Bayesian inference for biological time series

Richard Creswell
Hertford College
University of Oxford

A thesis submitted for the degree of Doctor of Philosophy (24 July 2023).

Supervisors:
Prof. David Gavaghan
Dr. Ben Lambert
Dr. Martin Robinson
Dr. Chon Lok Lei

Examiners:
Prof. Ruth Baker
Dr. Louise Dyson

Acknowledgements:
The author acknowledges support from the Engineering and Physical Sciences Research Council and the University of Oxford for studentship support.

Abstract

Inferring the parameters of time series models from observed data is essential across many areas of science. Bayesian statistics provides a powerful framework for this purpose, but significant challenges arise when time series models are misspecified due to complexities in the underlying process (e.g., heterogeneity in the modelled population, or when parameter values fluctuate over time), inaccurate numerical approximation of the forward model (e.g., in models involving differential equations), or the presence of non-stationary, non-independent error terms. We introduce a series of models and computational strategies for dealing with misspecification in time series inference problems, with a particular focus on time series problems arising in epidemiology and problems involving ordinary differential equations.

The models and inference strategies discussed include: 1. A generalisation of the Poisson renewal model to allow heterogeneous behaviour between local and imported cases, which we use to show that accounting for such heterogeneous behaviour is essential for accurate inference of the time-varying reproduction number (R_t); 2. A Bayesian nonparametric approach to flexibly learn time variation in R_t, which we show is capable of learning accurate and precise estimates of the parameter; 3. Estimates of the gradient and the error in the log-likelihood arising from numerical approximation of differential equations derived from *a posteriori* error analysis; and 4. A flexible noise process accommodating correlated and heteroscedastic error terms and whose form can be inferred from time series data using kernel functions. We motivate our methodological innovation by a comprehensive examination of the biases in inference that result from insufficiently accurate numerical approximation of differential equations, as well as time series inverse problems and models drawn from epidemiology, hydrology, and cardiac electrophysiology.

Contents

Chapter 1

Introduction

1.1 Motivation

A wide range of scientific phenomena involve time-varying observables. The rate at which water flows through a river, the number of people presently infected with a particular disease, or the current flowing out of a cell are a few selected examples of such time-varying outputs. These observable quantities are recorded at a discrete set of points in time, yielding a *time series*, which is often assumed to obey some parametric model derived from scientific theory. Often, the next step is to learn which values of the model parameters are compatible with the observed data. Developing computational tools to perform this task of model parameterisation more efficiently, practically, and accurately is the main goal of this thesis.

Many subfields of computational biology are replete with challenging time series inference problems. However, this thesis focuses particularly (though not exclusively) on epidemiological time series inference problems (see Chapter 1, §1.3). Epidemiological modelling of infectious disease depends heavily on the analysis of time series data. Time-varying data in infectious disease epidemiology often involves cases, deaths, or prevalence; these data can be used to inform the progression of an epidemic or the effectiveness of interventions intended to control an epidemic by fitting them to appropriate models of the transmission of an infectious disease through a population.

Throughout the thesis, we generally adopt a Bayesian approach. Bayesian statistics provides a powerful formalism for updating our beliefs about parameter values conditional on observed data, and is attractive for its principled handling of uncertainty in recovered parameter estimates (as discussed further in Chapter 2). However, performing Bayesian inference for time series models involves a number of challenges.

Many of these challenges are because scientific models inevitably fail to account for all of the observed variation in real data. In some cases, these challenges can be addressed through the development of more realistic models, which, for example, more comprehensively account for various features of the modelled system. However, more complex models are more difficult to fit to data and more likely to suffer from problems with practical identifiability, requiring care in the development and deployment of inference algorithms and, where possible, the collection of more comprehensive data. Thus, the appropriate level of model complexity to explain a particular dataset may be difficult to determine in advance, and this choice must be guided by the information in the data and the modelling or inference task at hand.

Models involving differential equations are widespread in computational biology and other fields, and because the outputs of these models must typically be approximated using numerical methods, additional difficulties may arise. Even once a differential equation system has been determined whose true solution is an appropriate model for the data, discrepancies between the true solution and its numerical approximation may interfere with Bayesian inference for the parameters of the model. However, highly accurate numerical approximation may be prohibitively computationally expensive, particularly during inference where large numbers of simulations of the same differential equations at different parameter values must be performed.

Observed time series data will almost always be affected by a variety of unmodelled influences. In these cases, even correctly specified and accurately computed deterministic models must incorporate a stochastic component to capture all the variation in observed data. Standard choices for this component such as independent and identically distributed Gaussian noise are justifiable in some cases, but they are inaccurate models of data affected by heteroscedasticity or autocorrelation, which may occur, for example, in time series where observation noise scales with the magnitude of the signal (heteroscedasticity) or when measurement are unable to capture short-term fluctuations in observables (autocorrelation).

1.2 Aims of the thesis

In this thesis, we aim to develop and apply a series of models and Bayesian inference strategies which address the challenges described in §1.1 above. Specifically, we aim to:

1. Develop more accurate models for biological phenomena;

2. Develop a flexible framework for accurately learning time variation in model

parameters;

3. Demonstrate the importance of highly accurate numerical approximation of ODEs when performing inference for their parameters, and efficiently infer the ODE parameters while controlling numerical error in the parameter posteriors;

4. Learn accurate parameter posteriors even in the presence of non-stationary heteroscedastic and autocorrelated noise.

5. Develop reliable and reusable software for performing all of these inference tasks (see Chapter 2, §2.5).

Throughout the thesis, a key focus is on applying novel methodologies to relevant biological problems. The problem which motivates the first portion of the thesis, and which provides the context in which we address points 1 and 2 above, is drawn from epidemiology, and concerns the development of effective inference algorithms for learning the time-varying reproduction number (R_t) from incidence data. We aim to develop a more accurate stochastic renewal model of infectious disease outbreaks incorporating differing transmission risk between local and imported cases, in order to increase the accuracy of R_t estimates when such heterogeneities are present in the population. Subsequently, we aim to develop a flexible framework for efficiently learning patterns of time variation in R_t; however, our work here is generally applicable to other problems in epidemiology and biology where choosing the appropriate model complexity is challenging.

The remainder of the thesis chiefly focuses on inference for differential equation models. We first aim to motivate and introduce our investigations of differential equations via a study of compartmental differential equation models as used in epidemiology, where we will show that the outputs of these models can be subject to significant uncertainty arising from parameter uncertainty or inaccuracy in numerical solvers. Subsequently, we aim to study inference for differential equations more broadly, addressing point 3 above, and we will demonstrate the importance of controlling the error on the likelihood to ensure accurate inference and develop methods for doing this efficiently. In the final chapter of the thesis, we will address point 4 above by developing a flexible noise process which can more accurately capture heteroscedasticity and time-varying autocorrelation in the error terms than the simpler, standard assumptions typically made when fitting time series data to differential equation models.

1.3 Context of the thesis

The doctoral studies which led to this thesis commenced in October of 2019. My original goal was to develop Bayesian inference algorithms for misspecified time series models, drawing on several motivating examples from cardiac electrophysiology and electrochemistry which are known to suffer from misspecification. The work on general noise processes, which appears as Chapter 8 in this thesis, was my first work in this direction.

However, before working further on the problem of misspecification in general, or on electrophysiological applications, the COVID-19 pandemic struck England about halfway through the first year of my doctoral studies. Realizing that many of the techniques for learning parameters from time-varying data that I was studying were applicable to COVID-19 time series data, I selected epidemiology as a central area of application for my computational work. I involved myself in several collaborative projects developing and applying computational modelling to relevant questions in understanding and controlling the spread of COVID-19. The research underlying Chapters 3 and 5 in this thesis arose from these projects. Epidemiology, including for diseases other than COVID-19, remains an important area for further methodological innovation and application of inference methods, as discussed further in Chapters 4 and 9.

Chapter 2

Background

Overview

This chapter provides a review of the background material which underlies the research presented later in the thesis. First, we provide an overview of the principles of Bayesian inference. Next, we discuss some of the methods which are used to perform Bayesian inference. We also provide some background information on differential equations and the algorithms which are used to numerically approximate their solutions. Additionally, we define appropriate likelihood functions which can be used to perform inference for the parameters of models involving ordinary differential equations.

Finally, we discuss epidemiological time series data and the variety of models which have been developed of the spread of an infectious disease.

Publications

Although this chapter primarily concerns general background information, Figure 2.1 in this chapter was taken from the following preprint on which I was co-author:

- K. Gallagher,[†] I. Bouros,[†] N. Fan,[†] E. Hayman,[†] L. Heirene,[†] P. Lamirande,[†] A. Lemenuel-Diot, B. Lambert, D. J. Gavaghan, **and R. Creswell:** "Epidemiological Agent-Based Modelling Software (Epiabm)," arXiv:2212.04937 (2022). [Gallagher et al., 2022]

  ([†]= joint first authorship.)

  Contributions to [Gallagher et al., 2022]: I contributed to the supervision of the project, made suggestions on the writing and revision of the manuscript, led the

students who were working on the software implementation, and designed the majority of the figure which appears as Figure 2.1 in this thesis (with the exception of the portion of the figure indicating places with variable members).

Additionally, some material in this chapter (portions of §2.3) is taken from [Creswell et al., 2023c].

2.1 Bayesian Inference

2.1.1 Data and Models

Our data consist of observations $y = (y_1, y_2, \ldots, y_N)$; each $y_i \in \mathcal{Y}_i \subseteq \mathbb{R}^n$. For the data, we propose a STATISTICAL MODEL $(\mathcal{S}, \mathcal{P})$, where $\mathcal{S}$ is the SAMPLE SPACE (all possible realisations of y) and $\mathcal{P}$ is a finite or infinite set of probability distributions on $\mathcal{S}$. We label the elements of $\mathcal{P}$ by unique values of the PARAMETERS θ, i.e., $\mathcal{P} = \{p(y|\theta) : \theta \in \Theta\}$; Θ is the PARAMETER SPACE.

Example 1 *The observed data are* $y = (1.2, 3.43, -0.325)$. *We model the data according to* $y_i \overset{\text{IID}}{\sim} \mathcal{N}(\mu, \sigma)$, *for* $\sigma > 0$, $\mu \in \mathbb{R}$; *i.e., with a statistical model* $(\mathcal{S} = \mathbb{R}^3, \mathcal{P} = \{\mathcal{N}(y|(\mu, \mu, \mu), \sigma I) : \sigma > 0, \mu \in \mathbb{R}\})$ *where* I *is the* 3×3 *identity matrix. The parameter space is two-dimensional:* $\Theta = \{(\sigma, \mu) : \sigma > 0, \mu \in \mathbb{R}\}$.

Sometimes, statistical models are constructed in order to characterize relationships or trends that may exist in the observations without containing any direct description of the physical mechanisms which gave rise to the data (this is common, for example, in regression modelling). We use the term *mechanistic* models to refer to those models which are parameterized by quantities with a direct biological or physical interpretation, and aim to summarize in mathematical equations the actual physical processes which generated the data (see, e.g., [Hilborn and Mangel, 2013, Baker et al., 2018]). Mechanistic models may be deterministic or stochastic. Deterministic mechanistic models are often used in concert with some assumed form of stochasticity in the observed data: see §2.3.3.

2.1.2 Likelihood functions

The LIKELIHOOD takes the same form as the joint probability density of the data, but treated as a function of θ, with the data treated as fixed. Thus, the likelihood is not

a probability distribution of θ. Higher values of the likelihood indicate a value of θ which gives a better fit to the data. The form of the likelihood follows from the statistical model adopted for the data. The likelihood encompasses both assumptions about the mechanistic process underlying the system or experiment as well as the noise or error properties of the observed data. To avoid numerical underflow error, computations involving the likelihood are typically done on the log scale and involve the LOG-LIKELIHOOD: $\log(p(y|\theta))$. Throughout this thesis, we reserve the calligraphic $\mathcal{L}(\theta|y)$ notation to refer to the log-likelihood, while we use L to refer to the likelihood function itself.

Example 2 *The data $y = (y_1, y_2, \ldots, y_N)$ are modelled according to $y_i \sim \mathcal{N}(\mu_i, \sigma)$, i.e., as independent (but not identically distributed) Gaussian. The log-likelihood for the parameters $\mu = (\mu_1, \mu_2, \ldots, \mu_N)$ and σ is:*

$$\mathcal{L}(\mu, \sigma|y) = \log(p(y|\mu, \sigma)) = -\frac{N}{2}\log(2\pi) - N\log(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \mu_i)^2. \tag{2.1}$$

2.1.3 Inference

In many fields of scientific inquiry, statistical models are proposed as explanations of observable phenomena. Once experimental data are collected, a ubiquitous research task is to identify which values of the parameters θ are compatible with the data. This task, known in various settings as INFERENCE or the INVERSE PROBLEM, is central to all of the research presented in this thesis.

Some approaches to inference yield a single, best fit estimate of θ for a particular dataset y. One such approach is the method of MAXIMUM LIKELIHOOD, which selects the value of θ which maximizes the likelihood function:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \log(p(y|\theta)).$$

In general, likelihood functions are not convex, and the maximum likelihood estimation relies on non-convex optimisation algorithms in order to find $\hat{\theta}$ (e.g., CMA-ES [Hansen et al., 2003]).

However, methods such as maximum likelihood, which yield only a best fit estimate of θ, are of limited usefulness in many scientific applications; rather, information about the uncertainty in θ implied by the data is needed. In such applications, the *probability distribution of θ conditional on y*, which provides not only a point estimate of θ (i.e.,

the mean or median of the distribution) but also information about its uncertainty implied by the data is more useful. Information about the uncertainty in θ is essential to satisfactorily answer many scientific questions.

Example 3 *In epidemiology, a common inference task is to learn the* TIME DEPENDENT REPRODUCTION NUMBER, R_t *(the expected number of secondary cases caused by each primary infection)—values of* $R_t > 1$ *indicate that an infectious disease will continue to spread, while values* $R_t < 1$ *indicate that an outbreak will die out. However, even if the best fit* R_t *to a particular incidence time series falls below 1, it would be imprudent to conclude on this basis that the disease is under control if, say, a* 75^{th} *percentile estimate of* R_t *were still greater than 1. Only when the* probability that $R_t > 1$ *is at a suitably small value should the disease be treated as under control. For this reason, inference for* R_t *has often adopted a Bayesian approach, e.g., [Thompson et al., 2019, Creswell et al., 2022].*

2.1.4 Bayes' Rule

Bayes' Rule [Bayes and Price, 1763] states:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}. \tag{2.2}$$

That is, the POSTERIOR distribution of θ (i.e., $p(\theta|y)$) is the product of the likelihood, $p(y|\theta)$ and the PRIOR, $p(\theta)$ over the MARGINAL LIKELIHOOD, $p(y)$.

The prior expresses our beliefs about the values of θ before observing any data. The prior is a central aspect of Bayesian inference. The choice of prior can be guided by many considerations. In problems where the parameters must be constrained to fall in certain intervals, or knowledge about their possible values is available from previous experiments, the prior distribution is a natural way to incorporate this knowledge into inference. In other problems, it is desirable for the prior to have as little influence on the inference results as possible, letting the shape of the posterior be dominated by the data (i.e., the likelihood function). Careful selection of the prior is an essential step when performing Bayesian inference.

Priors are often selected from a parametric family of distributions (e.g., a Gaussian distribution); the parameters of the prior distribution are termed HYPERPARAMETERS. (Similarly, when the posterior distribution is expressed as a member of a parametric family of distributions, we may refer to its parameters as hyperparameters.) When performing inference, the hyperparameters of the prior may be set to fixed values representing the assumption of a particular fixed prior distribution; or the prior hyper-

parameters may themselves be treated as unknown and assumed to obey their own prior distribution, termed a HYPERPRIOR, and inferred along with the other parameters of the model.

The marginal likelihood:

$$p(y) = \int_{\Theta} p(y|\theta)p(\theta)d\theta, \tag{2.3}$$

is challenging to calculate in practical problems where no analytical result exists and Θ often contains at least several dimensions. However, the marginal likelihood's lack of dependence on θ means that it can be viewed as a mere normalisation term for the posterior, and characterization of the location and shape of the UNNORMALISED POSTERIOR, $p(y|\theta)p(\theta)$, is sufficient to learn θ.

2.2 Bayesian inference in practice

2.2.1 Conjugate priors and motivation for sampling methods

In certain problems, the posterior eq. (2.2) is expressible as some standard probability distribution whose hyperparameters are closed-form functions of the data. This convenient situation often depends upon the choice of a CONJUGATE PRIOR—a distributional assumption for the prior such that the posterior is a distribution from the same parametric family, with updated hyperparameters.

Example 4 *The data $y = (y_1, y_2, \ldots, y_N)$, where each y_i is a non-negative integer, are modelled according to $y_i \overset{\text{IID}}{\sim}$ Poisson(λ). We choose a gamma prior[1] for the unknown parameter λ; i.e., $p(\lambda) =$ gamma(α, β). The posterior distribution of λ is given by:*

$$p(\lambda|y) = \textit{gamma}\left(\alpha + \sum_i y_i, \beta + N\right). \tag{2.4}$$

Because the gamma distribution is a conjugate prior *for the Poisson likelihood, the posterior can also be expressed as another gamma distribution whose hyperparameters are closed-form expressions of the data. Once the conjugate posterior has been derived, it is fast to compute.*

Unfortunately, for many likelihoods arising in scientific applications, a conjugate prior does not exist.

[1]Throughout this thesis, we parametrise the gamma distribution with a SHAPE α and RATE β such that its density function is: $f(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$.

Assuming that we can evaluate both the likelihood and the prior, eq. (2.2) allows us to calculate the unnormalised posterior at any particular value of θ, but pointwise evaluations of the unnormalised posterior are not on their own useful. What we want the posterior to tell us are what values of θ are compatible with the data, and how much uncertainty there is in these values—i.e., we want the *moments* and *percentiles* of the posterior. However (as we mentioned above in relation to the marginal likelihood), because $p(y|\theta)p(\theta)$ is not expected to be analytically integrable the moments and percentiles are in general computationally expensive to obtain.

In easy problems (say, dim $\Theta = 1$) where evaluations of the likelihood are fast, integrals of $p(y|\theta)p(\theta)$ could be evaluated using quadrature or similar numerical methods. Somewhat equivalently, the unnormalised posterior could be computed on a dense grid of θ values, and the shape and location of the posterior could easily be obtained from the graph of the unnormalised posterior on these values. However, in most problems of scientific interest, Θ is potentially high-dimensional and evaluations of the likelihood are slow, and thus any naïve grid-based calculation is prohibitively slow. For these reasons, in most applied Bayesian inference the unnormalised posterior must be interrogated using specialized algorithms. Specifically, we next describe two broad classes of algorithms which are used for this purpose: variational inference and sampling.

2.2.2 Variational Inference

In VARIATIONAL INFERENCE, the posterior is assumed to be well-approximated by some member of a set of tractable distributions $\mathcal{Q} = \{q_\phi(\theta) : \phi \in \Phi\}$ [Blei et al., 2017]. The appropriate hyperparameters ϕ of the approximate posterior are selected by minimization of the KL-divergence between $q(\theta)$ and the posterior $p(\theta|y)$. Variational inference is advantageous for its speed and scalability. However, challenges of the method include that for poor choices of $\mathcal{Q}$ the approximate posterior may differ significantly from the actual posterior, calculation of the KL-divergence can be difficult, and inefficient or non-convergent minimization algorithms will cause slow performance or further inaccuracy.

2.2.3 Sampling

Sampling methods perform Bayesian inference by generating samples from the posterior distribution. Once enough samples have been collected, the relevant properties of the posterior, such as its mean and percentiles, can simply be approximated from the empirical distribution of the samples. In particular, for a function f integrable on Θ', we

have by the law of large numbers:

$$\int_{\Theta'} f(\theta)p(\theta)d\theta \approx \frac{1}{K}\sum_{k=1}^{K} f(\theta^{(k)})$$

for sufficiently large K, where p is a probability density supported on Θ', and $\theta^{(k)}$ are distributed according to p. Note that, e.g., the mean of the posterior can thus be approximated from posterior samples using $f(\theta) = \theta$; $p(\theta) = p(\theta|y)$ in this formula.

A widely used class of algorithms for generating samples from a posterior distribution are MARKOV CHAIN MONTE CARLO (MCMC) methods. These algorithms involve the construction and simulation of a Markov chain whose equilibrium distribution is the posterior. Many MCMC algorithms are variants of the METROPOLIS-HASTINGS algorithm, which is described in Algorithm 1 [Metropolis et al., 1953, Hastings, 1970]. This algorithm proposes new values of θ according to a proposal distribution, and then either accepts or rejects them with a probability given by the ratio of the posterior density at the proposed value to the posterior density at the current value in the chain (corrected by the ratio of the proposal densities at the current value to the proposed value)—this is the variable denoted r in Algorithm 1. This procedure causes the sampler to move towards regions of higher posterior density while exploring the parameter space; the equilibrium distribution of the Markov chain constructed in this way is the targeted posterior distribution [Metropolis et al., 1953, Hastings, 1970].

Algorithm 1 Metropolis-Hastings MCMC

Set the initial state $\theta^{(0)}$
Set $k = 0$
while $k <$ Num. iterations **do**
    Draw θ^{prop} according to the proposal density $g(\theta^{\text{prop}}|\theta^{(k)})$
    Calculate $r = \min\left(1, \frac{p(y|\theta^{\text{prop}})p(\theta^{\text{prop}})g(\theta^{(k)}|\theta^{\text{prop}})}{p(y|\theta^{(k)})p(\theta^{(k)})g(\theta^{\text{prop}}|\theta^{(k)})}\right)$
    Draw $u \sim \text{uniform}(0, 1)$
    if $u \leq r$ **then**
        $\theta^{(k+1)} = \theta^{\text{prop}}$ (accept)
    else
        $\theta^{(k)} = \theta^{(k)}$ (reject)
    end if
    Set $k = k + 1$
end while

Assessment of convergence

The set of samples $\{\theta^{(k)}\}$ obtained according to an MCMC sampler such as Algorithm 1 is only informative of the posterior distribution once the Markov chain which generated it has converged to its equilibrium distribution. Convergence can be assessed by running multiple MCMC chains, initialized at different positions in parameter space, and then computing the Gelman $\hat{R}$ statistic [Gelman et al., 2013]. $\hat{R}$ measures the ratio of the variance within a chain to the variance between chains; values of $\hat{R}$ near one (e.g., $\hat{R} < 1.05$) suggest that the different chains may have "mixed" with each other, i.e., are moving around the same region of parameter space; if these chains were initialized in different locations, it is possible that the chains may have converged to a posterior mode. However, small values of $\hat{R}$ may still be obtained even when chains have not converged to the equilibrium distribution (for example, in a multimodal posterior, where all chains have happened to meet in one of the modes but are not exploring the full posterior); for this reason, MCMC algorithms should be deployed with caution, likelihood or posterior surfaces should be visualised to the extent possible, and care should be taken that MCMC chains are initialized in diverse locations in parameter space.

Efficient proposal distributions

The efficiency of the Metropolis-Hastings sampler and its variants is closely tied to the shape of the proposal density, denoted $g(\theta^{\text{prop}}|\theta^{(j)})$ in Algorithm 1. If the proposal distribution generates proposed parameter values which are too widely dispersed, only a small proportion of the proposals will be likely under the posterior, and the rejection rate of the algorithm will be too high. However, if the proposal distribution has too small a variance, the Markov chain will only be able to move around the parameter space slowly.

For this reason, more efficient MCMC samplers do not employ a fixed proposal distribution, and instead automatically tune the proposal based on the shape of the posterior distribution being explored. One algorithm employing this strategy is the HAARIO-BARDENET ADAPTIVE COVARIANCE sampler [Haario et al., 2001, Johnstone et al., 2016]. This sampler uses a multivariate Gaussian proposal distribution, with the covariance matrix of the proposal being set adaptively based on the previously accepted samples in the chain.

Gradient-based sampling

Another strategy for improving the efficiency of MCMC sampling is to use information about the gradient of the posterior with respect to the parameters to generate proposals in regions of parameter space where the posterior density is higher. Such proposals are more likely to be accepted, increasing the efficiency of the chain in exploring the posterior. Additionally, before the chain has converged, the gradient information may help the chain move towards the posterior modes more efficiently than is possible with non-gradient-based samplers, particularly in high dimensional parameter spaces. Some of the standard gradient-based MCMC samplers are HAMILTONIAN Monte Carlo and the NO-U-TURN sampler [Gelman et al., 2013].

Gibbs sampling

In some situations, the conditional posteriors for each element of θ (i.e., $p(\theta_1|y, \theta_2, \ldots, \theta_M)$, and so forth) are easy to sample from, even though the joint posterior of all elements of θ (i.e., $p(\theta_1, \ldots \theta_M|y)$) is still intractable and requires MCMC sampling for inference. In these situations, it may be attractive to employ the GIBBS sampler, which is a special case of Metropolis-Hastings. The Gibbs sampler generates a Markov chain by drawing a new value for each parameter in turn directly from its conditional distribution, conditional on the chain's current values for all the other parameters.

In many problems, the conditional posteriors are not easy to sample from, and the Gibbs sampler is not an appropriate choice. When conditional posteriors are readily available, however, the Gibbs sampler is attractive as it avoids the need for specifying or tuning a proposal density and rejecting proposals. However, it may be inefficient because (in its most simple form) it only updates one parameter at a time, and thus cannot make diagonal moves through the parameter space. In some cases, this deficiency can be addressed by performing sampling from the conditional posteriors of multivariate blocks of parameters rather than each parameter individually.

Another strategy for improving the efficiency of Gibbs sampling is the COLLAPSED Gibbs sampler, in which, when sampling for, say, θ_1, some or all of the other parameters $\theta_2, \ldots, \theta_M$ are marginalized out.

Example 5 *The model has two parameters, θ_1 and θ_2. A collapsed Gibbs sampler is derived in which the next value of θ_1 in the chain is sampled according to $\theta_1^{(k+1)} \sim p(\theta_1|y)$, where $p(\theta_1|y) = \int p(\theta_1|y, \theta_2)d\theta_2$. Thus, θ_1 gets updated without having to depend on the chain's current value of θ_2.*

Efficient collapsed Gibbs sampling may depend on analytic integrability of some of the conditional posteriors, which is often not possible. However, in problems where this is possible (which often involve an appropriate conjugate prior), it can lead to very efficient MCMC samplers by integrating out the dependence on nuisance parameters.

2.3 Forward and inverse problems involving ordinary differential equations

2.3.1 Time series data and ODEs

We assume that time series data $\{y_i\}_{i=1}^N$; $y_i \in \mathcal{Y}_i \subseteq \mathbb{R}^n$; $y = (y_1, \ldots, y_N) \in \mathcal{Y}$ are measured at time points $\{t_i\}_{i=1}^N$. These data are believed to be related to some FORWARD MODEL $\mathcal{F} : \Theta \to \mathcal{X}$, which, for each value of the parameter vector θ, defines some model output $x \in \mathcal{X}$. It is useful to distinguish the output of the forwards model (which is often continuous over time) from the typically discrete observed data (which may also be assumed to depend on other functionals of the model output not included in the forward map), so we introduce the OBSERVATION OPERATOR $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ which maps outputs from the forward model to the sample space. The following example (Example 6) illustrates the usage of these two functions. For a more rigorous treatment of this framework, see [Capistrán et al., 2022].

Example 6 *The data consist of measurements of the size of a population, $y_1, y_2, \ldots, y_N$ recorded at time points $t_1, t_2, \ldots, t_N$. The size of the population $f(t)$ is modelled according to the logistic growth model,*

$$f(t) = \frac{k}{1 + (k/p_0 - 1)\exp(-rt)},$$

with the unknown parameters growth rate r, carrying capacity k, and initial population size p_0 [Clerx et al., 2019]. The forward model is $\mathcal{F}(r, k, p_0) = f(t)$ as defined above, and the observation operation is $\mathcal{H}(\mathcal{F}(r, k, p_0)) = (f(t_1), f(t_2), \ldots, f(t_N))$.

We are specifically interested in forward maps involving deterministic ORDINARY DIFFERENTIAL EQUATIONS (ODEs) in time, i.e., where the model outputs x are the solutions to:

$$\begin{aligned} \frac{dx}{dt} &= h(t, x, \theta); \\ x(t = t_0) &= x_0 \end{aligned} \tag{2.5}$$

for some RIGHT-HAND-SIDE (RHS) function h which is informed by scientific theory, and an initial condition x_0 at the initial time t_0. Eq. (2.5) is an *ordinary* differential

equation because it involves derivatives of the STATE x with respect to one independent variable (in our case, time). Eq. (2.5) has been written as a *first order* equation (i.e., only involving the first order derivative of x); higher order differential equations may be rewritten as systems of first order equations.

ODEs are used throughout the biological and physical sciences to express dynamic processes; a few examples amongst the myriad of their application areas include epidemiology [van der Vegt et al., 2022], hydrology [Kavetski et al., 2003], cardiac electrophysiology [Whittaker et al., 2020], and population dynamics [Shertzer et al., 2002].

2.3.2 Numerical solution of ODEs

For some RHS functions h, eq. (2.5) can be solved analytically, i.e., an analytical expression for $x(t)$ can be derived. However, this is rarely observed for the ODEs used in scientific problems. Instead, most forward maps of interest in scientific applications must be approximated using numerical algorithms.

A wide range of algorithms have been developed which generate an approximation to $x(t)$ when $\mathcal{F}$ involves an ODE [Gautschi, 1997]. Typically, these algorithms first generate a pointwise approximation to $x(t)$ consisting of values $\{\hat{x}_j\}_{j=0}^J$ on a grid of solver time points $\{t_j\}_{j=0}^J$, and then use interpolation of these points to generate an approximate solution $\hat{x}(t)$ at arbitrary time points within the time interval under consideration.

If the numerical scheme is appropriate, $\hat{x}(t)$ approximates $x(t)$ but still inevitably involves some error. The accuracy can be characterized by the LOCAL TRUNCATION ERROR (the error introduced by each iteration of the solver, i.e., in advancing from t_j to t_{j+1}) and the GLOBAL TRUNCATION ERROR (the difference between $\hat{x}(t)$ and $x(t)$ at a particular time t, e.g., at the final time point). The properties of the local and global truncation error depend on the choice of solver used to generate $\hat{x}(t)$.

Example 7 *One of the simplest numerical solvers for ordinary differential equations is the Forward Euler method on a uniform grid. Given a first order differential equation eq.* (2.5) *and the value at the initial condition $x(t = t_0)$, this method constructs an approximate solution $\{\hat{x}_i\}_{j=0}^J$ on a time grid $\{t_i = t_0 + j\Delta t\}_{j=0}^J$ according to:*

$$\hat{x}_0 = x(t = t_0);$$

$$\hat{x}_{j+1} = \hat{x}_j + \Delta t h(t_j, \hat{x}_j), \quad j = 1, \ldots, J-1.$$

The local truncation error of the solver is given by:

$$\begin{aligned} x(t_1) - \hat{x}_1 &= x(t_0) + \Delta t x'(t_0) + \frac{\Delta t^2}{2} x''(t_0) + O(\Delta t^3) - x(t_0) - \Delta t h(t_0, x(t_0)) \\ &= \frac{\Delta t^2}{2} x''(t_0) + O(\Delta t^3), \end{aligned}$$

which, for small Δt, is proportional to Δt^2.

Fixed step size solvers

Most simply, the grid of solver time points, $\{t_j\}_{j=0}^{J}$, may be set in advance, as in Example 7. In the simplest case, the grid would be uniform, corresponding to a uniform solver time step of $\Delta t = t_{j+1} - t_j$. To improve the accuracy of the solution, Δt must be refined to smaller values.

Uniformly spaced grids are likely to be inefficient, because the spacing of grid points needed to achieve a given level of accuracy in the solution depends on the rate at which h is changing in time, and this required spacing could vary significantly over the time interval on which the ODE is being solved. In particular, in regions of time where h is changing rapidly, a high density of solver grid points is required to capture the behaviour of the solution; conversely, when h is changing slowly, larger spacing between grid points can be tolerated without causing much error.

For this reason, non-uniform solver grids are preferred in most problems. Because the regions of time where higher or lower densities of grid points are needed are typically not known in advance, algorithms have been developed which adaptively tune the step size based on the local features of the solution as the ODE is being solved.

Adaptive step size solvers

More sophisticated ODE solvers select the grid of solver time points $\{t_j\}_{j=0}^{J}$ based on the properties of the ODE being solved. Typically, these algorithms work by requiring that the user pre-specify a TOLERANCE, which is some threshold that the local truncation error should not exceed (often expressed as an absolute value and/or a value relative to the magnitude of the solution). At each iteration of the solver, the error in the solution caused by advancing from solver grid point t_j to $t_j + \Delta t_j$ is estimated; if the magnitude of this error exceeds the tolerance, Δt_j is repeatedly refined to a smaller value until the estimated error falls below the threshold. Conversely, if the magnitude of the estimated error already falls significantly below the threshold, a higher value of Δt_j will

be attempted for the subsequent solver iteration.

2.3.3 Inference for ODE parameters

For any given parameter values, deterministic models will always yield the same outputs. Stochastic models, however, incorporate randomness, and repeated simulations of a stochastic model at the same parameter values may yield different outputs. However, both deterministic and stochastic models of biological phenomena may fail to capture all of the observed variation in real observations. This is because many real observations are affected by a variety of influences—for example, fluctuations arising from imperfections in the measurement devices—which are difficult or impossible to incorporate into a mechanistic model of the process. For this reason, forward models are often combined with an additional stochastic component representing otherwise unmodelled elements (for example, processes involved in the measurement of the signal). Many choices are possible for this stochastic measurement or error component.

Assuming a deterministic forward model $\mathcal{F}$, the measurement stochasticity in the observations is often incorporated in an additive form, such that the data are proposed to have been generated according to:

$$y = \mathcal{H}(\mathcal{F}(\theta)) + \varepsilon, \tag{2.6}$$

where ε is an appropriately specified (multivariate) random variable expressing the noise process. In Chapter 8, we study flexible multivariate distributions which may be used to model ε. However, throughout this thesis we also make use of noise processes which we express in the form:

$$y_i = \mathcal{H}_i(\mathcal{F}(\theta)) + \varepsilon_i, \tag{2.7}$$

where $\mathcal{H}_i$ is the ith component of the observation output, and ε_i is a random variable modelling the noise term on observation y_i.

IID Gaussian noise

A standard choice for ε_i is

$$\varepsilon_i \overset{\text{IID}}{\sim} \mathcal{N}(0, \sigma). \tag{2.8}$$

With this choice of ε_i, the log-likelihood of the data is (cf. Example 2):

$$\mathcal{L}(\theta, \sigma|y) = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \mathcal{H}_i(\mathcal{F}(\theta)))^2. \tag{2.9}$$

The IID Gaussian assumption can be justified on several grounds, including the principle of maximum entropy (i.e., that the IID Gaussian is the distribution which makes the fewest additional assumptions beyond the mean and variance of the process), as well as the central limit theorem (i.e., that the noise terms ε_i should be modelled as arising from the additive contributions of many approximately independent random variables) [Lambert et al., 2023]. However, much real time series data does not obey the IID Gaussian noise distribution.

Autoregressive noise

Another choice for ε_i is:

$$\varepsilon_i = \rho\varepsilon_{i-1} + \nu_i \tag{2.10}$$

where the parameter $\rho \in [-1, 1]$, and $\nu \overset{\text{IID}}{\sim} \mathcal{N}(0, \sigma\sqrt{1-\rho^2})$.

This noise process, termed AUTOREGRESSIVE ORDER 1 or AR(1), involves positive correlation between the noise terms for $\rho > 1$. Such correlation may be appropriate when, for example, signals are measured very frequently, or models are misspecified causing persistent underestimation or overestimation of the signal at certain regions of time even at the best fit parameter values [Lei et al., 2020b, Lambert et al., 2023].

Because the distributional assumption for ε determines the form of the likelihood function, the assumed form of the noise process has a significant effect on the shape of the posterior distribution. In particular, failing to account for positive correlation in the noise terms can cause ODE parameter posteriors to have insufficient variance [Lambert et al., 2023]. For this reason, the noise process used to fit a time series should be selected carefully.

2.4 Statistical models of infectious disease

Epidemiological time series data often includes one or more of PREVALENCE, the number of individuals in the population who are infected at a particular time point; INCIDENCE, the number of new infections arising at a particular time point; and DEATHS. A wide variety of models of the spread of an infectious disease through a population have been

developed; selecting an appropriate model and fitting it to incidence or prevalence time series data is essential to understanding and forecasting the spread of a disease, and evaluating the effects of interventions intended to control the spread of a disease.

We divide the most widely used models of infectious disease time series data into three broad categories: AGENT-BASED models, COMPARTMENTAL models, and STOCHASTIC RENEWAL models.

2.4.1 Agent-based models

Agent-based models (ABMs) (or individual-based models, IBMs) involve the simulation of a population of individuals and their actions and interactions. Probabilistic or deterministic rules for how the infection spreads from one individual to another, and how long an infection lasts within an individual, are pre-specified, and these rules are simulated for an artificial population for the specified time interval.

Different ABMs vary widely in how they model the population and infection, and how detailed their simulation rules are. The COVIDSIM model is an illustrative example. CovidSim was initially developed for influenza modelling, and subsequently adapted for COVID-19, where it was influential in determining government policy in England [Adam, 2020, Ferguson et al., 2020a, Ferguson et al., 2006, Ferguson et al., 2020b]. The geographical region under consideration is divided into cells, and cells into microcells which represent the smallest geographical units. Within microcells, individuals are assigned to households and places. Individuals are initially classified as susceptible to the disease; if infected, they progress through a series of disease states representing different levels of severity, and may ultimately die or recover. Infected individuals may transmit the disease to susceptible individuals via a series of transmission modes which are illustrated in Figure 2.1.

Complex agent-based models such as CovidSim are useful for simulating the effects of realistic interventions and studying the interplay of disease transmission and demographic or geographic factors, but they are slow to run and difficult to parameterise. The CovidSim model, for example, contains over 900 parameters [Edeling et al., 2021]. Fitting even a small fraction of these to time series data via methods such as MCMC would be prohibitively computationally expensive. Thus, several categories of simpler, faster models are also employed in epidemiology.

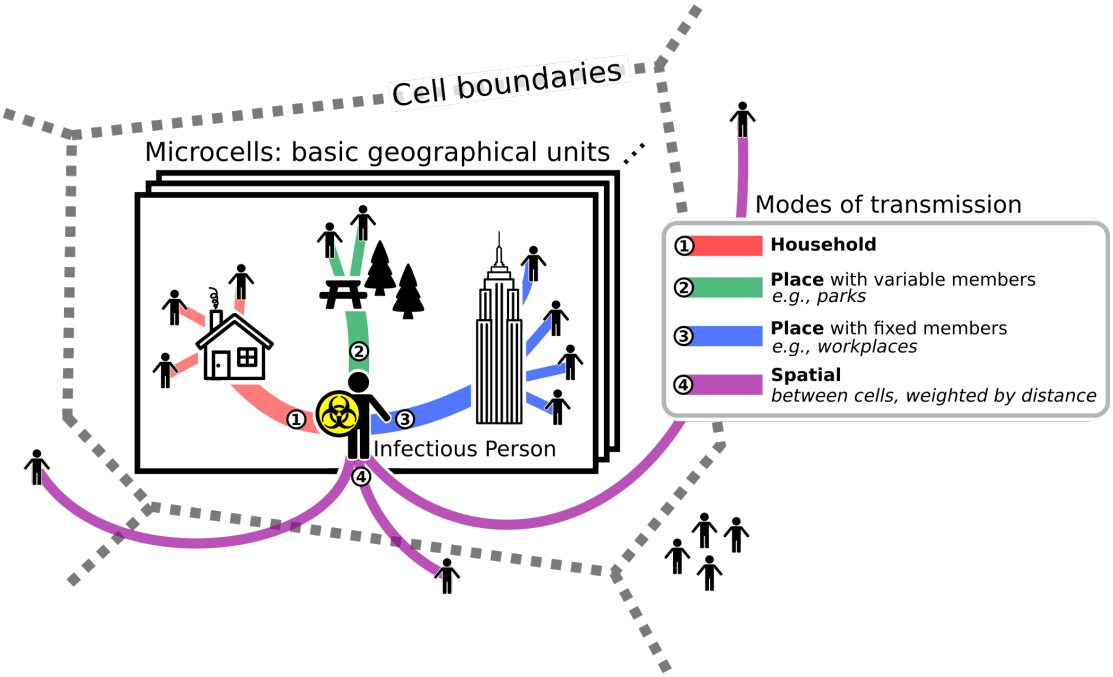


Figure 2.1: Schematic description of the modes of disease transmission implemented in the CovidSim model (this figure from [Gallagher et al., 2022]).

2.4.2 Compartmental models

Compartmental models divide the population into a number of compartments representing different diseased or non-diseased states and specify the rates at which individuals move from one compartment to another. These models are often expressed using differential equations.

Example 8 *A simple deterministic compartmental model for modelling epidemics is the SIR (susceptible-infected-recovered) model [Weiss, 2013]. This model keeps track of the number of susceptible individuals S (those who can be infected with the disease), infected individuals I (those who are currently infectious with the disease), and recovered individuals R (those who have recovered from the disease and are assumed immune). In the simplest case, births and deaths are neglected, and the model is expressed by the following system of differential equations:*

$$\frac{dS}{dt} = -\beta \frac{SI}{N}$$

$$\frac{dI}{dt} = \beta \frac{SI}{N} - \gamma I$$

$$\frac{dR}{dt} = \gamma I,$$

where $\beta > 0$ is the spreading rate of the disease, $\gamma > 0$ is the recovery rate, and $N > 0$ is the total size of the population. The system additionally requires the specification of initial conditions for each compartment ($S(t = 0)$, $I(t = 0)$, $R(t = 0)$).

More sophisticated compartmental differential equation models incorporate further compartments (e.g., an exposed compartment representing individuals who have been infected but are not yet infectious to others) and dynamics (e.g., movement of individuals from the recovered compartment back to the susceptible compartment to represent waning immunity) to relax the simplistic assumptions inherent in the SIR model [van der Vegt et al., 2022].

2.4.3 Stochastic renewal models

Renewal equations assume that cases of an infectious disease arise from historical cases, with the number of new cases arising at a particular time point depending on the historical case counts, the generation time distribution (amount of time between primary and secondary infections), and a time-varying REPRODUCTION NUMBER giving the expected number of secondary infections caused by each primary case [Fraser, 2007, Nishiura and Chowell, 2009, Thompson et al., 2019]. An appropriate discrete probability distribution is used to model the observed cases arising in discrete time (usually daily).

2.4.4 Reproduction numbers

The reproduction number, the expected number of secondary infections caused by each primary case, is a readily interpretible parameter for predicting whether an epidemic will continue to grow and evaluating the effects of interventions or behavioural changes intended to control the spread of a disease. Some epidemiological models (e.g., the stochastic renewal model described above) are directly parameterised in terms of a reproduction number; in other models, the reproduction number may be computed as a function of the inferred model parameters. We distinguish several different formal definitions of the reproduction number in the epidemiological context.

The BASIC REPRODUCTION NUMBER is the expected number of cases generated by one primary case, assuming that everyone in the population is susceptible to the disease. It is often denoted R_0.

TIME-DEPENDENT or EFFECTIVE reproduction numbers indicate the expected number of cases generated by each primary case, at a particular point in time, i.e., accounting

for (amongst other factors) a decrease in the proportion of the population which is susceptible as the disease spreads. Time-dependent reproduction numbers are often denoted R_t.

Two formal definitions of R_t are in wide use [Fraser, 2007]. The case reproduction number measures the expected number of cases generated by a primary case who was infected a time t over the course of their infectious period, accounting for changes in the level of transmissibility that may occur during this period. Meanwhile, the instantaneous reproduction number measures the expected number of cases that would be generated by a primary case who was infected at time t, if the level of transmissibility were to remain constant from time t onward.

Inference of the reproduction number in the presence of heterogeneity between local and imported cases, and the development of flexible models of time variation in R_t, are the subjects of the next two chapters of this thesis (§3 and §4).

2.5 Software implementations

The reliability and reproducibility of research in computational biology depends upon the development of well-documented, tested, and open-source software libraries. Additionally, many of the algorithms described in this chapter and used in this thesis (e.g., MCMC samplers and numerical solvers for ODEs) involve computationally expensive procedures, which require high-quality, optimised software to execute with reasonable runtimes.

For the software implementations developed for this thesis, we draw upon the PROBABILISTIC INFERENCE FOR NOISY TIME SERIES (PINTS) Python library [Clerx et al., 2019]. This library implements classes for expressing time series inference problems, as well as a wide range of optimisation algorithms and MCMC samplers which can be used to perform maximum likelihood estimation and Bayesian inference via MCMC. The models and algorithms described in the rest of the thesis are accompanied by open-source software libraries written primarily in Python, and have been designed to interface with the PINTS classes and MCMC samplers where possible. To increase the reliability and reusability of the developed software, a variety of software engineering techniques including open source GitHub repositories, unit testing, and continuous integration have been employed throughout this thesis. Each chapter (from Chapter 3 to Chapter 8) contains a section indicating details of the software which was developed to perform the research in that chapter.

Chapter 3

Modelling heterogeneity in onwards transmission risk between local and imported cases

Overview

Poisson renewal models are convenient tools for inferring time-varying reproduction numbers (R_t) from incidence time series data. However, these models assume that the risk of onwards transmission is the same for all individuals in the population; when populations are composed of multiple groups behaving in different ways, this assumption may be violated. We introduce a generalisation of a widely used stochastic branching process model of infectious disease incidence to allow heterogeneous behaviour between two groups in the population. Using this model, we focus on the distinction between local cases (those infected in the territory under consideration) and imported cases (those infected elsewhere before travelling to the territory under consideration). Using data from the early COVID-19 outbreak in selected countries and regions worldwide, we show that failing to account for potentially heterogeneous behaviour between local and imported cases may significantly bias inference results for R_t. Finally, we draw on age-structured and transmission network data from Hainan, China and Hong Kong to parameterise our model, and infer more accurate estimates of R_t for these territories accounting for heterogeneity between local and imported cases at the beginning of the COVID-19 outbreak.

Publications

The research presented in this chapter was published as:

- **R. Creswell,†** D. Augustin,† I. Bouros,† H. J. Farm,† S. Miao,† A. Ahern,† M. Robinson, A. Lemenuel-Diot, D. Gavaghan, B. Lambert, and R. N. Thompson: "Heterogeneity in the onwards transmission risk between local and imported cases affects practical estimates of the time-dependent reproduction number," *Philosophical Transactions of the Royal Society, A,* vol. 380 (2022). [Creswell et al., 2022]

  (†= joint first authorship.)

  Contributions: The paper cited above ([Creswell et al., 2022]) was written as part of a collaborative project conducted by the 2020–2021 SABS Epidemiology student cohort in the software engineering module. SABS (the EPSRC Sustainable Approaches to Biomedical Science: Responsible and Reproducible Research Centre for Doctoral Training) assigns first-year doctoral students to small groups working on research and industry-motivated software projects. My role in the epidemiology project was demonstrator and group lead for branching process models. In this role, I worked directly with the students and supervised and contributed to the software development, data processing, data analysis, model derivation, generation of figures, and interpretation of results appearing in the paper; these contributions were made in collaboration with the student working on the branching process who also did much of the coding implementation. My work was additionally performed in collaboration with the other demonstrator on the project (David Augustin), who also derived the generalisation of the posterior distribution of R_t with heterogenous imported cases; with the other SABS Epidemiology students making occasional contributions to software development and data analysis; and with Robin Thompson and the other senior supervisors of the project.

  This thesis chapter uses the same model, datasets, and figures as the publication [Creswell et al., 2022], but much of the text has been rewritten and reorganized. Discussion and analysis of the sliding window method of regularization has been expanded as this is an important comparator method for the results presented in Chapter 4 of this thesis. Other parts of the introduction and discussion have been abbreviated relative to the published paper.

3.1 Introduction

Several summary statistics are widely used (either in near-real time, or retrospectively) to describe the current status of an outbreak of an infectious disease, predict the future number of cases, or evaluate the effects of interventions designed to control the disease [Parag et al., 2022]. One of the most readily interpretable and widely used summary statistics for this purpose is the time-varying reproduction number R_t (introduced in §2.4.4), which specifies the expected number of secondary infections caused by each primary case. Values of $R_t > 1$ indicate that an outbreak will tend to grow, while values $R_t < 1$ indicate that the number of new cases will tend to fall. When, for example, $R_t > 1$, its magnitude immediately indicates what proportion of transmission must be halted to bring an outbreak under control: for example, if $R_t = 2$, an intervention halving the number of transmissions might be targeted in order to prevent further growth of the outbreak. Throughout the COVID-19 epidemic, R_t has been widely estimated in different locations and used to assess appropriate policy responses, or the effectiveness of past interventions (e.g., [Flaxman et al., 2020, Li et al., 2021, Parag et al., 2021, Brauner et al., 2021, Mendez-Brito et al., 2021]).

A variety of modelling approaches can be employed to learn R_t from available data. One approach is to fit stochastic renewal models, which assume that new cases arise from historical cases according to a distribution of generation times representing the time elapsed between a primary and secondary infection [Fraser, 2007, Nishiura and Chowell, 2009]. Our focus in this chapter is on the Poisson renewal model, which assumes that the number of new cases on day t (I_t, or incidence) obeys:

$$I_t \sim \text{Poisson}\left(R_t \Lambda_t\right), \text{ where } \Lambda_t = \sum_{s=1}^{t-1} w_s I_{t-s}. \tag{3.1}$$

In this equation, the w_s terms represent the discretized generation time distribution, i.e., conditional on a primary case infecting a secondary case, w_s is the probability of the primary case taking between $s-1$ and s days to cause the secondary case, and we thus have $\sum_{s=1}^{\infty} w_s = 1$ and $0 \leq w_s \leq 1$. We call Λ_t the transmission potential, and R_t, the unknown parameter of the model, is the time-varying reproduction number. Advantages of the Poisson renewal model include that it requires only incidence data and an estimate of the generation time to fit, and (with the specification of an appropriate conjugate prior) enables computationally efficient Bayesian inference for R_t. Fast and widely used software implementations for learning R_t are available which rely on the

Poisson renewal model [Cori et al., 2013, Thompson et al., 2019].

However, this simple version of the Poisson renewal model assumes that the population is homogenous in its risk of onwards transmission. This may not be the case. Work on the COVID-19 pandemic has indicated that subgroups of the population in different residential settings [Ladhani et al., 2020], with different ages [Thompson et al., 2020, Davies et al., 2020, Keeling et al., 2021b, Lovell-Read et al., 2022, Pooley et al., 2022], or with different vaccination statuses [Keeling et al., 2021a, Sachak-Patwa et al., 2021] have varying risks of transmitting COVID-19 or becoming infected with it. Another key subgroup of the infected population are *imported cases*: those who became infected with the disease elsewhere, before travelling to the region for which R_t is being calculated. Recent work shows that discriminating between local and imported cases is essential for accurate inference of R_t, and that treating imported cases as if they were infected locally may cause significant overestimation of R_t [Thompson et al., 2019]; however, underpinning this previous work is the assumption that local and imported cases are identical in their risks of onwards transmission. This assumption may be violated when local and imported cases behave differently.

In this chapter, we introduce a model that allows local and imported cases to differ in their risks of onwards transmission, and, by fitting this model to data from the COVID-19 outbreak in selected regions worldwide, we demonstrate the importance of accounting for heterogeneity between local and imported cases for accurate estimation of R_t.

The importance of differing transmission risk for the interpretation of an incidence time series is illustrated in Figure 3.1, where we show how a particular incidence time series could have been generated by a range of possible transmission scenarios. This has significant implications for the optimal policy response to control an outbreak of an infectious disease. If transmission is driven by imported cases, the appropriate interventions may include measures such as quarantine of international arrivals; however, if transmission is driven by local cases, more useful interventions would be those that reduce transmission amongst the local population.

