Accelerating Pseudo-marginal Metropolis-Hastings Schemes for Stochastic Kinetic Models

Tom Lowe

Thesis submitted for the degree of
Doctor of Philosophy

School of Mathematics, Statistics & Physics
Newcastle University
Newcastle upon Tyne
United Kingdom

October 2022

Abstract

Stochastic kinetic models (SKMs) provide a natural framework for modelling continuous-time physical processes with inherent stochasticity. As such, they are frequently used to model interacting species populations in areas such as epidemiology, population ecology and systems biology. This thesis focuses on the challenging problem of performing fully Bayesian inference for the rate constants governing these models, using discrete-time observations of the species populations that may be incomplete and subject to measurement error. The SKM is often represented by either a Markov jump process (MJP) or an Itô diffusion process. In either case, the observed data likelihood is intractable, necessitating the use of computationally intensive techniques such as pseudo-marginal Metropolis-Hastings (PMMH). One prominent example of PMMH is particle Markov chain Monte Carlo (particle MCMC), whereby the observed data likelihood is unbiasedly estimated using a particle filter. Whilst powerful, such schemes are often impractical due to their large computational expense.

This thesis aims to increase the computational and statistical efficiency of these schemes using various techniques. Several of these techniques leverage a tractable surrogate model, the linear noise approximation (LNA), which can be derived directly from the MJP or the diffusion process. The LNA can be used in three ways: in the design of a gradient-based parameter proposal such as the Metropolis-adjusted Langevin algorithm (MALA); in the first stage of a delayed-acceptance step; and to construct an appropriate bridge construct within the particle filter. Further computational savings can be made if several of these techniques are used in tandem, as the equations governing the LNA need only be solved once for use in all three techniques. A further acceleration technique involves inducing positive correlation between successive likelihood estimates within the particle filter. A novel approach to MALA is also proposed, whereby the gradient is approximated to reduce the number of differential equations required to estimate it. The proposed acceleration techniques are then applied to several models utilising real-world and synthetic data, to compare their performance.

Acknowledgements

There are so many people who have helped me get to this point, and I'm very grateful to all of you. The challenge here is to adequately thank everyone without this turning into an entire thesis chapter. Here goes...

First thanks of course goes to my supervisors, Andrew Golightly and Colin Gillespie. In particular, without Andy's exemplary supervision I highly doubt I'd have made it to this point. From (sometimes repeatedly) explaining concepts to me until I was clear with them, to seemingly always knowing the best way to go about things and the incredible speed with which you'll reply to an email with any problem I have, to being endlessly friendly and interested during meetings, even when I was consistently late to them; I could write pages about the things I've got to thank you for, but suffice to say I couldn't have had a better supervisor, and Newcastle's loss is Durham's gain!

Thanks as well to my PhD office, from the Daves, Tom, Joe, and Matt who welcomed me into PhD life at the beginning, to Nathan and Bev for being friendly faces at the end (on the rare occasions that we were in the office at the same time!)

A huge thanks to Mum, Dad, Simon, Jackie, Dan, and Jamie, for all your help in getting me this far, whether it was financial help, moving all my things between houses, supportive phone calls or just chatting about music and football. Mum, you may never fully understand the concept behind why I was getting "paid to go to university" for my PhD, but let's all be glad I made it to the end before I had to start paying tuition myself.

To all of the maths lot who finished university and decided they weren't yet ready to let go of Newcastle: Alyssa, Becky, Joe, Stanno, James, Kenny, Magda, Dan, Marsden, Rachel, Martha, Frances, and Vic. Some of us are now eventually leaving the city, but we all made a lot of good memories, and for some of them we were even sober enough to actually commit them to memory. Special mention goes to Jack, for pioneering the PhD life for the rest of us, and for generally being the nicest man in the world; and to Towen and Cameron, for being so much better at doing a PhD than I was that you managed to help me with my own work many times, and for being great housemates for two years - it's just a shame that Cooking With Cameron didn't become the global success it could have been.

Outside of that core maths group, thanks to Dave, Dylan, Sarah and Naomi for accompanying me to gigs from London to Leeds to Newcastle (although they do always seem to involve either Frank Carter or Enter Shikari), and to Ryan and Amelia for surviving living with me during a pandemic, and for getting me into D&D (still haven't quite managed to get me into Taylor Swift unfortunately Amelia). Thanks to Clare, for helping motivate me, for your many suggestions on how to live a more competent lifestyle (some of which

I listened to more than others), and for so much more - hopefully one day you'll let me forget how long it took me to cook a stroganoff.

Finally, to Rowan, Kieran and Caz - where do I start. We've lived with each other, invented games, had countless wine and film nights, set each other ridiculous challenges, thrown cans at the TV, created the maddest seven course abomination imaginable, got weirdly obsessed with conversion rates, thrown away mattresses, relentlessly made fun of each other, concocted some very last minute costumes, and gone on many, MANY nights out. You've also helped me out enormously, whether that's proof reading my code, giving me advice when I've needed it, or even letting me stay in your spare room for a while. Thank you for all of it - you're good people. Even though you do bad things.

It's been a long journey through my PhD, but one that I feel like I've made the most of. Thank you to everyone (including all the many others I haven't had a chance to explicitly name), for either helping me get to the end, or for making the experience a pleasant one.

Declaration

Parts of this thesis have been previously submitted for publication by the author:

- Parts of Chapter 4, Chapter 5 and Chapter 6 have previously been submitted as: Lowe, T.E., Golightly, A. and Sherlock, C. 'Accelerating inference for stochastic kinetic models', *Computational Statistics and Data Analysis*. Under review. Available from `https://arxiv.org/pdf/2206.02644.pdf`.

- Parts of Chapter 5 have previously been published as: Golightly, A., Bradley, E., Lowe, T. and Gillespie, C.S. 'Correlated pseudo-marginal schemes for time-discretised stochastic kinetic models', *Computational Statistics and Data Analysis*. Published 2019.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A stochastic kinetic model (SKM) typically refers to a reaction network, an associated rate law and a probabilistic description of the reaction dynamics. Reactions occur continuously in time with a reaction occurrence resulting in a discrete change to the system state. A Markov jump process (MJP) provides a natural description of the time-course behaviour of the species involved in the reaction network. A concise introduction to SKMs can be found in Wilkinson (2018).

Whilst exact simulation of the MJP is straightforward (using for example the direct method of Gillespie (1976)), performing exact fully Bayesian inference is made problematic by the intractability of the observed data likelihood. Consequently, several approaches have been developed that make use of computationally intensive methods. These include the use of data augmentation (Boys & Giles, 2007; Boys *et al.*, 2008; Fuchs, 2013) together with Markov chain Monte Carlo (MCMC), reversible jump MCMC (Boys *et al.*, 2008; Wang *et al.*, 2010), population Monte Carlo (Koblents & Miguez, 2015), approximate Bayesian computation (ABC) (Minter & Retkute, 2019; Wu *et al.*, 2014), and particle MCMC (Andrieu *et al.*, 2010; Golightly & Wilkinson, 2011; Owen *et al.*, 2015). Such methods typically require many simulations of the jump process, prohibiting their use for SKMs with many reactions and species. Consequently, there has been much interest in the development of exact (simulation-based) inference schemes for cheap approximations of the MJP. In particular, approximations based on time discretisation do not require simulation of every reaction event, but rather update the state of the system in one go, after a particular time step (typically chosen by the practitioner). One such approach is the Poisson leap method of Gillespie (2001); another approach is to approximate the MJP with an Itô stochastic differential equation known as the chemical Langevin equation (CLE; Gillespie, 2000), and then discretise this using the Euler-Maruyama discretisation. The modelling framework arising from either the MJP, CLE, or Poisson leap method is fairly flexible, and thus has been used ubiquitously in areas such as epidemiology (O'Neill

& Roberts, 1999; Lin & Ludkovski, 2013; McKinley *et al.*, 2014), population ecology (Ferm *et al.*, 2008; Boys *et al.*, 2008; Gillespie & Golightly, 2010; Sun *et al.*, 2015) and systems biology (Wilkinson, 2009; Golightly & Wilkinson, 2015; Koblents & Miguez, 2015; Hey *et al.*, 2015; Owen *et al.*, 2015; Georgoulas *et al.*, 2017; Golightly *et al.*, 2019). However, even when working with a time discretisation of the MJP, inference remains far from straightforward. Typically, to maintain a desired level of accuracy, inference requires the imputation of the latent process at pre-specified inter-observation time points. Since the latent process at these time points cannot be integrated out analytically, the observed data likelihood remains intractable even under these approximate models. Thus, although typically more efficient than working with the MJP, inference remains computationally expensive.

1.1 Thesis aims

The aim of this thesis is the development of fully Bayesian inference schemes for these SKMs that are both computationally and statistically efficient, given discrete-time data that may be incomplete and subject to error. To circumvent the problem of intractable observed data likelihoods, much of this thesis focuses on pseudo-marginal MCMC, in particular particle MCMC schemes (pMCMC), for performing fully Bayesian inference, and improvements in computational efficiency over basic implementations of these schemes in several ways.

A special case of the auxiliary particle filter of Pitt & Shephard (1999) is used to (unbiasedly) estimate the observed data likelihood. As shown by Golightly & Wilkinson (2015), this is crucial in avoiding highly variable likelihood estimates in scenarios where intrinsic stochasticity outweighs the error in the observation process. Essentially, particles are propagated conditional on future observations by using a suitable bridge construct, that is, a tractable approximation of the intractable end-point conditioned process. Several bridge constructs are considered in this thesis, including the modified diffusion bridge (MDB) of Durham & Gallant (2002), and different implementations of the residual bridges of Whitaker *et al.* (2017*b*) (see also van der Meulen & Schauer, 2017), and each of these constructs has an effect on the computational and statistical efficiency of the resulting inference scheme.

We also make use of the recently proposed correlated pseudo-marginal algorithm (Deligiannidis *et al.*, 2018; Dahlin *et al.*, 2015), which introduces positive correlation between successive likelihood estimates in order to reduce the variance of the acceptance ratio. Our approach is to introduce correlation between the bridges generated by the particle filter at iteration i and those generated at iteration $i+1$. Tran *et al.* (2016) and Choppala *et al.* (2016) describe a similar approach, known as the blockwise pseudo-marginal

method, and apply it to a univariate diffusion process and a Lotka-Volterra reaction network, respectively. In the blockwise pseudo-marginal method, the observed data likelihood is calculated by averaging several 'blocks' of unbiased estimates (which can be computed in parallel). Correlation is then introduced by only updating the likelihood in a randomly chosen block.

In addition, we look to use tractable approximations of the likelihood to further accelerate the inference schemes. Approximations such as the linear noise approximation (LNA; Kurtz, 1970; Komorowski *et al.*, 2009; Fearnhead *et al.*, 2014) are computationally inexpensive, and whilst the approximations may not be sufficiently accurate to use as the inferential model, they can have several benefits. Firstly, they can be used to estimate not only the log-likelihood but also its gradient, which can then be used to perform the Metropolis-adjusted Langevin algorithm (MALA), an algorithm proposed by Roberts & Stramer (2002) as an 'intelligent' proposal mechanism derived from a discretised Langevin diffusion. In essence, gradient information from the LNA is used to push proposals from a Metropolis-Hastings (MH) scheme towards areas of high posterior density. This requires the solution of a system of ordinary differential equations (ODEs), which in general do not have analytic solutions. Tractable approximations of the likelihood can also be used in a delayed acceptance stage of an inference scheme. The idea of delayed acceptance was proposed by Christen & Fox (2005), and used by Golightly *et al.* (2015) inside a pseudo-marginal scheme. The basic principle is to propose a set of parameter values, then use an initial MH step with an acceptance probability based on an approximate model. Proposed parameter values which are accepted at this initial screening stage proceed to another MH step with the marginal posterior as the target density. Hence, computationally expensive calculations of the observed data likelihood estimate are avoided for parameter proposals that are likely to be rejected.

The above acceleration techniques typically leverage the tractability of a common surrogate model such as the LNA, and as such further computational savings can be made by combining techniques in such a way that avoids unnecessary repeated solving of the ODE system that governs the surrogate model. Information from one solution of the ODE system can be used in three ways: firstly, in the design of a MALA proposal; secondly, to construct an appropriate bridge construct for use in the bootstrap particle filter; thirdly, in the first stage of a delayed acceptance step. This thesis presents a unified framework for applying a pMCMC algorithm, potentially with correlated bridges, a MALA proposal mechanism, and a delayed acceptance step, to a general class of time discretised stochastic kinetic models, that additionally allows a flexible observation regime. In particular, we consider incomplete observation of the model components as well as Gaussian measurement error. This framework can be applied whether the MJP, CLE, or Poisson leap method is used as the inferential model. The methodology is applied to several examples arising

in systems biology and epidemiology, using both real and synthetic data, including a birth-death model, immigration death model, the Lotka-Volterra predator-prey model, an autoregulatory network, an SIR epidemic model and a model for aphid populations. The remainder of this thesis is organised as follows.

1.2 Thesis layout

In Chapter 2, we review the necessary background material on Monte Carlo methods for intractable problems, including importance sampling and weighted resampling. We consider Markov chain Monte Carlo methods for generating (dependent) samples from target distributions known up to proportionality, and pseudo-marginal schemes for scenarios where the likelihood function involves an intractable integral. The use of correlation within a pseudo-marginal scheme is also considered here.

Chapter 3 introduces stochastic kinetic models. Starting with a pseudo-reaction network, the Markov jump process representation of species dynamics is considered. Updating reactions in discrete time steps leads to the Poisson leap approximation, and further ignoring state-space discreteness leads to a stochastic differential equation (SDE) approximation known as the chemical Langevin equation (CLE). This can be further approximated by a Gaussian process known as the linear noise approximation (LNA).

In Chapter 4, we consider the problem of performing fully Bayesian inference for the parameters and any unobserved component in a tractable SKM (namely the LNA). We describe a computationally efficient method for evaluating the observed data likelihood via a forward filter and outline a scheme utilising the Metropolis adjusted Langevin algorithm (MALA), which we illustrate with two synthetic data applications: a simple birth-death model, and the Lotka-Volterra predator-prey network.

In Chapter 5, we consider the challenging problem of performing fully Bayesian inference in the context of an intractable SKM, such as the MJP or CLE. We describe the use of particle filters within a particle MCMC scheme and consider the use of correlation to accelerate inference in this setting. Several applications of these techniques with both real and synthetic data are considered: an immigration-death model, the Lotka-Volterra system, an autoregulatory network, and an SIR epidemic model using real data from an influenza outbreak in a boarding school.

Chapter 6 gives a unified framework for using a tractable surrogate, namely the LNA, to accelerate inference for intractable SKMs. In particular, the tractability of the surrogate is exploited in a delayed acceptance step, to obtain an approximate log-likelihood gradient for use in MALA, and to drive a particular bridge construct within a particle filter. Moreover, we consider a strategy for only solving the system of ordinary differential equations (ODEs) governing the LNA solution once per pMCMC iteration. This is compared and contrasted

with a method that re-solves the ODE system for each particle in a given particle MCMC iteration. These techniques are considered within several synthetic and real-world data applications: a synthetic data model of aphid population dynamics, a real-world epidemic example studying the outbreak of plague in Eyam, and finally revisiting the Lotka-Volterra system.

In chapter 7, we summarise the findings of this thesis and discuss several avenues for future research.

Chapter 2

Monte Carlo methods and stochastic differential equations

The bulk of this thesis concerns Bayesian inference where a critical component of this inference, namely the posterior density, is intractable. To proceed, we employ Monte Carlo methods, which use repeated sampling of random variables to approximate a desired quantity, such as an expectation, probability, or posterior density. This chapter will review standard Monte Carlo methods such as importance sampling and weighted resampling, before giving a brief overview of Markov chain Monte Carlo and some standard techniques within this field. Some models considered in this thesis utilise stochastic differential equations, and so this chapter shall finish by providing a brief introduction to stochastic differential equations and Itô calculus.

2.1 Monte Carlo integration and importance sampling

Consider an integral of the form

$$I = \int_D \phi(\theta) d\theta,$$

that is, the integral of a function ϕ of a quantity θ over all possible values of θ within a domain D. I may be intractable, but if the integrand can be re-written in the form

$$\phi(\theta) = \tilde{\phi}(\theta) f(\theta)$$

for some density function $f(\cdot)$ with the same domain D and $\int_D f(\theta) d\theta = 1$, then the integral takes the form of an expectation

$$I = \int_D \tilde{\phi}(\theta) f(\theta) d\theta = E_f \left[\tilde{\phi}(\Theta) \right],$$

where Θ is a random variable with probability density function (PDF) $f(\theta)$. If we can generate N independent realisations $\theta^{(1)}, \ldots, \theta^{(N)}$ from $f(\cdot)$, then we can evaluate $\tilde{\phi}(\theta^{(i)})$ for $i = 1, \ldots, N$, and use the arithmetic mean of these as an estimator for I, that is

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} \tilde{\phi}(\theta^{(i)}).$$

Estimating integrals in this way is known as Monte Carlo integration. In the simplest case, D is just an interval on the real line $[a, b]$, and Θ has a uniform distribution within this interval,

$$f(\theta) = \frac{1}{b-a}, \quad a \leq \theta \leq b.$$

The estimator $\hat{I}$ has some desirable properties in that it is an unbiased and consistent estimator of I, provided the variance of $\tilde{\phi}(\Theta)$ is finite. Unbiasedness can be seen by checking the expectation of the estimator

$$E\left[\hat{I}\right] = \frac{1}{N} \sum_{i=1}^{N} E\left[\tilde{\phi}(\Theta^{(i)})\right] = E\left[\tilde{\phi}(\Theta)\right] = I.$$

Assuming a finite variance of $\tilde{\phi}(\Theta)$, consistency can be seen by noting the variance of the estimator

$$Var\left[\hat{I}\right] = Var\left[\frac{1}{N} \sum_{i=1}^{N} \tilde{\phi}(\Theta^{(i)})\right] = \frac{1}{N} Var\left[\tilde{\phi}(\Theta^{(i)})\right],$$

which will tend to 0 as N tends to infinity. Thus, as N increases the estimator converges to the true value of I.

There are cases where it is prohibitively difficult to write the integrand of I in the form $\tilde{\phi}(\theta) f(\theta)$ for an easily sampled density $f(\cdot)$, or where estimators using Monte Carlo integration have a very high variance for computationally practical values of N. In these cases, if we can easily sample from another density $g(\cdot)$, with $\tilde{\phi}(\cdot) f(\cdot) > 0 \implies g(\cdot) > 0$, then we can multiply and divide our integrand by this density and re-express the resulting integral as the sum of integrals over different domains

$$\begin{aligned} I &= \int_D \frac{\tilde{\phi}(\theta) f(\theta)}{g(\theta)} g(\theta) d\theta \\ &= \int_G \frac{\tilde{\phi}(\theta) f(\theta)}{g(\theta)} g(\theta) d\theta + \int_{D \setminus G} \frac{\tilde{\phi}(\theta) f(\theta)}{g(\theta)} g(\theta) d\theta - \int_{G \setminus D} \frac{\tilde{\phi}(\theta) f(\theta)}{g(\theta)} g(\theta) d\theta. \end{aligned}$$

Note now that for $\theta \notin D$ we have $f(\theta) = 0$, and for $\theta \in D \cap G^{\complement}$, we have that $f(\theta) > 0$ and $g(\theta) = 0$, meaning that we must have $\tilde{\phi}(\theta) = 0$ for our assumption on $g(\cdot)$ to hold. Therefore, the latter two integrals above are 0, and the remaining integral can now be

written as an expectation with respect to $g(\cdot)$

$$I = \int_G \frac{\tilde{\phi}(\theta)f(\theta)}{g(\theta)} g(\theta)d\theta = E_g\left[\frac{\tilde{\phi}(\Theta)f(\Theta)}{g(\Theta)}\right].$$

Thus, sampling N realisations from $g(\cdot)$ and following an analogous process to Monte Carlo integration, we can construct an estimate for I as

$$\hat{I}_{IS} = \frac{1}{N}\sum_{i=1}^{N} \frac{\tilde{\phi}(\theta^{(i)})f(\theta^{(i)})}{g(\theta^{(i)})}.$$

This is known as importance sampling. It can be shown using the same process as for standard Monte Carlo integration that importance sampling leads to unbiased and consistent estimators for I. Moreover, the variance of the estimator is given by

$$\frac{1}{N}Var\left[\frac{\tilde{\phi}(\Theta^{(i)})f(\Theta^{(i)})}{g(\Theta^{(i)})}\right].$$

Thus, the variance can be reduced by increasing N, similar to Monte Carlo integration, but can additionally be reduced by choosing $g(\cdot)$, known as the importance density, such that $Var\left[\frac{\tilde{\phi}(\cdot)f(\cdot)}{g(\cdot)}\right]$ is small.

Sometimes we may only know $f(\cdot)$ up to a constant of proportionality k. In these cases, we can use the self-normalised estimator

$$\hat{I}_{SNIS} = \frac{\frac{1}{N}\sum_{i=1}^{N} \tilde{\phi}(\theta^{(i)})f(\theta^{(i)})/g(\theta^{(i)})}{\sum_{i=1}^{N} f(\theta^{(i)})/g(\theta^{(i)})}.$$

If we replace $f(\cdot)$ in $\hat{I}_{SNIS}$ with $\tilde{f}(\cdot) = kf(\cdot)$, then we see that k cancels in the numerator and denominator to leave $f(\cdot)$. This is a valuable property in Bayesian statistics, where often a target density of interest is known only up to proportionality. Unfortunately, the resulting estimator is no longer unbiased. However, it is still consistent, as when $N \to \infty$

Algorithm 1 Weighted resampling

1. Generate N realisations, $\theta^{(1)}, \ldots, \theta^{(N)}$, from some proposal density $g(\cdot)$ with the same domain as $f(\cdot)$.
2. Assign a normalised weight to each realisation
$$w^{(j)} = \frac{f(\theta^{(j)})/g(\theta^{(j)})}{\sum_{i=1}^{N} f(\theta^{(i)})/g(\theta^{(i)})}, \quad j = 1, \ldots, N.$$
3. Resample M times with replacement from $\{\theta^{(1)}, \ldots, \theta^{(N)}\}$, with probabilities according to the weights calculated in step 2.

we have

$$\begin{aligned}
\hat{I}_{SNIS} &\to \frac{E_g\left[\tilde{\phi}(\Theta) f(\Theta)/g(\Theta)\right]}{E_g\left[f(\Theta)/g(\Theta)\right]} \\
&= \frac{\int_D \frac{\tilde{\phi}(\theta) f(\theta)}{g(\theta)} g(\theta) d\theta}{\int_D \frac{f(\theta)}{g(\theta)} g(\theta) d\theta} \\
&= \frac{\int_D \tilde{\phi}(\theta) f(\theta) d\theta}{\int_D f(\theta) d\theta} \\
&= \frac{I}{1} = I,
\end{aligned}$$

and so the estimator converges to the true value of the integral.

2.2 Weighted resampling

Weighted resampling is a method used to generate realisations from a continuous distribution with density $f(\cdot)$ that may otherwise be difficult to sample from. The principle is to sample from one distribution that is easier to generate realisations from, and then correct that value using a weighting and resampling technique to ensure that the resulting realisations are from the desired distribution. The steps to generate realisations from $f(\cdot)$ using weighted resampling are outlined in algorithm 1. The result is a sample $\theta^{(1)}, \ldots, \theta^{(M)}$ which is approximately distributed according to $f(\cdot)$. Note that typically $M = N$ in practice, and so we shall use N for both the initial realisations and the resampled realisations for the remainder of this section. As with self-normalised importance sampling in Section 2.1, this algorithm can be applied even if the target $f(\cdot)$ is only known up to proportionality, making it another useful tool in Bayesian statistics.

Weighted resampling approximates a continuous distribution with a discrete one. How-

ever, as the number of samples N tends to infinity, the algorithm converges to sample from the exact target distribution. To show this, consider for simplicity the univariate case, with a random variable Θ. Let $F(\cdot)$ denote the cumulative distribution function (CDF) under the target density $f(\cdot)$, and let $\tilde{F}(\cdot)$ be the CDF of the distribution generated by the algorithm. $\tilde{F}(\theta^*)$ is the probability that a sample from the set $\{\theta^{(1)}, \ldots, \theta^{(N)}\}$ is less than or equal to some new value θ^* and is given by the sum of all weights for which the corresponding sample is less than or equal to θ^*, that is

$$\tilde{F}(\theta^*) = \sum_{j=1}^{N} w^{(j)} \mathcal{I}(\theta^{(j)} \leq \theta^*) = \frac{\sum_{j=1}^{N} f(\theta^{(j)})/g(\theta^{(j)}) \mathcal{I}(\theta^{(j)} \leq \theta^*)}{\sum_{i=1}^{N} f(\theta^{(i)})/g(\theta^{(i)})},$$

where $\mathcal{I}(\theta^{(j)} \leq \theta^*)$ is an indicator variable that takes a value of 1 if $\theta^{(j)} \leq \theta^*$ and 0 otherwise. Note that if we multiply the numerator and denominator of our CDF by $\frac{1}{N}$, then both are Monte Carlo integration estimates for the expectations of $\mathcal{I}(\Theta \leq \theta^*) f(\Theta)/g(\Theta)$ and $f(\Theta)/g(\Theta)$ respectively. Thus, as $N \to \infty$, we have

$$\begin{aligned}
\tilde{F}(\theta^*) &\to \frac{\int_D [f(\theta)/g(\theta)] \mathcal{I}(\theta \leq \theta^*) g(\theta) d\theta}{\int_D [f(\theta)/g(\theta)] g(\theta) d\theta} \\
&= \frac{\int_D f(\theta) \mathcal{I}(\theta \leq \theta^*) d\theta}{\int_D f(\theta) d\theta} \\
&= F(\theta^*).
\end{aligned}$$

2.3 Markov chain Monte Carlo

Consider a target density $\pi(\theta)$, with parameter vector $\theta = (\theta_1, \ldots, \theta_p)' \in \mathcal{S}$, for some state-space $\mathcal{S} \subseteq \mathbb{R}^p$, where the $'$ denotes the transpose of the vector. Markov chain Monte Carlo (MCMC) is a technique used to simulate from distributions whose densities may only be known up to proportionality, by simulating from a specially constructed continuous Markov chain with the target density as its stationary distribution. Providing the chain has converged (at least approximately), any value sampled will be (approximately) from our target density $\pi(\theta)$. In general, a chain will not converge exactly to its stationary distribution in finite time. However, in practice, a commonly used technique is to run the chain for a long period of time, then discard the initial portion of the chain as "burn-in", and assume that the chain has approximately converged after this point. We can therefore use these samples to evaluate integrals and perform inference. However, note that samples from a Markov chain will not be independent from one another. In the following sections we discuss some key properties of continuous Markov chains, and then detail some MCMC algorithms used to construct such Markov chains with the required stationary distribution.

2.3.1 Continuous Markov chains

Consider a continuous state-space, discrete-time Markov chain Θ_n, with state-space $\mathcal{S}$. We can define the conditional cumulative distribution function of the chain as

$$P(\theta|\phi) = \mathbb{P}(\Theta_{n+1} \leq \theta|\Theta_n = \phi).$$

It is often more convenient to work instead with the transition density $p(\theta|\phi)$, where

$$p(\theta|\phi) = \frac{\partial}{\partial\theta}P(\theta|\phi).$$

Now, denote the stationary distribution of this Markov chain by $\pi(\cdot)$. A stationary density satisfies

$$\pi(\phi) = \int_{\mathcal{S}} \pi(\theta)p(\phi|\theta)d\theta. \tag{2.1}$$

Determining whether a density is a stationary density of the Markov chain is often done by checking whether it satisfies the detailed balance equation

$$\pi(\phi)p(\theta|\phi) = \pi(\theta)p(\phi|\theta), \quad \forall\, \phi, \theta \in \mathcal{S}. \tag{2.2}$$

To see that (2.2) implies (2.1), we can integrate both sides over $\mathcal{S}$ with respect to θ to give

$$\begin{aligned} & \int_{\mathcal{S}} \pi(\phi)p(\theta|\phi)d\theta = \int_{\mathcal{S}} \pi(\theta)p(\phi|\theta)d\theta \\ \implies \quad & \pi(\phi)\int_{\mathcal{S}} p(\theta|\phi)d\theta = \int_{\mathcal{S}} \pi(\theta)p(\phi|\theta)d\theta \\ \implies \quad & \pi(\phi) = \int_{\mathcal{S}} \pi(\theta)p(\phi|\theta)d\theta. \end{aligned}$$

A Markov chain is known as π-*irreducible* if, for any initial state Θ_0, the chain has a positive probability of entering any set $A \subseteq \mathcal{S}$ for which $\pi(\cdot)$ has a positive probability, at some point in the future. Furthermore, if there are portions of the state space that a Markov chain can only visit at certain regularly spaced times, then the chain is known as *periodic*, otherwise the chain is known as *aperiodic*. For rigorous definitions of these terms, see for example Tierney (1994) or Roberts & Rosenthal (2004).

If a Markov chain is π-irreducible and aperiodic, and a proper stationary distribution $\pi(\cdot)$ exists (proper in the sense that $\pi(\cdot)$ integrates to 1), then it can be shown that no matter the initial state Θ_0, the chain will converge to this stationary distribution as $n \longrightarrow \infty$. A proof of this statement can be found in Roberts & Rosenthal (2004).

Algorithm 2 The Metropolis-Hastings algorithm

1. Initialise the iteration counter to $i = 1$, and initialise the chain at $\theta^{(0)}$ from somewhere in the domain of $\pi(\theta)$.
2. Propose a new value θ^* using the proposal density $q(\theta^*|\theta^{(i-1)})$.
3. Evaluate the acceptance probability $\alpha(\theta^*|\theta^{(i-1)})$ of the proposed move using (2.3).
4. Set $\theta^{(i)} = \theta^*$ with probability $\alpha(\theta^*|\theta^{(i-1)})$, otherwise set $\theta^{(i)} = \theta^{(i-1)}$.
5. Set $i = i + 1$ and return to step 2.

2.3.2 Metropolis-Hastings algorithm

One of the fundamental algorithms in the field of MCMC is the Metropolis-Hastings algorithm. Metropolis *et al.* (1953) introduced the concept, which was then generalised by Hastings (1970) (see e.g. Gamerman & Lopes, 2006, for a more recent review). Central to the Metropolis-Hastings algorithm is the idea of a proposal density $q(\cdot|\cdot)$. The proposal density does not need to have $\pi(\theta)$ as its stationary distribution, and it can be advantageous to use a proposal distribution which is easy to simulate from. The Metropolis-Hastings algorithm is outlined in algorithm 2. At each stage a new value θ^* is generated from the proposal distribution. This value is then either accepted, in which case the chain moves to the proposed value, or rejected, in which case the chain will remain at its current position. The probability of a move from θ to θ^* being accepted is given by the acceptance probability

$$\alpha(\theta^*|\theta) = \min\left\{1, \frac{\pi(\theta^*)q(\theta|\theta^*)}{\pi(\theta)q(\theta^*|\theta)}\right\}. \tag{2.3}$$

As the target density $\pi(\theta)$ only enters the acceptance probability as a ratio, this algorithm can be used when the target is known only up to proportionality. This is commonly the case in Bayesian inference, where these algorithms are used frequently, and are often referred to as Metropolis-Hastings schemes, or M-H schemes.

A key user-defined element of the Metropolis-Hastings algorithm is the choice of proposal density $q(\cdot|\cdot)$. A good choice of proposal density is one that leads to a chain that converges rapidly towards its stationary distribution, and traverses efficiently around the parameter space, known as a well-mixing chain. Some commonly used classes of proposal distribution are now considered.

Symmetric proposals

If the proposal distribution is symmetric, that is

$$q(\theta^*|\theta) = q(\theta|\theta^*), \quad \forall \theta, \theta^* \in \mathcal{S},$$

then the acceptance probability simplifies to become

$$\alpha(\theta^*|\theta) = \min\left\{1, \frac{\pi(\theta^*)}{\pi(\theta)}\right\}.$$

This means that if the proposal density is symmetric then the acceptance probability does not depend on the proposal density.

Random walk Metropolis

The proposal density $q(\cdot|\cdot)$ can take the form

$$\theta^* = \theta + \omega,$$

where ω, known as an innovation, is a $d \times 1$ random vector that is independent from the state of the chain, and d is the dimension of the chain. Typically, ω has a Gaussian distribution centred around zero. Metropolis-Hastings schemes with this form of symmetric proposal mechanism are known as random walk Metropolis (RWM) schemes.

The variance of the innovations ω is a tuning parameter chosen by the user, and affects the mixing of the Markov chain. If the variance is too small, the chain will accept many proposed values but will not explore the parameter space well as each move will be small. Conversely, if the variance is too high, any moves will be large, but the chain will remain in place for long periods of time as few proposed values will be accepted. Subject to some constraints on the target distribution, for large values of d the asymptotic optimal acceptance rate of the chain is 0.234, and the optimal choice of the variance is

$$\frac{2.38^2}{d}\,\mathrm{Var}(\Theta).$$

For more details on this see e.g. Roberts & Rosenthal (2001); Sherlock *et al.* (2015). This optimal acceptance rate does not need to be reached precisely in practice, particularly for small d - an acceptance rate between 0.1 and 0.4 is often seen as acceptable (see e.g. Schmon & Gagnon, 2021). In general, $\mathrm{Var}(\Theta)$ will not be available, and so typically a pilot run with a somewhat arbitrary choice of variance is used in order to obtain an estimate of $\mathrm{Var}(\Theta)$.

Independence samplers

If the proposal density takes the form $q(\theta^*|\theta) = g(\theta^*)$ for some density $g(\cdot)$, independently of the current value θ, then the chain is known as an independence sampler. This leads to an acceptance probability of

$$\alpha(\theta^*|\theta) = \min\left\{1, \frac{\pi(\theta^*)}{g(\theta^*)} \times \frac{g(\theta)}{\pi(\theta)}\right\}.$$

For an independence sampler, the optimal acceptance probability is 1, and so increasing the acceptance probability as much as possible is desirable. This can be achieved by choosing $g(\cdot)$ to be as close to $\pi(\cdot)$ as possible.

2.3.3 Validity of Metropolis-Hastings

For the M-H algorithm to be valid, the target density $\pi(\theta)$ must be a stationary distribution of the Markov chain, and the chain must converge to this distribution. Recall from Section 2.3.1 that a density is stationary for a Markov chain if it satisfies detailed balance, and that a chain will converge towards its stationary distribution, assuming one exists, if it is aperiodic and π-irreducible. We show here that the target density for M-H satisfies detailed balance, and refer the reader to Tierney (1994) for discussion on the convergence of M-H schemes, and to Meyn *et al.* (2009) for a more in-depth analysis of convergence of continuous state-space Markov chains.

To see that $\pi(\theta)$ satisfies detailed balance, we must first obtain the transition density. When the chain moves, this takes the form

$$p(\theta^*|\theta) = q(\theta^*|\theta)\alpha(\theta^*|\theta), \quad \theta^* \neq \theta.$$

When the proposed move is rejected, the chain remains at θ, which happens with a probability of $\omega(\theta)$, where

$$\omega(\theta) = 1 - \int_{\mathcal{S}} q(\theta^*|\theta)\alpha(\theta^*|\theta)d\theta^*,$$

or 1 minus the marginal probability of the chain moving. Thus, the transition density is

$$p(\theta^*|\theta) = q(\theta^*|\theta)\alpha(\theta^*|\theta) + \omega(\theta)\delta(\theta^* - \theta),$$

where $\delta(\theta^* - \theta)$ is the Dirac delta function, equal to 1 if $\theta^* = \theta$ and 0 otherwise. Note that this function is trivially symmetric in θ^* and θ, that is $\delta(\theta^* - \theta) = \delta(\theta - \theta^*)$. Moreover, any function multiplied by this function is also trivially symmetric in θ^* and θ.

We can now check detailed balance as follows

$$\begin{aligned}
\pi(\theta)p(\theta^*|\theta) &= \pi(\theta)q(\theta^*|\theta)\alpha(\theta^*|\theta) + \pi(\theta)\omega(\theta)\delta(\theta^* - \theta) \\
&= \pi(\theta)q(\theta^*|\theta)\min\left\{1, \frac{\pi(\theta^*)q(\theta|\theta^*)}{\pi(\theta)q(\theta^*|\theta)}\right\} + \pi(\theta)\omega(\theta)\delta(\theta^* - \theta) \\
&= \min\left\{\pi(\theta)q(\theta^*|\theta), \pi(\theta^*)q(\theta|\theta^*)\right\} + \pi(\theta)\omega(\theta)\delta(\theta^* - \theta) \\
&= \min\left\{\pi(\theta^*)q(\theta|\theta^*), \pi(\theta)q(\theta^*|\theta)\right\} + \pi(\theta^*)\omega(\theta^*)\delta(\theta - \theta^*) \\
&= \pi(\theta^*)q(\theta|\theta^*)\alpha(\theta|\theta^*) + \pi(\theta^*)\omega(\theta^*)\delta(\theta - \theta^*) \\
&= \pi(\theta^*)p(\theta|\theta^*).
\end{aligned}$$

Thus, detailed balance is satisfied.

2.4 Pseudo-marginal Metropolis-Hastings (PMMH)

Sometimes, the target of interest $\pi(\theta)$ may not be known, even up to proportionality. This can often be the case if the target is an integral of the form discussed at the beginning of this chapter. If a non-negative, unbiased estimate of $\pi(\theta)$ can be obtained (for instance, using Monte Carlo integration or importance sampling), then a technique known as pseudo-marginal Metropolis-Hastings (PMMH) may be employed to sample from $\pi(\theta)$ exactly. Let $U \sim g(u)$ denote the auxiliary random variables (such as the proposal distribution in an importance sampler) used to generate an estimator $\hat{\pi}_U(\theta)$ of the target density. The corresponding estimate will then be denoted $\hat{\pi}_u(\theta)$. The PMMH algorithm is an M-H algorithm targeting the joint density

$$\hat{\pi}(\theta, u) \propto \hat{\pi}_u(\theta)g(u).$$

For a joint proposal density $q(\theta^*|\theta)g(u^*)$, the acceptance probability is

$$\begin{aligned}
\alpha\{(\theta^*, u^*)|(\theta, u)\} &= \min\left\{1, \frac{\hat{\pi}(\theta^*, u^*)}{\hat{\pi}(\theta, u)} \times \frac{q(\theta|\theta^*)g(u)}{q(\theta^*|\theta)g(u^*)}\right\} \\
&= \min\left\{1, \frac{\hat{\pi}_{u^*}(\theta^*)g(u^*)}{\hat{\pi}_u(\theta)g(u)} \times \frac{q(\theta|\theta^*)g(u)}{q(\theta^*|\theta)g(u^*)}\right\} \\
&= \min\left\{1, \frac{\hat{\pi}_{u^*}(\theta^*)}{\hat{\pi}_u(\theta)} \times \frac{q(\theta|\theta^*)}{q(\theta^*|\theta)}\right\}, \qquad (2.4)
\end{aligned}$$

and therefore the density associated with the auxiliary variables need not be evaluated. Comparing this acceptance probability with that found in 2.3, we see that it is of the same form, but with $\pi(\theta)$ and $\pi(\theta^*)$ replaced by their estimates $\hat{\pi}_u(\theta)$ and $\hat{\pi}_{u^*}(\theta^*)$. If the estimator of $\pi(\cdot)$ is non-negative, and unbiased up to a multiplicative constant so that $E_U[\pi_U(\theta)] \propto \pi(\theta)$, then the PMMH algorithm will exactly target the density of interest

$\pi(\theta)$ when the auxiliary variables are marginalised out of the joint density, as

$$\begin{aligned}\int_S \hat{\pi}(\theta, u)du &= \int_S \hat{\pi}_u(\theta)g(u)du \\ &= E_U[\hat{\pi}_U(\theta)] \\ &\propto \pi(\theta). \end{aligned} \tag{2.5}$$

2.4.1 PMMH illustrative example

Consider a standard Normal target density $\pi(\theta) \propto \exp\left\{-\theta^2/2\right\}$. Clearly, this density is readily available to sample from directly, but to illustrate the approach discussed in Section 2.4 we can "estimate" this target density by generating samples directly and then multiplying these samples by a random variable U with an expected value of 1. Thus the estimator for $\pi(\theta)$ is $\hat{\pi}_U(\theta) = \pi(\theta)U$, and the joint density is $\hat{\pi}(\theta, u) \propto \pi(\theta)ug(u)$. For this example, we will take $U \sim Ga(a, a)$. Thus $E[U] = 1$, $Var[U] = 1/a$, and so it is easy to verify that the estimator is unbiased as

$$E_U[\hat{\pi}_U(\theta)] = \pi(\theta)E[U] = \pi(\theta).$$

This unbiasedness means that an M-H scheme targeting $\hat{\pi}(\theta, u)$ will marginally target $\pi(\theta)$ by (2.5).