3.2 Methods

3.2.1 Bayesian inference for R_t

Our method is an extension of the Cori method for estimating the unknown parameter R_t from incidence data $\{I_t\}$ [Cori et al., 2013, Thompson et al., 2019]. In this method,

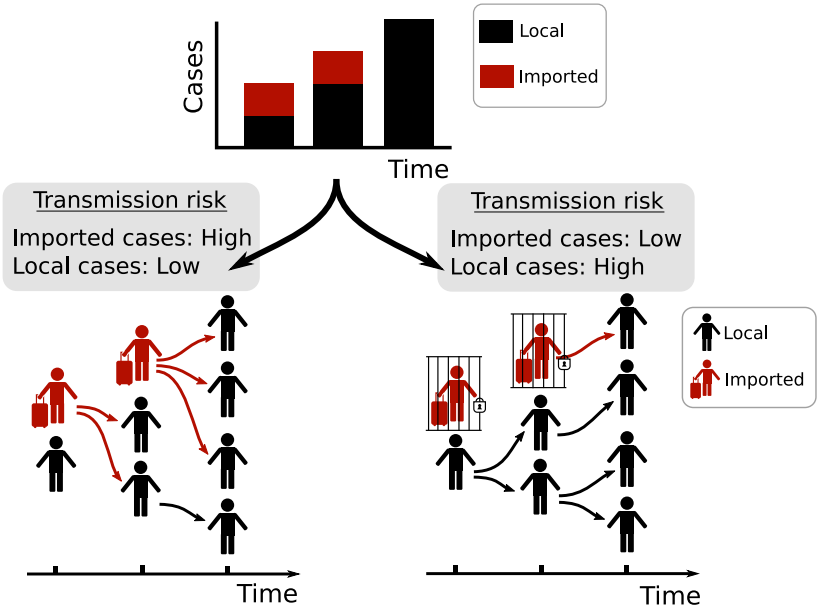


Figure 3.1: **Diagram of heterogeneous transmission between local and imported cases.** Schematic illustration of how a particular incidence time series (top) could have arisen from two different transmission scenarios (bottom left and bottom right). In the bottom left, imported cases are more transmissible than local, and most local infections are infected from imported cases. In the bottom right, imported cases are less transmissible than local (e.g., due to an effective quarantine policy), and transmission is driven by other local cases. Both scenarios lead to the same observed incidence time series but differ significantly in the source of the infections and the appropriate policy response.

R_t is allowed to take a separate value for each day t. A gamma distributed prior is placed on each daily R_t value with shape parameter α and rate parameter β. Using the conjugate relationship between the gamma prior and the Poisson likelihood (eq. (3.1)), the posterior for R_t can be calculated analytically (see Example 4 in Chapter 2).

In [Thompson et al., 2019], the Cori method is extended to separately account for local and imported cases; however, these two groups of cases are still assumed to have the same risk of onwards transmission.

3.2.2 Regularization of the R_t posterior via sliding windows

Naïve computation of the posterior of R_t using the conjugate relationship as described above may lead to highly imprecise estimates of R_t which exhibit spurious fluctuations from day to day, due to the lack of sufficient data to inform precise daily estimates of the parameter.

For this reason, the Cori method regularizes the R_t posterior via a sliding window

heuristic technique. When calculating the posterior distribution for R_t, it is assumed that R_t remained constant for the previous τ days; the set of incidence data $(I_{t-\tau}, \ldots, I_t)$ is thus used to compute the posterior update for R_t. τ is a hyperparameter of the method, and must be tuned by the user.

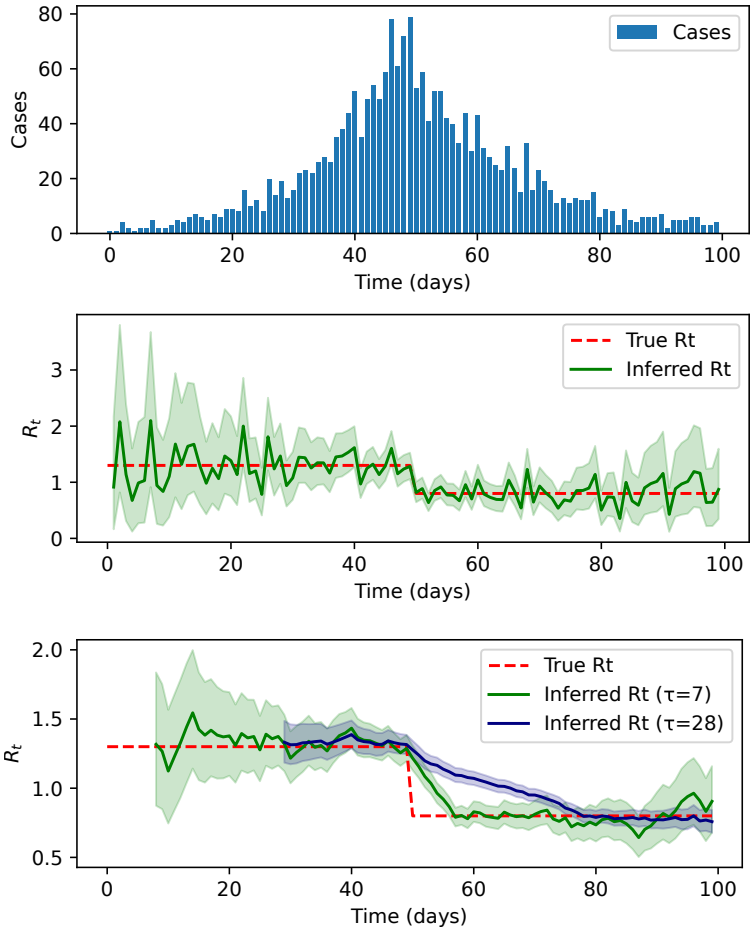


Figure 3.2: **Effect of sliding window width on inference for R_t.** (Top) incidence data generated from the Poisson renewal model, with R_t prespecified as a step function. (Middle) R_t inferred from this data using the Cori method without a sliding window looking backwards. (Bottom) R_t inferred from these data using the Cori method with two different choices of the sliding window width.

By using more data to inform each daily estimate of R_t, the sliding window technique enables smoother and more precise estimates of the parameter and reduces overfitting. However, larger values of the sliding window width τ make the method slower to detect changes in R_t—an example of tradeoff between bias and variance. We illustrate this effect in Figure 3.2. Incidence data was generated from $t = 1$ to $t = 100$ according to the

Poisson renewal model, eq. (3.1), using a step function R_t profile:

$$R_t = \begin{cases} 1.3 & t < 50 \\ 0.7 & t \geq 50, \end{cases}$$

and values of w_s resembling the serial interval of COVID-19 ([Nishiura et al., 2020]; we provide full details of our serial interval in §3.2.5). Then, we used the Cori method to infer R_t from the generated synthetic incidence data. We used a gamma prior on R_t with $\alpha = 1$ and $\beta = 0.2$, resulting in a prior mean and standard deviation both equal to 5. Without using a sliding window, the posterior for R_t is extremely imprecise, and the posterior median fluctuates drastically over time about the true value of the parameter. Conversely, the posteriors obtained using the sliding window ($\tau = 7$ and $\tau = 28$) are smoother in time and have significantly less uncertainty; however, the step change in R_t is not learned accurately but instead a gradual change in R_t is inferred as the sliding window slides over the true change point in the parameter. Thus, tuning of the sliding window width τ must be guided by the desired level of precision in the posterior and the time scale over which changes in R_t are expected.

3.2.3 Modelling heterogeneous transmission between local and imported cases

We assume that each local case will generate an average of R_t infections, while each imported case will generate ϵR_t for some $\epsilon \geq 0$ indicating the relative transmission risk of an imported case compared to a local case. Values of $\epsilon < 1$ indicate imported cases causing on average fewer infections than local cases, while $\epsilon > 1$ indicates that imported cases are more infectious. In this parameterisation, R_t characterizes the reproduction number of local transmission (rather than reflecting an average of local and imported transmission).

We denote the total number of new cases which arise at time step t by I_t. I_t is composed of local cases I_t^{loc} and imported cases I_t^{imp}, i.e., $I_t = I_t^{\text{loc}} + I_t^{\text{imp}}$. We model the expected number of local cases according to:

$$\mathbb{E}\left[I_t^{\text{loc}} | \{I_k^{\text{loc}}\}_{k=0}^{t-1}, \{I_k^{\text{imp}}\}_{k=0}^{t-1}, \epsilon, R_t, w\right] = R_t \sum_{s=1}^{t} (I_{t-s}^{\text{loc}} + \epsilon I_{t-s}^{\text{imp}}) w_s. \tag{3.2}$$

where w is the discretized serial interval distribution, i.e., w_s is the probability that the time between successsive cases in a transmission chain is s time steps (in our applied

modelling we use the *serial interval*, i.e., the time between symptom onsets, rather than the *generation time*, i.e., the time between infections, as it is more directly observable from data on known infectors and infectees, and has a similar mean [Svensson, 2007]). For notational convenience, we define the transmission potential:

$$\Lambda_t(w, \epsilon) = \sum_{s=1}^{t} (I_{t-s}^{\text{loc}} + \epsilon I_{t-s}^{\text{imp}}) w_s.$$

We use the Poisson distribution to model the stochasticity in the number of local cases appearing at each time step. Thus, the likelihood for the local incidence data within the sliding window of width τ, $\{I_k^{\text{loc}}\}_{k=t-\tau}^{t}$, conditional each time step on the previous local and imported incidence data, is given by:

$$P(\{I_k^{\text{loc}}\}_{k=t-\tau}^{t} | \{I_k^{\text{loc}}\}_{k=0}^{t-\tau-1}, \{I_k^{\text{imp}}\}_{k=0}^{t-1}, \epsilon, R_t, w) = \prod_{k=t-\tau}^{t} \frac{(R_t \Lambda_k(w, \epsilon))^{I_k^{\text{loc}}} \exp(-R_t \Lambda_k(w, \epsilon))}{I_k^{\text{loc}}!}. \tag{3.3}$$

We place a gamma prior on R_t with shape hyperparameter α and rate hyperparameter β. Using the conjugate relationship between this prior and the likelihood eq. (3.3), the posterior for each R_t can be computed analytically, according to:

$$p(R_t | w, \epsilon, I_{\leq t}) = \text{gamma}\left(R_t, \alpha + \sum_{k=0}^{\tau} I_{t-k}^{\text{loc}}, \beta + \sum_{k=0}^{\tau} \Lambda_{t-k}(w, \epsilon)\right), \tag{3.4}$$

where, for brevity, $I_{\leq t}$ denotes the historical incidence data $\{\{I_k^{\text{loc}}\}_{k=0}^{t}, \{I_k^{\text{imp}}\}_{k=0}^{t}\}$.

3.2.4 Modelling uncertainty in the serial interval distribution

The discretized generation time distribution w appearing in the posterior for R_t, which we approximate by the serial interval, is a property of the disease and outbreak being modelled, and in our approach the value of w must be specified in order to infer R_t. When neglecting uncertainty in w, eq. (3.4) can be used directly to infer R_t. However, if the knowledge of the serial interval distribution is subject to considerable uncertainty, it may be desirable to propagate this uncertainty to the inference results for R_t. In this section, we describe our procedure, used for the results in this chapter, for incorporating uncertainty in the serial interval distribution into our posterior estimates of R_t.

We assume that uncertainty in w is characterized by a set of n equally plausible serial interval distributions $\{w^{(i)}\}_{i=1}^{n}$. Such a set of samples may arise, for example, from sampling of the posterior distribution for w when learning this parameter from data

of known infector–infectee transmission pairs (e.g., [Nishiura et al., 2020]). For each separate plausible $w^{(i)}$, we compute a posterior distribution for R_t conditional on $w^{(i)}$ according to eq. (3.4). Next, we combine these separate posteriors according to:

$$p(R_t|\epsilon, I_{\leq t}) = \frac{1}{n}\sum_{i=1}^{n} p(R_t|w^{(i)}, \epsilon, I_{\leq t}); \tag{3.5}$$

this procedure approximates a marginalization over the distribution expressing our beliefs about the value of w, and the resultant posterior $p(R_t|\epsilon, I_{\leq t})$ incorporates the uncertainty in the serial interval distribution reflected by the samples $\{w^{(i)}\}_{i=1}^{n}$.

3.2.5 Selection and processing of data

Using our proposed model, we inferred R_t at the beginning of the COVID-19 outbreak in a range of regions worldwide. These regions were considered appropriate for our modelling approach because they published daily incidence data in which local and imported cases where distinguished. The full details of each COVID-19 dataset are presented in §3.3. We additionally inferred R_t for the MERS outbreak in Saudi Arabia; this dataset distinguished between cases infected by other humans (treated as local in our analysis) and cases likely infected from the animal resorvoir of the disease (treated as imported in our analysis).

To infer R_t for COVID-19, we used the serial interval distribution estimated in a previous study by fitting (via MCMC) a log-normal distribution to infector–infectee transmission pairs [Nishiura et al., 2020]. We used the MCMC samples which were obtained using both probable and certain infector-infectee pairs and correcting for right-truncation (the fact that infector–infectee pairs with longer serial intervals may not yet have been observed when the dataset of infector–infectee pairs was collected). We randomly selected $n = 1000$ MCMC samples of the log-normal parameters from [Nishiura et al., 2020] for use in our inference procedure (eq. (3.5)); for each set of plausible log-normal parameters, we obtained a discretized vector of daily values w_s by integrating the continuous serial interval distribution over each day, as described by [Cori et al., 2013], Appendix 11.

3.2.6 Tuning model hyperparameters

We set the hyperparameters of the gamma distribution prior on R_t according to $\alpha = 1$; $\beta = 0.2$. This results in a prior with a mean and standard deviation both equal to five. The prior mean substantially above 1 ensures that we will tend not to infer $R_t < 1$ (i.e.,

the outbreak is under control) unless evidence for this exists in the data, while the large prior standard deviation ensures that the prior is relatively uninformative.

For COVID-19, we tuned $\tau = 6$ representing a weekly sliding window. For MERS, we set $\tau = 27$, due to the smaller magnitude of incidence for that disease in the outbreak studied.

3.3 Results

3.3.1 Effect of differing relative transmissibility between local and imported cases on inference for R_t

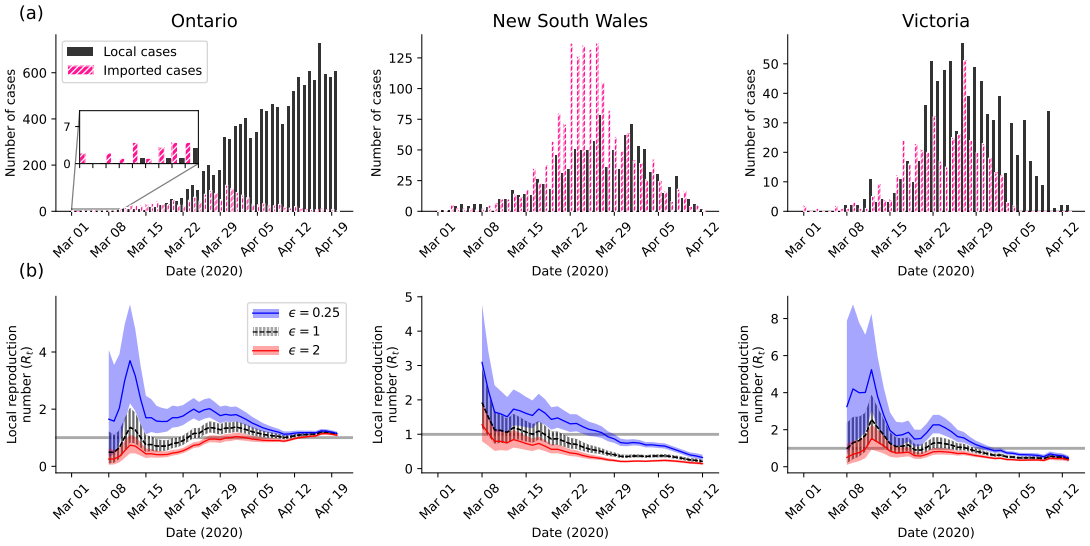


Figure 3.3: **Effect of choices of ϵ on inference for R_t.** (a) Incidence data for COVID-19, with local and imported cases distinguished, for three selected regions: Ontario, Canada (left); New South Wales, Australia (middle), and Victoria, Australia (right). (b) For each region, the inferred profile of R_t values using our model, with three different choices for ϵ: 0.25 (blue), 1 (gray), and 2 (red). Shaded regions indicate the central 95% of the posterior. The gray horizontal line indicates the $R_t = 1$ threshold.

First, we study the effect of assumptions made about the relative transmissibility of local and imported cases on the obtained posterior distributions for R_t (Figure 3.3). We first performed this analysis at the beginning of the COVID-19 outbreak for three regions:

1. **Ontario, Canada** (Figure 3.3a left). Data on the incidence of local and imported cases were obtained from 1 March 2020 to 20 April 2020 [Government of Ontario, 2021]. Any cases who reported travelling outside

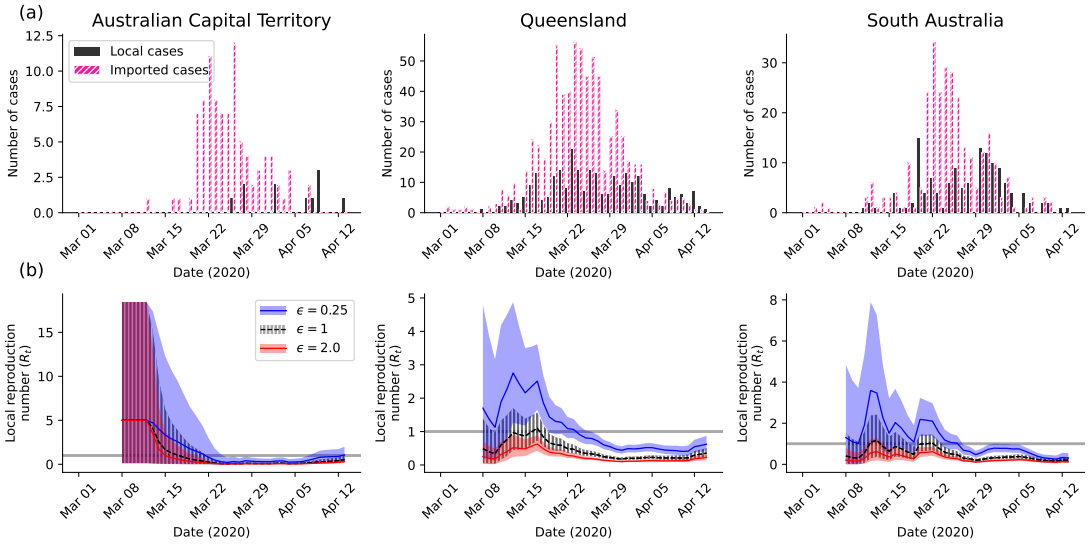


Figure 3.4: **Effect of choices of ϵ on inference for R_t, additional regions I.** (a) Incidence data for COVID-19, with local and imported cases distinguished, for three selected regions: Australian Capital Territory (left); Queensland, Australia (middle), and South Australia (right). (b) For each region, the inferred profile of R_t values using our model, with three different choices for ϵ: 0.25 (blue), 1 (gray), and 2 (red). Shaded regions indicate the central 95% of the posterior. The gray horizontal line indicates the $R_t = 1$ threshold.

Ontario within the 14 day period prior to the onset of symptoms were counted as imported. Any cases whose recent travel was unknown were treated as if they had been infected locally.

2. **New South Wales, Australia** (Figure 3.3a middle). Data on the incidence of local and imported cases were obtained from 1 March 2020 to 13 April 2020 [Price et al., 2020]. Any cases who were reported as "overseas acquired" were counted as imported, and cases with unknown origin were treated as if they had been infected locally.

3. **Victoria, Australia** (Figure 3.3a right). Data on the incidence of local and imported cases were obtained from 1 March 2020 to 13 April 2020 [Price et al., 2020]. Any cases who were reported as "overseas acquired" were counted as imported, and cases with unknown origin were treated as if they had been infected locally.

For each region, we consider three different choices of the parameter ϵ. First, we made the default assumption [Thompson et al., 2019] that local and imported cases have the same transmission risk, i.e., $\epsilon = 1$ (Figure 3.3b, gray). We also considered the

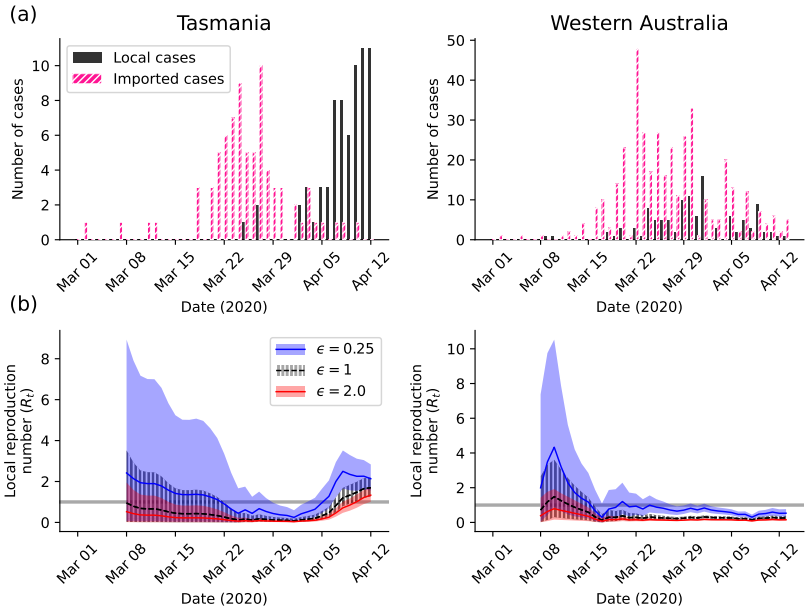


Figure 3.5: **Effect of choices of ϵ on inference for R_t, additional regions II.** (a) Incidence data for COVID-19, with local and imported cases distinguished, for two selected regions: Tasmania, Australia (left); Western Australia (right). (b) For each region, the inferred profile of R_t values using our model, with three different choices for ϵ: 0.25 (blue), 1 (gray), and 2 (red). Shaded regions indicate the central 95% of the posterior. The gray horizontal line indicates the $R_t = 1$ threshold.

situation where imported cases on average generate fewer infections than local ($\epsilon = 0.25$, Figure 3.3b, blue) and where imported cases cause on average twice the infections of local cases ($\epsilon = 2.0$, Figure 3.3b, red). The different choices of ϵ lead to significantly different posterior distributions for R_t. Larger imposed values of ϵ correspond to smaller R_t estimates, as expected because in the case of more infectious imported cases, less transmission must be attributed to the local cases.

Next, we performed the same analysis for a more comprehensive selection of regions where data discriminating between local and imported cases were available. These datasets included:

1. The early COVID-19 outbreak in Australian Capital Territory, Queensland, South Australia, Tasmania, and Western Australia (Figures 3.4 and 3.5). Data were obtained from [Price et al., 2020]. Any cases who were reported as "overseas acquired" were counted as imported, and cases with unknown origin were treated as if they had been infected locally.

2. The early COVID-19 outbreak in New Zealand (Figure 3.6).

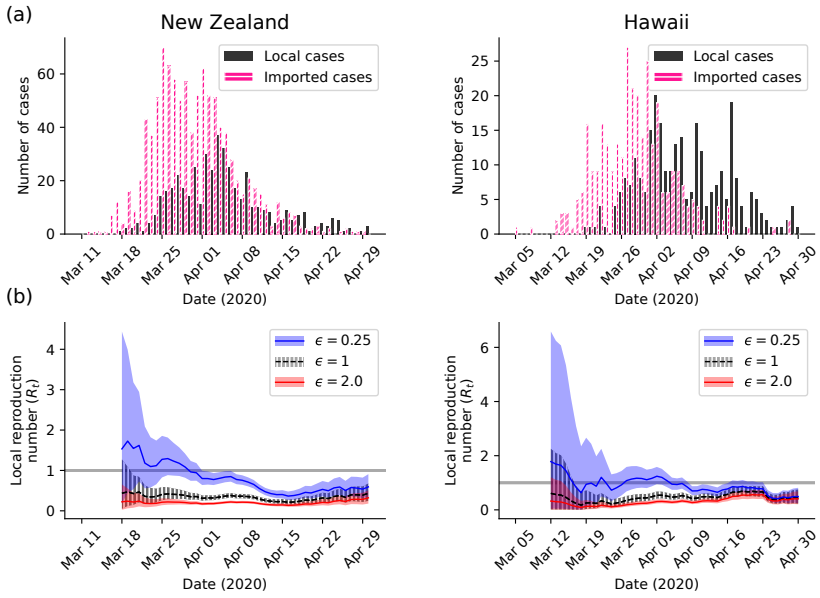


Figure 3.6: **Effect of choices of ϵ on inference for R_t, additional regions III.** (a) Incidence data for COVID-19, with local and imported cases distinguished, for two selected regions: New Zealand (left); Hawaii (right). (b) For each region, the inferred profile of R_t values using our model, with three different choices for ϵ: 0.25 (blue), 1 (gray), and 2 (red). Shaded regions indicate the central 95% of the posterior. The gray horizontal line indicates the $R_t = 1$ threshold.

Data were obtained from the New Zealand COVID Dashboard [Institute of Environmental Science and Research, 2022]. Cases labelled in the dashboard as "imported" or "import-related" were assumed to be imported cases, with all other cases assumed to be local cases.

3. The early COVID-19 outbreak in Hawaii (Figure 3.6). Data were obtained from [State of Hawaii Department of Health Disease Outbreak Control Division, 2022]. Cases labelled in the dataset as having recent "travel history" were assumed to be imported cases, with all other cases assumed to be local cases.

4. The 2014–2015 Middle East respiratory syndrome (MERS) outbreak in Saudi Arabia (Figure 3.7). Data were obtained from [Thompson et al., 2019]. Cases that reported contact with camels were assumed to be imported, while the remainder of cases were assumed to be local. For the analysis of MERS, we used a serial interval given by a gamma distribution with mean 6.8 days and standard deviation of 4.1 days [Thompson et al., 2019]. We additionally used a longer sliding window width of $\tau = 27$.

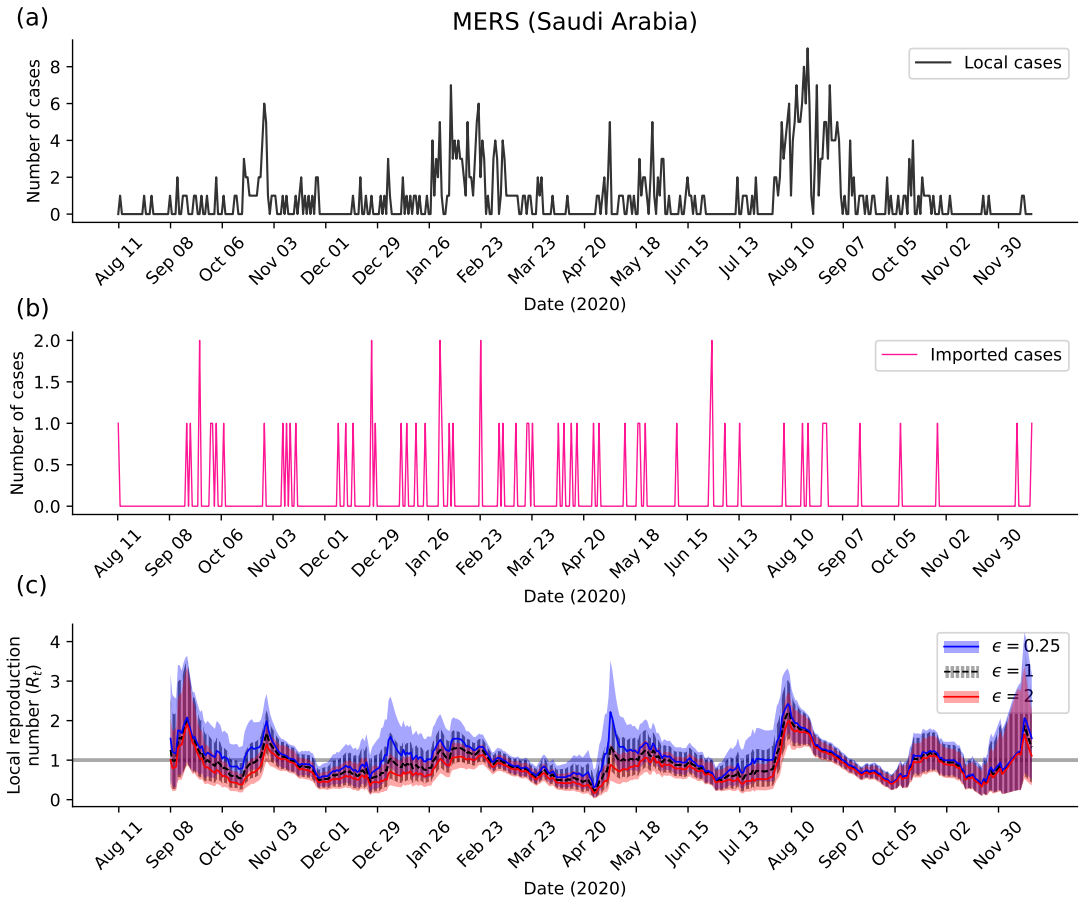


Figure 3.7: **Effect of choices of ϵ on inference for R_t for MERS.** (a) Local incidence data for MERS for Saudi Arabia. (b) Imported incidence data for MERS for Saudi Arabia. The cases labelled as imported are those believed to be infected from the animal reservoir. (c) The inferred profile of R_t values using our model, with three different choices for ϵ: 0.25 (blue), 1 (gray), and 2 (red). Shaded regions indicate the central 95% of the posterior. The gray horizontal line indicates the $R_t = 1$ threshold.

Our results for these additional regions provide further evidence that the choice of ϵ may have a significant effect on R_t estimates. In regions with few local and imported cases (e.g., Australian Capital Territory) the effect of the value of ϵ on R_t is less pronounced, and the posteriors of R_t corresponding to different choices of ϵ largely overlap in certain regions of time. However, for those incidence time series containing enough local cases for R_t posteriors to deviate significantly from the prior, and a large proportion of imported cases relative to the number of local cases, the different choices of ϵ considered here can lead to highly divergent posteriors for R_t: this is particularly apparent in Queensland, Australia and New Zealand, where the posteriors for the different choices

of ϵ show little overlap for much of the time interval considered.

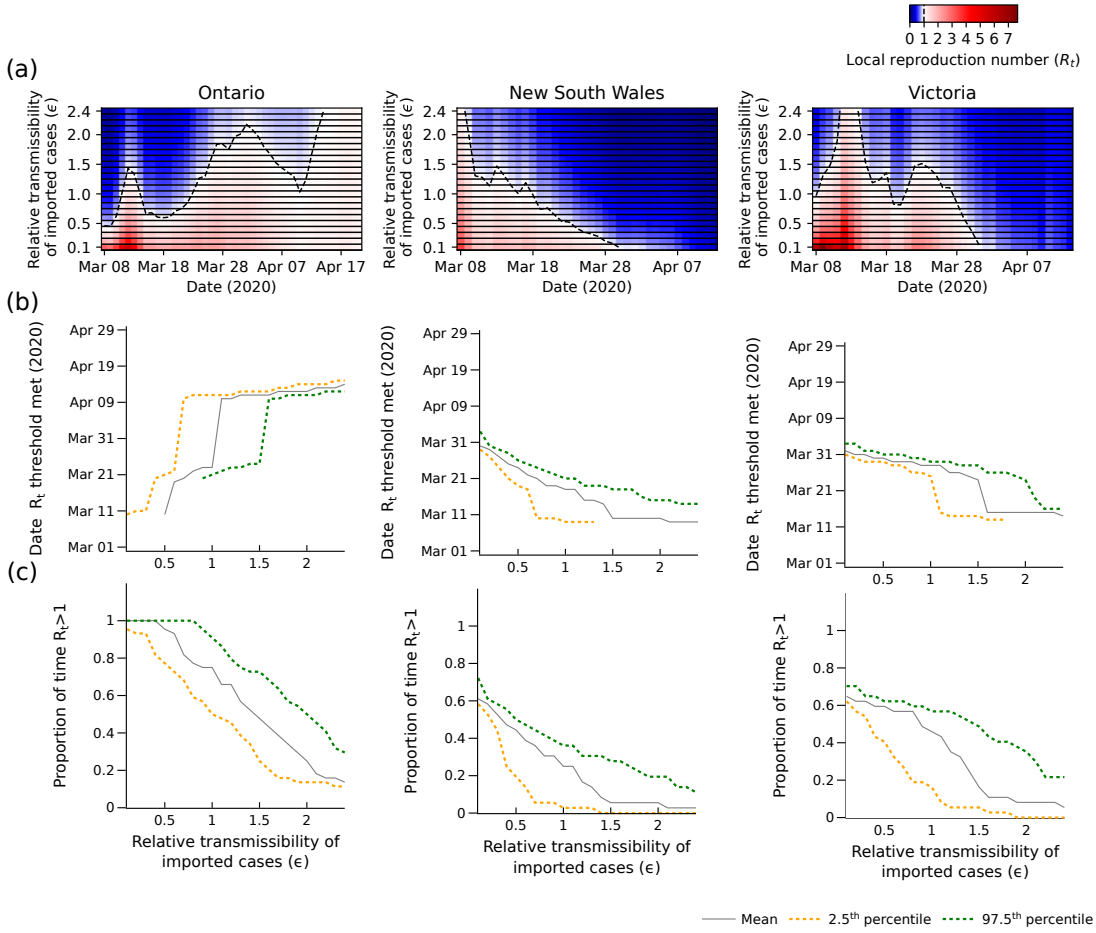


Figure 3.8: **Effect of ϵ on inference for R_t.** (a) The posterior mean R_t for Ontario, Canada (left), New South Wales, Australia (middle), and Victoria, Australia (right) for various values of ϵ (y-axis of each panel). Dashed black lines indicate the $R_t = 1$ threshold. (b) As a function of ϵ, the dates on which the inferred values of R_t cross policy-relevant thresholds. For Ontario, the date indicated is the first date when the estimated R_t value is above one and remains so for the remainder of the time period considered; for New South Wales and Victoria, the date indicated is the first date on which the estimated R_t value is below one and remains so for the remainder of the time period considered. (c) As a function of ϵ, the proportion of the time period that $R_t > 1$. For (b) and (c), the quantities are computed for the posterior mean R_t (gray) as well as the 2.5th percentile (yellow) and the 97.5th percentile (green) of the posterior for R_t.

Next, we considered a continuous range of ϵ values (rather than just three discrete choices of the parameter) for the same three regions that were studied in Figure 3.3. We again computed the effect of ϵ on inference for R_t, and visualised the results with a particular focus on two policy-relevant questions: "Is R_t greater than (or less than)

1?" and "For what proportion of the time was $R_t > 1$?" These results are shown in Figure 3.8. In Figure 3.8a, we plot the posterior mean estimate of R_t for a range of values of ϵ between 0.1 and 2.4; in all three regions considered, the choice of ϵ significantly shifts the times when the posterior mean of R_t is inferred to switch from above to below 1 or vice versa. In Figure 3.8b, we plot as a function of ϵ the first day when the posterior mean of R_t goes above 1 and remains above one for the rest of the time period (Ontario), or the first day when the posterior mean of R_t falls below 1 and remains below it for the rest of the time period (New South Wales and Victoria). Smaller assumed values of ϵ are observed to lead to significantly earlier dates for inferring that R_t has climbed greater than 1 in Ontario, and significantly later dates for inferring that R_t has fallen below 1 in New South Wales and Victoria. Finally, in Figure 3.8c, we plot the proportion of the time period when the posterior mean estimate of R_t is above one; in line with our earlier results, smaller values of ϵ lead one to conclude that R_t is greater than 1 for a larger proportion of the time period.

3.3.2 Realistic values of ϵ

Our results in §3.3.1 demonstrate that the value of ϵ has a significant effect on the recovered posteriors when learning R_t for the COVID-19 outbreak across a wide range of countries and regions. Incidence time series data is not on its own informative of the value of ϵ, making the parameter difficult to set without other sources of data. In this section, we study two regions where additional data which can be used to approximate ϵ is available: Hong Kong (using transmission networks) and Hainan, China (using age-structured contact data).

A previous study in Hong Kong [Liu et al., 2021] reconstructed the transmission network of COVID-19 cases in that country from 23 January 2020 to 8 January 2021. Using their network ([Liu et al., 2021], Table 1), we used the ratio of the outdegree of imported cases (0.74) and the outdegree of local cases (3.68) to approximate the value of $\epsilon = 0.2$. In the left panel of Figure 3.9b, we compare the posterior of R_t computed using this value of $\epsilon = 0.2$ to the posterior of R_t computed using the default value of $\epsilon = 1.0$. The results show that with the default choice of $\epsilon = 1.0$, R_t is significantly underestimated in the later stages of the outbreak relative to the choice of $\epsilon = 0.2$.

Next, we considered Hainan, China, where demographic information for local and imported cases, including their age groups, has been collected in a previous study [Wu et al., 2020]. Using an age-structured contact matrix for China [Prem et al., 2017] (see §5.3), we computed the expected number of daily contacts for local and imported

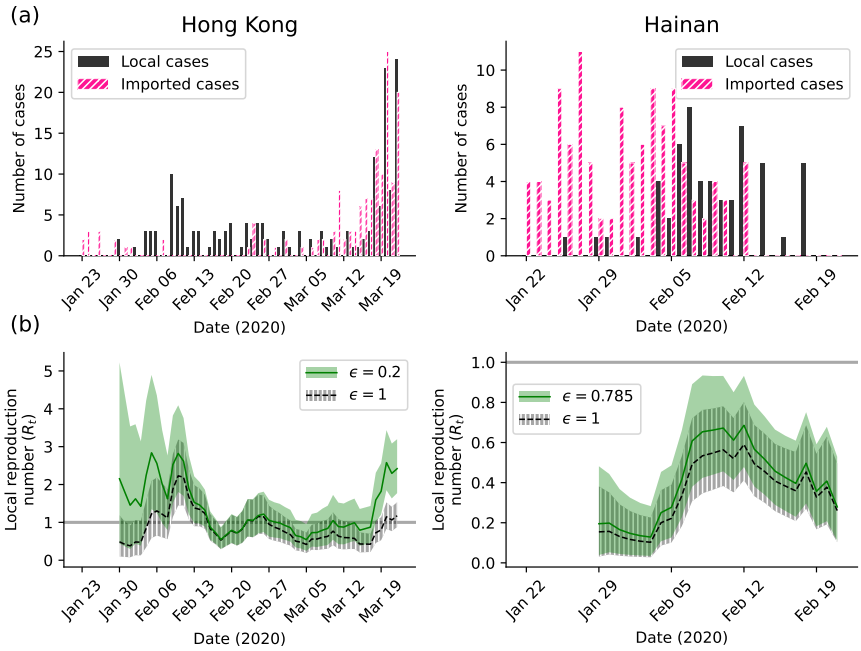


Figure 3.9: **Inference for R_t using realistic values of ϵ.** (a) Incidence data for COVID-19, with local and imported cases distinguished, for two selected regions: Hong Kong (left); Hainan, China (right). (b) For each region, the inferred profile of R_t values using our model, with two different choices for ϵ: 1 (gray), and a value of ϵ estimated based on data which informs this parameter (green, see text for details). Shaded regions indicate the central 95% of the posterior. The gray horizontal line indicates the $R_t = 1$ threshold.

cases based on their ages; we then approximated ϵ as the ratio of the expected number of contacts for imported cases to the expected number of contacts for local cases. This procedure yielded the approximation for Hainan of $\epsilon = 0.785$. In the right panel of Figure 3.9b we compare the posterior of R_t for Hainan using $\epsilon = 0.785$ to that computed using $\epsilon = 1.0$; only minor differences are apparent between the two assumptions.

3.4 Discussion

Summary statistics such as R_t are valuable for monitoring the spread of an infectious disease and informing policy decisions. In this chapter, we have introduced a generalisation of the widely-used Cori method for inferring R_t to account for differing risks of onwards transmission between local and imported cases. By using our model to infer R_t at the beginning of the COVID-19 outbreak in a range of countries worldwide, we have shown that, if local and imported cases differ in their risk of onwards transmission, failing to account for this may cause R_t to be significantly overestimated or underestimated.

We found that our modelling approach was also applicable to the MERS outbreak in Saudi Arabia, in which we treated cases infected by animals (often believed to be camels [Haagmans et al., 2014]) as imported and cases infected amongst the human population as local.

The correct value for the relative transmissibility of imported cases to local cases (ϵ) is difficult to determine in general, and may vary from region to region. Values of ϵ both above and below 1 are plausible: for example, with an effective home quarantine program on all international arrivals, ϵ would be expected to be below 1; conversely, in the absence of such interventions, and in a region where imported cases are more likely to be drawn from the most mobile subsets of the population, values of ϵ greater than 1 would be plausible. In our work, we considered two different approaches for parameterizing ϵ: transmission networks (Hong Kong) and comparing the age structure of the local and imported cases to contact matrix data (Hainan, China). If sufficient data were available, ϵ could also be approximated from, e.g., contact tracing data.

However, an advantage of our approach is that it does not require such data which is often not available. Once plausible values of ϵ have been determined, for example by one of the strategies mentioned above, the method can be applied as long as local and imported cases are distinguished. This differs from the recent approach [Tsang et al., 2021], which is more flexible by allowing independent values of R_t for local and imported cases, but requires the origin of local cases to be available (i.e., local cases infected by imported cases must be counted separately from local cases infected by other local cases): this data is often not available.

In Hong Kong, our model indicated that failing to account for heterogeneity between local and imported cases could lead to significant underestimation of R_t in certain regions of time. This is particularly important, since underestimation of R_t may cause policy makers to inaccurately conclude that a disease outbreak is under control when in fact sustained local transmission is still ongoing. Our results in Hainan, China did not indicate a significant effect of ϵ on inference for R_t; however, our estimate for ϵ in Hainan was likely highly approximate, as it was driven only by the difference in age distribution between local and imported cases. More accurate values of ϵ, informed by, e.g., transmission networks, may lead to different conclusions.

Our model involves several simplifying assumptions, which enable fast inference but may influence the results. First, we assumed that ϵ takes a constant value throughout the time period for which R_t is being inferred. More realistically, ϵ would be allowed to change over time. For example, new policies affecting international arrivals or

changing behaviours in the population could shift the relative transmissibility of local and imported cases. It would be straightforward, however, to adapt our method to involve a time-varying ϵ. Additionally, our model considers only one heterogeneity in the population: that between local and imported cases. In fact, populations are composed of many subgroups who may have differing risks of onwards transmissibility, and more realistic models would account for these factors (e.g., superspreading events). Finally, we assume that perfect knowledge of local or imported status is available for each case; however, in some incidence datasets which discriminate between local and imported cases, some cases are categorized as unknown.

An additional simplifying assumption made in this chapter is that the local cases arising at each day could be described by the Poisson distribution. Due to conjugacy between the Poisson likelihood and gamma prior on R_t, this choice enables fast accurate inference for R_t. However, the Poisson distribution is not an accurate model of *overdispersed* incidence series: those in which variance in daily incidence is larger than the mean. As we discuss further in Chapter 4, further work is necessary to develop and implement efficient inference methods which do not rely on a conjugate prior: this would enable the use of, for example, negative binomial renewal models, which can capture additional variance in the data and would lead to more accurate estimates of posterior uncertainty in R_t when overdispersion is present.

In our model, imported cases may have a different transmissibility but any cases which they infect locally are treated no differently than local cases infected by other local cases. Thus, our model could not be realistically applied to the situation where imported cases and local cases differ in the variants they tend to be infected with, and different variants have differing levels of transmissibility (see, e.g, [Challen et al., 2021]): such situations would require more complex models.

Throughout this chapter, we relied heavily on the sliding window heuristic method to regularize our posterior distributions for R_t. However, our results in Figure 3.2 indicate how the results of the sliding window method depend upon tuning of the window width parameter τ, and the relatively large values of τ used to analyse the real data in this chapter ($\tau = 6$ and $\tau = 27$) may have hurt our ability to detect rapid changes in R_t. In the next chapter, we propose an alternative approach for leveraging information across multiple data points to increase the precision of R_t estimates, while retaining the ability to learn rapid changes in R_t the data provide evidence that such changes exist.

3.5 Data and software

The Python software implementation of the model is available at `https://github.com/SABS-R3-Epidemiology/branchpro`. All data and scripts used to generate the results are available at `https://github.com/SABS-R3-Epidemiology/transmission-heterogeneity-results`.

Chapter 4

A flexible Bayesian nonparametric method for detecting rapid changes in disease transmission

Overview

Inferring precise estimates of the time-varying reproduction number (R_t) from incidence data requires combining information from consecutive time points, using methods such as the sliding window method used in Chapter 3. Drawing upon the *Pitman-Yor process* from Bayesian nonparametrics, and a previously developed framework for restricting this process such that it can be readily applied to time series data, we introduce an alternative inference approach for R_t which we term "EpiCluster." This approach assumes that R_t is piecewise constant, and infers the number of times that R_t changes (and the locations of these changes) from the data. By specifying an informative prior on the hyperparameters governing the process, EpiCluster automatically favours parsimonious fits and prevents overfitting.

First, we use EpiCluster to learn R_t for a range of synthetic examples, and compare its results to the sliding window approach described in Chapter 3 and another state-of-the-art method. We show that EpiCluster is capable of learning highly precise and accurate estimates of R_t. Next, we use EpiCluster to learn R_t for the early COVID-19 outbreak in selected regions worldwide, where it detects changes in R_t corresponding to the introduction of known interventions. Finally, we apply EpiCluster to time series incidence data for measles, SARS, and smallpox, and compare its results to existing methods on these diseases.

Publications

The contents of this chapter were published as:

- **R. Creswell,** M. Robinson, D. Gavaghan, K. V. Parag, C. L. Lei, and B. Lambert: "A Bayesian nonparametric method for detecting rapid changes in disease transmission," *Journal of Theoretical Biology*, vol. 558 (2023). [Creswell et al., 2023a]

  Contributions: I conducted the development of the model and inference algorithm, software development, data analysis, and interpretation and visualisation of results. Figure 1 from [Creswell et al., 2023a], which is reproduced as Figure 4.1 in this chapter, was designed and drawn by Ben Lambert, based in part on discussions with me. Ben Lambert also made contributions to the software, including writing the function to generate synthetic data, and a portion of the R script used to run EpiFilter (an existing method which we use as a comparator for our method). All authors made contributions and suggestions to the writing and revision of [Creswell et al., 2023a], and some of these contributions are reflected in the wording of parts of this chapter.

  The methods and results sections in this chapter follow the equivalent sections of the paper very closely (with most of the supplementary content of the paper included in the results section of this chapter), while the introduction and discussion have been more extensively rewritten for the thesis chapter.

4.1 Introduction

In Chapter 3, we extended the renewal model approach for inference of R_t to account for heterogeneous transmission risks between local and imported cases. We showed that accounting for this heterogeneity is important for accurate inference of R_t. However, in that analysis we continued to rely upon the sliding window heuristic method for regularizing the R_t posterior trajectories (see §3.2.2). As illustrated in our earlier results in Figure 3.2, this heuristic technique helps to increase the precision and smoothness of the posterior estimate of R_t over time, but it comes at a cost of making rapid changes in R_t impossible to learn, particularly when the sliding window width is tuned to a large value.

Existing approaches for inferring R_t have employed several other assumptions about the time variation of the parameter: in addition to the assumption of piecewise-constant R_t within a sliding window of a given prespecified length [Wallinga and Teunis, 2004,

Thompson et al., 2019], smooth variation controlled by a Gaussian filter has also been employed [Abbott et al., 2020, Parag, 2021], as well as the assumption that the R_t is piecewise constant with the optimal number of segments inferred according to a criterion derived from information theory [Parag and Donnelly, 2020].

In this chapter, we introduce a different modelling approach for learning R_t from incidence time series within a renewal model framework. Our method (called EpiCluster) also makes the assumption that R_t is piecewise constant. However, unlike the existing approaches mentioned above, no assumptions are made about the number of times that R_t should change (i.e., the number of constant regimes), the sizes of the regimes (which, in our approach, need not be equal), or the location of the times when R_t jumps to a new value. Instead, we aim to learn an appropriate number and location of regimes from the incidence data itself.

We adopt a Bayesian approach, due to the importance of quantifying uncertainty in R_t and its changes (e.g., Example 3). To learn the division of the R_t values into constant regimes, we use a prior distribution over configurations of the time points into consecutive clusters (each cluster having its own value of R_t) which is derived from the Pitman-Yor process [Pitman and Yor, 1997]. For appropriate values of the hyperparameters of this process, we can incorporate our prior preference for sparsity in the number of changepoints and thus prevent overfitting. Our approach based on the Pitman-Yor process is an example of a Bayesian nonparametric model. Such models are parameterised by a potentially infinite set of parameters; for any particular dataset, the complexity of the model can scale with the complexity and size of that dataset [Ghahramani, 2013].

Although we tune our prior such that sparsity in the number of change points in R_t is preferred, we (by default) express no prior knowledge about the timings of interventions or other changes in R_t, and the locations of any inferred changes in R_t are driven entirely by the data. This differs from existing approaches to assess the effects of interventions, in which particular timings are included explicitly in the model or prior (e.g., [Dehning et al., 2020, Flaxman et al., 2020, Brauner et al., 2021]). This makes EpiCluster a particularly useful tool for retrospectively determining the effectiveness of interventions in an unbiased way, since assumptions concerning intervention timing [Soltesz et al., 2020] or modelling [Sharma et al., 2020] may significantly affect estimates and their interpretation.

Additionally, in order to fit our model to data, we employ a highly efficient MCMC algorithm which samples the assignments of the time points into regimes. We derive this

sampler by exploiting a conjugate relationship between the renewal model likelihood and the prior on R_t. Our inference algorithm enables EpiCluster to be fit to typical disease incidence time series data with a runtime of several seconds to several minutes, depending on the length of the time series and the inferred complexity of the inferred R_t profile.

The rest of this chapter is organized as follows. In §4.2, we provide the full details of our model and inference algorithm. Next, in §4.3 we use EpiCluster to learn R_t from simulated data with known R_t values. Our results show that in both real time and retrospectively, EpiCluster is adept at identifying rapid changes in R_t of the sort that may occur after effective interventions are imposed [Dehning et al., 2020, Flaxman et al., 2020, Brauner et al., 2021]. Our results on synthetic data also show that EpiCluster is able to learn slow, gradual changes in R_t by automatically fitting a "ladder" of piecewise constant steps across the period of change, but alternative methods (e.g., [Thompson et al., 2019, Creswell et al., 2022, Parag, 2021]) may outperform when R_t changes slowly. In §4.3, we use EpiCluster on real data from the COVID-19 outbreaks in Australia and Hong Kong. On these datasets, we detect changepoints in R_t corresponding to the imposition of known interventions. Finally, in the second portion of §4.3, we apply EpiCluster to outbreaks of other diseases.

4.2 Methods

4.2.1 Renewal process model

As discussed in §2.4.4, the instantaneous reproduction number, R_t, represents the average number of secondary cases that would be generated by an infected case at time t assuming that future transmission remains the same as at time t [Fraser, 2007]. We assume that the data consist of a series of daily case counts[1] for each day, t, from $t = 1$ to $t = T$: $\{I_t\}_{t=1}^{T}$ and that the case counts are perfectly known. These case counts are modelled according to Poisson renewal model discussed in Chapter 3, eq. (3.1),

$$I_t \sim \text{Poisson}\left(R_t \Lambda_t\right), \text{ where } \Lambda_t = \sum_{s=1}^{t-1} w_s I_{t-s}, \tag{4.1}$$

[1]Technically, the renewal equation is formulated in terms of infections rather than cases, but, since we use the serial interval distribution in place of the generation time distribution, we keep with defining I_t as a case count.

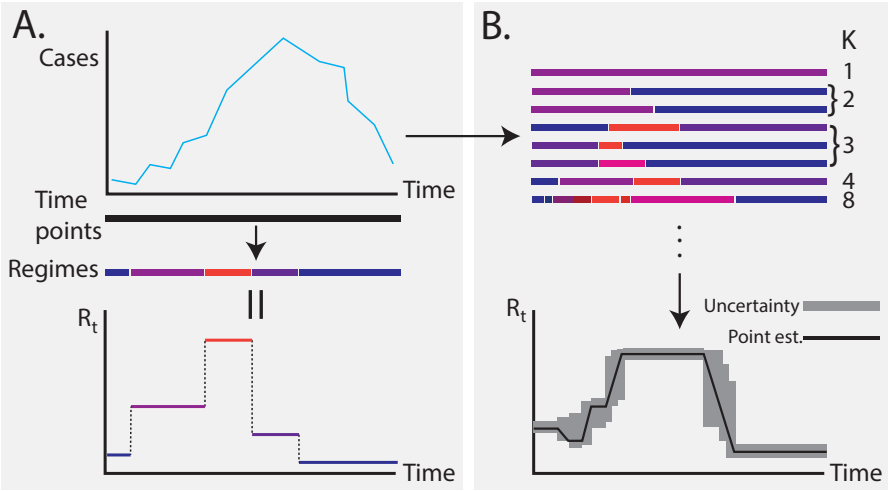


Figure 4.1: **Pitman-Yor based inference for the time-varying reproduction number.** (A) A schematic representation of our modelling assumptions: R_t is piecewise constant over time. (B) We do not prespecify K (the number of pieces of R_t), but instead allow our inference algorithm (see Algorithm 2) to explore the space over partitions (top). This results in posterior uncertainty in the R_t profile (bottom).

where, as before, $R_t \geq 0$ is the time-varying reproduction number on day t, $\Lambda_t \geq 0$ is the transmission potential, and the w_s terms represent the generation time distribution (which we approximate by the serial interval).