Figure 2.1 shows the output of a PMMH scheme with 10^4 iterations, a Gaussian random walk proposal with a variance of $\sigma^2 = 1$, an initial value of $\theta^{(0)} = 1$ and varying values of a. As we can see, in each case the scheme is targeting the correct density, but with varying levels of success. This is because as a decreases from 1 to 0.1 to 0.01, the variance of the estimator increases from 1 to 10 to 100. Having an estimator with a high variance can lead to significant over-estimates or under-estimates of $\pi(\theta)$. Significant under-estimates are likely to lead to a small numerator in the acceptance probability and therefore be rejected, and so should not have much of an effect. However, significant over-estimates are likely to lead to a large numerator and therefore acceptance, at which point the estimate will appear in the denominator of the acceptance probability in future iterations, leading to a small acceptance probability and many consecutive rejections, until a similarly large or larger value of $\pi(\theta)$ is generated, either by another over-estimate or by a proposal into an area of much higher density. This "sticky" behaviour affects the mixing of the chain, as can be seen in the trace plots in Figure 2.1.

A useful diagnostic check and measure of the statistical efficiency of the chain is the *effective sample size* (ESS), which is the equivalent number of samples if each realisation

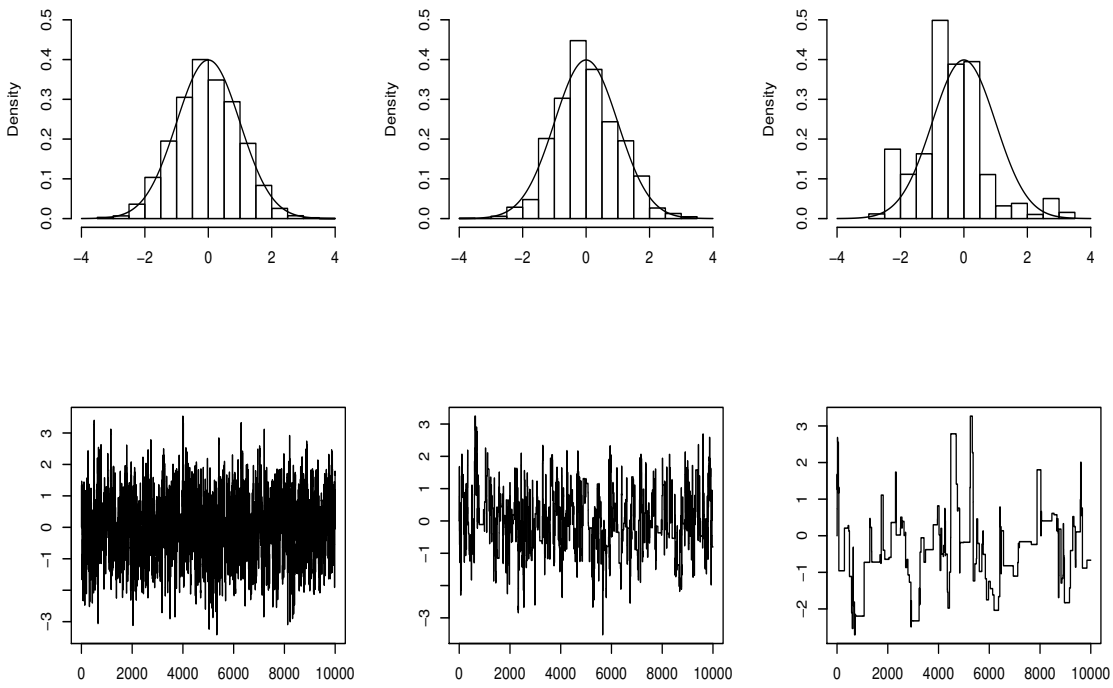

Figure 2.1: Histograms with overlaid target density, and trace plots of samples of θ from output of a PMMH scheme with 10^4 iterations and an initial value of $\theta^{(0)} = 0$. Left panels: $a = 1$. Middle panels: $a = 0.1$. Right panels: $a = 0.01$.

of the chain was completely independent from every other. It is given by the formula

$$ESS = \frac{n_{iters}}{1 + \sum_{k=1}^{\infty} \psi(k)},$$

where n_{iters} is the number of realisations of the distribution (also the number of iterations in the MCMC scheme), and $\psi(k)$ is the lag-k autocorrelation. The effective sample sizes for each chain can be found using the R package `coda` (Plummer *et al.*, 2006), and for $a = 1, 0.1, 0.01$ these are $826, 202$, and 29 respectively, to the nearest integer. As the ESS increases, the distribution of the realisations converges closer towards the target density. Thus, a high ESS is desirable, which further highlights the benefits of having an estimator with a low variance.

2.5 Correlated pseudo-marginal Metropolis-Hastings (CPMMH)

As seen in Section 2.4.1, a low variance estimator can improve effective sample size and reduce "sticky" behaviour in MCMC chains. One method of reducing the variance of an estimator in PMMH is by inducing correlation between successive estimates. This leads to a technique known as correlated pseudo-marginal Metropolis-Hastings (CPMMH) (Deligiannidis *et al.*, 2018; Dahlin *et al.*, 2015). The proposal density is not restricted to using $g(u^*)$, and so we may use a proposal density $q(\theta^*|\theta)K(u^*|u)$ that induces correlation between u and u^*, which will in turn induce correlation between $\hat{\pi}_u(\theta)$ and $\hat{\pi}_{u^*}(\theta^*)$. We must choose $K(\cdot,\cdot)$ such that it satisfies the detailed balance equation

$$g(u)K(u^*|u) = g(u^*)K(u|u^*).$$

One such choice of $g(\cdot)$ and $K(\cdot|\cdot)$ that satisfies these conditions is a standard Gaussian density and a Crank-Nicolson proposal density (Cotter *et al.*, 2013), that is

$$g(u) = N(u; 0, I_u), \quad K(u^*|u) = N(u^*; \rho u, (1-\rho^2)I_u), \tag{2.6}$$

where I_u is the identity matrix with dimension equal to the number of elements in u, and $\rho \in (-1, 1)$ is a tuning parameter controlling the correlation between successive values of u. To show that the detailed balance equation is satisfied, consider the densities written out in full

$$g(u)K(u^*|u) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-u^2}{2}\right) \times \frac{1}{\sqrt{2\pi(1-\rho^2)}} \exp\left(-\frac{(u^* - \rho u)^2}{2(1-\rho^2)}\right).$$

Collecting terms and multiplying out brackets gives

$$\begin{aligned} g(u)K(u^*|u) &= \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{u^2 - u^2\rho^2 + (u^*)^2 - 2\rho u u^* + u^2\rho^2}{2(1-\rho)^2}\right) \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{u^2 - 2\rho u u^* + (u^*)^2}{2(1-\rho)^2}\right). \end{aligned} \tag{2.7}$$

By completing the square we see that

$$u^2 - 2\rho u u^* = (u - \rho u^*)^2 - (u^*)^2\rho^2,$$

and substituting this into (2.7) gives

$$\begin{aligned} g(u)K(u^*|u) &= \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{(u-\rho u^*)^2 + (u^*)^2 - (u^*)^2\rho^2}{2(1-\rho)^2}\right) \\ &= \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{(u-\rho u^*)^2 + (u^*)^2(1-\rho^2)}{2(1-\rho)^2}\right) \\ &= \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-(u^*)^2}{2}\right) \times \frac{1}{\sqrt{2\pi(1-\rho^2)}} \exp\left(-\frac{(u-\rho u^*)^2}{2(1-\rho^2)}\right) \\ &= g(u^*)K(u|u^*), \end{aligned} \tag{2.8}$$

and so detailed balance is satisfied. In practice ρ is chosen to be close to 1 to induce strong positive correlation between u and u^*. However, if ρ is too close to 1 then this will negatively impact the mixing of the chain, resulting in long term dependence in the θ values. One approach is to choose ρ so that the ESS of the auxiliary chain for u is similar to the ESS of the parameter chain for θ. Taking $\rho = 0$ gives the special case that $K(u^*|u) = g(u)$, which is equivalent to uncorrelated PMMH.

The acceptance probability of the CPMMH algorithm is identical to 2.4. The rationale behind correlating the innovations is that if (θ, U) and (θ^*, U^*) are sufficiently close together, then it is expected that the ratio of the estimators in 2.4 will have a reduced variance (relative to if (θ, U) and (θ^*, U^*) were independent). If we consider the variance of the log-ratio of the estimators, we have

$$\begin{aligned} \mathrm{Var}\left(\log\left(\frac{\hat{\pi}_{u^*}(\theta^*)}{\hat{\pi}_u(\theta)}\right)\right) &= \mathrm{Var}(\log(\hat{\pi}_{u^*}(\theta^*))) + \mathrm{Var}(\log(\hat{\pi}_u(\theta))) \\ &\quad - 2\ \mathrm{Cov}(\log(\hat{\pi}_{u^*}(\theta^*)), \log(\hat{\pi}_u(\theta))), \end{aligned}$$

which will be smaller if the estimators are positively correlated than if they are independent. Deligiannidis *et al.* (2018) consider the asymptotic properties of the error of the (log)likelihood ratio for both PMMH and CPMMH and found that, under certain conditions, the variance of this ratio is lower for CPMMH than for PMMH, and that at stationarity the CPMMH scheme is less prone to sticky behaviour. If U does not follow a standard Gaussian distribution, we may generate from a standard Gaussian distribution, transform the realisations using the CDF of a standard Gaussian distribution to give standard uniform variates, and then transform again using the inverse CDF of the desired distribution to achieve realisations of our required distribution whilst still satisfying the detailed balance equation in 2.6.

2.5.1 CPMMH illustrative example

To illustrate the use of CPMMH and its benefits over PMMH, we revisit the example of Section 2.4.1. Note that the random variable U used in the estimator of $\pi(\theta)$ follows a Gamma distribution, rather than a standard Gaussian distribution. However, consider another random variable V which does follow a standard Gaussian distribution. Denote the PDF and CDF of V by $\phi(\cdot)$ and $\Phi(\cdot)$ respectively, and the PDF and CDF of U by $g(\cdot)$ and $G(\cdot)$ respectively. Then

$$U = G^{-1}(\Phi(V)).$$

Thus, we may generate a value v from $\phi(\cdot)$ and transform it using the method described in Section 2.5, to obtain the same estimator and joint density as in Section 2.4.1. Using V means that we can use the Crank-Nicolson proposal density for $K(v^*|v)$, and it will satisfy the detailed balance equation

$$\phi(v)K(v^*|v) = \phi(v^*)K(v|v^*),$$

as shown in (2.8). The estimator for $\pi(\theta)$ is now $\hat{\pi}_V(\theta) = \pi(\theta)G^{-1}(\Phi(V))$, and the joint density is now $\hat{\pi}(\theta, v) \propto \pi(\theta)G^{-1}(\Phi(v))\phi(v)$. As we still have that $E[U] = 1$, it is straightforward to verify that this estimator remains unbiased as

$$E_V[\hat{\pi}_V(\theta)] = \pi(\theta)E_V[G^{-1}(\Phi(V))] = \pi(\theta)E_U[U] = \pi(\theta).$$

Thus, as before, an M-H scheme targeting $\hat{\pi}(\theta, v)$ will marginally target the density of interest, $\pi(\theta)$.

We can induce correlation between the successive "estimates" of $\pi(\theta)$ by proposing a new value of θ from a Gaussian random walk, proposing a new value of v from $K(v^*|v)$ with a suitably large value of ρ, and then transform v to obtain a new value of $u = G^{-1}\Phi(v)$. In R these transformations can be achieved easily with the `pnorm` and `qgamma` functions. Thus, we can run this CPMMH scheme with $\rho = 0.999$, and all other parameters equal to the example in 2.4.1. We can see from figure 2.2 that decreasing the value of a hasn't caused the same level of sticky behaviour in the chain, and this is confirmed by looking at the effective sample sizes for each chain, which to the nearest integer are 1105, 982 and 191 for $a = 1$, 0.1, and 0.01 respectively, a clear improvement on the standard uncorrelated PMMH scheme.

2.6 Stochastic differential equations

Here, we provide some background on stochastic differential equations (SDEs) by first considering diffusion processes, in particular Brownian motion, and introducing the concept

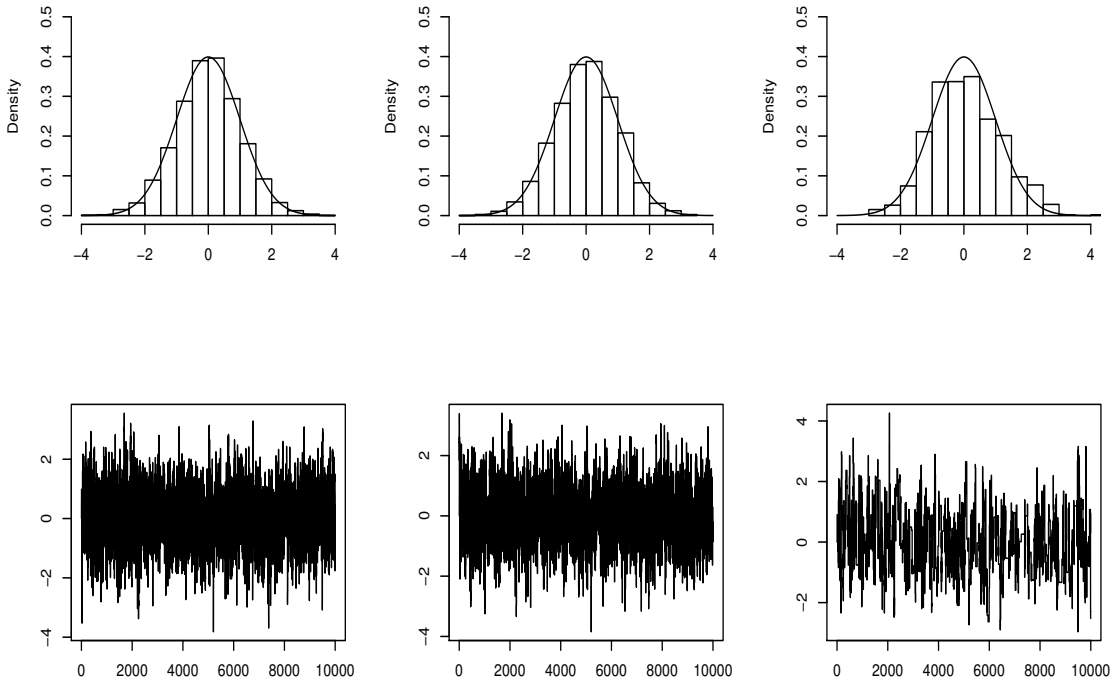

Figure 2.2: Histograms with overlaid target density, and trace plots of samples of θ from output of a CPMMH scheme with 10^4 iterations, $\rho = 0.999$ and an initial value of $\theta^{(0)} = 0$. Left panels: $a = 1$. Middle panels: $a = 0.1$. Right panels: $a = 0.01$.

of the Itô integral. For a more comprehensive introduction to SDEs, we refer the reader to Øksendal (2003).

2.6.1 Diffusion processes

Consider a stochastic Markov process $\{X_t, t \geq 0\}$, the continuous-time analogue to the continuous state-space Markov chains described in Section 2.3.1. A stochastic process is considered (first order) Markov if it satisfies the Markov condition. That is, given a sequence of n times $t_0 < t_1 < \ldots < t_n$, we have that

$$\mathbb{P}(X_{t_n} \leq x_{t_n} | X_{t_{n-1}} = x_{t_{n-1}}, X_{t_{n-2}} = x_{t_{n-2}}, \ldots X_{t_0} = x_{t_0}) = \mathbb{P}(X_{t_n} \leq x_{t_n} | X_{t_{n-1}} = x_{t_{n-1}}).$$

In other words, the future states of X_t depend only on the past states through the present state. For times $0 \leq t < t^* < \infty$, the transition density from x at time t to x^* at time t^* is denoted by $p(x^*, t^* | x, t)$, which is the continuous-time analogue to the transition density defined in Section 2.3.1. We also denote $\alpha(x, t)$ and $\beta(x, t)$ to be the infinitesimal mean

and variance of the process at time and state (x, t), also known as the drift and diffusion coefficient of the process, respectively.

Feller (1949) distinguished between two types of continuous-time Markov process: a Markov jump process, and a diffusion process. In a Markov jump process (MJP), there is an overwhelming probability that in a small time interval the state of the process will remain unchanged, however if the state does change then the change may be radical. This type of process leads to sample paths with discontinuities, and will be discussed further in Chapter 3. By contrast, in a diffusion process it is certain that some change will occur in any time interval, however small, but it is also certain that changes during small time intervals will also be small. In other words, sample paths of a diffusion process are (almost surely) continuous. The dynamics of the diffusion process going forwards and backwards in time are given by the Kolmogorov forward and backward equations (Kolmogorov, 1931). The backward equation is given by

$$-\frac{\partial p(x^*, t^*|x, t)}{\partial t} = \alpha(x, t)\frac{\partial p(x^*, t^*|x, t)}{\partial x} + \frac{1}{2}\beta(x, t)\frac{\partial^2 p(x^*, t^*|x, t)}{\partial x^2}. \tag{2.9}$$

Similarly, the forward equation is given by

$$\frac{\partial p(x^*, t^*|x, t)}{\partial t^*} = -\frac{\partial}{\partial x^*}\left(\alpha(x^*, t^*)p(x^*, t^*|x, t)\right) + \frac{1}{2}\frac{\partial}{\partial x^{*2}}\left(\beta(x^*, t^*)p(x^*, t^*|x, t)\right). \tag{2.10}$$

The forward equation is also known in this context as the Fokker-Planck equation (Fokker, 1914; Planck, 1917). Derivations of (2.9) and (2.10) can be found in Wilkinson (2018). Given $\alpha(\cdot)$ and $\beta(\cdot)$, (2.9) and (2.10) can be used to determine the transition density of the diffusion process. However, in general, these differential equations are analytically intractable.

2.6.2 Brownian motion

One diffusion process of particular interest is standard Brownian motion, first discovered by and named after Robert Brown when observing particles contained in the pollen of plants (Brown, 1828). It is also known as a Wiener process, and is thus typically denoted W_t, after Norbert Wiener, who proved its existence and provided a construction of the process (Wiener, 1923). Standard Brownian motion can be seen as a special case of a diffusion process with mean $\alpha(x, t) = 0$ and variance $\beta(x, t) = 1$. More formally, a univariate stochastic process $\{W_t, t \geq 0\}$ is a standard Brownian motion if $W_t \in \mathbb{R}$ depends continuously on t, and the following conditions are satisfied:

- $W_0 = 0$
- For all times $0 \leq t_0 < t_1 < t_2 < \infty$, $W_{t_2} - W_{t_1}$ is independent of $W_{t_1} - W_{t_0}$

- For all times $0 \leq t_0 < t_1 < \infty$, $W_{t_1} - W_{t_0} \sim N(0, t_1 - t_0)$

The second condition here ensures that standard Brownian motion has independent increments, and thus the process is (first order) Markovian as $W_{t_2} - W_{t_1}$ is independent of $\{W_t, 0 \leq t < t_1\}$. The third condition can be used to give the distribution of W_{t_1} conditional on W_{t_0}, as $W_{t_1} = W_{t_1} - W_{t_0} + W_{t_0}$, and thus $W_{t_1}|W_{t_0} = x \sim N(x, t_1 - t_0)$.

The transition density for this process is then given by

$$p(x^*, t^*|x, t) = \frac{1}{\sqrt{2\pi(t^* - t)}} \exp\left(-\frac{1}{2}\frac{(x^* - x)^2}{t^* - t}\right).$$

In this case, the Fokker-Planck equation (2.10) simplifies to

$$\frac{\partial p(x^*, t^*|x, t)}{\partial t^*} = \frac{1}{2}\frac{\partial\left(p(x^*, t^*|x, t)\right)}{\partial x^{*2}}.$$

We can differentiate the transition density with respect to t^* and x^* to show that the Fokker-Planck equation is satisfied. Using the product rule to differentiate with respect to t^*, we obtain

$$\begin{aligned}\frac{\partial p(x^*, t^*|x, t)}{\partial t^*} &= \frac{1}{\sqrt{2\pi(t^* - t)}}\frac{\partial}{\partial t^*}\left\{\exp\left(-\frac{1}{2}\frac{(x^* - x)^2}{t^* - t}\right)\right\} \\ &\quad + \frac{\partial}{\partial t^*}\left\{\frac{1}{\sqrt{2\pi(t^* - t)}}\right\}\exp\left(-\frac{1}{2}\frac{(x^* - x)^2}{t^* - t}\right) \\ &= \frac{(x^* - x)^2}{2(t^* - t)^2}\, p(x^*, t^*|x, t) - \frac{1}{2(t^* - t)}\, p(x^*, t^*|x, t).\end{aligned}$$

Differentiating the transition density with respect to x^* gives

$$\frac{\partial p(x^*, t^*|x, t)}{\partial x^*} = -\frac{x^* - x}{t^* - t}\, p(x^*, t^*|x, t),$$

and using the product rule to differentiate again with respect to x^* we obtain

$$\begin{aligned}\frac{\partial^2 p(x^*, t^*|x, t)}{\partial x^{*2}} &= -\frac{x^* - x}{t^* - t}\frac{\partial p(x^*, t^*|x, t)}{\partial x^*} + \frac{\partial}{\partial x^*}\left\{-\frac{x^* - x}{t^* - t}\right\} p(x^*, t^*|x, t) \\ &= \frac{(x^* - x)^2}{(t^* - t)^2}\, p(x^*, t^*|x, t) - \frac{1}{t^* - t}\, p(x^*, t^*|x, t).\end{aligned}$$

Multiplying this by $1/2$ shows that the Fokker-Planck equation is satisfied.

Generating a continuous-time realisation of Brownian motion is not possible, as the constant movement of the process would require infinitely many calculations. However, discrete-time sample paths of the process can be easily generated for an arbitrarily fine discretisation. For an equally spaced grid of $t_0 < t_1 < \ldots < t_m$ with $t_{i+1} - t_i = \Delta t$,

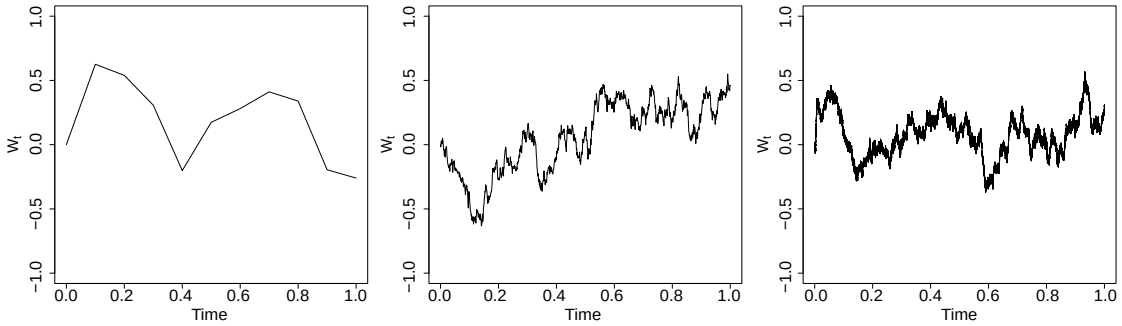

Figure 2.3: Sample paths of standard Brownian motion. Left panel: $\Delta t = 10^{-1}$. Middle panel: $\Delta t = 10^{-3}$. Right panel: $\Delta t = 10^{-5}$.

$i = 0, \ldots, m-1$, a sample path can be generated by recursively sampling

$$W_{t_{i+1}} | W_{t_i} = x_i \sim N(x_i, \Delta t).$$

Some illustrative sample paths of W_t for increasingly fine simulation grids are shown in Figure 2.3.

2.7 Itô calculus

Stochastic differential equations require the definition of stochastic integrals, as processes such as Brownian motion, whilst continuous almost everywhere, are almost nowhere differentiable (see e.g Breiman, 1968). Integrals of the form

$$\int_0^t f(X_s, s) dW_s$$

cannot be interpreted in the traditional Riemann sense. We proceed by defining the Itô stochastic integral, which is a stochastic generalisation of the Riemann integral. This integral, along with this branch of stochastic calculus, is named after Kiyosi Itô, who founded the concepts of stochastic integrals and stochastic differential equations.

Consider an equally spaced partition of $[0, t]$ as

$$0 = \tau_0 < \tau_1 < \ldots < \tau_m = t, \tag{2.11}$$

with $\Delta\tau = \tau_{i+1} - \tau_i$, $i = 0, \ldots, m-1$, so that $\tau \to 0$ as $m \to \infty$, and $\Delta W_{\tau_i} = W_{\tau_{i+1}} - W_{\tau_i}$, $i = 0, \ldots, m-1$. For a square-integrable function $f(X_s, s)$, the Itô stochastic integral is

given by

$$\int_0^t f(X_s, s)dW_s = \overset{ms}{\lim_{m\to\infty}} \sum_{i=0}^{m-1} f(X_{\tau_i}, \tau_i)\Delta W_{\tau_i}. \tag{2.12}$$

Here, $\overset{ms}{\lim}$ refers to the mean-square limit. That is, if a series S_m has a mean-square limit L as $m \to \infty$, then

$$\lim_{m\to\infty} \mathrm{E}\left((S_m - L)^2\right) = 0.$$

Note that our partition here uses the left endpoint of each sub-interval. For standard Riemann integration, it does not matter where the function was evaluated within each sub-interval, as the limit is the same in all cases. However, for stochastic calculus, this selection of the point within the sub-interval is important. The other common choice is to evaluate the function at the midpoint of each sub-interval, which leads to the Stratonovich stochastic integral (Stratonovich, 1966). For a discussion on the relative merits of Itô and Stratonovich interpretations of stochastic integrals, see Øksendal (2003). For the purposes of this thesis it is convenient to work with Itô integrals, and so stochastic integrals in this thesis shall take that form.

For some simple cases, (2.12) can be applied directly to evaluate an Itô integral. For instance, for the function $f(X_t, t) = 1$, we have

$$\begin{aligned}
\int_0^t dW_s &= \overset{ms}{\lim_{m\to\infty}} \sum_{i=0}^{m-1} \Delta W_{\tau_i} \\
&= \overset{ms}{\lim_{m\to\infty}} \left[(W_{\tau_1} - W_{\tau_0}) + (W_{\tau_2} - W_{\tau_1}) + \ldots + (W_{\tau_m} - W_{\tau_{m-1}})\right] \\
&= \overset{ms}{\lim_{m\to\infty}} (W_{\tau_m} - W_{\tau_0}) \\
&= \overset{ms}{\lim_{m\to\infty}} (W_t - W_0) \\
&= W_t - W_0 \\
&= W_t.
\end{aligned}$$

For a real-valued, square-integrable function $g(\cdot)$, a useful property of the Itô integral is that

$$\int_0^t g(s)dW_s \sim N\left(0, \int_0^t g(s)^2 ds\right). \tag{2.13}$$

To see this, we use (2.12) to write

$$\int_0^t g(s)dW_s = \overset{ms}{\lim_{m\to\infty}} \sum_{i=0}^{m-1} g(\tau_i)\Delta W_{\tau_i}$$

using the partition (2.11). Now, as $\Delta W_{\tau_i} \sim N(0, \Delta t)$, we have a linear combination of

Gaussian random variables, and so

$$\sum_{i=0}^{m-1} g(\tau_i)\Delta W_{\tau_i} \sim N\left(0, \sum_{i=0}^{m-1} g(\tau_i)^2 \Delta t\right).$$

Taking the limit as $m \to \infty$ gives a Riemann integral for the variance term, and thus we recover (2.13). An obvious corollary from this result is that

$$E\left[\int_0^t g(s)dW_s\right] = 0, \tag{2.14}$$

and

$$E\left[\left\{\int_0^t g(s)dW_s\right\}^2\right] = \int_0^t g(s)^2 ds. \tag{2.15}$$

The property given in (2.15) is known as Itô isometry.

A stochastic process $\{X_t, t \geq 0\}$ is known as an Itô process if it can be expressed as the sum of (Riemann) deterministic and (Itô) stochastic integrals. That is

$$X_t = X_0 + \int_0^t \alpha(s, X_s)dt + \int_0^t \sqrt{\beta(t, X_t)}dW_t.$$

This can be written equivalently in differential form as

$$dX_t = \alpha(t, X_t)dt + \sqrt{\beta(t, X_t)}dW_t. \tag{2.16}$$

Equation (2.16) is known as the stochastic differential equation (SDE), which shall be utilised throughout this thesis. In general, these SDEs do not permit analytic solutions (see e.g. Øksendal, 2003, for conditions and details on the existence and uniqueness of solutions to SDEs). However, there are SDEs for which an analytic solution can be obtained through Itô calculus. An example of one such process is given below.

2.7.1 SDE illustrative example

Geometric Brownian motion (GBM) is a model often used as a basic model of stock price. A GBM is a stochastic process $\{X_t, t \geq 0\}$ satisfying the SDE

$$dX_t = \theta_1 X_t dt + \theta_2 X_t dW_t, \quad X_0 = x_0 > 0. \tag{2.17}$$

Here we have a drift $\alpha(t, X_t) = \theta_1 X_t$, and diffusion coefficient $\sqrt{\beta(t, X_t)} = \theta_2 X_t$. This is an example of an SDE that can be solved analytically, using an identity known as Itô's lemma, which can be seen as the Itô calculus equivalent of the chain rule. Writing the drift and diffusion as α and β respectively for ease of notation, Itô's lemma states that

for an SDE of the form given in (2.16) and a function $f(t, x)$ that is differentiable at least once with respect to t and twice with respect to x, then

$$df(t,x) = \left(\frac{\partial f}{\partial t} + \alpha\frac{\partial f}{\partial x} + \frac{\beta}{2}\frac{\partial^2 f}{\partial x^2}\right)dt + \sqrt{\beta}\frac{\partial f}{\partial x}dW_t.$$

Applying Itô's lemma to (2.17) with $f(t, X_t) = \log(X_t)$ gives

$$\begin{aligned} d\log(X_t) &= \left(0 + \frac{\theta_1 X_t}{X_t} - \frac{\theta_2^2 X_t^2}{X_t^2}\right)dt + \frac{\theta_2 X_t}{X_t}dW_t \\ &= \left(\theta_1 - \frac{\theta_2^2}{2}\right)dt + \theta_2 dW_t. \end{aligned}$$

Integrating both sides between 0 and t gives

$$\begin{aligned} \log(X_t) - \log(X_0) &= \left(\theta_1 - \frac{\theta_2^2}{2}\right)t + \theta_2\int_0^t dW_s \\ \implies \log(X_t) &= \log(x_0) + \left(\theta_1 - \frac{\theta_2^2}{2}\right)t + \theta_2 W_t \\ \implies X_t &= x_0\exp\left\{\left(\theta_1 - \frac{\theta_2^2}{2}\right)t + \theta_2 W_t\right\}. \end{aligned}$$

Thus, X_t follows a log-normal distribution

$$X_t \sim LN\left(\log(x_0) + \left(\theta_1 - \frac{\theta_2^2}{2}\right)t, \theta_2^2 t\right).$$

Using standard results of the log-normal distribution we have that $E(X_t) = x_0 e^{\theta_1 t}$ and $\text{Var}(X_t) = x_0^2 e^{2\theta_1 t}\left(e^{\theta_2^2 t} - 1\right)$.

As with standard Brownian motion, we cannot generate a continuous-time realisation of GBM, but we can easily generate discrete-time sample paths for a given discretisation. To do so for an equally spaced grid of $t_0 < t_1 < \ldots < t_m$ with $t_{i+1} - t_i = \Delta t$, $i = 0, \ldots, m-1$, we recursively sample from

$$X_{t_{i+1}}|x_{t_i} = x_{t_i}\exp\left\{\left(\theta_1 - \frac{\theta_2^2}{2}\right)\Delta t + \theta_2\sqrt{\Delta t}Z\right\}, \quad Z \sim N(0,1).$$

Figure 2.4 shows a sample path of a GBM with $\theta_1 = 0.5$ and $\theta_2 = 1$, simulated using a discretisation of $\Delta t = 10^{-3}$.

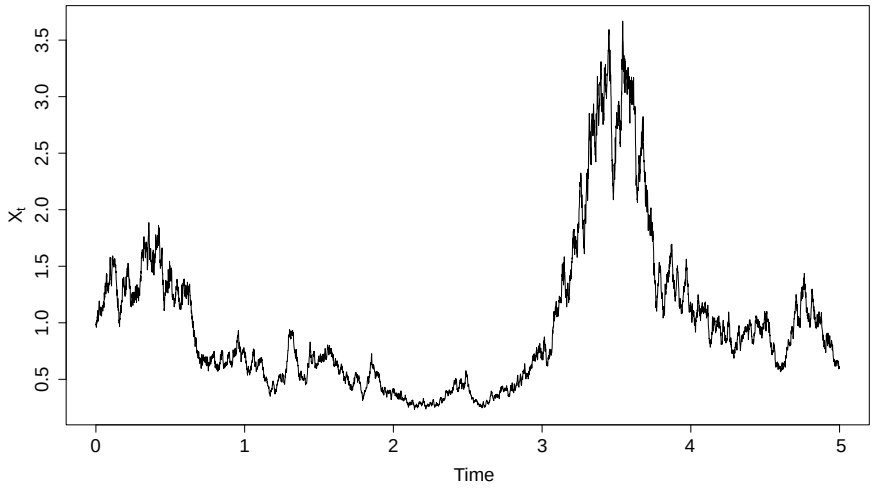

Figure 2.4: Sample path of geometric Brownian motion, with $\theta_1 = 0.5$, $\theta_2 = 1$, and $\Delta t = 10^{-3}$.

Chapter 3

Stochastic kinetic models

This chapter will provide an introduction to the class of model for which we aim to perform Bayesian inference in this thesis, namely stochastic kinetic models (SKMs). These are a flexible class of model, and have been used to model several different scenarios ranging from the spread of epidemics throughout a population (O'Neill & Roberts, 1999) to gene expression within organisms (Hey *et al.*, 2015). Typically, an SKM consists of a reaction network, an associated rate law and a probabilistic description of reaction dynamics. We first introduce a natural representation of an SKM, the Markov jump process, before detailing a stochastic differential equation approximation, the chemical Langevin equation, and a further, more tractable, approximation, the linear noise approximation. For a more comprehensive overview of stochastic kinetic models we refer the reader to Wilkinson (2018).

3.1 Markov jump processes

Consider a reaction network involving s species $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_s$ and r reactions $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_r$ such that the effect of reaction $\mathcal{R}_i$ is

$$\sum_{j=1}^{s} p_{ij}\mathcal{X}_j \longrightarrow \sum_{j=1}^{s} q_{ij}\mathcal{X}_j, \quad i = 1, \ldots, r$$

where the p_{ij} and q_{ij} are non-negative integers known as stoichiometric coefficients. Let $X_{j,t}$ denote the (discrete) number of species $\mathcal{X}_j$ at time t, and let X_t be the s-vector $X_t = (X_{1,t}, X_{2,t}, \ldots, X_{s,t})'$. A reaction $\mathcal{R}_i$ will instantaneously change the state of the system X_t, by removing p_{ij} and adding q_{ij} to $X_{j,t}$ for $j = 1, \ldots, s$. Thus for any $i \in \{1, \ldots, r\}$, if $\mathcal{R}_i$ occurs at time t the state becomes

$$X_t = X_{t-dt} + S^i,$$

Reaction type	Order	Reactants	Products	Hazard
Influx	0	$\emptyset$	$\mathcal{X}_1$	c_1
Reproduction	1	$\mathcal{X}_1$	$2\mathcal{X}_1$	$c_2 X_1$
Decay	1	$\mathcal{X}_1$	$\emptyset$	$c_3 X_1$
Catalysation	2	$\mathcal{X}_1 + \mathcal{X}_2$	$\mathcal{X}_3$	$c_4 X_1 X_2$
Absorption	2	$\mathcal{X}_1 + \mathcal{X}_2$	$2\mathcal{X}_2$	$c_5 X_1 X_2$
Dimerisation	2	$2\mathcal{X}_1$	$\mathcal{X}_2$	$c_6 X_1(X_1 - 1)/2$
Trimerisation	3	$3\mathcal{X}_1$	$\mathcal{X}_3$	$c_7 X_1(X_1 - 1)(X_1 - 2)/6$

Table 3.1: Some example reaction types and associated hazards.

where S^i is the ith column of the $s \times r$ stoichiometry matrix S whose (i, j)th element is given by $q_{ji} - p_{ji}$.

The time evolution of the process $\{X_t, t \geq 0\}$ is most naturally described by a Markov jump process (MJP), which is a continuous-time, discrete-valued Markov Process. We follow the representation of an MJP used in Golightly & Sherlock (2019), where the state of the system at time t is defined as

$$X_t = x_0 + \sum_{i=1}^{r} S^i R_{i,t}.$$

Here, $X_0 = x_0$ is the initial system state and $R_{i,t}$ is a counting process that denotes the number of times that reaction i has occured by time t. Following Kurtz (1972) (see also Wilkinson, 2018), it can be shown that

$$R_{i,t} = Y_i \int_0^t h_i(x_s, c_i) ds,$$

where Y_i are independent, unit rate Poisson processes for $i = 1, \ldots, r$, and $h_i(\cdot, c_i)$ is known as the reaction hazard. For an infinitesimal time increment dt and a reaction hazard $h_i(X_t, c_i)$, the probability of a type i reaction occurring in the time interval $(t, t + dt]$ is $h_i(X_t, c_i)dt$. Under the standard assumption of mass action kinetics, h_i is proportional to a product of binomial coefficients. Specifically

$$h_i(X_t, c_i) = c_i \prod_{j=1}^{s} \binom{X_{j,t}}{p_{ij}}. \tag{3.1}$$

Some examples of particular reactions and their hazards are given in table 3.1.Values for $c = (c_1, c_2, \ldots, c_r)'$ and the initial system state $X_0 = x_0$ complete specification of the Markov process.

The probability of observing a particular system state x_t at time t, $p(x_t)$, can be shown

Algorithm 3 Gillespie's direct method

1. Set $t = 0$. Initialise $x_0 = (x_{1,0}, \ldots, x_{u,0})'$, and set the stopping time T.
2. Calculate the hazards $h_i(x_t, c_i), i = 1, \ldots, v$ using 3.1 and the combined hazard $h_0(x_t, c)$ using 3.2.
3. Simulate the time to the next event, $t' \sim Exp(h_0(x_t, c))$.
4. Simulate the reaction index i from the set $\{1, \ldots, v\}$ with probabilities $h_i(x_t, c_i)/h_0(x_t, c)$.
5. Set $x_{t+t'} = x_t + S^i$, where S^i denotes the i-th column of S.
6. Set $t = t + t'$. Output x_t and t. If $t < T$, return to step 2.

(van Kampen, 2001) to satisfy the chemical master equation (CME):

$$\frac{d}{dt}p(x_t) = \sum_{j=1}^{r} \left[h_j(x_t - S^j)p(x_t - S^j) - h_j(x_t)p(x_t) \right].$$

Unfortunately, the CME can rarely be solved in practice, with the exactly solvable cases described in McQuarrie (1967). However, despite this intractability, generating exact realisations of the MJP is straightforward via a technique described in Algorithm 3, known in this context as *Gillespie's direct method* (Gillespie, 1977). In brief, if the current time and state of the system are t and X_t respectively, then the time to the next event will be exponential with a rate parameter equal to the *combined hazard*

$$h_0(X_t, c) = \sum_{i=1}^{r} h_i(X_t, c_i), \tag{3.2}$$

and the event will be a reaction of type $\mathcal{R}_i$ with probability $h_i(X_t, c_i)/h_0(X_t, c)$ independently of the inter-event time.

3.2 Time discretisation

Whilst generating simulations of the MJP description of the SKM is straightforward, capturing every occurrence of a reaction time and type can be computationally expensive, and this may preclude use of the MJP as an inferential model. We therefore consider two approximations to the MJP, the Poisson leap method and the chemical Langevin equation, and give an intuitive derivation of these approaches, before discussing a further approximation, namely the linear noise approximation.

Consider an infinitesimal time interval, $(t, t + dt]$, over which the reaction hazards will remain constant almost surely. The occurrence of reaction events can therefore be regarded as the occurrence of events of a Poisson process with independent realisations for each reaction type. Hence, for an interval $(t, t + \Delta t]$ of finite length, Δt, and the current system state X_t, the number of reaction events of type i, $\tilde{r}_i$, is approximately Poisson distributed with rate $h_i(X_t, c)\Delta t$. Let $\tilde{r} = (\tilde{r}_1, \ldots, \tilde{r}_r)'$. The system state can then be updated approximately, according to

$$X_{t+\Delta t} = X_t + S\tilde{r}\,. \tag{3.3}$$

This discrete approximate update is known as the Poisson leap method.

If we wish to make a further approximation, then from (3.3), and knowing the rate of each $\tilde{r}_i$, we see that the expectation and variance of the infinitesimal dX_t are

$$\mathrm{E}(dX_t) = S\,h(X_t, c)dt, \qquad \mathrm{Var}(dX_t) = S\,\mathrm{diag}\{h(X_t, c)\}S'dt,$$

where $h(X_t, c) = (h_1(X_t, c_1), \ldots, h_r(X_t, c_r))'$. Hence, we can construct an Itô stochastic differential equation (SDE) that has the same infinitesimal mean and variance as the true MJP. That is

$$dX_t = S\,h(X_t, c)dt + \sqrt{S\,\mathrm{diag}\{h(X_t, c)\}S'}\,dW_t, \tag{3.4}$$

where W_t is an s-vector of standard Brownian motion and $\sqrt{S\,\mathrm{diag}\{h(X_t, c)\}S'}$ is an $s \times s$ matrix B such that $BB' = S\,\mathrm{diag}\{h(X_t, c)\}S'$. Equation (3.4) is typically referred to as the chemical Langevin equation (CLE), and can be shown to approximate the SKM increasingly well in high concentration scenarios (Gillespie, 2000). The CLE can rarely be solved analytically, and it is common to work with a discretisation such as the Euler-Maruyama discretisation which gives

$$X_{t+\Delta t} = X_t + S\,h(X_t, c)\Delta t + \sqrt{S\,\mathrm{diag}\{h(X_t, c)\}S'\Delta t}\,Z, \tag{3.5}$$

where Z is a standard multivariate Gaussian random variable.

3.3 The linear noise approximation

The linear noise approximation (LNA) can be seen as a further approximation to an SDE such as the CLE, with increased tractability. The LNA first appeared in Kurtz (1970, 1971) as a functional central limit law for density dependent processes. Here we derive the LNA in an informal manner following that of Golightly *et al.* (2015) and Fearnhead *et al.* (2014); more formal derivations and detailed discussion can be found in Komorowski *et al.* (2009), Elf & Ehrenberg (2003) and Ferm *et al.* (2008). We begin by deriving the

LNA for a general SDE, and then use it to approximate the CLE specifically.

3.3.1 LNA derivation

Consider an SDE of the form

$$dX_t = \alpha(X_t)dt + \epsilon\beta(X_t)dW_t, \quad X_0 = x_0, \tag{3.6}$$

where ϵ is used to indicate that the stochastic term is small and is dominated by the drift, or deterministic term, of the SDE. We can then partition X_t as

$$X_t = \eta_t + \epsilon R_t, \tag{3.7}$$

where η_t is the deterministic part of X_t corresponding to the solution of

$$\frac{d\eta}{dt} = \alpha(\eta_t), \tag{3.8}$$

and R_t is a residual stochastic process. Assuming that the norm $||X_t - \eta_t||$ is $O(\epsilon)$ over a time interval of interest, we then substitute (3.7) into (3.6) to give

$$d(\eta_t + \epsilon R_t) = \alpha(\eta_t + \epsilon R_t)dt + \epsilon\beta(\eta_t + \epsilon R_t)dW_t.$$

Taylor expanding $\alpha(\cdot)$ about η_t up to terms of $O(\epsilon)$ gives

$$\alpha(\eta_t + \epsilon R_t) = \alpha(\eta_t) + \epsilon F_t R_t + \dots$$

where F_t is the Jacobian matrix with (i,j)th element $\partial\alpha_i(\eta_t)/\partial\eta_{j,t}$, and $\alpha_i(\eta_t)$ is the ith element of the vector $\alpha(\eta_t)$. Similarly, Taylor expanding $\varepsilon\beta(\cdot)$ about η_t up to terms of $O(\epsilon)$ gives

$$\epsilon\beta(\eta_t + \epsilon R_t) = \epsilon\beta(\eta_t) + \dots.$$

Collecting these terms, removing the terms relating to (3.8) and cancelling the remaining ϵ gives the approximate SDE for R_t of

$$dR_t = F_t R_t dt + \beta(\eta_t)dW_t, \tag{3.9}$$

Now that we have collected terms of $O(\epsilon)$, we may set $\epsilon = 1$ without loss of generality, as ϵ does not appear in the evolution of either η_t or R_t (equations (3.8) and (3.9) respectively).

3.3.2 LNA solution

Provided the SDE for R_t has fixed or Gaussian initial conditions, that is, $R_0 \sim N(m_0, V_0)$, then dR_t is a linear combination of Gaussians and so will have a Gaussian distribution for all t. We can then solve this SDE explicitly to give

$$R_t | R_0 = r_0 \sim N(G_t r_0, G_t \psi_t G_t'), \tag{3.10}$$

where G_t is known as the fundamental matrix and satisfies

$$\frac{dG_t}{dt} = F_t G_t, \quad G_0 = I_d, \tag{3.11}$$

and ψ_t satisfies

$$\frac{d\psi_t}{dt} = G_t^{-1} \beta^2(\eta_t) \left(G_t^{-1}\right)', \quad \psi_0 = V_0. \tag{3.12}$$

To see this, we can follow Whitaker (2016) by rewriting the identity matrix as $G_t G_t^{-1}$ and using the product rule to expand the time derivative

$$\frac{d}{dt} G_t G_t^{-1} = G_t \frac{dG_t^{-1}}{dt} + \frac{dG_t}{dt} G_t^{-1} = 0.$$

Rearranging and pre-multiplying by G_t^{-1} gives

$$\frac{dG_t^{-1}}{dt} = -G_t^{-1} \frac{dG_t}{dt} G_t^{-1}. \tag{3.13}$$

Substituting (3.11) into (3.13) gives

$$\frac{dG_t^{-1}}{dt} = -G_t^{-1} F_t G_t G_t^{-1} = -G_t^{-1} F_t. \tag{3.14}$$

Now define a new variable $A_t = G_t^{-1} R_t$. As $G_0 = G_0^{-1} = I_d$, we have that $A_0 = R_0$, and

$$dA_t = d\left(G_t^{-1} R_t\right) = R_t d\left(G_t^{-1}\right) + G_t^{-1} d\left(R_t\right).$$

Substituting in the expressions for $d\left(G_t^{-1}\right)$ and $d\left(R_t\right)$ from (3.14) and (3.9) respectively gives

$$\begin{aligned} dA_t &= -G_t^{-1} F_t R_t dt + G_t^{-1}(F_t R_t dt + \beta(\eta_t) dW_t) \\ &= G_t^{-1} \beta(\eta_t) dW_t. \end{aligned}$$

Integrating this gives

$$A_t = A_0 + \int_0^t G_t^{-1} \beta(\eta_t) dW_s.$$

From properties (2.14) and (2.15), we know that for any process X_t,

$$E\left[\int_0^t X_s dW_s\right] = 0$$

and

$$E\left[\left(\int_0^t X_s dW_s\right)^2\right] = \text{Var}\left[\int_0^t X_s dW_s\right] = E\left[\int_0^t X_s^2 ds\right].$$

Thus, as we have a linear combination of Gaussian quantities, the distribution for A_t given A_0 is

$$A_t|A_0 \sim N\left(A_0, \int_0^t G_s^{-1}\beta^2(\eta_s)(G_s^{-1})' ds\right).$$

Finally, we substitute in ψ_t from (3.12), along with $R_0 = A_0$, $R_t = G_t A_t$ to obtain (3.10). The system of coupled ODEs (3.8), (3.11) and (3.12) characterise the LNA, and must be solved either analytically or, more often, numerically. The approximating distribution of X_t is then given by

$$X_t \sim N(\eta_t + G_t r_0, G_t \psi_t G_t').$$

An equivalent representation of the LNA can be achieved by writing

$$R_t|R_0 = r_0 \sim N(m_t, V_t), \tag{3.15}$$

where

$$\frac{dm_t}{dt} = F_t m_t, \quad m_0 = r_0, \tag{3.16}$$

and the ODE for $V_t = G_t\psi_t G_t'$ can be found using the product rule, 3.11 and 3.12 to give

$$\begin{aligned}
\frac{dV_t}{dt} &= (G_t\psi_t)\frac{dG_t'}{dt} + \frac{d}{dt}(G_t\psi_t)\,G_t' \\
&= G_t\psi_t G_t' F_t' + \left(G_t\frac{d\psi_t}{dt} + \frac{dG_t}{dt}\psi_t\right)G_t' \\
&= V_t F_t' + G_t G_t^{-1}\beta^2(\eta_t)(G_t^{-1})' G_t' + F_t G_t \psi_t G_t' \\
&= V_t F_t' + \beta^2(\eta_t) + F_t V_t.
\end{aligned} \tag{3.17}$$

Thus this alternative LNA representation is characterised by the coupled ODE system of (3.8), (3.16) and (3.17), and the approximating distribution of X_t for this alternative representation is

$$X_t \sim N(\eta_t + m_t, V_t).$$

For the CLE, we have that

$$\alpha(X_t) = S\,h(X_t, c), \quad \beta(X_t, c) = \sqrt{S\,\text{diag}\{h(X_t, c)\}S'}$$

and so the LNA for the CLE has

$$\frac{d\eta}{dt} = S\,h(\eta_t, c), \tag{3.18}$$

$$\frac{dV}{dt} = V_t F_t' + S\,\mathrm{diag}\{h(\eta_t, c)\}S' + F_t V_t, \tag{3.19}$$

and

$$dR_t = F_t R_t dt + \sqrt{S\,\mathrm{diag}\{h(\eta_t, c)\}S'}dW_t.$$

3.3.3 Restarting the LNA

Fearnhead *et al.* (2014) (see also Golightly *et al.*, 2015; Minas & Rand, 2017) discuss how the accuracy of the LNA can become poor over time when the ODE satisfied by η_t is solved once over the whole time-course for a given initial condition. Essentially, for large t, it is possible that a significant discrepancy between the stochastic process and the deterministic ODE solution for η_t can emerge, leading to a poor approximation of X_t. To alleviate this problem, Fearnhead *et al.* (2014) propose 'restarting' the LNA by repeatedly re-initialising and re-integrating the ODE system. If X_t has a discrete set of n observations x_{t_i}, $i = 0, \ldots, n-1$, then we may restart the LNA by re-initialising $\eta_{t_i} = x_{t_i}$, $m_{t_i} = 0_s$ (that is, an s-length vector of zeroes) at each observation, and re-integrating forward to t_{i+1}. The rationale behind restarting the LNA is that the approximation relies on a first-order Taylor expansion about η_t, and so repeatedly realigning the point about which the expansion is performed aims to minimise the impact of the higher-order terms that have been disregarded in order to make the approximation. Note that this repeated re-initialisation has the added benefit of reducing the dimension of the ODE system, since (3.16) need no longer be solved as $m_t = 0_s$ for all $t \geq t_0$. Therefore, unless otherwise stated, all applications of the LNA in this thesis shall use this restarted version.

3.4 Examples

We illustrate the different SKM representations that we have introduced using two examples - a univariate birth-death process, and a bivariate Lotka-Volterra model.

3.4.1 Birth-death model

The birth-death model is a univariate model for the population size of a singular species. The size of the population at time t is denoted X_t, and the reaction network takes the

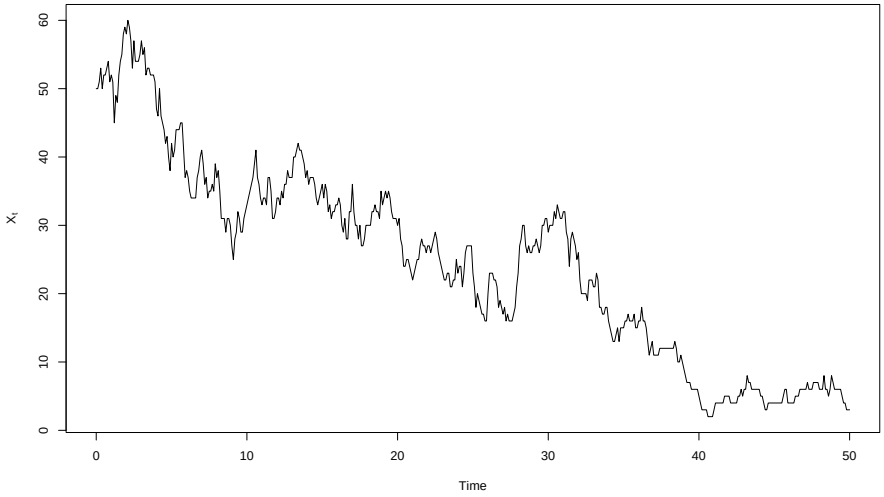

Figure 3.1: Birth-death model. A single simulation of the MJP for $t \in [0, 50]$.

form

$$\mathcal{R}_1 : \mathcal{X} \xrightarrow{c_1} 2\mathcal{X}$$
$$\mathcal{R}_2 : \mathcal{X} \xrightarrow{c_2} \emptyset,$$

where $\mathcal{R}_1$ and $\mathcal{R}_2$ denote a birth and a death in the system, respectively. A single simulation from the MJP for this model is shown in Figure 3.1. The stoichiometry matrix is given by

$$S = \begin{pmatrix} 1 & -1 \end{pmatrix}$$

and the associated hazard function is

$$h(X_t, c) = (c_1 X_t, c_2 X_t)'$$

Applying these to (3.4) gives the CLE as

$$dX_t = (c_1 - c_2)X_t dt + \sqrt{(c_1 + c_2)X_t}\, dW_t, \quad X_0 = x_0,$$

where W_t is a standard Brownian motion process. The Jacobian F_t is given by

$$F_t = c_1 - c_2.$$

Thus, the ODE system (3.18) and (3.19) governing the LNA for this model is

$$\frac{d\eta_t}{dt} = (c_1 - c_2)\eta_t, \tag{3.20}$$

$$\frac{dV_t}{dt} = 2(c_1 - c_2)V_t + (c_1 + c_2)\eta_t. \tag{3.21}$$

This is a tractable system of ODEs, which we can solve to get analytical solutions for the LNA. For η_t, we can rearrange (3.20) to find

$$\frac{d\eta_t}{\eta_t} = (c_1 - c_2)dt.$$

Integrating both sides gives

$$\begin{aligned} &\log(\eta_t) = (c_1 - c_2)t + C \\ \implies &\eta_t = \tilde{C}e^{(c_1 - c_2)t}, \end{aligned}$$

where C is the constant of integration and $\tilde{C} = e^C$. At $t = 0$, $\eta_0 = \tilde{C} = x_0$, so we have

$$\eta_t = x_0 e^{(c_1 - c_2)t}. \tag{3.22}$$

We can substitute this into (3.21) and rearrange to find

$$\frac{dV_t}{dt} - 2(c_1 - c_2)V_t = (c_1 + c_2)x_0 e^{(c_1 - c_2)t}.$$

To proceed, we multiply both sides of this equation by the integrating factor $e^{-2(c_1 - c_2)t}$, to obtain

$$\begin{aligned} \left(\frac{dV_t}{dt} - 2(c_1 - c_2)V_t\right) e^{-2(c_1 - c_2)t} &= (c_1 + c_2)x_0 e^{-(c_1 - c_2)t} \\ \implies \frac{d}{dt}(V_t e^{-2(c_1 - c_2)t}) &= (c_1 + c_2)x_0 e^{-(c_1 - c_2)t} \end{aligned}$$

Integrating both sides and then multiplying by $e^{2(c_1 - c_2)t}$ gives

$$V_t = e^{2(c_1 - c_2)t}\left(C - \frac{(c_1 + c_2)}{(c_1 - c_2)}x_0 e^{-(c_1 - c_2)t}\right),$$

where again C is the constant of integration. At $t = 0$, $V_0 = 0$, which implies that $C = \frac{(c_1 + c_2)}{(c_1 - c_2)}x_0$. Thus,

$$V_t = \frac{(c_1 + c_2)}{(c_1 - c_2)}x_0 e^{(c_1 - c_2)t}\left[e^{(c_1 - c_2)t} - 1\right]. \tag{3.23}$$

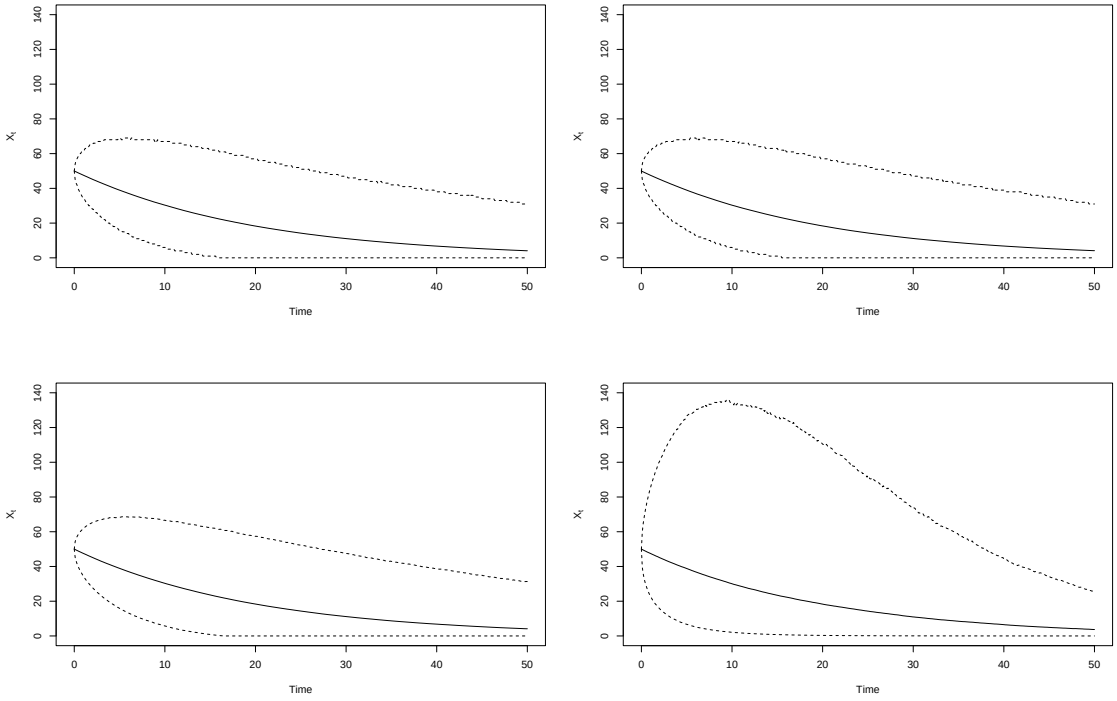

Figure 3.2: Birth-death model. Mean (solid lines) and 95% credible region (dashed lines) for 10^5 simulations of X_t with $x_0 = 50$ and $c = (0.5, 0.55)'$, with time step $\Delta t = 0.1$, using the MJP (top left), Poisson leap method (top right), CLE (bottom left), and LNA with restart (bottom right).

Figure 3.2 shows the mean and 95% credible regions for 10^5 simulations of the model, using the MJP, Poisson leap method, CLE and LNA (with restart). Note that although reactions under the MJP occur with continuous time, we have collected the state of the system in increments of 0.1 time units, to ensure direct comparisons between methods. We see that the CLE and Poisson leap method provide very good approximations to the MJP for this model. The LNA, on average, shows similar behaviour to the MJP for this model, however the LNA is far more variable, as shown by the wider credible region. Recall that one of the key assumptions for the LNA is that the stochasticity in the process is small relative to the drift. From Figure 3.1, we can see that a typical path for this process exhibits significant stochasticity relative to the expected path in Figure 3.2, which may explain why the LNA performs poorly relative to the other approximations for this model. Fearnhead *et al.* (2014) discuss further how the LNA can become more inaccurate when the perturbations of the system from the ODE solution are no longer small.

3.4.2 Lotka-Volterra model

The Lotka-Volterra reaction network comprises two biochemical species $\mathcal{X}_1$ (prey) and $\mathcal{X}_2$ (predator), and three reactions: $\mathcal{R}_1$ denotes the reproduction of a member of the prey

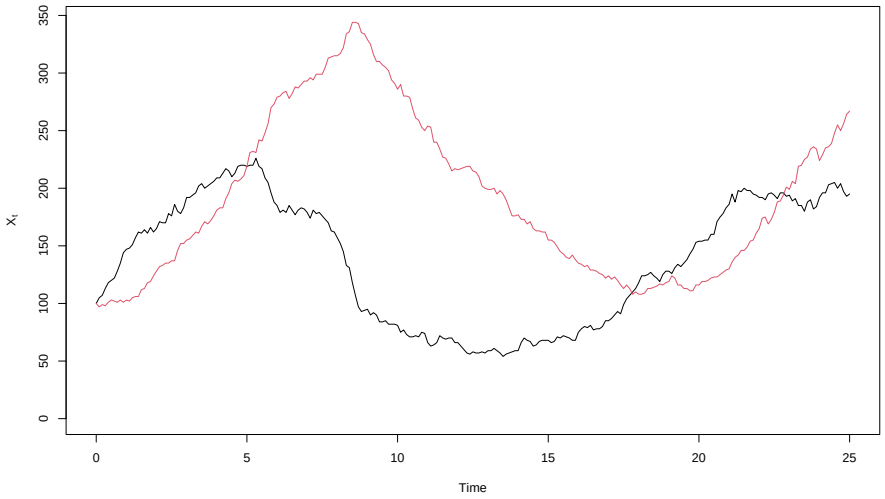

Figure 3.3: Lotka-Volterra model. A single simulation of the MJP for $X_{1,t}$ (black lines) and $X_{2,t}$ (red line), for $t \in [0, 25]$.

species, $\mathcal{R}_2$ denotes the death of a member of prey and the reproduction of a predator, and $\mathcal{R}_3$ denotes the death of a predator. The resulting reaction list is

$$\begin{aligned} \mathcal{R}_1 &: \mathcal{X}_1 \xrightarrow{c_1} 2\mathcal{X}_1 \\ \mathcal{R}_2 &: \mathcal{X}_1 + \mathcal{X}_2 \xrightarrow{c_2} 2\mathcal{X}_2 \\ \mathcal{R}_3 &: \mathcal{X}_2 \xrightarrow{c_3} \emptyset \end{aligned}$$

Let $X_t = (X_{1,t}, X_{2,t})'$ denote the system state at time t. The system is frequently used to benchmark competing inference algorithms; see e.g. Fearnhead *et al.* (2014) when using the LNA, Boys *et al.* (2008), Koblents & Miguez (2015) when using the MJP representation or Fuchs (2013), Ryder *et al.* (2021), Graham & Storkey (2017), Golightly *et al.* (2019) when using the CLE. For this reason, this model shall be revisited in subsequent chapters to illustrate different inference techniques. A single simulation from the MJP for this model is shown in Figure 3.3. The stoichiometry matrix associated with the system is given by

$$S = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}$$

and the associated hazard function is

$$h(X_t, c) = (c_1 X_{1,t}, c_2 X_{1,t} X_{2,t}, c_3 X_{2,t})'.$$

Applying these to (3.4) give the CLE as

$$d\begin{pmatrix} X_{1,t} \\ X_{2,t} \end{pmatrix} = \begin{pmatrix} c_1 X_{1,t} - c_2 X_{1,t} X_{2,t} \\ c_2 X_{1,t} X_{2,t} - c_3 X_{2,t} \end{pmatrix} dt + \begin{pmatrix} c_1 X_{1,t} + c_2 X_{1,t} X_{2,t} & -c_2 X_{1,t} X_{2,t} \\ -c_2 X_{1,t} X_{2,t} & c_2 X_{1,t} X_{2,t} + c_3 X_{2,t} \end{pmatrix}^{\frac{1}{2}} d\begin{pmatrix} W_{1,t} \\ W_{2,t} \end{pmatrix}$$

where $W_{1,t}$ and $W_{2,t}$ are independent standard Brownian motion processes. As in Section 3.3.2, we can approximate X_t as $X_t \sim N(\eta_t, V_t)$, where

$$\eta_t = \begin{pmatrix} \eta_{1,t} \\ \eta_{2,t} \end{pmatrix}, \quad V_t = \begin{pmatrix} V_{1,t} & V_{C,t} \\ V_{C,t} & V_{2,t} \end{pmatrix},$$

and $V_{C,t}$ denotes the covariance between $X_{1,t}$ and $X_{2,t}$. The Jacobian F_t is given by

$$F_t = \begin{pmatrix} c_1 - c_2\eta_{2,t} & -c_2\eta_{1,t} \\ c_2\eta_{2,t} & c_2\eta_{1,t} - c_3 \end{pmatrix}.$$

Substituting these into (3.18) and (3.19) gives the coupled ODE system that specifies the LNA for this model

$$\frac{d\eta_t}{dt} = (c_1\eta_{1,t} - c_2\eta_{1,t}\eta_{2,t}, c_2\eta_{1,t}\eta_{2,t} - c_3\eta_{2,t})', \tag{3.24}$$

$$\begin{aligned} \frac{dV_t}{dt} = {} & V_t \begin{pmatrix} c_1 - c_2\eta_{2,t} & c_2\eta_{2,t} \\ -c_2\eta_{1,t} & c_2\eta_{1,t} - c_3 \end{pmatrix} + \begin{pmatrix} c_1\eta_{1,t} + c_2\eta_{1,t}\eta_{2,t} & -c_2\eta_{1,t}\eta_{2,t} \\ -c_2\eta_{1,t}\eta_{2,t} & c_2\eta_{1,t}\eta_{2,t} + c_3\eta_{2,t} \end{pmatrix} \\ & + \begin{pmatrix} c_1 - c_2\eta_{2,t} & -c_2\eta_{1,t} \\ c_2\eta_{2,t} & c_2\eta_{1,t} - c_3 \end{pmatrix} V_t. \end{aligned} \tag{3.25}$$

These ODEs are intractable, and so must be solved numerically.

Figure 3.4 shows the mean and 95% credible regions for 10^4 simulations of this model, using the MJP (again with the system state collected every 0.1 time units), Poisson leap, CLE and LNA both with and without restart. We see that, generally, all methods approximate the MJP well at smaller times, but the accuracy of the approximation decreases at larger times. In particular, the LNA without restart approximates the behaviour of the MJP poorly at larger times. Conversely, the LNA with restart approximates the MJP particularly well across all times. In future chapters, unless specified otherwise, this thesis will restart the LNA whenever this approximation is used.

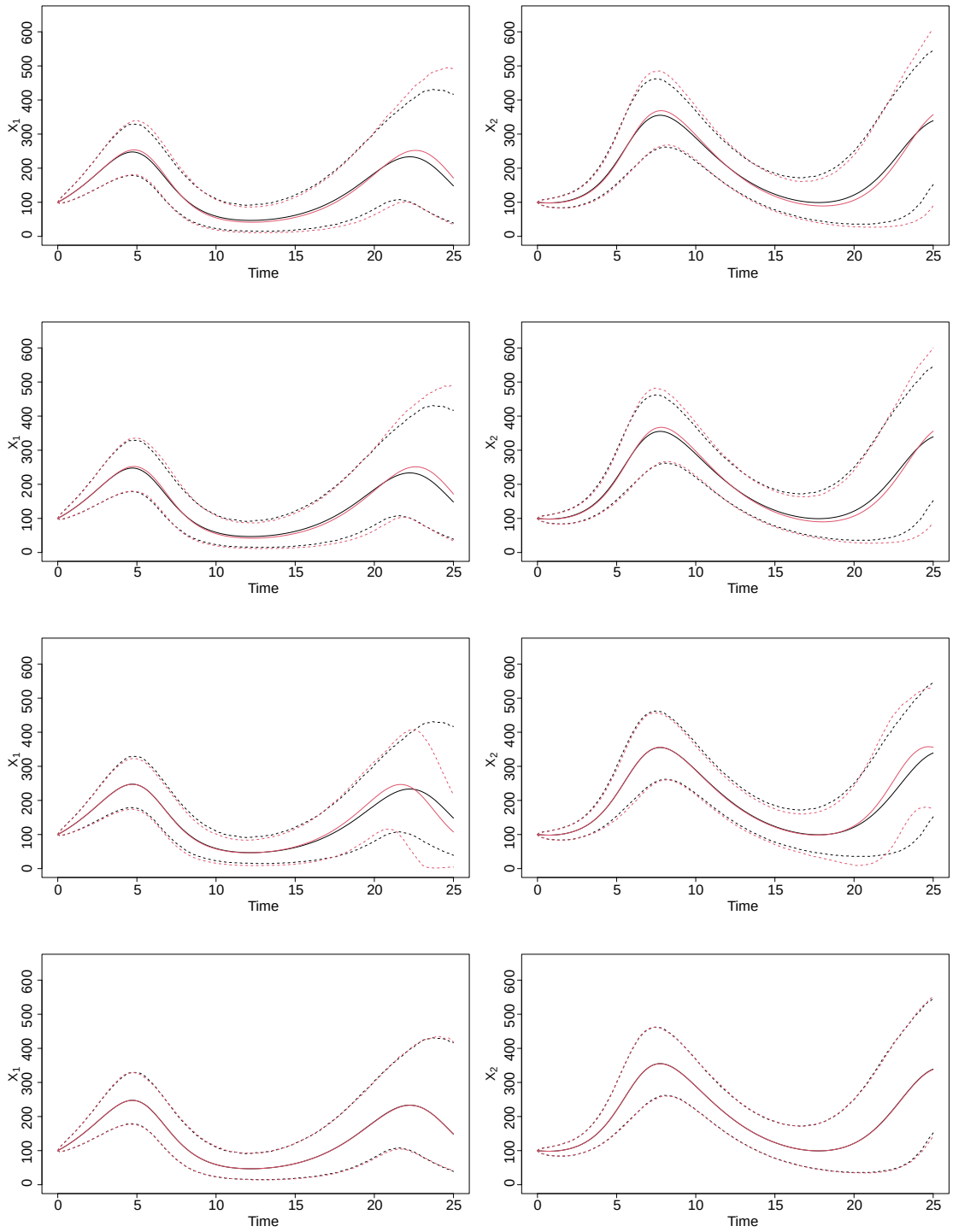

Figure 3.4: Lotka-Volterra model. Mean (solid lines) and 95% credible region (dashed lines) for 10^4 simulations of $X_{1,t}$ (left panels) and $X_{2,t}$ (right panels) with $x_0 = (100, 100)'$, $c = (0.5, 0.0025, 0.3)'$ and $\Delta t = 0.1$. In each case the black lines represent the true stochastic kinetic process (MJP), whilst the red lines represent differing approximations: the Poisson leap method (top row), CLE (second row), LNA without restart (third row), LNA with restart (bottom row).

Chapter 4

Bayesian inference for a tractable stochastic kinetic model

Suppose that the process X_t is not observed directly, but observations $\mathcal{D} = (y_{t_0}, y_{t_1}, \ldots, y_{t_n})$ are available and assumed conditionally independent (given the latent process) with conditional probability distribution obtained via the observation equation,

$$Y_{t_i} = P' X_{t_i} + \varepsilon_{t_i}, \qquad \varepsilon_{t_i} \sim \mathrm{N}(0, \Sigma), \qquad i = 0, 1, 2, \ldots, n. \tag{4.1}$$

Here, for some value d, Y_{t_i} is a length-d vector, P is a constant matrix of dimension $s \times d$ and ε_{t_i} is a length-d Gaussian random vector. Note that this allows for $d < s$ - a partial observation scenario where only a subset of the components of the network are observed. The density linking the observed and latent process is denoted by $p(y_{t_i}|x_{t_i})$, and we assume that the observations are conditionally independent given the latent process, so that the density for the observations given the latent process, $p(\mathcal{D}|x)$, is given by

$$p(\mathcal{D}|x) = \prod_{i=0}^{n} p(y_{t_i}|x_{t_i}),$$

where $p(y_{t_i}|x_{t_i}) \sim N(P' x_{t_i}, \Sigma)$ from (4.1). For simplicity we assume that Σ is known.

For the remainder of this thesis, interest lies in performing Bayesian inference on the rate constants c of stochastic kinetic models. Specifically, we seek to obtain samples from a posterior density $\pi(c|\mathcal{D})$, given a prior density $\pi(c)$ and a likelihood given observed data $p(\mathcal{D}|c)$ (or, in later chapters, an estimate of this likelihood). The posterior density can be written as

$$\pi(c|\mathcal{D}) \propto \pi(c) p(\mathcal{D}|c). \tag{4.2}$$

This thesis considers several different inferential models, and thus several different variations of $p(\mathcal{D}|c)$.

4.1 Marginal likelihood using the forward filter

In general, given data $\mathcal{D}$, the likelihoods for many representations of SKMs are intractable (see Chapter 5 for more details). One of the simplest methods for performing Bayesian inference for SKMs is therefore to replace the model with a tractable approximation. For the purposes of this chapter, we shall therefore use the LNA, introduced in Section 3.3, as the inferential model.

Denote the (approximate) posterior under the LNA by

$$\pi_a(c|\mathcal{D}) \propto \pi(c)p_a(\mathcal{D}|c).$$

The observed data (approximate) likelihood $p_a(\mathcal{D}|c)$ can be factorised as

$$p_a(\mathcal{D}|c) = p_a(y_{t_0}|c)\prod_{i=0}^{n} p_a(y_{t_{i+1}}|y_{t_0:t_i}, c), \tag{4.3}$$

where $y_{t_0:t_i} = (y_{t_0}, \ldots, y_{t_i})$. Constituent terms in (4.3) are tractable, and can be computed recursively using a forward filter, which is outlined in Algorithm 4.

The forward filter is a special case of the Kalman filter Kalman (1960). The filter leverages the linear Gaussian structure of both the LNA and the observation equation. Consider a time point t_i. At the next time point t_{i+1}, a prior distribution of the underlying process given the previous observations, $X_{t_{i+1}}|y_{t_0:t_i}$, is constructed. Bayes theorem is then applied to combine this prior with the likelihood given by the observation equation 4.1 using standard conditional multivariate Gaussian results, to obtain the posterior distribution $X_{t_{i+1}}|y_{t_0:t_{i+1}}$. This posterior distribution is then used to construct the prior distribution at the next time step. An intuitive derivation of such filters can be found in Barker *et al.* (1995).

Algorithm 4 utilises the approach introduced in Section 3.3.3 of restarting the LNA at each observation time t_i. In this case, as the observations are subject to error, we instead initialise η_{t_i} at a_{t_i}, the posterior mean of X_{t_i} given the observations up to time t_i. Also, the algorithm assumes a fixed initial condition $X_{t_0} = a$. Extending the algorithm to allow for a Gaussian initial condition $X_{t_0} \sim N(a, B)$ is straightforward: the initialisation step will just follow steps 2(b) and 2(c) but with $\eta_{t_0} = a$ and $V_{t_0} = B$.

An M-H scheme targeting $\pi_a(c|\mathcal{D})$ can therefore be implemented, with the forward filter used to evaluate the likelihood, where necessary. Note that if all components of the network are observed and observations are without error, that is, $P = I_s$ and $\Sigma = 0_s$, then Algorithm 4 simplifies considerably. In this case $a_{t_i} = y_{t_i}$, and $B_{t_i} = 0$ for all i. Thus, the (approximate) marginal likelihood may be obtained by simply integrating the ODEs

Algorithm 4 Forward Filter: Marginal likelihood under the LNA

1. Initialisation. Set $a_{t_0} = a$, $B_{t_0} = 0$. Compute

$$p_a(y_{t_0}|c) = \phi(y_{t_0}; P'a, \Sigma),$$

where $\phi(\cdot; a, B)$ denotes the Gaussian density with mean vector a and variance matrix B. The posterior at time t_0 is therefore $X_{t_0}|y_{t_0} = a$.

2. For times t_i, $i = 0, 1, \ldots, n-1$:

   (a) Prior at t_{i+1}. Initialise the LNA with $\eta_{t_i} = a_{t_i}$, $m_{t_i} = 0$ and $V_{t_i} = B_{t_i}$. Integrate (3.8) and (3.17) (and (4.7) if performing MALA) forward to t_{i+1} to obtain $\eta_{t_{i+1}}$ and $V_{t_{i+1}}$. Hence

$$X_{t_{i+1}}|y_{t_0:t_i} \sim N(\eta_{t_{i+1}}, V_{t_{i+1}}).$$

   (b) One step forecast. Using the observation equation, we have that

$$Y_{t_{i+1}}|y_{t_0:t_i} \sim N(P'\eta_{t_{i+1}}, P'V_{t_{i+1}}P + \Sigma).$$

   Compute

$$p_a(y_{t_0:t_{i+1}}|c) = p_a(y_{t_0:t_i}|c)p_a(y_{t_{i+1}}|y_{t_0:t_i}, c).$$

   (c) Posterior at t_{i+1}. Combining the distributions of $X_{t_{i+1}}$ and $Y_{t_{i+1}}$ and then conditioning on $y_{t_0:t_{i+1}}$ and c gives $X_{t_{i+1}}|y_{t_0:t_{i+1}} \sim N(a_{t_{i+1}}, B_{t_{i+1}})$ where

$$\begin{aligned} a_{t_{i+1}} &= \eta_{t_{i+1}} + V_{t_{i+1}}P(P'V_{t_{i+1}}P + \Sigma)^{-1}(y_{t_{i+1}} - P'\eta_{t_{i+1}}) \\ B_{t_{i+1}} &= V_{t_{i+1}} - V_{t_{i+1}}P(P'V_{t_{i+1}}P + \Sigma)^{-1}P'V_{t_{i+1}}. \end{aligned}$$

forward from each observation, and computing individual terms of (4.3) as

$$p_a(y_{t_{i+1}}|y_{t_0:t_i}, c) = \phi(y_{t_{i+1}}; \eta_{t_{i+1}}, V_{t_{i+1}}).$$

In scenarios where observations are not error-free, and interest lies in learning the latent process (at the observation times), we consider the joint posterior density

$$\pi_a(c, x|\mathcal{D}) \propto \pi(c)p(x_{t_0})p(x_{t_1:t_n}|x_{t_0}, c)p(\mathcal{D}|x),$$

where $x_{t_1:t_n} = (x_{t_1}, \ldots, x_{t_n})$ and $x = (x_{t_0}, \ldots, x_{t_n})$. Note that the joint posterior can also be factorised directly as

$$\pi_a(c, x|\mathcal{D}) = \pi_a(c|\mathcal{D})\pi_a(x|c, \mathcal{D}),$$

which suggests a two-stage sampling procedure as follows:

1. Draw $c \sim \pi_a(c|\mathcal{D})$

2. Draw $x \sim \pi_a(x|c, \mathcal{D})$.

The conditional posterior is tractable and can be efficiently sampled using backward sampling. This method is shown in Algorithm 5 - for further details on forward filtering and backward sampling, see West & Harrison (1997).