4.2.2 Model of changing R_t

Exchangeable partition probability functions and the Pitman-Yor process

Here, we assume that the R_t profile can be decomposed into a number of regimes within which R_t is constant. Our goal is to avoid prespecifying the location of changepoints—representing the boundary between two different R_t regimes—nor their count, since these choices can bias analyses, but rather to learn an appropriate configuration of the time points into regimes using Bayesian inference. We develop a probabilistic model of the division of the time points into regimes. To do so, we use a Pitman-Yor process [Pitman and Yor, 1997][2] to account for a probabilistic decomposition of data points into clusters and, following [Martínez and Mena, 2014], we adjust this model to account for the time series nature of our data. The remainder of this subsection serves as a brief review of this model, starting with a treatment of the nonparametric clustering of unordered data points via *exchangeable partition probability functions* (EPPFs) and followed by appropriate modifications for the time series case (see §4.2.2).

[2] Also known as the two-parameter Poisson-Dirichlet process.

In the standard clustering problem, we have a set $[T] = \{1, \ldots, T\}$ (i.e., the labels of T data points), which we would like to divide into K mutually exclusive subsets $\{A_1, \ldots, A_K\}$ such that $\cup_k A_k = [T]$ where none of the A_k are empty. We denote the set of all such groupings by $\mathcal{P}_{[T]}$; each element of $\mathcal{P}_{[T]}$ is called a *partition*. Random variables Π_T taking values in $\mathcal{P}_{[T]}$ are termed *random partitions* of $[T]$. A random partition has the property of *exchangeability* if its probability distribution can be written as a symmetric function p of the subset sizes, i.e.,

$$\text{Prob}(\Pi_T = \{A_1, \ldots, A_K\}) = p(n_1, \ldots, n_K)$$

where $n_k = |A_k|$ (i.e. n_k is the size of the subset, A_k).

Under these conditions p is known as an EPPF. A more complete treatment of the concept of EPPFs can be found in [Pitman, 2002, Lijoi and Prunster, 2010]. A fairly general EPPF, which we will employ in this work, is derived from the Pitman-Yor process, a generalisation of the Dirichlet process [Teh, 2010]. This EPPF is given by [Pitman, 2002, eq. (3.6)]:

$$p(n_1, \ldots, n_K | \theta, \sigma) = \frac{\prod_{i=1}^{k-1}(\theta + i\sigma)}{(\theta+1)_{T-1\uparrow}} \prod_{j=1}^{K} (1-\sigma)_{n_j - 1\uparrow}, \tag{4.2}$$

where $x_{m\uparrow} := \prod_{j=0}^{m-1}(x+j)$, and $\sigma \in [0, 1)$ and $\theta > -\sigma$ are the two hyperparameters governing the process: σ is called the discount parameter, which essentially controls how the number of regimes, K, grows with the size of the dataset; θ is called the strength parameter with larger values giving greater weight to series with more regimes. In the limit $\sigma \to 0$, a Pitman-Yor process becomes a Dirichlet process which permits a slower growth (of order $\log T$ as opposed to T^σ) in the number of regimes with increases in data size [Pitman, 2002, section 3.3].

Applicability of EPPFs to time series problems

Unlike the general clustering problem, in the time series case, the data points have an ordering which the clusters must respect. For example, consider an incidence series of length three: (I_1, I_2, I_3). For this series, allowable effective reproduction number allocations include: $\{\{I_1, I_2, I_3\}\}$, where all the data points are generated from a process with the same effective reproduction number: i.e. there is a single regime ($K = 1$); $\{\{I_1\}, \{I_2, I_3\}\}$, where the first data point was generated from a process with one effective reproduction number and the latter two data points from a process with a different

one: i.e. there are two regimes ($K = 2$); $\{\{I_1, I_2\}, \{I_3\}\}$, where the first two points are grouped; and $\{\{I_1\}, \{I_2\}, \{I_3\}\}$, where each data point is generated from a process with a different reproduction number: i.e. there are three regimes ($K = 3$).

An allocation which would be disallowed is: $\{\{I_1, I_3\}, \{I_2\}\}$, where the first and third data points come from the same process which is distinct from that governing the second. Whilst, it is possible that transmission could return to a previous level, it is an assumption of our modelling process that only consecutive data points share the same R_t. By avoiding recurrence to historical regimes, we ensure that the changepoints identified are straightforward to interpret.

For a given EPPF, p', we can obtain a distribution p which is supported only on those partitions which respect an ordering of the labels using the following result [Martínez and Mena, 2014]:

$$p(n_1, \ldots, n_K) = \begin{cases} \frac{1}{K!} \binom{T}{n_1, \ldots, n_K} p'(n_1, \ldots, n_K), & \text{if allowable partitioning} \\ 0, & \text{otherwise,} \end{cases} \tag{4.3}$$

where the large bracketed term indicates the multinomial coefficient.

Combining eqs. (4.3) and (4.2), we obtain the following result for the prior distribution on the sequence of regime sizes in the time series case:

$$p(n_1, \ldots, n_K | \theta, \sigma) = \frac{T!}{K!} \frac{\prod_{i=1}^{k-1} (\theta + i\sigma)}{(\theta + 1)_{T-1\uparrow}} \prod_{j=1}^{K} \frac{(1 - \sigma)_{n_j - 1\uparrow}}{n_j!}. \tag{4.4}$$

4.2.3 Hyperparameters of the process

In order to learn parsimonious assignments of the time points into regimes, our prior, given by eq. (4.4), should favour configurations consisting of longer regimes. We favour longer regimes because they mitigate against overfitting—for typical data, the likelihood of the renewal process would be maximised by assigning each time point to its own cluster with an idiosyncratic value of R_t; the resulting profile of R_t values will tend to be jagged and exhibit spurious fluctuations. Additionally, longer regimes have the advantage of allowing more data to be leveraged in order to learn more precise estimates of R_t. However, by favouring longer regimes, it is possible that we miss shorter term fluctuations in R_t—this is akin to the issue of choosing window lengths for a number of existing methods; see, e.g., [Thompson et al., 2019].

Eq. (4.4) induces a marginal distribution over the number of clusters whose mean

has been derived as [Martínez and Mena, 2014, Pitman, 2002, eq. (3.13)]:

$$\mathbb{E}[K] = \frac{(\theta + \sigma)_{T\uparrow}}{\sigma(\theta + 1)_{T-1\uparrow}} - \frac{\theta}{\sigma}, \tag{4.5}$$

for $\sigma \neq 0$. For small values of the hyperparameters θ and σ, $\mathbb{E}[K]$ is significantly smaller than the number of time points T, and the marginal distribution of K places little weight on values of K close to T, thus preferring sparsity in the number of clusters.

4.2.4 Tuning process hyperparameters

As discussed above, values of θ and σ such that *a priori* there are expected to be substantially fewer clusters than time points are attractive as they can help prevent overfitting. These parameter values may be tuned based the degree of prior knowledge about changes in R_t: if, *a priori*, R_t is not expected to change over the time period where inference is being performed, θ and σ should be chosen such that the marginal prior on the number of regimes is centered near 1 regime; alternatively, if R_t is *a priori* expected to fluctuate drastically, θ and σ can be chosen such that the marginal prior on the number of regimes places more weight on greater numbers of regimes.

We use the following heuristic strategy to tune θ and σ for the results in this chapter: we set $\theta = 0$ and choose σ as a function of T such that $E[K] = 1.5$ (with the appropriate value of σ selected by numerical optimisation of eq. (4.5)); this represents a prior belief that R_t is generally constant over the time series, but allows flexibility to add clusters when the data provides evidence that they are needed. For a time series of length $T = 100$, our choice of prior hyperparameters induces a marginal distribution over the number of clusters whose 2.5$^{\text{th}}$ percentile is one cluster and 97.5$^{\text{th}}$ percentile is four clusters. In Figure 4.3 and Figure 4.12 (see §4.3 below), we use EpiCluster to perform inference for R_t for a grid of different values of the hyperparameters θ and σ. As discussed below, in these results, we observe some degree of sensitivity to hyperparameter choices, but the existence of a wide range of hyperparameter values for which estimated R_t values remain broadly compatible, with similar changepoint locations.

4.2.5 Marginal likelihood of the data

In this section, we calculate the marginal likelihood of the data conditional on a particular arrangement of the time points into regimes, which involves integrating out R_t with respect to its prior distribution. This marginal likelihood enables efficient inference for

the posterior distribution over regime configurations via MCMC sampling (see § 4.2.6).

The marginal likelihood for an incidence series conditional on a particular set of subset sizes $n_1, \ldots, n_K$ (see §4.2.2) can be written as a product of marginal likelihoods for each regime:

$$p(I_1, \ldots, I_T | n_1, \ldots, n_K) = \prod_{k=1}^{K} L_k(I_{k,1}, \ldots, I_{k,n_k} | I_{-k}), \tag{4.6}$$

where $I_{k,j}$ denotes the jth data point in regime k, and L_k is the marginal likelihood of the data in the kth regime, which we assume is conditional on all cases observed prior to regime k (denoted by I_{-k}). We derive the regime-specific marginal likelihoods using the renewal model (eq. (4.1)):

$$L_k(I_{k,1}, \ldots, I_{k,n_k} | I_{-k}) = \int_0^\infty p(R_k) \prod_{j=1}^{n_k} \text{Poisson}(I_{k,j} | R_k \Lambda_{k,j}) dR_k,$$

where $\Lambda_{k,j}$ is the transmission potential calculated for the jth time point in regime k, R_k is the value of the effective reproduction number for the kth regime, and $p(R_k)$ is the prior on R_k.

We choose a gamma distribution prior for R_k with shape parameter α and rate parameter β.[3] With this choice of prior, the integral in the formula for the regime-specific marginal likelihood can be evaluated analytically, resulting in:

$$\begin{aligned} L_k(I_{k,1}, \ldots, I_{k,n_k} | I_{-k}) = & \frac{\beta^\alpha}{\Gamma(\alpha)} \Gamma\left(\alpha + \sum_{j=1}^{n_k} I_{k,j}\right) \left(\beta + \sum_{j=1}^{n_k} \Lambda_{k,j}\right)^{-\left(\alpha + \sum_{j=1}^{n_k} I_{k,j}\right)} \\ & \times \prod_{j=1}^{n_k} \frac{\Lambda_{k,j}^{I_{k,j}}}{I_{k,j}!}, \end{aligned}$$

where $\Gamma(\cdot)$ is the gamma function.

Additionally, with the gamma prior on R_k, the posterior distribution of each R_k, conditional on the data assigned to regime k, is given by the conjugate gamma posterior (see Chapter 3):

$$p(R_k | I_{k,1}, \ldots, I_{k,n_k}, I_{-k}) = \text{gamma}(R_k | \text{shape} = \alpha + \sum_{j=1}^{n_k} I_{k,j}, \text{rate} = \beta + \sum_{j=1}^{n_k} \Lambda_{k,j}). \tag{4.7}$$

[3] $p(R|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} R^{\alpha-1} e^{-\beta R}$.

As prior hyperparameters, we follow the same procedure as in Chapter 3 and select $\alpha = 1$ and $\beta = 0.2$. With this choice, the prior mean and standard deviation are both equal to 5. The high standard deviation provides a relatively uninformative prior, and the high mean ensures that the outbreak is unlikely to be determined as under control (since $> 81\%$ of prior probability is for $R_t > 1$) unless there is considerable evidence to suggest otherwise.

4.2.6 Inference

At particular values of the hyperparameters σ and θ, the target posterior of regime configurations, which we denote by $p(n_1, \ldots, n_K | I_1, \ldots, I_T, \sigma, \theta)$ is proportional to the product of eq. (4.4) and eq. (4.6):

$$p(n_1, \ldots, n_K | I_1, \ldots, I_T, \sigma, \theta) \propto p(I_1, \ldots, I_T | n_1, \ldots, n_K) \times p(n_1, \ldots, n_K | \theta, \sigma).$$

For brevity, we suppress the dependence on cases and hyperparameters and denote the unnormalised posterior by $p(\gamma_K)$, where $\gamma_K := (n_1, \ldots, n_K)$ indicates a particular configuration of the time points into K regimes.

Inference for this posterior is performed via Markov Chain Monte Carlo (MCMC) which provides a distribution over the number of regimes by jumping between models of different numbers of parameters. We use the same split-merge-shuffle structure as [Martínez and Mena, 2014]. Each step of our MCMC algorithm is given in Algorithm 2, and we now describe it.

Different configurations of the time points into regimes are explored through the use of *split*, *merge*, and *shuffle* proposals. The split proposal takes an existing regime and proposes to split it into two regimes at some randomly located changepoint. The merge proposal takes two consecutive regimes and proposes to merge them into one. Both of these proposals consider an update to the total number of regimes, thus allowing the sampler to explore the marginal posterior distribution over the number of regimes. Additionally, the shuffle proposal shifts the boundary between two consecutive regimes, thus keeping the same number of regimes but efficiently exploring uncertainty in the location of a changepoint. At each iteration of the MCMC sampler, we make one shuffle proposal and randomly choose whether to make a split or merge proposal, with the MCMC tuning parameter q giving the probability of making the split proposal. For the results presented in this chapter, we fix $q = 0.5$. The acceptance probabilities for the split, merge, and shuffle proposals are derived in [Martínez and Mena, 2014] and are

given by $\min(1, \alpha_e)$, with $e \in \{\text{split}, \text{merge}, \text{shuffle}\}$.

α_{split} is calculated by:

$$\alpha_{\text{split}} = \begin{cases} (1-q)(T-1)\frac{p(\gamma_{K+1})}{p(\gamma_K)}, & \text{if } K = 1, \\ \frac{1-q}{q}\frac{p(\gamma_{K+1})}{p(\gamma_K)}\frac{n_{\text{splittable}}(n_s-1)}{K}, & \text{if } K > 1, \end{cases}$$

where $n_{\text{splittable}}$ is the number of splittable regimes (i.e., those with more than one time point assigned to them) in the original configuration, and n_s is the length of the regime selected for a split; γ_K is the current regime configuration, and γ_{K+1} is the split configuration.

The corresponding quantity for a merge move is given by:

$$\alpha_{\text{merge}} = \begin{cases} \frac{q}{1-q}\frac{p(\gamma_{K-1})}{p(\gamma_K)}\frac{K-1}{n^*_{\text{splittable}}(n_s+n_{s+1}-1)}, & \text{if } K < T, \\ q(T-1)\frac{p(\gamma_{K-1})}{p(\gamma_K)}, & \text{if } K = T, \end{cases}$$

where $n^*_{\text{splittable}}$ is the number of splittable regimes in the proposed configuration, and n_s and n_{s+1} are the sizes of the regimes which are proposed to be merged; γ_{K-1} is the merged regime configuration.

The equivalent quantity for a shuffle move is given by:

$$\alpha_{\text{shuffle}} = \frac{p(\gamma^*_K)}{p(\gamma_K)},$$

where γ^*_K is the shuffled configuration obtained from γ_K as described in Algorithm 2.

The values of R_t are updated using Gibbs steps conditional on the current regime configuration using eq. (4.7).

For all EpiCluster results presented in this chapter, we run four separate MCMC chains, two initialised with all time points assigned to a single regime (i.e. $K = 1$) and the other two initialised with all time points assigned to their own singleton regime (i.e. with $K = T$). We assessed convergence of our MCMC algorithm (Algorithm 2) by monitoring convergence in K, the number of regimes. To do so, we computed the $\hat{R}$ statistic [Gelman et al., 2013] and required $\hat{R} < 1.05$. Once convergence was determined, we discarded the first 50% of each of the MCMC chains as warm-up and combined the rest of the samples in order to calculate posterior percentiles and means.

Algorithm 2 One step of the MCMC sampler.

$q \leftarrow$ User specified value (MCMC tuning parameter)
$K \leftarrow$ Current number of regimes
for k in $1, \ldots, K$ **do** ▷ *Update the R_t via Gibbs steps.*
    Draw a value for R_t in the kth regime from its conditional posterior, eq. (4.7).
end for
if $K = 1$ **then**
    $q \leftarrow 1$
else if $K = T$ **then**
    $q \leftarrow 0$
end if
$Sp \sim$ Bernoulli(q) ▷ *Draw binary variable to allow random choice between split and merge proposals.*
if $Sp = 1$ **then** ▷ *Perform a split proposal.*
    Uniformly at random propose a regime to split.
    Uniformly at random propose an index within that regime at which to split.
    Accept the split regime configuration with probability α_{split}.
else ▷ *Perform a merge proposal.*
    Uniformly at random propose a regime (not the last) which will be merged with following regime.
    Accept the merged regime configuration with probability α_{merge}.
end if
$K \leftarrow$ Current number of regimes
if $K > 1$ **then** ▷ *Perform a shuffle proposal.*
    Uniformly at random propose a regime j (not the last) to shuffle.
    Uniformly at random propose an index within either regime j or $j + 1$ to be the new changepoint between these two regimes.
    Accept the shuffled regime configuration with probability α_{shuffle}.
end if

4.2.7 Comparator methods

In §4.3, we compare the posterior distribution for R_t yielded by our nonparametric method to those yielded by two comparator methods. This first is the Cori sliding window method from Chapter 3 [Cori et al., 2013, Thompson et al., 2019], which assumes that R_t is constant over a sliding window of τ days looking backwards. The sliding window width has a significant effect on the posterior and the effective bias-variance trade-off. As a result, we consider two choices of τ (7 days and 28 days) when applying the method to synthetic data. The second comparator is the EpiFilter method [Parag, 2021], which applies sequential Bayesian smoothing and controls change in R_t under a random walk prior.

Runtimes

We ran the sliding window method using the `branchpro` Python package [Creswell et al., 2022]. We ran the EpiFilter method through its R package [Parag, 2021]. Using our software library and typical consumer hardware (3.6GHz CPU), EpiCluster takes from several seconds to several minutes to learn the posterior,

depending on the complexity of the R_t profile. By comparison, the sliding window method and EpiFilter methods are effectively instantaneous to compute on the time series studied here.

4.2.8 Handling imported cases

Some of the real data examples we consider (see §4.3) consist of case counts in locations where a substantial proportion of the case loads are due to imported cases. To account for this, we adapt our renewal model using the methods described above in Chapter 3 (i.e., [Creswell et al., 2022]). In summary, cases are classified as either *local* or *imported*. Local cases $\{I_t^{\text{loc}}\}_{t=1}^T$ (denoted simply I_t for brevity) are those arising from local transmission in the spatial region under consideration, while imported cases $\{I_t^{\text{imp}}\}_{t=1}^T$ are those who were infected elsewhere before travelling to the region. Thus, imported cases contribute to local transmission, but did not arise from it. We allow local and imported cases to have different risks of onwards transmission by weighting the imported cases by some number $\epsilon > 0$, as defined in Chapter 3, and we set ϵ to appropriate values (see §4.2.9 and §3.3.2). The default choice of $\epsilon = 1$ corresponds to an equal risk of onwards transmission between local and imported cases. Note, any case and any subsequent lineages begot by an imported case are classified as local: it is only the rate at which newly imported cases infect others which is assumed to differ from purely local transmission.

We adapt eq. (4.1) to model the dynamics of local cases I_t, resulting in:

$$I_t \sim \text{Poisson}\left(R_t \sum_{s=1}^{t-1} w_s (I_{t-s} + \epsilon I_{t-s}^{\text{imp}})\right), \tag{4.8}$$

where R_t is the effective reproduction number that characterises local transmission on day t. For problems where imported cases are not considered, we use eq. (4.1).

4.2.9 Real incidence data

We fit to real case incidence data for local and imported COVID-19 cases for three regions: Victoria and Queensland in Australia and Hong Kong. In each of these three locations, we used cases with dates given by the date of symptom onset. We selected these regions as they exhibit a variety of different trends in R_t: a gradual decrease in Victoria, a more rapid decrease in Queensland, and a fall in R_t followed by the sudden appearance of a second wave in Hong Kong. Data for the Australian regions were obtained from the Australian national COVID-19 database [Price et al., 2020]; data

for Hong Kong were obtained from the Hong Kong Department of Health COVID-19 database [Hong Kong Department of Health, 2022]. For the Australian states, cases of unknown origin were assumed to be local, and in Hong Kong, all cases other than those listed as "imported case confirmed" were treated as local.

The proportion of cases whose local or imported status is unknown varies significantly from region to region. For the time periods we considered, 57% of cases in Victoria, 8% of cases in Queensland, and 20% of cases in Hong Kong were not confirmed as either local or imported in the datasets and thus were treated as local. This assumption, if incorrect, would tend to bias our estimates for the reproduction number towards larger values, as we attribute as many cases as possible to local transmission.

We assumed $\epsilon = 1$ in eq. (4.8) for Victoria and Queensland; however, for Hong Kong, transmission networks suggest that imported cases were significantly less infective than local cases [Liu et al., 2021], so we set $\epsilon = 0.2$, according to the methods described in §3.3.2. In all three instances, we assumed that under-reporting and delays were negligible given the strong surveillance in these countries. We note that since different assumptions for ϵ tend to shift rather than warp the inferred R_t series, they are unlikely to affect the position of changepoints.

Whereas in Chapter 3 we incorporated uncertainty in the serial interval distribution into our estimates of R_t (see §3.2.4), in this chapter we neglect uncertainty in w and use a single distribution for this parameter. The method described in §3.2.4 could be straightforwardly adapted to EpiCluster, if uncertainty in the serial interval were considered highly important, but this would come at the cost of significant extra computational runtime.

4.3 Results

EpiCluster reliably estimates sudden changes in R_t in retrospective analyses

To evaluate the performance of our model, we generated synthetic incidence data using eq. (4.1) where the R_t profile was known (see Figure 4.2). We considered three R_t profiles: one with a precipitous decline in R_t ("fast drop-off"); another, with a decline in R_t followed by a later resurgence ("fast resurgence"; we included this profile since resurgences may be more difficult to infer than declines in transmission strength [Parag and Donnelly, 2022]); and another with a more gradual decline in R_t ("slow drop-off"). The fast drop-off and slow drop-off time series were initialized with 5 cases on each of three days preceding the beginning of simulation, while the fast

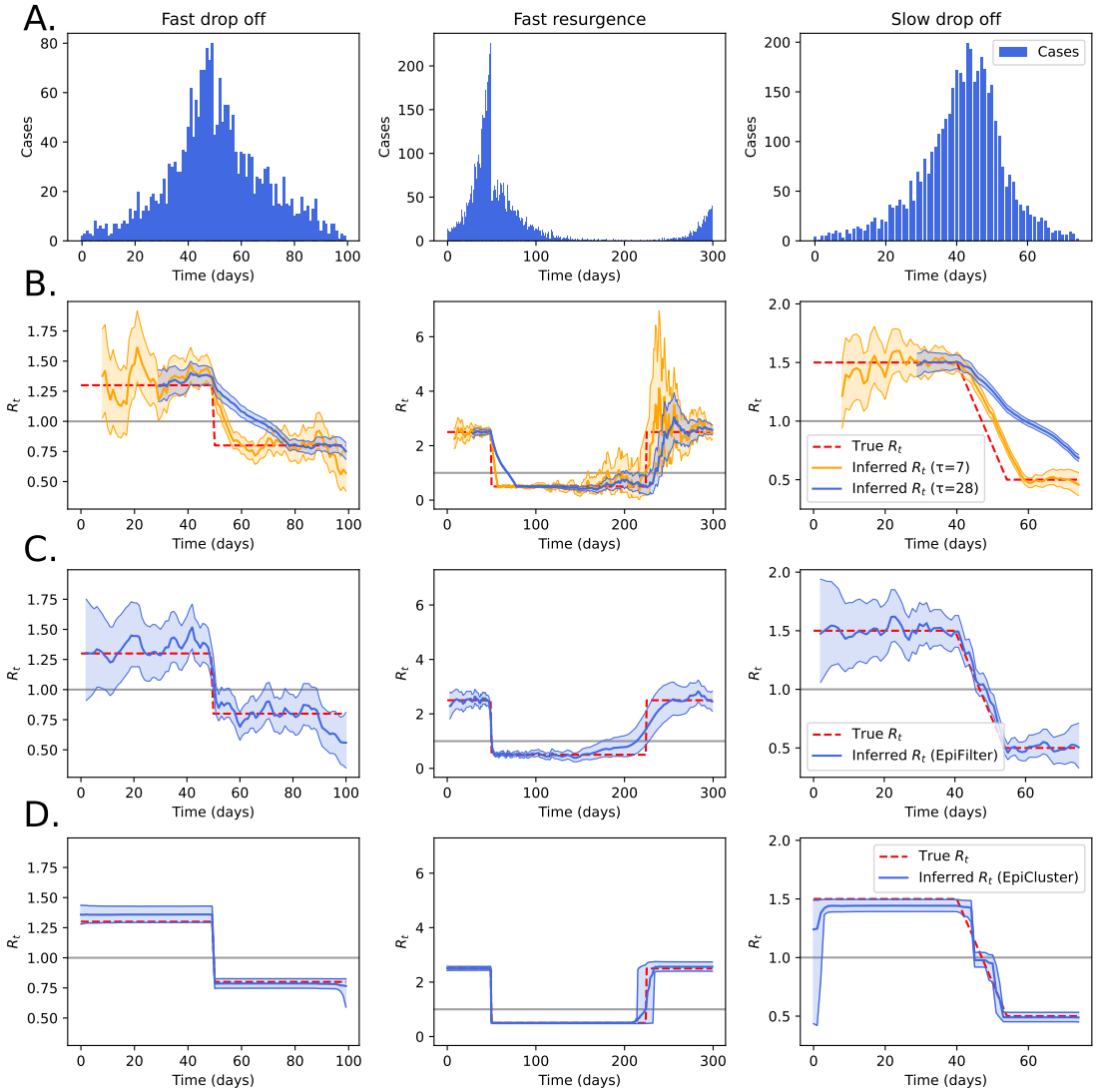


Figure 4.2: **Recovering synthetic R_t profiles in retrospective analyses.** We generated synthetic case data (panel A) using the Poisson renewal model (eq. (4.1)) with three prespecified profiles for R_t (dashed red lines in panels B / C / D). In panel B, we show the inferred R_t profile using a sliding window method [Thompson et al., 2019] for two different choices of the sliding window size ($\tau = 7$ and 28 days). In panel C, we show the inferred R_t profile using the EpiFilter method [Parag, 2021]. In panel D, we show the inference results when using EpiCluster to recover R_t. The hyperparameters θ and σ were set as described in §4.2.3. In panels B, C and D, shaded regions indicate the central 90% of the posterior distribution of R_t, while the central line indicates the posterior mean, and the background gray line indicates $R_t = 1$.

resurgence was initialized with 5 cases on each of fifty days preceding the beginning of simulation. Simulations for fast drop-off and slow drop-off used the COVID-19 serial

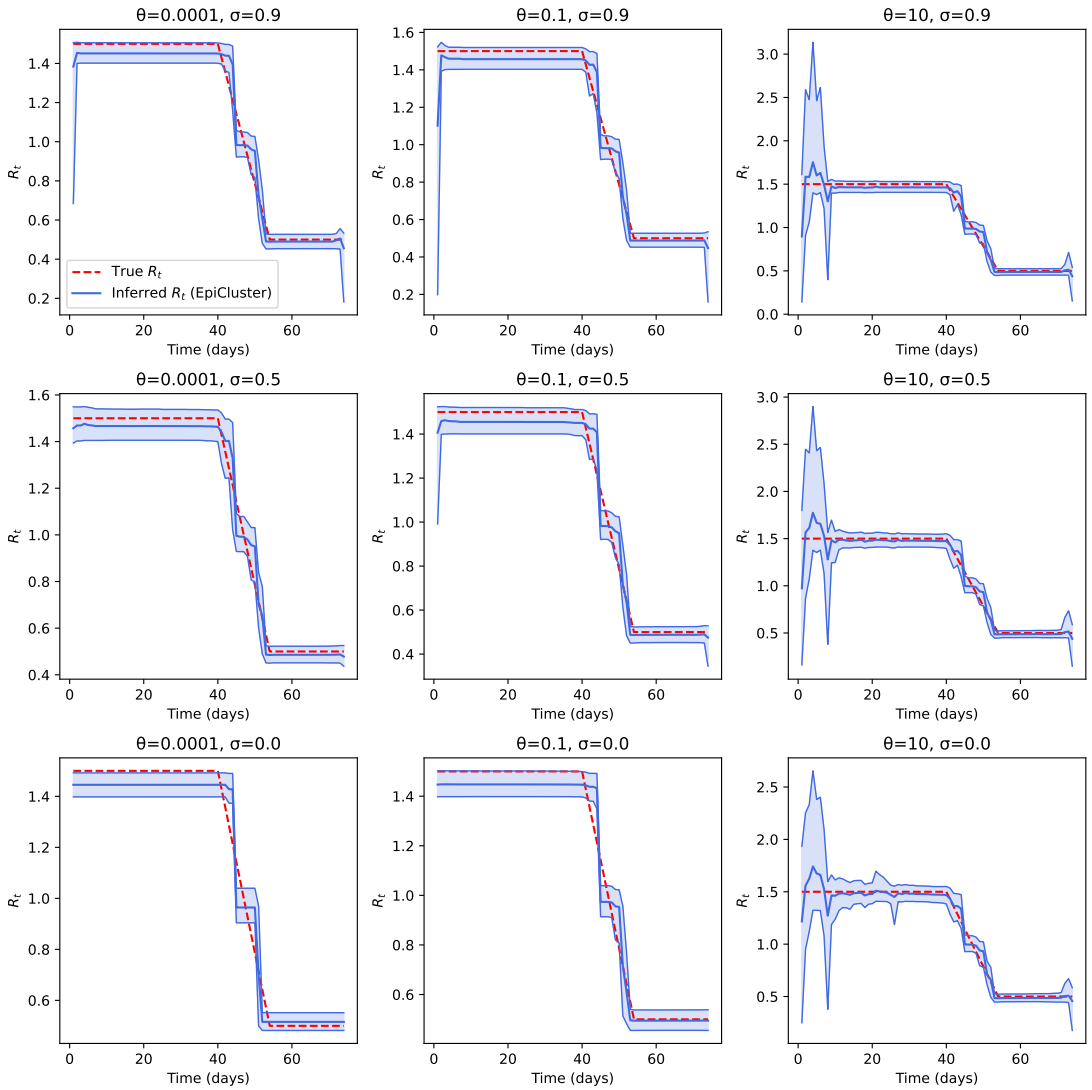


Figure 4.3: **Effect of θ and σ on inference for R_t.** We used the same slow drop-off synthetic data from Figure 4.2, and performed inference for R_t using the indicated fixed values of θ and σ, the two hyperparameters of the Pitman-Yor process (see eq. (4.4)). In all panels, shaded regions indicate the central 90% of the posterior distribution of R_t, while the central line indicates the posterior mean.

interval [Nishiura et al., 2020], while the fast resurgence used the Ebola serial interval as estimated for the 2014 West African Outbreak [Van Kerkhove et al., 2015].

In Figure 4.2, we compare R_t estimates from our method with those from two comparator methods: the sliding window method [Thompson et al., 2019] with two different choices of the sliding window width (7 days and 28 days), and the EpiFilter method [Parag, 2021].

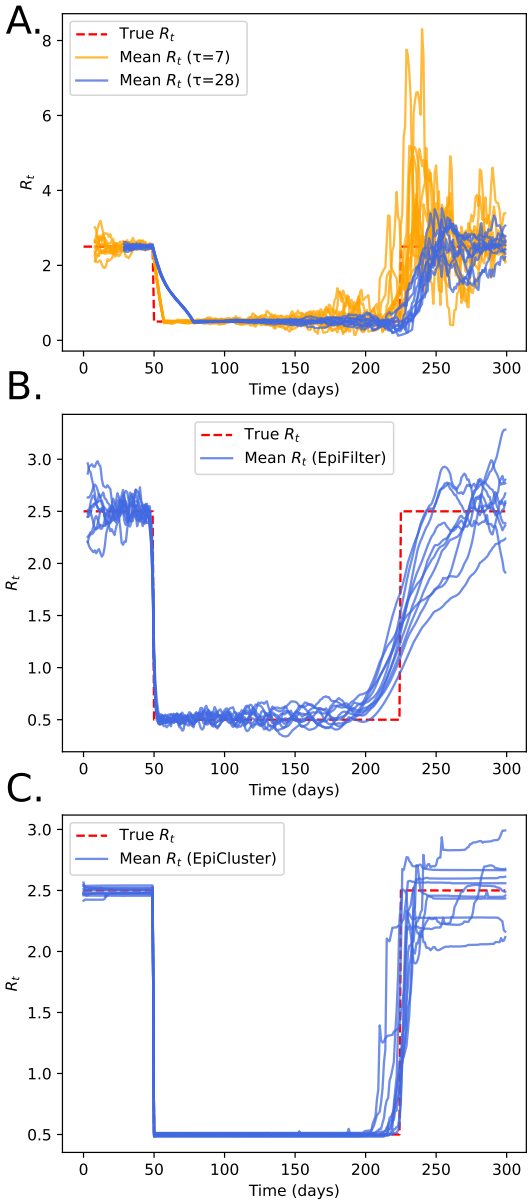


Figure 4.4: **Posterior mean estimates of R_t in the fast resurgence synthetic example.** Using the fast resurgence R_t profile (Fig. 4.2), we repeated the generation of synthetic data 10 times and performed inference for R_t for each synthetic dataset. Panel A shows the posterior means according to the sliding window method [Thompson et al., 2019] for two different choices of the sliding window size ($\tau = 7$ and 28 days). In panel B, we show the inferred mean R_t profiles using the EpiFilter method [Parag, 2021]. In panel C, we show the inferred means using EpiCluster. The hyperparameters θ and σ were set as described in §4.2.3.

Across the three R_t profiles considered, the estimates from the sliding window method lag behind the true values (Fig. 4.2B), since the windows are inherently backward-

looking—the longer the window width, the longer the moving average and the slower it is to respond to changes in R_t; the estimates are also very variable. The EpiFilter method fares better and is able to reliably infer downward shifts in R_t (Fig. 4.2C), corresponding to suppression; this method overly smooths over the upward tick in transmission in the fast-resurgence example. Our method performs favourably in the two "fast" examples (Fig. 4.2D). Like the EpiFilter method, our approach is less able to infer resurgence than suppression [Parag and Donnelly, 2022]. In the slow example, our piecewise-constant method approximates the linear decline in R_t with a staircase-like profile, which is better estimated by EpiFilter. In Figure 4.3, we show the effect of changing the hyperparameters of our method on inference for the slow drop-off example on the number and location of the regimes which are learned. As the two hyperparameters, θ and σ increase, more weight is given to a partitioning consisting of more regimes (see also Fig. 4.1), and the staircase steps become finer.

To account for stochastic variation in the synthetic data generation, we repeated inference for the fast resurgence example 10 times (Fig. 4.4). For the three methods, the posterior means are qualitatively similar across all runs, suggesting that these results are consistent across different realisations of the renewal process. The results in this figure indicate that the strong performance achieved by EpiCluster in Figure 4.2 (Fast resurgence), where the increase in R_t is detected, is consistent across realisations of the renewal process. We further investigated these inference results in Figure 4.5. At $t = 200$, before the increase in R_t, the inference results are consistent between replicates of the synthetic data generation and EpiFilter infers a precise and accurate value of R_t around 0.5 for all synthetic datasets. At $t = 250$, EpiFilter consistently learns that $R_t > 1$ for all synthetic datasets, although there is significant variability from replicate to replicate in the inferred posterior variance. This is unsurprising given that different replicates correspond to different incidence series, which vary in how much information they provide about the resurgence in R_t. Finally, at $t = 295$, EpiCluster consistently learns posteriors of R_t which are near the true value and place little probability mass on values $R_t < 1$ for all synthetic data replicates.

In the fast drop off and fast resurgence examples, EpiCluster estimates R_t with low bias and high precision. This is because the R_t profiles in the simulated examples align well with the assumptions made in our modelling: namely, that the R_t profile is piecewise-constant. We now consider R_t profiles with notable deviations from this assumption. In Figure 4.6, we compare the same methods on both noisy (left and middle columns) and oscillatory R_t profiles. When the magnitude of the noise is low (left

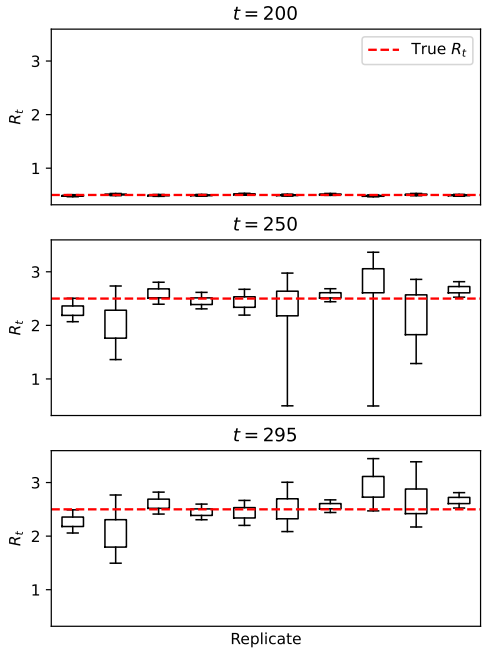


Figure 4.5: **Posterior distributions of R_t in the fast resurgence synthetic example at selected time points.** Using the fast resurgence R_t profile (Fig. 4.2), we repeated the generation of synthetic data 10 times and performed inference for R_t for each synthetic dataset. In each panel, we show the inferred posterior distribution for R_t using EpiCluster at the indicated time point. One box indicates the inferred posterior for each replicate of the synthetic data generation, with the box indicating the central 50% of the posterior and the lines spanning the central 90% of the posterior. The dotted red lines indicate the true value of R_t at that time point. The hyperparameters θ and σ were set as described in §4.2.3.

column), the results mirror those from the previous example. When the noise level increases (middle column), all methods are late to predict the precipitous decline in R_t, and EpiFilter provides a better quantification of uncertainty than the nonparameteric model. For the sinusoid example (right column), EpiFilter performs best, since the assumptions underpinning that method—that R_t follows a random walk—are closer to the reality of the generated data.

To evaluate the comparative inference performance of the methods quantitatively, for each R_t profile studied in Figures 4.2 and 4.6, we repeated the generation of synthetic data 100 times and studied the distributions of mean squared error (MSE) between the inferred posterior mean of R_t and the true R_t profile for each method. These distribu-

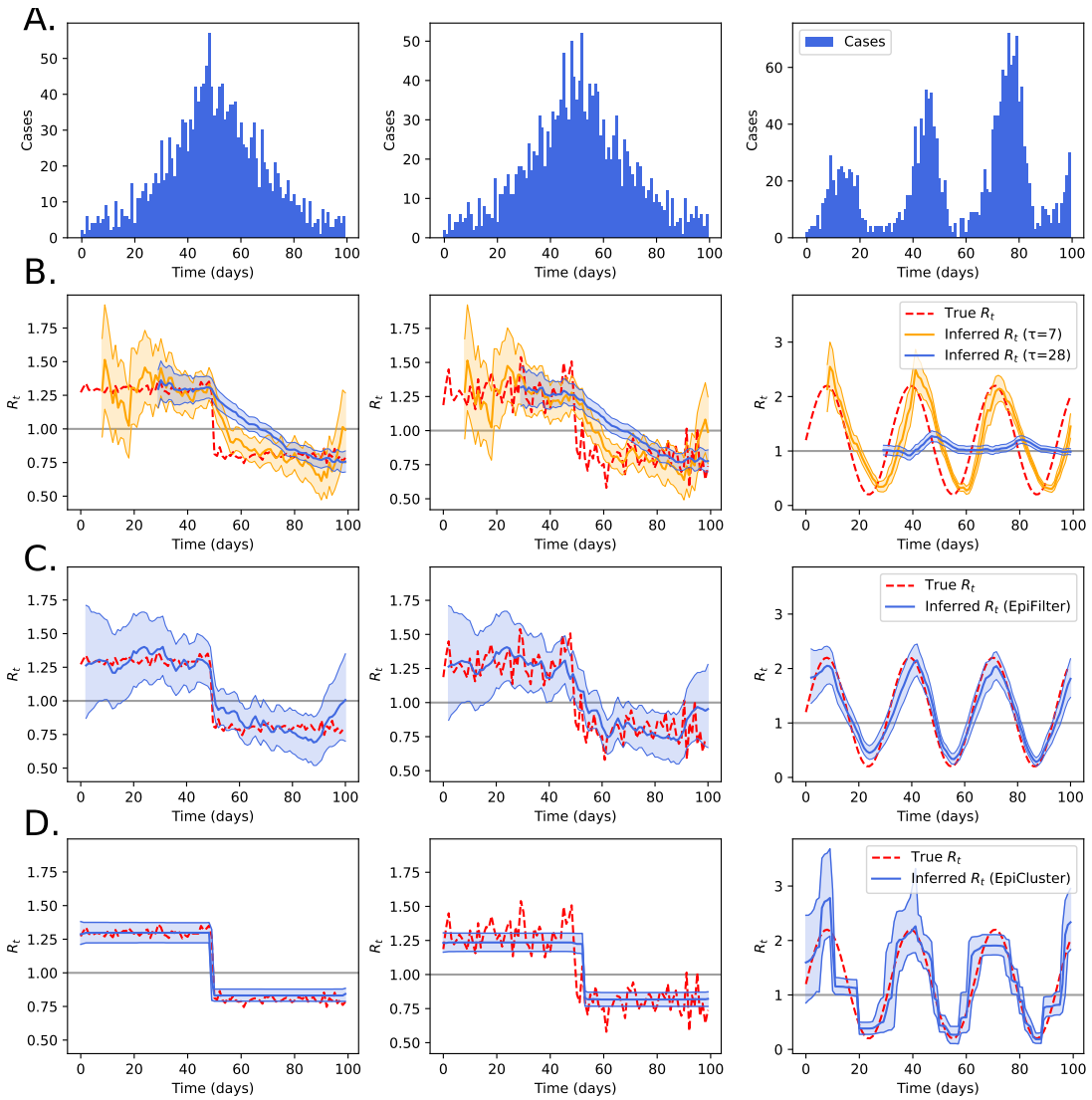


Figure 4.6: **Recovering noisy and oscillatory R_t profiles in retrospective analyses.** We generated synthetic case data (panel A) using the Poisson renewal model with three prespecified profiles for R_t (dashed red lines in panels B / C / D). The R_t profiles were calculated using step functions with additive IID Gaussian noise of standard deviation 0.025 (left) and 0.1 (middle). In the right column, we show results when R_t follows a sine wave. In panel B, we show the inferred R_t profile using a sliding window method [Thompson et al., 2019] for two different choices of the sliding window size ($\tau = 7$ and 28 days). In panel C, we show the inferred R_t profile using the EpiFilter method [Parag, 2021]. In panel D, we show the inference results when using EpiCluster to recover R_t. The hyperparameters θ and σ were set as described in §4.2.3. In panels B, C and D, shaded regions indicate the central 90% of the posterior distribution of R_t, while the central line indicates the posterior mean, and the background gray line indicates $R_t = 1$.

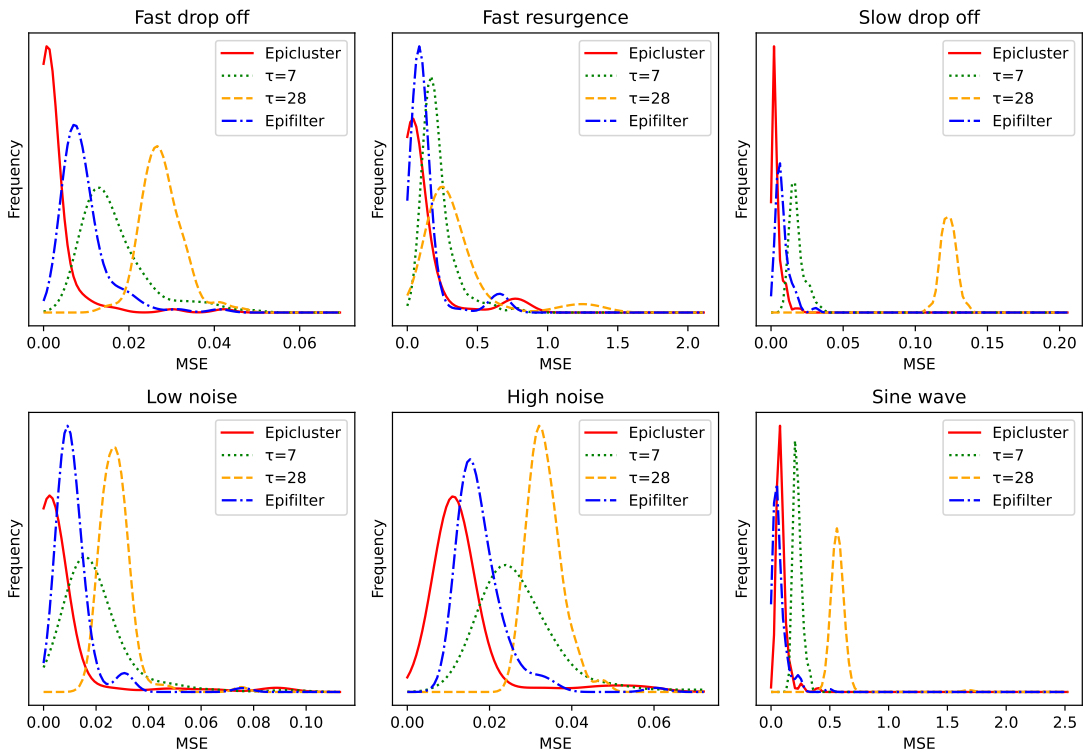


Figure 4.7: **Quantifying inference performance for R_t on synthetic data examples.** For each synthetic R_t profile in Figure 4.2 (top row) and Figure 4.6 (bottom row), we generated incidence data and performed inference 100 times. We then calculated the mean squared error (MSE) between the posterior mean and the true R_t profile for each inference method. The distributions (estimated via kernel density estimation) of MSE values for each dataset and method are shown.

tions of MSE values, estimated via kernel density estimation, are shown in Figure 4.7. For the majority of the examples, EpiCluster tended to produce R_t estimates with the lowest MSE values followed by EpiFilter, with the sliding window methods performing worse. On the sinusoid example (Figure 4.6, right column), EpiFilter achieves lower MSE values than EpiCluster, presumably because the changes in R_t were more gradual in this case.

EpiCluster is effective at detecting sharp changes in transmission in real-time

The results thus far have considered retrospective analysis of outbreaks; these analyses are important for understanding the timing and impact of interventions following their imposition (e.g. [Flaxman et al., 2020, Brauner et al., 2021]). But, in unfolding epidemics of novels pathogens, it is crucial to know in as close to real time as data allows whether

transmission changes rapidly either after an intervention is instituted or after it is discontinued. In this section, we compare how the three R_t estimation methods fared in inferring an epidemic resurgence in real-time: as new case data becomes available subsequent to a jump upwards in transmission. We used the same fast resurgence data as in Fig. 4.2 and fit each method for a series of datasets of different lengths. Each of these datasets began at the same point (at $t = 0$); the datasets ended at different points. The endpoints ranged from 5 days to 35 days post-resurgence with gaps of 5 days between them.

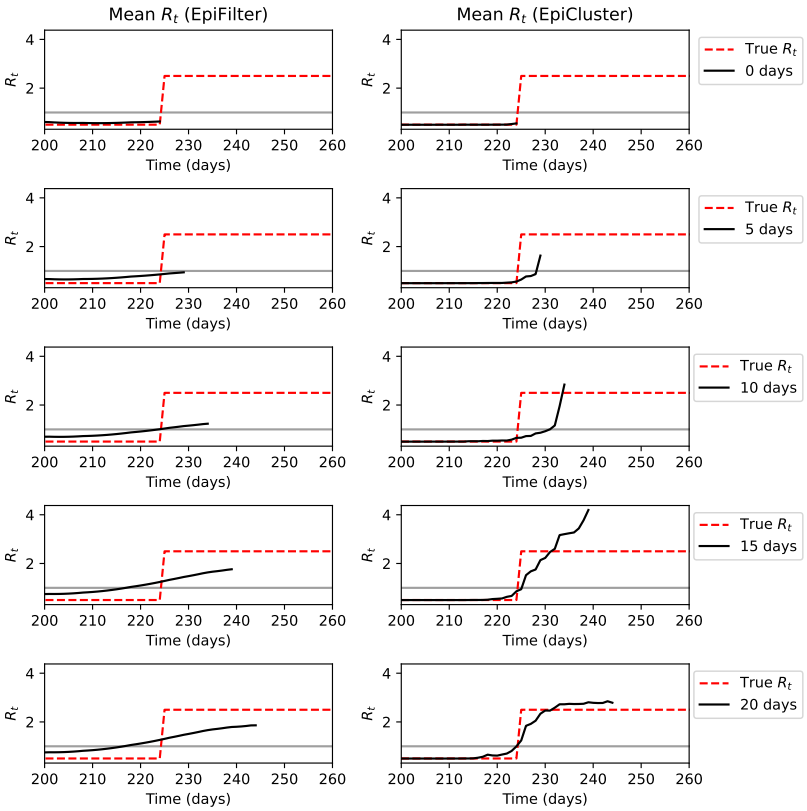


Figure 4.8: **Real-time estimation of a resurgence in R_t.** We used the same fast resurgence synthetic data from Figure 4.2 and performed inference for R_t based only on the time series up till the number of days after the resurgence indicated in the legend. In the left panel, we show the mean inferred R_t profile using the EpiFilter method [Parag, 2021]. In the right panel, we show the results when using EpiCluster to recover the mean of R_t. The hyperparameters θ and σ were set as described in §4.2.3. The background gray line indicates $R_t = 1$.

The posterior means of the inferred R_t series are shown in Fig. 4.8, while the full posteriors are shown in Fig. 4.13. The results illustrate that all three methods needed considerable data post resurgence to infer changes in transmission. For each series,

EpiCluster generally fared best in inferring the timing and magnitude of resurgence, with the posterior uncertainty interval reliably including the true R_t profile.

Data generating processes with greater variability pose issues for all methods and EpiFilter generally performs best

Variation in transmissibility across different individuals within a population can lead to greater variation in cases than is accounted for by a Poisson renewal model, and different pathogens vary in their degree of overdispersion [Lloyd-Smith et al., 2005]: SARS, for example, is prone to many superspreading events [Shen et al., 2004]; whereas pneumonic plague exhibits less variation in offspring cases [Lloyd-Smith et al., 2005].

To study the robustness of EpiCluster under more variable data generating processes, we generated data using the fast drop-off R_t profile and a negative binomial (NB) renewal model with inverse-dispersion parameter $\kappa > 0$: as $\kappa \to \infty$, the NB model approaches the Poisson, so low values of κ therefore correspond to more overdispersed data. Using the fast drop-off R_t profile, we generated case data under different values of κ, and, for each series, we fit the sliding window, EpiFilter and EpiCluster methods.

The results are shown in Fig. 4.9. When κ is large (i.e., the data are effectively generated from a Poisson distribution), the results match those observed in Fig. 4.6. As the data generating process exhibits more variation, all methods perform worse: generally failing to correctly identify the change in R_t and inferring a highly noisy R_t profile with many spurious fluctuations. However, the sliding window and EpiFilter methods generally produced more robust estimates in the presence of strong overdispersion.

EpiCluster estimates sharp changes in R_t for real COVID-19 incidence series

Next, we performed retrospective inference of R_t for the early COVID-19 outbreaks in three selected regions: Victoria and Queensland, Australia, and Hong Kong (these are the same datasets which were analysed in Chapter 3; see §3.2.5 and §4.2.9), which were selected for the variety of transmission profiles they encompass. The R_t estimates for these regions are shown in Figure 4.10 again comparing the sliding window approach (panel B) with the EpiFilter approach (panel C) and EpiCluster (panel D).