Algorithm 5 LNA Backward Sampler

1. From the output of Algorithm 4, sample x_{t_n} from $X_{t_n}|\mathcal{D} \sim N(a_{t_n}, B_{t_n})$.
2. For times t_i, $i = n-1, n-2, \ldots, 0$:
   (a) Joint distribution of X_{t_j} and $X_{t_{j+1}}$. Note that $X_{t_j}|y_{t_0} : y_{t_j} \sim N(a_{t_j}, B_{t_j})$. The joint distribution of X_{t_j} and $X_{t_{j+1}}$ conditional on $y_{t_0} : y_{t_j}$ is

$$\begin{pmatrix} X_{t_j} \\ X_{t_{j+1}} \end{pmatrix} \sim N\left(\begin{pmatrix} a_{t_j} \\ \eta_{t_{j+1}} \end{pmatrix}, \begin{pmatrix} B_{t_j} & B_{t_j} G'_{t_j} \\ G_{t_j} B_{t_j} & V_{t_{j+1}} \end{pmatrix} \right).$$

   (b) Backwards distribution. Conditioning further on $X_{t_{j+1}}$ gives the distribution of $X_{t_j}|X_{t_{j+1}}, y_{t_0} : y_{t_j}$ as $N(\tilde{a}_{t_j}, \tilde{B}_{t_j})$, where

$$\begin{aligned} \tilde{a}_{t_j} &= a_{t_j} + B_{t_j} G'_{t_j} V^{-1}_{t_{j+1}} (x_{t_{j+1}} - \eta_{t_{j+1}}), \\ \tilde{B}_{t_j} &= B_{t_j} - B_{t_j} G'_{t_j} V^{-1}_{t_{j+1}} G_{t_j} B_{t_j}. \end{aligned}$$

   Sample x_{t_j} from $X_{t_j}|X_{t_{j+1}}, y_{t_0} : y_{t_j} \sim N(\tilde{a}_{t_j}, \tilde{B}_{t_j})$.

4.2 Metropolis adjusted Langevin algorithm

A common choice of proposal mechanism in MCMC schemes is the random walk Metropolis (RWM) proposal discussed in Chapter 2.3.2, which we write here as

$$q(c^*|c) = N(c^*; c, \lambda \Sigma_T),$$

for some tuning covariance matrix Σ_T and step size λ, where both are typically chosen to try to optimise the mixing of the chain. For example it is common to take $\Sigma_T = \widehat{\mathrm{Var}}(c|\mathcal{D})$ estimated from a pilot run, with λ tuned to meet a desired acceptance rate. This proposal mechanism is computationally inexpensive, but does not use any information about the target density and as such can sometimes result in a lower statistical efficiency than other proposals.

Ideally, we seek a proposal using local information about the posterior to sample from areas of higher posterior density. The Metropolis adjusted Langevin Algorithm (MALA) was proposed by Roberts & Tweedie (1996) as an 'intelligent' proposal mechanism derived

from a discretised Langevin diffusion. Consider the posterior density of the rate constants, $\pi(c|\mathcal{D})$. One way of sampling from the posterior would be to construct a Markov process that has $\pi(c|\mathcal{D})$ as its stationary distribution. The gradient of the log-posterior density, $\nabla \log \pi(c|\mathcal{D})$, can be used to construct a Langevin SDE of the form

$$dc = \frac{1}{2}\nabla \log\left(\pi(c|\mathcal{D})\right) dt + dW_t. \tag{4.4}$$

This Langevin SDE has been constructed to admit the target density $\pi(c|\mathcal{D})$ as a stationary distribution. Further insight can be gleaned by considering the Fokker-Planck equation governing density of c at time t, $p(c,t)$ (where the dependence of this density on $\mathcal{D}$ has been suppressed). That is,

$$\frac{\partial}{\partial t}p(c,t) = -\frac{\partial}{\partial c}\left\{\alpha(c,t)p(c,t)\right\} + \frac{1}{2}\frac{\partial^2}{\partial c^2}\left\{\beta(c,t)p(c,t)\right\}, \tag{4.5}$$

where here $\alpha(c,t) = (1/2)\nabla \log \pi(c|\mathcal{D})$ and $\beta(c,t) = 1$. If the process is in equilibrium, then $p(c,t) = p(c)$, and $\frac{\partial}{\partial t}p(c) = 0$. Thus we can show that $\pi(c|\mathcal{D})$ is a solution of (4.5) since $\frac{\partial}{\partial t}\pi(c|\mathcal{D}) = 0$ and the right-hand side of (4.5) becomes

$$\begin{aligned}
-\frac{\partial}{\partial c}\left\{\frac{1}{2}\left(\nabla \log \pi(c|\mathcal{D})\right)\pi(c|\mathcal{D})\right\} + \frac{1}{2}\frac{\partial^2 \pi(c|\mathcal{D})}{\partial c^2} &= -\frac{\partial}{\partial c}\left\{\frac{1}{2}\frac{\pi'(c|\mathcal{D})}{\pi(c|\mathcal{D})}\pi(c|\mathcal{D})\right\} + \frac{1}{2}\pi''(c|\mathcal{D}) \\
&= -\frac{1}{2}\pi''(c|\mathcal{D}) + \frac{1}{2}\pi''(c|\mathcal{D}) \\
&= 0,
\end{aligned}$$

and so $\pi(c|\mathcal{D})$ is stationary. Note here that $\pi'(c|\mathcal{D})$ and $\pi''(c|\mathcal{D})$ denote the first and second (partial) derivatives of $\pi(c|\mathcal{D})$ with respect to c.

In general the SDE (4.4) cannot be solved analytically, and so we cannot just sample from this solution to obtain samples from the posterior distribution. However, we can discretise using the Euler-Maruyama discretisation introduced in Section 3.2 to give

$$c^* = c + \frac{\lambda}{2}\nabla \log\left(\pi(c|\mathcal{D})\right) + \sqrt{\lambda}Z, \quad Z \sim N(0, I_d),$$

for some step size λ. This will introduce a discretisation error. Taking λ to be small would reduce this discretisation error, but the resulting chain would fail to rapidly mix over the parameter space. Therefore, for a given λ, a Metropolis-Hastings step is used to correct for the discretisation. Consequently, we have a proposal mechanism that takes into account the gradient information of the target density. Statistical gains can be made by employing a preconditioning matrix (Roberts & Stramer, 2002), such that the proposal

density takes the form

$$q(c^*|c) \sim N\left(c^*; c + \frac{\lambda}{2}\nabla \log\left(\pi(c|\mathcal{D})\right), \lambda\Sigma_T\right),$$

where Σ_T is the preconditioning matrix and plays a similar role to the tuning matrix in the random walk proposal density. The extra drift term in the MALA proposal density serves to 'push' the proposed values towards regions of high posterior density. As such, MALA algorithms have a larger optimal proposal variance than RWM algorithms, and a larger asymptotic optimal acceptance rate of 0.574, compared to 0.234 for RWM. However, as with RWM, this acceptance rate does not need to be reached precisely, and in practice acceptance rates between 0.4 and 0.8 are generally seen as acceptable. It can also be shown that as the number of parameters to be inferred increases, the efficiency of MALA decreases at a slower rate than RWM, meaning that the algorithm scales better in large dimensions than RWM. For further details on the scaling and optimal acceptance rates of MALA, see e.g. Roberts & Rosenthal (1998, 2001).

Using the LNA as the inferential model we construct the likelihood using the factorisation (4.3), and note that each constituent term of the factorisation can be calculated using the forward filter, with

$$p(y_{t_{i+1}}|y_{t_0:t_i}, c) = N(y_{t_{i+1}}; P'\eta_{t_{i+1}}, P'V_{t_{i+1}}P + \Sigma).$$

Note that $\eta_{t_{i+1}}$ and $V_{t_{i+1}}$ depend on initial conditions a_{t_i} and B_{t_i}, which themselves depend implicitly on the rate constants c through the forward filter. It is not obvious how to determine this dependence analytically, and so we disregard it and treat a_{t_i} and B_{t_i} as constants independent of c, in order to obtain a closed form expression for our (approximate) gradient, and note that schemes using this approximation work well empirically. Thus, the (approximate) gradient of the constituent log-likelihood terms is given by

$$\nabla \log p(y_{t_{i+1}}|y_{t_0:t_i}, c) = \nabla \log N(y_{t_{i+1}}; P'\eta_{t_{i+1}}, P'V_{t_{i+1}}P + \Sigma).$$

For ease of notation, in what follows we shall work with a general multivariate normal distribution with mean $\mu(c,t)$ and variance $\Psi(c,t)$, and note that for our purposes $\mu(c,t) = P'\eta_{t_{i+1}}$ and $\Psi(c,t) = P'V_{t_{i+1}}P + \Sigma$, where $\eta_{t_{i+1}}$ and $V_{t_{i+1}}$ are both implicitly dependent on the rate parameters c. The form of each partial derivative for the log of a multivariate normal distribution with respect to a rate constant c_i is given by

$$\frac{\partial \log N(y; \mu(c,t), \Psi(c,t))}{\partial c_i} = \frac{1}{2}\mathrm{Tr}\left\{(\gamma\gamma^T - \Psi^{-1}(c,t))\frac{\partial \Psi(c,t)}{\partial c_i}\right\} + \gamma^T \frac{\partial \mu(c,t)}{\partial c_i}, \qquad (4.6)$$

where $\gamma = \Psi^{-1}(c,t)\{y - \mu(c,t)\}$. Evaluating (4.6) requires the first order sensitivities

$\partial\mu(c,t)/\partial c_i$ and $\partial\Psi(c,t)/\partial c_i$. These are not in general available analytically, but as we have expressions for $d\mu(c,t)/dt$ and $d\Psi(c,t)/dt$ we can find expressions for the time derivatives of the first order sensitivities by augmenting the system of ODEs giving the LNA solution. Let ξ be the vector of all elements of $\mu(c,t)$ and all lower triangular elements of $\Psi(c,t)$, and let the number of these elements be denoted N_S. Then the first order sensitivity of the jth element of ξ with respect to the ith rate constant c_i is given by

$$S_j^i = \frac{\partial \xi_j}{\partial c_i}, \quad i = 1, \ldots, r,\ j = 1, \ldots, N_S.$$

The time derivatives of these sensitivities can then be written using the total derivative as

$$\frac{d}{dt}S_j^i = \sum_{l=1}^{N_S} \frac{\partial}{\partial \xi_l}\frac{d\xi_j}{dt}S_l^i + \frac{\partial}{\partial c_i}\frac{d\xi_j}{dt}, \quad i = 1, \ldots, r,\ j = 1, \ldots, N_S. \tag{4.7}$$

For further insight into first-order sensitivity equations, see Calderhead & Girolami (2011). Given an initial condition of $S_j^i = 0$ at time t_0, these time derivatives can then be integrated forward numerically along with the rest of the component ODEs giving the LNA solution. Conveniently, this can be done as part of step 2a in Algorithm 4.

The augmentation of the LNA ODE system does come with an additional computational cost. For a reaction network with s species and r rate constants, the number of ODEs to solve excluding sensitivities is given by $s + s(s+1)/2$. With the addition of the sensitivity ODEs, the augmented system has $(r+1)\,(s + s(s+1)/2)$ ODEs in total to be solved. This means that using this method will at least double the number of ODEs to be solved, and this number can grow quickly as s and r increase, which can be computationally prohibitive for reaction systems with many species and/or reactions. As one of the underlying assumptions of the LNA is that the stochastic perturbations are small compared to the deterministic process, we can alleviate this computational cost by making a further approximation and basing our gradient information solely on the deterministic part of the LNA. This is equivalent to ignoring the dependence of $\Psi(c,t)$ on c. The partial derivative in (4.6) becomes

$$\gamma^T \frac{\partial \mu(c)}{\partial c_i}, \tag{4.8}$$

thereby reducing the number of ODE components to $s(r+1) + s(s+1)/2$. We denote the use of a MALA proposal with this additional approximation as simplified MALA, or sMALA.

4.2.1 Tail behaviour in RWM and MALA

The performance of both MALA and RWM algorithms can depend on the shape of the target density, in particular the size of the tails. If a target density $\pi(\theta)$ has most of its

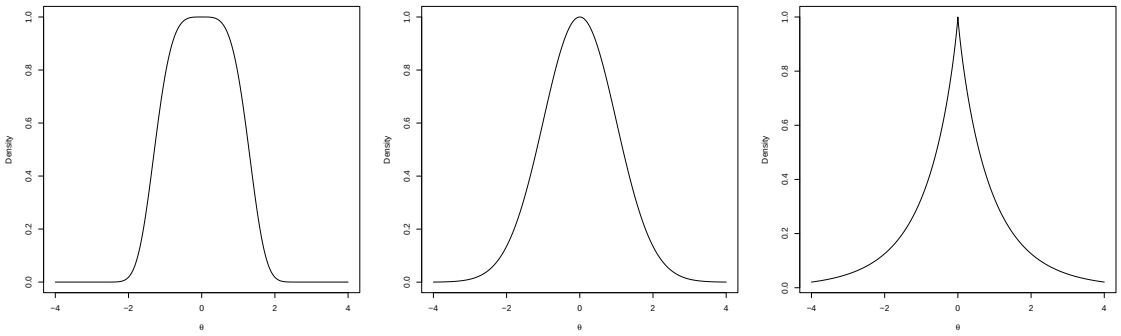

Figure 4.1: From left to right panels: illustrations of a light, standard and heavy-tailed distribution.

area centred around the mode, and the density tails off quickly as θ moves away from the mode, then the density is said to be *light-tailed*. Conversely, if the density tails off slowly away from the mode, then the density is said to be *heavy-tailed*. Illustrations of these types of distributions, along with a reference distribution proportional to a standard Gaussian, are shown in Figure 4.1.

To illustrate the behaviour of both schemes for light and heavy tails, we shall consider a simple univariate target density

$$\pi(\theta) \propto \exp\left(-\frac{|\theta|^a}{a}\right), \quad \theta \in \mathbb{R}.$$

Note that for $a = 2$, this corresponds to a standard Gaussian density, and for all a the mode of this target density is at $\theta = 0$. In particular, we shall focus on the positive tail of this distribution where $\theta \gg 0$, noting that by symmetry the behaviour will be the same for $\theta \ll 0$. The density can then be written as $\pi(\theta) \propto \exp(-\theta^a/a)$, and the gradient of the log-density becomes

$$\nabla \log \pi(\theta) = \frac{d}{d\theta}\left(\frac{-\theta^a}{a}\right) = -\theta^{a-1}. \tag{4.9}$$

The expected value of the MALA proposal distribution for this density is therefore

$$\mathrm{E}[\theta^*|\theta] = \theta - \frac{\lambda}{2}\theta^{a-1}.$$

The variance of this proposal distribution does not depend on θ or θ^*.

When $a > 2$ so that the target density has lighter tails than a standard Gaussian distribution, this proposal runs into problems. As λ is fixed, no matter how small it is fixed at, there will be some $\theta \gg 0$ for which $\lambda\theta^{a-1}/2 > 2\theta$. At this point, MALA 'overshoots', as the expectation of the proposed value θ' is beyond the mode of the target

density, and is in fact further away from this mode than the original value θ. As $\pi(\theta)$ is light-tailed, moving further away from the mode will quickly result in much lower values of $\pi(\theta)$, and thus the acceptance probability for this move will be close to 0, leading to an inefficient 'sticky' chain. For $a = 2$, the expected proposal value becomes

$$\mathrm{E}[\theta^*|\theta] = \theta(1 - \frac{\lambda}{2}).$$

Thus, a similar problem arises for $\lambda > 4$, but for $\lambda < 4$ the expected proposed value will be closer to the mode than the current value. Note that RWM does not suffer from this issue, as the expected proposed value for the RWM algorithm is θ, regardless of the tails of the distribution.

For $a < 1$, note that (4.9) tends to 0 as θ tends to infinity. Thus, for large θ, RWM and MALA exhibit similar behaviour. To examine this behaviour, recall that the acceptance probability for RWM is the minimum of 1 and a ratio R, where in this case

$$R = \frac{\pi(\theta^*)}{\pi(\theta)} = \frac{\exp(-|\theta^*|^a/a)}{\exp(-|\theta|^a/a)} = \exp\left(\frac{|\theta|^a - |\theta^*|^a}{a}\right).$$

For $\theta \gg 0$ and proposal mechanism $\theta^* = \theta + \omega$, the numerator for this exponent can be written as

$$\theta^a - (\theta + \omega)^a = \theta^a(1 - (1 + \omega/\theta)^a).$$

We can Taylor expand the $(1 + \omega/\theta)^a$ term, keeping the first two terms, to obtain

$$|\theta|^a - |\theta^*|^a \approx \theta^a(1 - (1 + a\omega/\theta)) = -\frac{a\omega}{\theta^{1-a}},$$

which tends to 0 as θ tends to ∞ or $-\infty$. Thus the acceptance probability for large $|\theta|$ becomes very close to 1, no matter the proposed value. Thus, in the tails of a heavy tailed target density, both MALA and RWM exhibit behaviour similar to a simple random walk, scarcely more likely than not to move into a region of higher density, which leads to very inefficient sampling of the target density.

4.3 Applications

To illustrate the proposed approach, and to assess the effectiveness of MALA over the basic implementation of the M-H scheme, we consider two synthetic data examples. We first fit the LNA to a simple birth-death model. Note that in this case the ODEs governing the LNA are analytically tractable, and thus the first order sensitivities required for MALA are analytically available. Our second example concerns the Lotka-Volterra predator-prey model. In this case the ODE system is analytically intractable and thus we utilise the

`lsoda` solver within the R package `deSolve` to numerically solve these equations.

In each example we use the effective sample size, as mentioned in Section 2.4.1, as a measure of statistical efficiency. This is calculated using the function `effectiveSize` in the R package `coda`. We report the minimum effective sample size over all components of the chain, denoted by mESS. To measure computational efficiency, we use wall clock time in seconds. We then use mESS/s as a comparator of overall efficiency between schemes. All algorithms are coded in R and were run on a desktop computer with an Intel Core i7-4770 processor and a 3.40GHz clock speed.

Since the rate constants must be strictly positive we target the posterior density of $log(c)$, so that our chain has $\mathbb{R}$ as its support. To tune the schemes, we ran short pilot schemes to obtain an estimate of the posterior variance $\widehat{\text{Var}}(c|\mathcal{D})$. We then used this as a basis for Σ_T, and scaled the jump size accordingly with λ to obtain acceptance rates within the range $(0.1, 0.4)$ for RWM, and $(0.4, 0.8)$ for MALA.

4.3.1 Birth-death process

Recall the birth-death process introduced in Section 3.4.1. In particular, recall that the LNA for this model had analytic solutions given by (3.22) and (3.23). From these solutions, we can derive analytical expressions for the sensitivities $\frac{\partial \eta_t}{\partial c_i}$ and $\frac{\partial V_t}{\partial c_i}, i \in \{1, 2\}$. For η_t, these sensitivities are trivially

$$\begin{aligned}\frac{\partial \eta_t}{\partial c_1} &= t x_0 e^{(c_1 - c_2)t} \\ \frac{\partial \eta_t}{\partial c_1} &= -t x_0 e^{(c_1 - c_2)t}.\end{aligned}$$

To differentiate V_t with respect to c_1, we can first rearrange to obtain

$$\frac{\partial V_t}{\partial c_1} = \frac{\partial}{\partial c_1}\left(\frac{(c_1 + c_2)}{(c_1 - c_2)} e^{2(c_1 - c_2)t} - \frac{(c_1 + c_2)}{(c_1 - c_2)} e^{(c_1 - c_2)t}\right) x_0. \tag{4.10}$$

We now have two terms to differentiate, both of which require the use of the product and quotient rules. Differentiating the first term with respect to c_1 gives

$$\begin{aligned}& e^{2(c_1 - c_2)t}\left[\frac{2t(c_1 + c_2)}{c_1 - c_2} + \frac{\partial}{\partial c_1}\left(\frac{c_1 + c_2}{c_1 - c_2}\right)\right] \\ =& 2e^{2(c_1 - c_2)t}\left[\frac{t(c_1 + c_2)}{c_1 - c_2} - \frac{c_2}{(c_1 - c_2)^2}\right].\end{aligned}$$

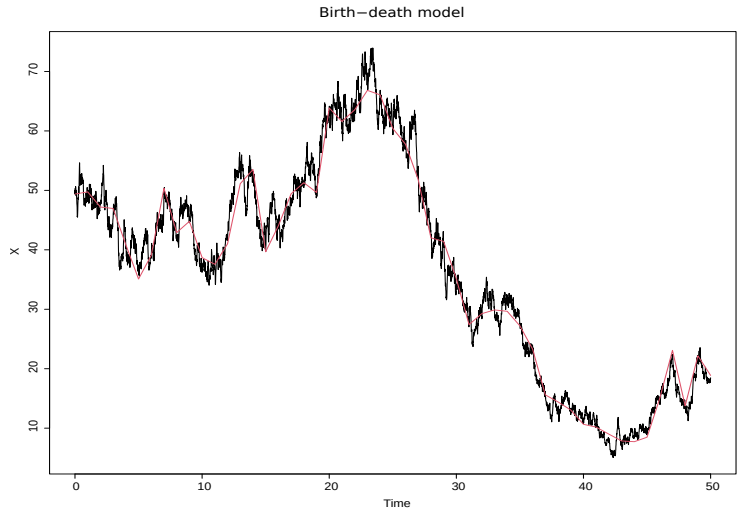

Figure 4.2: Birth-death model. Dataset (red line) and underlying Markov Jump Process (black line).

Similarly, differentiating the second term with respect to c_1 gives

$$\begin{aligned} & e^{(c_1-c_2)t}\left[\frac{t(c_1+c_2)}{c_1-c_2}+\frac{\partial}{\partial c_1}\left(\frac{c_1+c_2}{c_1-c_2}\right)\right] \\ =& e^{(c_1-c_2)t}\left[\frac{t(c_1+c_2)}{c_1-c_2}-\frac{2c_2}{(c_1-c_2)^2}\right]. \end{aligned}$$

Substituting these derivatives into (4.10), collecting terms and rearranging gives

$$\frac{\partial V_t}{\partial c_1}=\left[t(c_1^2-c_2^2)\left(2e^{(c_1-c_2)t}-1\right)-2c_2\left(e^{(c_1-c_2)t}-1\right)\right]\frac{x_0e^{(c_1-c_2)t}}{(c_1-c_2)^2}.$$

An analogous process shows that

$$\frac{\partial V_t}{\partial c_2}=\left[2c_1\left(e^{(c_1-c_2)t}-1\right)-t(c_1^2-c_2^2)\left(2e^{(c_1-c_2)t}-1\right)\right]\frac{x_0e^{(c_1-c_2)t}}{(c_1-c_2)^2}.$$

A synthetic dataset of 51 observations was generated from this model by simulating from the MJP using algorithm 3 with $c_1 = 0.5, c_2 = 0.55, X_0 = 50$ and $T = 50$, and retaining the size of the population at integer times. This data was then corrupted with additive Gaussian noise with a variance of $\sigma^2 = 1$, to give

$$Y_t \sim N(X_t, 1), \quad t = 0, 1, \ldots, 50.$$

The data, along with the underlying process, are shown in Figure 4.2.

Using the LNA as the inferential model, we took independent $N(0, 10^2)$ priors for

Proposal	α	CPU (s)	mESS	mESS/s	Rel.
RWM	0.35	40	13171	329	1
MALA	0.47	80	45481	569	1.7

Table 4.1: Birth-death model. Acceptance rate α, CPU time (in seconds), minimum ESS, minimum ESS per second and relative (to RWM) minimum ESS per second. All results are based on 10^5 iterations of each scheme.

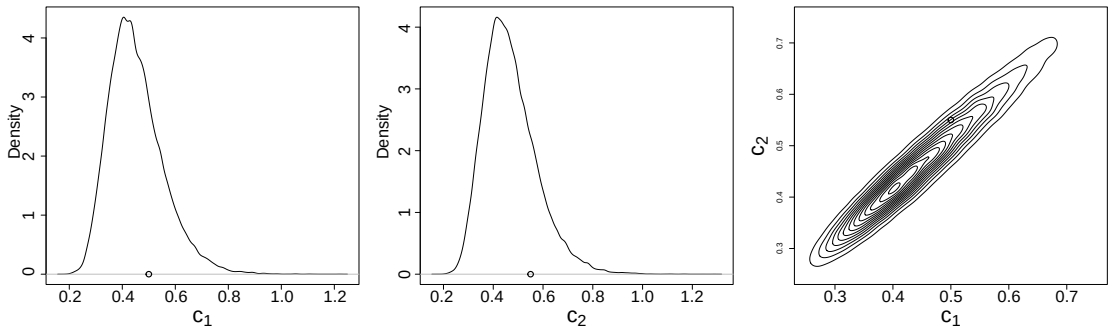

Figure 4.3: Birth-death model. Left and middle panels: marginal posterior distributions based on the RWM proposal. Right panel: contour plot of the joint posterior. The true values of c_1 and c_2 are indicated.

log c_1 and log c_2, and performed runs for M-H schemes consisting of 10^5 iterations with both RWM and MALA proposals. Results are summarised in Table 4.1 and Figure 4.3.

Figure 4.3 shows the marginal and joint posterior distributions from the RWM scheme - plots from the MALA scheme showed the same behaviour and so are omitted. We can see that the posterior samples are consistent with the true values that produced the dataset, despite using the LNA as the inferential model rather than the MJP. We also see a strong posterior correlation between the two parameters. From Table 4.1 we can see that the extra computational cost required to calculate the sensitivities and gradient for MALA is outweighed by the increase in minimum ESS achieved by the scheme. In terms of overall efficiency (as measured by minimum ESS per second), MALA outperforms RWM by almost a factor of 2.

Figure 4.4 shows the first 50 iterations of the output from RWM and MALA schemes where the chains were initialised away from the values that produced the data. As can be seen from the figure, the RWM scheme meanders more slowly towards the area of high posterior density, whereas the MALA scheme enters the area of high posterior density more quickly, and thus spends more time exploring the space of high posterior density. One consequence of this is that if a chain is initialised away from the posterior mean, MALA may require a shorter "burn-in" period than RWM, thus leading to a greater efficiency, as fewer iterations of the scheme will need to be discarded. However, care must

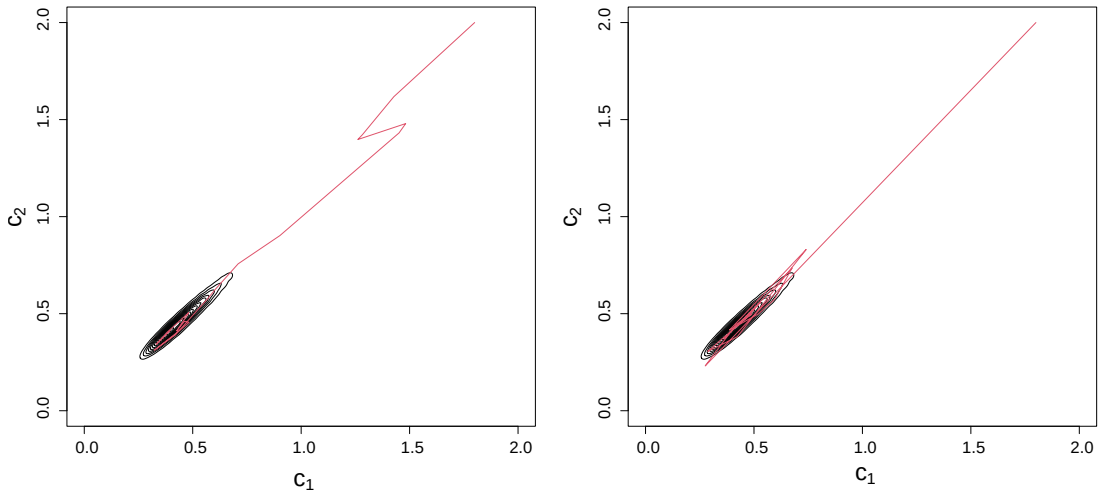

Figure 4.4: Birth-death model. Joint posterior densities and the first 50 iterations of the chain for two different schemes. Left panel: RWM. Right panel: MALA.

be taken, as if the chain is initialised too far into the tail of the posterior distribution, then the tail behaviour problems of Sections 4.2.1 may arise. In particular, if the posterior distribution is light-tailed, then MALA may actually perform worse than RWM, as if the chain is initialised far enough into the tail of the distribution it may immediately become subject to the "overshooting" problem, and thus have an intolerably low acceptance rate.

4.3.2 Lotka-Volterra model

Recall the Lotka-Volterra system introduced in Section 3.4.2. First order sensitivity ODEs for this model can be found in Appendix A.1. We generated a single realisation of the jump process at 51 integer times via Algorithm 3 with rate constants as in Boys *et al.* (2008), that is $c = (0.5, 0.0025, 0.3)'$ and an initial condition of $x_0 = (100, 100)'$, as in Golightly & Wilkinson (2011). We then corrupted the system state according to

$$Y_t \sim \mathrm{N}\left(X_t, \sigma^2 I_{2\times 2}\right), \quad t = 0, 1, \ldots, 50,$$

where $I_{2\times 2}$ is the 2×2 identity matrix and $\sigma = 1$.

Using the LNA as the inferential model, we took independent $N(0, 10^2)$ priors for the $\log c_i$, and performed runs for M-H schemes consisting of 10^5 iterations with RWM, sMALA and MALA proposal mechanisms. Results are summarised in Figure 4.5 and Table 4.2.

Figure 4.5 shows the marginal posterior distributions from the full MALA scheme - plots from the RWM and sMALA schemes showed the same behaviour and so are omit-

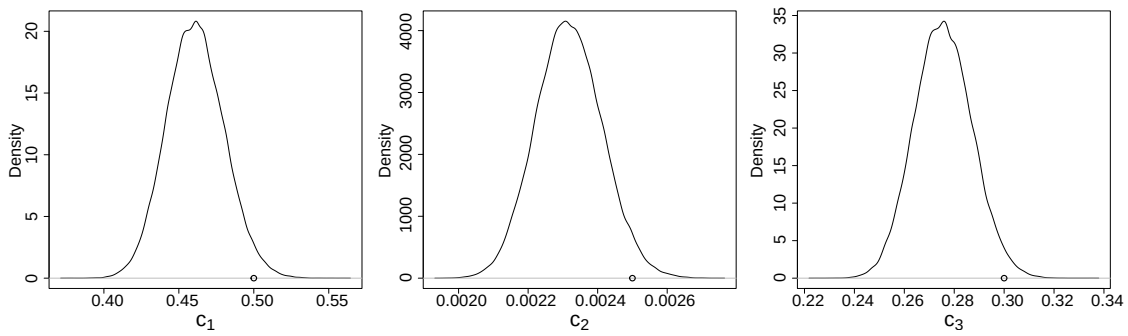

Figure 4.5: Lotka-Volterra model. Marginal posterior distributions for c_1, c_2 and c_3 respectively, based on the full MALA proposal. The true values of each parameter are indicated.

Proposal	α	CPU (s)	mESS	mESS/s	Rel.
RWM	0.34	5090	8414	1.65	1
sMALA	0.64	7740	34181	4.42	2.68
MALA	0.66	24738	37688	1.52	0.92

Table 4.2: Lotka-Volterra model. Acceptance rate α, CPU time (in seconds), minimum ESS, minimum ESS per second and relative (to RWM) minimum ESS per second. All results are based on 10^5 iterations of each scheme.

ted. Again, we can see that the posterior samples are consistent with the true values that produced the dataset. From Table 4.2 we can see that despite the clear improvement in minimum ESS gained from the full MALA proposal, this is negated by the large additional computational overhead required to solve the large system of ODEs governing the sensitivities for V_t. However, simplified MALA requires the solution of a much smaller system of ODEs, and thus has a much lower computational cost. Despite the additional gradient approximation required to reduce the size of the ODE system, the minimum ESS is almost as high as for full MALA, meaning that sMALA combines the benefits of both approaches in this scenario, outperforming both other schemes by a factor of almost 3.

4.4 Limitations of the LNA as an inferential model

As we have seen in this chapter, approximating the MJP with the LNA as an inferential model can be advantageous due to its tractability, which leads to computationally inexpensive algorithms for performing Bayesian inference for the rate constants of SKMs. However, there are certain scenarios where the LNA is not appropriate as an inferential model. As discussed in Section 3.4.1 (see also Fearnhead *et al.*, 2014), the LNA can approximate the MJP poorly in scenarios where the reaction network exhibits a large amount of stochasticity relative to its deterministic properties. Moreover, Fintzi *et al.* (2021) warn

against utilising the LNA as an inferential model in certain situations, such as the extinction and emergence of epidemic outbreaks, due to certain dynamics that cannot be captured by approximating MJP transition densities with a simple Gaussian distribution. Golightly *et al.* (2015) also found an epidemic modelling scenario where the posterior density under the LNA significantly differed from that under the MJP, and Grima *et al.* (2011) found that the CLE gave a more accurate approximation than the LNA, particularly for low-volume systems. One previously discussed limitation of the LNA is the decreasing accuracy of the approximation over time. Although the technique described in Section 3.3.3 of restarting the LNA at each observation can alleviate the problem, this technique becomes less effective for large inter-observation times, as the accuracy of the approximation may degrade even between observations. Furthermore, recall from Section 3.2 that the Poisson leap and CLE approximations share the same infinitesimal mean and variance as the true MJP. For reaction networks involving multiple species, it is in general not true that the LNA matches even the infinitesimal mean of the true stochastic process, due to correlation between different species. To illustrate this point, consider the Lotka-Volterra model of Section 3.4.2. The ODE that governs the drift of the LNA for this model is given by

$$d\begin{pmatrix}\eta_{1,t}\\ \eta_{2,t}\end{pmatrix} = \begin{pmatrix}c_1\eta_{1,t} - c_2\eta_{1,t}\eta_{2,t}\\ c_2\eta_{1,t}\eta_{2,t} - c_3\eta_{2,t}\end{pmatrix} dt.$$

The *expectation* of the CLE for this model is given by

$$d\,\mathrm{E}(X_t) = \begin{pmatrix}c_1\,\mathrm{E}(X_{1,t}) - c_2\,\mathrm{E}(X_{1,t}X_{2,t})\\ c_2\,\mathrm{E}(X_{1,t}X_{2,t}) - c_3\,\mathrm{E}(X_{2,t})\end{pmatrix} dt.$$

Whilst it is true that $\mathrm{E}(X_{1,t}) = \eta_{1,t}$ and $\mathrm{E}(X_{2,t}) = \eta_{2,t}$, the correlation between the two species means that $\mathrm{E}(X_{1,t}X_{2,t}) \neq \eta_{1,t}\eta_{2,t}$. Thus the LNA has the undesirable property that its drift does not, in general, match that of the process that it is approximating. For more limitations of the LNA, see e.g. Scott *et al.* (2006); Minas & Rand (2017). For these reasons, the next chapter will explore inference techniques when using the more intractable, but generally more accurate, CLE or Poisson leap approximations to the MJP.

Chapter 5

Bayesian inference for intractable stochastic kinetic models

Recall the scenario introduced in Chapter 4, where the process X is not observed directly, but observations (on a regular grid) $\mathcal{D} = (y_{t_0}, y_{t_1}, \ldots, y_{t_n})$ are available, related to the underlying process via (4.1). Recall that the marginal posterior density (4.2) is given by

$$\pi(c|\mathcal{D}) \propto \pi(c)p(\mathcal{D}|c),$$

where $\pi(c)$ is the prior density ascribed to c, and $p(\mathcal{D}|c)$ is the observed data likelihood. The observed data likelihood can be constructed as

$$p(\mathcal{D}|c) = \int p(x|c)p(\mathcal{D}|x)dx,$$

where

$$p(\mathcal{D}|x) = \prod_{i=0}^{n} N(y_{t_i}; x_{t_i}, \Sigma). \tag{5.1}$$

The term $p(x|c)$ is typically referred to as the complete data likelihood, whose form depends on the inferential model. In the case of the MJP, we have $x = \{x_t, t_0 \leq t \leq t_n\}$, and

$$p(x|c) = \left\{ \prod_{i=1}^{n_r} h_{\nu_i} \left(x_{\tau_{i-1}} \right) \right\} \exp \left\{ - \int_{t_0}^{t_n} \sum_{i=1}^{r} h_i \left(x_t \right) dt \right\},$$

where n_r denotes the total number of reaction events; reaction times (assumed to be in increasing order) and types are denoted by (τ_i, ν_i), $i = 1, \ldots, n_r$, $\nu_i \in \{1, \ldots, r\}$ and we take $\tau_0 = t_0$. A complete data scenario for the MJP requires knowledge of the times and types of every reaction within the system, and as such is likely to be practically infeasible, particularly for large systems.

When working with the CLE, the transition density under the Euler-Maruyama discretisation, $p_e(x_{t_{i+1}}|x_{t_i}, c)$, is given by (3.5), with $\Delta t = t_{i+1} - t_i$. However, as with the LNA (see Section 4.4), this transition density is likely to be inaccurate if the inter-observation time Δt is too large. Hence, it is commonplace to introduce intermediate time points between observation instants allowing the discretisation to operate over a time step chosen by the practitioner. To this end, consider an equally spaced partition of $[t_i, t_{i+1}]$ as

$$t_i = \tau_{i,0} < \tau_{i,1} < \ldots < \tau_{i,m-1} < \tau_{i,m} = t_{i+1}, \tag{5.2}$$

with $\tau_{i,j+1} - \tau_{i,j} = \Delta\tau = 1/m$ for $j = 0, \ldots, m-1$. Hence, the approximation is applied recursively over each sub-interval $[\tau_{t-1,i}, \tau_{t-1,i+1}]$ rather than in a single instance over $[t_i, t_{i+1}]$, with m controlling both the accuracy and computational cost of the approximation. It is then straightforward to evaluate the complete data likelihood $p(x|c)$, where $x = (x_{[t_0,t_1)}, x_{[t_1,t_2)}, \ldots, x_{[t_{n-1},t_n]})$. This takes the form

$$p(x|c) = p(x_{t_0}) \prod_{t=0}^{n-1} \prod_{i=0}^{m-1} \mathrm{N}\left(x_{\tau_{t,i+1}}; x_{\tau_{t,i}} + S\,h(x_{\tau_{t,i}}, c)\Delta\tau\,,\, S\,\mathrm{diag}\{h(x_{\tau_{t,i}}, c)\}S'\Delta\tau\right) \tag{5.3}$$

For the Poisson leap approximation we have that

$$p(\mathcal{D}|c) = \sum_{x_{t_0}, \tilde{r}} p(x_{t_0}) p(\tilde{r}|x_{t_0}, c) p(\mathcal{D}|\tilde{r}, x_{t_0})$$

where $\tilde{r} = (\tilde{r}_{\tau_{0,1}}, \ldots, \tilde{r}_{\tau_{0,m}}, \tilde{r}_{\tau_{1,1}}, \tilde{r}_{\tau_{1,2}} \ldots, \tilde{r}_{\tau_{n-1,m}})$ and for example, $\tilde{r}_{\tau_{i,j}} = (\tilde{r}_{\tau_{i,j,1}}, \ldots, \tilde{r}_{\tau_{i,j,v}})'$ is the length-r vector containing the number of reactions of each type in the interval $[\tau_{i,j-1}, \tau_{i,j}]$. It should be clear that given x_{t_0} and $\tilde{r}$, x can be obtained deterministically through recursive application of (3.3). Hence $p(\mathcal{D}|\tilde{r}, x_{t_0})$ coincides with $p(\mathcal{D}|x)$ in (5.1) and

$$p(\tilde{r}|x_{t_0}, c) = \prod_{i=0}^{n-1} \prod_{j=0}^{m-1} \prod_{k=1}^{v} \mathrm{Po}\left(\tilde{r}_{\tau_{i,j+1,k}}\,;\, h_k(x_{\tau_{i,j}}, c_k)\Delta\tau\right)$$

where $\mathrm{Po}(\cdot; h)$ denotes the mass function of a Poisson random variable with mean h.

Irrespective of whether the MJP, CLE, or Poisson leap method is used as the inferential model, the observed data likelihood $p(\mathcal{D}|c)$ remains intractable. On the other hand, whilst a complete data scenario is typically impractical, the complete data likelihood is tractable. This motivates simulation based approaches to inference based on data augmentation, whereby a sampler is constructed to target the joint posterior of c and the latent jump process between observation instants, and the uncertainty for the latent process is then integrated over via Monte Carlo. For this thesis, we focus specifically on construction of PMMH (Beaumont, 2003; Andrieu & Roberts, 2009; Andrieu *et al.*, 2010) or CPMMH

(Dahlin *et al.*, 2015; Deligiannidis *et al.*, 2018; Golightly *et al.*, 2019) schemes using either the CLE or Poisson leap as the inferential model. In the next section, we describe how these approaches, introduced in Sections 2.4 and 2.5, can be applied to perform inference for these particular SKMs.