The first case of COVID-19 in Australia was reported in Victoria state on 25th January 2020 [Storen and Corrigan, 2020]. Subsequently, Victoria quickly became a hub of transmission and declared a state of emergency on 16th March, including a ban on non-essential gatherings of over 500 people [Storen and Corrigan, 2020]. On 18th March, more restrictions on movement followed with indoor public gatherings

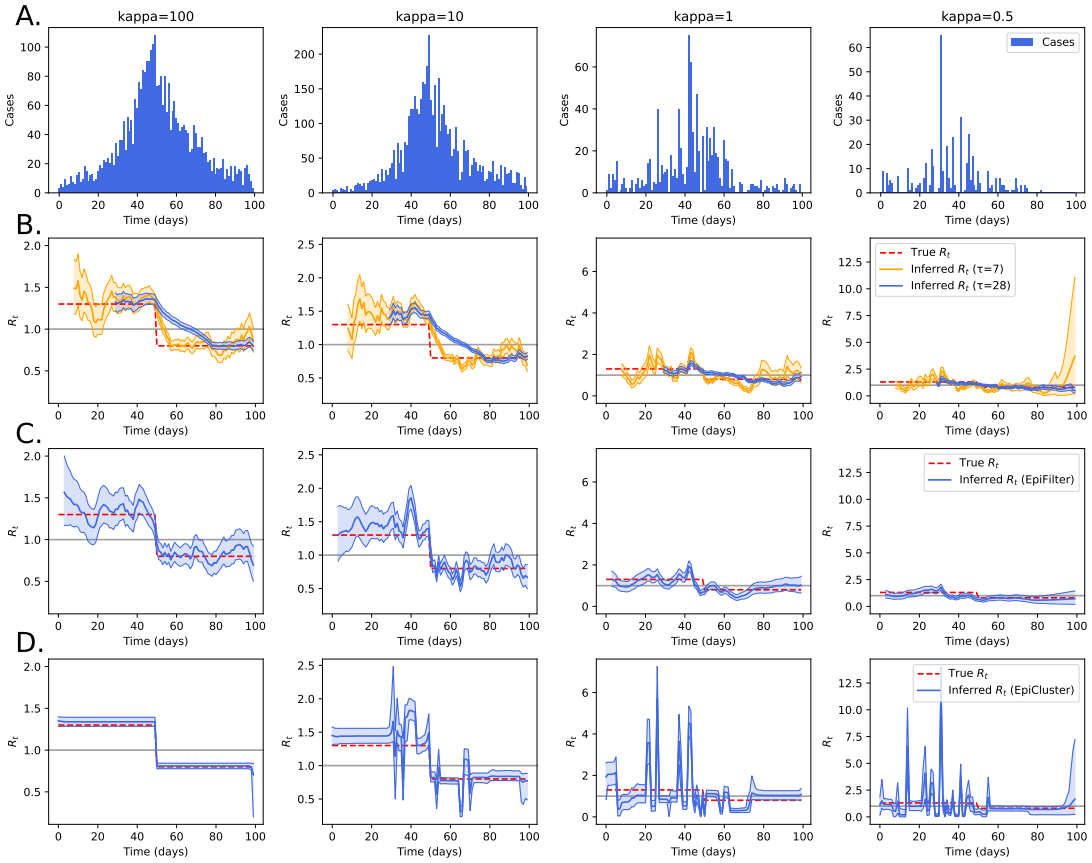


Figure 4.9: **Effect of negative binomial noise on posterior estimates of R_t.** We used the same fast drop-off R_t profile from Figure 2 but generated data according to a negative binomial renewal process with the inverse overdispersion (kappa) indicated at the top of each column. In panel B, we show the inferred R_t profile using a sliding window method [Thompson et al., 2019] for two different choices of the sliding window size ($\tau = 7$ and 28 days). In panel C, we show the inferred R_t profile using the EpiFilter method [Parag, 2021]. In panel D, we show the inference results when using EpiCluster to recover R_t. The hyperparameters θ and σ were set as described in §4.2.3. In panels B, C and D, shaded regions indicate the central 90% of the posterior distribution of R_t, while the central line indicates the posterior mean, and the background gray line indicates $R_t = 1$.

of more than 100 people banned and restrictions in aged care facilities introduced across Australia [Storen and Corrigan, 2020]. On the 22nd March, the state Premier announced that Victoria would implement a shutdown of all non-essential activity across the state [Storen and Corrigan, 2020]. The sliding window approach (Fig. 4.10B) and EpiFilter (Fig. 4.10C) both estimated declines in transmission starting around 22nd March; EpiCluster infers a sharper decline around 25th March. All methods inferred that transmission subsequently remained below the level for sustained transmission,

apart from an uptick in transmission estimated from EpiCluster coinciding with a burst of cases around 10th April, which likely reflects a violation of the assumptions of the model (the burst of cases causing this spike in the inferred R_t value only lasts one day, and thus may by occurring due to noise in the reported data rather than a genuine change in R_t).

The first case of COVID-19 in Queensland, Australia occurred on 29th January 2020 [Storen and Corrigan, 2020], and the first wave began in early March. All three estimation methods inferred that, since imported cases were the dominant cause of the wave, there was relatively low community transmission, and the bulk of local R_t estimates were below 1 (Fig. 4.10). All methods inferred a decline in transmission beginning around the 16th March—the date when Victoria declared a state of emergency, and Australia introduced a self-isolation requirement for all international arrivals—and EpiCluster estimated a rapid decline on 17th March. To combat the insurgence of imported cases, the Queensland Premier announced that the state would restrict access to the border on 24th March: this included termination of all rail services and border road closures [Storen and Corrigan, 2020], and EpiCluster inferred a small decline occurring on this date.

Hong Kong, like Singapore and Taiwan, was quick to act on learning of the outbreak of COVID-19 in Wuhan, China, and the government enacted intensive surveillance campaigns and declared a state of emergency on 25th January, 2020 [Cowling et al., 2020, Fig. 1]. On the 7th February 2020, Hong Kong introduced prison sentences for anyone breaching quarantine rules [OT&P Healthcare, 2022]. This date broadly coincides with the decline in R_t detected across all three methods, and the decline detected by EpiCluster is especially rapid.

Hong Kong's second wave of COVID-19 began in March 2020 driven by imported cases from North America and Europe [Parag et al., 2021], and all three methods detect an increase in the local R_t shortly after 15th March. Policy responses to this wave by the Hong Kong government included a quarantine requirement on international arrivals (effective 19th March; [Xinhua News Agency, 2020]), a ban on foreign travellers (effective 25th March; [OT&P Healthcare, 2022]), and a ban on gatherings of more than four people (effective 27th March; [OT&P Healthcare, 2022]); a significant decrease in R_t is detected by all three methods around the times when these interventions were imposed. The EpiCluster results mirror the timing of this intervention most closely, suggesting that there was a short time lag between when the interventions were imposed and their effect.

To explore the sensitivity of our estimates for Hong Kong to the hyperparameters of the method, we performed a series of sensitivity analyses where these parameters were fixed at different values and inference was performed (Fig. 4.12). These experiments illustrate that, as either of the hyperparameters are increased, the R_t profile comprises a greater number of regimes, and there is greater uncertainty in the R_t estimates. The qualitative behaviour of the majority of estimates, however, remains the same, with a large decline in transmission around 7th February 2020 and a resurgence in mid March.

EpiCluster estimates sharp changes in R_t for other disease outbreaks

To study the applicability of EpiCluster to infectious diseases other than COVID-19, we applied the method to several outbreaks: the 1972 Smallpox outbreak in Yugoslavia, the 1861 Measles outbreak in Hagelloch, Germany, and the 2003 SARS outbreak; these datasets were obtained from the EpiEstim package [Cori et al., 2013]. These results are shown in Figure 4.11 and show R_t estimates excluding an initial period when cases are low, since these early data are more likely to be unreliable due to data limitations. In all three outbreaks, EpiFilter learns the smoothest R_t profile, while EpiCluster infers a more jagged R_t series with high uncertainty and larger, rapid fluctuations in the value of the effective reproduction number. For SARS, the large variation in R_t inferred by EpiCluster is almost certainly due to the model's assumptions being violated, and we return to this point in the discussion. For these outbreaks, the sliding window method [Thompson et al., 2019] learns an R_t which resembles the results from EpiCluster but typically with smaller and smoother fluctuations in the value of R_t.

4.4 Discussion

EpiCluster assumes that R_t is piecewise constant, and presents a flexible approach for learning the locations and duration of each constant regime of R_t. When R_t does truly change rapidly (Figure 4.7), we observed that EpiCluster could outperform existing methods in accurately recovering R_t from incidence data. Thus, the method might be particularly appropriate to apply in situations where disease transmission is believed to have suffered some rapid change in time (for example, the immediate institution of a non-pharmaceutical intervention such as a lockdown or social distancing). In such situations, EpiCluster provides a principled, data-driven approach to evaluating whether and when the intervention affected transmission. However, when rates of disease transmission change more gradually over time (for example, if non-pharmaceutical interventions

intended to control spread of the disease are relaxed gradually), other methods which can directly learn gradual changes in R_t (such as EpiFilter) may be more appropriate.

It is likely that some of the rapid changes in R_t detected by EpiCluster on the COVID-19 time series (Fig. 4.10) reflect violations of the model assumptions, rather than genuine changes in the transmission of the disease, and further work is required to adapt the framework presented in this chapter to handle more complex processes. EpiCluster relies on the Poisson distribution (eq. (4.1)) for its stochastic renewal model; this distribution does not accurately account for the fact that many infectious diseases—including smallpox, measles, and SARS (which we analysed in Fig. 4.11)—exhibit significant variation in transmissibility from person to person [Lloyd-Smith et al., 2005].

These factors would be more accurately captured by an overdispersed renewal model, e.g., the negative binomial distribution [Lloyd-Smith et al., 2005]. If we incorporated such a distribution into the renewal model underlying EpiCluster, we anticipate that the method may be able to infer R_t more robustly for diseases characterized by heterogenous transmission or superspreading, with fewer spurious clusters and dramatic changes in R_t value.

Unfortunately, an appropriate conjugate prior does not exist for the negative binomial distribution. The MCMC sampler which we derive in §4.2 could not be applied in this situation, and inference would be slower and more challenging. Further methodological work on non-conjugate inference is needed.

Our approach additionally assumes that the incidence data are perfect, when in fact reporting biases are present [Gostic et al., 2020, Pitzer et al., 2021]. Our approach could be modified to correct R_t estimates for reporting biases [Gostic et al., 2020]; in the absence of these corrections, we anticipate that the changepoints we have inferred for R_t on real data may be biased by several days.

Bayesian nonparametric methods such as EpiCluster are attractive for their ability to allow model complexity to scale with the volume and complexity of the data, and this principle could enable wider applications of Bayesian nonparametrics throughout epidemiology, representing an advance upon any methods which require the number of parameters used to represent the underlying process to be determined in an ad hoc way.

However, nonparametric methods such as EpiCluster depend on process hyperparameters which do require user tuning. (This situation is not dissimilar to the sliding window width hyperparameter which we used to perform inference in Chapter 3.) In this chapter, we proposed a heuristic strategy for tuning the two process hyperparameters of EpiCluster (θ and σ) as a function of the number of time points such that *a priori*

few changepoints in R_t were expected; this assumption causes the method to prevent overfitting. Our sensitivity analyses (Figure 4.3 and Figure 4.12) show that EpiCluster produces sensible and compatible inference results for a range of hyperparameter values. Further work may help to develop a more flexible, hierarchical inference approach in which hyperpriors are placed on the process hyperparameters, and these are then themselves inferred from the data. An example of this approach has been developed for Dirichlet processes, in which a gamma prior is placed on the concentration hyperparameter of a Dirichlet process and then this hyperparameter is updated according to a Gibbs sampler [Escobar and West, 1995].

We developed EpiCluster within the stochastic renewal model framework (which we also used in Chapter 3). Despite the usefulness of such stochastic models for learning R_t, more complex models which enable more detailed specification of certain features of the population are also useful for epidemiological research (see §2.4). Thus, in the next chapter, we move away from the stochastic renewal model and begin our investigations of compartmental differential equation models of infectious disease outbreaks.

4.5 Data and software

The Python software implementation of the model is available at `https://github.com/SABS-R3-Epidemiology/epicluster`. All data and scripts use to generate the results are available at `https://github.com/SABS-R3-Epidemiology/epicluster-results`.

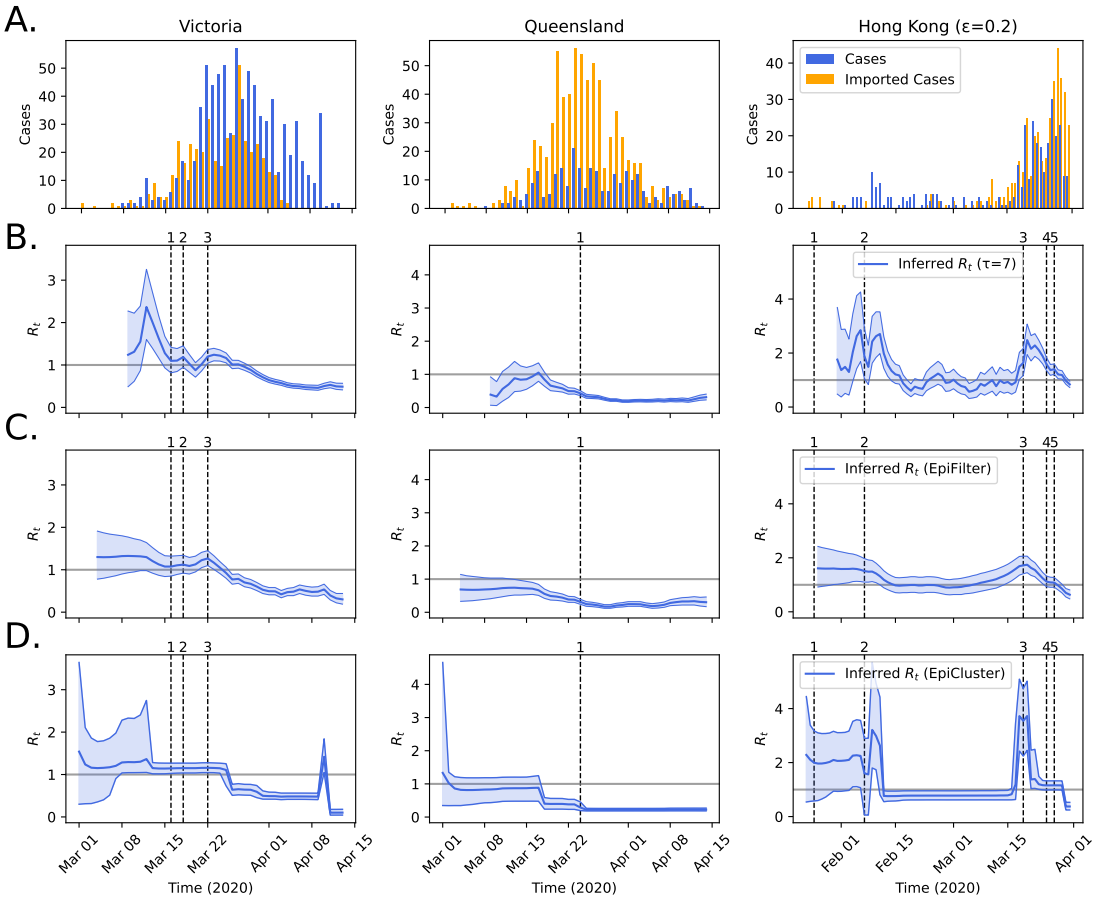


Figure 4.10: **Learning R_t from early COVID-19 epidemic incidence curves in three locations.** Data on local and imported cases from the early COVID-19 pandemic in three selected regions is shown in panel A. In panel B, we show the inferred R_t profile using a sliding window method [Thompson et al., 2019] for two different choices of the sliding window size ($\tau = 7$ and 28 days). In panel C, we show the inferred R_t profile using the EpiFilter method [Parag, 2021]. In panel D, we show the inference results when using EpiCluster to recover R_t. The hyperparameters θ and σ were set as described in §4.2.3. In panels B, C and D, shaded regions indicate the central 90% of the posterior distribution of R_t, while the central line indicates the posterior mean, and the background gray line indicates $R_t = 1$. Vertical dotted lines indicate policy-relevant dates. For Victoria: 1: ban on non-essential gatherings of over 500 people; 2: movement restrictions and ban on indoor gatherings of over 100 people; 3: shutdown of all non-essential activity. Queensland: 1: border restrictions and termination of rail services. Hong Kong: 1: state of emergency declared; 2: prison sentences introduced for those breaking quarantine; 3: compulsory quarantine of all arrivals; 4: ban on foreign travellers; 5: ban on gatherings over four people.

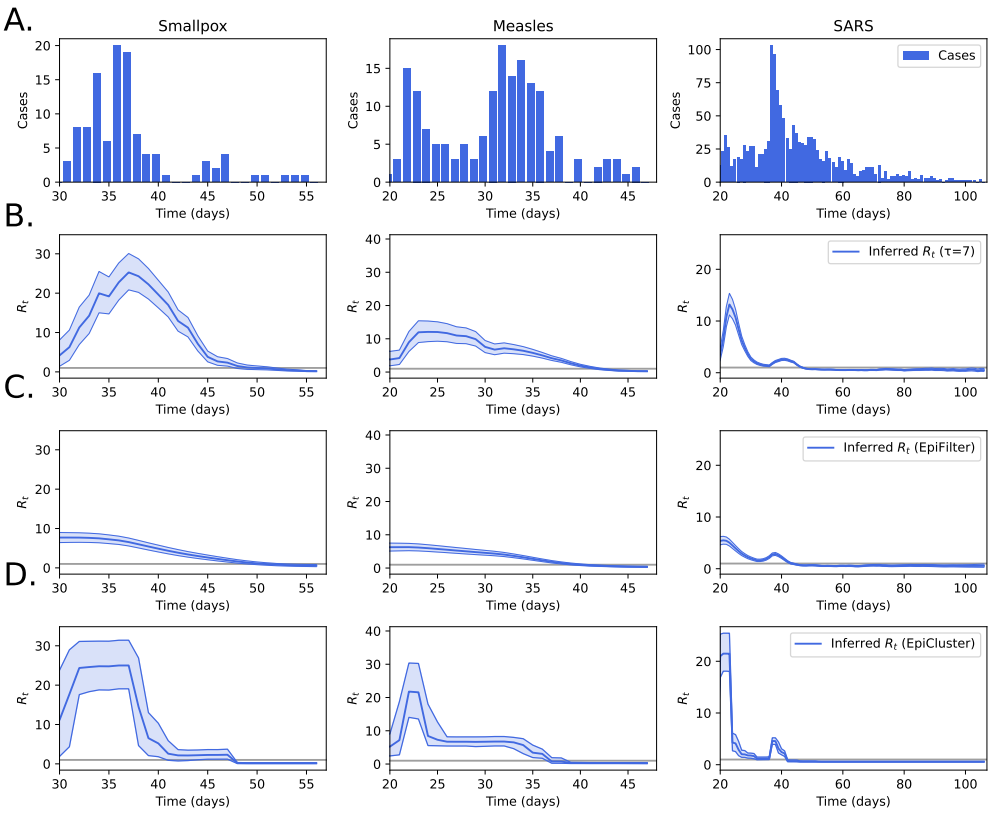


Figure 4.11: **Learning R_t profiles for non-COVID outbreaks.** Data on local cases for three selected outbreaks are shown in A. In panel B, we show the inferred R_t profile using a sliding window method [Thompson et al., 2019] for two different choices of the sliding window size ($\tau = 7$ and 28 days). In panel C, we show the inferred R_t profile using the EpiFilter method [Parag, 2021]. In panel D, we show the inference results when using EpiCluster to recover R_t. The hyperparameters θ and σ were set as described in §4.2.3. In panels B, C and D, shaded regions indicate the central 90% of the posterior distribution of R_t, while the central line indicates the posterior mean, and the background gray line indicates $R_t = 1$.

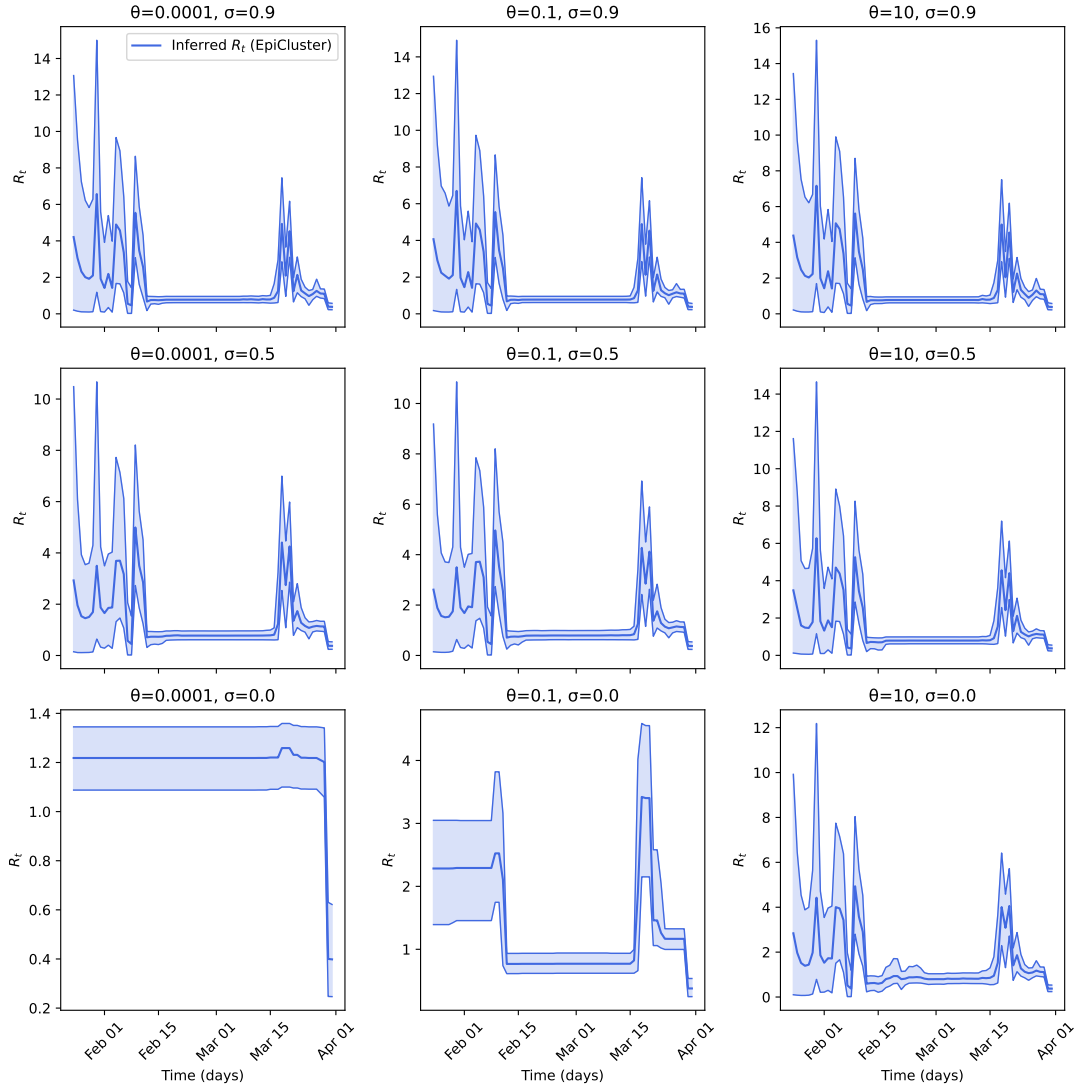


Figure 4.12: **Effect of θ and σ on inference for R_t for the Hong Kong COVID-19 dataset.** We used the Hong Kong data from Figure 4, and performed inference for R_t using the indicated fixed values of θ and σ, the two hyperparameters of the Pitman-Yor process (see eq. (4.4)). In all panels, shaded regions indicate the central 90% of the posterior distribution of R_t, while the central line indicates the posterior mean.

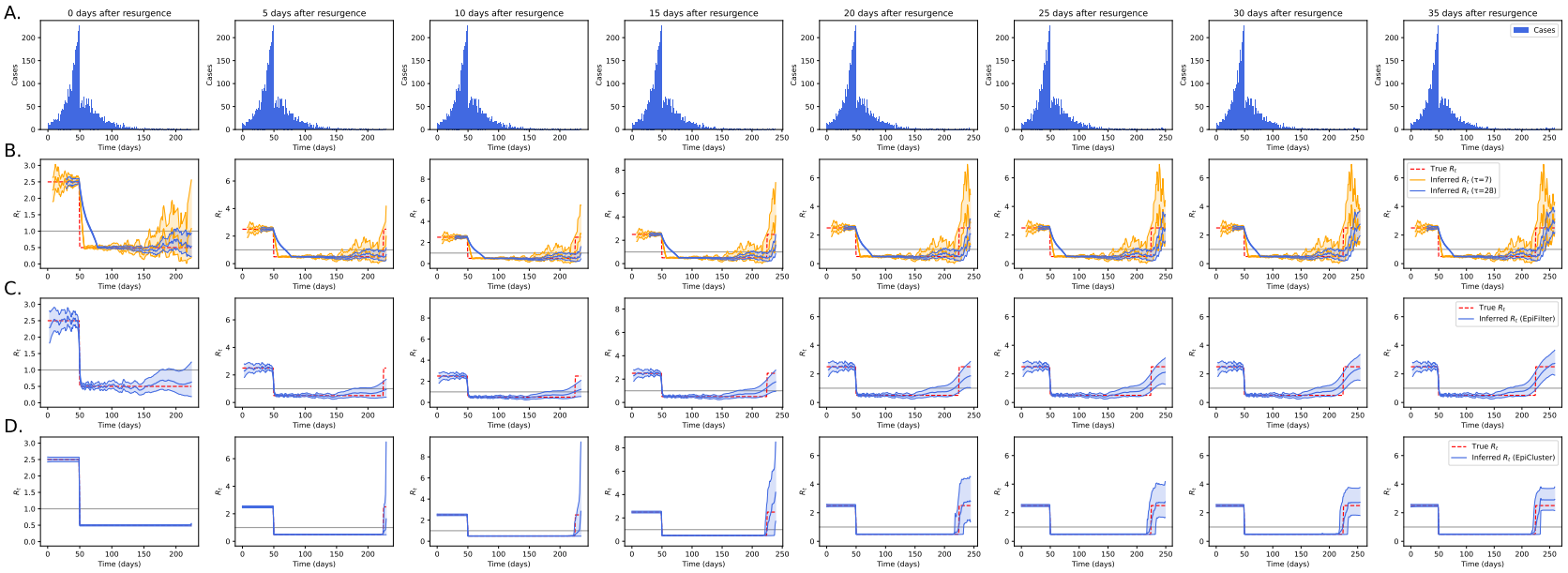


Figure 4.13: **Real-time estimation of a resurgence in R_t.** We used the same fast resurgence synthetic data from Figure 2, and performed inference for R_t based only on the data indicated at the top of each column. In panel B, we show the inferred R_t profile using a sliding window method [Thompson et al., 2019] for two different choices of the sliding window size ($\tau = 7$ and 28 days). In panel C, we show the inferred R_t profile using the EpiFilter method [Parag, 2021]. In panel D, we show the inference results when using EpiCluster to recover R_t. The hyperparameters θ and σ were set as described in §4.2.3. In panels B, C and D, shaded regions indicate the central 90% of the posterior distribution of R_t, while the central line indicates the posterior mean, and the background gray line indicates $R_t = 1$.

Chapter 5

Accounting for uncertain contact structure and the accuracy of numerical approximation in compartmental models

Overview

Alongside the stochastic renewal process models that we employed in Chapters 3 and 4, compartmental differential equation models such as SIR (see Example 8 in Chapter 2) are workhorses of epidemiological modelling. In this chapter, we analyse two significant forms of uncertainty in the simulated outputs of epidemiological compartmental models.

First, we consider compartmental models which include age structure; these models take as an input parameter a *contact matrix* specifying the rate of contact between each age group in the population. Although contact matrices may be set to fixed values, the finite survey data from which they are estimated may in fact correspond to significant uncertainty in the values of the contact matrix. In a synthetic study, we demonstrate that uncertainty in the values of the contact matrix may lead to significant discrepancy in forward simulations of the SEIR model.

Subsequently, we analyse another potentially significant source of uncertainty in the solution of compartmental models: insufficiently accurate numerical integration of the underlying differential equations. We show that insufficient solver step sizes can significantly bias the simulated number of cases and deaths, and in particular, we show that a 1-day solver step size is not sufficiently accurate for simulation of cases and deaths

at a daily resolution. Our results motivate the use of adaptive step size solvers for such models. We additionally show how interventions expressed as step functions can cause further inaccuracies in the simulation of compartmental models, and demonstrate that smoothing approximations can significantly increase the accuracy of model simulations when interventions are involved.

Motivation

One of the primary goals of the publication [van der Vegt et al., 2022] and the associated notebooks, from which this thesis chapter is adapted, was to provide a pedagogical resource for newcomers to the field of epidemiological modelling, teaching them some of the ways in which compartmental models can be used and some of the things that can go wrong with them. For this reason, the focus in this chapter is in exploring the modelling concepts, explaining potential pitfalls of these models, and setting up the work on inference for differential equations which appears later in the thesis, rather than building a highly realistic compartmental model of the transmission of COVID-19 or any other specific disease outbreak.

Publications

The contents of this chapter are derived from a portion of:

- S. A. van der Vegt,[†] L. Dai,[†] I. Bouros,[†] H. J. Farm,[†] **R. Creswell,[†]** O. Dimdore-Miles,[†] I. Cazimoglu, S. Bajaj, L. Hopkins, D. Seiferth, F. Cooper, C. L. Lei, D. Gavaghan, and B. Lambert: "Learning transmission dynamics modelling of COVID-19 using comomodels," *Mathematical Biosciences*, vol. 349 (2022). [van der Vegt et al., 2022]

  ([†]= joint first authorship.)

  Contributions: The research underlying this chapter was conducted as part of a collaboration between the Doctoral Training Center (DTC) at Oxford and the COVID-19 International Modelling Consortium (CoMo). The Como Consortium was formed in 2020 in response to the COVID-19 outbreak, and was engaged in the development and deployment of infectious disease modelling to inform COVID-19 policy, in collaboration with public health officials in 40 countries in Asia, Africa, North America, and South America. In October 2020, I volunteered

to join the organizing team for a collaboration between the CoMo Consortium and Oxford's DTC. Amongst other work, this collaboration developed an R software library and a series of R notebooks containing code, figures, and results illustrating the application of compartmental models to COVID-19 data. This collection of notebooks was published as part of a special issue in *Mathematical Biosciences* [van der Vegt et al., 2022].

This thesis chapter is derived from two of the notebooks, "The importance of uncertainty in age-specific contact patterns for quantifying COVID-19 risk," and "On the numerical solution of compartmental models." I was the primary author of these two notebooks and handled the data analysis, code sections, and interpretation of results contained within them. All authors of the paper made contributions and suggestions to the writing and revision of the notebooks, and some of these contributions are reflected in the wording of parts of this chapter.

5.1 Introduction

The Poisson renewal model of the incidence of an infectious disease, eq. (3.1), underlies our work in Chapters 3 and 4. As we have shown, this model is highly attractive for the immediate interpretability of its unknown parameter (R_t), the rapid inference enabled by a conjugate relationship between the Poisson model and the gamma prior on R_t, the ease of incorporating flexible models of time variation in R_t (e.g., sliding windows or EpiCluster), and requiring only incidence data and an estimate of the generation time distribution or serial interval distribution for fitting. For these reasons, the Poisson renewal model is a powerful tool for monitoring and analysing outbreaks of infectious disease.

However, eq. (3.1) suffers from several disadvantages. It assumes that the population is homogeneous and well-mixed, failing to account for the effects on transmission of, e.g., spatial or age structure. It does not model the fact that, for serious diseases, a significant proportion of cases may subsequently die. These limitations can be addressed using another modelling framework for infectious diseases: *compartmental models*, which we introduced in Example 8, Chapter 2. Compartmental models assume that each member of the population can be classified as belonging to one of a finite number of diseased (or non-diseased) states. Examples of typical states used in compartmental models include susceptible (describing individuals who can be infected with the disease, but are not currently), recovered (individuals who have recovered from the disease and may

have full or partial immunity), and infected (individuals who have the disease and may spread it to others). Model complexity can be increased as necessary for a particular modelling or inference task by adding more specialized disease states. For example, multiple infected states corresponding to different levels of transmissibility or severity, exposed states where individuals have been infected but are not yet infectious to others (representing the latent period of a disease), or an absorbing state representing death can be used. States can also be added for different age groups, spatial subsets, or other characteristics of the population. Given a set of states, compartmental models express the rates at which individuals move from one state to another, yielding a system of differential equations.

Compartmental models have been widely adopted for modelling the COVID-19 pandemic, e.g., [Birrell et al., 2021, Dehning et al., 2020], where they have enabled the specification of more complex and more realistic processes underlying the transmission of the virus. However, the additional flexibility of the compartmental modelling framework comes at the cost of simulation and inference potentially being more challenging than in the Poisson renewal model. Thus, in this chapter, we present an analysis of two particular problems inherent in the simulation of compartmental models, and we illustrate the importance of these problems by running compartmental model simulations. First, we study compartmental models which include age structure. Because the mortality of many diseases is highly variable between different age groups, age structured models are useful, but they require the specification of a *contact matrix* giving the rates of contact between the various age groups. Contact matrices are informed by survey data, e.g., [Prem et al., 2017], which may not be numerous enough to inform highly precise estimates of the contact matrix.

Compartmental models are defined in terms of ordinary differential equations (ODEs) which in general have no analytical solution. Thus, an essential step in the simulation of compartmental models is numerical approximation of the ODEs using an appropriate solver. Thus, we conclude this chapter by providing an analysis of the effects of inaccuracy in the ODE solver on the numbers of cases and deaths in a compartmental model. Later in this thesis (Chapters 6 and 7), we will study the effects of solver inaccuracy on model simulation and parameter inference in more detail and for a wide range of different models; the results in this chapter provide a preliminary motivation for the importance of ensuring solver accuracy.

5.2 Methods

In this section, we define the three compartmental models which will employ to run simulations of a synthetic outbreak of an infectious disease.

5.2.1 The SEIRD model

A straightforward and more realistic extension of SIR (Chapter 2, Example 8), the *SEIRD* model, adds an exposed state, representing individuals who have been exposed to the disease but are not yet infectious to others, and a death state, representing individuals who have died of the disease. This model is given by:

$$\frac{dS}{dt} = -\beta SI \tag{5.1}$$
$$\frac{dE}{dt} = \beta SI - \kappa E \tag{5.2}$$
$$\frac{dI}{dt} = \kappa E - (\gamma + \mu)I \tag{5.3}$$
$$\frac{dR}{dt} = \gamma I, \tag{5.4}$$
$$\frac{dD}{dt} = \mu I, \tag{5.5}$$

where $\beta > 0$ is the spreading rate of the disease, $\kappa > 0$ is the rate at which individuals move from exposed to infectious, $\gamma > 0$ is the recovery rate, and $\mu > 0$ is the death rate.

5.2.2 Adding age structure to the SEIRD model

Age is a major risk factor for more severe symptoms of many diseases, so accounting for the contact patterns of different age groups is particularly important for quantifying the risk that each age group faces. We extend the SEIRD model to include age structure by assuming that each member of the population belongs to one of a finite set of age

groups. The system of differential equations is:

$$\frac{dS_i}{dt} = -\beta S_i \sum_j C_{i,j} I_j \tag{5.6}$$

$$\frac{dE_i}{dt} = \beta S_i \sum_j C_{i,j} I_j - \kappa E_i \tag{5.7}$$

$$\frac{dI_i}{dt} = \kappa E_i - (\gamma_i + \mu_i) I_i \tag{5.8}$$

$$\frac{dR_i}{dt} = \gamma_i I, \tag{5.9}$$

$$\frac{dD_i}{dt} = \mu_i I, \tag{5.10}$$

where S_i indicates susceptibles belonging to the ith age group, and so forth. We allow age-structured values of γ_i and μ_i, because the rates of recovery and death can be highly age-dependent. $C_{i,j}$ is the contact matrix, discussed in detail in the next section §5.2.3.

This model provides no mechanism for individuals to move from one age group to another. Thus, it is appropriate for simulation over shorter time scales in which aging of the population is not expected to be a significant factor.

5.2.3 Contact matrix data

In the age-structured SEIRD model, differing rates of contact between age groups are modelled using the contact matrix, $C_{i,j} \geq 0$. Each element of $C_{i,j}$ is proportional to the expected number of daily contacts that individuals of age group i have with individuals of age group j.

We obtain country-specific estimates of the contact matrix in 152 countries from [Prem et al., 2017]. These contact matrices are constructed from survey data, in which participants in the study record their contacts throughout the day and include information on the age and location of each contact. For each country, a contact matrix is provided for four different locations where people may mix with others: at home, at school, at work, and elsewhere. For a contact matrix, C, each element, $C_{i,j}$, indicates the expected number of contacts someone from age group i has per day with people from age group j, which is given by:

$$C_{i,j} = \frac{\text{total \# contacts between } i \text{ and } j}{\text{size of group } i}. \tag{5.11}$$

Because $C_{j,i}$ has the size of group j as its denominator, typically $C_{i,j} \neq C_{j,i}$ due to demographic patterns, meaning contact matrices are not typically exactly symmetric.

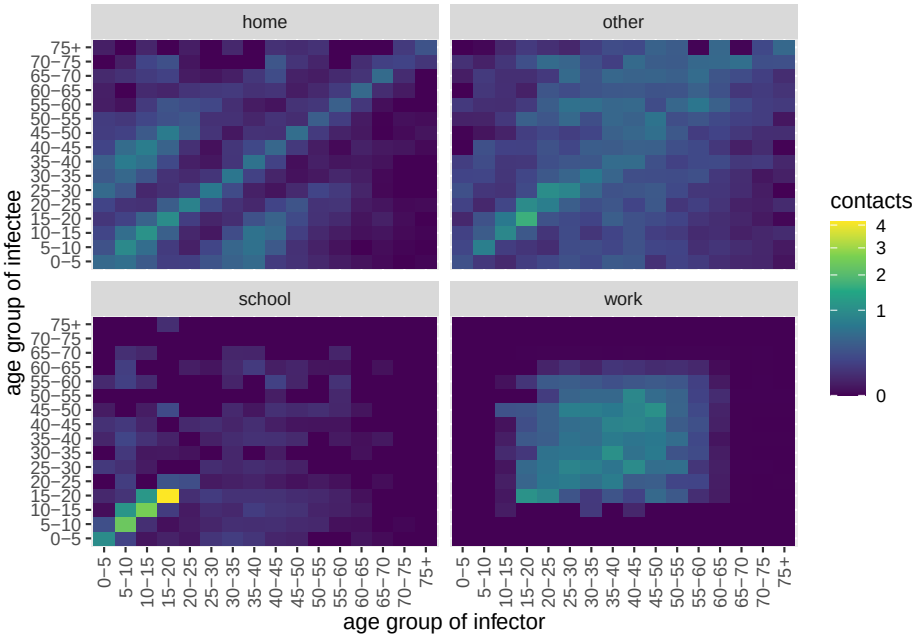


Figure 5.1: **Location-specific contact matrices for the United Kingdom (UK).** Contact matrix data for the UK published by [Prem et al., 2017], with contacts divided into four locations indicated at the top of each matrix: home, school, work, and other.

Since the age-demographics of a population affect its contact matrices, a contact matrix estimated for a given country should not be repurposed for another without due care [Arregui et al., 2018].

In the contact matrices provided by [Prem et al., 2017], the oldest age group is 75–80 year olds; in the remainder of this section, we assume that the contact patterns are the same for individuals aged 80+. This may be a strong assumption, since it neglects the change of circumstances that may occur for many in this age group.

The contact matrices for the United Kingdom are shown in Figure 5.1. These matrices illustrate rich contact patterns for the UK, which are markedly different between locations. At school, unsurprisingly, students mix with many others of similar ages. At home, there is considerable intergenerational mixing. At work, there is more uniform mixing—the vast majority between people of working age. These suggest a common transmission pattern, in which schoolchildren, who have the most daily contacts, infect one another. They then pass infection onto their parents at home, who then pass their infection onto work colleagues.

To use the location-specific contact matrix data in our model, eq. (5.6), we sum the

four location-specific contact matrices:

$$C_{i,j} = C_{i,j}^{\text{home}} + C_{i,j}^{\text{school}} + C_{i,j}^{\text{work}} + C_{i,j}^{\text{other}}.$$

5.3 The importance of uncertainty in age-specific contact patterns for compartmental model simulations

5.3.1 Bootstrapped sampling of the UK contact matrices

When performing forward simulations of the age-structured SEIRD model or when fitting these models to data, the contact matrix is typically provided as a fixed input. However, using point estimates for the contact matrix neglects the considerable uncertainty inherent in them. Thus, we now investigate the sensitivity of the outputs of the age-structured SEIRD model to the uncertainty in contact matrix estimates. To do so, we use bootstrapped samples of the contact matrix to represent this uncertainty. The bootstrap algorithm works by selecting a random sample (with replacement) of the survey respondents, and a contact matrix is then constructed on the basis of this sample of respondents. Across many such samples, the set of contact matrices provides a measure of uncertainty in the number of daily contacts across different age groups.

We obtain bootstrapped samples of the contact matrix using the `socialmixr` library which accesses the POLYMOD data [Funk, 2020, Mossong et al., 2017]. Uncertainty in the contact matrix, estimated via the bootstrap procedure, has previously been used as part of a study to compute age-structured estimates of the immunity levels needed to eliminate transmission of measles [Funk et al., 2019].

We base our analysis on the age-structured population of the UK and generate 200 bootstrapped contact matrix draws to represent its uncertainty. In Figure 5.2, we show the variation in within-age-group daily contacts: i.e., we plot the samples of the diagonal elements of the contact matrix. The plot shows that ages 5–20 (i.e., mostly school children) have the greatest variation in contacts. Most likely, this is because this age group has the most contacts.

5.3.2 The influence of contact matrix uncertainty on epidemic dynamics

To explore the sensitivity of model outputs to the elements of the contact matrix, we simulate the age-structured SEIRD model for one year for each bootstrapped sample of the contact matrix. We use the following fixed values for the other parameters of the

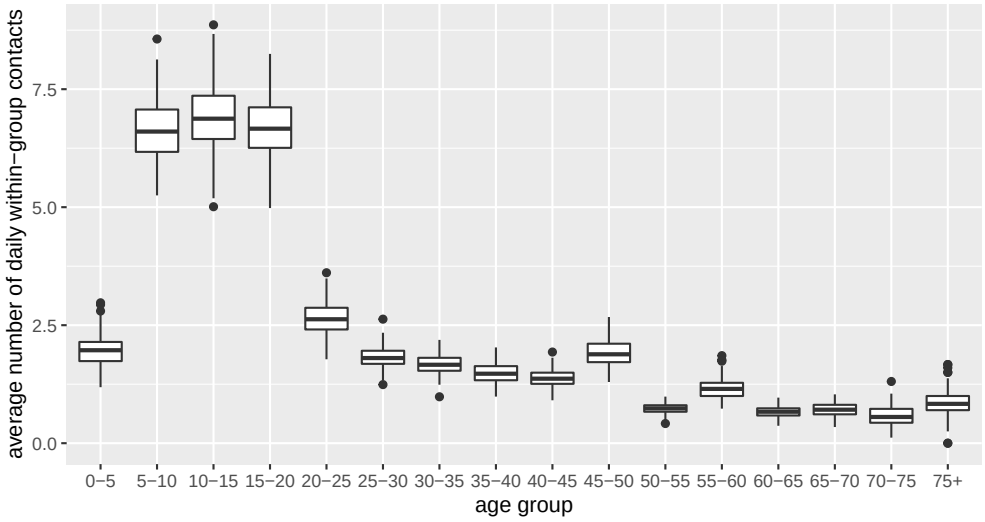


Figure 5.2: **Uncertainty in diagonal elements of the UK contact matrix.** For each age group, the distribution of the number of contacts members of that age group have with other members of their age group estimated from the bootstrap samples of the contact matrix is shown. Each boxplot indicates the median (central line) and the 25th and 75th percentiles (bottom and top of the box). The whiskers extend from the 25th percentile minus 1.5 times the interquartile range (IQR) to the 75th percentile plus 1.5 times the IQR; samples falling outside the whiskers are indicated by dots.

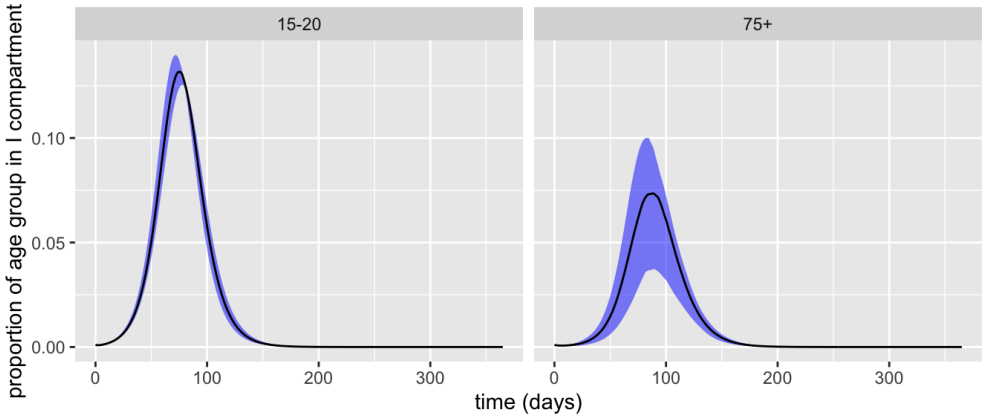


Figure 5.3: **Uncertainty in age-structured SEIR model simulations arising from uncertainty in the contact matrix.** (left) the size of the I compartment (as a proportion of the population size in that compartment) over time for the 15–20 age group, with the blue line indicating the median and the shaded region the central 90% of the simulations for each bootstrap sample of the contact matrix. (right) as left, but for the 75+ age group.

model [van der Vegt et al., 2022]: $\beta = 0.46, \kappa = \frac{1}{5.5}, \gamma_i = \frac{1}{7}(1 - \text{IFR}_i), \mu_i = \frac{1}{7}\text{IFR}_i$ where IFR_i is the infection fatality ratio for the ith age group as reported by [Verity et al., 2020] for COVID-19.

Because COVID-19 exhibits significant variation in mortality across age groups, its IFR values are a useful choice for studying the effect of contact matrix uncertainty. However, we note that due to the simplicity of our compartmental model, our simulation outputs are unlikely to resemble the actual course of the COVID-19 outbreak in the United Kingdom. See, e.g., [Birrell et al., 2021, Dehning et al., 2020], for more realistic compartmental models of the COVID-19 outbreak.

As an initial condition, we assume that 0.1% of the population is infectious, with the remainder of the population susceptible. No interventions or behavioural changes are included.

In Figure 5.3, we plot the central 90% quantiles computed from these simulations of the infectious population size for two age groups: 15–20 year olds and the 75+ age group. In both age groups, the results show marked uncertainty in the peak infectious counts but with more substantial variation for the older age group—a result that is particularly worrying given the greater risk of severe disease faced by older individuals for diseases such as COVID-19 [Verity et al., 2020]. Although, in our simulations, those in the 75+ age group see fewer daily contacts than those in the younger 15-20 age group (see Figure 5.1), the SEIR model simulations for the 75+ age group show more uncertainty than those for the 15-20 age group. This is due to the large relative uncertainty in the number of contacts for the 75+ age group indicated by the bootstrap samples (Figure 5.2).

Next, we examine the proportion of individuals infected with the disease at the end of the year, which we plot in Figure 5.4. Here, we consider only those individuals who have survived infection. This plot shows that those aged 5–20 are the most likely to have been infected: mainly because these individuals have the highest number of contacts and, because of this, are important drivers of infections within the population. Figure 5.4 also illustrates the pronounced uncertainty in the proportion exposed to infection in the oldest age groups.

Finally, in Figure 5.5, we study the effect of uncertainty in the contact matrix on the number of individuals that die of the infection. Although younger people are more likely to be infected due to their higher number of contacts, deaths occur mainly in the elderly, as expected based on the age-structured IFR which was used to parametrise our simulations. The bootstrapped samples of the contact matrix generate a wide range of deaths, particularly in the oldest age groups.

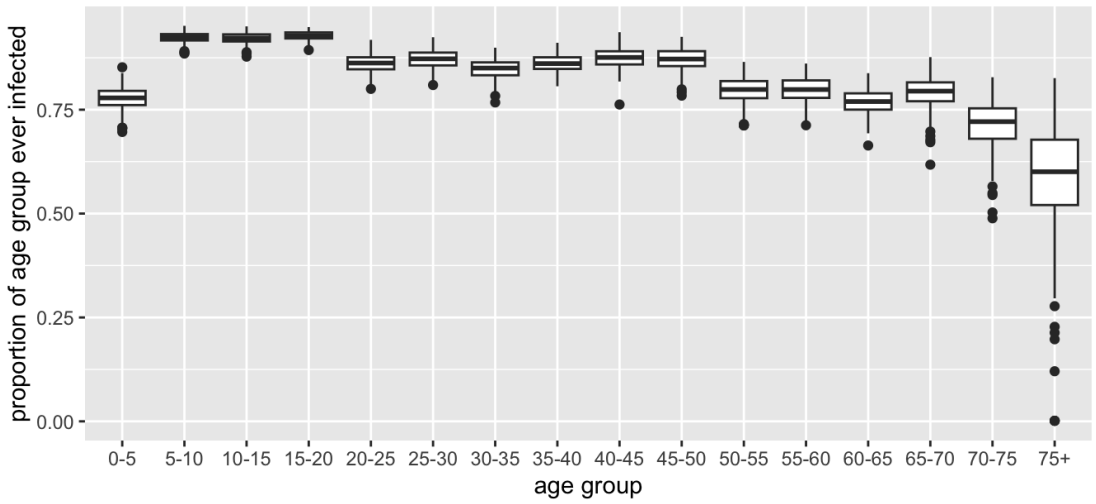


Figure 5.4: **Uncertainty in proportion ever infected arising from uncertainty in the contact matrix.** After running the SEIRD simulation for 1 year, we computed the proportion of each age group who had ever been in the infectious compartment (including those that subsequently died or recovered). The boxplots indicate the distribution of this proportion across the bootstrap samples of the contact matrix. Each boxplot indicates the median (central line) and the 25th and 75th percentiles (bottom and top of the box). The whiskers extend from the 25th percentile minus 1.5 times the interquartile range (IQR) to the 75th percentile plus 1.5 times the IQR; samples falling outside the whiskers are indicated by dots.

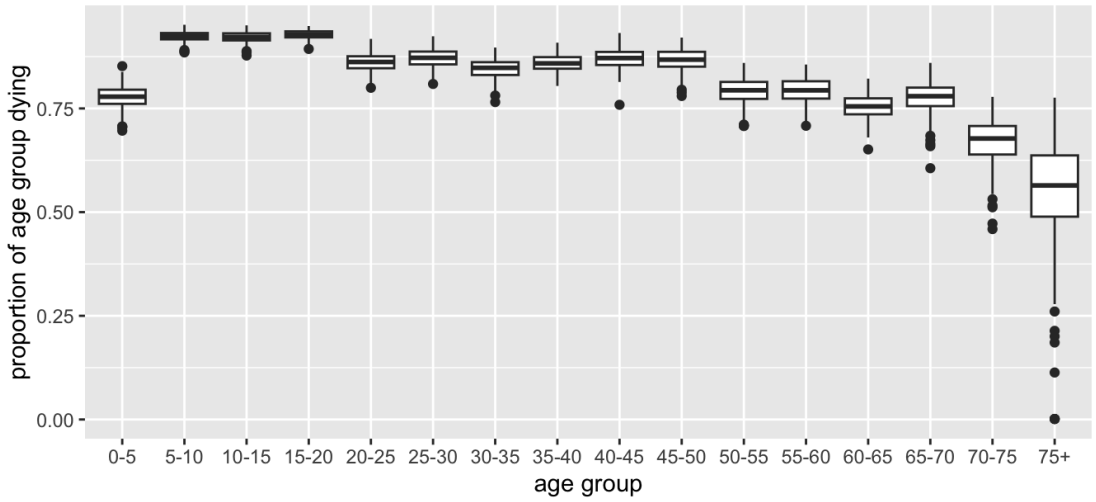


Figure 5.5: **Uncertainty in proportion dying arising from uncertainty in the contact matrix.** After running the SEIRD simulation for 1 year, we computed the proportion of each age group who have died from the disease. The boxplots indicate the distribution of this proportion across the bootstrap samples of the contact matrix. Each boxplot indicates the median (central line) and the 25th and 75th percentiles (bottom and top of the box). The whiskers extend from the 25th percentile minus 1.5 times the interquartile range (IQR) to the 75th percentile plus 1.5 times the IQR; samples falling outside the whiskers are indicated by dots.