5.1 Correlated pseudo-marginal Metropolis-Hastings

Suppose that auxiliary variables $U \sim g(u)$ can be used to generate a non-negative unbiased estimator $\hat{p}_U(\mathcal{D}|c)$ of $p(\mathcal{D}|c)$. Therefore, an unbiased (up to a multiplicative constant) estimator of the posterior is

$$\hat{\pi}_U(c|\mathcal{D}) = \pi(c)\hat{p}_U(\mathcal{D}|c).$$

In this context, the PMMH scheme of Section 2.4 becomes an M-H scheme targeting

$$\tilde{\pi}(c, u|\mathcal{D}) = \pi(c)g(u)\hat{p}_u(\mathcal{D}|c) \tag{5.4}$$

which, following the method in (2.5), has marginal distribution

$$\int \pi(c)g(u)\hat{p}_u(\mathcal{D}|c)\, du \propto \pi(c|\mathcal{D}).$$

For a proposal kernel of the form $q(c^*|c)g(u^*)$, the M-H acceptance probability is

$$\begin{aligned}\alpha\left\{(c^*, u^*)|(c, u)\right\} &= \min\left\{1\,,\, \frac{\tilde{\pi}(c^*, u^*|\mathcal{D})}{\tilde{\pi}(c, u|\mathcal{D})} \times \frac{q(c|c^*)g(u)}{q(c^*|c)g(u^*)}\right\} \\ &= \min\left\{1\,,\, \frac{\pi(c^*)\hat{p}_{u^*}(\mathcal{D}|c^*)}{\pi(c)\hat{p}_u(\mathcal{D}|c)} \times \frac{q(c|c^*)}{q(c^*|c)}\right\}\end{aligned} \tag{5.5}$$

and therefore the density associated with the auxiliary variables need not be evaluated.

Recall from Section 2.5 that the proposal kernel need not be restricted to the use of $g(u^*)$. The CPMMH scheme (Deligiannidis *et al.*, 2018; Dahlin *et al.*, 2015) generalises the PMMH scheme by using a proposal kernel of the form $q(c^*|c)K(u^*|u)$ where $K(\cdot|\cdot)$ satisfies the detailed balance equation

$$g(u)K(u^*|u) = g(u^*)K(u|u^*). \tag{5.6}$$

It is straightforward to show that an M-H scheme with proposal kernel $q(c^*|c)K(u^*|u)$ and acceptance probability (5.5) satisfies detailed balance with respect to the target $\tilde{\pi}(c, u)$.

Upon negating the trivial scenario that the chain does not move, we have that

$$\begin{aligned}
&\tilde{\pi}(c,u|\mathcal{D})q(c^*|c)K(u^*|u)\alpha\left\{(c^*,u^*)|(c,u)\right\} \\
&\quad= \min\left\{\pi(c)g(u)\hat{p}_u(\mathcal{D}|c)q(c^*|c)K(u^*|u)\,,\,\pi(c^*)g(u)\hat{p}_{u^*}(\mathcal{D}|c^*)q(c|c^*)K(u^*|u)\right\} \\
&\quad= \min\left\{\pi(c)g(u)\hat{p}_u(\mathcal{D}|c)q(c^*|c)K(u^*|u)\,,\,\pi(c^*)g(u^*)\hat{p}_{u^*}(\mathcal{D}|c^*)q(c|c^*)K(u|u^*)\right\} \\
&\quad= \tilde{\pi}(c^*,u^*|\mathcal{D})q(c|c^*)K(u|u^*)\alpha\left\{(c,u)|(c^*,u^*)\right\}
\end{aligned}$$

where (5.6) is used to deduce the third line.

In practice, the approach detailed in Section 2.5 of taking $g(u)$ to be a standard Gaussian density and $K(u^*|u)$ to be the kernel associated with a Crank-Nicolson proposal is commonplace. That is

$$g(u) = \mathrm{N}\left(u;\,0\,,\,I_d\right) \qquad \text{and} \qquad K(u^*|u) = \mathrm{N}\left(u^*;\,\rho u\,,\,\left(1-\rho^2\right)I_d\right)$$

where I_d is the identity matrix whose dimension d is determined by the number elements in u and ρ is chosen to be close to 1, to induce positive correlation between $\hat{p}_U(\mathcal{D}|c)$ and $\hat{p}_{U^*}(\mathcal{D}|c^*)$. Recall that taking $\rho = 0$ gives the special case that $K(u^*|u) = g(u^*)$, which corresponds to the PMMH scheme. The motivation for taking $\rho \approx 1$ is to reduce the variance of the acceptance probability in (5.5) - for further details, see Section 2.5, or Deligiannidis *et al.* (2018). Consequently, significant gains in statistical efficiency (relative to the standard PMMH scheme) may be expected. The use of correlation here is likely to be of most benefit in low dimensional models, since it is likely that N can be scaled at rate $n^{1/2}$ for univariate models and $n^{2/3}$ for bivariate models (Deligiannidis *et al.*, 2018), as opposed to at rate n for the standard PMMH scheme (Bérard *et al.*, 2014). Recall that in scenarios where U is not normally distributed it is straightforward to generate uniform random variates via $\Phi(U)$ (where $\Phi(\cdot)$ is the cdf of a standard normal random variable). These uniform draws can then be transformed to give draws from the required distribution via the inversion method.

The CPMMH scheme is summarised in Algorithm 6. After initialisation, each iteration requires computation of $\hat{p}_{u^*}(\mathcal{D}|c^*)$. This is achieved by executing a diffusion bridge particle filter (for each proposed value (c^*, u^*)), which we describe in the next section.

5.1.1 Diffusion bridge particle filter

The marginal likelihood $p(\mathcal{D}|c)$ can be factorised as

$$p(\mathcal{D}|c) = p(y_{t_0}|c)\prod_{i=0}^{n} p(y_{t_{i+1}}|y_{t_0:t_i},c), \tag{5.7}$$

Algorithm 6 Correlated PMMH scheme (CPMMH)

1. Initialisation. For $i = 0$:
   (a) Set $c^{(0)}$ in the support of $\pi(c|\mathcal{D})$ and draw $u^{(0)} \sim \mathrm{N}(0, I_d)$.
   (b) Compute $\hat{p}_{u^{(0)}}(\mathcal{D}|c^{(0)})$ by running Algorithm 7 with $(c, u) = (c^{(0)}, u^{(0)})$.
2. For iteration $i \geq 1$:
   (a) Draw $c^* \sim q(\cdot|c^{(i-1)})$ and $\omega \sim \mathrm{N}(0, I_d)$. Put $u^* = \rho u^{(i-1)} + \sqrt{1-\rho^2}\omega$.
   (b) Compute $\hat{p}_{u^*}(\mathcal{D}|c^*)$ by running Algorithm 7 with $(c, u) = (c^*, u^*)$.
   (c) With probability $\alpha\left\{(c^*, u^*)|(c^{(i-1)}, u^{(i-1)})\right\}$ given by (5.5), put $(c^{(i)}, u^{(i)}) = (c^*, u^*)$ otherwise store the current values $(c^{(i)}, u^{(i)}) = (c^{(i-1)}, u^{(i-1)})$.

where $y_{t_0:t_i} = (y_{t_0}, \ldots, y_{t_i})$. Although the constituent terms in (5.7) will typically be intractable, a particle filter provides an efficient mechanism for their estimation. Moreover, the particle filters that we consider here can be used to give an unbiased estimator of $p(\mathcal{D}|c)$ (Del Moral, 2004; Pitt *et al.*, 2012) and hence drive the CPMMH scheme described above.

The basic idea behind the particle filter is to recursively approximate the sequence of filtering densities $p(x_{t_i}|y_{t_0:t_i}, c)$ using a sequence of importance sampling and resampling steps, whereby N state particles are propagated forward, appropriately weighted using the complete data likelihood and observation density, and resampled with replacement (e.g. systematically as in Deligiannidis *et al.*, 2018) to prune out particle paths with low weight. Let $u = (u_1, \ldots, u_n)$ denote a realisation of the random variables required by the particle filter. We further adopt the partition $u_t = (\tilde{u}_t, \bar{u}_t)'$ to distinguish between the variables used to propagate state particles and those used in the resampling step, respectively. Note that $\tilde{u}_t = (\tilde{u}_t^1, \ldots, \tilde{u}_t^N)$ corresponding to a filter with N particles and $\tilde{u}_t^i = (\tilde{u}_{t,1}^i, \ldots, \tilde{u}_{t,m}^i)$ for $t > 1$, corresponding to the time discretisation introduced at the beginning of this chapter.

Given a weighted sample of 'particles' $\{x_{t_{i-1}}^j, w(u_{t_{i-1}}^j)\}_{j=1}^N$ approximately distributed according to $p(x_{t_{i-1}}|y_{t_0:t_{i-1}}, c)$, the particle filter uses the approximation

$$\hat{p}(x_{(t_{i-1},t_i]}|y_{t_0:t_i}, c) \propto p(y_{t_i}|x_{t_i}, c) \sum_{j=1}^{N} p(x_{(t_{i-1},t_i]}|x_{t_{i-1}}^j, c) w(u_{t_{i-1}}^j) \tag{5.8}$$

where, in the case of the CLE, $x_{(t_{i-1},t_i]} = (x_{\tau_{t_{i-1},1}}, \ldots, x_{\tau_{t_{i-1},m}})$. In the case of the Poisson leap approximation, $x_{(t_{i-1},t_i]}$ is replaced by $\tilde{r}_{(t_{i-1},t_i]} = (\tilde{r}_{\tau_{t_{i-1},1}}, \ldots, \tilde{r}_{\tau_{t_{i-1},m}})$, since x_{t_i} can be obtained deterministically, given $x_{t_{i-1}}$ and $\tilde{r}_{(t_{i-1},t_i]}$. In what follows, differences between the CLE and Poisson leap will be made explicit, and to avoid repetition $x_{(t_{i-1},t_i]}$ will be used as notation where both $x_{(t_{i-1},t_i]}$ and $\tilde{r}_{(t_{i-1},t_i]}$ are appropriate.

The auxiliary particle filter (APF) of Pitt & Shephard (1999) (see also Pitt *et al.*, 2012), which can be constructed by noting that

$$p(y_{t_i}|x_{t_i}, c)p(x_{(t_{i-1},t_i]}|x_{t_{i-1}}, c) = p(y_{t_i}|x_{t_{i-1}}, c)p(x_{(t_{i-1},t_i]}|x_{t_{i-1}}, y_{t_i}, c),$$

constructs a pre-weight $g(y_{t_i}|x^j_{t_{i-1}}, c)$, and then propagates particles via $x^j_{(t_{i-1},t_i]} = f_{t_i}(\tilde{u}^j_{t_i}) \sim g(\cdot|x^j_{t_{i-1}}, y_{t_i}, c)$, after initialising with $x^j_{t_0} = f_{t_0}(\tilde{u}^j_{t_0}) \sim g(\cdot|y_{t_0}, c)$. We use the diffusion bridge particle filter of (Golightly & Wilkinson, 2011), which can be seen as a special case of the APF with pre-weight $g(y_{t_i}|x^j_{t_{i-1}}, c) = 1$, and defer discussion on the construction of appropriate propagation constructs $g(x_{(t_{i-1},t_i]}|x^j_{t_{i-1}}, y_{t_i}, c)$ until later in this section. Note that taking $g(\cdot|x^j_{t_{i-1}}, y_{t_i}, c) = p(\cdot|x^j_{t_{i-1}}, c)$ so that each $x^j_{(t_{i-1},t_i]}$ is generated by simple forward simulation from the model (MJP, CLE or Poisson leap) gives the bootstrap particle filter of Gordon *et al.* (1993). As discussed in Golightly & Wilkinson (2015), this approach is likely to be inefficient when the inherent stochasticity in the latent process dominates the measurement error variance.

Resampling

For the resampling step we follow Deligiannidis *et al.* (2018) and use systematic resampling, which only requires simulating a single uniform random variable at each time point. These can be constructed from $\bar{u}_{t_i} \sim \mathrm{N}(0, 1)$ via $\Phi(\bar{u}_{t_i})$. Sorted uniforms can then be found via $\bar{u}^j_{Rt_i} = (j-1)/N + \Phi(\bar{u}_{t_i})/N, j = 1, \ldots, N$ which are in turn used to choose indices $a^j_{t_{i-1}}$ that (marginally) satisfy $\Pr(a^j_{t_{i-1}} = k) = w(u^k_{t_{i-1}})$. Note that upon changing c and u the effect of the resampling step is likely to prune out different particles, thus breaking the correlation between successive estimates of marginal likelihood. To alleviate this problem, Deligiannidis *et al.* (2018) sort the particles before resampling via the Hilbert sort procedure of Gerber & Chopin (2015). We follow Choppala *et al.* (2016) by using a simple Euclidean sorting procedure, which is more resource-efficient. At observation time t_i (immediately after propagation), we sort the particle trajectories $x^j_{(t_{i-1},t_i]}$ as follows. The first sorted particle corresponds to that with the smallest value of the first component of the set $\{x^1_{t_i}, \ldots, x^N_{t_i}\}$. The remaining particles are chosen by minimising the Euclidean distance between the currently selected particle and the set of all other particles. Note one potential issue of these sorting procedures is that they sort particle trajectories based on the endpoints x_{t_i}, ignoring the preceding trajectories. Thus, particle trajectories that are dissimilar over the course of $(t_{i-1}, t_i]$ but are close at t_i will be sorted close together.

The diffusion bridge particle filter is described in Algorithm 7. Note that steps 1(c) and 2(e) give the particle filter's estimate of the constituent marginal likelihood terms in (5.7).

Algorithm 7 Diffusion bridge particle filter

1. Initialisation (t_0).
   (a) Sample $\tilde{u}^j_{t_0} \sim \mathrm{N}(0,1)$ and put $x^j_{t_0} = f_{t_0}(\tilde{u}^j_{t_0}) \sim g(\cdot|y_{t_0}, c)$, $j = 1, \ldots, N$.
   (b) Compute the weights. For $j = 1, \ldots, N$

$$\tilde{w}(u^j_{t_0}) = \frac{p(x^j_{t_0})p(y_{t_0}|x^j_{t_0}, c)}{g(x^j_{t_0}|y_{t_0}, c)}, \qquad w(u^j_{t_0}) = \frac{\tilde{w}(u^j_{t_0})}{\sum_{k=1}^N \tilde{w}(u^k_{t_0})}.$$

   (c) Compute the current estimate of marginal likelihood $\hat{p}_{u_{t_0}}(y_{t_0}|c) = \sum_{j=1}^N \tilde{w}(u^j_{t_0})/N$.
2. For times $t_2, t_3, \ldots, t_n$:
   (a) Euclidean sorting: for $j = 1, \ldots, N$ obtain the sorted index $s(j)$ and put $\left\{x^j_{t_{i-1}}, w(u^j_{t_{i-1}})\right\} := \left\{x^{s(j)}_{t_{i-1}}, w(u^{s(j)}_{t_{i-1}})\right\}$.
   (b) Sample $\bar{u}_{t_i} \sim \mathrm{N}(0,1)$ and put $\bar{u}^j_{Rt_i} = (j-1)/N + \Phi(\bar{u}_{t_i})/N$, $j = 1, \ldots, N$. Obtain indices $a^j_{t_{i-1}}$ using systematic resampling with weights $w(u^j_{t_{i-1}})$.
   (c) Propagate. Sample $\tilde{u}^j_{t_i} \sim \mathrm{N}(0_m, I_m)$ and put $x^j_{(t_{i-1},t_i]} = f_{t_i}(\tilde{u}^j_{t_i}) \sim g\big(\cdot\, |x^{a^j_{t_{i-1}}}_{t_{i-1}}, y_{t_i}, c\big)$, $j = 1, \ldots, N$.
   (d) Compute the weights. For $j = 1, \ldots, N$

$$\tilde{w}(u^j_{t_i}) = \frac{p(y_{t_i}|x^j_{t_i}, c)p\big(x^i_{(t_{i-1},t_i]}|x^{a^j_{t_{i-1}}}_{t_{i-1}}, c\big)}{g\big(x^j_{(t_{i-1},t_i]}|x^{a^j_{t_{i-1}}}_{t_{i-1}}, y_{t_i}, c\big)}, \qquad w(u^j_{t_i}) = \frac{\tilde{w}(u^j_{t_i})}{\sum_{k=1}^N \tilde{w}(u^k_{t_i})}$$

   (e) Compute the current estimate of marginal likelihood $\hat{p}_{u_{t_0:t_i}}(y_{t_0:t_i}|c) = \hat{p}_{u_{t_0:t_{i-1}}}(y_{t_0:t_{i-1}}|c)\hat{p}_{u_{t_i}}(y_{t_i}|y_{t_0:t_{i-1}}, c)$ where $\hat{p}_{u_{t_i}}(y_{t_i}|y_{t_0:t_{i-1}}, c) = \frac{1}{N}\sum_{j=1}^N \tilde{w}(u^j_{t_i})$.

5.1.2 Propagation

The form of (5.8) suggests a simple importance sampling/resampling strategy where particles are sampled (with replacement) in proportion to their weights, propagated myopically of any future observations via $p(x_{(t_{i-1},t_i]}|x^j_{t_{i-1}}, c)$ and reweighted by $p(y_{t_i}|x_{t_i}, c)$. The resulting algorithm gives the bootstrap particle filter of Gordon *et al.* (1993). However, as discussed in Del Moral & Murray (2015) and Golightly & Wilkinson (2015) (see also Golightly *et al.*, 2019), this scheme is likely to perform poorly when observations are informative, since very few state particles will have reasonable weight, which leads to a highly variable estimator of the marginal likelihood. We therefore require a proposal mechanism

that can generate paths between observations for the particles, conditional on the current state of the particle, the next observation and the rate constants. These paths are often referred to as bridges, and the mechanisms for generating them are known as bridge constructs. These constructs play an important role in reducing the variance of $\hat{p}_U(\mathcal{D}|c)$ relative to the aforementioned myopic approach based on forward simulation.

Markov jump process

Without loss of generality, consider a time interval $(0, T]$ for which we require a bridge construct with density $g(x_{(0,T]}|x_0, y_T, c)$. Consider first the MJP as the inferential model and suppose that we have simulated as far as time t. A suitable bridge construct can be found by noting the conditioned hazard (CH) associated with reaction $\mathcal{R}_i$ is

$$h_i(x_t|y_T) = h_i(x_t)\frac{p(y_T|X_t = x^*)}{p(y_T|X_t = x_t)},$$

where $x^* = x_t + S^i$. The transition density $p(y_T|x_t)$ will typically be intractable. However, we may follow Golightly & Sherlock (2019) by replacing it with the transition density under the LNA

$$p_a(y_T|X_t = x_t) = \mathrm{N}\left(y_T; P'\left[\eta_{T|0} + G_{T|t}(x_t - \eta_{t|0})\right],\ P'V_{T|t}P + \Sigma\right).$$

Here, we use the notation $\eta_{t^*|t}$, $G_{t^*|t}$ and $V_{t^*|t}$ to denote the solution of the ODE system in (3.8), (3.11) and (3.17) at time $t^* > t$, integrated over $(t, t^*]$ with initial conditions $z_t = x_t$, $G_t = I_s$ and $V_t = 0_s$. Hence, a single integration of the ODE system over $[0, T]$ gives $\eta_{t|0}$, $G_{t|0}$ and $V_{t|0}$ for $t \in [0, T]$. We can then re-express $G_{T|t}$ and $V_{T|t}$ via two identities, which we derive here following a similar method to that of Golightly & Sherlock (2019). Recall from (3.10) and (3.15) that we may write

$$R_T|R_0 = r_0 \sim N(G_{T|0}r_0, V_{T|0}),$$

which we can naturally extend to

$$R_T|R_t \sim N(G_{T|t}R_t, V_{T|t}).$$

Thus we have

$$\begin{aligned}\mathrm{E}(R_T|r_0) &= G_{T|0}r_0 \\ &= G_{T|t}\,\mathrm{E}(R_t|r_0) \\ &= G_{T|t}G_{t|0}r_0.\end{aligned}$$

Our first identity is therefore

$$G_{T|t} = G_{T|0} G_{t|0}^{-1}. \tag{5.9}$$

Similarly,

$$\begin{aligned} \mathrm{Var}(R_T|r_0) &= V_{T|0} \\ &= G_{T|t}\,\mathrm{Var}(R_t|r_0) G'_{T|t} + V_{T|t} \\ &= G_{T|t} V_{t|0} G'_{T|t} + V_{T|t}, \end{aligned}$$

which leads to our second identity

$$V_{T|t} = V_{T|0} - G_{T|t} V_{t|0} G'_{T|t}. \tag{5.10}$$

Use of (5.9) and (5.10) avoids reintegration of the ODE system at each jump event. By ignoring the explicit dependence of $h_i(x_t|y_T)$ on t, sampling the resulting bridge proposal $g(x_{(0,T]}|x_0, y_T, c)$ can be achieved by executing Algorithm 3 with $h_i(x_t)$ replaced by $h_i(x_t|y_T)$. Evaluating $g(x_{(0,T]}|x_0, y_T, c)$ is straightforward via the complete data likelihood of $x_{(0,T]}$, again with $h_i(x_t)$ replaced by the conditioned hazard function.

Chemical Langevin equation

Consider now the discretised CLE as the inferential model. Recall the partition in (5.2) which we will write as

$$0 = \tau_0 < \tau_1 < \ldots < \tau_{m-1} < \tau_m = T$$

for notational simplicity, with $\Delta\tau = 1/m$ as before. We adopt the following factorisation,

$$g(x_{(0,T]}|x_0, y_T, c) = \prod_{k=0}^{m-1} g(x_{\tau_{k+1}}|x_{\tau_k}, y_T, c),$$

and seek suitable expressions for the constituent terms in the product. One option is to use the modified diffusion bridge (MDB) construct of Durham & Gallant (2002) (see also Golightly & Wilkinson (2008) for the generalisation to partial, noisy observations, and Whitaker *et al.* (2017*b*) for a recent discussion) which effectively uses a linear Gaussian approximation of $X_{\tau_{k+1}}|x_{\tau_k}, y_t, c$. Under the Euler-Maruyama approximation, the one step transition density is given by

$$X_{\tau_{k+1}}|x_{\tau_k}, c \sim N(x_{\tau_k} + \alpha_k \Delta\tau, \beta_k \Delta\tau),$$

were $\alpha_k = S\,h(x_{\tau_k}, c)$ is the drift of the CLE, $\beta_k = S\,\mathrm{diag}\{h(x_{\tau_k}, c)\}S'$ is its diffusion coefficient. The same Euler-Maruyama approximation, combined with the observation

equation (4.1) gives

$$Y_T|x_{\tau_{k+1}}, c \sim N(P'(x_{\tau_{k+1}} + \alpha_{k+1}\Delta_{k+1}), P'\beta_{k+1}P\Delta_{k+1} + \Sigma),$$

where $\alpha_{k+1} = S\,h(x_{\tau_{k+1}}, c)$, $\beta_{k+1} = S\,\mathrm{diag}\{h(x_{\tau_{k+1}}, c)\}S'$ and $\Delta_{k+1} = T - \tau_{k+1}$. To obtain a joint Gaussian distribution of $X_{\tau_{k+1}}$ and Y_T conditional only on x_{τ_k}, we make the further approximation that the hazard function is locally constant, and thus estimate α_{k+1} and β_{k+1} with α_k and β_k respectively. The (approximate) joint conditional distribution is then given by

$$\begin{pmatrix} X_{\tau_{k+1}} \\ Y_T \end{pmatrix} \Bigg| x_{\tau_k} \sim N\left(\begin{pmatrix} x_{\tau_k} + \alpha_k\Delta\tau \\ P'(x_{\tau_k} + \alpha_k\Delta_k) \end{pmatrix}, \begin{pmatrix} \beta_k\Delta\tau & \beta_k P\Delta\tau \\ P'\beta_k\Delta\tau & P'\beta_k P\Delta_k + \Sigma \end{pmatrix}\right),$$

were, $\Delta_k = T - \tau_k$. Finally, using standard multivariate Gaussian arguments we condition $X_{\tau_{k+1}}$ on $Y_T = y_t$ to obtain

$$g(x_{\tau_{k+1}}|x_{\tau_k}, y_T, c) = \mathrm{N}\left(x_{\tau_{k+1}}\,;\, x_{\tau_k} + \mu_{\mathrm{MDB}}(x_{\tau_k}, c)\Delta\tau\,,\, \Psi(x_{\tau_k}, c)\Delta\tau\right) \tag{5.11}$$

where

$$\mu_{\mathrm{MDB}}(x_{\tau_k}, c) = \alpha_k + \beta_k P\left(P'\beta_k P\Delta_k + \Sigma\right)^{-1}\left\{y_t - P'(x_{\tau_k} + \alpha_k\Delta_k)\right\}$$

and

$$\Psi(x_{\tau_k}, c) = \beta_k - \beta_k P\left(P'\beta_k P\Delta_k + \Sigma\right)^{-1} P'\beta_k\Delta\tau. \tag{5.12}$$

Given that the importance density in (5.11) is Gaussian, it is straightforward to perform the propagation step in Algorithm 7 by drawing $\tilde{u}_t^i \sim \mathrm{N}(0_m, I_m)$ and then setting

$$x_{\tau_{k+1}} = x_{\tau_k} + \mu(x_{\tau_k}, c)\Delta\tau + \sqrt{\Psi(x_{\tau_k}, c)\Delta\tau}\,\tilde{u}_{t,k+1}^i, \qquad k = 0, \ldots, m-1.$$

The effect of this bridge construct is to effectively "push" particles linearly towards observations, so that on average the particle travels in a straight line between observations.

Poisson leap

For the Poisson leap approximation, we factorise as

$$g(\tilde{r}_{(0,T]}|x_0, y_T, c) = \prod_{k=0}^{m-1} g(\tilde{r}_{\tau_{k+1}}|x_{\tau_k}, y_T, c),$$

and again seek suitable expressions for the constituent terms in the product. We take $g(\tilde{r}_{\tau_{k+1}}|x_{\tau_k}, y_T, c)$ to be a Poisson probability, with rate given by the (approximate) expected number of reactions $\mathrm{E}(\tilde{R}_{\tau_{k+1}})$ in $(\tau_k, \tau_{k+1}]$ given the current state of the system

x_{τ_k} and the observation y_T. To obtain this approximate rate, we follow the derivation of Golightly & Wilkinson (2015), combined with the MDB approach above. First assume a constant reaction hazard $h(x_{\tau_k}, c)$ over $(\tau_k, T]$, and then take a Normal approximation to the corresponding Poisson distribution for $\tilde{R}_{\tau_{k+1}}$ as

$$\tilde{R}_{\tau_{k+1}}|X_{\tau_k} = x_{\tau_k} \sim N\left(h(x_{\tau_k}, c)\Delta\tau, \text{diag}\{h(x_{\tau_k}, c)\}\Delta\tau\right).$$

Let $\tilde{R}_{T^-}$ denote the number of reactions over $(\tau_{k+1}, T]$. Then

$$\tilde{R}_{T^-}|X_{\tau_{k+1}} = x_{\tau_{k+1}} \sim N\left(h(x_{\tau_{k+1}}, c)\Delta_{k+1}, \text{diag}\{h(x_{\tau_{k+1}}, c)\}\Delta_{k+1}\right).$$

As we have Gaussian observation error from (4.1) we have that

$$Y_T|X_{\tau_{k+1}} = x_{\tau_{k+1}} \sim N\left(P'(x_{\tau_{k+1}} + S\tilde{R}_{T^-}), P'\beta_k P\Delta_{k+1} + \Sigma\right).$$

Thus the (approximate) joint density of $\tilde{R}_{\tau_{k+1}}$ and Y_T (conditional on x_{τ_k}) is

$$\begin{pmatrix}\tilde{R}_{\tau_{k+1}} \\ Y_T\end{pmatrix} \sim N\left\{\begin{pmatrix}h(x_{\tau_k}, c)\Delta\tau \\ P'(x_{\tau_k} + \alpha_k\Delta_k)\end{pmatrix}, \begin{pmatrix}\text{diag}\{h(x_{\tau_k}, c)\}\Delta\tau & \text{diag}\{h(x_{\tau_k}, c)\}S'P\Delta\tau \\ P'S\,\text{diag}\{h(x_{\tau_k}, c)\}\Delta\tau & P'\beta_k P\Delta_k + \Sigma\end{pmatrix}\right\}.$$

We can then take the expectation of $\tilde{R}_{\tau_{k+1}}|Y_T = y_T$ using standard multivariate Gaussian arguments, and divide the resulting expression by $\Delta\tau$ to give an (approximate) conditioned reaction hazard

$$h_{\text{PL}}(x_{\tau_k}, c|y_T) = h(x_{\tau_k}, c) + \text{diag}\{h(x_{\tau_k}, c)\}S'P(P'\beta_k P\Delta_k + \Sigma)^{-1}\left[y_T - P'(x_{\tau_k} + \alpha_k\Delta_k)\right].$$

Hence, we obtain

$$g(\tilde{r}_{\tau_{k+1}}|x_{\tau_k}, y_T, c) = \prod_{j=1}^{r} \text{Po}(\tilde{r}_{\tau_{k+1},j}; h^{PL}(x_{\tau_k}, c|y_T)\Delta\tau). \tag{5.13}$$

The propagation step in Algorithm 7 can be performed by drawing $\tilde{u}^j_{t_i,k+1} \sim N(0, I_r)$ and then applying the inverse Poisson CDF to each component of $\Phi(\tilde{u}^j_{t_i,k+1})$ to give $\tilde{r}_{\tau_{k+1}}$ for $k = 0, 1, \ldots, m-1$. We then set

$$x_{\tau_{k+1}} = x_{\tau_k} + S\tilde{r}_{\tau_{k+1}}, \quad k = 0, 1, \ldots, m-1.$$

5.1.3 Tuning

A single iteration of the CPMMH scheme described in Algorithm 7 requires $n-1 \times m \times N$ draws of the bridge construct with density (5.11) when using the CLE, and mass function

(5.13) when using the Poisson leap (when using the MJP, the bridge construct requires the use of Algorithm 3, and so the number of draws is dependent on the number of reactions). Recall that n is the number of observations, m is the number of latent process values per observation interval and N is the number of particles in the auxiliary particle filter. The cost of drawing from (5.11) and (5.13) is dictated by the number of observed components d, where $d \leq s$, since the inversion of $d \times d$ matrices is required. It remains to choose m and N to balance posterior accuracy with the cost and variance of the particle filter estimator.

To choose m (or equivalently, $\Delta\tau$), we follow Stramer & Bognar (2011) and Golightly & Wilkinson (2011) among others, by performing short pilot runs of the inference scheme (for a fixed, conservative value of N), with increasing values of m (decreasing values of $\Delta\tau$, until no discernible difference in the posterior output is detected (this is typically done heuristically by visual inspection of kernel density estimates of the marginal parameter posteriors).

The number of particles N used in the scheme can be chosen by following the practical advice proposed by Tran *et al.* (2016) for their block PMMH method, which was extended to the CPMMH method by Choppala *et al.* (2016). The variance of the log-posterior (σ_N^2 computed with N particles) at a central value of c (e.g. estimated posterior mean) should satisfy $\sigma_N^2 = 2.16^2/1 - \rho_l^2$ where ρ_l is the correlation between $\hat{p}_u(y|c)$ and $\hat{p}_{u'}(y|c')$, estimated from a short pilot run with parameters fixed at the same central value of c. Note that $\rho_l = 0$ corresponds to the vanilla PMMH case in which case the aforementioned tuning advice is broadly consistent with Sherlock *et al.* (2015).

5.2 Applications

To illustrate the proposed approaches we consider four applications of increasing complexity. A simple immigration-death model is considered in Section 5.2.1. We fit the CLE to synthetic data and compare CPMMH with PMMH and additionally, the state-of-the-art MCMC scheme, that is, the modified innovation scheme (MIS) of Golightly & Wilkinson (2008), described briefly in Appendix B.1. In Section 5.2.2, we fit the CLE associated with a Lotka-Volterra model to synthetic data. We also investigate the effect of increasing observation noise on the performance of the CPMMH scheme. The autoregulatory network of Sherlock *et al.* (2014) is considered in Section 5.2.3. We generate synthetic data that is inherently discrete, precluding the use of the CLE as an inferential model. We therefore perform inference using the Poisson leap, and additionally explore the effect of using a bootstrap particle filter on the performance of the CPMMH scheme. In Section 5.2.4, the CLE approximation of a Susceptible–Infected–Removed (SIR) epidemic model is fitted using data on an influenza outbreak in a boys' boarding school in Britain (BMJ News and

Notes, 1978). It is assumed that the infection rate is a mean reverting diffusion process giving a model with two unobserved components.

Since the rate constants must be strictly positive we update $\log(c)$, as in Section 4.3, and use an RWM proposal with Gaussian innovations. We took the innovation variance to be the posterior variance of $\log(c)$ (estimated from a pilot run) scaled by a factor of $2.56^2/r$ for (C)PMMH and $2.38^2/r$ for MIS, as in Sherlock *et al.* (2015). Recall that r is the number of rate constants. This scaling factor can then be fine-tuned to achieve the parameter dimension-dependent optimal acceptance rates in Schmon *et al.* (2021) for (C)PMMH and in Schmon & Gagnon (2021) for MIS. We choose m and N following the advice in Section 5.1.3, and choose ρ according to the approach mentioned in Section 2.5, that is, choosing the largest possible ρ such that the ESS of the auxiliary chain is broadly consistent with the ESS of the parameter chain to mitigate long term dependence between parameter draws. As in Section 4.3, we use effective sample size (calculated using the function `effectiveSize` in the R package `coda`) as a comparator, as well as wall clock time. We report the minimum effective sample size over all components of the chain, denoted by mESS. All algorithms are coded in R and were run on a desktop computer with an Intel Core i7-4770 processor and a 3.40GHz clock speed.

5.2.1 Immigration-death model

The immigration-death reaction network takes the form

$$\begin{aligned} \mathcal{R}_1 &: \emptyset \xrightarrow{c_1} \mathcal{X}_1 \\ \mathcal{R}_2 &: \mathcal{X}_1 \xrightarrow{c_2} \emptyset \end{aligned}$$

with immigration and death reactions shown respectively. The stoichiometry matrix is given by

$$S = \begin{pmatrix} 1 & -1 \end{pmatrix}$$

and the associated hazard function is

$$h(X_t, c) = (c_1, c_2 X_t)'$$

where X_t denotes the state of the system at time t. Applying (3.4) directly gives the CLE as

$$dX_t = (c_1 - c_2 X_t)\, dt + \sqrt{(c_1 + c_2 X_t)}\, dW_t.$$

We generated a synthetic data set consisting of 101 observations by simulating from the Markov jump process via Algorithm 3 and retaining the system state at integer times. To provide a challenging scenario for the CLE, we took $c_1 = 4$ and $c_2 = 0.8$ giving inherently

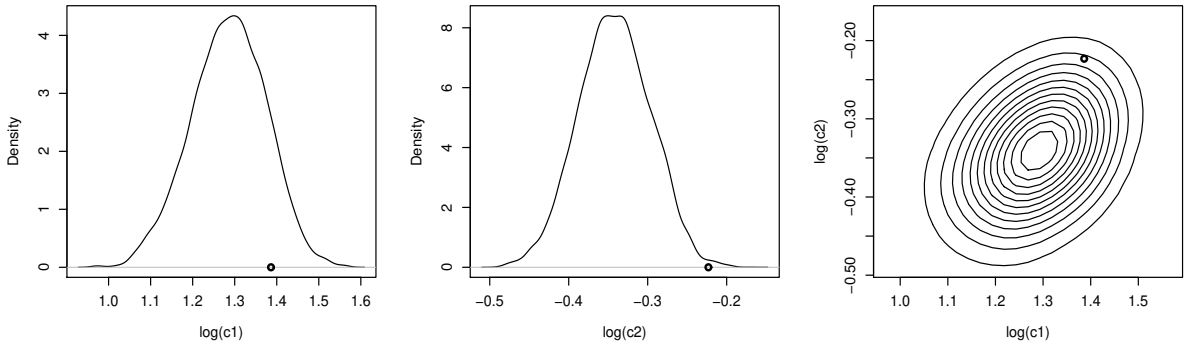

Figure 5.1: Immigration death model. Left and middle panels: marginal posterior distributions based on the output of CPMMH ($\rho = 0.99$). Right panel: Contour plot of the joint posterior. The true values of $\log(c_1)$ and $\log(c_2)$ are indicated.

discrete trajectories that 'mean revert' around the value 5. Moreover, we took $X_0 = 500$ so that typical trajectories exhibit nonlinear dynamics over the time interval $[0, 10]$, but are reasonably linear between observation times. We assume error-free observation of X_t so that the latent path between observation times, which is propagated according to equation(5.11), becomes

$$g(x_{\tau_{k+1}}|x_{\tau_k}, x_t, c) = \mathrm{N}\left(x_{\tau_{k+1}}\,;\, x_{\tau_k} + \frac{x_t - x_{\tau_k}}{t - \tau_k}\Delta\tau\,,\, \frac{t - \tau_{k+1}}{t - \tau_k}\beta(x_{\tau_k}, c)\Delta\tau\right),$$

which can be sampled for $k = 0, 1, \ldots, m-2$. We also note in the case of error-free observation of all components of X_t (as is considered in this application), the particle filter of Section 5.1.1 reduces to a simple importance sampler. Consequently, the sorting and resampling steps of Algorithm 7 are not required here.

We took independent $N(0, 10^2)$ priors for $\log(c_1)$ and $\log(c_2)$, and determined an appropriate discretisation level by performing short runs of MIS with $\Delta\tau \in \{0.05, 0.1, 0.2, 05\}$. Since there was very little difference in posteriors beyond $\Delta\tau = 0.2$, we used this value in the main monitoring runs which consisted of 2×10^4 iterations of MIS, CPMMH and PMMH. The results are summarised by Figures 5.1–5.2 and Table 5.1.

Posterior samples are consistent with the true values that produced the data, despite using the CLE (rather than the MJP from which the data were generated) as an inferential model. Table 5.1 shows a comparison of each competing inference scheme. As the tuning advice in Section 5.1.3 suggests, CPMMH can tolerate much smaller values of N than PMMH, with the scheme only requiring a value of N around 2 (and we report results for $N = 1, 2$) when $\rho = 0.99$ compared to $N = 50$ for PMMH. Moreover, we found that the PMMH scheme often exhibited 'sticky' behaviour, resulting in relatively low effective sample sizes. Consequently, in terms of minimum ESS per second, CPMMH with $\rho = 0.99$

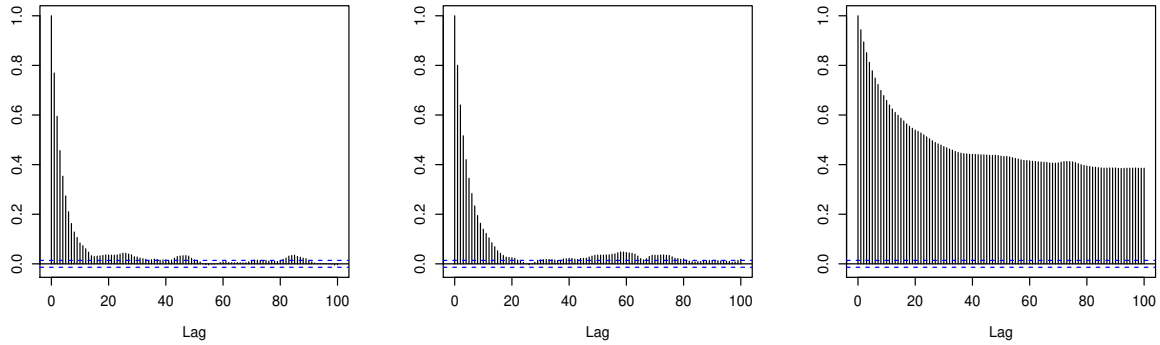

Figure 5.2: Immigration death model. Correlogram based on $\log(c_2)$ samples from the output of MIS (left panel), CPMMH with $\rho = 0.99$ (middle panel) and PMMH (right panel).