5.3.3 Discussion

Many infectious diseases are predominantly spread from close contact with infected individuals and exhibit a strong relationship between age and risk of severe disease. Thus, mathematical models of disease transmission often depend on estimates of age-specific contact patterns. Sensitivity analyses of models with respect to contact matrix inputs are essential. Our results in this section indicate that considerable uncertainty exists in the values of the contact matrix for the United Kingdom on the basis of the contact data from [Funk, 2020, Mossong et al., 2017], and this uncertainty corresponds to a significant variation in model outputs.

There are a range of factors which the bootstrapped approach to uncertainty quantification, as employed in this section, does not consider. The algorithm assumes that the original survey from which the contact matrices are calculated is representative of the underlying population, which may not be true. For example, if contact data are collected primarily from an urban area in a country whose population is mostly rural, the resulting contact matrices would likely be unrepresentative of the country as a whole. Because the bootstrap algorithm does not allow for such biases in quantifying

uncertainty, it likely understates the true population-level uncertainty.

5.4 Data and software

The original R notebooks from which this thesis chapter is derived are available at `https://github.com/Como-DTC-Collaboration/como-models-math-biosci`. The R package implementing the models is available at `https://github.com/Como-DTC-Collaboration/como-models`.

Chapter 6

Understanding the impact of numerical solvers on inference for differential equation models

Overview

Most ordinary differential equation (ODE) models used to describe biological or physical systems must be solved approximately using numerical methods. In this chapter, we study *parameter inference* for models involving differential equations. Perniciously, even those solvers which seem sufficiently accurate for the *forward problem*, i.e., for obtaining an accurate simulation, may not be sufficiently accurate for the *inverse problem*, i.e., for inferring the model parameters from data. We show that for both fixed step and adaptive step ODE solvers, solving the forward problem with insufficient accuracy can distort likelihood surfaces, which may become jagged, causing inference algorithms to get stuck in local "phantom" optima. We demonstrate that biases in inference arising from numerical approximation of ODEs are potentially most severe in systems involving low noise and rapid nonlinear dynamics. We reanalyse an ODE changepoint model previously fit to the COVID-19 outbreak in Germany and show the effect of the step size on simulation and inference results. We then fit a more complicated rainfall-runoff model to hydrological data and illustrate the importance of tuning solver tolerances to avoid distorted likelihood surfaces. Our results indicate that when performing inference for ODE model parameters, adaptive step size solver tolerances must be set cautiously and likelihood surfaces should be inspected for characteristic signs of numerical issues.

Publications

The contents of this chapter are available as a preprint at:

- **R. Creswell,** K. M. Shepherd, B. Lambert, C. L. Lei, M. Robinson, and D. J. Gavaghan: "Understanding the impact of numerical solvers on inference for differential equation models," arXiv:2307.00749 (2023) [Creswell et al., 2023c]

  This manuscript is currently submitted to *Journal of the Royal Society Interface.*

  Contributions: I was the primary author of this preprint and conducted the investigation, methodology, theoretical analysis bounding the error in the log-likelihood, software implementations, and visualisation of results appearing in this chapter. Katherine Shepherd conducted preliminary work on smoothing approximations (based on a different ODE model) which informed the work presented §6.5. All authors of [Creswell et al., 2023c] made contributions and suggestions to the writing and revision of the preprint, and some of these contributions are reflected in the wording of parts of this chapter.

6.1 Introduction

In this chapter, we focus on the problem of inference for ODEs, which inevitably involves a sequence of numerical simulations of the ODE at different parameter values. We present a series of results elucidating the interplay between solver accuracy and biases in likelihoods and inference results.

In addition to finding widespread application in epidemiological modelling, ODEs are widely used throughout computational biology and other scientific fields (e.g., hydrology [Kavetski et al., 2003], cardiac electrophysiology [Whittaker et al., 2020], and population dynamics [Shertzer et al., 2002]). Thus, in this chapter, our treatment is more general and we do not focus exclusively on epidemiological problems. Our theoretical results are proved for ODE systems in general, and we then study an illustrative toy model drawn from mechanics (the driven oscillator) before studying inference problems involving real data drawn from both epidemiology and hydrology.

Numerical algorithms for solving the forward problem introduce error, but the properties of this error are generally well understood and can be controlled. In solvers using a fixed time step (discussed in §6.4.1), the error can be reduced by decreasing the size of the time step [Gautschi, 1997]. In solvers in which the time step is set adaptively (discussed in §6.4.2), the error is typically controlled through user-specified relative and

absolute tolerances on the local truncation error (the error in the solution introduced by a single time step of the solver) [Dormand and Prince, 1980]. Our focus in this chapter is on the interplay between the numerical approximations inherent in the forward problem and the *inverse problem*, which consists of learning values of the parameters that are compatible with an observed time series. As discussed in the earlier chapters of the thesis, some widely used approaches to the inverse problem include optimisation of an objective function which measures the quality of fit between the model and the data (e.g., maximum likelihood), or Bayesian approaches which generate samples from the posterior distribution of the parameters (e.g., Markov chain Monte Carlo (MCMC)). These approaches to the inverse problem require the forward problem to be solved at multiple different parameter values. The errors in each numerical solution of the forward problem, even when individually small, are liable to introduce significant bias to inference results.

The rest of this chapter is organised as follows. In §6.2, we present the widely used independent and identically distributed Gaussian noise log-likelihood function for fitting ODE models and derive a bound on the error in this log-likelihood arising from the use of an approximate solution to the ODE. On the basis of this bound, and results presented subsequently, we argue that the biases in inference results arising from numerical solvers are likely to be most severe in systems which have low noise and rapid nonlinear dynamics. In §6.3, we study two broad classes of ODE solvers: those involving a fixed time step, and those involving a time step set adaptively to control the error on the solution. Using forward simulations of a compartmental epidemiological model, we show that adaptive step size solvers can be significantly more efficient than fixed steps solvers. In §6.4, we study the effects that solver inaccuracy may have on inference, and illustrate this using synthetic data. Additionally, in §6.5, we study how smoothing approximations can reduce the influence of numerical error on computation of the likelihood. Finally, in §6.6 and §6.7 we consider inference of ODE models for real data series. in §6.6 we reanalyse an ODE model of disease transmission fit to the COVID-19 outbreak in Germany and show that, when using a solver with a fixed time step, the choice of time step can alter inference and simulation results, and in §6.7 we fit a rainfall-runoff model to hydrological data to illustrate the pitfalls of performing parameter inference using an adaptive step size solver with insufficient local tolerances.

6.2 Effects of numerical error on computation of the log-likelihood

6.2.1 Log-likelihood function for an ODE model

We assume that time series data $\{y_i\}_{i=1}^N$; $y_i \in \mathbb{R}^n$ are measured at time points $\{t_i\}_{i=1}^N$. These data are believed to obey some function $g : \mathbb{R}^l \to \mathbb{R}^n$ of $x(t;\theta) \in \mathbb{R}^l$, where x is the solution to an ordinary differential equation:

$$\begin{aligned} \frac{dx}{dt} &= h(t, x, \theta); \\ x(0) &= x_0, \end{aligned} \tag{6.1}$$

for some function h which is informed by scientific theory and parameterised by the (potentially unknown) values $\theta \in \mathbb{R}^m$.

We assume that the noise in the data obeys the standard independent and identically distributed (IID) Gaussian noise model (eq. (2.9)):

$$\log p(y_1, \ldots, y_N|\theta, \sigma) = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - g(x(t_i;\theta)))^2. \tag{6.2}$$

6.2.2 Error in the log-likelihood arising from approximation of the forward solution

The data are assumed to obey the IID Gaussian log-likelihood, eq. (6.2). We assume that $x(t_i;\theta)$ is the *true* solution to the ODE at time point t_i, which is unavailable and approximated by $\hat{x}_i$. The deviation between $x(t_i;\theta)$ and $\hat{x}_i$ at any time point is given by the global truncation error, $e(t_i)$:

$$e(t_i) = x(t_i;\theta) - \hat{x}_i.$$

In general, $e(t_i)$ is unknown, although, for particular numerical solvers, its magnitude may be bounded by some function of the step size or some other quantity which can be used to tune the accuracy of the solver.

The log-likelihood available to the inference algorithm takes the same form as eq. (6.2), but computed using the numerical approximation $\hat{x}_i$ instead of $x(t_i;\theta)$. For brevity, we denote the accurate log-likelihood by $\mathcal{L}$, and we denote the log-likelihood computed

using a numerical approximation by $\hat{\mathcal{L}}$, which is given by

$$\hat{\mathcal{L}} = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - g(\hat{x}_i))^2. \tag{6.3}$$

For brevity, let $g_i = g(x(t_i;\theta))$ and $\hat{g}_i = g(\hat{x}_i)$. We have:

$$\begin{aligned}
|\hat{\mathcal{L}} - \mathcal{L}| &= \left| -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \hat{g}_i)^2 + \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - g_i)^2 \right| && (6.4)\\
&= \frac{1}{2\sigma^2}\left| \sum_{i=1}^{N}(y_i - g_i)^2 - (y_i - \hat{g}_i)^2 \right| && (6.5)\\
&\leq \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left|(y_i - g_i)^2 - (y_i - \hat{g}_i)^2\right| && (6.6)\\
&= \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left|g_i^2 - \hat{g}_i^2 - 2y_i(g_i - \hat{g}_i)\right| && (6.7)\\
&= \frac{1}{2\sigma^2}\sum_{i=1}^{N}|(g_i - \hat{g}_i)(-2y_i + \hat{g}_i + g_i)| && (6.8)\\
&\leq \frac{1}{2\sigma^2}\sum_{i=1}^{N}|g_i - \hat{g}_i|\,|-2y_i + \hat{g}_i + g_i| && (6.9)\\
&\leq \frac{1}{2\sigma^2}\sum_{i=1}^{N}|g_i - \hat{g}_i|\,(|y_i - \hat{g}_i| + |y_i - g_i|) && (6.10)\\
&\leq \frac{1}{2\sigma^2}\sum_{i=1}^{N}|g_i - \hat{g}_i|\,(2|y_i - g_i| + |g_i - \hat{g}_i|)\,. && (6.11)
\end{aligned}$$

To proceed further, we impose the assumption of Lipschitz continuity of the observation function g with Lipschitz constant K, i.e., $|g(x_1) - g(x_2)| \leq K|x_1 - x_2|$ for all $x_1, x_2 \in \mathbb{R}^l$. We thus bound:

$$\begin{aligned}
|g(x(t_i;\theta)) - g(\hat{x}_i)| &\leq K|x(t_i;\theta) - \hat{x}_i| && (6.12)\\
&= K|e(t_i)|. && (6.13)
\end{aligned}$$

Using this in eq. (6.11), we have:

$$|\mathcal{L} - \hat{\mathcal{L}}| \leq \frac{1}{2\sigma^2} \sum_{i=1}^{N} K|e(t_i)| \left(2|y_i - g(x(t_i;\theta))| + K|e(t_i)|\right) \tag{6.14}$$

$$= \frac{1}{2\sigma^2} \sum_{i=1}^{N} K^2|e(t_i)|^2 + 2K|e(t_i)||y_i - g(x(t_i;\theta))| \tag{6.15}$$

$$= \sum_{i=1}^{N} \left(\frac{K^2}{2\sigma^2}|e(t_i)|^2 + \frac{K}{\sigma^2}|e(t_i)||y_i - g(x(t_i;\theta))| \right). \tag{6.16}$$

Assuming that the y_i are distributed according to the specified IID Gaussian likelihood, and that the likelihood is evaluated at the same parameter values that generated the data, we can easily compute the expectation of the bound. We have:

$$\mathbb{E}_{y_i \sim N(g(x(t_i;\theta)),\sigma)} \left[|y_i - g(x(t_i;\theta))|\right] = \sqrt{\frac{2}{\pi}}\sigma$$

so the bound follows:

$$\mathbb{E}_{y_i \sim N(g(x(t_i;\theta)),\sigma)} \left[|\mathcal{L} - \hat{\mathcal{L}}|\right] \leq \sum_{i=1}^{N} \left(\frac{K^2}{2\sigma^2}|e(t_i)|^2 + \sqrt{\frac{2}{\pi}}\frac{K}{\sigma}|e(t_i)| \right). \tag{6.17}$$

We observe an inverse relationship between σ and the expectation of the bound of $|\mathcal{L} - \hat{\mathcal{L}}|$ when $e(t_i)$ is held constant. Thus, when a solver is tuned to yield a particular global truncation error $e(t_i)$, we expect the absolute bias in the log-likelihood as a result of using this solver to be more severe at smaller values of σ. Additionally, at a fixed level of σ, we expect the bias in the log-likelihood to decrease as the global truncation error is decreased.

6.3 Effects of ODE solvers on forward simulations

6.3.1 Fixed step and adaptive step ODE solvers

A wide range of numerical algorithms have been developed to obtain approximate solutions to ODEs of the form given in eq. (6.1). These algorithms typically work by computing an approximate solution on a grid of time points (in general, distinct from the time points where the data are located) and then using an interpolation algorithm to obtain the solution at intermediate time points.

Most simply, the grid of solver time points can be prespecified in advance (we refer

to such methods as fixed time step solvers). However, in general, it is inefficient to use the same time step throughout the entire time range on which the ODE is being solved, particularly when solved repeatedly over a range of parameters. Solvers can employ large time steps in regions where the solution and its gradients change gradually without causing much error in the solution; however, in regions where the derivative changes rapidly, small time steps are required to maintain a low error. This motivated the development of ODE solvers which adjust the step size throughout the time domain over which the ODE is solved. While fixed step solvers are still commonly used, adaptive step solvers are standard in high performance computing and are widely implemented in software libraries for ODE solving.

When using an adaptive step size solver, the user does not specify a step size, but rather a local error tolerance. The algorithm then selects a time-varying sequence of step sizes such that the local error in the solution falls below the specified tolerance. The total number of time steps used by the solver thus depends on the selected tolerance and the properties of the solution. Typically, an interpolation scheme is then used to obtain the solution at intermediate time values. Tolerances can be expressed either as an absolute value or relative to the magnitude of the solution. In many implementations, both are available to the user: for example, the SciPy library allows the user to specify both an absolute tolerance `atol` and a relative tolerance `rtol`, and chooses step sizes such that the magnitude of the local truncation error on the solution x does not exceed $\texttt{atol} + \texttt{rtol}|x|$ [Virtanen et al., 2020].

6.3.2 Effect of integrator step size on the SEIRD model

We now revisit the SEIRD model from Chapter 5. We set its parameter values to be representative of transmission of the Delta variant of SARS-CoV-2: $\beta = 0.714, \kappa = \frac{1}{3.7}, \gamma_i = \frac{1}{7}(1 - \text{IFR}), \mu_i = \frac{1}{7}\text{IFR}$ [van der Vegt et al., 2022, Verity et al., 2020]. As in Chapter 5, we note that although these parameter choices ensure that our simulation outputs are not wildly unrealistic, due to the simplicity of the modelling approach taken in this section our simulation outputs are not intended to resemble the actual outbreak of the Delta variant in the United Kingdom or any other country. We solve the model using the forward Euler method, for two different choices for the solver step size. The first is a daily time step, which might be considered a reasonable choice given that we are interested in simulating daily cases and deaths. We also consider a much smaller time step of 0.001 days. The daily deaths and cases are plotted in Figure 6.1. The results indicate that for the SEIRD model with these parameters, a step size of 1 day results in

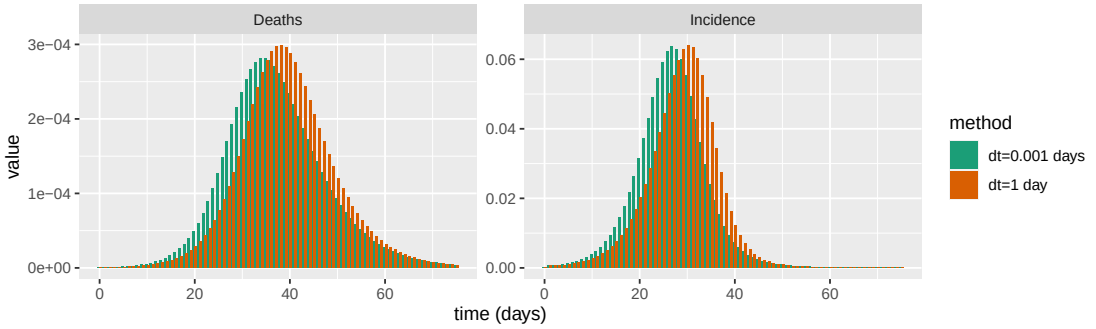


Figure 6.1: **Effect of daily solver time step on simulations of daily deaths and cases.** (left) daily deaths simulated from the SEIRD model using the Forward Euler solver with two different choices of the step size. (right) daily incidence simulated from the SEIRD model using the Forward Euler solver with two different choices of the step size.

substantial differences in simulation outputs relative to a time step of 0.001 days. The larger step size resulted in a higher peak case load which occurred later than observed with the finer time step. It also delayed the peak in deaths.

6.3.3 Adaptive step size solvers for compartmental models

The chief advantage of numerical solvers such as the uniformly spaced forward Euler used above, which take a fixed grid of time points on which to calculate the approximate solution, is their ease of implementation. However, as seen above, these methods suffer because:

1. To obtain an accurate solution, the solver step size must be set to some small value, but it is often unknown how small this value should be to ensure a reasonable approximation.

2. When the solver step size is set to a small value, the method may be impractically slow.

Both of these defects are addressed by adaptive step size methods. In these, the user specifies not a step size but a tolerance—some relative or absolute value which the local error in the approximate solution should not exceed. Then, the solver algorithm selects the appropriate step size in order for the solution to meet this tolerance, using the theoretical error properties of the solver and various techniques for step size adaptation. Adaptive solvers are able to vary the step size over the course of the solve, selecting very small values only in those regions of time where this is necessary, such as where the solution is changing rapidly over time, and otherwise increasing the step size to

larger values. For this reason, they are much more efficient than fixed step solvers [Gautschi, 1997].

A wide variety of approaches to step size adaptation have been proposed. At each step, these methods typically use an estimator of the error introduced at that time step, and then—at least roughly—they select the largest possible step such that the threshold imposed by the user-supplied tolerance is not exceeded.

In order to demonstrate the advantages of adaptive step solvers, in Figure 6.2, we compare the performance of the forward Euler solver with a fixed, small step size to that of the LSODA adaptive step size solver [Hindmarsh and Petzold, 2005]. Both methods are seen to achieve similarly accurate solutions. However, we note the significant speed advantage of the LSODA solver, which here is roughly two orders of magnitude faster (see Table 6.1).

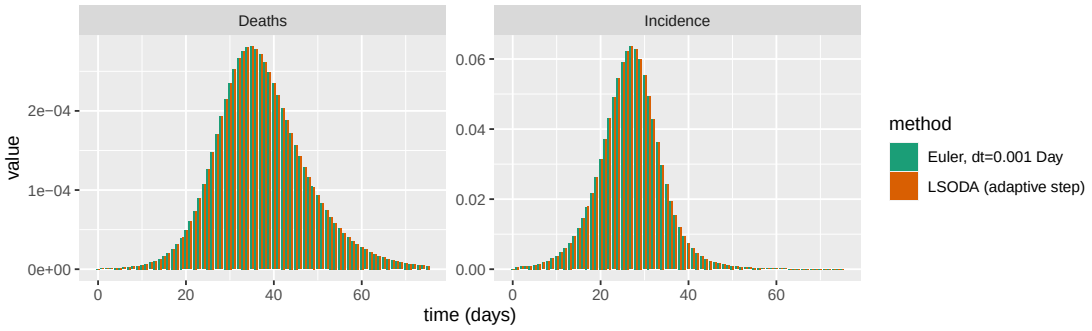


Figure 6.2: **Comparison of simulations of the SEIRD model computed using adaptive and fixed step solvers.** (left) daily deaths simulated from the SEIRD model using the fixed-step Forward Euler solver and the LSODA solver. (right) daily incidence simulated from the SEIRD model using the fixed-step Forward Euler solver and the LSODA solver.

Solver	Runtime
Forward Euler $\Delta t = 0.001$ Day	1.885 seconds
LSODA	0.022 seconds

Table 6.1: Runtimes for the simulations plotted in Figure 6.2. The simulations were performed in R version 4.2.1 on a 2.42 GHz core.

6.4 Effects of ODE solvers on inference

To study the interplay between ODE solvers and inference, we introduce the following differential equation problem which describes an oscillatory system with damping and

forcing:

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = F(t).$$

The model has three parameters which will be treated as unknown: (m, c, k). In classical mechanics, these represent the mass, damping coefficient, and spring constant respectively. $F(t)$ represents the forcing function or stimulus, and in this chapter takes a variety of forms throughout our results. This damped and forced oscillator is described by a second order differential equation; to apply ODE solvers straightforwardly, we rewrite it as a first order differential equation of two state variables:

$$\frac{d}{dt}\begin{pmatrix} x \\ \dot{x} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \frac{F(t)}{m} - \frac{c}{m}\dot{x} - \frac{k}{m}x \end{pmatrix}, \tag{6.18}$$

where $\dot{x} = dx/dt$. Structural identifiablity of the ODE model was assessed using the SIAN toolbox [Hong et al., 2019], which showed that the unknown parameters were structurally identifiable.

Typical log-likelihood surface shapes

We now consider the influence of the two numerical solution methods for parameter inference. Because fixed step solvers use the same grid throughout parameter space, while adaptive step solvers may employ different grids at different parameter values, these two classes of solvers differ in the characteristics of the error that they may introduce into the likelihood function.

For the inference results presented in this chapter, we fix `atol` to a value of 10^{-9} and tune `rtol` to control the accuracy of the solver. Adaptive step sizes have been implemented for a wide variety of ODE solver algorithms. For our subsequent inference results, we focus on Runge-Kutta methods of the form RK$p(q)$, which use the qth order method to estimate the error (and thus control the time step), while making the actual steps using the pth order method [Dormand and Prince, 1980]. Runge-Kutta methods are not described in detail here for brevity—they are widely used and details can be found in many standard texts (for example, [Gautschi, 1997]). We rely on the SciPy adaptive time step Runge-Kutta implementation, which employs a quartic interpolation polynomial for RK5(4) and a cubic Hermite interpolation polynomial for RK3(2) [Virtanen et al., 2020].

We illustrate this by computing the likelihood surface for the k parameter in the oscillator problem, eq. (6.18). 75 evenly spaced data points were generated from and

including $t = 0$ to $t = 50$ from the model with an exact solution, using true parameter values $k = 1$, $c = 0.2$, $m = 1$, initial conditions of $x(t = 0) = 0$, $\dot{x}(t = 0) = 0$, and

$$F(t) = \begin{cases} 1, & t < 25, \\ 0.9, & t \geq 25. \end{cases}$$

Then IID Gaussian noise was added to the solution at each of the sampled locations with $\sigma = 0.01$. Holding all other parameters fixed at their true values, the log-likelihood was calculated for a range of values of k, using three different ODE solvers. First, the exact solution to the ODE was used to compute the accurate ('True') likelihood. Next, the Forward Euler solver with a fixed time step of $\Delta t = 0.01$ was used. Finally, we used the RK5(4) solver, but with its relative tolerance tuned so the observed magnitude in the error in the log-likelihood at the true parameter values was equal to that produced by the Forward Euler solver (for this problem, this resulted in relative tolerance tuned to 0.00944). These results are shown in Figure 6.3.

At the true parameter value, both solvers result in a slight underestimation of the log-likelihood. Across the parameter range considered, the fixed time step solver results in a log-likelihood which is shifted relative to the true one, but retains the smooth, unimodal shape. However, the adaptive step solver results in a log-likelihood surface which in addition to being shifted exhibits jagged, discontinuous fluctuations. In the remainder of §6.4, we examine these two phenomena in more detail.

6.4.1 Fixed time step solvers

Forward Euler solver

One of the simplest numerical solvers for ordinary differential equations is the Forward Euler method with a uniform step size Δt (see Chapter 2, Example 7). This solver is easily implemented and thus has achieved wide usage despite its simplicity and typically mediocre performance.

Forward Euler has been used for inference in some recent high-profile epidemiological research where Δt was set to a value comparable to the time scale of the behaviour of the system (e.g., [Birrell et al., 2021, Dehning et al., 2020]). Whether these applications are representative of the use of Forward Euler more generally is unclear, but our results in §6.6 indicate that such choices of Δt may alter both forward model solutions and parameter inference results.

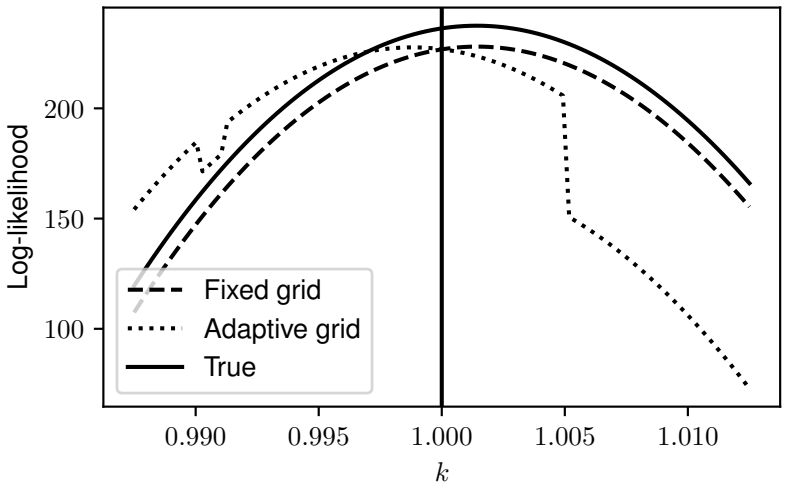


Figure 6.3: **Comparison of log-likelihood surfaces calculated using fixed step and adaptive step solvers.** Log-likelihood for the parameter k calculated from data generated from the oscillator model eq. (6.18), with all other parameters held at their true values. The log-likelihood was calculated from eq. (2.9) using the exact solution (True), a Forward Euler solver with a fixed time step $\Delta t = 0.01$, and an adaptive step RK5(4) solver with tolerance tuned such that at the true parameter values (vertical line) it introduces the same magnitude of error in the log-likelihood as the fixed step Forward Euler solver (corresponding to a relative tolerance of 0.00944).

Inference for the damped, driven oscillator using Forward Euler

We now exemplify the impact of using Forward Euler with insufficiently small time steps on inference by using synthetic noisy data generated from the (accurate) solution of eq. (6.18). 25 evenly spaced data points were generated from and including $t = 0$ to $t = 5$ from the model using true parameter values $k = 1$, $c = 0.2$, $m = 1$, an initial condition of $x(t = 0) = 0$, $\dot{x}(t = 0) = 0$, and $F(t) = 1$. Then, IID Gaussian noise was added to the solution at each of the sampled locations with $\sigma = 0.1$. Holding all other parameters fixed at their true values, the log-likelihood was calculated for a range of values of k, using the Forward Euler solver with various time steps.

Figure 6.4 shows the impact of using Forward Euler on the likelihood surface. The results show the typical effect of a fixed step solver with insufficiently small time steps: the likelihood surface maintains a smooth shape, but it is shifted relative to its true location. The longest time step considered in this study, $\Delta t = 0.1$, causes substantial inaccuracy in the likelihood even though $\Delta t = 0.1$ is small compared to the time scale of the dynamics of the system and the system with $F(t) = 1$ contains no discontinuities or other challenging features.

As the step size is refined, the log-likelihood curves converge. This suggests a diagnostic technique which could be incorporated into inference algorithms: once the optimal parameter values have been determined, the log-likelihood should be evaluated at those parameter values with the step size on the solver slightly adjusted; if the solver is sufficiently accurate, the value of the log-likelihood should not be a strong function of the step size.

6.4.2 Adaptive step size solvers

Adaptive step size solvers enable increased efficiency in obtaining accurate solutions to ODEs. However, when used in inference problems, they can convert a smooth likelihood surface into a rough one, characterized by rapid (and entirely phantom) changes in the likelihood which interfere with inference algorithms. These inaccuracies in the likelihood can be observed even at tolerances in the solution error where further refinements do not visibly influence the solution. For example, in cardiac electrophysiology, jagged parameter likelihoods have been observed with adaptive step size ODE solvers with tolerances as low 10^{-7} [Johnstone, 2018, Mirams, 2018]. Next, we investigate the origin of the jagged likelihoods using synthetic data from the oscillator model described in eq. (6.18).

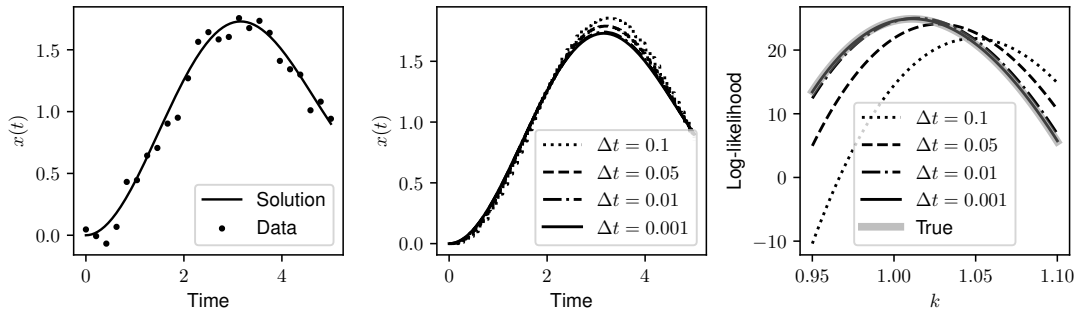


Figure 6.4: **Damped oscillator inference using Forward Euler.** (Left) Synthetic data for the damped driven oscillator. The curved line indicates the accurate solution to the ODE with these parameters, while the points indicate the noisy data. (Center) Solution for oscillator computed using a Forward Euler solver with four different choices for the time step Δt. (Right) Log-likelihood for the parameter k calculated from the noisy data, with all other parameters held at their true values. The log-likelihood was calculated from eq. (2.9) using a Forward Euler solver with four different choices for the time step Δt.

Inference for the damped, driven oscillator using an adaptive step size solver

We first study the effects of adaptive time step solvers on inference using the model system that was introduced at the beginning of §6.4 (eq. (6.18)). Here, we set the input stimulus according to

$$F(t) = \begin{cases} 1, & t < t_{\text{change}}, \\ f_1, & t \geq t_{\text{change}}. \end{cases} \tag{6.19}$$

Thus, f_1 controls the strength of a pulse provided to the system at $t = t_{\text{change}}$.

First, we consider the problem where $f_1 = -1$ and $t_{\text{change}} = 2.5$ for different choices of the RK5(4) solver tolerance. 25 evenly spaced data points were generated from and including $t = 0$ to $t = 5$ from the model, using true parameter values $k = 1$, $c = 0.2$, $m = 1$ and an initial condition of $x(t = 0) = 0$, $\dot{x}(t = 0) = 0$. Then, IID Gaussian noise was added to the solution at each of the sampled locations with $\sigma = 0.1$. Holding all other parameters fixed at their true values, the log-likelihood was calculated for a range of values of k, using the RK5(4) solver with various tolerances. These results are shown in Figure 6.5. At insufficient tolerances, the log-likelihood surface exhibits significant erroneous jaggedness. Notably, visual changes between the forward simulations are minor even at tolerances which cause drastic differences in the log-likelihood.

Next, we fix the adaptive solver tolerance and study how introducing more rapid changes in the system's behaviour affects the log-likelihood surface. In Figure 6.6, we

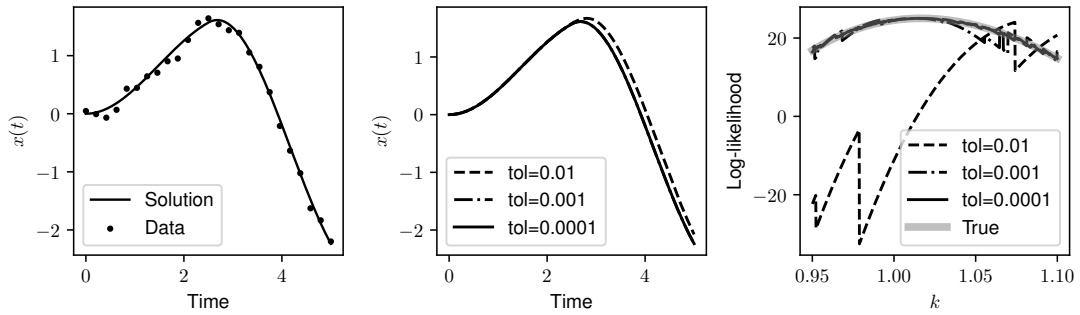


Figure 6.5: **Damped oscillator inference using adaptive time step Runge-Kutta.** (Left) Synthetic data for the damped driven oscillator. The curved line indicates the accurate solution to the ODE with these parameters, while the points indicate the noisy data. (Center) Solution for oscillator computed using an RK5(4) solver with three different choices for the relative tolerance (indicated by tol in the legend). (Right) Log-likelihood for the parameter k calculated from the noisy data, with all other parameters held at their true values. The log-likelihood was calculated from eq. (2.9) using an RK5(4) solver with three different choices for the tolerance.

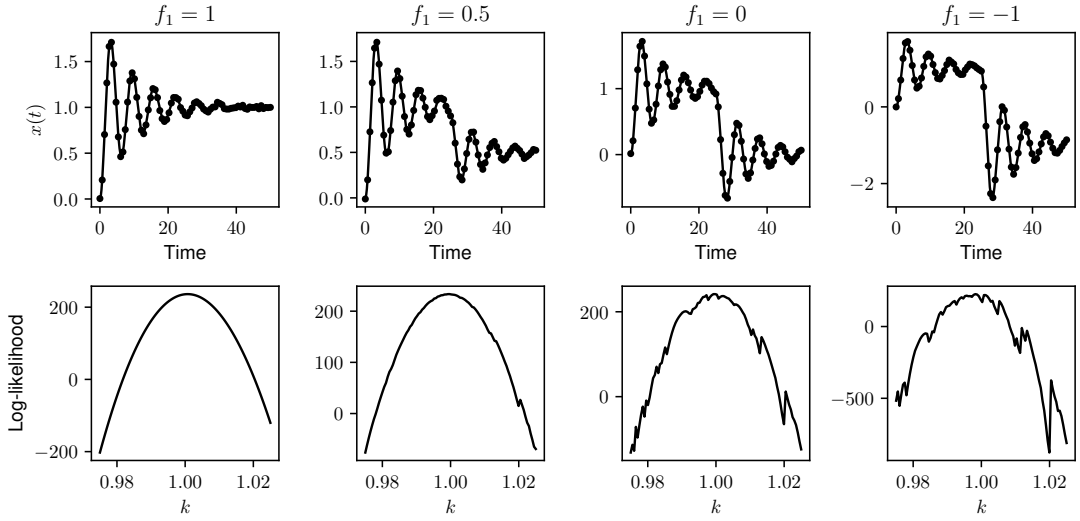


Figure 6.6: **Damped oscillator model: forward simulations and inference using an adaptive solver.** Time series data and parameter likelihood surfaces are shown for four values of f_1 in the oscillator problem: eqs. (6.18) & (6.19). For each value of f_1, the top plot shows the accurate ODE solution (line) and noisy synthetic data (points) generated from it. The bottom plot panels show the corresponding log-likelihood surface for k over an interval centred on the true value, $k = 1$, while all other parameters are held at their true values. For generating the likelihood surfaces, an RK5(4) solver was used with $\texttt{rtol} = 10^{-3}$.

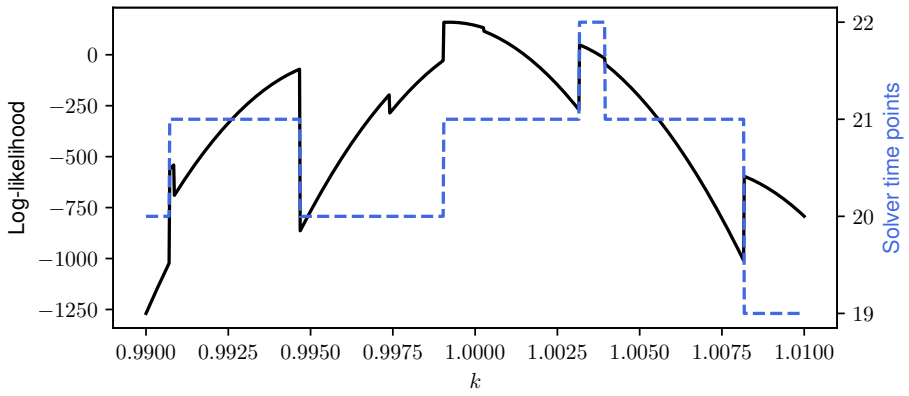


Figure 6.7: **Damped oscillator model: likelihood discontinuities caused by variation in the number of adaptive steps.** The log-likelihood surface for the parameter 5 in the oscillator problem (black solid line) and the number of time points used by the adaptive step size ODE solver in the calculation of each value of the likelihood (blue dashed line) are shown. An RK5(4) solver was used with $\texttt{rtol} = 10^{-3}$.

fix $t_{\text{change}} = 25$ and consider four different values of f_1 and plot the likelihood surface for the model parameter m calculated according to an RK5(4) solver with $\texttt{rtol} = 10^{-3}$. For each value of f_1, 75 evenly spaced data points were generated on the interval from and including $t = 0$ to $t = 50$, using parameter values $k = 1$, $c = 0.2$, and $m = 1$. IID Gaussian noise was added to the solution at each of the sampled locations with $\sigma = 0.01$. The likelihood was then calculated over a range of values of k, with all other parameters held at their correct values. For $f_1 = 1$, the stimulus $F(t)$ is constant over time, and the likelihood surface appears smooth. However, as f_1 is adjusted so the stimulus is a stronger pulse, the likelihood becomes jagged with large deviations away from the true likelihood surface. (This is an example of a challenging RHS which could be made more tractable for inference using smoothing approximations, which we analyse in §6.5.) Overall, these results indicate that the more rapid the changes in a system's behaviour, generally the tighter solver tolerances are required to solve the inverse problem.

A fundamental point to note is that these inaccuracies arise because different values of the parameters represent different forward problems, and the solver selects a different sequence of step sizes for each. When the solution contains regions of rapid change, differences in the positions of the solver time steps, and, particularly, the inevitably discontinuous jumps in the total number of time steps used by the solver, cause errors in the likelihood. This phenomenon is investigated more closely in Figure 6.7. For this study, the oscillator model eq. (6.18) was again used. 50 evenly spaced data points were generated on the time interval from and including $t = 0$ to $t = 10$, with $t_{\text{change}} = 5$ and

$f_1 = -5$, using parameter values $m = 1$, $c = 0.2$, and $k = 1$. IID Gaussian noise was added to the solution at each of the sampled locations with $\sigma = 0.01$. The likelihood for k was calculated as before and is plotted in Figure 6.7. In this case, the figure is restricted to a very narrow range of k values, and the total number of time points selected by the adaptive solver for the calculation of the likelihood at each value of k is overlaid on the plot. Here, the large jumps in the likelihood correspond to the addition or removal of a solver time point. Smaller spikes and jaggedness where the total number of solver time points is constant correspond to shifting of the solver time points.

Effect of jaggedness on inference algorithms

The jagged spikes appearing in the likelihood surface as a result of insufficiently accurate adaptive step size solvers plague computational inference algorithms. A common approach to Bayesian inference is to use the Metropolis MCMC algorithm, or variants of it [Gelman et al., 2013]. This algorithm generates a sequence of parameter values via a Markov chain whose stationary distribution is the posterior distribution of the parameters. Given the most recent parameter values in the chain θ^{old}, the Metropolis algorithm proposes new parameter values θ^{prop} according to a proposal distribution and then accepts θ^{prop} with a probability of:

$$r = \min\left(1, \frac{p(\theta^{\text{prop}})}{p(\theta^{\text{old}})} \frac{p(y|\theta^{\text{prop}})}{p(y|\theta^{\text{old}})}\right),$$

where $p(\theta^{\text{prop}})$ is the prior and $p(y|\theta^{\text{prop}})$ is the likelihood. To illustrate the detrimental effects of jagged errors in the likelihood, we consider a situation where θ^{old} and θ^{prop} have identical values under the prior and the accurately computed likelihood (this is a plausible assumption when θ^{old} and θ^{prop} are nearby), but we assume that the log-likelihoods at these two parameter values computed using the numerical approximation differ by some factor c driven by numerical error in the adaptive step size solver (i.e., $\log p(y|\theta^{\text{prop}}) = \log p(y|\theta^{\text{old}}) - c$, for $c > 0$). This assumption of a jump in computed likelihood values at nearby parameter values is analogous to the spikes appearing in the log-likelihood in our results in Figures 6.6 and 6.7.

Under these assumptions, $\log r = -c$ or $r = \exp(-c)$. For a value of $c = 10$ (smaller than many of the magnitudes of spikes observed in our results), the probability of accepting the proposal is less than 1 in 20,000. Even a relatively small jump of magnitude $c = 3$ will be traversed by the sampler with a probability of only about 5%. Although these computations are based on simplistic assumptions, they suggest that even minor

warping of the log-likelihood may severely restrict the ability of a Metropolis-Hastings sampler (or similar inference algorithm) to traverse the parameter space efficiently.

6.4.3 The impact of observation error magnitude on inference and sampling performance

In this section, we empirically study the effects of different levels of observation noise on inference. We performed Bayesian inference using MCMC for the oscillator problem with varying levels of noise in the data. We considered two values of σ (0.01 and 0.1) to generate the data, fixed $f_1 = -1$, and otherwise generated data exactly as described for Figure 6.6. We set a uniform prior on $[0.1, 1.5]$ for the three model parameters m, c, and k, and a uniform prior on $[0, 1]$ for the σ. Three MCMC chains were run, initialized at random samples from the prior (with the same MCMC starting point being used for both choices of the true σ). 1500 iterations of MCMC were performed using the Haario-Bardenet adaptive covariance algorithm as implemented in PINTS to sample from the posterior [Haario et al., 2001, Clerx et al., 2019]. The MCMC chains for the m parameter are plotted in the left column of Figure 6.8 using the RK5(4) solver with $\texttt{rtol} = 10^{-3}$, while the right column of Figure 6.8 shows the chains using the same solver but with more accurate tolerances of $\texttt{rtol} = 10^{-8}$.

At the lowest noise level considered ($\sigma = 0.01$), the three MCMC chains using the less accurate solver move towards the true value of the parameter but fail to mix with each other. Instead, each chain remains stuck in a narrow region of parameter space near the true parameter value, corresponding to the phantom local maxima in the likelihood surface observed in Figure 6.6. Reducing the solver tolerance to 10^{-8} was, however, sufficient to ensure chain mixing, indicating that the lack of convergence was purely an artefact of using an inaccurate solver. At the highest level of noise considered here ($\sigma = 0.1$), the three MCMC chains mix well for either tolerance choice,[1] which can be explained by our bound given in eq. (6.11): that larger σ values lead to gentler variation in the log-likelihood surface and so easier exploration by inference algorithms.

[1] We note that, for this level of noise, the centers of the sampling distributions are shifted slightly away from the true parameter value because the noise limits our ability to estimate this parameter.

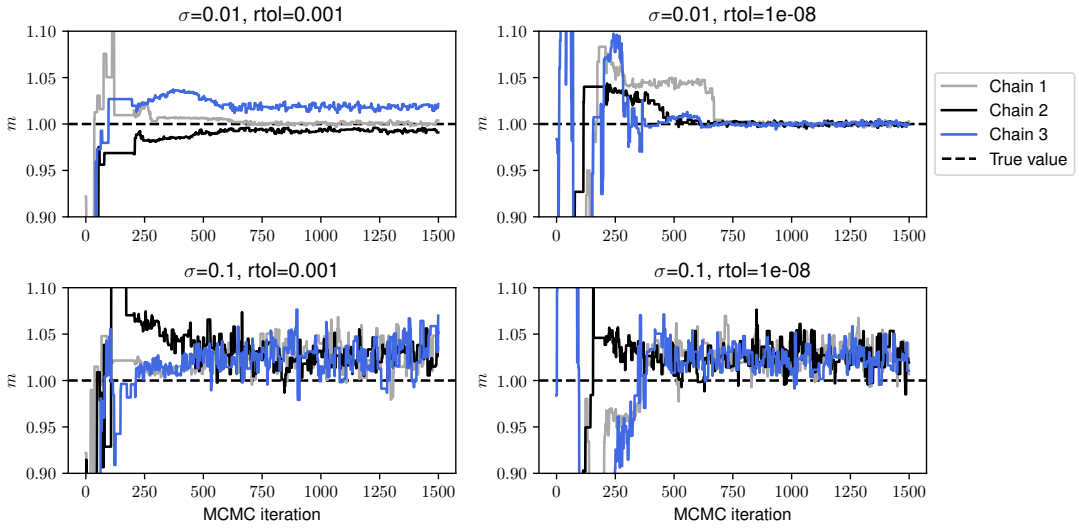


Figure 6.8: **Effect of noise on MCMC convergence.** Data were generated according to the same specifications as for Figure 6.6, with $f_1 = -1$, and the indicated values of σ. Inference was performed for the three parameters m, c, and k, as well as σ, via adaptive covariance MCMC [Haario et al., 2001] with three independent chains initialized at random samples from the prior (uniform on $[0.1, 1.5]$ for the model parameters, and uniform on $[0, 1]$ for σ). 1500 MCMC iterations were performed. The plots show the three chains for the m parameter. (Left) Forward simulation was performed using the RK5(4) solver with $\texttt{rtol} = 10^{-3}$. (Right) Forward simulation was performed using the RK5(4) solver with $\texttt{rtol} = 10^{-8}$.

6.5 Smoothing forcing terms to reduce numerical errors in the likelihood

As indicated by our results in Figure 6.6, discontinuities in the right-hand side (RHS) of an ODE can lead to substantial errors in the likelihood when adaptive step size solvers are used. In general, errors in the likelihood arising from numerical errors in the solution can be reduced by refining the tolerance of the adaptive solver. However, when the RHS suffers from a discontinuity, the required solver tolerance to obtain an acceptable likelihood surface may employ a prohibitively large grid of solver points. Several approaches to remove discontinuities from the RHS have been developed to enable more accurate forward simulations, including smoothing approximations and solving the ODE separately within regions where the RHS is continuous [Stewart, 2011]. These techniques may be particularly advantageous when performing inference. In this section, we study the effects on the computation of the log-likelihood of one of these approaches, which is to smooth discontinuities in the RHS of the ODE. Smoothing is often a particularly appropriate assumption for biological models, where a continuous rather than an instant change may in fact more realistically represent the true behaviour of the system. For example, in epidemiology, interventions (such as the introduction of a vaccination campaign) may be naively represented by discontinuous step functions in the RHS of a compartmental epidemiological model; however, a function smoothly moving between two values (corresponding to the intervention reaching its full effect gradually over an appropriate period of time) is both more realistic and more tractable for numerical solvers for the forward problem [van der Vegt et al., 2022].

The hyberbolic tangent function ($\tanh$) is a useful smooth approximation to a step function. In the forced oscillator problem, we can use $\tanh$ to approximate the step function stimulus, (6.19), with $f_1 = -1$ according to:

$$F^{\text{smooth}}(t) = -\tanh\left(\frac{t - t_{\text{change}}}{a}\right) \tag{6.20}$$

where a is a tuning parameter controlling the level of smoothing, with larger values of a leading to a more gradual change in the stimulus, and t_{change} is the time when the stimulus changes in value.

To examine the effect of the smoothing approximation on inference, we computed the likelihood surface for the k parameter in the forced oscillator model using a variety of choices for the smoothing parameter, with results shown in Figure 6.9. Using $f_1 = -1$ and $t_{\text{change}} = 2.5$, 25 evenly spaced data points were generated from and including $t = 0$

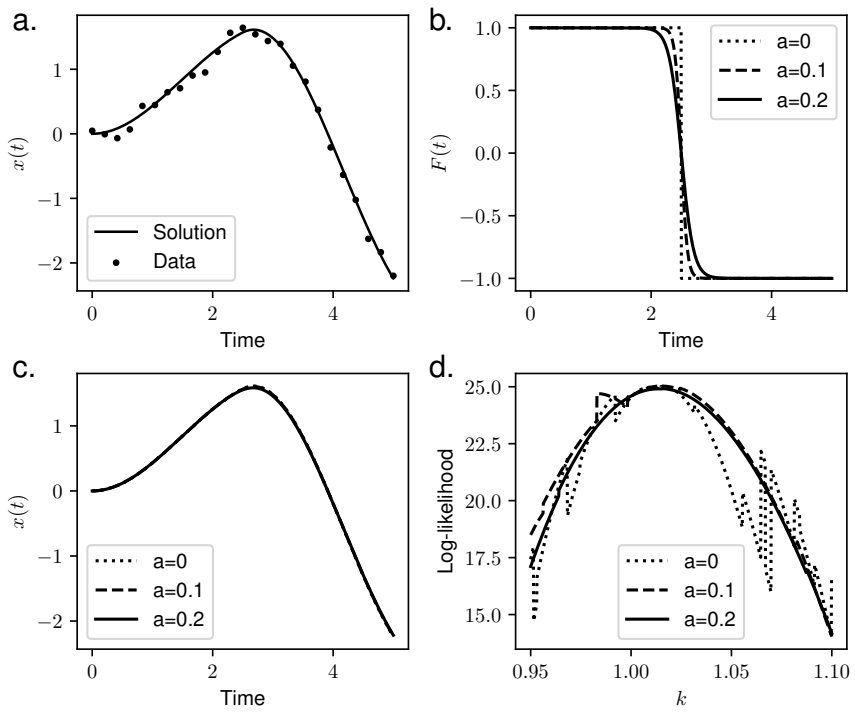


Figure 6.9: **Effect of tanh-smoothing on likelihood surface.** (a.) Synthetic data for the damped driven oscillator. The curved line indicates the accurate solution to the ODE with these parameters, while the points indicate the noisy data. (b.) The three considered forms of the stimulus. $a = 0$ indicates the unsmoothed stimulus (eq. (6.19)), while the positive values of a indicate the tanh-smoothed stimulus according to eq. (6.20). (c.) Solution for oscillator computed using an RK5(4) solver with relative tolerance 10^{-3}, with three different forms of the stimulus, at the true parameter values. (d.) Log-likelihood for the parameter k calculated from the noisy data, with all other parameters held at their true values. The log-likelihood was calculated from eq. (6.2) using an RK5(4) solver with relative tolerance 10^{-3}.

to $t = 5$ from the model with an accurate solver (the RK5(4) solver with relative tolerance set to 10^{-8}), using true parameter values $k = 1$, $c = 0.2$, $m = 1$ and an initial condition of $x(t = 0) = 0$, $\dot{x}(t = 0) = 0$. Then, IID Gaussian noise was added to the solution at each of the sampled locations with $\sigma = 0.1$. Holding all other parameters fixed at their true values, the log-likelihood was calculated for a range of values of k, using the RK5(4) solver with relative tolerance tuned to 10^{-3}. The likelihood was computed using both the original step function stimulus eq. (6.19) (indicated in Figure 6.9 by $a = 0$), as well as the smooth approximation eq. (6.20) with two different choices of $a > 0$. Without smoothing, we observe significant jagged biases in the likelihood, as expected due to the insufficient solver tolerance. However, with smoothing of the RHS, a smooth, tractable

likelihood surface is obtained despite the mediocre solver tolerance. This is despite the fact that all forward simulations are visually very similar. This is in accordance with our results in Figure 6.5, where even visibly small changes in the forward solution may hide the fact that there lurks substantial distortions of the likelihood surface.

6.6 Fixed step solvers applied to an SIR change point model of the spread of COVID-19 in Germany

As discussed in the previous chapter (Chapter 5), a widely used class of differential equation models in epidemiology are compartmental models, which divide the population into a number of compartments representing different diseased or non-diseased states and specify the rates at which individuals move from one compartment to another [van der Vegt et al., 2022]. A simple yet commonly used example is the SIR model (susceptible-infected-recovered) [Weiss, 2013]. This model keeps track of the number of susceptible individuals $S(t)$ (those who can be infected with the disease), infected individuals $I(t)$ (those who are currently infectious with the disease), and recovered individuals $R(t)$ (those who have recovered from the disease and are assumed immune). Neglecting births and deaths, the model is expressed by the following system of differential equations:

$$\frac{dS}{dt} = -\lambda \frac{SI}{N} \tag{6.21}$$

$$\frac{dI}{dt} = \lambda \frac{SI}{N} - \mu I \tag{6.22}$$

$$\frac{dR}{dt} = \mu I, \tag{6.23}$$

where $\lambda > 0$ is the spreading rate of the disease, $\mu > 0$ is the recovery rate, and $N > 0$ is the total size of the population. The system additionally requires the specification of initial conditions for each compartment ($S(0)$, $I(0)$, $R(0)$). $I(0)$ must exceed zero for an infection to spread.

The qualitative behaviour of the SIR model can be determined by the basic reproduction number, R_0, where

$$R_0 = \frac{\lambda}{\mu}.$$

Assuming that $S(0) \approx N$ and $I(0) > 0$, when $R_0 > 1$, the number of infected individuals will tend to grow, for $R_0 < 1$, the number of infected individuals will fall. Thus, fitting an SIR model to infection data, and estimating the spreading rate λ and reproduction

number R_0, are important steps in understanding and predicting the progression of an epidemic.

An extension to the standard SIR model has λ vary over time, allowing the model to capture changes in the spread of a disease caused by behavioural changes or government policy. In the aftermath of the outbreak of COVID-19 in Europe in early 2020, an SIR model allowing changes in λ through time was used in a high profile paper which attempted to capture the impact of major public health policy interventions on COVID-19 transmission in Germany [Dehning et al., 2020]. The authors used the model eqs. (6.21)–(6.23), discretised with a one day time step, equivalent to a Forward Euler solver with $\Delta t = 1$:

$$S_t = S_{t-1} - \lambda(t)\Delta t \frac{S_{t-1}I_{t-1}}{N} \tag{6.24}$$

$$I_t = I_{t-1} + \lambda(t)\Delta t \frac{S_{t-1}I_{t-1}}{N} - \mu\Delta t I_{t-1} \tag{6.25}$$

$$R_t = R_{t-1} + \mu\Delta t I_{t-1}. \tag{6.26}$$

The initial condition was given by an unknown parameter $I_0 = I(0)$. The system was closed with $R(0) = 0$ and $S(0) = N - I_0$. The spreading rate λ was assumed to be a continuous function of time and was allowed to shift at three time points, whose locations were estimated from the data. Specifically, these three time points, $t_i, i \in \{1, 2, 3\}$ denoted the times at which λ began to (linearly) change to a new, constant value, and the time taken for these shifts was dictated by durations d_i. The λ profile then has the following piecewise representation:

$$\lambda(t) = \begin{cases} \lambda_0, & t < t_1, \\ \lambda_0 + \frac{\lambda_1 - \lambda_0}{d_1}(t - t_1), & t_1 \le t < t_1 + d_1, \\ \lambda_1, & t_1 + d_1 \le t < t_2, \\ \lambda_1 + \frac{\lambda_2 - \lambda_1}{d_2}(t - t_2), & t_2 \le t < t_2 + d_2, \\ \lambda_2, & t_2 + d_2 \le t < t_3, \\ \lambda_2, + \frac{\lambda_3 - \lambda_2}{d_3}(t - t_3) & t_3 \le t < t_3 + d_3, \\ \lambda_3, & t_3 + d_3 \le t. \end{cases}$$

Additional features of the model included a reporting delay and a weekly modulation. The reporting delay was characterised by a single parameter D indicating the number of days between the time at which new infections occur and the time at which they are reported. The modulation accounts for the weekly periodicity evident in the

data and is characterised by two parameters f_w and Φ_w. This significant periodicity likely arises from processes involved in the reporting of COVID-19 cases and deaths [Gallagher et al., 2023]. Specifically, cases C_t are modelled by:

$$C_t = (1 - f(t))I^{\text{new}}_{t-D}, \tag{6.27}$$

where

$$f(t) = (1 - f_w)\left(1 - \left|\sin\left(\frac{\pi}{7}t - \frac{1}{2}\Phi_w\right)\right|\right), \tag{6.28}$$

where $I^{\text{new}}_t = S_{t-1} - S_t$. [Dehning et al., 2020] assumed a Student-t distribution with four degrees of freedom and multiplicative noise for the likelihood, such that the likelihood for observed cases $\hat{C}_t$ was given by:

$$p(\hat{C}_t|\theta, \sigma) = \text{Student-t}_{\nu=4}(\text{mean} = C_t(\theta), \text{scale} = \sigma\sqrt{C_t(\theta)}),$$

where $\theta = (\lambda_0, \lambda_1, \lambda_2, \lambda_3, t_1, t_2, t_3, d_1, d_2, d_3, \mu, D, I_0, f_w, \Phi_w, \sigma)$ is the full vector of parameters for the differential equation model, and $C_t(\theta)$ is the deterministic solution which may be computed using a range of different time steps. The prior distributions for the parameters are given in Table 6.2.

Parameter	Prior
λ_0	log normal$(\log(0.4), 0.5)$
λ_1	log normal$(\log(0.2), 0.5)$
λ_2	log normal$(\log(0.125), 0.5)$
λ_3	log normal$(\log(0.0625), 0.5)$
t_1	$N(2020 \text{ March } 9, 3 \text{ days})$
t_2	$N(2020 \text{ March } 16, 1 \text{ day})$
t_3	$N(2020 \text{ March } 23, 1 \text{ day})$
d_i	log normal$(\log(3), 0.3)$
μ	log normal$(\log(0.0625), 0.2)$
D	log normal$(\log(8), 0.2)$
I_0	half Cauchy(100)
f_w	beta$(0.7, 0.17)$
Φ_w	Von-Mises$(0, 0.01)$
σ	half Cauchy(10)

Table 6.2: Prior distributions for parameters in the SIR changepoint model.

6.6.1 Effect of time step on the forward solution

We first study the effect of assuming $\Delta t = 1$ day on forward simulations of the model. We set up the forward simulations using the same settings that [Dehning et al., 2020] used to generate their Figure 2. The parameters of an SIR model without change points or weekly modulation (i.e., a single value of λ, μ, D I_0, and σ) were inferred from an early period of the German daily reported COVID-19 cases, from 2 March 2020 to 15 March 2020. The posterior median values of these parameters (excepting λ) were then used to generate forward simulations according to the full model without weekly modulation (eqs. (6.24)–(6.27)), with one change point, and pre-specified values of λ_0 and λ_1.

As in [Dehning et al., 2020], the first set of simulations considered how different levels of social restrictions could influence the course of disease transmission, as measured by cases. Three levels of social restrictions (assumed to be captured by different λ values) are considered, which each yield two sets of simulations: one corresponding to Forward Euler with $\Delta t = 1$ day (as in [Dehning et al., 2020]) and another with $\Delta t = 0.1$ days. The results of this are shown in Figure 6.10A. Our second set of simulations, shown in Figure 6.10B, considered only our "strong" social distancing scenario and explored three different times at which the change in λ might occur (e.g., if a public health intervention were implemented at different times). These simulations illustrate how, for constant values of the parameter λ, using a model with a large time step generally leads to a substantial underestimation of case counts relative to a model with a smaller time step, particularly during the (crucial) growth phase of the epidemic.

6.6.2 Effect of time step on the posterior distributions

We also studied the effect of the time step on parameter inference for the full model (eqs. (6.24)–(6.28)) using the German daily cases data from 2 March 2020 to 21 April 2020 as was done in [Dehning et al., 2020]. Inference was performed using the PyMC3 No-U-Turn MCMC Sampler (NUTS) [Salvatier et al., 2016, Gelman et al., 2013] using the model developed by [Dehning et al., 2020], modified to allow the 0.1 day step size. To initialize the chains, automatic differentiation variational inference [Kucukelbir et al., 2017] as implemented in PyMC3 [Salvatier et al., 2016] was performed to generate an approximate posterior (which, however, does not capture correlations between the parameters). Four MCMC chains were then initialized by sampling from this approximation of the posterior. The chains were run for 500 iterations of NUTS, with the first 100 discarded as burn-in, and convergence assessed by requiring that $\hat{R} < 1.05$ [Gelman et al., 2013].

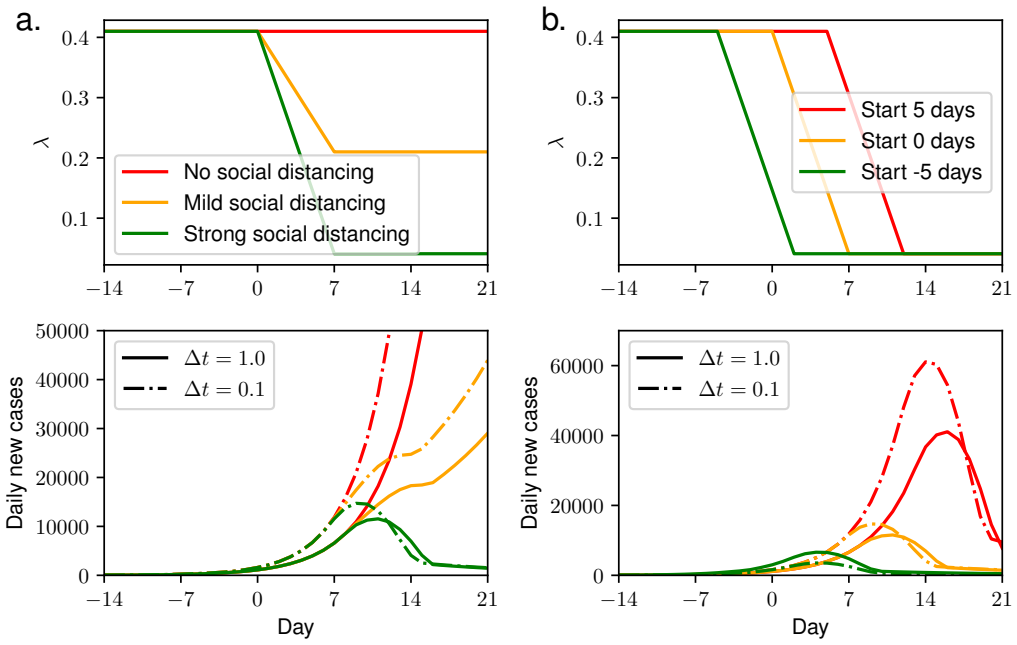


Figure 6.10: **COVID-19 model: forward simulations using Forward Euler.** In both (a) and (b), the top panel shows three different pre-specified trajectories of $\lambda(t)$, and the bottom panel shows the number of daily cases resulting from these trajectories for each choice of the time step Δt.

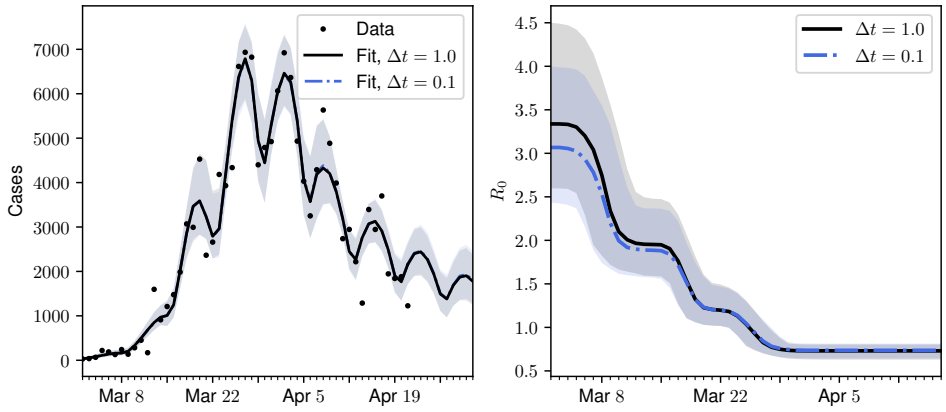


Figure 6.11: **COVID-19 model: inference using Forward Euler** (Left) Real data and model fits for the number of daily COVID-19 cases in Germany over the period 2 March 2020 to 21 April 2020. Note that the model fits for $\Delta t = 1$ and $\Delta t = 0.1$ overlap almost completely. (Right) Inferred basic reproduction number over time for the Germany COVID-19 data, using the SIR model with change points in λ (eqs. (6.24)–(6.28)) and two different values for the ODE solver time step, Δt. In both panels, lines indicate the posterior median and shaded regions indicate the central 95% of the posterior.

These results are shown in Figure 6.11.

Both models achieve a near identical visual fit to the data, using the median values of the recovered parameters. However, the parameter estimates of the two models differed. We focus on the posterior distribution for the basic reproduction number R_0, which is calculated using the MCMC samples of the joint posterior for (λ, μ). The one day time step results in overestimation of R_0 (by approximately 10% relative to the 0.1 day time step) during the early stages of the epidemic (i.e., before the first change point). This is because, during the growth phase of the epidemic, the larger time step results in slower growth for a given λ value (cf. Figure 6.10), meaning a larger λ value is estimated to compensate. During the later stages of the epidemic, the values of R_0 are more similar between the two models. Additionally, the change point locations are not much affected by the choice of time step (though, this is expected as the change points have fairly informative priors).

Our results indicate that while the discrete version of the SIR change point model using $\Delta t = 1$ appears visually to obtain a good fit to German COVID-19 data, the growth parameters of the discrete model using this time step vary markedly from those recovered using $\Delta t = 0.1$, and thus care should be taken in the deployment of such discrete models and the reporting of their results.

6.7 Numerical errors arising in rainfall-runoff models of river streamflow data

In this section we use real data from the French Broad River at Asheville, North Carolina to investigate the impact of adaptive solvers in performing inference for rainfall-runoff models used in hydrology [Schoups and Vrugt, 2010, Schoups et al., 2010].

Rainfall-runoff models divide the flow of water through a river basin into several spatially grouped components representing different hydrological processes. The model we consider here is governed by a system of five ODEs:

$$\frac{dS_i}{dt} = \text{Precip}(t) - \text{InterceptEvap}(t) - \text{EffectPrecip}(t) \tag{6.29}$$

$$\frac{dS_u}{dt} = \text{EffectPrecip}(t) - \text{UnsatEvap}(t) - \text{Percolation}(t) - \text{Runoff}(t) \tag{6.30}$$

$$\frac{dS_s}{dt} = \text{Percolation}(t) - \text{SlowStream}(t) \tag{6.31}$$

Term	Definition	Description
S_i	Interception storage	Water which strikes vegetative surfaces.
S_u	Unsaturated storage	Storage of water in the soil above the water table.
S_s	Slow reservoir	Water moving to the river via percolation.
S_f	Fast reservoir	Water moving to the river via surface runoff.
z	River discharge	Water flowed out of the river at the measuring location.
$f(S, a)$	$\frac{1-e^{-aS}}{1-e^{-a}}$	Nonlinear flux function.
Precip(t)	Precipitation	Areal precipitation in the river basin, provided as input to the model.
Evap(t)	Evaporation	Evaporation from the river basin, provided as input to the model.
InterceptEvap(t)	Evap$(t)f(S_i/I_{\text{max}}, \alpha_i)$	Evaporation from interception.
EffectPrecip(t)	Precip$(t)f(S_i/I_{\text{max}}, -\alpha_i)$	Effective precipitation which reaches unsaturated storage.
UnsatEvap(t)	$\max(0,$ Evap(t) $-$ InterceptEvap$(t))f(S_u/S_{\text{u,max}}, \alpha_e)$	Evaporation from unsaturated storage.
Percolation(t)	$Q_{\text{s,max}}f(S_u/S_{\text{u,max}}, \alpha_s)$	Trickling of water through the ground.
Runoff(t)	EffectPrecip$(t)f(S_u/S_{\text{u,max}}, \alpha_f)$	Flow of water on the surface.
SlowStream(t)	S_s/K_s	Slow component of the river flow.
FastStream(t)	S_f/K_f	Fast component of the river flow.

Table 6.3: Description of the terms which appear in the rainfall-runoff model.

$$\frac{dS_f}{dt} = \text{Runoff}(t) - \text{FastStream}(t) \tag{6.32}$$

$$\frac{dz}{dt} = \text{SlowStream}(t) + \text{FastStream}(t), \tag{6.33}$$

Each term in this equation is defined in Table 6.3, and the seven unknown parameters of the model and their prior distributions are defined in Table 6.4. The data consist of daily streamflow measurements (dz/dt), and the authors assume an IID Gaussian likelihood with unknown standard deviation σ.

Parameter	Definition	Prior
I_{max}	Maximum interception	Uniform$(0, 10)$
$S_{\text{u,max}}$	Unsaturated storage capacity	Uniform$(10, 1000)$
$Q_{\text{s,max}}$	Maximum percolation	Uniform$(0, 100)$
α_e	Evaporation flux shape	Uniform$(0, 100)$
α_f	Runoff flux shape	Uniform$(-10, 10)$
K_s	Slow reservoir time constant	Uniform$(0, 150)$
K_F	Fast reservoir time constant	Uniform$(0, 10)$
$\alpha_s = 0$	Percolation flux shape	-
$\alpha_i = 50$	Interception flux shape	-
σ	Noise standard deviation	Uniform(0, 10)

Table 6.4: Description of the seven unknown parameters of the model, and the two parameters with fixed values.

Previous work has shown that using large time steps with such hydrological models can bias inferences [Kavetski et al., 2003]. We show that using an adaptive step size method (as suggested by [Schoups et al., 2010]) can also cause inaccurate inference results, unless the error is tightly controlled.

Using a fast and accurate ODE solver (the CVODE multistep solver from the SUNDIALS library [Hindmarsh et al., 2005] with $\texttt{rtol} = \texttt{atol} = 10^{-7}$), we obtained the posterior distributions for the seven parameters of the model, using USGS data for the streamflow at Asheville, North Carolina (USGS station 03451500) over a 200 day period starting 1 January 1960. Sampling was performed using the Dream multi-chain MCMC algorithm as implemented in PINTS [Vrugt et al., 2009, Clerx et al., 2019], using 6 chains with each initialized by a sample from the prior (Table S3, supplementary information). 25000 MCMC iterations were performed, and convergence of the chains was assessed by requiring that $\hat{R} < 1.05$ [Gelman et al., 2013]. In Figure 6.12, we plot the likelihood surfaces of the parameters for slices through parameter space near the estimated posterior medians. Likelihood surfaces are plotted for two adaptive step size solvers: the RK3(2) solver from SciPy with $\texttt{rtol} = \texttt{atol} = 10^{-3}$, and the CVODE solver as described above. For all parameters, the 10^{-3} tolerance solver causes highly jagged likelihoods, of sufficient magnitude to interfere with inference via MCMC or maximum likelihood estimation. This is in accordance with our earlier results using the oscillator model in §6.4, as rapid changes in the RHS cause spurious jaggedness in the computed likelihood. The likelihoods calculated using the more accurate solver have

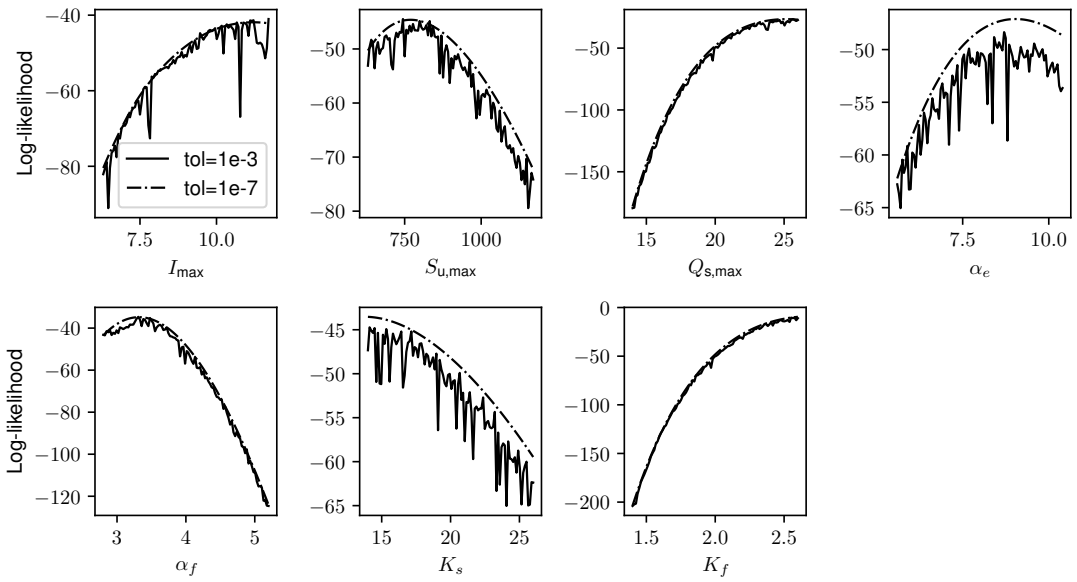


Figure 6.12: **Rainfall run-off model: inference using accurate and inaccurate adaptive solvers.** Here, we plot the likelihood surface for each parameter for the rainfall-runoff model defined by eqs. (6.29)–(6.33). For each parameter, the solid line indicates the likelihood calculated using an RK3(2) adaptive solver with $\texttt{rtol} = \texttt{atol} = 10^{-3}$, while the dashed line indicates the likelihood calculated using the CVODE adaptive solver with $\texttt{rtol} = \texttt{atol} = 10^{-7}$.

similar broadscale shapes but are smooth enough for accurate inference to be performed.

6.8 Discussion

Inaccurate solution of ODEs through either fixed time step or adaptive solvers can lead to biased inferences which are generally exacerbated when there is low observation noise. For adaptive solvers, these biases may manifest through the presence of phantom jaggedness in the likelihood surface, which can wreak havoc for inference algorithms attempting to navigate the surface. For the oscillator model studied in this chapter, the three model parameters could be precisely learned when the ODE was solved accurately, but when a solver with insufficient tolerance was used MCMC chains failed to mix (Figure 6.8). Our results indicate that, when applying adaptive solver grids to inference, sufficient accuracy of the numerics is a prerequisite for intelligible inference results. Researchers facing intractable likelihood surfaces or objective functions arising due to numerical inaccuracy in an adaptive grid solver may be motivated to modify the model in arbitrary ways when, in fact, all that was required to render inference soluble was a reduction in solver tolerances. Tolerances which seem good enough for forward simulation are likely insufficient for solving inverse problems. For example, a relative tolerance of 10^{-3} was insufficient for both the synthetic data and real data studied in §6.4.2 and §6.7. When using an ODE solver library to perform inference, default settings may well not suffice and, ideally, the solver tolerance should be set by inspection of the likelihood surface.

In inference problems involving ODEs, numerical solution of the differential equations at each parameter value is likely to be the dominant computational expense. For this reason, researchers may be motivated to implement coarser solvers to decrease runtimes. When using fixed solver grids, the results in this chapter suggest that the parameter values associated with models using larger time steps cannot be considered equivalent to those associated with finer time steps. Using adaptive solver grids may enable significant speedups over fixed grid methods. However, in this chapter we observed highly intractable likelihood surfaces computed using adaptive solvers with moderate tolerances, arising from the fact that these solvers use different solver grids at nearby parameter values. For this reason, when using adaptive time step solvers for inference researchers must ensure that their computational budget allows for a sufficiently refined tolerance that their likelihood surfaces are tractable for inference.

Unless there is a bifurcation in system behaviour at points in parameter space,

likelihood surfaces should not have abrupt discontinuities. So, the presence of such changes may well be an artefact of using an adaptive ODE solver with insufficient tolerances. MCMC and optimisation algorithms could be augmented by monitoring for such jumps and warning the user should they occur.

ODEs involving discontinuous RHS functions are known to be particularly challenging to solve accurately. Our results indicate that RHS functions involving rapid changes over time, such as those involving discontinuities, also curse computational inference when adaptive ODE solvers are used. However, our results in §6.5 also indicate that errors in the likelihood arising from discontinuous RHS functions can be ameliorated through the use of simple smoothing approximations—a potentially more computationally efficient alternative to increasing tolerances. We argue that in many scientific systems such smoothing approximations are additionally more realistic descriptions of the phenomena being modelled. Although the appropriate degree of smoothing may be difficult to determine in general, for certain systems, the level of smoothing can be tuned based on knowledge of the process being modelled.

Much of the work on error control for ODE solvers has focused merely on the accuracy of the forward problem. The accuracies of widely used ODE solvers are typically tuned via their step sizes or local truncation error tolerances, but these are not the most relevant quantities for inference—instead, it is the error in the log-likelihood which must be controlled. ODE solvers which control the error on the log-likelihood directly would avoid much of the problems highlighted in this chapter, and we suggest this as a fruitful research direction.

6.9 Data and software

The code to perform the computer experiments presented in this chapter was written in Python 3.7 and is available in an open source Python library at `https://github.com/rccreswell/ode_inference`. To run the COVID-19 simulations, we adapted the software library developed by [Dehning et al., 2020]. The version of the code including our modifications is available at `https://github.com/rccreswell/covid19_inference_forecast`.

Chapter 7

Enhancing gradient-based inference using the adjoint

Overview

Our results in Chapter 6 indicate the importance of controlling the error on the likelihood during parameter inference for models involving numerically approximated differential equations; however, existing approaches for solving ordinary differential equations tend to merely control the local truncation error on the model solution and not the error which is relevant for accurate inference (namely, the error in the likelihood or objective function). In this chapter, our goal is to develop a technique for controlling error on the log-likelihood while performing inference for differential equation models. Performing inference for large, multi-parameter differential equation models presents an additional challenge: the most efficient inference algorithms require the gradient of the log-posterior, but this gradient is expensive to approximate. We propose drawing upon *adjoint*-based methods to simultaneously address both of these challenges. We show how the adjoint equation to an ODE system can be used to simultaneously compute the error in the log-likelihood arising due to numerical approximation of an underlying ODE model as well as compute gradients of the log-likelihood with respect to the parameters of the model. To motivate the proposed approach, we show how controlling error on the log-likelihood via methods such as those discussed in the chapter bounds the Bayes factor between the numerically approximated model and the true model.

Publications

This chapter is derived from the unpublished working paper:

- **R. Creswell,** M. Robinson, B. Lambert, C. L. Lei, D. J. Gavaghan, and S. Tavener: "Enhancing gradient-based Bayesian inference for initial value problems using the adjoint, " (Unpublished working paper). [Creswell et al., 2023b]

  This manuscript is currently under preparation for eventual submission to *Bayesian Analysis*.

  Contributions: I designed and performed the numerical experiments, and visualised and interpreted the results. I derived the bound on the expected absolute Bayes factor as a function of error in the log-likelihood in collaboration with Simon Tavener. The derivations of the adjoint-based approximations to the error in the likelihood and the gradient were performed by Simon Tavener; because these quantities are employed in the proposed inference strategy, this chapter includes a rewriting of his derivations. Martin Robinson developed a Python library for solving the adjoint equation and using the error estimate and gradient for parameter optimisation according to the methods discussed in this chapter; I relied on parts of this library while developing my software implementation. All authors made contributions and suggestions to the writing and revision of the working paper cited above, and some of these contributions are reflected in the wording of parts of this chapter.

7.1 Introduction

In Chapter 6, we analysed the interplay between ODE solvers and inference problems. Existing adaptive time step solvers of the sort studied in that chapter tune the solver grid in order to control an estimate of error in the local truncation error. However, in inference problems, we showed that the relevant quantity whose error must be kept at an acceptably small level is not the local truncation error but rather the log-posterior density or log-likelihood. Thus, our results in Chapter 6 demonstrated the significant biases that can appear in computations of the log-likelihood and Bayesian inference algorithms when solver tolerances appear good enough for forward simulation, but are not good enough for inference.

In this chapter, we study methods by which the numerical error in the log-likelihood arising from numerical approximation of the underlying ODEs can be controlled. Our

goal is thus to place a tolerance directly on the error in the log-likelihood (or other relevant quantity of interest for inference) and ensure that the ODE is solved sufficiently accurately that this tolerance is not exceeded. Applying these methods involves significant extra computational expense—however, we show how they can simultaneously yield the gradient of the log-likelihood with respect to the ODE model parameters, enabling faster gradient-based sampling or optimisation algorithms.

In the remainder of this section (§7.1), we briefly review relevant work on inference for ODEs. Subsequently, in §7.2, we introduce the adjoint equation to an ODE system and show how it can be used to estimate the error in a quantity of interest (some functional of the ODE solution, such as a likelihood or objective function) arising due to numerical approximation of the underlying ODE. Next, in §7.3, we show how bounding the error in the log-likelihood also bounds the Bayes factor between the true and numerically approximate posterior, motivating the methods described in §7.2. Finally, in §7.4, we present empirical results using the methods discussed in the chapter.

7.1.1 Inverse problems and numerical error

Inference via optimisation or MCMC requires many solutions of the forward problem at different values of the parameters θ. Each forward problem must be solved numerically with sufficient accuracy so that the algorithms used to explore the parameter space are not biased by numerical error. The impact of numerical errors on parameter inference has motivated the development of a variety of methods for performing inference for ODEs, which we briefly review here.

Several of these methods fall under the field of *probabilistic numerics* (PN) [Hennig et al., 2015], which concerns the development of numerical methods in which uncertainty is treated in a probabilistic manner. For example, in the method developed by [Chkrebtii et al., 2016], Gaussian distributed error is assumed for the deviation between the true model and its numerical approximation at each time step; the solution to the differential equation is modelled as unknown with a Gaussian process prior. Another PN method for handling numerical uncertainty in inverse problems involving differential equations involves modifying deterministic differential equation solvers to include random terms (for example, adding an IID Gaussian term to the solution at each solver time step [Conrad et al., 2017]); the differential equation solutions then obey stochastic differential equations (SDEs) which have been shown to converge to the original differential equations under appropriate limits [Conrad et al., 2017, Teymur et al., 2016, Teymur et al., 2018].

An alternative approach employs importance sampling in an attempt to correct inaccuracies in the posterior samples resulting from numerical approximation of the forward model [Timonen et al., 2022]. In this approach, MCMC samples are first obtained using a faster solver which may introduce some numerical bias. Once the set of biased samples are obtained, each is reweighted using the ratio of the target densities computed under two numerical solvers: one faster and less accurate (the one used in the MCMC algorithm), and one slower and more accurate (used only in the calculation of the importance weights), which offers a significant speed up relative to having to use the accurate solver at all parameter values proposed by the MCMC algorithm. [Timonen et al., 2022] additionally propose heuristic strategies using the observed distribution of the importance weights by which sufficient accuracy of the slower solver may be ensured.

A recent line of inquiry, which we employ in this chapter, focuses on the use of Bayes factors to characterize the effects of numerical error in the ODE solution on the posterior distribution. It considers the Bayes factor between the hypothetical "true" model which assumes the ODE is solved exactly, and the "approximate" model assuming the ODE is solved numerically with finite accuracy. When this Bayes factor is close to one, the numerical approximation can be considered sufficiently accurate for inference [Capistrán et al., 2016]. This approach has been generalized to PDEs and to include discretization error on the prior [Christen et al., 2017, Capistrán et al., 2022, Daza-Torres et al., 2021]. In particular, [Capistrán et al., 2022] bounds the error in the forward solver that can be tolerated while keeping the Bayes factor close to one, and shows that the Bayes factor tends towards one at the same order as that of the numerical solver. Later in this chapter (§7.3), we adopt a similar approach by considering the Bayes factor between the true and numerical models, but by using the adjoint-based *a posteriori* methods to estimate the numerical error in the log-likelihood function that we discuss in §7.2, we are able to obtain a simpler bound on the error that is required to keep the Bayes factor close to one.

7.1.2 Gradient-based inference methods

ODE inference problems of interest in scientific applications typically have multiple parameters, and present high-dimensional, non-convex likelihood surfaces. Such surfaces are difficult for inference algorithms to explore without using information about the gradient of the likelihood with respect to the parameters θ. Gradient information can be integrated into MCMC samplers in order to guide the generation of more efficient

proposals.

A standard gradient-based sampler is Hamiltonian Monte Carlo (HMC) [Neal, 2011]. HMC augments the parameters to be learned θ with an auxiliary "momentum" variable. At each iteration, the momentum variable is updated according to a random walk, and then the parameters θ and momentum are updated jointly according to approximate Hamiltonian dynamics with the negative log posterior treated as a potential energy term. Symplectic integration for HMC has typically employed the leapfrog method [Gelman et al., 2013]. Although HMC is often used in problems where exact analytical gradients can be calculated, the algorithm has also been used with approximate gradients [Chen et al., 2014, Li et al., 2019].

In order to avoid having to tune or adapt the hyperparameters governing HMC (the number of steps and the step size of the symplectic integration), a more sophisticated sampler, the No-U-Turn sampler (NUTS), has been developed [Gelman et al., 2013]. At each MCMC iteration, NUTS sets the number of leapfrog steps in order to avoid inefficient "U-Turns" (where the proposal starts to turn around towards the starting point) and tunes the step size in order to achieve a certain MCMC acceptance ratio.

In ODE problems, analytical gradients are rarely available and numerical approximations to the gradient via finite differences may be prohibitively slow due to the high dimensionality of Θ and the high computational cost of each solution to the forward problem. However, the gradient of a functional of the solution to an ODE with respect to its parameters may also be obtained by solving the adjoint equation as we describe in §7.2.3. [Melicher et al., 2017] shows how to use the adjoint method to obtain gradients, which may be used for Bayesian inference or optimisation. Adjoint-based *a posteriori* error analysis is a well-established technique ([Ainsworth and Oden, 2000, Bangerth and Rannacher, 2003, Barth, 2004, Becker and Rannacher, 2001, Eriksson et al., 1995, Estep, 1995, Giles and Süli, 2002]); gradient based optimisation and Bayesian inference using the adjoint-derived approximate gradient has found application in epidemiology [Kabanikhin and Krivorotko, 2020] and neural mass models [Sengupta et al., 2016], amongst other fields.

7.2 Two applications of the adjoint equation

In this section, we aim to derive the adjoint equation to an ODE. Using the adjoint equation, we show to simultaneously estimate the error in a functional of the ODE

solution arising due to numerical approximation of the solution, and the gradient of the same functional with respect to the ODE's unknown parameters.

7.2.1 Error estimation for a functional of the numerical solution to an ODE

As in Chapter 6, we consider problems of the form:

$$\begin{aligned} \frac{dx}{dt} &= h(t, x, \theta), \qquad t \in (0, T], \\ x(0; \theta) &= x_0, \end{aligned} \tag{7.1}$$

where $x \in \mathbb{R}^n$, $\theta \in \Theta \subset \mathbb{R}^m$ and $h : (0, T] \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. We assume that $x(t; \theta)$ cannot be exactly determined, and is instead approximated at a set of discrete solver grid points, denoted by $\alpha = (\bar{t}^{(1)}, \ldots, \bar{t}^{(L)})$.

For solutions to the ODE (eq. (7.1)), we begin by defining a nonlinear functional $Q(x)$ or "quantity of interest" of the solution in terms of an integral over [0,T], i.e.,

$$Q(x) = \int_0^T q(x)\ dt\ , \tag{7.2}$$

for some $q : \mathbb{R}^n \to \mathbb{R}$.

Let $\hat{x}$ be an approximate solution to the ODE eq. (7.1), and define the error

$$e(t) = x(t) - \hat{x}(t)\ , \tag{7.3}$$

Our goal is to approximate e_Q, the error in Q corresponding to the use of the approximate solution. We use the first order Taylor expansion of $Q(x)$ about $\hat{x}$ to obtain an expression for e_Q in terms of e and dq/dx, i.e.,

$$e_Q = Q(x) - Q(\hat{x}) \approx \frac{dQ}{dx_i}(x_i - \hat{x}_i) = \int_0^T e_i \frac{\partial q}{\partial x_i}\ dt = \int_0^T \left(e, \frac{\partial q}{\partial x}\right)\ dt \tag{7.4}$$

where summation over repeated indices is implied, $(\cdot, \cdot)$ is the inner product and all derivatives are evaluated at solution value $\hat{x}$ and parameter values θ.

We define the adjoint $\phi \in \mathbb{R}^n$ as the solution to the (backwards) differential equation

$$\begin{aligned} -\dot{\phi} - \frac{\partial h}{\partial x}^\top \phi &= \frac{\partial q}{\partial x}, \qquad t \in (T, 0]\ , \\ \phi(T) &= 0, \end{aligned} \tag{7.5}$$

where all derivatives are evaluated at $(\hat{x}, \theta)$ and $\dfrac{\partial h}{\partial x}$ denotes the matrix with entries $\dfrac{\partial h_i}{\partial x_j},\ i, j = 1, \ldots, n.$

From (7.4) and (7.5), using integration by parts and defining the residual of the ODE as

$$R(t) = h(t, \hat{x}, \theta) - \dot{\hat{x}}, \tag{7.6}$$

we now have

$$\begin{aligned} e_Q &= \int_0^T \left(e, \frac{\partial q}{\partial x} \right)\ dt \\ &= \int_0^T \left(e, -\dot{\phi} - \frac{\partial h}{\partial x}^\top \phi \right)\ dt \\ &= -[(e, \phi)]_0^T + \int_0^T (\dot{e}, \phi) - \left(e, \frac{\partial h}{\partial x}^\top \phi \right)\ dt \\ &= e(0)\phi(0) + \int_0^T \left(\dot{e} - \frac{\partial h}{\partial x} e, \phi \right)\ dt \\ &= e(0)\phi(0) + \int_0^T (R, \phi)\ dt \end{aligned} \tag{7.7}$$

where we have used the fact that

$$\dot{e} = \dot{x} - \dot{\hat{x}} = h(x) - \dot{\hat{x}} \approx h(\hat{x}) + \frac{\partial h}{\partial x} e - \dot{\hat{x}} = R + \frac{\partial h}{\partial x} e.$$

7.2.2 Accuracy of the adjoint-based error estimate

The adjoint-based estimate of the error in a quantity of interest depending on the solution, eq. (7.7), itself involves several approximations: firstly, the first order Taylor expansion of $h(x)$ about $\hat{x}$ (i.e., we neglect second order and higher terms of $(x - \hat{x})$) and similarly of $Q(x)$ about $\hat{x}$, and secondly, the adjoint state ϕ must also be obtained using a numerical solver. Finally, computation of the integral in eq. (7.7) may also introduce error if this must be done using approximate numerical integration. In this section we comment on the relative importance of these sources of error on the practical application of the adjoint-based error estimate to inference problems.

Letting $e_{Q,\text{True}}$ denote the true error in Q and e_Q denote the approximation to the error as derived in eq. (7.7), we consider the *effectivity ratio*, given by:

$$\frac{e_Q}{e_{Q,\text{True}}}.$$

When the effectivity ratio equals one, the estimated error is completely accurate, while

values of the effectivity ratio above or below one indicate that the error in the quantity of interest is being underestimated or overestimated by the adjoint method.

For simplicity, we assume a uniform mesh with spacing Δt. Suppose that $e_{Q,\text{True}} = c_1 \Delta t$, i.e., the actual error in the quantity of interest converges to zero at the same rate as the step size of the solution grid. If the accuracy of the estimated error were to also converge at the same rate ($e_Q - e_{Q,\text{True}} = c_2 \Delta t$), we would have for the effectivity ratio:

$$\frac{e_Q}{e_{Q,\text{True}}} = \frac{c_1 \Delta t + c_2 \Delta t}{c_1 \Delta t} = 1 + \frac{c_2}{c_1},$$

i.e., the effectivity ratio fails to converge to 1 for any step size, and even for highly accurate grids the adjoint-based error estimate will suffer from some inaccuracy. Conversely, supposing that the estimated error converges at a higher rate, $e_Q - e_{Q,\text{True}} = c_3 \Delta t^2$, we have for the effectivity ratio:

$$\frac{e_Q}{e_{Q,\text{True}}} = 1 + \frac{c_3 \Delta t}{c_1},$$

which converges to 1 as $\Delta t \to 0$.

In practice, the rate of convergence of the error and its estimate may be more complicated; however, this argument motivates solving the adjoint equation using a higher order solver than the forwards model.

Furthermore, because second order and higher terms of $(x - \hat{x})$ are neglected, the adjoint-based error estimate may not be appropriate for use when the solver grid is so poor that x and $\hat{x}$ deviate from each other drastically.

7.2.3 Gradient calculation for a functional of the solution to an ODE

We can take a similar approach to that taken in §7.2.1 for obtaining efficient estimates of the gradient of Q with respect to the parameters θ, by again making use of the adjoint problem. Here we begin by considering a small perturbation to the parameters φ which induces a change in the solution, z. The resulting solution $x + z$ satisfies the perturbed ODE

$$\begin{aligned} \frac{d}{dt}(x+z) &= h(t, x+z, \theta+\varphi), \qquad t \in (0, T], \\ (x+z)(0) &= x_0 + z_0. \end{aligned}$$

Expanding h as a Taylor series, we find that the perturbation $z(t)$ satisfies

$$\begin{aligned} \frac{dz}{dt} &= \frac{\partial h}{\partial x} z + \frac{\partial h}{\partial \theta} \varphi \qquad t \in (0, T], \\ z(0) &= z_0, \end{aligned} \tag{7.8}$$

where $\dfrac{\partial h}{\partial \theta}$ is the matrix with entries given by $\dfrac{\partial h_i}{\partial \theta_j},\ i = 1, \ldots, n,\ j = 1, \ldots, m$.

The corresponding change in the quantity of interest is

$$Q(x+z) \approx Q(x) + \frac{\partial Q}{\partial x_i} z_i = Q(x) + \int_0^T \left(z, \frac{\partial q}{\partial x} \right) \, dt.$$

Hence, employing integration by parts and (7.8)

$$\begin{aligned} Q(x+z) - Q(x) &= \int_0^T \left(z, \frac{\partial q}{\partial x} \right) \, dt \\ &= \int_0^T \left(z, \dot{\phi} - \frac{\partial h}{\partial x}^T \phi \right) \, dt \\ &= [(z, \phi)]_0^T + \int_0^T \left(\dot{z} - \frac{\partial h}{\partial x} z, \phi \right) \, dt \\ &= (z_0, \phi(0)) + \int_0^T \left(\frac{\partial h}{\partial \theta} \varphi, \phi \right) \, dt \end{aligned}$$

or

$$\frac{\partial Q}{\partial x_0} = \phi(0) \quad \text{and} \quad \frac{\partial Q}{\partial \theta_j} = \int_0^T \left(\frac{\partial h}{\partial \theta_j}, \phi \right) \, dt. \tag{7.9}$$

In the derivation of the gradient, we have used the same adjoint state ϕ whose backwards ODE was derived in §7.2.1. Thus, although solving for ϕ does involve extra computational expense, we observe that, once obtained, it can be used twice: once to obtain the error estimate, and once to obtain the gradient. We aim to maximise the utility of the adjoint solution and the associated cost of constructing and solving an adjoint problem to compute both gradients and errors.

7.3 Bounding the Expected Absolute Bayes' Factor

The adjoint methods described in the previous section provide an estimate of the error in a functional of the ODE solution (such as the log-likelihood) arising from error in the numerical solver. In this section, we show how such a bound on the error in the log-likelihood also bounds the Bayes factor between the true and numerically approximate posteriors.

Our results in this section are a variation of the approach of [Capistrán et al., 2022]. In our approach, rather than assuming a bound on the error in the forward solution, we assume that the error in the log-likelihood can be controlled (e.g., via the adjoint-based method presented above in §7.2.1). Furthermore, while [Capistrán et al., 2022] considers

error in the computation of the prior distribution, in our approach we assume that the prior is computed with perfect accuracy and focus exclusively on numerical error arising from the likelihood.

7.3.1 Definition of the EABF

Let Θ be the parameter space and V the space of outputs of the forward model. We define the forward map (FM)

$$\mathcal{F} : \Theta \to V \ , \tag{7.10}$$

and the *observation operator*

$$\mathcal{H} : V \to \mathcal{Y} \ . \tag{7.11}$$

The likelihood is given by $L(\tilde{y}|\theta)$, where $\tilde{y} \in V$ is the observed data and $\theta \in \Theta$ is the parameter value. In practice, the forward map $\mathcal{F}$ is solved using a numerical approximation on a grid of solver points $\alpha = (\bar{t}^{(1)}, \dots, \bar{t}^{(L)})$. We indicate the likelihood computed using the numerical approximation to the forward map on the mesh α with a superscript α.

The Bayes factor, or the ratio of the marginal probability density computed according to each of two candidate models, is a widely used quantity for Bayesian model comparison [Gelman et al., 2013]. Our goal is to calculate the Bayes factor between the true model and the model relying on the numerical approximation to the forward map. Letting π indicate the prior measure on the parameters, which we assume is normalised, the marginal probability densities are given by

$$Z(y) = \int_\Theta L(y|\theta) \ \pi(d\theta) \tag{7.12}$$

for the true model, and

$$Z^\alpha(y) = \int_\Theta L^\alpha(y|\theta) \ \pi(d\theta) \tag{7.13}$$

for the model involving a numerical approximation, while the Bayes factor (BF) is given by

$$\text{BF} = \frac{Z^\alpha(y)}{Z(y)} \ .$$

Keeping the Bayes factor close to 1 ensures that the models do not differ significantly. In order to study the convergence of the Bayes factor, we introduce the Absolute Bayes factor (ABF)

$$\text{ABF}(y) = \left| \frac{Z^\alpha(y)}{Z(y)} - 1 \right| \tag{7.14}$$

which we wish to approach zero for the models to not differ significantly.

The ABF depends on the data $\tilde{y}$, which we assume falls in the observation space, $\tilde{y} \in \mathcal{Y}$. To avoid dependence on a particular value of the data, we compute the expectation of the ABF given the *distribution of the data* under the true model. This quantity is called the Expected Absolute Bayes factor (EABF). Note that the density of the data $\tilde{y}$ under the true model is simply $Z(\tilde{y})$ with respect to some measure λ (e.g., for continuous data, the Lebesgue measure); thus, we have for the EABF

$$\begin{aligned} \text{EABF} &= \int_{\mathcal{Y}} \text{ABF}(y) Z(y)\ \lambda(dy) \\ &= \int_{\mathcal{Y}} \left| \frac{Z^{\alpha}(y)}{Z(y)} - 1 \right| Z(y)\ \lambda(dy) \\ &= \int_{\mathcal{Y}} |Z^{\alpha}(y) - Z(y)|\ \lambda(dy). \end{aligned} \tag{7.15}$$

As used above, $L(\tilde{y}|\theta)$ indicates the likelihood of data $\tilde{y}$ given a value of the parameters θ. We also introduce the notation $L_o(\tilde{y}|\eta)$, indicating the likelihood of data $\tilde{y}$ given a vector of observations based on the solution to the ODE, η. That is,

$$L(\tilde{y}|\theta) = L_o(\tilde{y}|\mathcal{H}(\mathcal{F}(\theta)))\ . \tag{7.16}$$

We consider likelihoods which are *normalised when viewed as a function of the data*, i.e.,

$$\int_{\mathcal{Y}} L_o(\tilde{y}|\eta)\ \lambda(d\tilde{y}) = 1. \tag{7.17}$$

7.3.2 Bounding the EABF based on error in the solution

Traditional approaches to the solution of differential equations, such as those employed in Chapter 6, control error on the solution itself. [Capistrán et al., 2022] represent this by assuming that the error in the forward model obeys

$$||\mathcal{H}(\mathcal{F}(\theta)) - \mathcal{H}(\mathcal{F}^{\alpha}(\theta))|| < C_0 \langle \alpha \rangle^p\ , \tag{7.18}$$

for some positive order p, where $\langle \alpha \rangle$ indicates some functional of the mesh α. They consider those likelihoods which can be written in the form

$$L_o(\tilde{y}|\eta) = \prod_{k=1}^{K} \frac{1}{\sigma}\ \rho\left(\frac{\tilde{y}^{(k)} - \eta^{(k)}}{\sigma} \right)\ ,$$

where ρ is some continuous function, $\sigma > 0$ is the noise scale, and $\eta^{(k)}$ are the expected values of each data point. This assumption encompasses many standard likelihoods used in time series inference, including those based on IID Gaussian. For such likelihoods, in order to achieve EABF $< b$, [Christen et al., 2017, Capistrán et al., 2022] show that the following condition must be satisfied,

$$C = C_0|\alpha|^p < \frac{2\sigma}{K}\frac{b}{\rho(0)} \ .$$

7.3.3 Bounding the EABF based on error in the log-likelihood

In our approach, we assume an adjoint-based adaptive strategy that ensures:

$$\begin{aligned} |\log L(\tilde{y}|\theta) - \log L^\alpha(\tilde{y}|\theta)| < b \quad &\text{for all} \quad \theta \in \Theta,\ \tilde{y} \in \mathcal{Y}, \\ \Rightarrow \left|\log \frac{L(\tilde{y}|\theta)}{L^\alpha(\tilde{y}|\theta)}\right| < b \quad &\text{for all} \quad \theta \in \Theta,\ \tilde{y} \in \mathcal{Y}, \\ \Rightarrow \left|\frac{L(\tilde{y}|\theta)}{L^\alpha(\tilde{y}|\theta)} - 1\right| < b \quad &\text{for all} \quad \theta \in \Theta,\ \tilde{y} \in \mathcal{Y}, \end{aligned} \tag{7.19}$$

for $L(\tilde{y}|\theta) \approx L^\alpha(\tilde{y}|\theta)$ (the last line of eq. (7.19) depends on the approximation $\log(1+x) \approx x$ for small x; see §7.2.2). From (7.12), (7.13) and (7.19), we have

$$\begin{aligned} |Z^\alpha(\tilde{y}) - Z(\tilde{y})| &= \left| \int_\Theta \left(L(\tilde{y}|\theta) - L^\alpha(\tilde{y}|\theta)\right)\ \pi(d\theta) \right| \\ &\leq \int_\Theta |L(\tilde{y}|\theta) - L^\alpha(\tilde{y}|\theta)|\ \pi(d\theta) \\ &= \int_\Theta \left| L(\tilde{y}|\theta) \left(\frac{L^\alpha(\tilde{y}|\theta)}{L(\tilde{y}|\theta)} - 1 \right) \right|\ \pi(d\theta) \\ &< b \int_\Theta |L(\tilde{y}|\theta)|\ \pi(d\theta) \quad \text{for all} \quad \tilde{y} \in \mathcal{Y}. \end{aligned}$$

The EABF (7.15) is then bounded as

$$\text{EABF} < \int_\mathcal{Y} \left(b \int_\Theta L(\tilde{y}|\theta)\ \pi(d\theta) \right)\ \lambda(d\tilde{y}) = b \int_\mathcal{Y} \int_\Theta L(\tilde{y}|\theta)\ \pi(d\theta)\ \lambda(d\tilde{y}) \ .$$

Reversing the order of integration,

$$\text{EABF} < b \int_\Theta \int_\mathcal{Y} L_o(\tilde{y}|\mathcal{H}(\mathcal{F}(\theta)))\ \lambda(d\tilde{y})\ \pi(d\theta).$$

Recalling from (7.17) that the likelihood viewed as a function of the data is normalised, we have

$$\text{EABF} < b \int_{\Theta} \pi(d\theta) = b \; . \tag{7.20}$$

7.4 Results

7.4.1 Controlling the numerical error in the log-likelihood

Our first result is an empirical investigation of the relationship between the solver mesh step size, the error in the likelihood, and the order of the solver. These results are shown in Figure 7.1.

Three different synthetic ODE systems were studied. The first is the logistic growth model:

$$\begin{aligned} \dot{x} &= Rx(1 - x/K) \; , \qquad t \in (0, T] \; , \\ x(0) &= x_0 \; . \end{aligned} \tag{7.21}$$

where the parameter $R > 0$ is the growth rate and the parameter $K > 0$ is the carrying capacity and x_0 is the initial population size. We set the true parameter values to $R = 1, K = 1$ and $x_0 = 0.1$.

We additionally studied the damped oscillator model, which is given by:

$$\begin{aligned} \ddot{x} + k\dot{x} + cx &= F(t) \; , \qquad t \in (0, T] \; , \\ x(0) &= x_0 \; , \\ \dot{x}(0) &= \dot{x}_0 \; , \end{aligned} \tag{7.22}$$

where x is the position, $k > 0$ is the damping constant, $c > 0$ is the spring constant, and $F(t)$ is the forcing function. Rewriting as a first order system

$$\frac{d}{dt} \begin{pmatrix} x \\ \dot{x} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ -k\dot{x} - cx + F(t) \end{pmatrix} \; , \quad \begin{pmatrix} x(0) \\ \dot{x}(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ \dot{x}_0 \end{pmatrix} \; .$$

We set the true values of the parameters to be $c = 1, k = 2$ and $x_0 = 0.1, \dot{x}_0 = 2$.