Algorithm	ρ	N	CPU (s)	mESS	mESS/s	Rel.
MIS	–	–	121	2190	18	90
CPMMH	0.99	1	45	1910	42	210
	0.99	2	78	2370	30	150
	0.90	1	45	820	18	90
PMMH	0	50	1740	380	0.2	1

Table 5.1: Immigration death model. Correlation parameter ρ, number of particles N, CPU time (in seconds), minimum ESS, minimum ESS per second and relative (to PMMH) minimum ESS per second. All results are based on 2×10^4 iterations of each scheme.

and $N = 1$ outperforms PMMH by a factor of 210, reducing to 150 when $N = 2$.

As noted by Deligiannidis *et al.* (2018), values of ρ close to 1 can result in slow mixing of the auxiliary variables U, in turn giving parameter correlograms that exhibit long range dependence. This does not appear to be the case for $\rho = 0.99$ (see middle panel of Figure 5.2). Nevertheless, we note that reducing ρ to 0.9 still gives an increase in overall efficiency of almost two orders of magnitude over PMMH. Finally, we compare CPMMH to the modified innovation scheme. We obtain similar ESS values between the two schemes for $\rho = 0.99$. However, the relatively low computational cost of CPMMH for these parameter choices results in an improvement in overall efficiency (with an mESS/s of 42 vs 18 for $N = 1$, or 30 vs 18 for $N = 2$).

5.2.2 Lotka-Volterra model

Recall the Lotka-Volterra system introduced in Section 3.4.2. We generated a single realisation of the jump process at 51 integer times via Algorithm 3 with rate constants as in Boys *et al.* (2008), that is $c = (0.5, 0.0025, 0.3)'$ and an initial condition of $x_0 = (100, 100)'$.

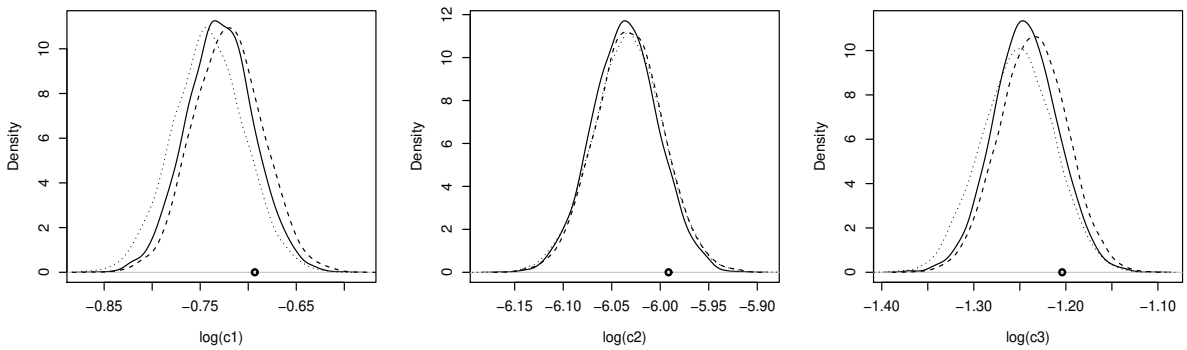

Figure 5.3: Lotka-Volterra model. Marginal posterior distributions based on the output of CPMMH ($\rho = 0.99$) using data sets $\mathcal{D}_1$ (solid lines), $\mathcal{D}_2$ (dashed lines) and $\mathcal{D}_3$ (dotted lines). The true values of $\log(c_1)$, $\log(c_2)$ and $\log(c_3)$ are indicated.

Algorithm	N	CPU (s)	mESS	mESS/s	Rel.
$\mathcal{D}_1$ ($\sigma = 1$)					
MIS	–	14697	9218	0.627	13.5
CPMMH	3	11278	8023	0.711	16.3
PMMH	16	59730	2771	0.046	1.0
$\mathcal{D}_2$ ($\sigma = 5$)					
MIS	–	14598	8139	0.558	14.3
CPMMH	8	29779	3681	0.124	3.2
PMMH	20	75929	2959	0.039	1.0
$\mathcal{D}_3$ ($\sigma = 10$)					
MIS	–	14690	6436	0.438	15.3
CPMMH	19	71524	3516	0.049	1.7
PMMH	28	105770	3031	0.029	1.0

Table 5.2: Lotka-Volterra model. Number of particles N, CPU time (in seconds), minimum ESS, minimum ESS per second and relative (to PMMH) minimum ESS per second. All results are based on 10^5 iterations of each scheme.

We then obtained 3 data sets by corrupting the system state according to

$$Y_t \sim \mathrm{N}\left(X_t, \sigma^2 I_{2\times 2}\right)$$

where $I_{2\times 2}$ is the 2×2 identity matrix and $\sigma \in \{1, 5, 10\}$ giving data sets designated as $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ respectively. We took independent $N(0, 10^2)$ priors for each $\log(c_i)$, $i = 1, 2, 3$, and followed Golightly & Wilkinson (2011) by setting $\Delta\tau = 0.2$. The main monitoring runs consisted of 10^5 iterations of MIS, CPMMH (with $\rho = 0.99$) and PMMH. The results are summarised in Figure 5.3 and Table 5.2.

Figure 5.3 shows that posterior samples are consistent with the true values that produced the data, despite using an approximate inferential model (the CLE). Table 5.2

shows a comparison of each competing inference scheme. When using data set $\mathcal{D}_1$ ($\sigma = 1$), CPMMH outperforms PMMH by an order of magnitude (in terms of overall efficiency) and compares favourably with MIS. However, it is clear that as the measurement error standard deviation (σ) increases, PMMH and CPMMH require more particles, in order to effectively integrate over increasing uncertainty in the observation process. Consequently, MIS outperforms PMMH and CPMMH when using $\mathcal{D}_2$ ($\sigma = 5$) and $\mathcal{D}_3$ ($\sigma = 10$), although the relative difference is less than an order of magnitude for MIS vs CPMMH. It is worth noting that the rate of increase in N is greater for CPMMH than for PMMH. Increasing σ appears to break down the correlation between successive estimates of the log-posterior. Fixing the parameter values at the posterior mean and estimating the correlation, denoted by ρ_l, between $\hat{p}_u(\mathcal{D}|c)$ and $\hat{p}_{u^*}(\mathcal{D}|c)$ gave $\rho_l = 0.97$ for $\mathcal{D}_1$, $\rho_l = 0.91$ for $\mathcal{D}_2$ and $\rho_l = 0.57$ for $\mathcal{D}_3$. Nevertheless, we still observe a worthwhile increase in overall efficiency of a factor of 2 for CPMMH vs PMMH, when using data set $\mathcal{D}_3$ corresponding to the relatively extreme $\sigma = 10$.

5.2.3 Autoregulatory network

In this section, we consider a simple autoregulatory network with two species, $\mathcal{X}_1$ and $\mathcal{X}_2$ whose time course behaviour evolves according to the following set of coupled reactions,

$$\begin{aligned}
\mathcal{R}_1 &: \emptyset \xrightarrow{c_1} \mathcal{X}_1 \\
\mathcal{R}_2 &: \emptyset \xrightarrow{c_2} \mathcal{X}_2 \\
\mathcal{R}_3 &: \mathcal{X}_1 \xrightarrow{c_3} \emptyset \\
\mathcal{R}_4 &: \mathcal{X}_2 \xrightarrow{c_4} \emptyset \\
\mathcal{R}_5 &: \mathcal{X}_1 + \mathcal{X}_2 \xrightarrow{c_5} 2\mathcal{X}_2
\end{aligned}$$

Essentially, reactions R_1 and R_2 represent immigration and reactions R_3 and R_4 represent death. The species interact via R_5. Let $X_t = (X_{1,t}, X_{2,t})'$ denote the system state at time t. The stoichiometry matrix associated with the system is given by

$$S = \begin{pmatrix} 1 & 0 & -1 & 0 & -1 \\ 0 & 1 & 0 & -1 & 1 \end{pmatrix}$$

and the associated hazard function is

$$h(X_t, c) = (c_1, c_2, c_3 X_{1,t}, c_4 X_{2,t}, c_5 X_{1,t} X_{2,t})'.$$

We simulated a single realisation of the jump process at 101 integer times via Algorithm

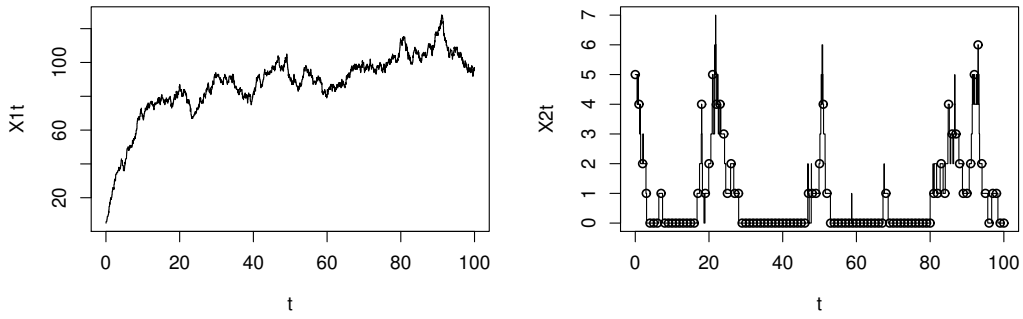

Figure 5.4: Autoregulatory network. A single realisation of the jump process with $c = (10, 0.1, 0.1, 0.7, 0.008)'$ and $X_0 = (5, 5)'$. Observations are indicated by circles.

Algorithm	N	CPU (s)	mESS	mESS/s	Rel.
CPMMH (Diffusion bridge)	20	15575	1272	0.082	6.2
PMMH (Diffusion bridge)	55	42014	1302	0.031	2.4
PMMH (Myopic)	200	95802	1263	0.013	1

Table 5.3: Autoregulatory network. Number of particles N, CPU time (in seconds), minimum ESS, minimum ESS per second and relative (to myopic filter driven PMMH) minimum ESS per second. All results are based on 10^5 iterations of each scheme.

3 with rate constants $c = (10, 0.1, 0.1, 0.7, 0.008)'$ and an initial condition of $X_0 = (5, 5)'$. We then discarded the values of $X_{1,t}$ to leave observations of $X_{2,t}$ only. The full data trace used to generate the data set is given in Figure 5.4. The inherently discrete nature of the data set coupled with long time periods where $X_{2,t} = 0$ make applying the CLE impractical. We therefore use the Poisson leap approximation as the inferential model. To provide a challenging scenario, we assume error-free observation of $X_{2,t}$ so that step 2(d) of Algorithm 7 assigns a weight of 0 to the particle x_t^i unless $x_{2,t}^i$ coincides with the observation at time t. We took a weakly informative Gamma$(10, 1)$ prior for c_1 and Gamma$(0.1, 0.1)$ priors for the remaining rate constants. We found little difference in sampled posterior values for a value of $\Delta\tau$ beyond 0.2 and therefore used this value in our main monitoring runs which consisted of 10^5 iterations of CPMMH (with $\rho = 0.996$, which we found to work well for the partial observation scenario) and PMMH. We report results for schemes driven by both the myopic particler filter of Gordon *et al.* (1993), and by the diffusion bridge particle filter of Section 5.1.2. The results are summarised in Table 5.3 and Figure 5.5.

Again, we chose the number of particles N by following the practical advice of Tran *et al.* (2016) for CPMMH and Sherlock *et al.* (2015) for PMMH. Inspection of Table 5.3 reveals that the myopic particle filter driven PMMH scheme required $N = 200$ particles.

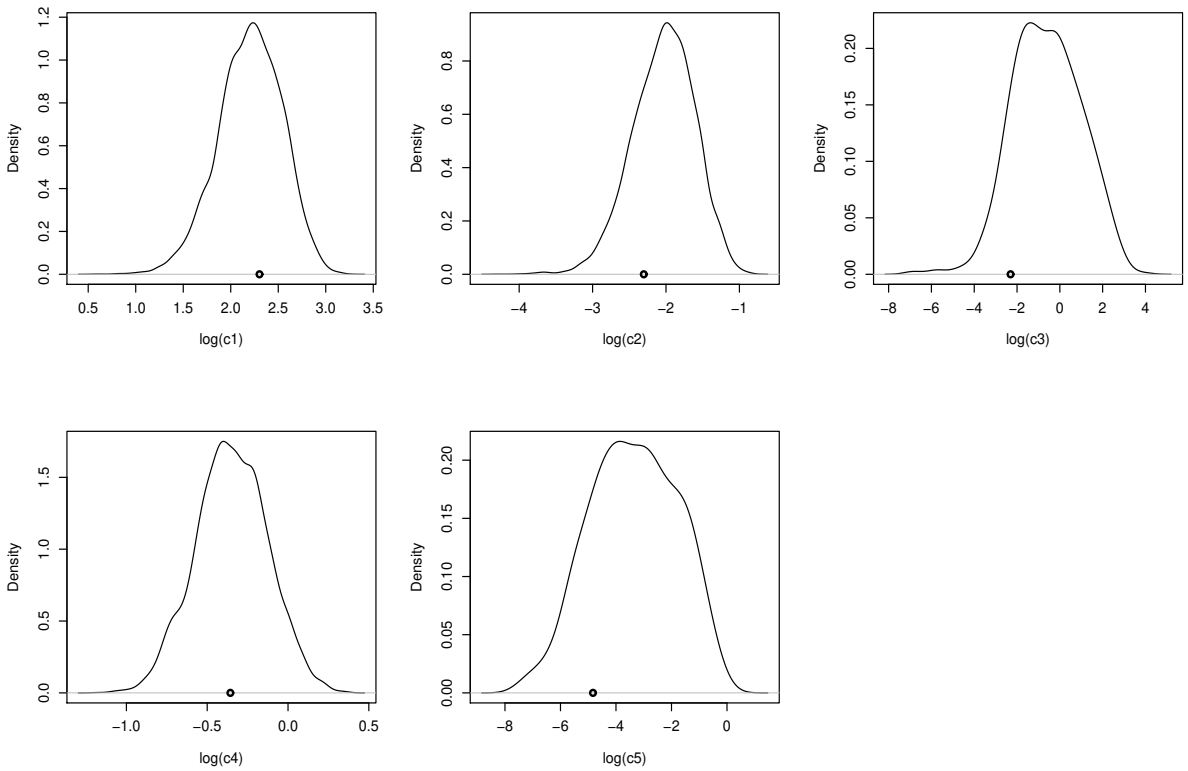

Figure 5.5: Autoregulatory network. Marginal posterior distributions based on the output of CPMMH ($\rho = 0.996$). The true values of $\log(c_i)$, $i = 1, \ldots, 5$, are indicated.

This reduces to $N = 55$ when using the diffusion bridge particle filter, and reduces further still to $N = 20$ when strong and positive correlation is introduced between successive values of the random variables that drive the diffusion bridge particle flter. Despite the diffusion bridge driven scheme requiring many fewer particles than the myopic scheme, overall efficiency (as measured by minimum ESS per second) is only increased by a factor of 2.4 due to the computational complexity of the conditioned hazard, which is used to propagate state particles within the diffusion bridge particle filter. The correlated implementation gives a further increase of a factor of 2.6, giving a 6-fold increase in overall efficiency over the most basic PMMH scheme.

5.2.4 Epidemic model

The Susceptible–Infected–Removed (SIR) epidemic model (see e.g. Andersson & Britton, 2000) describes the evolution of two species (susceptibles $\mathcal{X}_1$ and infectives $\mathcal{X}_2$) via two reaction channels which correspond to an infection of a susceptible individual and a removal

of an infective individual. The reaction equations are

$$\mathcal{R}_1 : \mathcal{X}_1 + \mathcal{X}_2 \xrightarrow{c_1} 2\mathcal{X}_2$$
$$\mathcal{R}_2 : \mathcal{X}_2 \xrightarrow{c_2} \emptyset.$$

The stoichiometry matrix is given by

$$S = \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix}$$

and the associated hazard function is

$$h(X_t, c) = (c_1 X_{1,t} X_{2,t}, c_2 X_{2,t})'.$$

We consider a data set consisting of the daily number of pupils confined to bed (out of a total of 763) during an influenza outbreak in a boys' boarding school in Great Britain, instigated by a single pupil. Hence, $X_0 = (762, 1)'$. The data are displayed graphically in BMJ News and Notes (1978) and converted into counts in Fuchs (2013). For completeness, we give the data in Table 5.4. We work with the CLE which has the form

$$d\begin{pmatrix} X_{1,t} \\ X_{2,t} \end{pmatrix} = \begin{pmatrix} -c_1 X_{1,t} X_{2,t} \\ c_1 X_{1,t} X_{2,t} - c_2 X_{2,t} \end{pmatrix} dt + \begin{pmatrix} c_1 X_{1,t} X_{2,t} & -c_1 X_{1,t} X_{2,t} \\ -c_1 X_{1,t} X_{2,t} & c_1 X_{1,t} X_{2,t} + c_2 X_{2,t} \end{pmatrix}^{1/2} d\begin{pmatrix} W_{1,t} \\ W_{2,t} \end{pmatrix}. \tag{5.14}$$

We further assume that the infection rate is a mean reverting diffusion process governed by the SDE

$$d\log(c_{1,t}) = c_3(c_4 - \log(c_{1,t}))dt + c_5 dW_{3,t}. \tag{5.15}$$

Hence, the inferential model is specified by (5.14) and (5.15), where c_1 is replaced by $c_{1,t}$ in (5.14). We wish to infer $c = (c_2, c_3, c_4, c_5)'$ based on measurements of $X_{2,t}$ only, giving a partially observed system. We took a normal $\mathrm{N}(0, 10^2)$ prior on the reversion level c_4 of $\log(c_{1,t})$, and exponential $\mathrm{Exp}(1)$ priors for the remaining parameters. For simplicity, we fixed the initial unobserved infection rate by taking $\log(c_{1,0}) = -6$. The discretisation level was fixed by taking $\Delta\tau = 0.1$. The main monitoring runs consisted of 2×10^5 iterations of CPMMH and PMMH. The results are summarised in Figure 5.6 and Table 5.5. It is evident that CPMMH outperforms PMMH in terms of overall efficiency (as measured here by minimum ESS per minute) by a factor of 7.

Day	1	2	3	4	5	6	7	8	9	10
No. of infectives	1	3	6	25	73	221	294	257	236	189
Day	11	12	13	14	15					
No. of infectives	125	67	26	10	3					

Table 5.4: Boarding school data.

Algorithm	N	CPU (m)	mESS	mESS/m	Rel.
CPMMH	90	2765	226	0.08	7.2
PMMH	600	26338	299	0.01	1

Table 5.5: Epidemic model. Number of particles N, CPU time (in minutes m), minimum ESS, minimum ESS per minute and relative minimum ESS per minute. All results are based on 2×10^5 iterations of each scheme.

5.2.5 Summary of Application results

In all applications considered in this chapter, CPMMH outperformed its standard PMMH counterpart in terms of overall efficiency. This is due to the lower number of bridges required in the particle filter (or importance sampler in application 5.2.1) to achieve a similar effective sample size, leading to reduced computational effort for a similar standard of output. In the most extreme case, with full, error-free observations, the CPMMH scheme was able to use a single bridge compared to 50 bridges for the PMMH scheme, and the resulting mESS was still five times greater for CPMMH than for PMMH. In such cases, where little to no noise in observations meant that high ESS could be achieved with few bridges, the CPMMH scheme performs favourably compared to a competing scheme, the MIS. However, as noise in the system increases, the performance of CPMMH degrades faster than that of PMMH or MIS, leading it to underperform compared to MIS in high noise scenarios, and outperform PMMH by a smaller margin.

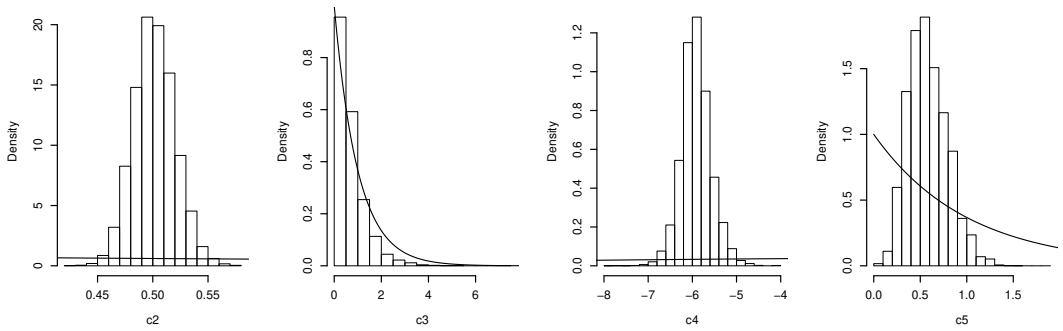

Figure 5.6: Epidemic model. Marginal posterior distributions based on the output of CPMMH (histograms). Prior densities are given by the solid lines.

Chapter 6

Accelerating inference for intractable models using tractable surrogates

The previous two chapters have considered performing Bayesian inference for stochastic kinetic models using both tractable and intractable inferential models. Using a tractable approximation to the MJP, such as the LNA, as an inferential model is significantly more computationally efficient, but may not be suitably accurate in practice. Using a more accurate, but intractable, approximation such as the CLE or Poisson leap method is often desirable, but may come with a large computational overhead. This chapter therefore considers methods of using the LNA as a tractable surrogate likelihood in order to improve either the statistical or computational efficiency of inference schemes utilising an intractable inferential model. Firstly, we use the LNA in the first stage of a delayed acceptance scheme. We then consider the solution of some or all of the ODEs governing the LNA for use in improved bridge constructs within the particle filter. We then discuss how the use of several of the techniques used in the thesis can be applied in tandem to further increase computational savings.

6.1 Delayed acceptance pseudo-marginal Metropolis Hastings using the LNA

Consider now the particle MCMC scheme of Section 5.1 targeting the joint posterior $\hat{\pi}(c, u|\mathcal{D})$ in (5.4) for which $\pi(c|\mathcal{D})$ is a marginal. Whilst a particle filter is useful for constructing an unbiased estimator of the observed data likelihood, it can be computationally expensive, particularly when many particles are required to keep the variance of the estimator low. Therefore, we would ideally like to avoid running the particle filter to compute

$\hat{p}_{u^*}(\mathcal{D}|c^*)$ if c^* is likely to be rejected. This motivates the use of a screening step, whereby the particle filter is only run for proposals accepted under the surrogate posterior. This is known as *delayed acceptance* (DA). This method was first proposed by Christen & Fox (2005), and applied to stochastic kinetic models by Golightly *et al.* (2015).

For a given iteration with current state (c, u), Stage One of the DA scheme proposes $c^* \sim q(\cdot|c)$, computes $p_a(\mathcal{D}|c^*)$ using a surrogate likelihood such as the LNA (as discussed in Section 4.1) and runs a M-H screening step with acceptance probability

$$\alpha_1\left(c^*|c\right) = \min\left\{1, \frac{\pi(c^*)p_a(\mathcal{D}|c^*)}{\pi(c)p_a(\mathcal{D}|c)} \times \frac{q(c|c^*)}{q(c^*|c)}\right\}. \tag{6.1}$$

If this screening step is successful, Stage Two of the DA scheme is to propose $u^* \sim g(\cdot)$, construct the estimate $\hat{p}_{u^*}(\mathcal{D}|c^*)$ using a particle filter, and the Stage Two acceptance probability

$$\begin{aligned}\alpha_{2|1}\left\{(c^*, u^*)|(c, u)\right\} &= \min\left\{1, \frac{\pi(c^*)\hat{p}_{u^*}(\mathcal{D}|c^*)}{\pi(c)\hat{p}_u(\mathcal{D}|c)} \times \frac{\pi(c)p_a(\mathcal{D}|c)}{\pi(c^*)p_a(\mathcal{D}|c^*)}\right\} \\ &= \min\left\{1, \frac{\hat{p}_{u^*}(\mathcal{D}|c^*)}{\hat{p}_u(\mathcal{D}|c)} \times \frac{p_a(\mathcal{D}|c)}{p_a(\mathcal{D}|c^*)}\right\}. \end{aligned} \tag{6.2}$$

Thus the overall acceptance probability for the scheme is

$$\alpha_{\mathrm{DA}}\left\{(c^*, u^*)|(c, u)\right\} = \alpha_1\left(c^*|c\right)\alpha_{2|1}\left\{(c^*, u^*)|(c, u)\right\}. \tag{6.3}$$

This algorithm, known as delayed acceptance CPMMH (DA-CPMMH) is shown in Algorithm 8.

In much the same way as we showed that a CPMMH scheme satisfied detailed balance in section 5.1, we can show that detailed balance is satisfied for daCPMMH. Upon negating the trivial scenario that the chain does not move, we have that

$$\begin{aligned}&\tilde{\pi}(c, u|\mathcal{D})q(c^*|c)K(u^*|u)\alpha_{\mathrm{DA}}\left\{(c^*, u^*)|(c, u)\right\} \\ &\quad= \min\left\{\pi(c)g(u)\hat{p}_u(\mathcal{D}|c)q(c^*|c)K(u'|u)\,,\, \pi(c)g(u)\hat{p}_u(\mathcal{D}|c)q(c^*|c)K(u^*|u)\right. \\ &\qquad\qquad\left.\times\frac{\pi(c^*)p_a(\mathcal{D}|c^*)q(c|c^*)}{\pi(c)p_a(\mathcal{D}|c)q(c^*|c)} \times \frac{\hat{p}_{u^*}(\mathcal{D}|c^*)p_a(\mathcal{D}|c)}{\hat{p}_u(\mathcal{D}|c)p_a(\mathcal{D}|c^*)}\right\} \\ &\quad= \min\left\{\pi(c)g(u)\hat{p}_u(\mathcal{D}|c)q(c^*|c)K(u^*|u)\,,\, \pi(c^*)g(u)\hat{p}_{u^*}(\mathcal{D}|c^*)q(c|c^*)K(u^*|u)\right\} \\ &\quad= \min\left\{\pi(c)g(u)\hat{p}_u(\mathcal{D}|c)q(c^*|c)K(u^*|u)\,,\, \pi(c^*)g(u^*)\hat{p}_{u^*}(\mathcal{D}|c^*)q(c|c^*)K(u|u^*)\right\} \\ &\quad= \tilde{\pi}(c^*, u^*|\mathcal{D})q(c|c^*)K(u|u^*)\alpha_{\mathrm{DA}}\left\{(c, u)|(c^*, u^*)\right\}, \end{aligned}$$

where again (5.6) is used to deduce the third line.

As mentioned in Christen & Fox (2005), a delayed acceptance algorithm will always be less statistically efficient than an equivalent scheme that does not employ delayed accep-

Algorithm 8 Delayed acceptance correlated PMMH scheme (DA-CPMMH)

1. Initialisation. For $i = 0$:
   (a) Set $c^{(0)}$ in the support of $\pi(c|\mathcal{D})$ and draw $u^{(0)} \sim \mathrm{N}(0, I_d)$.
   (b) Compute $p_a(\mathcal{D}|c^{(0)})$ by running Algorithm 4 with $c = c^{(0)}$.
   (c) Compute $\hat{p}_{u^{(0)}}(\mathcal{D}|c^{(0)})$ by running Algorithm 7 with $(c, u) = (c^{(0)}, u^{(0)})$.
2. For iteration $i \geq 1$:
   (a) Draw $c^* \sim q(\cdot|c^{(i-1)})$ and $\omega \sim \mathrm{N}(0, I_d)$. Put $u^* = \rho u^{(i-1)} + \sqrt{1-\rho^2}\omega$.
   (b) **Stage 1**
      (i) Compute $p_a(\mathcal{D}|c^*)$ by running Algorithm 4 with $c = c^*$.
      (ii) With probability $\alpha(c^*|(c^{(i-1)})$ given by (6.1), compute $\hat{p}_{u^*}(\mathcal{D}|c^*)$ by running Algorithm 7 with $(c, u) = (c^*, u^*)$ and go to step 2(c); otherwise store the current values $(c^{(i)}, u^{(i)}) = (c^{(i-1)}, u^{(i-1)})$, increment i and go to step 2(a).
   (c) **Stage 2**
   With probability $\alpha\left\{(c^*, u^*)|(c^{(i-1)}, u^{(i-1)})\right\}$ given by (6.2), put $(c^{(i)}, u^{(i)}) = (c^*, u^*)$ otherwise store the current values $(c^{(i)}, u^{(i)}) = (c^{(i-1)}, u^{(i-1)})$. Increment i and go to step 2(a).

tance (unless the Stage 1 acceptance rate is 1, which would render the delayed acceptance step redundant). Essentially this is because some proposals that may have been accepted at Stage 2 will be rejected at Stage 1. This means that, for a given number of iterations, a delayed acceptance scheme will generally have a lower ESS than the non-delayed acceptance equivalent. Thus, in order to be more efficient overall, the computational savings made by the scheme must outweigh the loss in statistical efficiency. As noted by Christen & Fox (2005), this is most likely when the acceptance rate is low, and when the computational cost of the approximation is negligible compared to the cost of the Stage 2 calculation. This concept has been formalised by Sherlock *et al.* (2021), who provide tuning advice and optimal acceptance rates for pseudo-marginal delayed acceptance schemes, which differ depending on how many orders of magnitude cheaper it is to calculate Stage 1 than Stage 2.

6.2 Improved Bridge constructs

As discussed in Section 5.1.2, the MDB construct guides particles towards observations in a linear fashion. Thus, if the underlying stochastic process exhibits nonlinear dynamics between observations, the MDB will fail to adequately capture these dynamics. This can lead to few particles having a reasonable weight in the particle filter, necessitating a much

larger number of particles to keep the variance of the estimator to a reasonable level. At worst, parameter choices that lead to accurate dynamics of the underlying process may be rejected as the bridge construct does not recreate these dynamics, leading to inaccurate inference. The MDB is therefore inadequate when dealing with such inter-observation nonlinearity, and so another class of bridge constructs is required for time-discretised approximations to the MJP.

As in Section 5.1.2, consider without loss of generality a time interval $(0, T]$ partitioned as

$$0 = \tau_0 < \tau_1 < \ldots < \tau_{m-1} < \tau_m = T, \quad \Delta\tau = \frac{1}{m}.$$

Whitaker *et al.* (2017*b*) (see also Botha *et al.*, 2021) propose a class of bridge constructs known as residual bridge (RB) constructs. These involve partitioning X_t as $X_t = \eta_t + R_t$, for a deterministic process $\{\eta_t, t \geq 0\}$ satisfying

$$\frac{d\eta_t}{dt} = \alpha(\eta_t), \quad \eta_0 = x_0, \tag{6.4}$$

and a residual stochastic process $\{R_t, t \geq 0\}$ satisfying

$$dR_t = \{\alpha(X_t) - \alpha(\eta_t)\}dt + \sqrt{\beta(X_t)}dW_t. \tag{6.5}$$

We can then solve (6.4) (either directly or using an ODE solver), and construct the MDB for the residual stochastic process. To do this, note that our partition of X_T can be substituted into the observation equation (4.1) and rearranged to obtain

$$Y_T - P'\eta_T = P'R_t + \varepsilon_T.$$

Thus, $Y_T - P'\eta_T$ is a partial, noisy observation of R_T, and so we can approximate the joint distribution of $R_{\tau_{k+1}}$ and $Y_T - P'\eta_T$ given r_{τ_k} in the same manner as Section 5.1.2 to obtain

$$\begin{pmatrix} R_{\tau_{k+1}} \\ Y_T - P'\eta_T \end{pmatrix} \Bigg| r_{\tau_k} \sim N\left(\begin{pmatrix} r_{\tau_k} + (\alpha_k - \alpha_k^\eta)\Delta\tau \\ P'(r_{\tau_k} + (\alpha_k - \alpha_k^\eta)\Delta_k) \end{pmatrix}, \begin{pmatrix} \beta_k\Delta\tau & \beta_k P\Delta\tau \\ P'\beta_k\Delta\tau & P'\beta_k P\Delta_k + \Sigma \end{pmatrix}\right),$$

where $\alpha_k^\eta = \alpha(\eta_{\tau_k}) = S\,h(\eta_{\tau_k}, c)$. Recall from Section 5.1.2 that $\alpha_k = S\,h(x_{\tau_k}, c)$, $\beta_k = S\ \mathrm{diag}\{h(x_{\tau_k}, c)\}S'$, and $\Delta_k = T - \tau_k$. We follow Whitaker *et al.* (2017*b*) by replacing α_k^η with δ_k^η to approximate $d\eta_t/dt$, where

$$\delta_k^\eta = \frac{\eta_{\tau_{k+1}} - \eta_{\tau_k}}{\Delta\tau}.$$

Making this replacement gives

$$\begin{pmatrix} R_{\tau_{k+1}} \\ Y_T - P'\eta_T \end{pmatrix} \Bigg| r_{\tau_k} \sim N\left(\begin{pmatrix} r_{\tau_k} + (\alpha_k - \delta_k^\eta)\Delta\tau \\ P'(r_{\tau_k} + (\alpha_k - \delta_k^\eta)\Delta_k) \end{pmatrix}, \begin{pmatrix} \beta_k \Delta\tau & \beta_k P \Delta\tau \\ P'\beta_k \Delta\tau & P'\beta_k P \Delta_k + \Sigma \end{pmatrix} \right).$$

Conditioning on $Y_T - P'\eta_T$ as in Section 5.1.2 gives an equation for $R_{\tau_{k+1}}|y_T, r_{\tau_k}$, which can then be simplified when obtaining the corresponding equation for $X_{\tau_{k+1}}|y_T, r_{\tau_k}$ by noting that

$$X_{\tau_{k+1}} = \eta_{\tau_{k+1}} + R_{\tau_{k+1}} = \eta_{\tau_k} + \delta_k^\eta \Delta\tau + R_{\tau_{k+1}}.$$

The $\delta_k^\eta \Delta\tau$ term above cancels with the term in the expectation of $R_{\tau_{k+1}}$, and the η_{τ_k} term combines with the r_{τ_k} term in the expectation to give x_{τ_k}. Thus we obtain the simple residual bridge construct, henceforth referred to as RB, which takes the form of (5.11) but with $\mu_{\text{MDB}}(x_{\tau_k}, c)$ replaced with

$$\mu_{\text{RB}}(x_{\tau_k}, c) = \alpha_k + \beta_k P \left(P'\beta_k P \Delta_k + \Sigma\right)^{-1} \left[y_T - P' \left\{\eta_T + r_{\tau_k} + \left(\alpha_k - \delta_k^\eta\right) \Delta_k\right\}\right], \quad (6.6)$$

A more advanced residual bridge construct, henceforth referred to as RB$^-$, can be found by further partitioning X_t as $X_t = \eta_t + \hat{\rho}_t + R_t^-$, where $\hat{\rho}_t$ is the expectation of the approximate conditioned residual process, that is $\hat{\rho}_t = \text{E}(\hat{R}_t|r_0, y_T)$, and $\{R_t^-, t \in [0,T]\}$ is the additional residual stochastic process arising from this new partition. Our approximate conditioned residual process $\{\hat{R}_t, t \in [0,T]\}$ can be found using the LNA. For a general $t \in [0,T]$ we can extend the representation of the LNA given in (3.15) to

$$\hat{R}_T|\hat{R}_t \sim N\left(G_{T|t}\hat{R}_t, V_{T|t}\right), \quad (6.7)$$

where $G_{T|t}$ and $V_{T|t}$ are as defined in Section 5.1.2. Using (5.9), we can rewrite the expectation for (6.7) as

$$\text{E}\left(\hat{R}_T|\hat{R}_t\right) = G_T G_t^{-1} \text{E}\left(\hat{R}_t\right) = G_T \hat{R}_0. \quad (6.8)$$

As $\hat{R}_T|\hat{R}_t$ can be written as a linear combination of $G_{T|t}\hat{R}_t$ and some independent noise, we can determine the covariance of $\hat{R}_T$ and $\hat{R}_t$ to be

$$\begin{aligned} \text{Cov}\left(\hat{R}_T, \hat{R}_t\right) &= \text{Cov}\left(G_{T|t}\hat{R}_t, \hat{R}_t\right) && (6.9) \\ &= G_{T|t} \text{Var}\left(\hat{R}_t\right) && (6.10) \\ &= G_T G_t^{-1} V_t. && (6.11) \end{aligned}$$

As $Y_T - P'\eta_T$ can be seen as a partial, noisy observation of $\hat{R}_T$, we can use (6.8) and (6.11)

to construct the joint distribution of $\hat{R}_t$ and $Y_T - P'\eta_T$ conditional on $\hat{r}_0$ as

$$\begin{pmatrix} \hat{R}_t \\ Y_T - P'\eta_T \end{pmatrix} \Bigg| \hat{r}_0 \sim N\left(\begin{pmatrix} G_t\hat{r}_0 \\ P'G_T\hat{r}_0 \end{pmatrix}, \begin{pmatrix} V_t & V_t\{G_t^{-1}\}'G_T'P \\ P'G_TG_t^{-1}V_t & P'V_TP + \Sigma \end{pmatrix} \right).$$

Note that for a known X_0, $\hat{r}_0 = r_0$. Thus, as $\hat{\rho}_t = \mathrm{E}(\hat{R}_t | r_0, y_T)$ we can condition on $Y_T - P'\eta_T$ to obtain

$$\hat{\rho}_t = G_t r_0 + V_t\{G_t^{-1}\}'G_T'P\left(P'V_T'P + \Sigma\right)^{-1}\left(y_T - P'(\eta_T + G_T r_0)\right). \tag{6.12}$$

In general $r_0 = 0$, and so (6.12) simplifies to

$$\hat{\rho}_t = V_t\{G_t^{-1}\}'G_T'P\left(P'V_T'P + \Sigma\right)^{-1}\left(y_T - P'\eta_T\right).$$

We can then construct an approximate joint distribution for the further residual process $R^-_{\tau_{k+1}}$ and $Y_T - P'\eta_T$, and condition further on the observation in an analogous manner to the construction of the RB construct. This leads to the RB$^-$ construct, whereby we replace $\mu_{\mathrm{MDB}}(x_{\tau_k}, c)$ in (5.11) with

$$\mu_{\mathrm{RB}^-}(x_{\tau_k}, c) = \alpha_k + \beta_k P\left(P'\beta_k P\Delta_k + \Sigma\right)^{-1}\left[y_T - P'\left\{\eta_T + \hat{\rho}_T + r^-_{\tau_k} + \left(\alpha_k - \delta_k^\eta - \delta_k^\rho\right)\Delta_k\right\}\right], \tag{6.13}$$

where

$$\delta_k^\rho = \frac{\rho_{\tau_{k+1}} - \rho_{\tau_k}}{\Delta\tau},$$

and δ_k^η is as before. Note that this construct requires the solution of a larger ODE system than other implementations of the LNA, since the ODEs governing G_t must be explicitly solved. Recall that in general we avoid solving these ODEs through the use of restarting the LNA, as mentioned in Section 3.3.3.

Although not considered in this thesis, it should be noted that for applications where the system is observed fully and without error, both the RB and RB$^-$ constructs simplify considerably. For RB, $\mu_{\mathrm{RB}}(x_{\tau_k}, c)$ becomes

$$\mu^*_{\mathrm{RB}}(x_{\tau_k}, c) = \delta_k^\eta + \frac{x_T - \eta_T - (x_{\tau_k} - \eta_{\tau_k})}{\Delta_k},$$

and for RB$^-$, $\mu_{\mathrm{RB}^-}(x_{\tau_k}, c)$ becomes

$$\mu^*_{\mathrm{RB}}(x_{\tau_k}, c) = \delta_k^\eta + \delta_k^\rho + \frac{x_T - \eta_T - \hat{\rho}_k - (x_{\tau_k} - \eta_{\tau_k} - \hat{\rho}_{\tau_k})}{\Delta_k}.$$

We can compare the accuracy of the MDB, RB and RB$^-$ constructs by using them to propose conditioned paths between two observations from an SKM. To illustrate this, we

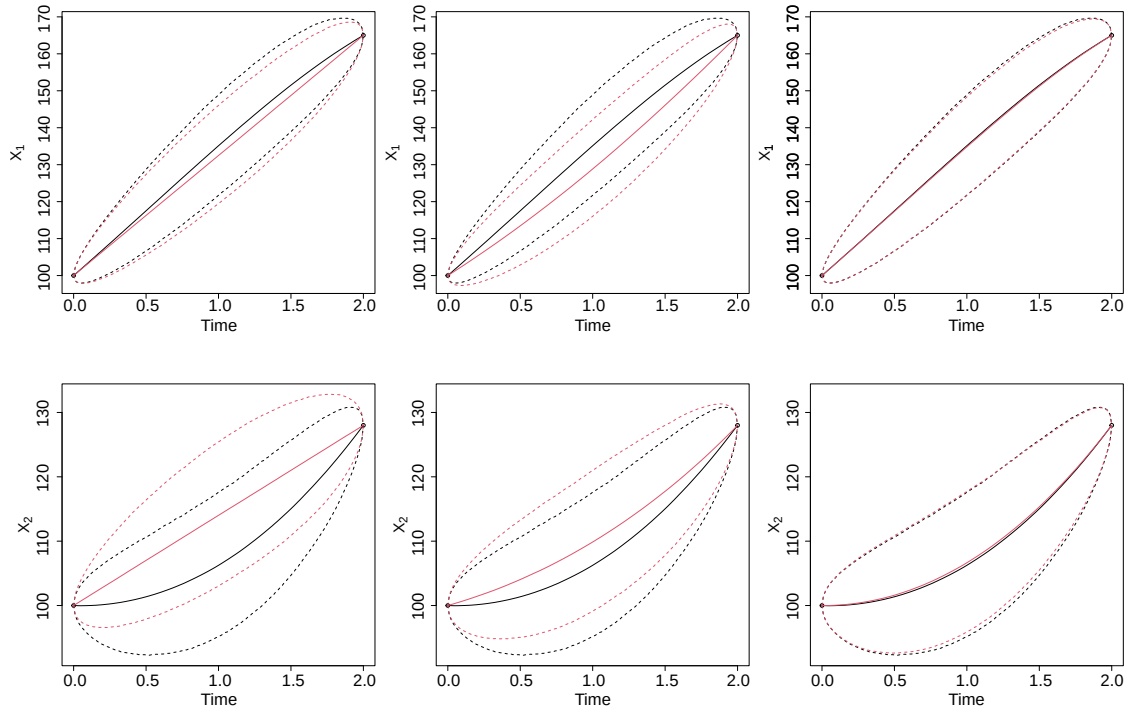

Figure 6.1: 95% credible region (dashed lines) and mean (solid lines) of the Lotka-Volterra model. Black lines are the true conditioned process; red lines are bridge constructs. Top row: prey component; bottom row: predator component. Left: MDB; middle: RB; right: RB$^-$.

shall use the Lotka-Volterra model - for full details of this model, see Section 3.4.2. Using the parameters of Boys *et al.* (2008), that is $c = (0.5, 0.0025, 0.3)'$, we simulated a path from $x_0 = (100, 100)'$ to time $T = 2$ using Algorithm 3. We treated the observation x_2 as very informative and so use the observation equation

$$Y_2 \sim N(X_2, 0.01^2 I_2),$$

where I_2 is the 2×2 identity matrix. We then proposed 10^5 bridges between x_0 and x_2 using the MDB, RB and RB$^-$ constructs with $m = 200$ intermediate time points. Figure 6.1 compares the means and 95% credible regions of these bridges for the predator component with those of the "true" conditioned process, obtained using weighted resampling. It can be seen that the MDB pushes the path towards the observation in a linear fashion, and does not capture the dynamics of the true conditioned process. The simple residual bridge, RB, is a modest improvement, and begins to capture these dynamics, whilst the RB$^-$ construct captures the dynamics very effectively, with paths very close to the true conditioned process.

6.3 Combining techniques

By combining the techniques presented so far in this thesis, we can construct a unified inference framework that simultaneously aims to avoid unnecessary calculations of $\hat{p}_{u^*}(\mathcal{D}|c^*)$, reduce the variance of the likelihood estimator for a given N and use a parameter proposal mechanism informed by an approximation of the marginal posterior density.

Deriving the gradient of the $\log(\pi(c|\mathcal{D}))$ for use with MALA is challenging when using the CLE, Poisson leap, or MJP as the inferential model, due to the intractable observed data likelihood $p(\mathcal{D}|c)$. It is possible to estimate the gradient directly from the inferential model (see e.g. Poyiadjis *et al.*, 2011; Nemeth *et al.*, 2016). However, the behaviour of the algorithm in these methods depends heavily on the number of particles used to estimate the gradient. Instead, we estimate the gradient using the surrogate LNA model, as in Section 4.2, and note that the additional approximation used in the proposal will be corrected for in the Metropolis-Hastings step.

Use of the surrogate LNA model in a delayed acceptance step, the MALA parameter proposal and to construct bridge proposals inside the particle filter each require the solution of an ODE system. However, there is some overlap in the ODE components that must be solved to perform each technique, and as such, if implemented correctly, further computational savings can be made when using several of these techniques at once.

Computing the observed data likelihood under the LNA for use in a delayed acceptance step requires the solution of (3.8) and (3.17), restarted at the posterior mean and variance given by the forward filter at each observation time. Computing the gradient information to use full MALA requires the solution of (3.8) and (3.17), as well as the first order sensitivities $\partial\mu(c,t)/\partial c_i$ and $\partial\Psi(c,t)/\partial c_i$ for $i = 1, \ldots, r$. The gradient information using simplified MALA does not require the solution of the $\partial\Psi(c,t)/\partial c_i$. The simple residual bridge, RB, requires only the solution of (3.8). The residual bridge with additional subtraction, RB^-, and conditioned hazard, CH, require the solution of (3.8), (3.11) and (3.17).

As we can see, all of these techniques require the solution of (3.8) and the majority also require the solution of (3.17). Thus, it is desirable to solve these ODEs *once per (C)PMMH iteration* and use the output in several different techniques. Running the forward filter to obtain the surrogate likelihood used in delayed acceptance also solves several of the ODE components used in determining the gradient of the log posterior for MALA. Care must be taken when implementing the RB^- bridge construct, which, for an arbitrary observation interval $[t_i, t_{i+1}]$ and time $t \in (t_i, t_{i+1}]$, requires the LNA variance $V_{t|t_i}$ initialised at 0_s to calculate $\hat{\rho}_t$, whereas the forward filter restarts this variance at the filtering mean B_i (see Algorithm 4). This "disconnect" is alleviated via (5.10) which we may write as

$$V_{t|t_i} = V_t - G_t V_{t_i} G_t',$$

	(C)PMMH	da(C)PMMH	Simplified MALA	Full MALA
RB_{iter}	$O(s)$	$+O(s^2)$	$+O(sr)$	$+O(s^2r)$
$\text{RB}^{-}_{\text{iter}}$	$O(s^2)$	$-$	$+O(sr)$	$+O(s^2r)$
RB_{part}	$O(sN)$	$+O(s^2)$	$+O(sr)$	$+O(s^2r)$
$\text{RB}^{-}_{\text{part}}$	$O(s^2N)$	$+O(s^2)$	$+O(sr)$	$+O(s^2r)$

Table 6.1: Order of complexity in terms of ODE components required to be solved for different bridge construct implementations, and the additional computational cost required to enact delayed acceptance, simplified or full MALA. Note that N, s and r denote the number of particles, species and parameters respectively.

where V_t and G_t are obtained from the forward filter. We denote the resulting bridge constructs in this setting by RB_{iter}, $\text{RB}^{-}_{\text{iter}}$ and CH_{iter}. The accuracy of the bridges over $[t_i, t_{i+1}]$ can be improved by re-integrating the ODE system given by (3.8) and (3.17) *for each particle* $x^{(k)}_{t_i}$. That is, η_{t_i} is set at $x^{(k)}_{t_i}$ and $V_{t_i} = 0_s$. We denote the resulting bridge constructs by RB_{part}, $\text{RB}^{-}_{\text{part}}$ and CH_{part}. Although use of the latter compared to the "once per iteration" approach is likely to result in an estimator of observed data likelihood with lower variance and in turn, better mixing of the (C)PMMH scheme, it comes with an additional computational cost. Given s species and N particles, "once per particle" bridges require the solution of an additional sN ODE components. Table 6.1 shows the relative computational complexity (in terms of the number of ODE components that must be solved) for different acceleration techniques. Note that CH_{iter} and CH_{part} have the same computational complexities as $\text{RB}^{-}_{\text{iter}}$ and $\text{RB}^{-}_{\text{part}}$.

6.3.1 Tuning

Schemes employing CPMMH require specification of a correlation parameter ρ, and irrespective of the acceleration technique employed, all schemes require specification of several other tuning parameters. These include a number of particles N, a preconditioning matrix Σ_T and scaling parameter λ, with the latter two tuning parameters used in the RWM or MALA proposal mechanism. In all cases, we take the usual choice of $\Sigma_T = \widehat{\text{Var}}(c|\mathcal{D})$ to be estimated from a short pilot run, and find the largest permissible ρ that gives an effective sample size (ESS) value for the auxiliary variable chain consistent with the minimum (over parameter chains) ESS value (mESS) obtained from the pilot run (as in Chapter 5).

Practical advice for choosing the number of particles N for PMMH can be found in Doucet *et al.* (2015) and Sherlock *et al.* (2015); see also Schmon *et al.* (2021) for parameter dimension guidelines. For CPMMH, we follow Deligiannidis *et al.* (2018) by choosing N so that the variance of the logarithm of the ratio $\hat{p}_{u^*}(\mathcal{D}|c)/\hat{p}_u(\mathcal{D}|c)$ is around 1 with c set at some central posterior value. For RWM, we use a starting point of $\lambda = 2.56^2/r$ as in Sherlock *et al.* (2015), and then follow Schmon *et al.* (2021) by fine tuning λ to give an

empirical acceptance rate of around 20%, depending on the number of parameters to be inferred. When using MALA, we apply the practical advice of Nemeth *et al.* (2016) and aim for an acceptance rate of around $40\% - 50\%$. Guidance on tuning delayed acceptance (RWM) schemes can be found in Sherlock *et al.* (2021). For DA-CPMMH schemes with either a RWM or MALA proposal, we first choose the number of particles following the procedure above, and then conditional on this choice, tune the scaling to optimise efficiency and finally, with this scaling, choose the number of particles to optimise efficiency (e.g. mESS).

6.4 Applications

To illustrate the proposed acceleration approaches, we consider three applications. In Section 6.4.1, we show the benefit of using the residual bridge constructs, even separately from other acceleration techniques, by fitting the CLE approximation to synthetic data generated from an aphid model. The underlying process in this model exhibits strong nonlinearity between observations, precluding the use of the MDB, and so we instead compare the RB constructs with a myopic but computationally inexpensive approach based on forward simulating from the underlying process. In Section 6.4.2, we fit the MJP representation of an SIR model to real data from the Eyam plagues data set. The discrete nature of the data and the inferential model requires the use of the conditioned hazard (CH) bridge construct. Finally in Section 6.4.3, we return to the well-studied Lotka-Volterra model to implement a scheme comparing several different acceleration techniques.

In what follows, all algorithms are coded in R and were run on a desktop computer with an Intel quad-core CPU. For all applications, we again compare the performance of competing algorithms using minimum (over each parameter chain) effective sample size per second (mESS/s), computed using the R coda package (Plummer *et al.*, 2006) and wall clock computing time. The latter is based on main monitoring runs of the MCMC scheme considered and we note that the CPU cost of tuning was small relative to the cost of the main run and comparable across competing schemes. When using the discretised chemical Langevin equation as the inferential model (second application), we fixed $\Delta\tau$ at 0.1, which gave a reasonable balance between accuracy and computational efficiency.

6.4.1 Aphid model

Aphids, also known as greenflies, are small, sap-sucking insects that feed on plants, often on the underside of leaves. Cotton aphids (*Aphis gossypii*) are a species of aphid that are hosted on several plants, including cotton. When aphids initially infest a plant, they tend to reproduce far faster than they die. However, as well as damaging the plant directly, they also secrete honeydew over the plant leaf, and whilst this can damage the plant further,

it also forms a cover over the leaf which prevents the aphids from moving or sucking more sap, and so causes starvation (Prajeshnu, 1998). The more aphids that have been on a leaf, the more honeydew there is and so the faster the aphids die, until the rate of death overtakes the rate of reproduction. Matis *et al.* (2006) describe a model for the population growth of aphids with two species, the current population size N_t, and the cumulative population size C_t. The reaction list is

$$\begin{aligned} \mathcal{R}_1 &: N \xrightarrow{c_1} 2N + C \\ \mathcal{R}_2 &: N + C \xrightarrow{c_2} C \end{aligned}$$

Let $X_t = (N_t, C_t)'$ denote the state of the system at time t. The stoichiometry matrix associated with the reaction system is

$$S = \begin{pmatrix} 1 & -1 \\ 1 & 0 \end{pmatrix}$$

and the associated hazard function is

$$h((N_t, C_t)', c) = (c_1 N_t, c_2 N_t C_t)'.$$

The CLE for this model, which we take to be the inferential model, is

$$d\begin{pmatrix} N_t \\ C_t \end{pmatrix} = \begin{pmatrix} c_1 N_t - c_2 N_t C_t \\ c_1 N_t \end{pmatrix} dt + \begin{pmatrix} c_1 N_t + c_2 N_t C_t & c_1 N_t \\ c_1 N_t & c_1 N_t \end{pmatrix}^{1/2} d\begin{pmatrix} W_{1,t} \\ W_{2,t} \end{pmatrix}.$$

Similarly, the LNA for this model is specified by the coupled ODE system

$$\begin{aligned} \frac{d\eta_t}{dt} &= (c_1 \eta_{N,t} - c_2 \eta_{N,t} \eta_{C,t}, c_1 \eta_{N,t})', \\ \frac{dG_t}{dt} &= \begin{pmatrix} c_1 - c_2 \eta_{C,t} & -c_2 \eta_{N,t} \\ c_1 & 0 \end{pmatrix} G_t \\ \frac{dV_t}{dt} &= V_t \begin{pmatrix} c_1 - c_2 \eta_{C,t} & c_1 \\ -c_2 \eta_{N,t} & 0 \end{pmatrix} + \begin{pmatrix} c_1 \eta_{N,t} + c_2 \eta_{N,t} \eta_{C,t} & c_1 \eta_{N,t} \\ c_1 \eta_{N,t} & c_1 \eta_{N,t} \end{pmatrix} + \begin{pmatrix} c_1 - c_2 \eta_{C,t} & -c_2 \eta_{N,t} \\ c_1 & 0 \end{pmatrix} V_t. \end{aligned}$$

Using parameter values inspired by the real data example in Whitaker *et al.* (2017*a*), synthetic data were generated at 8 integer times using Algorithm 3 with $c = (1.75, 0.001)'$ and $N_0 = C_0 = 5$. The data for $\{C_t\}$ were then discarded to obtain a challenging partial observation scenario, and the resulting data set was corrupted with Gaussian error. We followed Whitaker *et al.* (2017*a*) by taking the variance proportional to the current number of aphids in the system, which was found to give a better predictive fit in real data

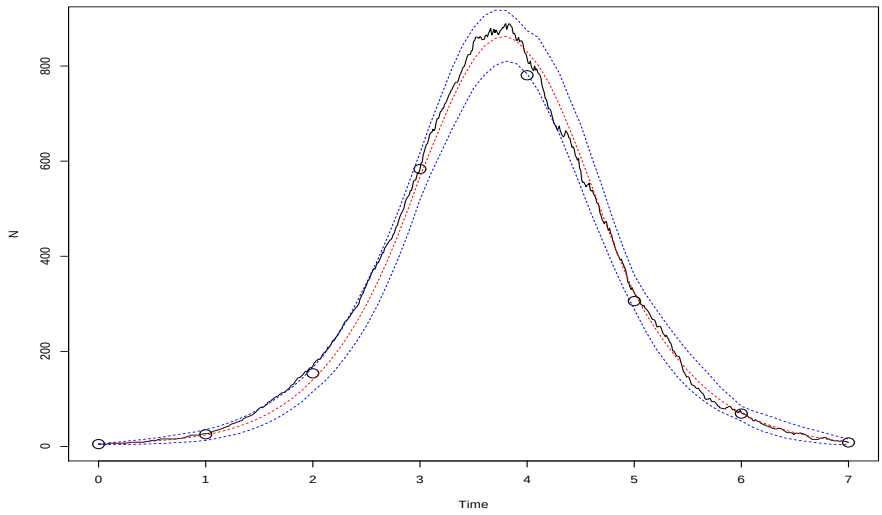

Figure 6.2: Observations from the aphid data set, with the latent process (solid line) overlaid. The dashed lines are the mean, 2.5% and 97.5% quantiles of 1000 bridges generated with the RB construct, using the ground truth for c_1 and c_2.

applications. Hence, we have that

$$Y_t = P'X_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathrm{N}\left(0, \sigma^2 P'X_t\right), \qquad t = 0, \ldots, 7 \tag{6.14}$$

where $X_t = (N_t, C_t)'$, $P' = (1, 0)$ and we chose $\sigma = 1$.

The data are shown in Figure 6.2, alongside the underlying latent N_t process that produced the data. It is clear that the behaviour of the latent process between observations is nonlinear. As discussed in Section 6.2, this precludes the use of bridge constructs such as the MDB, that push the particles towards the observations in a linear fashion. As well as using residual bridges as discussed in Section 6.2, a computationally inexpensive option is to generate particles from the model myopically from one time point to the next, without taking into consideration the observation at the end point. This will ensure that particle trajectories capture the nonlinear dynamics of the process, however, as mentioned in Section 5.1.2, this implementation leads to a highly variable estimator when the observation variance is small relative to the intrinsic stochasticity of the latent process, thus necessitating a far larger number of particles. This in turn can negate any computational benefit arising from the simplified form of the simulator and associated weight (compared to when using a bridge construct).

We therefore compared the performance of PMMH and CPMMH using either myopic simulation or the simple RB construct. We adopted an independent prior specification with $N(0, 10^2)$ distributions assigned to $\log c_1$ and $\log c_2$. We treated σ, N_0 and C_0 as fixed and known. Using $\rho \approx 1$ in this application led to long term dependence between

Scheme	N	α	CPU (s)	mESS	mESS/s	Rel.
PMMH / RWM / Myopic	100	0.10	15320	4172	0.272	1.0
CPMMH / RWM / Myopic	35	0.09	4452	2482	0.558	2.1
PMMH / RWM / RB_{part}	5	0.19	6527	4857	0.744	2.7
PMMH / RWM / RB_{iter}	5	0.17	5030	4226	0.840	3.1
CPMMH / RWM / RB_{part}	2	0.19	2593	3563	1.374	5.1
CPMMH / RWM / RB_{iter}	2	0.18	2493	3737	1.499	5.5

Table 6.2: Aphid model. Number of particles N, acceptance rate α, CPU time (in seconds), minimum ESS, minimum ESS per second, and relative (to the worst performing scheme) minimum ESS per second. All results are based on 10^5 iterations of each scheme.

	Time (months)							
	0	0.5	1	1.5	2	2.5	3	4
Susceptibles	254	235	201	153	121	110	97	83
Infectives	7	14	22	29	20	8	8	0

Table 6.3: Eyam plague data.

parameter draws, which reduced the effective sample size of the schemes. Therefore, we reduced ρ to 0.75 for this application, which we found to be optimal. To implement the residual bridge construct, we replaced $\sigma^2 P' X_t$ by $\sigma^2 P' \eta_t$ where η_t is the solution to (3.8) at observation time t but emphasise that this is necessary to obtain a tractable bridge and does not introduce any further approximation in terms of the posterior output.

Figure 6.3 and Table 6.2 summarise the output of each scheme. Table 6.2 shows that despite the computational complexity required to solve the ODEs for this construct, the resulting schemes outperform the myopic schemes in terms of overall efficiency by around a factor of 3. The behaviour of the simple residual bridge can be seen in Figure 6.2, and adequately captures the dynamics of the latent process. Indeed, we found no improvement in overall efficiency by using the residual bridge with additional subtraction (results omitted). Finally, we note a small improvement in overall efficiency by solving the ODE system used by the residual bridge, once per iteration as opposed to once per particle.

6.4.2 Epidemic model

We consider the well studied Eyam plague data set (see e.g. Raggett, 1982) consisting of 8 observations on susceptible and infective individuals during the outbreak of plague in the village of Eyam, England, taken over a four month period from June 18th 1666 and are presented here in Table 6.3.

We assume that the data can be modelled by a susceptible–Infected–Removed (SIR) compartment model which has two species (susceptibles $\mathcal{X}_1$ and infectives $\mathcal{X}_2$) and two

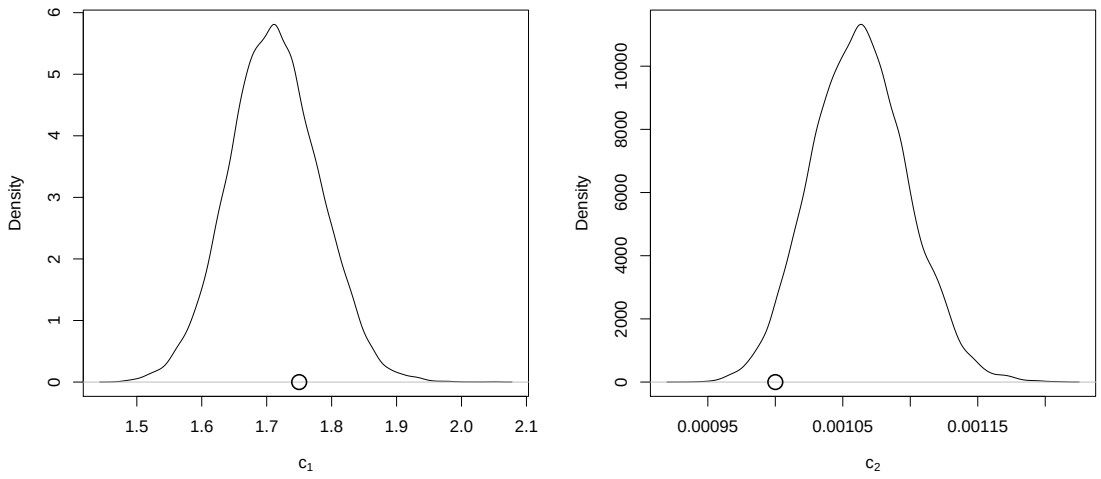

Figure 6.3: Aphid model. Marginal posterior plots for the two parameters. The ground truth is indicated on each plot.

reaction channels (infection of a susceptible and removal of an infective):

$$\begin{aligned} \mathcal{R}_1 : \quad & \mathcal{X}_1 + \mathcal{X}_2 \xrightarrow{c_1} 2\mathcal{X}_2 \\ \mathcal{R}_2 : \quad & \mathcal{X}_2 \xrightarrow{c_2} \emptyset. \end{aligned}$$

For this application, we took the Markov jump process representation of species dynamics as the inferential model. We additionally assumed the challenging scenario of exact observation of all model components (albeit at discrete times), for which the particle filter in Algorithm 7 assigns a non zero weight to the particle $x^{(k)}_{(t_i,t_{i+1}]}$ if and only if $x^{(k)}_{t_{i+1}}$ is equal to the observation $y_{t_{i+1}}$. That is, simulated trajectories must "hit" the observation or else receive zero weight. In this exact observation setting, no resampling is required and the particle filter coincides with a series of independent importance samplers (over each observation interval). Hence, the ODE solution required to implement the conditioned hazard approach of Section 5.1.1 need not be re-initialised for each particle and therefore $\mathrm{CH}_{\mathrm{iter}}$ and $\mathrm{CH}_{\mathrm{part}}$ coincide.

We followed Ho *et al.* (2018) by taking an independent prior specification with a $N(0, 100^2)$ distribution assigned to the logarithm of each rate constant. We then ran bridge-based CPMMH ($\rho = 0.99$) with and without MALA, with and without delayed acceptance. For bench-marking, we also ran standard PMMH (based on forward simulation, denoted "Myopic") and bridge-based PMMH. The main monitoring runs consisted of 10^4 iterations and this output is summarised in Table 6.4. Use of the conditioned hazard and correlating reaction times / types between successive runs of the particle filter gives

Scheme	N	α_1	$\alpha_{2\|1}$	α	CPU (s)	mESS	mESS/s	Rel.
PMMH / RWM (Myopic)	5000	–	–	0.16	68177	863	0.013	1.0
PMMH / RWM	100	–	–	0.19	25752	644	0.025	2.0
CPMMH / RWM	75	–	–	0.25	15796	609	0.039	3.0
CPMMH / MALA	75	–	–	0.41	16040	1360	0.085	6.7
CPMMH / sMALA	75	–	–	0.42	15799	891	0.056	4.4
daCPMMH / RWM	75	0.28	0.49	0.13	4746	340	0.071	5.6
daCPMMH / MALA	75	0.15	0.45	0.07	2840	386	0.136	10.7

Table 6.4: Epidemic model. Number of particles N, acceptance rates α_1, $\alpha_{2|1}$ and α, CPU time (in seconds), minimum ESS, minimum ESS per second, and relative (to the worst performing scheme) minimum ESS per second. All results are based on 10^4 iterations of each scheme.

a modest improvement in overall efficiency (by a factor of 3) compared to the most basic PMMH scheme. For this particular target posterior (see Figure 6.4) MALA is clearly more effective than RWM and is more than twice as efficient (in terms of minimum ESS per second) compared to RWM. Combining CPMMH, delayed acceptance and MALA gives the best performing scheme. The increased performance due to delayed acceptance is unsurprising, given the accuracy of the surrogate (as evidenced by the Stage-Two acceptance probability) and its computational efficiency (with the relative cost of calculating the observed data likelihood under the surrogate versus an estimate from the particle filter scaling as around 1:100).

Finally, we note from Figure 6.5 and Table 6.4 that although simplified MALA (sMALA, using equation (4.8)) gives gradients of the log posterior that are generally comparable to full MALA (using equation (4.6)), the reduction in CPU time is not sufficient to overcome the reduction in mixing efficiency. This is unsurprising given that CPU time is dominated by the particle filter, as noted above.

6.4.3 Lotka-Volterra

Recall again the Lotka-Volterra system of Section 3.4.2, and the sensitivities for this model discussed in Section 4.3.2 and derived in Appendix A.1. To implement the RB$^-$ schemes, we additionally need to augment the system of ODEs with the time derivative of the fundamental matrix, given by

$$\frac{dG_t}{dt} = \begin{pmatrix} c_1 - c_2\eta_{2,t} & -c_2\eta_{1,t} \\ c_2\eta_{2,t} & c_2\eta_{1,t} - c_3 \end{pmatrix} G_t$$

We generated a single realisation of the jump process at 51 integer times via Algorithm 3 with rate constants as in Boys *et al.* (2008), that is $c = (0.5, 0.0025, 0.3)'$ and an initial condition of $X_0 = (100, 100)'$. We then corrupted the data for both species with inde-

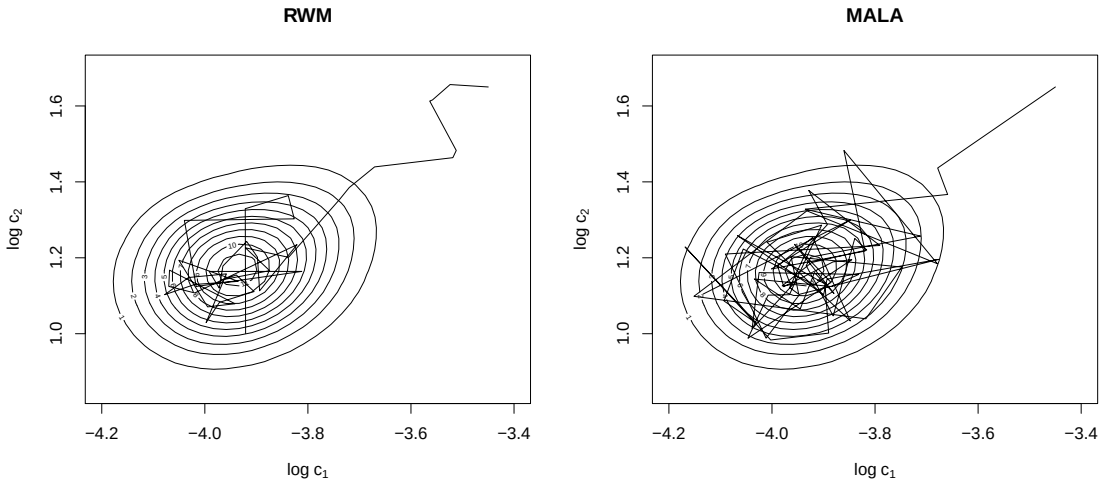

Figure 6.4: Epidemic model. Joint posterior density and the first 100 iterations of CPMMH-RWM (left) and CPMMH-MALA (right).

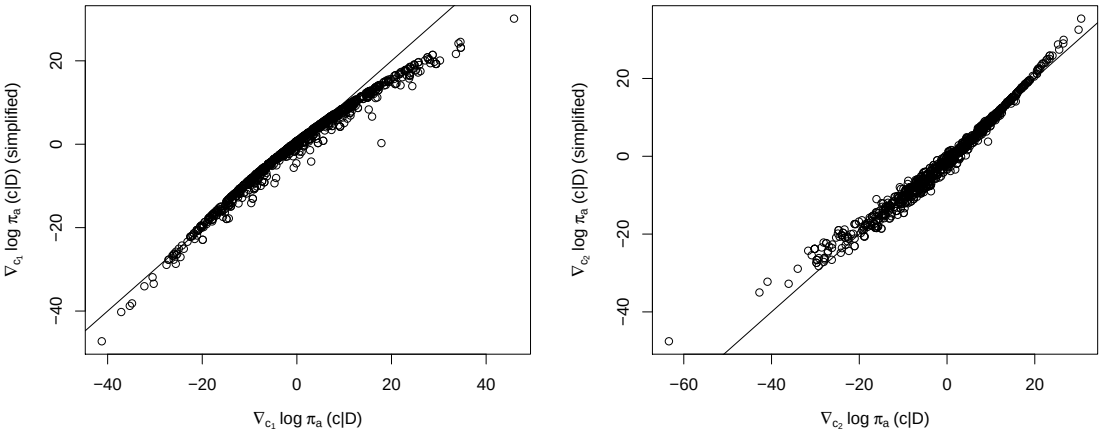

Figure 6.5: Epidemic model. Full versus simplified gradient of the log posterior density with respect to c_1 (left) and c_2 (right) computed for 1000 draws from the joint posterior over c.

pendent, additive Gaussian error and standard deviation $\sigma = 1$, so that the observation equation (4.1) becomes

$$Y_t = X_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathrm{N}\left(0, I_2\right), \qquad t = 0, \ldots, 50,$$

where I_2 is the 2×2 identity matrix. We ran CPMMH with 4 different bridge implementations: $\mathrm{RB}_{\mathrm{iter}}$, $\mathrm{RB}_{\mathrm{part}}$, $\mathrm{RB}^{-}_{\mathrm{iter}}$, and $\mathrm{RB}^{-}_{\mathrm{part}}$, along with the presence or absence of two techniques: simplified MALA and delayed acceptance. Since the residual bridge with extra

subtraction (RB^-) typically outperformed the simple residual bridge (RB) up to a factor of 2 in terms of overall efficiency (depending on the acceleration technique employed), we report results for RB^- only. Similarly, we found little difference between the gradients employed by simplified MALA versus full MALA (see Figure 6.6) and report results for the former.

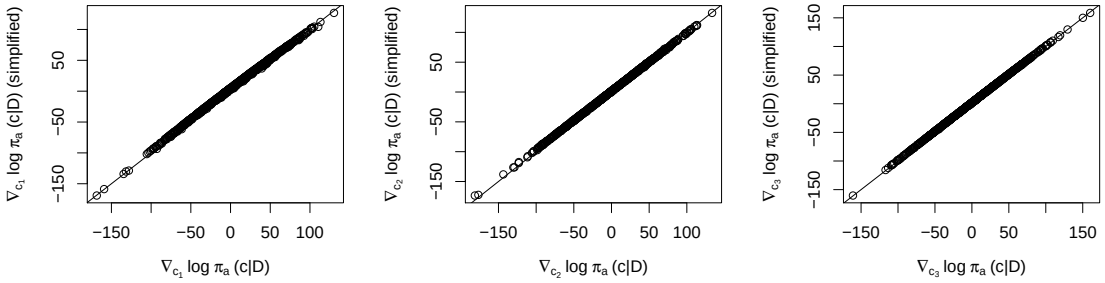

Figure 6.6: Lotka-Volterra model. Full versus simplified gradient of the log posterior density with respect to c_1 (left), c_2 (centre) and c_3 (right) computed for 1000 draws from the joint posterior over c.

We took an independent prior specification for c and assigned $N(0, 10^2)$ distributions to the logarithm of each rate constant. We then ran all schemes for 10^5 iterations, including PMMH (with $\text{RB}^-_{\text{iter}}$ which performed best of all bridge implementations) for benchmarking. Figure 6.7 and Table 6.5 summarise our findings. The former gives marginal parameter posterior densities from the output of the best performing inference scheme (with consistent results obtained from other schemes but not shown) from which we see consistency with the ground truth values. From Table 6.5, we see that the most basic CPMMH scheme (without MALA or delayed acceptance) gives an improvement in overall efficiency over PMMH of around a factor of 3. It is also clear that while the *per iteration* implementation of RB^- results in a small reduction in minimum effective sample size compared to the *per particle* implementation, the computational saving is worthwhile. Replacing the RWM parameter proposal with MALA gives a relative increase in overall efficiency by a factor of 3. It is evident that the combination of delayed acceptance and MALA gives the best performing scheme, with an order of magnitude increase in mESS/s against the benchmark.

6.4.4 Summary of Application results

Application 6.4.1 shows that in scenarios where a linear diffusion bridge such as the MDB is infeasible for inference, the residual bridge constructs outperform a computationally cheap myopic scheme of generation by forward simulation from the model. The additional

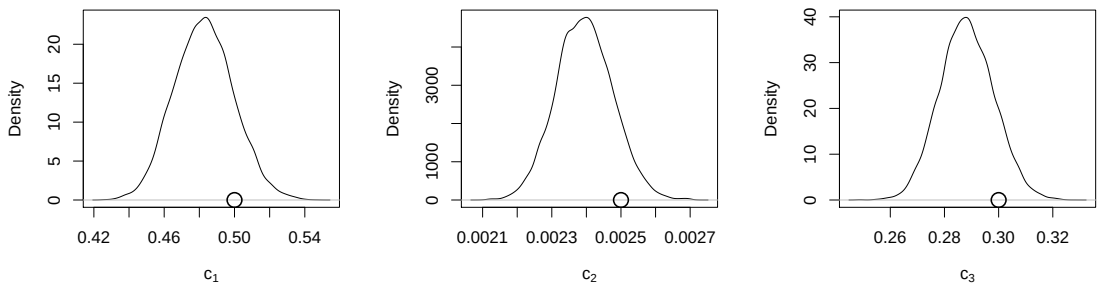

Figure 6.7: Lotka-Volterra model. Marginal posterior plots for the three parameters. The ground truth is indicated on each plot.

Scheme	N	α_1	$\alpha_{2\|1}$	α	CPU (s)	mESS	mESS/s	Rel.
PMMH / RWM / $\text{RB}^-_{\text{iter}}$	2	–	–	0.13	25735	2809	0.109	1.0
CPMMH / RWM / $\text{RB}^-_{\text{iter}}$	2	–	–	0.21	25372	7705	0.304	2.8
CPMMH / RWM / $\text{RB}^-_{\text{part}}$	2	–	–	0.25	31568	8445	0.280	2.6
CPMMH / sMALA / $\text{RB}^-_{\text{iter}}$	2	–	–	0.44	28898	24709	0.855	7.8
CPMMH / sMALA / $\text{RB}^-_{\text{part}}$	2	–	–	0.44	39643	25545	0.644	5.9
daCPMMH / RWM / $\text{RB}^-_{\text{iter}}$	2	0.22	0.85	0.19	10877	6415	0.590	5.4
daCPMMH / sMALA / $\text{RB}^-_{\text{iter}}$	2	0.46	0.84	0.39	18339	19944	1.088	10.0

Table 6.5: Lotka-Volterra model. Number of particles N, acceptance rates α_1, $\alpha_{2|1}$ and α, CPU time (in seconds), minimum ESS, minimum ESS per second, and relative (to the worst performing scheme) minimum ESS per second. All results are based on 10^5 iterations of each scheme.

computational expense of the residual bridges was offset by the smaller number of bridges required, leading to a CPMMH RB scheme that took less CPU time to generate a larger mESS than its myopic counterpart.

Applications 6.4.2 and 6.4.3 show the benefits of combining techniques such as (s)MALA and delayed acceptance with CPMMH to increase efficiency. In both applications, (s)MALA led to a greater increase in efficiency than delayed acceptance when considered separately, but the combination of both techniques led to greater improvements than when used individually. Comparing figures 6.5 and 6.6 shows that the additional assumptions made in sMALA compared to full MALA will affect the mixing efficiency more in some cases than others - this reduction in mixing efficiency may outweigh the computational savings, as in application 6.4.2. Due to the intractable form of the LNA variance sensitivity to the rate parameters, it is difficult to gain insight into when this sensitivity is small enough to be ignored without incurring a noticeable effect on the gradient estimate, as was the case in application 6.4.3.

The performance of RB against RB^- was also case dependent. In application 6.4.1,

the simple residual bridge captured the dynamics of the latent process well enough that the additional cost of the additional subtraction did not outweigh any gains in mESS. However, for application 6.4.3, the additional subtraction was worthwhile, with gains in mESS leading to a greater overall efficiency despite the additional cost. In both applications, solving the ODE system once per iteration was more efficient than solving once per particle, reducing the computational complexity whilst sacrificing very little in terms of mESS.

Chapter 7

Conclusions

This thesis considers the problem of performing Bayesian inference, given discrete, time-course data that may be subject to error, for the parameters governing a stochastic kinetic model (SKM), which is most naturally represented by a Markov Jump Process (MJP). Exact (simulation-based) inference for MJPs is complicated by the intractability of the observed data likelihood, and inference methods based on estimates of this likelihood such as pseudo-marginal Metropolis-Hastings (PMMH) are often computationally impractical due to the overhead required for integrating over every reaction time and type in the process. One possibility is to replace the MJP as the inferential model with a tractable approximation such as the linear noise approximation (LNA), which requires the solution of a system of coupled ODEs (see e.g. Komorowski *et al.*, 2009; Fearnhead *et al.*, 2014). This allows for the likelihood to be efficiently calculated via a forward filter, and so inference may proceed via traditional MCMC methods. However, there are scenarios in which the LNA is not suitable as an approximation to the MJP. In these cases, a more accurate approximation of the MJP based on time-discretisation may be used. Two such approximations are the chemical Langevin equation (CLE), and the Poisson leap method. For these models, the observed data likelihood remains intractable, necessitating the use of PMMH techniques such as particle MCMC (pMCMC), but the advantage here is that such schemes only require integrating over a user-specified number of intermediate states between each observation. Although less expensive, inference can still be very computationally intensive, and so various techniques may be required to accelerate inference using these schemes.

This thesis contributes a unified framework for performing inference for the rate constants governing an array of different classes of SKM, potentially using multiple different acceleration techniques at once. These models include an LNA approximation, the chemical Langevin equation, the Poisson leap method and the full Markov jump process, of which inference for the latter three classes of SKM utilises the pMCMC algorithm. Ac-

celeration techniques considered in this thesis cover the parameter proposal mechanism of the scheme, as well as methods of reducing both the computational cost of the particle filter and the number of times that it needs to be run.