For the unforced oscillator, $F(t) = 0$. For the forced oscillator, we choose

$$F(t) = \begin{cases} 0 & t < 4, \\ -20 & 2 \leq t < 4, \\ 0 & 4 \leq t. \end{cases} \tag{7.23}$$

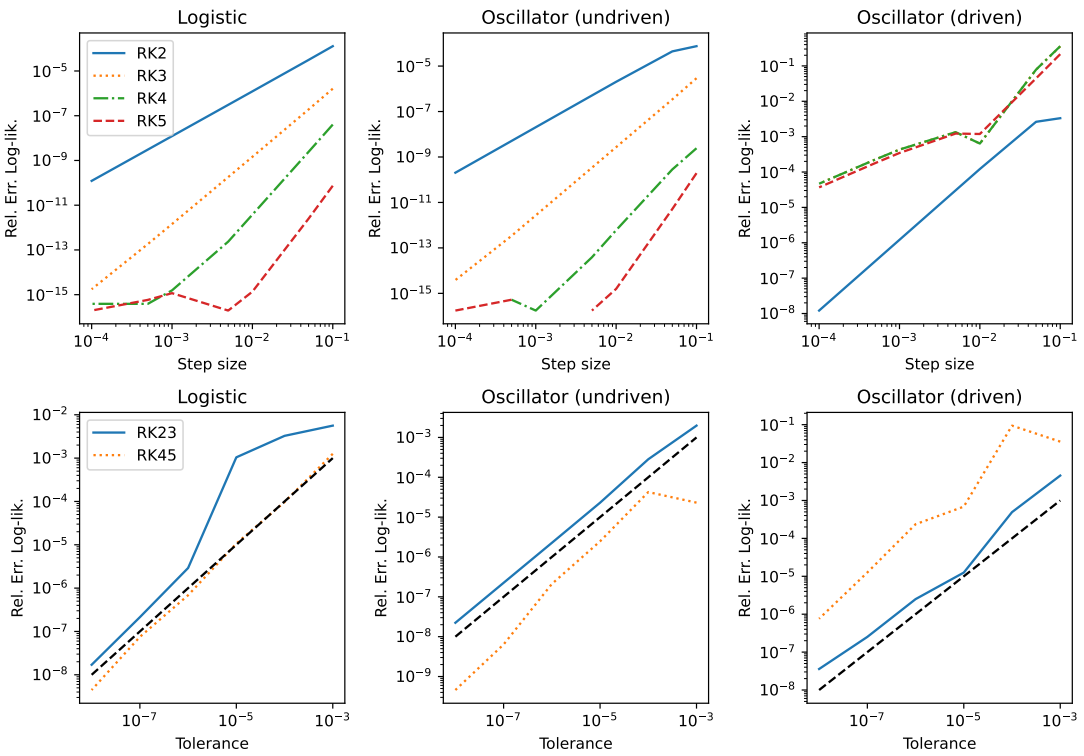


Figure 7.1: Relative error in the log-likelihood as a function of step size (top) and tolerance for the local truncation error (bottom), for the logistic model (7.21), the unforced oscillator model (7.22) and the forced oscillator (7.22). Log-likelihoods were evaluated at the true parameter values. RKN indicates the Runge-Kutta method of order N; RKMN indicates the adaptive step size Runge-Kutta method of order N where local truncation error is approximated assuming accuracy of the method of order M. In the bottom panels, the dotted black line indicates Relative error=Tolerance.

For each model, synthetic data was generated using 101 data points uniformly spaced between $t = 0$ and $t = 10$, with IID Gaussian noise of standard deviation 0.05. We first employed a uniformly spaced solver mesh of various step sizes (Figure 7.1, top). At each solver mesh size, the numerical solution at the true parameter values was obtained using a Runge-Kutta method of order n, for $n = 2, 3, 4, 5$. In order to calculate the error in the likelihood, we also require a "true" solution. The logistic model has an analytical solution, which is used for this purpose. However, the general solution to the oscillator model can only be expressed in terms of a challenging integral which itself requires numerical approximation; thus, our "true" solution is that yielded by the SciPy RK5(4) solver with tolerance set to 10^{-15}. In these results, we observe that the error in the likelihood converges at the same order as that of the solver used, except in the driven oscillator, where the second order method outperforms all the others. This poor

performance of Runge-Kutta methods of order 3 and above is presumably caused by the lack of continuity of the third order and higher derivatives of the solution to the driven oscillator. We also considered the approach in which the solver mesh is adapted based on a relative tolerance on the local truncation error (Figure 7.1, bottom). In this approach, we sometimes observe relative errors in the log likelihood which significantly exceed the relative tolerance, as expected since the tolerance here applies only to the local truncation error and not to the log-likelihood itself (this further supports our findings presented in Chapter 6).

7.4.2 Estimating the error in the log-likelihood using adjoint methods

In this section, we use the adjoint-based techniques derived in §7.2.1 to estimate the error in the log-likelihood arising from numerical approximation of the underlying ODE.

First, we studied the logistic model, eq. (7.21). Synthetic data was generated using 21 data points evenly spaced between $t = 0$ and $t = 20$, using the parameter values $R = 1$, $K = 1$, and $x_0 = 0.01$. IID Gaussian noise of standard deviation 0.01 was added. The true value of the log-likelihood at the true parameter values was computed using the analytical solution to the logistic model. Then, for a range of solver step sizes, the numerical solution at the true parameter values was obtained using the fourth order Runge-Kutta method (RK4) on a uniform solver grid, and the approximate log-likelihood was computed using this solution. Subsequently, the adjoint equation was solved on the same solver grid using a fifth order Runge-Kutta method (RK5). The adjoint solution was interpolated between mesh points using piecewise, continuous cubic interpolation, allowing the error estimate (7.7) to be approximated using 2-point Gaussian quadrature on each subinterval between solver grid points.

In Figure 7.2, we plot the adjoint-based estimates of the error in the log-likelihood and compare them to the actual observed error in the log-likelihood for the range of solver step sizes considered. For the coarsest solver grid considered (with a step size of 1), the adjoint-based error estimate is seen to substantially differ from the actual error in the log-likelihood, as expected based on our discussion in §7.2.2. As the solver step size is refined, the adjoint-based error estimate closely approximates the actual error in the log-likelihood.

Next, we repeated the same experiment for the oscillator model, eq. (7.22). We set

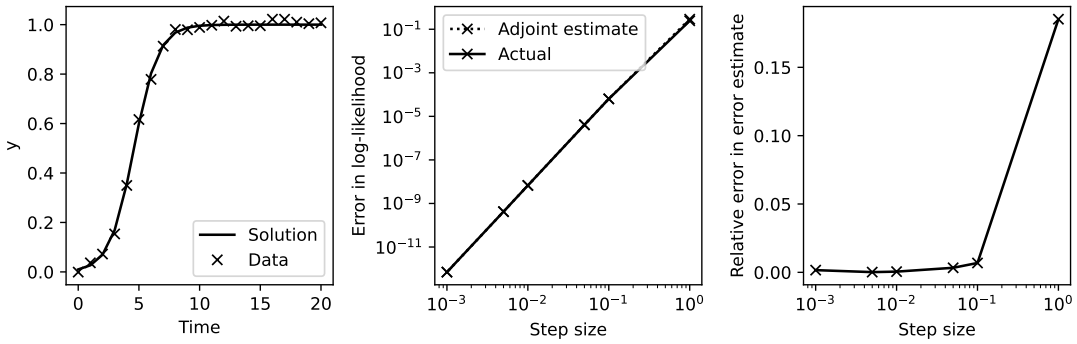


Figure 7.2: **Adjoint-based error estimation for the logistic model.** (Left) The solution and synthetic data points. (Middle) The adjoint-based estimate of the error in the log-likelihood, compared to the actual observed error in the log-likelihood, for a range of solver grid step sizes. (Right) The relative error in the adjoint-based error estimate, for a range of solver grid step sizes.

the stimulus function according to:

$$F(t) = \begin{cases} 0 & t < 4, \\ 1 & 4 \leq t < 8, \\ 0 & 8 \leq t < 12, \\ 1 & 12 \leq t < 16, \\ 0 & 16 \leq t. \end{cases}$$

Synthetic data was generated using 101 data points evenly spaced between $t = 0$ and $t = 20$, using the parameter values $k = 0.1$, $c = 1$, and $x_0 = 0.01, \dot{x}_0 = 1$. IID Gaussian noise of standard deviation 0.1 was added. The true solution at the true parameter values was approximated using the SciPy RK5(4) solver with tolerance set to 10^{-15}. Then, the log-likelihood at the true parameter values, as well as its approximation based on numerical solutions to the ODE and the adjoint-based estimates of the errors in those approximations were computed for a range of solver grid step sizes exactly as described above for the logistic model. We plot the results in Figure 7.3. The adjoint-based estimate of the error in the log-likelihood is seen to be most accurate for solver step sizes around 10^{-2}.

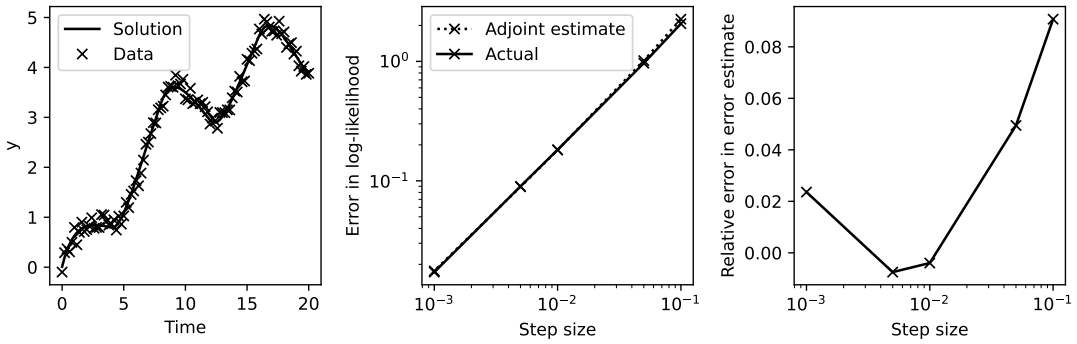


Figure 7.3: **Adjoint-based error estimation for the oscillator model.** (Left) The solution and synthetic data points. (Middle) The adjoint-based estimate of the error in the log-likelihood, compared to the actual observed error in the log-likelihood, for a range of solver grid step sizes. (Right) The relative error in the adjoint-based error estimate, for a range of solver grid step sizes. We compute the relative error according to $(e_Q - e_{Q,\text{True}})/e_{Q,\text{True}}$; negative values of the relative error indicate that $e_Q - e_{Q,\text{True}} < 0$.

7.5 Discussion

The adjoint can usefully play two roles when seeking to infer the parameters of an ODE model. The first is to estimate the error in the log-likelihood (§7.2.1) and thereby ensure the magnitude of error in the log-likelihood is low enough for accurate inference. The second is to compute sensitivities of a quantity of interest to changes in the parameter that are necessary for gradient based MCMC samplers (§7.2.3). We propose that the error estimate and the gradient be used together to develop an efficient MCMC inference algorithm. As we saw in Chapter 6, just controlling error in the local truncation error is not necessarily sufficient to accurately infer the parameters of an ODE model; instead, the error in the log-likelihood must be kept at a small level (the allowable tolerance for errors on the log-likelihood could be informed by, for example, our discussion in §6.4.2). Although solving the adjoint equation at each parameter value proposed by the inference algorithm would involve extra computational expense, we anticipate that much of this expense could be offset by the additional efficiency afforded by a gradient based sampler. Most simply, existing solver step size adaptation algorithms (such as RK5(4)) could be employed: the adjoint-based error estimate would be used as a check to ensure that a given tolerance on the local truncation error did not cause an unacceptable error for inference at each parameter value; if error on the log-likelihood appeared too high, the tolerance on the local truncation error could be refined downwards. Such an algorithm is proposed in Algorithm 3. However, the adjoint-based error estimate is potentially informative enough to drive an approach to grid refinement based directly

on the error in the likelihood. By solving the integral in eq. (7.7) cumulatively, the error in the log-likelihood from each segment of the solver grid could be determined; this information could be used to directly refine or coarsen the grid in the most appropriate regions of time.

The adjoint-based methods discussed in this chapter yield approximations to the gradient and error in the likelihood. In §7.2.2, we provided some initial investigations of the error in the adjoint-based approximations, and in the test problems studied in this chapter, we observed that our selected numerical methods for solving the adjoint problem achieved sufficient accuracy. However, in order to increase the robustness of the proposed inference approach as it may be applied to a variety of problems, further work is necessary to ensure that the adjoint-based approximations to the gradient and error in the likelihood are themselves sufficiently accurate not to interfere with inference.

Throughout this chapter, as well as Chapter 6, we made simplistic assumptions about the noise processes specifying the stochastic deviations between the (accurately solved) differential equation and the observed data. Typically, we assumed that they were IID Gaussian (see eq. (2.9)). Many real datasets do not obey this assumption, however, and in fact the level of variance or autocorrelation in the error terms may vary over time. Thus, in the next chapter, we develop more flexible noise processes for fitting ODE models to data.

7.6 Data and software

The code to perform the computer experiments presented in this chapter was written in Python 3 and is available in an open source Python library at `https://github.com/rccreswell/adjinf`.

Algorithm 3 Evaluation of the log-likelihood at parameter values proposed by a gradient-based MCMC sampler (e.g., NUTS).

$\theta \leftarrow$ Parameter values proposed by NUTS
$r_{\mathcal{L}} \leftarrow$ User specified value (tolerance of log-likelihood)
$r \leftarrow$ Value of r from the previous MCMC iteration (tolerance of local truncation error)
$e \leftarrow \infty$ (Estimated error in log-likelihood at θ)
while $e > r_{\mathcal{L}}$ **do**
    Solve the ODE model at parameters θ using the RK5(4) adaptive solver with local truncation error tolerance r, to obtain an approximate solution $\hat{x}$ on a grid α.
    Solve the adjoint equation (eq. (7.5)) using the RK6 solver with grid α, to obtain the adjoint solution ϕ.
    $e \leftarrow$ error estimate (eq. (7.7))
    $dQ/d\theta \leftarrow$ gradient estimate (eq. (7.9))
    if $e < 0.25 r_{\mathcal{L}}$ **then**
        $r \leftarrow 2r$
        Return (Use $\hat{x}$ to compute the log-likelihood, and $dQ/d\theta$ as the gradient.)
    end if
    if $e < r_{\mathcal{L}}$ **then**
        Return (Use $\hat{x}$ to compute the log-likelihood, and $dQ/d\theta$ as the gradient.)
    end if
    $r \leftarrow r/10$

Chapter 8

Using flexible noise models to avoid noise model misspecification in inference of differential equation time series models

Overview

Our work throughout this thesis has focused on the development of accurate (and accurately solved) models for the processes underlying time series data. When modelling time series, however, it is inevitable that some of the observed variation cannot be modelled by the "signal" process (the process of interest). Instead, this is handled through stochastic "noise" terms, representing nuisance factors. Throughout this thesis, we have often made the typical choice of independent Gaussian noise for the noise process, which defines a statistical model that is simple to implement but may mischaracterise the measurement process. There are a range of alternative noise processes available but, in practice, none of these may be entirely appropriate, as actual noise may be better characterised as a time-varying mixture of various types. Here, we present classes of flexible noise processes that adapt to a system's characteristics, using a multivariate normal kernel where Gaussian processes allow for non-stationary persistence and variance. These noise processes faithfully reproduce parameter estimate uncertainty when doing inference using the correct noise model. We apply our models to time series problems using real data from electrophysiology, and we detect regions of autocorrelation and heteroscedasticity in the noise terms, with a significant difference in the estimated

parameters obtained relative to an IID Gaussian assumption.

Publications

This chapter is based on a preprint available at:

- **R. Creswell,** B. Lambert, C. L. Lei, M. Robinson, D. J. Gavaghan: "Using flexible noise models to avoid noise model misspecification in inference of differential equation time series models," arXiv:2011.04854 (2020) [Creswell et al., 2020].

  Contributions: I was the primary author of this preprint and conducted the development of the noise models, their application to the problems, the software implementation, and the visualisation and interpretation of results. All authors made contributions and suggestions to the writing and revision of the preprint, and some of these contributions are reflected in the wording of parts of this chapter.

8.1 Introduction

We model a noise-free trajectory $\{\bar{y}_i\}_{i=1}^N$ at time points $\{t_i\}_{i=1}^N$ according to,

$$\bar{y}_i = f(t_i; \theta). \tag{8.1}$$

Here, we assume that f represents the solution to an ODE. Even if the model underlying f is appropriately specified, and even if it is numerically computed with sufficient accuracy (see Chapter 6), it is inevitable that observed data will not obey f exactly. Instead, for eq. (8.1) to be a viable model of real data, it must be combined with a noise process modelling the myriad of factors affecting the data which are not (and, often, realistically cannot) be included in f itself.

Assumptions made about the form of the noise can substantially change estimated posterior uncertainty of θ [Lambert et al., 2023]. Notably, when the noise model is misspecified, posterior variance in model parameters may be drastically underestimated or overestimated. Misspecification may also lead to biased estimates. The standard assumption of independent and identically distributed (IID) Gaussian noise is applicable in some cases, but there are many other possible forms. For example, consecutive observations may be correlated due to imperfections in measurement rather than the shape of the signal itself; the magnitude of measurement noise may scale with function

values; there may be time periods with higher observation volatility due to environmental variation; or even a mixture of these various types of noise within a single time series. Non-Gaussian and non-IID noise is also likely to appear in cases of time series model misspecification: when the best available model does not coincide with the hypothetical true process which generated the data, regions of poor fit may be accompanied by residual autocorrelation and spikes in the magnitude of the noise terms.

In applied circumstances, the exact noise process is never known. Some form for the noise must therefore be assumed, with consequences for inference. Whatever choice is made should have some rational basis but be flexible enough to account for the particular sample of data to hand. In this vein, parametric models likely fall short and, instead, more adaptable non-parametric methods prosper. Here, we describe nonparametric models for capturing noise processes that defy characterisation into existing boxes. Through a host of toy examples with predetermined noise processes, we show that parameter inference using our noise models faithfully reproduces the true posterior distributions; that is, those distributions that result when using the correct noise process. Figure 8.1 gives an overview of the proposed approach to noise modelling in a synthetic example where the magnitude of the noise terms increases over time, and some level of autocorrelation is present..

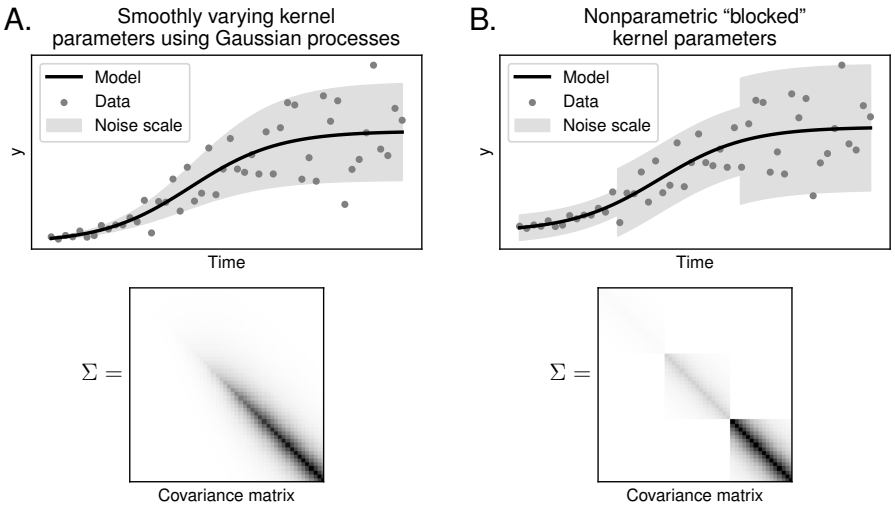


Figure 8.1: **Two noise processes for time series modelling.** Panel (A) shows how non-stationary covariance kernels with continuously time-varying parameters can be used to learn the covariance matrix; and panel (B) shows how a covariance matrix can be built from non-overlapping constituent blocks.

The remainder of this chapter is organised as follows. In §8.2, the multivariate

normal distribution is introduced as a general model for noisy time series data, and we show how an appropriate covariance matrix can be learned from data using positive definite kernels. In §8.3, we describe a method where the parameters of a kernel vary smoothly over time governed by Gaussian processes. In §8.4, we discuss performance considerations for long time series, and §8.5 shows the application of our methods to real data from experiments conducted on the hERG potassium ion channel.

8.2 The multivariate Gaussian likelihood for time series noise

Flexible noise processes depend on a suitably general distributional assumption governing the difference between observed data and the data predicted by the deterministic model, and we use the multivariate normal for this purpose. To learn the covariance matrices of the multivariate normal, we use positive definite kernel functions (see, e.g., [Delisle et al., 2020]); in this section, we show how they can be used to correctly infer parameter posteriors for a time series model with stationary but non-IID noise.

8.2.1 Description of multivariate likelihood

The dataset consists of time points $\{t_i\}_{i=1}^N$ and corresponding noisy data $\{y_i\}_{i=1}^N$. A typical modelling assumption, widely used, for example, in Chapter 6 of this thesis, is to treat the noise on each data point as IID Gaussian with a variance parameter σ^2, so that,

$$y_i = f(t_i; \theta) + \varepsilon_i, \quad i = 1, \ldots, N, \tag{8.2}$$

$$\varepsilon_i \overset{\text{IID}}{\sim} \mathcal{N}(0, \sigma). \tag{8.3}$$

Our first step is to generalize eq. (8.3) so that the variance of noise terms can vary (i.e., allow the noise to be non-identically distributed), and each noise realisation can be correlated with its neighbours (i.e., be non-independent). A multivariate Gaussian can handle both of these generalisations, where we model a random vector, $\mathbf{y} = (y_1, \ldots, y_N)^\top$, as having a mean, $\mathbf{f}(\theta) = (f(t_1; \theta), \ldots, f(t_N; \theta))^\top$,

$$\mathbf{y} \sim \mathcal{N}(\mathbf{f}(\theta), \Sigma). \tag{8.4}$$

For appropriate values of the covariance matrix Σ, this distributional assumption encompasses a wide variety of noise forms which may include correlated and heteroscedastic noise terms. For example, eq. (8.4) could describe heteroscedastic noise which scales

with the magnitude of the trajectory with $\Sigma = \text{diag}(\mathbf{f}(\theta)\sigma^2)$. However, for autocorrelated noise terms, Σ would contain off-diagonal elements.

8.2.2 Learning the covariance matrix, Σ

Multiple methods have been proposed for inference of covariance matrices [Hoffbeck and Landgrebe, 1996, Lam and Fan, 2009, Diggle and Verbyla, 1998, Bickel and Levina, 2008, Cai and Liu, 2011, Schäfer and Strimmer, 2005]. A standard Bayesian approach places a prior on Σ and infers it along with ODE model parameters, θ. Typical choices for priors include the conjugate inverse-Wishart [Gelman et al., 2013, Huang and Wand, 2013], or a prior based around the LKJ correlation matrix [Lewandowski et al., 2009, Stan Development Team, 2016]. However, these methods are not designed to handle the covariance of a single time series. For a single time series obeying eq. (8.4), there is just one multivariate data point (that is, the vector $\mathbf{y}$) available to inform the matrix Σ. With such limited data, these methods for estimating covariance matrices have too much freedom, resulting in dense matrices that overfit the data.

A more productive strategy is to impose a positive definite covariance function $C : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, which generates a covariance matrix according to the rule,

$$\Sigma_{ij} = C(t_i, t_j). \tag{8.5}$$

For example, heteroscedastic errors, where $\Sigma = \text{diag}(\mathbf{f}(\theta)\sigma^2)$, could be represented by the following covariance function:

$$C(t_i, t_j) = f(t_i; \theta)\sigma^2\delta_{ij}. \tag{8.6}$$

where $\delta_{ij} = 1, \text{ if } i = j;\ 0, \text{otherwise}$. In this chapter, we consider positive definite kernels which are flexible enough to capture a wide variety of noise forms, with parameters that can, nonetheless, be learned from a single time series.

8.2.3 Kernels for time series noise

In this section, we introduce the kernels used throughout this chapter. Notwithstanding the important differences discussed in §8.2.4, much of the work on kernel functions for Gaussian processes is applicable to ODE noise models as well, and the three kernels we discuss have seen extensive use in Gaussian process regression. One of the most

widely used positive definite kernels is the Gaussian kernel (also called the Radial Basis Function, or RBF) [Fasshauer, 2011],

$$C(t_i, t_j) = \sigma^2 e^{-(t_i - t_j)^2 / 2L^2}. \tag{8.7}$$

We also consider the Laplacian kernel [Feragen et al., 2015] for specifying time series autocovariances, since it more faithfully reproduces the types of persistence emergent from basic univariate time series models,

$$C(t_i, t_j) = \sigma^2 e^{-|t_i - t_j| / L}. \tag{8.8}$$

The kernels in eqs. (8.7) & (8.8) are each characterised by two parameters which control the size and autocovariance in the errors. A more general class of kernels is the Matérn [Williams and Rasmussen, 2006],

$$C(t_i, t_j) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{L} |t_i - t_j| \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}}{L} |t_i - t_j| \right), \tag{8.9}$$

where K_ν is the modified Bessel function of the second kind. For $\nu = 1/2$, the Matérn kernel simplifies to the Laplacian kernel.

8.2.4 Comparison to Gaussian processes (GPs)

Consider a function $g : \mathcal{X} \to \mathbb{R}$ obeying a Gaussian process (GP) with mean function m and kernel C, i.e. $g \sim \mathcal{GP}(m, C)$ (see, for example, [Rasmussen, 2003]). For every finite set of inputs $\{t_i\}_{i=1}^N$, $t_i \in \mathcal{X}$, the vector of function values $\mathbf{g} = (g(t_1), \ldots, g(t_N))^\top$ has a multivariate Gaussian distribution,

$$\mathbf{g} \sim \mathcal{N}(\mathbf{m}, \Sigma), \tag{8.10}$$

where $\mathbf{m} = (m(t_1), \ldots, m(t_N))^\top$ and Σ is generated as in eq. (8.5). This distribution, identical with eq. (8.4) for $m(\cdot) = f(\,\cdot\,; \theta)$, illustrates an apparent resemblance between the multivariate normal likelihood for time series noise and the GP. Our proposed noise model, however, differs from a GP regression in several key aspects:

1. In GP regression, eq. (8.10) determines a *prior* over functions, and the posterior over functions is inferred. Our proposed noise model uses the multivariate normal specification as a *likelihood* for finite observed data, and posterior inference applies only to the parameters of f, not the functional form of the noise-free relationship

between y and t which is assumed fixed and fully determined by θ.

2. To handle noisy data, Gaussian process regression typically adds an extra noise term—often IID Gaussian. No such terms are used in our multivariate normal noise process.

That is, in full, the likelihood for our multivariate normal model is given by eq. (8.4), with covariance matrix given by eq. (8.5). An example of the utility of the multivariate normal noise process is shown in the next section. In this example, we show that the Laplacian kernel can faithfully capture autoregressive order 1 (AR(1)) noise in an ODE time series model, enabling accurate posterior inference for the ODE model parameters.

8.2.5 Stationary AR(1) noise with Laplacian kernel

Before studying non-stationary covariance functions in the subsequent sections, we first study the applicability of the covariance function approach when the noise terms are stationary. We show that accurate inference for stationary non-IID noise can be achieved using the standard Laplacian kernel,

$$C(t_i, t_j) = \sigma^2 e^{-|t_i - t_j|/L}. \tag{8.11}$$

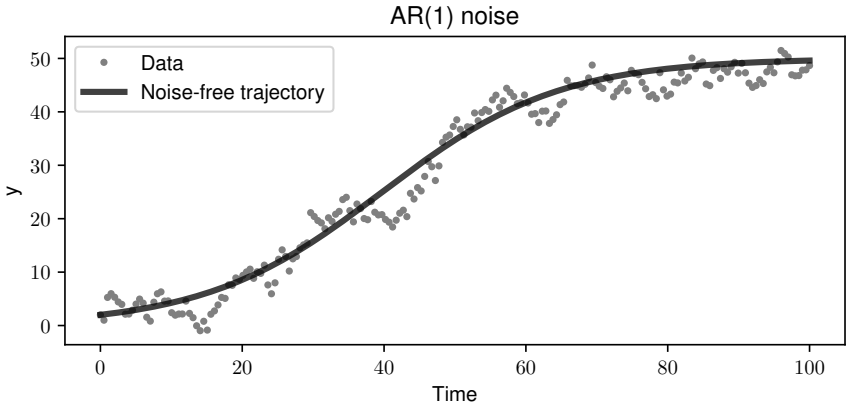
Here, we show the results of the model applied to a synthetic logistic growth time series with autoregressive order 1 (AR(1)) error terms. These results are shown in Figure 8.2. Panel (a) shows a synthetic noisy time series. The underlying model trajectory, labelled "Noise-free trajectory", is calculated from a logistic growth model,

$$\frac{dy}{dt} = ry(1 - y/K). \tag{8.12}$$

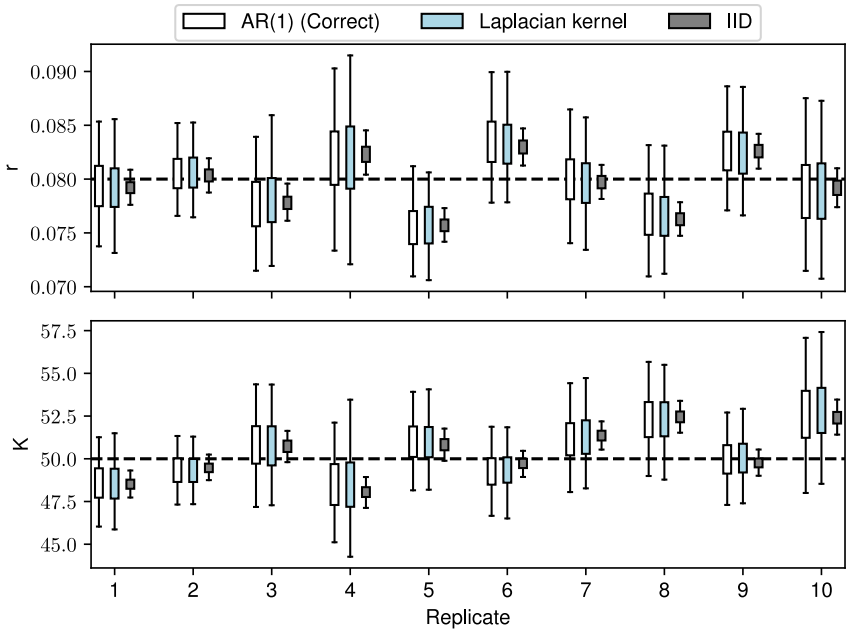
The AR(1) time series shows persistence in the error terms: the error term at any given time points depends both on a random fluctuation as well as the previous observation. Specifically, we model each error term $\varepsilon_i = y_i - f(t_i; \theta)$ according to:

$$\epsilon_i = \rho\varepsilon_{i-1} + v_i, \tag{8.13}$$

where v_i is Gaussian white noise, $v_i \sim N(0, \sigma\sqrt{1-\rho^2})$. In these simulations, we used $\rho = 0.8$ and $\sigma = 3$. Ten replicates of the time series with AR(1) noise were generated. For each time series, Bayesian inference for the parameters r and K was performed for each of three noise processes we consider: IID Gaussian with unknown variance



(a)



(b)

Figure 8.2: **Capturing AR(1) noise using a stationary Laplacian kernel**. Panel (a) shows a logistic growth time series with AR(1) noise. Ten replicates of the AR(1) time series were generated. Panel (b) shows the posterior distributions for logistic growth model parameters under three different assumptions for the noise process for each replicate. The boxes cover the central 50% posterior estimates, while the whiskers cover the central 95% posterior estimates. The dashed lines indicate true parameter values.

(incorrectly specified), AR(1) with two unknown parameters (correctly specified), and the multivariate Gaussian likelihood with Laplacian kernel covariance. MCMC sampling was performed using three chains of the Haario Bardenet adaptive covariance algorithm, with a total of 20000 iterations in each chain [Haario et al., 2001, Johnstone et al., 2016]. The first half of each chain was discarded as warm-up, and convergence was assessed using the Gelman $\hat{R}$ statistic [Gelman et al., 2013]. In Panel (b), the results of posterior inference for r and K are shown under the three noise processes, across 10 replicates. In each replicate, the bars indicate the central 95% of the posterior, while the dashed lines indicate the true values of the posterior. In each replicate, the first posterior with the correctly specified AR(1) noise process shows relatively high posterior uncertainty. The general multivariate normal noise process with Laplacian kernel reproduces the high level of posterior uncertainty in model parameters. By contrast, the incorrectly specified IID assumption underestimates posterior uncertainty.

8.3 Flexible noise for ODEs using Gaussian processes

In this section, we describe a flexible noise process which can learn effective covariance matrices from a time series. Standard positive definite kernels such as eq. (8.7) and eq. (8.8) are appropriate for simple covariance matrices. They are, however, stationary: depending only on the difference between two time points and not on absolute time. In this section, we consider models that allow kernel parameters (for example, σ and L in eq. (8.8)) to vary smoothly over time, allowing distinct sections of a time series to have different noise magnitudes and persistences. First, a brief overview of existing work on non-stationary covariance functions is provided in §8.3.1. In §8.3.2, the non-stationary version of the Laplacian kernel is presented. In §8.3.3, inference for non-stationary kernel parameters is introduced, and in §8.3.4, GP hyperparameter selection is discussed. In §8.3.5, results are presented on synthetic data using the non-stationary Laplacian kernel.

8.3.1 Background on non-stationary covariance functions

Non-stationary covariance functions have been used for spatial modelling and Gaussian process regression. Unlike stationary kernels which depend only on the distance between the two inputs, in the non-stationary case, the kernel shape itself must depend on the input location. This is expressed using the notation $k_s(u)$ for a kernel centred at location s and evaluated at location u. For example, for the Laplacian kernel with one

dimensional input t, we would take

$$k_t(u) = \sigma(t)^2 e^{-|t-u|/L(t)}, \tag{8.14}$$

with the kernel parameters σ and L being functions of the kernel centre location t.

If eq. (8.14) is used to construct a covariance matrix, there are no guarantees that it will be positive definite. Instead, non-stationary modelling has relied on the following general formula for a non-stationary positive definite covariance function:

$$C(x_i, x_j) = \int_{\mathbb{R}^N} k_{x_i}(u) k_{x_j}(u) du, \tag{8.15}$$

for inputs $x_i, x_j, u \in \mathbb{R}^N$ [Higdon et al., 1999, Paciorek, 2003]. The non-stationary version of the Gaussian RBF covariance function can be derived from this formula, which has been used in non-stationary Gaussian process regression [Gibbs, 1998, Paciorek and Schervish, 2004]. To learn time-varying kernel parameters, a Gaussian process prior can be placed on each [Paciorek and Schervish, 2004, Heinonen et al., 2016].

8.3.2 Non-stationary Laplacian covariance function

In this section, we present a non-stationary version of the Laplacian kernel. The techniques presented here are equally applicable to any appropriate positive definite kernel, however.

The one-dimensional non-stationary Laplacian covariance function is:

$$C(t_i, t_j) = \sigma(t_i)\sigma(t_j)\sqrt{\frac{2L(t_i)L(t_j)}{L(t_i)^2 + L(t_j)^2}} \exp\left(-\frac{|t_i - t_j|}{\sqrt{L(t_i)^2 + L(t_j)^2}}\right). \tag{8.16}$$

Eq. (8.16) may be derived as a special case of the non-stationary Matérn kernel [Paciorek and Schervish, 2004]; it also follows directly from the one-dimensional case of eq. (8.15) using reparameterised versions of the respective stationary kernels, with the reparameterisations chosen to ensure that the final non-stationary covariance functions have a sensible form (cf. eq. (3.69) in [Gibbs, 1998]). The logarithms of L and σ each vary over time governed by Gaussian process priors:

$$\log L \sim \mathcal{GP}(\mu_L, K_L),\ \log \sigma \sim \mathcal{GP}(\mu_\sigma, K_\sigma), \tag{8.17}$$

where μ_L, K_L, μ_σ, and K_σ are the GP hyperparameters.

8.3.3 Inference for non-stationary kernel parameters

Having specified a non-stationary covariance function such as

$$C(t_i, t_j) = \sigma(t_i)\sigma(t_j)\sqrt{\frac{2L(t_i)L(t_j)}{L(t_i)^2 + L(t_j)^2}}\exp\left(-\frac{|t_i - t_j|}{\sqrt{L(t_i)^2 + L(t_j)^2}}\right), \tag{8.18}$$

the next task is to infer the posterior distribution of model and covariance parameters. However, analytic expressions for the posterior mean and variance of the Gaussian processes $L(t)$ and $\sigma(t)$ are not available. Instead, MCMC sampling or maximum a posteriori (MAP) estimation can be used to infer the values of $L(t)$ and $\sigma(t)$ at each time point [Heinonen et al., 2016]. For both MCMC and MAP estimation, we recommend the use of gradient-based methods (e.g., Hamiltonian MCMC and gradient-based optimisers) for improved convergence rates in the high dimensional parameter space [Neal, 2011]. When analytic gradients are not available, automatic differentiation can be used. Indeed, all our GP examples presented in this chapter involve an interpolation scheme discussed in §8.4, and we resort to using automatic differentiation.

We use the following procedure for long ODE time series problems with non-stationary covariance functions, which is specified in Algorithm 4. First, the joint MAP estimate of model parameters and covariance parameters is obtained using a gradient-based optimiser. Then, MCMC sampling is used to obtain the posterior distribution of model parameters conditional on the previously obtained MAP estimate of covariance parameters. For both optimisation and MCMC sampling, random or uniform initialisation of the covariance parameters will work for easier problems but will delay convergence on longer time series. In long time series problems with intelligible noise patterns, we recommend a data-driven initialisation of the covariance parameters in order to accelerate convergence of MCMC or optimisation. To initialise L and σ in the non-stationary Laplacian covariance function, we use the procedure given by Algorithm 5. In practice, gradient-based optimisers such as L-BFGS-B [Zhu et al., 1997] may settle at local maxima. Thus, we perform optimisation with multiple restarts, with each restart taking a different initial value. A set of variable yet plausible initial values for the restarts can be generated by rerunning Algorithm 4 multiple times with different sliding window widths.

Algorithm 4 MCMC estimates of ODE parameters using MAP estimates for kernel parameters.

1: Initialise ϕ, for example using to Algorithm 5 for the Laplacian kernel
2: Use gradient-based optimisation to find $(\theta_{\text{MAP}}, \phi_{\text{MAP}}) = \arg\max p(\theta, \phi|y)$
3: Calculate the fixed covariance matrix Σ_{MAP} such that $\Sigma_{i,j} = C_{\phi_{\text{MAP}}}(t_i, t_j)$
4: Use the covariance matrix defined above to form the likelihood $\mathcal{N}(y|f(t;\theta), \Sigma_{\text{MAP}})$
5: Use MCMC to sample from the conditional posterior $p(\theta|y, \phi_{\text{MAP}})$ =0

Algorithm 5 Initialisation for non-stationary Laplacian kernel parameters.

1: Use optimisation to find the MAP estimate of model parameters assuming an IID noise model, $\theta_{\text{MAP,IID}}$
2: Subtract $f(t; \theta_{\text{MAP,IID}})$ from the observed data to obtain an estimate of the noise terms ϵ_i
3: At each time point t_i, calculate the empirical variance v_i and 1st order autocorrelation ρ_i of the noise terms within a sliding window centred on that time point
4: Smooth both estimates using a Wiener filter [Wiener, 1950]
5: At each time point, t_i, set $\sigma_i = \sqrt{v_i}$ and $L_i = -\Delta t / \log(|\rho_i|)$

8.3.4 Gaussian process hyperparameters

For the GPs defined by eqs. (8.17), we used squared exponential kernels with constant mean functions [Heinonen et al., 2016]. With this assumption, there are six Gaussian process hyperparameters for the model (for each of L and σ, a mean μ, noise level α, and length scale β). Prior knowledge or a grid search can be used to set these values, although existing work suggests that β is the most important parameter [Heinonen et al., 2016]. We set $\alpha = 1$ for both processes, and $\mu_\sigma = 1$.

For new problems, we propose the following procedure for tuning the β hyperparameter. β controls how the Gaussian process can change over time. This behaviour is crucial to the adaptivity of the method. If β is too short, the GP will overfit local fluctuations; too large and it will fail to account for real changes in the process over time. To set the length scale, we used a heuristic based on the expected rate of change of the noise process. Given evenly spaced time points with spacing Δt and a user-specified number of time points N_c, we set β as the solution of

$$\zeta = e^{-(N_c \Delta t)^2/(2\beta^2)}, \tag{8.19}$$

for some small value $\zeta = 0.01$. This equation imposes that the prior covariance between two values of the Gaussian process N_c time points apart is close to zero, thus summarising the prior belief that the noise structure can change over that time scale.

Good choices for N_c will generally be problem specific. For non-uniform spacing, $N_c\Delta t$ could replaced by an appropriate time interval.

For the mean of L, a choice such as $\mu_L = 0$ may result in the prior mean corresponding to a significant amount of autocorrelation present in the noise process. This is not necessarily an undesirable property for the prior: as higher values of autocorrelation in the noise process tend to lead to higher uncertainty in the model parameters, a prior preference for significant autocorrelation is conservative in the sense that it is unlikely to cause parameter uncertainty to be underestimated.

However, when the simplest plausible noise process is desired, a more natural choice is for the prior mean to correspond to a negligible autocorrelation (i.e., independence of the noise terms), such that the more complex autocorrelated noise process will only be preferred *a posteriori* if it is supported by the data. These considerations are particularly important for shorter time series, where there is not enough data to overwhelm the prior, and a choice of $\mu_L = 0$ may cause autocorrelation to be inferred even when no evidence of this exists in the data.

Thus, we propose that μ_L be set according to

$$\mu_L = -\frac{\Delta t}{\log a_0} \tag{8.20}$$

for some "default" autocorrelation a_0 which is close to 0 (or some other value, if justified). In our results, we set $a_0 = 0.001$. In time series with non-uniform spacing, this formula can be used by replacing Δt with the smallest spacing at which independence of consecutive noise terms is considered plausible for that time series.

8.3.5 Example with synthetic data

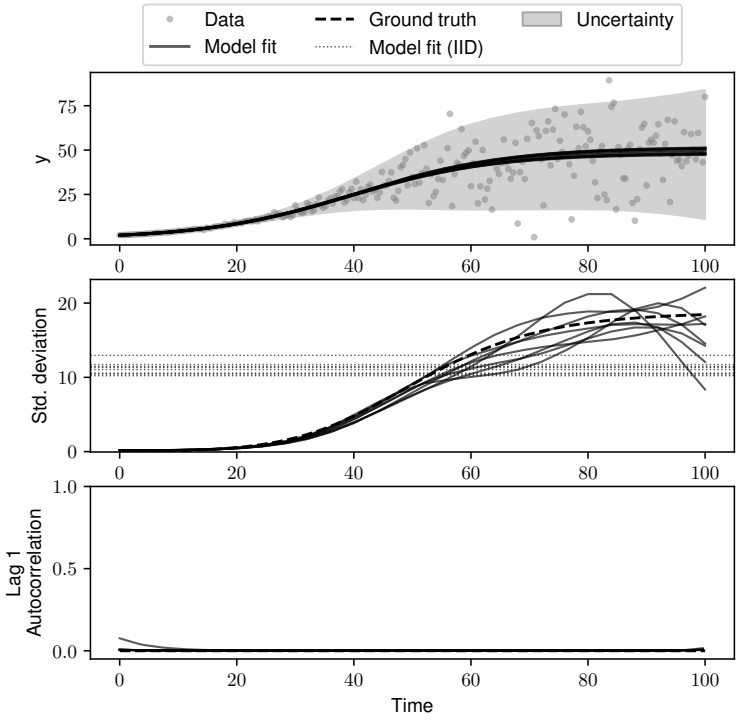
In this example, we use the two-parameter logistic growth model:

$$\frac{dy}{dt} = ry(1 - y/K). \tag{8.21}$$

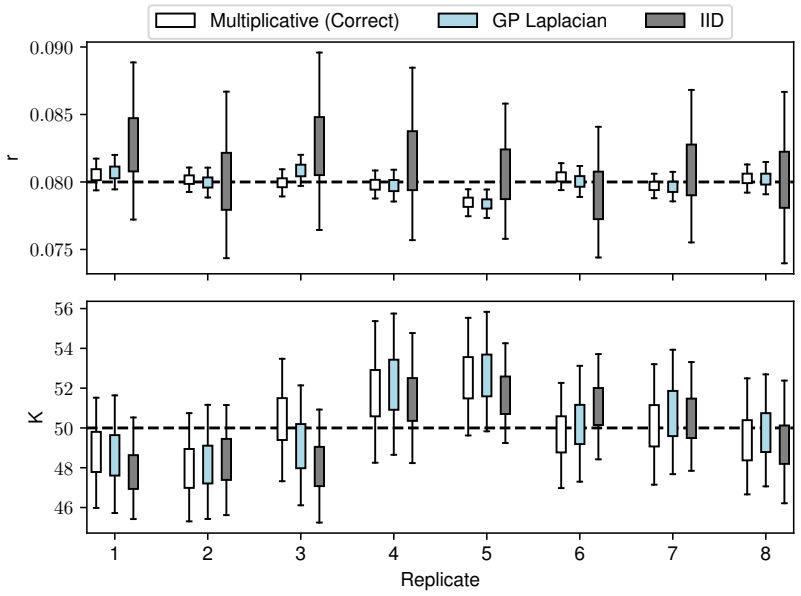
We demonstrate the results of fitting the non-stationary kernel to synthetic data, generated from a logistic growth model with $r = 0.08$, $K = 50$, and $f(t = 0) = 2$ with multiplicative Gaussian noise:

$$y_i = f(t_i; \theta) + f(t_i; \theta)^\eta v_i, \tag{8.22}$$

where y_i is an observed data point, $f(t; \theta)$ is the ODE model solution, and $v_i \overset{\text{IID}}{\sim} \mathcal{N}(0, \sigma^2)$.



(a)



(b)

Figure 8.3: **Non-stationary Laplacian kernel fits to logistic data.** The top plot of panel (a) shows an example logistic growth time series with multiplicative noise, with 250 time points. In the other two plots of (a) and in panel (b), results for model fits to eight replicate datasets are shown. In the middle plot of panel (a), the true standard deviation $\sqrt{C(t_i, t_i)}$ is shown, along with model estimates of it at the MAP estimates for L and σ (one line per each replicate). In this plot, we also indicate the standard deviations estimated by the IID assumption as horizontal dashed lines. In the bottom plot of panel (a), the same is shown as in the middle plot, except with results for the lag 1 autocorrelation, $C(t_i, t_{i+1})/(\sigma(t_i)\sigma(t_{i+1}))$. Panel (b) shows MCMC estimates of the posterior distributions for the logistic growth model parameters under three different assumptions for the noise process; the boxes cover the central 50% posterior estimates, while the whiskers cover the central 95% posterior estimates, and the dashed lines indicate the true values of the parameters.

We set $\eta = 2$ and $\sigma = 0.0075$. Eqs. (8.21)&(8.22) were used to generate eight replicate time series, each with 250 time points. We considered parameter inference for each set of series under three different noise processes: multiplicative (i.e. the true noise process), the non-stationary Laplacian kernel, and IID Gaussian. In each case, Algorithm 4 was used to generate posterior samples from r and K. MCMC sampling for model parameters was performed using Pints inference software [Clerx et al., 2019] with three Markov chains and a total of 20,000 iterations on each, using the Haario Bardenet method [Haario et al., 2001, Johnstone et al., 2016]. On a desktop processor, each chain took approximately 20 minutes to run. The first half of each chain was discarded as warm-up, and convergence was assessed using the Gelman $\hat{R}$ statistic, requiring $\hat{R} < 1.05$ for all parameters [Gelman et al., 2013]. To set the GP hyperparameter β, we used eq. (8.19) with $N_c = 200$. The results are shown in Figure 8.3. In panel (a), the data (from the first replicate) is shown in the top panel, along with the fitted model trajectory. Below, the standard deviation and lag 1 autocorrelation are shown based on the MAP estimates for each replicate and indicate good correspondence with the ground truth. In panel (b), the posterior distributions for the model parameters are shown. The growth parameter, r, was most affected by incorrectly assuming IID Gaussian noise, where the IID noise model resulted in estimates with overly inflated uncertainty. This is because model output is most sensitive to r in the first half of the series, where the IID noise model overestimates the noise level. In all cases, the GP method provided a higher fidelity estimate of uncertainty than IID noise; in most cases the location of the posterior is also improved. Another example of the GPs fitted to synthetic data is given in Figure 8.4. In this example, the true data generating process consists of discrete blocks of different noise models, and the results show the ability of the non-stationary kernel method to find an appropriate smooth approximation.

8.3.6 Non-stationary Laplacian kernel on blocked synthetic data

This section shows an example of the GP non-stationary kernel method being applied to a synthetic time series with very sharp changes in the true noise parameters. The noise process had 5 regimes, and was used with a logistic growth ODE model. The first, third and fifth regimes had IID Gaussian noise with $\sigma = 3$, the second regime had AR(1) noise with $\rho = 0.85$ and $\sigma = 3$, and the fourth regime had IID Gaussian noise with $\sigma = 30$. We found the MAP estimates of the non-stationary Laplacian kernel parameters, using Algorithm 2 for initialisation. In Figure 8.4, the results are shown. The top panel shows one replicate of the data and the MAP estimate of the model trajectory. In the bottom

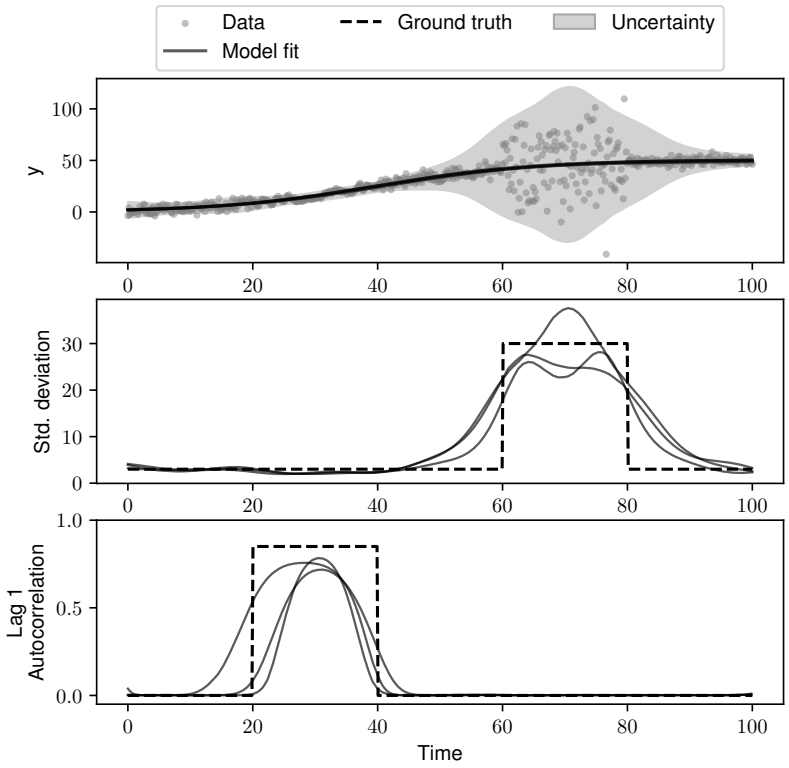


Figure 8.4: **Non-stationary kernel method fit to blocked noise data**. This figure shows how the Gaussian processes in the non-stationary Laplacian kernel handle a noise process with blocks of different types of noise. The top plot shows a logistic growth time series with 5 blocks of different noise forms. In the middle and bottom plots, the true values of standard deviation and lag-1 autocorrelation are shown as dotted lines, while the inferred MAP estimates for standard deviation and lag 1 autocorrelation are shown in solid lines.

two panels, the inferred standard deviation and lag 1 autocorrelation are shown for three replicates. The GPs are unable to learn the sharp corners in the ground truth for standard deviation and autocorrelation, but they do reach a smooth approximation of the ground truth.

8.4 Efficient computation for long time series

Simple noise processes, such as IID Gaussian, are advantageous for their high scalability to time series involving larger numbers of data points. Because the methods described in this chapter, however, require learning time-variation in kernel parameters, scalability is more challenging. In real-life time series problems, we often encounter numbers of

data points in the hundreds or thousands. In these cases, the computational cost of the methods mentioned above becomes a serious hindrance. In this section, we thus provide two computational strategies which can significantly decrease the runtime, and enable scaling to long time series.

The computational cost of the multivariate normal likelihood given in eq. (8.4) is sensitive to the number of time series points, n: the covariance matrix, Σ, is $n \times n$, and the multivariate normal requires its inverse and determinant to be calculated. For highly sampled time series data, n is large enough that these computations are impossible.

To scale to long time series, we take advantage of the relative sparsity of the covariance matrix. Any reasonable kernel, including the kernels we study in this chapter, will generate matrices whose values are close to zero sufficiently far away from the diagonal. We truncate the entries in our covariance matrix, setting all those below a small threshold (10^{-9}) to zero. This results in a sparse matrix whose inverse and determinant can be computed using sparse Cholesky decomposition.