For parameter proposal, the standard random walk Metropolis (RWM) proposal mechanism can often be improved upon by utilising the Metropolis-adjusted Langevin algorithm (MALA), which uses the gradient of the posterior density of the parameters to propose values that are more likely to be in regions of higher posterior density, thus increasing the acceptance rate and effective sample size of the scheme. Whilst it is possible to estimate the gradient directly from the inferential model when using pMCMC (Poyiadjis *et al.*, 2011; Nemeth *et al.*, 2016), we eschew this approach in favour of an approximate gradient based on the LNA (Stathopoulos & Girolami, 2013), which can clearly also be utilised when the LNA is used as the inferential model. This approximation requires the solution of a system of coupled ODEs, which scales at rate rs^2 with the number of rate constants r and species s. As one of the assumptions of the LNA is that the deterministic movement of the process dominates its intrinsic stochasticity, a further acceleration technique can be made to reduce the dimension of this ODE system by ignoring the dependence of the variance of the LNA on the parameters, instead estimating the gradient solely from the deterministic term. The success of this approach, denoted sMALA, naturally depends on the extent to which the process is dominated by its deterministic movement. In Section 4.3.2, we observed an increase in overall efficiency (as measured by minimum effective sample size per second, mESS/s) of a factor of 3 for sMALA over MALA when considering a Lotka-Volterra system with the LNA approximation as the inferential model.

To reduce the number of times the expensive particle filter must be run in pMCMC schemes, a delayed acceptance step is implemented (Golightly *et al.*, 2015), using the LNA as an inexpensive surrogate likelihood in an initial Metropolis-Hastings acceptance step. The rationale behind this step is to prune out parameter proposals that are clearly unsuitable and therefore likely to be rejected in the final Metropolis-Hastings step, therefore reserving evaluation of the particle filter for parameter proposals that are more likely to be accepted. In Section 6.4.2, we found in a real data SIR model that the use of a delayed acceptance step in schemes that employed other acceleration techniques improved mESS/s by a factor of 2.

To reduce the computational cost of the particle filter, two acceleration techniques may be used. The first of these is to induce strong, positive correlation between successive observed data likelihood estimates by correlating the innovations that drive the proposal mechanism in the particle filter (see e.g. Dahlin *et al.*, 2015; Deligiannidis *et al.*, 2018; Golightly *et al.*, 2019). To do this, innovations are drawn from a Crank-Nicolson proposal with a user-defined correlation parameter ρ, and then transformed if necessary so that the resulting kernel satisfies detailed balance with respect to the innovation density (such

as transforming the Gaussian innovations from the Crank-Nicolson proposal to Poisson variates in the case of the Poisson leap method). Typically ρ is chosen to be close to 1, with the aim of inducing strong correlation between successive estimates of the observed data likelihood. This then leads to a reduced variance in their ratio which necessitates fewer particles to be used in the particle filter to keep the variance of the acceptance probability low. However, in practice, strong correlation between the auxiliary variables does not necessarily translate to strong correlation in the observed data likelihood estimates, as a high degree of measurement error in the system can lead to a breakdown of correlation between estimates. This breakdown can impede the performance of CPMMH as the measurement error increases, as the performance appears to degrade faster than that of standard PMMH. However, when comparing CPMMH to PMMH for synthetic data from a Lotka-Volterra system corrupted with increasingly variable additive Gaussian noise in Section 5.2.2, we found that even in the most extreme scenario where the measurement error variance and average species sizes are of a similar order of magnitude, CPMMH still outperforms PMMH by a factor of 2. In other applications, CPMMH outperformed PMMH by a factor of 3 when applied to a Poisson Leap approximation of an autoregulatory network (Section 5.2.3), and by a factor of 7 when applied to an SIR model of a real-world influenza outbreak (Section 5.2.4). Unsurprisingly, the biggest gains in efficiency were achieved for a fully observed, error-free immigration-death model, using the CLE approximation as the inferential model (Section 5.2.1). With no measurement error to break the correlation between successive estimates, CPMMH here outperformed PMMH by around two orders of magnitude.

The other acceleration technique considered here is the use of three residual bridge constructs, denoted RB, RB$^-$ and CH (Whitaker *et al.*, 2017*b*; Golightly & Sherlock, 2019). These bridge constructs use the ODEs associated with the LNA surrogate (or in the case of simple RB, a subset of them) to create a bridge that aims to capture the dynamics of the underlying process, whilst still conditioning on the endpoint observation of the bridge. We considered two implementations of these surrogate-based bridge constructs: one in which the ODE system is re-solved per particle (with initial conditions informed by the current state particle) which we denote e.g. RB_{part}, and one in which the ODE system is solved once per iteration (with initial conditions informed by the output of the forward filter), which we denote e.g. RB_{iter}. The relative efficiency of both approaches likely depends on the number of particles in the particle filter, as well as which construct is being implemented, as RB requires the solution of $O(s)$ ODEs, compared to $O(s^2)$ ODEs for RB$^-$ or CH. Regardless of implementation, use of these bridge constructs leads to significantly fewer particles being required in the particle filter compared to less sophisticated bridge implementations such as myopic forward simulation, and are crucial in capturing nonlinear dynamics that may be missed by linear bridge constructs such as

the modified diffusion bridge (MDB). When comparing the residual bridges with myopic forward simulation for a partially observed aphid population model in Section 6.4.1, we observed that schemes using the residual bridge constructs required an order of magnitude fewer particles, which corresponded to an efficiency increase of a factor of 3 (as the residual bridges are more computationally expensive to implement than simple forward simulation bridges).

As the acceleration techniques considered here (with the exception of inducing correlation) leverage the tractability of the LNA, further computational savings can be made by implementing multiple acceleration techniques in tandem. In particular, from a single run of Algorithm 4, one can obtain an estimate of the posterior gradient to be used in the MALA proposal, an estimate of the likelihood under the LNA for use in a delayed acceptance step, and the solutions of the LNA ODEs for use in the $\mathrm{RB}_{\mathrm{iter}}$, $\mathrm{RB}^{-}_{\mathrm{iter}}$ or $\mathrm{CH}_{\mathrm{iter}}$ constructs. However, not all acceleration techniques are suitable for use in all contexts. For example, the use of delayed acceptance is likely to be most effective in scenarios where the overall computational cost is dominated by the particle filter and thus the cost of obtaining the likelihood under the LNA is relatively inexpensive (Sherlock *et al.*, 2021). In fact, in scenarios where this is not the case, using delayed acceptance may lead to a less efficient algorithm, as schemes using delayed acceptance are typically less statistically efficient (Christen & Fox, 2005), meaning that in general they will have a lower ESS for a given number of iterations than schemes that do not employ delayed acceptance. Similarly, the increase in statistical efficiency gained from using MALA in a pMCMC scheme is most likely to outweigh its additional computational cost in scenarios where this cost is negligible relative to the cost of estimating the likelihood through the particle filter. The optimal scenarios for implementing these acceleration techniques naturally run counter to the aim of the other techniques, which is to reduce the cost of running the particle filter. Nevertheless, there are applications where the use of all acceleration techniques together results in the most efficient scheme, as in Section 6.4.3.

A natural question is how well these acceleration techniques will scale as either the length of the dataset or the number of parameters to infer grows. It is believed that CPMMH scales better than standard PMMH as the length of the data set grows, leading to larger efficiency gains for CPMMH over PMMH for particularly long data sets. For a number of observations n, it may be possible for univariate models to scale the number of particles N at rate $n^{1/2}$, compared to at rate n for PMMH (Bérard *et al.*, 2014). For bivariate models, it may be possible to scale N at rate $n^{2/3}$. This suggests that the benefits of CPMMH over PMMH degrade as the dimension of the problem increases. See Deligiannidis *et al.* (2018) for further discussion on the scaling of CPMMH relative to PMMH. Roberts & Rosenthal (2001) show that the statistical efficiency of MALA scales better with parameter dimension than RWM schemes. However, as noted in Section 4.2,

the computational cost of implementing full MALA using LNA gradient estimation grows quickly with the parameter dimension, due to the increasing number of ODEs to be solved, which may nullify its increased statistical efficiency. Use of sMALA will greatly mitigate the additional computational cost in higher dimensions, and so should scale better. The scaling properties of other acceleration techniques remains an open area of research.

7.1 Future Work

There are a number of directions in which future research in this area can extend the work of this thesis. For example, Golightly *et al.* (2019) suggest that one way of preserving correlation between successive observed data likelihood estimates may be to perform the weighted resampling step less often, or even to avoid resampling altogether and replace the particle filter with an importance sampler. They note that the feasibility of this may depend on the accuracy of the bridge construct. As the paper considered only the modified diffusion bridge construct, it is possible that using the residual bridge constructs of Chapter 6 may lend itself well to this idea.

Another area that merits further research is the simplified MALA proposal mechanism of Section 4.2. We have given an informal justification for this approach and demonstrated that it can work well empirically, but a more theoretically rigorous treatment may be beneficial. Alternatively, other proposal mechanisms could be considered. One option is to use Hamiltonian Monte Carlo (HMC), of which MALA can be seen as a special case. In brief, HMC aims to estimate the Hamiltonian dynamics of the posterior density, and use these dynamics to propose a move to a point in the parameter space that is far away from the current state of the chain, but maintains a high acceptance probability. The resulting algorithm can be computationally intensive, but extremely statistically efficient, and the question of whether the statistical efficiency outweighs the computational burden when applied to the SKMs considered in this thesis is an interesting problem. Another proposal mechanism of which MALA is a special case is manifold MALA, which has been applied to the LNA by Stathopoulos & Girolami (2013).

As mentioned in Section 5.1.1, the particle filter of Golightly & Wilkinson (2011) used in this thesis can be seen as a special case of the auxiliary particle filter of Pitt *et al.* (2012), with pre-weight $g(y_{t_i}|x^j_{t_{i-1}}, c) = 1$. However, other choices of pre-weight are available that could be investigated. For example, Golightly & Wilkinson (2015) describe a method of performing an initial "look ahead" step by resampling amongst the $x^j_{t_{i-1}}$ with weights proportional to some $g(\cdot|\cdot)$. The tractability of the LNA could once again be exploited to provide this pre-weight. However, as mentioned in Golightly & Wilkinson (2015), the Gaussian approximation of the LNA may be light-tailed relative to the target, and so this look-ahead step could lead to otherwise valid trajectories being pruned out.

Finally, an acceleration technique not considered within this thesis is the use of parallelisation. Although the state-dependent nature of MCMC schemes means that different iterations of the scheme cannot be computed in parallel, there are elements within each iteration that can be computed in parallel, notably the particles of the particle filter. For instance, Choppala *et al.* (2016) implement a block pseudo-marginal method which utilises several independent particle filters computed in parallel. An approach similar to this may be able to be implemented for the particle filters used in this thesis, and the resulting schemes could be compared to the block pseudo-marginal method when both are applied to SKMs.

Appendix A

Additional model details

A.1 First order sensitivities for the Lotka-Volterra model

Recall the Lotka-Volterra system introduced in Section 3.4.2. In addition to the equations therein, we require the first order sensitivities for the model to perform MALA. These sensitivities are intractable, however, we can find their time derivatives by applying (4.7) to (3.24) and (3.25). Let $\xi = (\eta_{1,t}, \eta_{2,t}, V_{1,t}, V_{C,t}, V_{2,t})$. Note that the final term of (4.7) is the only term that explicitly depends on c_i, and so there is significant overlap of the form of the sensitivities with respect to each parameter. The remaining terms of each sensitivity time derivative are given by

$$\frac{d}{dt}S_1^i - \frac{d}{dc_i}\frac{d\xi_1}{dt} = (c_1 - c_2\eta_{2,t})\, S_1^i - c_2\eta_{1,t}\, S_2^i \tag{A.1}$$

$$\frac{d}{dt}S_2^i - \frac{d}{dc_i}\frac{d\xi_2}{dt} = c_2\eta_{2,t}\, S_1^i + (c_2\eta_{1,t} - c_3)\, S_2^i \tag{A.2}$$

$$\begin{aligned}\frac{d}{dt}S_3^i - \frac{d}{dc_i}\frac{d\xi_3}{dt} &= (c_1 + c_2(\eta_{2,t} - 2V_{C,t}))\, S_1^i + c_2(\eta_{1,t} - 2V_{1,t})\, S_2^i \\ &\quad + 2(c_1 - c_2\eta_{2,t})\, S_3^i - 2c_2\eta_{1,t}\, S_4^i\end{aligned} \tag{A.3}$$

$$\begin{aligned}\frac{d}{dt}S_4^i - \frac{d}{dc_i}\frac{d\xi_4}{dt} &= c_2(V_{C,t} - V_{2,t} - \eta_{2,t})\, S_1^i + c_2(V_{1,t} - V_{C,t} - \eta_{1,t})\, S_2^i \\ &\quad + c_2\eta_{2,t}\, S_3^i + (c_1 + c_2(\eta_{1,t} - \eta_{2,t}) - c_3)\, S_4^i - c_2\eta_{1,t}\, S_5^i\end{aligned} \tag{A.4}$$

$$\begin{aligned}\frac{d}{dt}S_5^i - \frac{d}{dc_i}\frac{d\xi_5}{dt} &= c_2(2V_{2,t} + \eta_{2,t})\, S_1^i + (c_2(2V_{C,t} + \eta_{1,t}) + c_3)\, S_2^i \\ &\quad + 2c_2\eta_{2,t}\, S_4^i + 2(c_2\eta_{1,t} - c_3)\, S_5^i,\end{aligned} \tag{A.5}$$

for $i \in \{1, 2, 3\}$. Replacing i in (A.1) to (A.5) with the appropriate superscript in the following, the full sensitivity time derivatives can be written as

$$
\begin{aligned}
\frac{d}{dt}S_1^1 &= (A.1) + \eta_{1,t} \\
\frac{d}{dt}S_1^2 &= (A.1) - \eta_{1,t}\eta_{2,t} \\
\frac{d}{dt}S_1^3 &= (A.1) \\
\frac{d}{dt}S_2^1 &= (A.2) \\
\frac{d}{dt}S_2^2 &= (A.2) + \eta_{1,t}\eta_{2,t} \\
\frac{d}{dt}S_2^3 &= (A.2) - \eta_{2,t} \\
\frac{d}{dt}S_3^1 &= (A.3) + 2V_{1,t} + \eta_{1,t} \\
\frac{d}{dt}S_3^2 &= (A.3) - 2\eta_{2,t}V_{1,t} - 2\eta 1, tV_{C,t} + \eta_{1,t}\eta_{2,t} \\
\frac{d}{dt}S_3^3 &= (A.3) \\
\frac{d}{dt}S_4^1 &= (A.4) + V_{C,t} \\
\frac{d}{dt}S_4^2 &= (A.4) + (\eta_{1,t} - \eta_{2,t})V_{C,t} + \eta_{2,t}V_{1,t} - \eta_{1,t}V_{2,t} - \eta_{1,t}\eta_{2,t} \\
\frac{d}{dt}S_4^3 &= (A.4) - V_{C,t} \\
\frac{d}{dt}S_5^1 &= (A.5) \\
\frac{d}{dt}S_5^2 &= (A.5) + 2\eta_{1,t}V_{2,t} + 2\eta_{2,t}V_{C,t} + \eta_{1,t}\eta_{2,t} \\
\frac{d}{dt}S_5^3 &= (A.5) + \eta_{2,t} - 2V_{2,t}.
\end{aligned}
$$

These derivatives are then integrated forward with the ODEs governing the LNA for this process, (3.24) and (3.25) to numerically obtain the sensitivities.

A.2 First order sensitivities for the epidemic model

Recall the SIR model considered in Section 6.4.2. To perform MALA, we require an estimate of the gradient of the log-target density, which we obtain via the LNA surrogate as discussed in Section 6.3 and Section 4.2. To do so, we approximate X_t as $X_t \sim N(\eta_t, V_t)$ as in Section 3.3.2, where

$$\eta_t = \begin{pmatrix} \eta_{1,t} \\ \eta_{2,t} \end{pmatrix}, \quad V_t = \begin{pmatrix} V_{1,t} & V_{C,t} \\ V_{C,t} & V_{2,t} \end{pmatrix},$$

and $V_{C,t}$ denotes the covariance between $X_{1,t}$ and $X_{2,t}$.The Jacobian F_t for this model is given by

$$F_t = \begin{pmatrix} -c_1\eta_{2,t} & -c_1\eta_{1,t} \\ c_1\eta_{2,t} & c_1\eta_{1,t} - c_2 \end{pmatrix},$$

and the ODE system governing the LNA is given by

$$\frac{d\eta_t}{dt} = (-c_1\eta_{1,t}\eta_{2,t}, c_1\eta_{1,t}\eta_{2,t} - c_2\eta_{2,t})', \tag{A.6}$$

$$\begin{aligned} \frac{dV_t}{dt} = V_t &\begin{pmatrix} -c_1\eta_{2,t} & c_1\eta_{2,t} \\ -c_1\eta_{1,t} & c_1\eta_{1,t} - c_2 \end{pmatrix} + \begin{pmatrix} c_1\eta_{1,t}\eta_{2,t} & -c_1\eta_{1,t}\eta_{2,t} \\ -c_1\eta_{1,t}\eta_{2,t} & c_1\eta_{1,t}\eta_{2,t} + c_2\eta_{2,t} \end{pmatrix} \\ &+ \begin{pmatrix} -c_1\eta_{2,t} & -c_1\eta_{1,t} \\ c_1\eta_{2,t} & c_1\eta_{1,t} - c_2 \end{pmatrix} V_t. \end{aligned} \tag{A.7}$$

As in Appendix A.1, let $\xi = (\eta_{1,t}, \eta_{2,t}, V_{1,t}, V_{C,t}, V_{2,t})$. We find the time derivatives of the first order sensitivities by applying (4.7) to (A.6) and (A.7), and these derivatives can then be integrated forwards with (A.6) and (A.7). Following the convention set out in Appendix A.1, the terms of each sensitivity time-derivative that are independent of c_i are

given by

$$\frac{d}{dt}S_1^i - \frac{d}{dc_i}\frac{d\xi_1}{dt} = -c_1\eta_{2,t}S_1^i - c_1\eta_{1,t}S_2^i \tag{A.8}$$

$$\frac{d}{dt}S_2^i - \frac{d}{dc_i}\frac{d\xi_2}{dt} = c_1\eta_{2,t}S_1^i + (c_1\eta_{1,t} - c_2)S_2^i \tag{A.9}$$

$$\begin{aligned}\frac{d}{dt}S_3^i - \frac{d}{dc_i}\frac{d\xi_3}{dt} &= (\eta_{2,t} - 2V_{C,t})c_1S_1^i + (\eta_{2,t} - 2V_{1,t})c_1S_2^i \\ &\quad - 2c_1\eta_{2,t}S_3^i - 2c_1\eta_{1,t}S_4^i\end{aligned} \tag{A.10}$$

$$\begin{aligned}\frac{d}{dt}S_4^i - \frac{d}{dc_i}\frac{d\xi_4}{dt} &= (V_{C,t} - V_{2,t} - \eta_{2,t})c_1S_1^i + (V_{1,t} - V_{C,t} - \eta_{1,t})c_1S_2^i \\ &\quad + c_1\eta_{2,t}S_3^i + (c_1(\eta_{1,t} - \eta_{2,t}) - c_2)\,S_4^i - c_1\eta_{1,t}S_5^i\end{aligned} \tag{A.11}$$

$$\begin{aligned}\frac{d}{dt}S_5^i - \frac{d}{dc_i}\frac{d\xi_5}{dt} &= (2V_{2,t} + \eta_{2,t})c_1S_1^i + ((2V_{C,t} + \eta_{1,t}) + c_2)\,S_2^i \\ &\quad + 2c_1\eta_{2,t}S_4^i + 2(c_1\eta_{1,t} - c_2)S_5^i,\end{aligned} \tag{A.12}$$

for $i \in \{1, 2\}$. Replacing i in (A.8) to (A.12) with the appropriate superscript in the following, the full sensitivity time derivatives can then be written as

$$\begin{aligned}
\frac{d}{dt}S_1^1 &= (A.8) - \eta_{1,t}\eta_{2,t} \\
\frac{d}{dt}S_1^2 &= (A.8) \\
\frac{d}{dt}S_2^1 &= (A.9) + \eta_{1,t}\eta_{2,t} \\
\frac{d}{dt}S_2^2 &= (A.9) - \eta_{2,t} \\
\frac{d}{dt}S_3^1 &= (A.10) + \eta_{1,t}\eta_{2,t} - 2(\eta_{2,t}V_{1,t} + \eta_{1,t}V_{C,t}) \\
\frac{d}{dt}S_3^2 &= (A.10) \\
\frac{d}{dt}S_4^1 &= (A.11) + \eta_{2,t}(V_{1,t} - V_{C,t}) + \eta_{1,t}(V_{C,t} - V_{2,t}) - \eta_{1,t}\eta_{2,t} \\
\frac{d}{dt}S_4^2 &= (A.11) - V_{C,t} \\
\frac{d}{dt}S_5^1 &= (A.12) + 2(\eta_{2,t}V_{C,t} + \eta_{1,t}V_{2,t}) + \eta_1\eta_2 \\
\frac{d}{dt}S_5^2 &= (A.12) - 2V_{2,t} + \eta_{2,t}.
\end{aligned}$$

Appendix B

Alternative algorithm details

B.1 Modified innovation scheme

We give a brief description of the modified innovation scheme (MIS) and refer the reader to Whitaker *et al.* (2017*a*) in the references therein for further details.

Consider the joint posterior of c and the latent process x under the CLE given by

$$\pi(c, x) \propto \pi_0(c)p(x|c)p(\mathcal{D}|x)$$

where $p(x|c)$ and $p(\mathcal{D}|x)$ can be found in (5.3) and (5.1). A Gibbs sampler can be used to generate draws from $\pi(c, x)$ by alternately sampling from the full conditionals

1. $p(x|c, \mathcal{D})$,
2. $p(c|x)$.

It is straightforward to sample $p(x|c, \mathcal{D})$ using Metropolis within Gibbs coupled with a suitable blocking approach. For example, the latent process can be updated over each interval $[t-1, t+1]$, $t = 1, 2, \ldots, n-1$ with the modified diffusion bridge construct in (5.11) used as the proposal mechanism. The use of overlapping blocks in this way ensures that latent process is updated at the observation times (as well as at all other intermediate times). The full conditional $p(c|x)$ can be sampled via Metropolis within Gibbs however for small values of $\Delta\tau$, dependence between the parameters and latent process can render this approach impractical. This well known problem is discussed at length in Roberts & Stramer (2001). The issue is circumvented by the MIS via a reparameterisation. The basic idea is to draw parameter values conditional on a process whose quadratic variation does not determine c. For example, for a time interval $[0, T]$, conditioning on the innovations that drive the modified diffusion bridge construct (see e.g. Section 5.1.2; Durham &

Gallant, 2002) leads to the continuous-time innovation process

$$\begin{aligned} dZ_t &= \beta(X_t, c)^{-1/2} \left(dX_t - \frac{x_T - X_t}{T - t}\, dt \right), \\ &= \beta(X_t, c)^{-1/2} \left\{ \alpha(X_t, c) - \frac{x_T - X_t}{T - t} \right\} dt + dW_t \end{aligned} \tag{B.1}$$

where $\alpha(X_t, c) = S\, h(X_t, c)$ and $\beta(X_t, c) = S\, \mathrm{diag}\{h(X_t, c)\} S'$. A justification for conditioning on realisations of this process in a Gibbs sampler can be found in Fuchs (2013). In practice, we work with a discretisation of (B.1), that is, the modified diffusion bridge construct. For the induced invertible mapping $x = f(z)$ (where we have suppressed dependence of $f(\cdot)$ on c and the values of the latent process at the observation times), the full conditional density required in step 2 is easily shown to be

$$p(c|z) \propto \pi_0(c) p(f(z)|c) J(f(z)|c) \tag{B.2}$$

where $p(f(z)|c)$ is given by (5.3) and

$$J(f(z)|c) \propto \prod_{t=1}^{n-1} \prod_{k=1}^{m-1} \left| \beta(x_{\tau_{t,k-1}}, c) \right|^{-1/2}$$

is the Jacobian determinant of f. Naturally, the full conditional in (B.2) will typically be intractable, requiring the use of Metropolis-within-Gibbs updates. We propose to update $\log(c)$ using random walk Metropolis with Gaussian innovations.

Bibliography

Andersson, H. k. & Britton, T. 2000 *Stochastic epidemic models and their statistical analysis*, *Lecture Notes in Statistics*, vol. 151. Springer-Verlag, New York.

Andrieu, C., Doucet, A. & Holenstein, R. 2010 Particle Markov chain Monte Carlo methods (with discussion). *J. R. Statist. Soc. B* **72** (3), 1–269.

Andrieu, C. & Roberts, G. O. 2009 The pseudo-marginal approach for efficient computation. *Annals of Statistics* **37**, 697–725.

Barker, A. L., Brown, D. E. & Martin, W. N. 1995 Bayesian estimation and the kalman filter. *Computers & Mathematics with Applications* **30** (10), 55–77.

Beaumont, M. A. 2003 Estimation of population growth or decline in genetically monitored populations. *Genetics* **164**, 1139–1160.

Bérard, J., Del Moral, P. & Doucet, A. 2014 A lognormal central limit theorem for particle approximations of normalizing constants. *Electronic Journal of Probability* **19**, 1–28.

BMJ News and Notes 1978 Influenza in a boarding school. *British Medical Journal* p. 587.

Botha, I., Kohn, R. & Drovandi, C. 2021 Particle Methods for Stochastic Differential Equation Mixed Effects Models. *Bayesian Analysis* **16** (2), 575 – 609.

Boys, R. J. & Giles, P. R. 2007 Bayesian inference for stochastic epidemic models with time-inhomogeneous removal rates. *J. Math. Biol.* **55**, 223–247.

Boys, R. J., Wilkinson, D. J. & Kirkwood, T. B. L. 2008 Bayesian inference for a discretely observed stochastic kinetic model. *Statistics and Computing* **18**, 125–135.

Breiman, L. 1968 *Probability*. Addison-Wesley Publishing Company.

Brown, R. 1828 A brief account of microscopical observations made in the months of june, july and august, 1827, on the particles contained in the pollen of plants; and on

the general existence of active molecules in organic and inorganic bodies. *Philosophical Magazine Series 2* **4**, 161–173.

Calderhead, B. & Girolami, M. 2011 Statistical analysis of nonlinear dynamical systems using differential geometric sampling methods. *Interface Focus* **1**, 821–835.

Choppala, P., Gunawan, D., Chen, J., Tran, M.-N. & Kohn, R. 2016 Bayesian inference for state space models using block and correlated pseudo marginal methods. Available from `http://arxiv.org/abs/1311.3606`.

Christen, J. A. & Fox, C. 2005 Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics* **14** (4), 795–810.

Cotter, S. L., Roberts, G. O., Stuart, A. M. & White, D. 2013 MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science* **28** (3), 424–446.

Dahlin, J., Lindsten, F., Kronander, J. & Schon, T. B. 2015 Accelerating pseudo-marginal Metropolis-Hastings by correlating auxiliary variables. Available from `https://arxiv.1511.05483v1`.

Del Moral, P. 2004 *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. New York: Springer.

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

Deligiannidis, G., Doucet, A. & Pitt, M. K. 2018 The correlated pseudo-marginal method. *Journal of the Royal Statistical Society Series B* **80**, 839–870.

Doucet, A., Pitt, M. K., Deligiannidis, G. & Kohn, R. 2015 Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika* **102** (2), 295–313.

Durham, G. B. & Gallant, R. A. 2002 Numerical techniques for maximum likelihood estimation of continuous time diffusion processes. *Journal of Business and Economic Statistics* **20**, 279–316.

Elf, J. & Ehrenberg, M. 2003 Fast evolution of fluctuations in biochemical networks with the linear noise approximation. *Genome Res.* **13** (11), 2475–2484.

Fearnhead, P., Giagos, V. & Sherlock, C. 2014 Inference for reaction networks using the Linear Noise Approximation. *Biometrics* **70**, 457–466.

Feller, W. 1949 On the theory of stochastic processes, with particular reference to applications. *Berkeley Symposium on Mathematical Statistics and Probability* **1**, 403–432.

Ferm, L., Lötstedt, P. & Hellander, A. 2008 A hierarchy of approximations of the master equation scaled by a size parameter. *J. Sci. Comput.* **34** (2), 127–151.

Fintzi, J., Wakefield, J. & Minin, V. 2021 A linear noise approximation for stochastic epidemic models fit to partially observed incidence counts. *Biometrics* pp. 1–12.

Fokker, A. D. 1914 Die mittlere energie rotierender elektrischer dipole im strahlungsfeld. *Ann. Physik* pp. 810–820.

Fuchs, C. 2013 *Inference for diffusion processes with applications in Life Sciences*. Heidelberg: Springer.

Gamerman, D. & Lopes, H. F. 2006 *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, 2nd edn. Chapman & Hall.

Georgoulas, A., Hillston, J. & Sanguinetti, G. 2017 Unbiased Bayesian inference for population Markov jump processes via random truncations. *Statistics and Computing* **27**, 991–1002.

Gerber, M. & Chopin, N. 2015 Sequential quasi Monte Carlo (with discussion). *Journal of the Royal Statistical Society, Series B* **77**, 509–579.

Gillespie, C. S. & Golightly, A. 2010 Bayesian inference for generalized stochastic population growth models with application to aphids. *Journal of the Royal Statistical Society, Series C* **52**, 341–357.

Gillespie, D. T. 1976 A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics* **22** (4), 403–434.

Gillespie, D. T. 1977 Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry* **81**, 2340–2361.

Gillespie, D. T. 2000 The chemical Langevin equation. *Journal of Chemical Physics* **113** (1), 297–306.

Gillespie, D. T. 2001 Approximate accelerated stochastic simulation of chemically reacting systems. *Journal of Chemical Physics* **115** (4), 1716–1732.

Golightly, A., Bradley, E., Lowe, T. & Gillespie, C. S. 2019 Correlated pseudo-marginal schemes for time-discretised stochastic kinetic models. *CSDA* **136**, 92–107.

Golightly, A., Henderson, D. A. & Sherlock, C. 2015 Delayed acceptance particle MCMC for exact inference in stochastic kinetic models. *Statistics and Computing* **25** (5), 1039–1055.

Golightly, A. & Sherlock, C. 2019 Efficient sampling of conditioned markov jump processes. *Statistics and Computing* pp. 1–15.

Golightly, A. & Wilkinson, D. J. 2008 Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics and Data Analysis* **52** (3), 1674–1693.

Golightly, A. & Wilkinson, D. J. 2011 Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface Focus* **1** (6), 807–820.

Golightly, A. & Wilkinson, D. J. 2015 Bayesian inference for Markov jump processes with informative observations. *SAGMB* **14** (2), 169–188.

Gordon, N. J., Salmond, D. J. & Smith, A. F. M. 1993 Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F* **140**, 107–113.

Graham, M. M. & Storkey, A. J. 2017 Asymptotically exact inference in differentiable generative models. *Electronic Journal of Statistics* **11** (2), 5105 – 5164.

Grima, R., Thomas, P. & Straube, A. V. 2011 How accurate are the nonlinear chemical fokker-planck and chemical langevin equations? *The Journal of Chemical Physics* **135** (8), 084–103.

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

Hey, K. L., Momiji, H., Featherstone, K., Davis, J. R. E., White, M. R. H., Rand, D. A. & Finkenstädt, B. 2015 A stochastic transcriptional switch model for single cell imaging data. *Biostatistics* **16** (4), 655–669.

Ho, L., Xu, J., Crawford, F., Minin, V. & Suchard, M. 2018 Birth / birth-death processes and thier computable transition probabilities with biological applications. *Journal of Mathematical Biology* **76** (4), 911–944.

Kalman, R. E. 1960 A new approach to linear filtering and prediction problems. *Trans. ASME, D* **82**, 35–44.

van Kampen, N. G. 2001 *Stochastic Processes in Physics and Chemistry*. North-Holland.

Koblents, E. & Miguez, J. 2015 A population Monte Carlo scheme with transformed weights and its application to stochastic kinetic models. *Statistics and Computing* **25**, 407–425.

Kolmogorov, A. 1931 Über die analytischen methoden in der wahrscheinlichkeitsrechnung,. *Math. Annalen* **104**, 415–458.

Komorowski, M., Finkenstadt, B., Harper, C. & Rand, D. 2009 Bayesian inference of biochemical kinetic parameters using the linear noise approximation. *BMC Bioinformatics* **10** (1), 343.

Kurtz, T. G. 1970 Solutions of ordinary differential equations as limits of pure jump Markov processes. *J. Appl. Probab.*. **7**, 49–58.

Kurtz, T. G. 1971 Limit theorems for sequences of jump Markov processes approximating ordinary differential processes. *J. Appl. Probab.*. **8**, 344–356.

Kurtz, T. G. 1972 The relationship between stochastic and deterministic models for chemical reactions. *The Journal of Chemical Physics* **57**, 2976–2978.

Lin, J. & Ludkovski, M. 2013 Sequential Bayesian inference in hidden Markov stochastic kinetic models with application to detection and response to seasonal epidemics. *Statistics and Computing* **24**, 1047–1062.

Matis, J. H., Kiffe, T. R., Matis, T. I. & Stevenson, D. E. 2006 Application of population growth models based on cumulative size to pecan aphids. *Journal of Agricultural, Biological, and Environmental Statistics* **11** (4), 425–449.

McKinley, T. J., Ross, J. V., Deardon, R. & Cook, A. R. 2014 Simulation-based Bayesian inference for epidemic models. *Computational Statistics and Data Analysis* **71**, 434–447.

McQuarrie, D. A. 1967 Stochastic approach to chemical kinetics. *Journal of Applied Probability* **4**, 413–478.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. 1953 Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* **21**, 1087–1092.

van der Meulen, F. & Schauer, M. 2017 On residual and guided proposals for diffusion bridge simulation. Available from `https://arxiv.org/pdf/1708.04870.pdf`.

Meyn, S., Tweedie, R. L. & Glynn, P. W. 2009 *Markov Chains and Stochastic Stability*, 2nd edn. Cambridge University Press.

Minas, G. & Rand, D. A. 2017 Long-time analytic approximation of large stochastic oscillators: Simulation, analysis and inference. *PLOS Computational Biology* **13** (7), 1–23.

Minter, A. & Retkute, R. 2019 Approximate Bayesian computation for infectious disease modelling. *Epidemics* **29**.

Nemeth, C., Sherlock, C. & Fearnhead, P. 2016 Particle Metropolis-adjusted Langevin algorithms. *Biometrika* **103**, 701–717.

Øksendal, B. 2003 *Stochastic differential equations: An introduction with applications*, 6th edn. Springer-Verlag, Berlin Heidelberg New York.

O'Neill, P. D. & Roberts, G. O. 1999 Bayesian inference for partially observed stochastic epidemics. *J. R. Statist. Soc. A.* **162**, 121–129.

Owen, J., Wilkinson, D. J. & Gillespie, C. S. 2015 Likelihood free inference for Markov processes: a comparison. *Statistical Applications in Genetics and Molecular Biology* **14** (2), 189–209.

Pitt, M. K., dos Santos Silva, R., Giordani, P. & Kohn, R. 2012 On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *J. Econometrics* **171** (2), 134–151.

Pitt, M. K. & Shephard, N. 1999 Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association* **446**, 590–599.

Planck, M. 1917 Über einen satz der statistischen dynamik und seine erweiterung in der quantentheorie. *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin* .

Plummer, M., Best, N., Cowles, K. & Vines, K. 2006 CODA: convergence diagnosis and output analysis for MCMC. *R News* **6** (1), 7–11.

Poyiadjis, G., Doucet, A. & Singh, S. S. 2011 Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika* **98**, 65–80.

Prajeshnu 1998 A nonlinear statistical model for aphid population growth. *Journal of the Indian Society of Agricultural Statistics* **51** (1), 73–80.

Raggett, G. 1982 A stochastic model of the Eyam plague. *Journal of Applied Statistics* **9**, 212–225.

Roberts, G. O. & Rosenthal, J. S. 1998 Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society Series B* **60** (1), 255–268.

Roberts, G. O. & Rosenthal, J. S. 2001 Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science* **16** (4), 351–367.

Roberts, G. O. & Rosenthal, J. S. 2004 General state space markov chains and mcmc algorithms. *Probability Surveys* **1**, 20–71.

Roberts, G. O. & Stramer, O. 2001 On inference for non-linear diffusion models using Metropolis-Hastings algorithms. *Biometrika* **88** (3), 603–621.

Roberts, G. O. & Stramer, O. 2002 Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and Computing in Applied Probability* **4**, 337–357.

Roberts, G. O. & Tweedie, R. L. 1996 Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* **2** (4), 341–363.

Ryder, T., Prangle, D., Golightly, A. & Matthews, I. 2021 The neural moving average model for scalable variational inference of state space models. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, *Proceedings of Machine Learning Research*, vol. 161, pp. 12–22. PMLR.

Schmon, S. M., Deligiannidis, G., Doucet, A. & Pitt, M. K. 2021 Large sample asymptotics of the pseudo-marginal method. *Biometrika* **108** (1), 37–51.

Schmon, S. M. & Gagnon, P. 2021 Optimal scaling of random walk Metropolis algorithms using Bayesian large-sample asymptotics. *https://arxiv.org/abs/2104.06384* .

Scott, M., Ingalls, B. & Kæ rn, M. 2006 Estimations of intrinsic and extrinsic noise in models of nonlinear genetic networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **16** (2), 026107.

Sherlock, C., Golightly, A. & Gillespie, C. S. 2014 Bayesian inference for hybrid discrete-continuous systems biology models. To appear in Inverse Problems.

Sherlock, C., Thiery, A. & Golightly, A. 2021 Efficiency of delayed-acceptance random walk metropolis algorithms. *The Annals of Statistics* **49** (5), 2972–2990.

Sherlock, C., Thiery, A., Roberts, G. O. & Rosenthal, J. S. 2015 On the efficiency of pseudo-marginal random walk Metropolis algorithms. *The Annals of Statistics* **43** (1), 238–275, available from `http://arxiv.org/abs/1309.7209`.

Stathopoulos, V. & Girolami, M. 2013 Markov chain Monte Carlo inference for Markov jump processes via the linear noise approximation. *Philosophical Transactions of the Royal Society A* .

Stramer, O. & Bognar, M. 2011 Bayesian inference for irreducible diffusion processes using the pseudo-marginal approach. *Bayesian Analysis* **6**, 231–258.

Stratonovich, R. L. 1966 A new representation for stochastic integrals and equations. *SIAM Journal on Control* **4** (2), 362–371.

Sun, L., Lee, C. & Hoeting, J. A. 2015 Parameter inference and model selection in deterministic and stochastic dynamical models via approximate Bayesian computation: modeling a wildlife epidemic. *Environmetrics* **26**, 451–462.

Tierney, L. 1994 Markov chains for exploring posterior distributions. *The Annals of Statistics* **22** (4), 1701–1762.

Tran, M.-N., Kohn, R., Quiroz, M. & Villani, M. 2016 Block-wise pseudo-marginal Metropolis-Hastings. Available from `http://arxiv.org/abs/1603.02485`.

Wang, Y., Christley, S., Mjolsness, E. & Xie, X. 2010 Parameter inference for discretely observed stochastic kinetic models using stochastic gradient descent. *BMC Systems Biology* **4**, 99.

West, M. & Harrison, J. 1997 *Bayesian Forecasting and Dynamic Models*, 2nd edn. Springer-Verlag.

Whitaker, G. A. 2016 Bayesian inference for stochastic differential mixed-effects models. PhD thesis, Newcastle University, Newcastle, UK.

Whitaker, G. A., Golightly, A., Boys, R. J. & Sherlock, C. 2017*a* Bayesian inference for diffusion driven mixed-effects models. *Bayesian Analysis* **12**, 435–463.

Whitaker, G. A., Golightly, A., Boys, R. J. & Sherlock, C. 2017*b* Improved bridge constructs for stochastic differential equations. *Statistics and Computing* **27**, 885–900.

Wiener, N. 1923 Differential-space. *Journal of Mathematics and Physics* pp. 131–174.

Wilkinson, D. J. 2009 Stochastic modelling for quantitative description of heterogeneous biological systems. *Nature Reviews Genetics* **10**, 122–133.

Wilkinson, D. J. 2018 *Stochastic Modelling for Systems Biology*, 3rd edn. Boca Raton, Florida: Chapman & Hall/CRC Press.

Wu, Q., Smith-Miles, K. & Tian, T. 2014 Approximate Bayesian computation schemes for parameter inference of discrete stochastic models using simulated likelihood density. *BMC Bioinformatics* **15**.