The non-stationary kernel for the GP method presents another scaling challenge as it requires inferring the value of the various GPs at each time point. For the non-stationary Laplacian kernel, this means that $L(t)$ and $\sigma(t)$ in eq. (8.17) are estimated for all t. For long time series, the number of parameters to infer then becomes prohibitive. To reduce this cost, we infer only the GP posterior on a sparser grid of time points. The GP functions are then interpolated to populate the covariance matrix at the original time points; here, we use linear interpolation but recognise that, if the GP value changes rapidly, more nuanced schemes may be appropriate.

The specific speedup enabled by these two computational approximations will vary greatly according to details of the problem at hand but, in our experience, can be quite dramatic. With a time series of length 150, we found that learning the non-stationary Laplacian kernel parameters at every fifth point and then interpolating resulted in a speedup of approximately 500% at each MCMC iteration, and using sparse covariance matrices resulted in a speedup of approximately 4100% for evaluation of the multivariate normal likelihood. On a typical desktop computer, these approximations enable reasonable runtimes for time series with lengths on the order of 10,000 points, as we demonstrate in the following section.

8.5 Application to hERG channel kinetics

In this section, we fit flexible noise processes to real data generated from experiments on the hERG potassium ion channel. This problem is challenging because the noise is clearly not IID, and, also because there may be misspecification of the underlying ODE model. In §8.5.1, we provide a brief description of the hERG channel and a model used to investigate its behaviour and also describe experimental data generated for this system. In §8.5.3, we show how a flexible noise process can capture non-IID noise trends leading to different estimates of model parameters compared to those from an IID noise model.

The hERG channel time series are long (7700 time points, after $10\times$ thinning), and we expect that the variation in the magnitude and autocorrelation of their noise terms can be captured using a continuously varying method. Thus, in this section we use the non-stationary covariance kernel method from §8.3 along with those modifications given in §8.4 to allow efficient computation.

8.5.1 Description of hERG problem

The *human Ether-à-go-go-Related Gene* (hERG) encodes the alpha subunit of the potassium channel Kv11.1 that conducts the rapid delayed rectifier potassium current I_{Kr}. This current is of great importance in cardiac electrophysiology and safety pharmacology; reduction of I_{Kr} by pharmaceutical compounds or mutations can induce fatal disturbances in cardiac rhythm. Interest in this model generally centres on understanding the current response of the hERG channel when a voltage stimuli V is applied. The current can be described with a Hodgkin & Huxley-style structure model [Hodgkin and Huxley, 1952] given by:

$$I_{\text{Kr}} = g_{\text{Kr}} \cdot a \cdot r \cdot (V - E_{\text{K}}), \tag{8.23}$$

where g_{Kr} is the maximal conductance, and E_{K} is the reversal potential (Nernst potential) for potassium ions which can be calculated directly from potassium concentrations using the Nernst equation.

The kinetic terms of the model, a and r, are governed by:

$$\frac{\mathrm{d}a}{\mathrm{d}t} = \frac{a_\infty - a}{\tau_a}, \qquad \frac{\mathrm{d}r}{\mathrm{d}t} = \frac{r_\infty - r}{\tau_r}, \tag{8.24}$$

$$a_\infty = \frac{k_1}{k_1 + k_2}, \qquad r_\infty = \frac{k_4}{k_3 + k_4}, \tag{8.25}$$

$$\tau_a = \frac{1}{k_1 + k_2}, \qquad \tau_r = \frac{1}{k_3 + k_4}, \tag{8.26}$$

where,

$$k_1 = p_1 \exp(p_2 V), \qquad k_3 = p_5 \exp(p_6 V), \tag{8.27}$$

$$k_2 = p_3 \exp(-p_4 V), \qquad k_4 = p_7 \exp(-p_8 V). \tag{8.28}$$

The model has 9 parameters $\theta = (g_{\mathrm{Kr}}, p_1, p_2, \ldots, p_8)$ to be inferred, all of which are positive. These parameters are the maximal conductance g_{Kr} [pS] and kinetic parameters $p_1, p_2, p_3, \cdots, p_8$ [s^{-1}, V^{-1}, s^{-1}, $\cdots$, V^{-1}].

Experimental data of the current are taken from a freely available dataset [Lei et al., 2019b, Lei et al., 2019a], where the voltage stimuli V were designed for parametrising the model.

The logarithm-transformation was applied to all model parameters θ, such that the transformed parameters $\phi = \log(\theta)$ are unconstrained. To account for the impact of this non-linear transformation on the posterior, a Jacobian transformation was applied. Priors for ϕ were selected using existing literature results (Lei et al., 2019a; 2019b), and, for each element of ϕ, a weakly informative prior Gaussian distribution was used (see Table 8.1 for the prior hyperparameters).

8.5.2 hERG Hodgkin-Huxley model parameter priors

In this section, we list the priors used for the 9 log-transformed model parameters in the hERG model introduced in §8.5.1. These values are given in Table 8.1.

8.5.3 Results

For six different cells, the model parameter posteriors were obtained via MCMC using the IID noise model and the non-stationary Laplacian kernel flexible noise model. To obtain posterior samples, the simulated tempering population MCMC algorithm was used [Jasra et al., 2007], with convergence assessed using the Gelman $\hat{R}$ statistic [Gelman et al., 2013], and the first half of each chain discarded as warm up. For the

Parameter	Prior
g_{Kr}	$\mathcal{N}(10.5, 1.0)$
p_1	$\mathcal{N}(-2.5, 3.0)$
p_2	$\mathcal{N}(4.5, 1.0)$
p_3	$\mathcal{N}(-3.5, 1.5)$
p_4	$\mathcal{N}(4.0, 0.5)$
p_5	$\mathcal{N}(4.5, 0.5)$
p_6	$\mathcal{N}(3.0, 1.5)$
p_7	$\mathcal{N}(2.0, 0.5)$
p_8	$\mathcal{N}(3.5, 0.5)$

Table 8.1: **hERG model prior parameters**. This table contains the prior distributions used for each parameter in the hERG model. For each parameter, the prior is a normal distribution with the mean and standard deviation given in the table.

non-stationary Laplacian kernel, we used Algorithm 4 for inference and Algorithm 5 for initialisation.

Figure 8.5 shows the central 95% posterior distribution ranges for all nine model parameters, assuming either IID Gaussian noise (horizontal axis) or the non-stationary noise process (vertical axis). There were significant differences in the parameter estimates for almost all parameters, with much of probability mass not overlapping the IID=Laplacian line. Additionally, the more sophisticated noise model resulted in substantially higher posterior variance for several model parameters, notably including g_{Kr}, p_2 and p_4. Cell A04 is an outlier: this is likely because this cell has a region of drastic misspecification in much of the time series, from $t = 6$ to $t = 10$. While the model fits for all six cells indicate short regions of misspecification, which is particularly apparent after the drops in current around $t = 2$ and $t = 14$, cell A04 (and to a lesser extent, A07) suffer from more extensive misspecification. The data and inferred fits for cell A04 are shown in Figure 8.6. The non-stationary noise model detects the central misspecified region by assigning high variance and autocorrelation in the middle of the time series. In the time series for cell A04, the poor fit between model and data may be largely explained by the fact that our model in this study (§8.5.1) fails to account for experimental artefacts in the voltage clamp experiment, such as leakage current—these artefacts may explain much of the cell-to-cell variability observed in these experiments [Lei et al., 2020a]. Thus, the high levels of standard deviation and autocorrelation detected in these time series suggest that a more detailed model of the experiment is necessary in order to understand these cells and correct the regions of obvious poor fit.

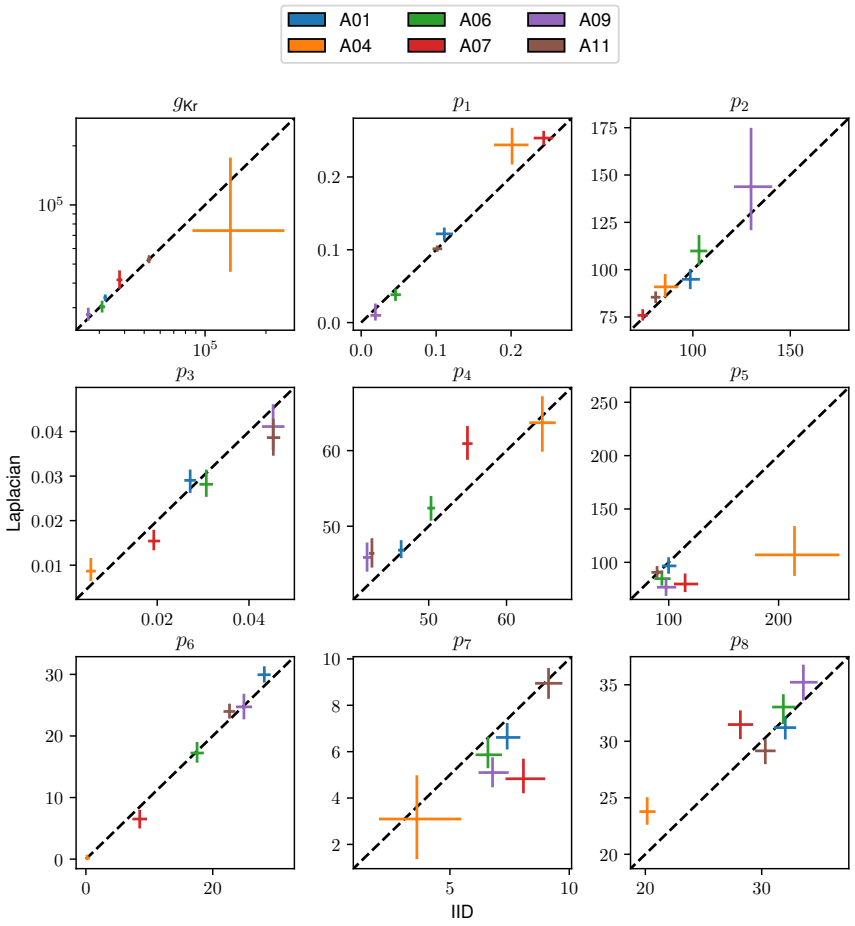


Figure 8.5: **Posterior distributions for hERG model parameters**. This figure compares the posterior distributions resultant from the IID Gaussian noise assumption ("IID") and non-stationary Laplacian kernel ("Laplacian") for the nine hERG model parameters for six cells. For each parameter, the central 95% range of the posterior is shown for each noise model as a bar, with the IID posterior shown on the horizontal axis and the non-stationary posterior shown on the vertical axis. Within each plot, a diagonal dashed line is drawn along $y = x$.

8.6 Discussion

When performing Bayesian inference for the parameters of time series models, the assumption made for the noise process may drastically alter the posterior estimates of parameter uncertainty. The flexible noise models described in this chapter have the ability to learn noise processes from the data, including complex, non-stationary noise processes. The utility of these methods has been demonstrated in constructed synthetic

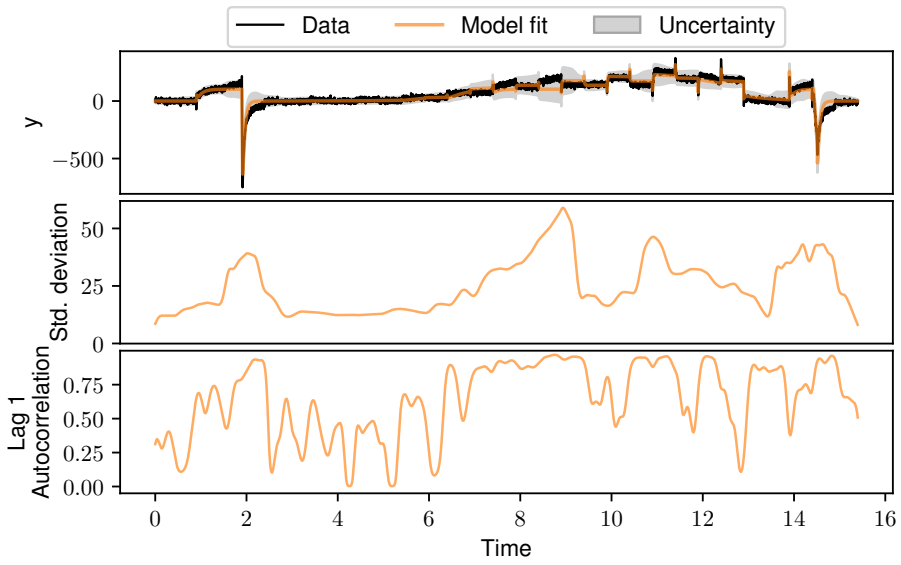


Figure 8.6: **Non-stationary Laplacian kernel noise model fit to hERG cell A04.** This figure shows the data and model fit (top panel), and the MAP estimate standard deviation and lag 1 autocorrelation over time (second and third panels) inferred by the non-stationary Laplacian kernel noise model for cell A04.

data examples.

In applied circumstances, noise terms which exhibit autocorrelation and time-varying magnitude often indicate model misspecification. This is what we observe in the hERG time series problem, in which the best fit model trajectory cannot fully express the signal that is clear in the data. In these cases, our non-stationary covariance noise process is able to pick out the regions of poor fit and model the spikes in magnitude and autocorrelation present at those time periods, with corresponding changes apparent in the model parameter posteriors. Our method is agnostic as to whether the non-stationary noise appears due to model misspecification or measurement imperfections, and future work applying our method to misspecified models may be worthwhile.

8.7 Data and software

The software and data used in this chapter are available in an open source Python package at `https://github.com/rccreswell/flexnoise`.

Chapter 9

Discussion

9.1 Summary of contributions

Applying Bayesian inference to parameterize the models of biological time series is liable to be challenging for several reasons:

1. Existing models may be inaccurate or insufficient, failing to account for all features of the modelled phenomena or population.

2. Models are liable to be parameterised in terms of quantities whose values may vary over time in an arbitrary way, but learning precise estimates of time-varying parameters is difficult.

3. Many biological models involve differential equations, and insufficiently accurate simulation of these models can cause highly erroneuous inference results.

4. The errors between observed data and the best-fit model may disobey standard assumptions such as IID Gaussian.

The investigations presented in this thesis address these challenges and provide generally useful models and approaches for applying Bayesian inference to biological time series.

In the first portion of the thesis, epidemiology was the chief area of application of the methodological investigations. Learning time-varying reproduction numbers (R_t) via stochastic renewal models was the main focus in Chapters 3 and 4. In Chapter 3, we developed a stochastic renewal model of infectious disease outbreaks to incorporate heterogeneity between local and imported cases; using our model, we showed that accounting for this heterogeneity is essential for accurate inference of R_t when there are

differences in the transmissibility of local and imported cases. Thus, we addressed point 1 above (for one particular model).

Making models more complex can be dangerous, however, if it makes the parameters of those models more difficult to infer (potentially leading to non-identifiability or slow inference performance). In Chapter 3, we were able to retain conjugacy between the model likelihood and the prior on R_t, ensuring that inference remained fast and tractable. However, we identified that the sliding window heuristic method, used to learn the pattern of time variation in R_t, depended on fixed hyperparameters which were difficult to tune. When we tuned the sliding window width to be large enough to obtain precise estimates of R_t, it was not possible to estimate rapid changes in R_t.

Thus, in Chapter 4, we introduced an alternative framework for learning time variation in R_t (the ideas presented in Chapter 4 are much more generally applicable to learning time-varying parameters for a variety of models in various fields, however). This approach used a Bayesian nonparametric prior to learn an arrangement of the time points into clusters having shared values of R_t. Using this approach, we were able to pool information of all data points within a given cluster, so that we could estimate precise R_t values within each cluster.

In the second portion of the thesis, we focused on differential equation models and addressed points 3 and 4 above. In Chapter 5, we considered the challenges that can arise when developing ODE-based compartmental models of infectious disease transmission.

In Chapter 6, we provided a thorough investigation of the interplay between numerical approximation of differential equations and inference for their parameters. We showed that insufficient accuracy in the numerical solution could lead to biased parameter estimates, and that even apparently small errors in the forward solution could be magnified and cause significant biases in the recovered parameters. In Chapter 6, we considered three models: a compartmental model of COVID-19; a toy model from elementary mechanics; and a streamflow model used to describe streamflows in rivers.

Chapter 6 showed that numerical solvers can introduce significant errors in the parameter likelihoods, and controlling this error is essential for accurate parameter inference. However, although we derived a bound on the error in the log-likelihood as a function of the local truncation error introduced by the solver, in Chapter 6 we did not provide a practical, general framework for controlling numerical error on the likelihood, aside from more ad-hoc strategies such as visualising slices of the likelihood surface and refining tolerances on the local truncation error. Thus, in Chapter 7, we

used adjoint-based techniques to develop a gradient-based MCMC Bayesian inference strategy which controls error on the log-likelihood while using the gradient to speed up inference. Although computing the actual error in the log-likelihood due to numerical error in the forward solution is computationally expensive, we showed that many of these computations could be reused to approximate the gradient of the log-likelihood with respect to the parameters.

Finally, in Chapter 8, we turned to the stochastic noise terms which are used to model deviations between deterministic differential equation models and noisy observed data. Moving away from the independent and identically distributed (IID) Gaussian assumption, we used a flexible model to learn noise terms whose standard deviation and correlation could vary over time. We showed that when heteroscedasticty and autocorrelation were present in the time series, failing to account for them via the standard IID Gaussian assumption could lead to highly biased parameter inference; however, when the flexible models described in the chapter were used, parameter posteriors could be recovered more accurately.

9.2 Directions for future work

Throughout Chapters 3 and 4, we relied on incidence data to inform values of R_t. Although our methods developed in these chapters were successfully applied to real incidence data from several regions worldwide, more widespread application of these methods would require more careful consideration of the significant levels of noise and bias that are liable to occur in epidemiological time series data such as incidence.

Noise and bias in epidemiological time series can arise from factors such as reporting delays [Gostic et al., 2020], cyclical factors in reporting of cases and deaths [Gallagher et al., 2022], geographic or demographic variability in test-taking behaviours [Nicholson et al., 2022], or population heterogeneity in disease transmission risk [Lloyd-Smith et al., 2005]. Naïve application of the standard Poisson renewal model which we built upon in Chapters 3 and 4 to time series data significantly affected by these factors is likely to cause poor inference results. This is particularly important for nonparameteric methods such as EpiCluster, which we introduced in Chapter 4. In such models, the ability to accurately infer the correct number of changes in parameter value is closely tied to accurate modelling of the stochasticity in the data: if the stochasticity is misspecified, the model may explain genuine changes in the parameter values just as stochastic artefacts, or, equally dangerously, it may overfit and falsely assume that

spurious fluctuations in the data reflect genuine changes in the parameter values.

Distributions such as the negative binomial allow overdispersion [Lloyd-Smith et al., 2005], potentially leading to more robust inference results when used as the renewal distribution in a stochastic model of a disease outbreak. However, such distributions lack the convenient Poisson-Gamma conjugate prior relationship, making inference for R_t more challenging. Developing efficient inference algorithms for R_t not reliant on the conjugacy between the prior and likelihood is therefore likely to be useful future work.

In Chapter 3, we generalized the Poisson renewal model to allow local and imported cases to have differing risks of onwards transmission. Our results in that chapter indicated that modelling this heterogeneity can be important for accurate inference of epidemiological parameters such as R_t. Thus, it would be valuable to apply the same idea—that local and imported cases have potentially different behaviours—to other epidemiological models beyond the Poisson renewal model—for example, the Hawkes process, another stochastic model which has recently seen use in infectious disease modelling [Garetto et al., 2021, Unwin et al., 2021].

The estimation of R_t is not the only epidemiological inference problem that stands to benefit from wider application of Bayesian nonparametric methods such as EpiCluster (Chapter 4). The ideas developed in Chapter 4 could be straightforwardly adapted to a variety of other problems involving parameters which vary over time, and when parameter values change rapidly, our work in this thesis suggests that nonparametric change point models may outperform existing methods. As an example, serocatalytic models (e.g., [Pons-Salort et al., 2023]) of the age-structured seroprevalence of a disease are often parameterised in terms of a time-varying historical force of infection (FOI). Learning time variation in FOI presents many of the same challenges as learning R_t: to obtain precise estimates, information across multiple time points must be leveraged, but *a priori* it is rarely obvious how to divide the time interval to achieve this.

As mentioned above, however, such models may not yield convenient conjugate relationships, necessitating further work on the development of efficient inference algorithms.

Bibliography

[Abbott et al., 2020] Abbott, S., Hellewell, J., Thompson, R. N., Sherratt, K., Gibbs, H. P., Bosse, N. I., Munday, J. D., Meakin, S., Doughty, E. L., Chun, J. Y., et al. (2020). Estimating the time-varying reproduction number of SARS-CoV-2 using national and subnational case counts. *Wellcome Open Research*, 5(112):112.

[Adam, 2020] Adam, D. (2020). Special report: The simulations driving the world's response to COVID-19. *Nature*, 580(7803):316–318.

[Ainsworth and Oden, 2000] Ainsworth, M. and Oden, T. (2000). A posteriori *error estimation in finite element analysis*. John Wiley-Teubner.

[Arregui et al., 2018] Arregui, S., Aleta, A., Sanz, J., and Moreno, Y. (2018). Projecting social contact matrices to different demographic structures. *PLoS Computational Biology*, 14(12):e1006638.

[Baker et al., 2018] Baker, R. E., Pena, J.-M., Jayamohan, J., and Jérusalem, A. (2018). Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biology letters*, 14(5):20170660.

[Bangerth and Rannacher, 2003] Bangerth, W. and Rannacher, R. (2003). *Adaptive finite element methods for differential equations*. Birkhauser Verlag.

[Barth, 2004] Barth, T. J. (2004). *A posteriori error estimation and mesh adaptivity for finite volume and finite element methods*, volume 41 of *Lecture Notes in Computational Science and Engineering*. Springer, New York.

[Bayes and Price, 1763] Bayes, T. and Price, R. (1763). An Essay towards Solving a Problem in the Doctrine of Chances. *Philosophical Transactions of the Royal Society of London*, 53:370–418.

[Becker and Rannacher, 2001] Becker, R. and Rannacher, R. (2001). An optimal control approach to *a posteriori* error estimation in finite element methods. *Acta Numerica*, pages 1–102.

[Bickel and Levina, 2008] Bickel, P. J. and Levina, E. (2008). Regularized estimation of large covariance matrices. *The Annals of Statistics*, 36(1):199–227.

[Birrell et al., 2021] Birrell, P., Blake, J., Van Leeuwen, E., Gent, N., and De Angelis, D. (2021). Real-time nowcasting and forecasting of COVID-19 dynamics in England: the first wave. *Philosophical Transactions of the Royal Society B*, 376(1829):20200279.

[Blei et al., 2017] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

[Brauner et al., 2021] Brauner, J. M., Mindermann, S., Sharma, M., Johnston, D., Salvatier, J., Gavenčiak, T., Stephenson, A. B., Leech, G., Altman, G., Mikulik, V., et al. (2021). Inferring the effectiveness of government interventions against COVID-19. *Science*, 371(6531).

[Cai and Liu, 2011] Cai, T. and Liu, W. (2011). Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association*, 106(494):672–684.

[Capistrán et al., 2022] Capistrán, M. A., Christen, J. A., Daza-Torres, M. L., Flores-Arguedas, H., and Montesinos-López, J. C. (2022). Error control of the numerical posterior with Bayes factors in Bayesian uncertainty quantification. *Bayesian Analysis*, 17(2):381–403.

[Capistrán et al., 2016] Capistrán, M. A., Christen, J. A., and Donnet, S. (2016). Bayesian analysis of ODEs: solver optimal accuracy and Bayes factors. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):829–849.

[Challen et al., 2021] Challen, R., Dyson, L., Overton, C., Guzman-Rincon, L., Hill, E., Stage, H., Brooks-Pollock, E., Pellis, L., Scarabel, F., Pascall, D., et al. (2021). Early Epidemiological Signatures of Novel SARS-CoV-2 Variants: Establishment of B. 1.617. 2 in England. *medRxiv*.

[Chen et al., 2014] Chen, T., Fox, E., and Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In *International conference on machine learning*, pages 1683–1691. PMLR.

[Chkrebtii et al., 2016] Chkrebtii, O. A., Campbell, D. A., Calderhead, B., Girolami, M. A., et al. (2016). Bayesian solution uncertainty quantification for differential equations. *Bayesian Analysis*, 11(4):1239–1267.

[Christen et al., 2017] Christen, J. A., Capistrán, M. A., Daza-Torres, M. L., Flores-Argüedas, H., and Montesinos-López, J. C. (2017). Posterior distribution existence and error control in banach spaces in the bayesian approach to uq in inverse problems. *arXiv preprint arXiv:1712.03299*.

[Clerx et al., 2019] Clerx, M., Robinson, M., Lambert, B., Lei, C., Ghosh, S., Mirams, G., and Gavaghan, D. (2019). Probabilistic inference on noisy time series (PINTS). *Journal of Open Research Software*, 7(1).

[Conrad et al., 2017] Conrad, P. R., Girolami, M., Särkkä, S., Stuart, A., and Zygalakis, K. (2017). Statistical analysis of differential equations: introducing probability measures on numerical solutions. *Statistics and Computing*, 27(4):1065–1082.

[Cori et al., 2013] Cori, A., Ferguson, N. M., Fraser, C., and Cauchemez, S. (2013). A new framework and software to estimate time-varying reproduction numbers during epidemics. *American Journal of Epidemiology*, 178(9):1505–1512.

[Cowling et al., 2020] Cowling, B. J., Ali, S. T., Ng, T. W., Tsang, T. K., Li, J. C., Fong, M. W., Liao, Q., Kwan, M. Y., Lee, S. L., Chiu, S. S., et al. (2020). Impact assessment of non-pharmaceutical interventions against coronavirus disease 2019 and influenza in hong kong: an observational study. *The Lancet Public Health*, 5(5).

[Creswell et al., 2022] Creswell, R., Augustin, D., Bouros, I., Farm, H., Miao, S., Ahern, A., Gavaghan, D., Lambert, B., and Thompson, R. (2022). Heterogeneity in transmission between hosts affects practical estimates of the time-dependent reproduction number. *Philosophical Transactions of the Royal Society, A*.

[Creswell et al., 2020] Creswell, R., Lambert, B., Lei, C. L., Robinson, M., and Gavaghan, D. (2020). Using flexible noise models to avoid noise model misspecification in inference of differential equation time series models. *arXiv preprint arXiv:2011.04854*.

[Creswell et al., 2023a] Creswell, R., Robinson, M., Gavaghan, D., Parag, K. V., Lei, C. L., and Lambert, B. (2023a). A Bayesian nonparametric method for detecting rapid changes in disease transmission. *Journal of Theoretical Biology*, 558:111351.

[Creswell et al., 2023b] Creswell, R., Robinson, M., Lambert, B., Lei, C. L., Gavaghan, D., and Tavener, S. (2023b). Enhancing gradient-based bayesian inference for initial value problems using the adjoint.

[Creswell et al., 2023c] Creswell, R., Shepherd, K. M., Lambert, B., Mirams, G. R., Lei, C. L., Tavener, S., Robinson, M., and Gavaghan, D. J. (2023c). Understanding the impact of numerical solvers on inference for differential equation models. *arXiv preprint arXiv:2307.00749*.

[Davies et al., 2020] Davies, N. G., Klepac, P., Liu, Y., Prem, K., Jit, M., and Eggo, R. M. (2020). Age-dependent effects in the transmission and control of COVID-19 epidemics. *Nature Medicine*, 26(8):1205–1211.

[Daza-Torres et al., 2021] Daza-Torres, M. L., Montesinos-López, J. C., Capistrán, M. A., Christen, J. A., and Haario, H. (2021). Error control in the numerical posterior distribution in the Bayesian UQ analysis of a semilinear evolution PDE. *International Journal for Uncertainty Quantification*, 11(4).

[Dehning et al., 2020] Dehning, J., Zierenberg, J., Spitzner, F. P., Wibral, M., Neto, J. P., Wilczek, M., and Priesemann, V. (2020). Inferring change points in the spread of COVID-19 reveals the effectiveness of interventions. *Science*, 369(6500):eabb9789.

[Delisle et al., 2020] Delisle, J.-B., Hara, N., and Ségransan, D. (2020). Efficient modeling of correlated noise-II. A flexible noise model with fast and scalable methods. *Astronomy & Astrophysics*, 638:A95.

[Diggle and Verbyla, 1998] Diggle, P. J. and Verbyla, A. P. (1998). Nonparametric estimation of covariance structure in longitudinal data. *Biometrics*, pages 401–415.

[Dormand and Prince, 1980] Dormand, J. R. and Prince, P. J. (1980). A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26.

[Edeling et al., 2021] Edeling, W., Arabnejad, H., Sinclair, R., Suleimenova, D., Gopalakrishnan, K., Bosak, B., Groen, D., Mahmood, I., Crommelin, D., and Coveney, P. V. (2021). The impact of uncertainty on predictions of the CovidSim epidemiological code. *Nature Computational Science*, 1(2):128–135.

[Eriksson et al., 1995] Eriksson, K., Estep, D., Hansbo, P., and Johnson, C. (1995). Introduction to adaptive methods for differential equations. *Acta Numerica*, 4:105–158.

[Escobar and West, 1995] Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588.

[Estep, 1995] Estep, D. (1995). *A posteriori* error bounds and global error control for approximation of ordinary differential equations. *SIAM Journal on Numerical Analysis*, 32(1):1–48.

[Fasshauer, 2011] Fasshauer, G. E. (2011). Positive definite kernels: past, present and future. *Dolomites Research Notes on Approximation*, 4:21–63.

[Feragen et al., 2015] Feragen, A., Lauze, F., and Hauberg, S. (2015). Geodesic exponential kernels: When curvature and linearity conflict. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3032–3042.

[Ferguson et al., 2020a] Ferguson, N, M., Laydon, D., Nedjati-Gilani, G., Imai, N., Ainslie, K., Baguelin, M., Bhatia, S., Boonyasiri, A., Cucunuba Perez, Z., Cuomo-Dannenburg, G., Dighe, A., Dorigatti, I., Fu, H., Gaythorpe, K., Green, W., Hamlet, A., Hinsley, W., Okell, L, C., Van Elsland, S., Thompson, H., Verity, R., Volz, E., Wang, H., Wang, Y., Walker, P., Winskill, P., Whittaker, C., Donnelly, C., Riley, S., and Ghani, A, C. (2020a). Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand.

[Ferguson et al., 2020b] Ferguson, N., Nedjati Gilani, G., and Laydon, D. (2020b). *COVID-19 CovidSim microsimulation model*. R Foundation for Statistical Computing.

[Ferguson et al., 2006] Ferguson, N. M., Cummings, D. A. T., Fraser, C., Cajka, J. C., Cooley, P. C., and Burke, D. S. (2006). Strategies for mitigating an influenza pandemic. *Nature*, 442(7101):448–452.

[Flaxman et al., 2020] Flaxman, S., Mishra, S., Gandy, A., Unwin, H. J. T., Mellan, T. A., Coupland, H., Whittaker, C., Zhu, H., Berah, T., Eaton, J. W., Molod, M., Imperial College COVID-19 Response Team, Ghani, A. C., Donnelly, C., Riley, S., Vollmer, M. A. C., Ferguson, N. M., Okell, L. C., and Bhatt, S. (2020). Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe. *Nature*, 584(7820):257–261.

[Fraser, 2007] Fraser, C. (2007). Estimating individual and household reproduction numbers in an emerging epidemic. *PLOS One*, 2(8).

[Funk, 2020] Funk, S. (2020). Socialmixr: Social Mixing Matrices for Infectious Disease Modelling. *The Comprehensive R Archive Network*.

[Funk et al., 2019] Funk, S., Knapp, J. K., Lebo, E., Reef, S. E., Dabbagh, A. J., Kretsinger, K., Jit, M., Edmunds, W. J., and Strebel, P. M. (2019). Combining serological and contact data to derive target immunity levels for achieving and maintaining measles elimination. *BMC medicine*, 17:1–12.

[Gallagher et al., 2022] Gallagher, K., Bouros, I., Fan, N., Hayman, E., Heirene, L., Lamirande, P., Lemenuel-Diot, A., Lambert, B., Gavaghan, D., and Creswell, R. (2022). Epidemiological Agent-Based Modelling Software (Epiabm). *arXiv preprint arXiv:2212.04937*.

[Gallagher et al., 2023] Gallagher, K., Creswell, R., Gavaghan, D., and Lambert, B. (2023). Identification and attribution of weekly periodic biases in epidemiological time series data. *medRxiv*.

[Garetto et al., 2021] Garetto, M., Leonardi, E., and Torrisi, G. L. (2021). A time-modulated Hawkes process to model the spread of COVID-19 and the impact of countermeasures. *Annual Reviews in Control*, 51:551–563.

[Gautschi, 1997] Gautschi, W. (1997). *Numerical Analysis*. Springer Science & Business Media.

[Gelman et al., 2013] Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, B. (2013). Bayesian Data Analysis. 3rd edition.

[Ghahramani, 2013] Ghahramani, Z. (2013). Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984).

[Gibbs, 1998] Gibbs, M. N. (1998). *Bayesian Gaussian processes for regression and classification*. PhD thesis, University of Cambridge.

[Giles and Süli, 2002] Giles, M. B. and Süli, E. (2002). Adjoint methods for pdes: *a posteriori* error analysis and postprocessing by duality. *Acta Numerica*, 11(1):145–236.

[Gostic et al., 2020] Gostic, K. M., McGough, L., Baskerville, E. B., Abbott, S., Joshi, K., Tedijanto, C., Kahn, R., Niehus, R., Hay, J. A., De Salazar, P. M., et al. (2020). Practical considerations for measuring the effective reproductive number, R_t. *PLOS Computational Biology*, 16(12).

[Government of Ontario, 2021] Government of Ontario (2021). COVID-19 data: likely source of infection.

[Haagmans et al., 2014] Haagmans, B. L., Al Dhahiry, S. H., Reusken, C. B., Raj, V. S., Galiano, M., Myers, R., Godeke, G.-J., Jonges, M., Farag, E., Diab, A., et al. (2014). Middle East respiratory syndrome coronavirus in dromedary camels: an outbreak investigation. *The Lancet Infectious Diseases*, 14(2):140–145.

[Haario et al., 2001] Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, pages 223–242.

[Hansen et al., 2003] Hansen, N., Müller, S. D., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18.

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

[Heinonen et al., 2016] Heinonen, M., Mannerström, H., Rousu, J., Kaski, S., and Lähdesmäki, H. (2016). Non-stationary Gaussian Process Regression with Hamiltonian Monte Carlo. In *Artificial Intelligence and Statistics*, pages 732–740.

[Hennig et al., 2015] Hennig, P., Osborne, M. A., and Girolami, M. (2015). Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150142.

[Higdon et al., 1999] Higdon, D., Swall, J., and Kern, J. (1999). Non-stationary spatial modeling. In *Proceedings of the Sixth Valencia International Meeting on Bayesian Statistics*, pages 761–768.

[Hilborn and Mangel, 2013] Hilborn, R. and Mangel, M. (2013). *The ecological detective: confronting models with data*. Princeton University Press.

[Hindmarsh and Petzold, 2005] Hindmarsh, A. and Petzold, L. (2005). LSODA, ordinary differential equation solver for stiff or non-stiff system. *International Nuclear Information System (INIS)*, 41.

[Hindmarsh et al., 2005] Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., and Woodward, C. S. (2005). SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396.

[Hodgkin and Huxley, 1952] Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544.

[Hoffbeck and Landgrebe, 1996] Hoffbeck, J. P. and Landgrebe, D. A. (1996). Covariance matrix estimation and classification with limited training data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):763–767.

[Hong et al., 2019] Hong, H., Ovchinnikov, A., Pogudin, G., and Yap, C. (2019). SIAN: software for structural identifiability analysis of ODE models. *Bioinformatics*, 35(16):2873–2874.

[Hong Kong Department of Health, 2022] Hong Kong Department of Health (2022). Latest local situation of COVID-19. `https://data.gov.hk/en-data/dataset/hk-dh-chpsebcddr-novel-infectious-agent`.

[Huang and Wand, 2013] Huang, A. and Wand, M. P. (2013). Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis*, 8(2):439–452.

[Institute of Environmental Science and Research, 2022] Institute of Environmental Science and Research (2022). New Zealand COVID Dashboard.

[Jasra et al., 2007] Jasra, A., Stephens, D. A., and Holmes, C. C. (2007). On population-based simulation for static inference. *Statistics and Computing*, 17(3):263–279.

[Johnstone, 2018] Johnstone, R. H. (2018). *Uncertainty characterisation in action potential modelling for cardiac drug safety*. PhD thesis, University of Oxford.

[Johnstone et al., 2016] Johnstone, R. H., Chang, E. T., Bardenet, R., De Boer, T. P., Gavaghan, D. J., Pathmanathan, P., Clayton, R. H., and Mirams, G. R. (2016). Uncertainty and variability in models of the cardiac action potential: Can we build trustworthy models? *Journal of Molecular and Cellular Cardiology*, 96:49–62.

[Kabanikhin and Krivorotko, 2020] Kabanikhin, S. and Krivorotko, O. (2020). Optimization methods for solving inverse immunology and epidemiology problems. *Computational Mathematics and Mathematical Physics*, 60:580–589.

[Kavetski et al., 2003] Kavetski, D., Kuczera, G., and Franks, S. W. (2003). Semidistributed hydrological modeling: A "saturation path" perspective on TOPMODEL and VIC. *Water Resources Research*, 39(9).

[Keeling et al., 2021a] Keeling, M., Dyson, L., Hill, E., Moore, S., and Tildesley, M. (2021a). Road map scenarios and sensitivity: Step 4.[cited 4 aug 2021]. *See https://assets. publishing. service. gov. uk/government/uploads/system/uploads/attachment_data/file/993358/s1288_Warwick_RoadMap_Step_4. pdf*.

[Keeling et al., 2021b] Keeling, M. J., Hill, E. M., Gorsich, E. E., Penman, B., Guyver-Fletcher, G., Holmes, A., Leng, T., McKimm, H., Tamborrino, M., Dyson, L., et al. (2021b). Predictions of COVID-19 dynamics in the UK: Short-term forecasting and analysis of potential exit strategies. *PLoS Computational Biology*, 17(1):e1008619.

[Kucukelbir et al., 2017] Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *Journal of Machine Learning Research*.

[Ladhani et al., 2020] Ladhani, S. N., Chow, J. Y., Janarthanan, R., Fok, J., Crawley-Boevey, E., Vusirikala, A., Fernandez, E., Perez, M. S., Tang, S., Dun-Campbell, K., et al. (2020). Increased risk of SARS-CoV-2 infection in staff working across different care homes: enhanced CoVID-19 outbreak investigations in London care Homes. *Journal of Infection*, 81(4):621–624.

[Lam and Fan, 2009] Lam, C. and Fan, J. (2009). Sparsistency and rates of convergence in large covariance matrix estimation. *Annals of Statistics*, 37(6B):4254.

[Lambert et al., 2023] Lambert, B., Lei, C. L., Robinson, M., Clerx, M., Creswell, R., Ghosh, S., Tavener, S., and Gavaghan, D. (2023). Autocorrelated measurement processes and inference for ordinary differential equation models of biological systems. *Journal of the Royal Society Interface*, 20.

[Lei et al., 2019a] Lei, C. L., Clerx, M., Beattie, K. A., Melgari, D., Hancox, J. C., Gavaghan, D. J., Polonchuk, L., Wang, K., and Mirams, G. R. (2019a). Rapid characterization of hERG channel kinetics II: temperature dependence. *Biophysical Journal*, 117(12):2455–2470.

[Lei et al., 2019b] Lei, C. L., Clerx, M., Gavaghan, D. J., Polonchuk, L., Mirams, G. R., and Wang, K. (2019b). Rapid characterization of hERG channel kinetics I: using an automated high-throughput system. *Biophysical Journal*, 117(12):2438–2454.

[Lei et al., 2020a] Lei, C. L., Clerx, M., Whittaker, D. G., Gavaghan, D. J., De Boer, T. P., and Mirams, G. R. (2020a). Accounting for variability in ion current recordings

using a mathematical model of artefacts in voltage-clamp experiments. *Philosophical Transactions of the Royal Society A*, 378(2173):20190348.

[Lei et al., 2020b] Lei, C. L., Ghosh, S., Whittaker, D. G., Aboelkassem, Y., Beattie, K. A., Cantwell, C. D., Delhaas, T., Houston, C., Novaes, G. M., Panfilov, A. V., et al. (2020b). Considering discrepancy when calibrating a mechanistic electrophysiology model. *Philosophical Transactions of the Royal Society A*, 378(2173):20190349.

[Lewandowski et al., 2009] Lewandowski, D., Kurowicka, D., and Joe, H. (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9):1989–2001.

[Li et al., 2019] Li, L., Holbrook, A., Shahbaba, B., and Baldi, P. (2019). Neural network gradient Hamiltonian Monte Carlo. *Computational Statistics*, 34(1):281–299.

[Li et al., 2021] Li, Y., Campbell, H., Kulkarni, D., Harpur, A., Nundy, M., Wang, X., Nair, H., for COVID, U. N., et al. (2021). The temporal association of introducing and lifting non-pharmaceutical interventions with the time-varying reproduction number (R) of SARS-CoV-2: a modelling study across 131 countries. *The Lancet Infectious Diseases*, 21(2):193–202.

[Lijoi and Prunster, 2010] Lijoi, A. and Prunster, I. (2010). Models beyond the Dirichlet process. In Hjort, N. L., Holmes, C., Muller, P., and Walker, S. G., editors, *Bayesian nonparametrics*. Cambridge University Press.

[Liu et al., 2021] Liu, Y., Gu, Z., and Liu, J. (2021). Uncovering transmission patterns of COVID-19 outbreaks: A region-wide comprehensive retrospective study in Hong Kong. *EClinicalMedicine*, 36.

[Lloyd-Smith et al., 2005] Lloyd-Smith, J. O., Schreiber, S. J., Kopp, P. E., and Getz, W. M. (2005). Superspreading and the effect of individual variation on disease emergence. *Nature*, 438(7066):355–359.

[Lovell-Read et al., 2022] Lovell-Read, F. A., Shen, S., and Thompson, R. N. (2022). Estimating local outbreak risks and the effects of non-pharmaceutical interventions in age-structured populations: SARS-CoV-2 as a case study. *Journal of Theoretical Biology*, 535:110983.

[Martínez and Mena, 2014] Martínez, A. F. and Mena, R. H. (2014). On a nonparametric change point detection model in Markovian regimes. *Bayesian Analysis*, 9(4):823–858.

[Melicher et al., 2017] Melicher, V., Haber, T., and Vanroose, W. (2017). Fast derivatives of likelihood functionals for ode based models using adjoint-state method. *Computational Statistics*, 32(4):1621–1643.

[Mendez-Brito et al., 2021] Mendez-Brito, A., El Bcheraoui, C., and Pozo-Martin, F. (2021). Systematic review of empirical studies comparing the effectiveness of non-pharmaceutical interventions against COVID-19. *Journal of Infection*, 83(3):281–293.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

[Mirams, 2018] Mirams, G. (2018). Numerical errors from ODE solvers can mess up optimisation and inference very easily. *Mathematical Matters of the Heart*.

[Mossong et al., 2017] Mossong, J., Hens, N., Jit, M., Beutels, P., Auranen, K., Mikolajczyk, R., Massari, M., Salmaso, S., Tomba, G. S., Wallinga, J., et al. (2017). POLYMOD social contact data. 1:10–5281.

[Neal, 2011] Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2(11):2.

[Nicholson et al., 2022] Nicholson, G., Lehmann, B., Padellini, T., Pouwels, K. B., Jersakova, R., Lomax, J., King, R. E., Mallon, A.-M., Diggle, P. J., Richardson, S., Blangiardo, M., and Holmes, C. (2022). Improving local prevalence estimates of SARS-CoV-2 infections using a causal debiasing framework. *Nature Microbiology*, 7(1):97–107.

[Nishiura and Chowell, 2009] Nishiura, H. and Chowell, G. (2009). The effective reproduction number as a prelude to statistical estimation of time-dependent epidemic trends. In *Mathematical and statistical estimation approaches in epidemiology*, pages 103–121. Springer.

[Nishiura et al., 2020] Nishiura, H., Linton, N. M., and Akhmetzhanov, A. R. (2020). Serial interval of novel coronavirus (COVID-19) infections. *International Journal of Infectious Diseases*, 93:284–286.

[OT&P Healthcare, 2022] OT&P Healthcare (2022). COVID-19 timeline of events. `https://www.otandp.com/covid-19-timeline`. Accessed: 22 June 2020.

[Paciorek, 2003] Paciorek, C. J. (2003). *Nonstationary Gaussian processes for regression and spatial modelling*. PhD thesis, Carnegie Mellon University.

[Paciorek and Schervish, 2004] Paciorek, C. J. and Schervish, M. J. (2004). Nonstationary covariance functions for gaussian process regression. In *Advances in Neural Information Processing Systems*, pages 273–280.

[Parag, 2021] Parag, K. V. (2021). Improved estimation of time-varying reproduction numbers at low case incidence and between epidemic waves. *PLOS Computational Biology*, 17(9).

[Parag et al., 2021] Parag, K. V., Cowling, B. J., and Donnelly, C. A. (2021). Deciphering early-warning signals of SARS-CoV-2 elimination and resurgence from limited data at multiple scales. *Journal of the Royal Society Interface*, 18(185).

[Parag and Donnelly, 2020] Parag, K. V. and Donnelly, C. A. (2020). Adaptive estimation for epidemic renewal and phylogenetic skyline models. *Systematic Biology*, 69(6):1163–1179.

[Parag and Donnelly, 2022] Parag, K. V. and Donnelly, C. A. (2022). Fundamental limits on inferring epidemic resurgence in real time using effective reproduction numbers. *PLOS Computational Biology*, 18(4):e1010004.

[Parag et al., 2022] Parag, K. V., Thompson, R. N., and Donnelly, C. A. (2022). Are epidemic growth rates more informative than reproduction numbers? *Journal of the Royal Statistical Society Series A: Statistics in Society*, 185:S5–S15.

[Pitman, 2002] Pitman, J. (2002). Combinatorial stochastic processes. *Lecture Notes in Mathematics*, 1875:7–24.

[Pitman and Yor, 1997] Pitman, J. and Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, pages 855–900.

[Pitzer et al., 2021] Pitzer, V. E., Chitwood, M., Havumaki, J., Menzies, N. A., Perniciaro, S., Warren, J. L., Weinberger, D. M., and Cohen, T. (2021). The impact of changes in diagnostic testing practices on estimates of COVID-19 transmission in the United States. *American Journal of Epidemiology*, 190(9):1908–1917.

[Pons-Salort et al., 2023] Pons-Salort, M., Lambert, B., Kamau, E., Pebody, R., Harvala, H., Simmonds, P., and Grassly, N. C. (2023). Changes in transmission of Enterovirus D68 (EV-D68) in England inferred from seroprevalence data. *Elife*, 12:e76609.

[Pooley et al., 2022] Pooley, C. M., Doeschl-Wilson, A. B., and Marion, G. (2022). Estimation of age-stratified contact rates during the covid-19 pandemic using a novel inference algorithm. *Philosophical Transactions of the Royal Society A*, 380(2233):20210298.

[Prem et al., 2017] Prem, K., Cook, A. R., and Jit, M. (2017). Projecting social contact matrices in 152 countries using contact surveys and demographic data. *PLoS Computational Biology*, 13(9):e1005697.

[Price et al., 2020] Price, D. J., Shearer, F. M., Meehan, M. T., McBryde, E., Moss, R., Golding, N., Conway, E. J., Dawson, P., Cromer, D., Wood, J., Abbott, S., McVernon, J., and McCaw, J. M. (2020). Early analysis of the Australian COVID-19 epidemic. *eLife*, 9.

[Rasmussen, 2003] Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer.

[Sachak-Patwa et al., 2021] Sachak-Patwa, R., Byrne, H. M., Dyson, L., and Thompson, R. N. (2021). The risk of SARS-CoV-2 outbreaks in low prevalence settings following the removal of travel restrictions. *Communications Medicine*, 1(1):39.

[Salvatier et al., 2016] Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55.

[Schäfer and Strimmer, 2005] Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1).

[Schoups et al., 2010] Schoups, G., Vrugt, J., Fenicia, F., and Van De Giesen, N. (2010). Corruption of accuracy and efficiency of Markov chain Monte Carlo simulation by inaccurate numerical implementation of conceptual hydrologic models. *Water Resources Research*, 46(10).

[Schoups and Vrugt, 2010] Schoups, G. and Vrugt, J. A. (2010). A formal likelihood function for parameter and predictive inference of hydrologic models with correlated, heteroscedastic, and non-Gaussian errors. *Water Resources Research*, 46(10).

[Sengupta et al., 2016] Sengupta, B., Friston, K. J., and Penny, W. D. (2016). Gradient-based MCMC samplers for dynamic causal modelling. *NeuroImage*, 125:1107–1118.

[Sharma et al., 2020] Sharma, M., Mindermann, S., Brauner, J., Leech, G., Stephenson, A., Gavenčiak, T., Kulveit, J., Teh, Y. W., Chindelevitch, L., and Gal, Y. (2020). How

robust are the estimated effects of nonpharmaceutical interventions against COVID-19? *Advances in Neural Information Processing Systems*, 33:12175–12186.

[Shen et al., 2004] Shen, Z., Ning, F., Zhou, W., He, X., Lin, C., Chin, D. P., Zhu, Z., and Schuchat, A. (2004). Superspreading sars events, Beijing, 2003. *Emerging Infectious Diseases*, 10(2):256.

[Shertzer et al., 2002] Shertzer, K. W., Ellner, S. P., Fussmann, G. F., and Hairston Jr, N. G. (2002). Predator-prey cycles in an aquatic microcosm: testing hypotheses of mechanism. *Journal of Animal Ecology*, pages 802–815.

[Soltesz et al., 2020] Soltesz, K., Gustafsson, F., Timpka, T., Jaldén, J., Jidling, C., Heimerson, A., Schön, T. B., Spreco, A., Ekberg, J., Dahlström, Ö., et al. (2020). On the sensitivity of non-pharmaceutical intervention models for SARS-CoV-2 spread estimation. *medRxiv*.

[Stan Development Team, 2016] Stan Development Team (2016). Stan modeling language users guide and reference manual. *Technical report*.

[State of Hawaii Department of Health Disease Outbreak Control Division, 2022] State of Hawaii Department of Health Disease Outbreak Control Division (2022). Hawaii COVID-19 Summary Metrics.

[Stewart, 2011] Stewart, D. E. (2011). *Dynamics with Inequalities: impacts and hard constraints*. SIAM.

[Storen and Corrigan, 2020] Storen, R. and Corrigan, N. (2020). COVID-19: a chronology of state and territory government announcements (up until 30 June 2020). `https://www.aph.gov.au/About_Parliament/Parliamentary_Departments/Parliamentary_Library/pubs/rp/rp2021/Chronologies/COVID-19StateTerritoryGovernmentAnnouncements#_Toc52275800`. Accessed: 22 June 2020.

[Svensson, 2007] Svensson, Å. (2007). A note on generation times in epidemic models. *Mathematical Biosciences*, 208(1):300–311.

[Teh, 2010] Teh, Y. W. (2010). Dirichlet process. *Encyclopedia of Machine Learning*, 1063:280–287.

[Teymur et al., 2018] Teymur, O., Lie, H. C., Sullivan, T., and Calderhead, B. (2018). Implicit probabilistic integrators for ODEs. *Advances in Neural Information Processing Systems*, 31.

[Teymur et al., 2016] Teymur, O., Zygalakis, K., and Calderhead, B. (2016). Probabilistic linear multistep methods. *Advances in Neural Information Processing Systems*, 29.

[Thompson et al., 2019] Thompson, R., Stockwin, J., van Gaalen, R. D., Polonsky, J., Kamvar, Z., Demarsh, P., Dahlqwist, E., Li, S., Miguel, E., Jombart, T., Lessler, J., Cauchemez, S., and Cori, A. (2019). Improved inference of time-varying reproduction numbers during infectious disease outbreaks. *Epidemics*, 29.

[Thompson et al., 2020] Thompson, R. N., Hollingsworth, T. D., Isham, V., Arribas-Bel, D., Ashby, B., Britton, T., Challenor, P., Chappell, L. H., Clapham, H., Cunniffe, N. J., et al. (2020). Key questions for modelling COVID-19 exit strategies. *Proceedings of the Royal Society B*, 287(1932):20201405.

[Timonen et al., 2022] Timonen, J., Siccha, N., Bales, B., Lähdesmäki, H., and Vehtari, A. (2022). An importance sampling approach for reliable and efficient inference in bayesian ordinary differential equation models. *arXiv preprint arXiv:2205.09059*.

[Tsang et al., 2021] Tsang, T. K., Wu, P., Lau, E. H., and Cowling, B. J. (2021). Accounting for imported cases in estimating the time-varying reproductive number of coronavirus disease 2019 in Hong Kong. *The Journal of Infectious Diseases*, 224(5):783–787.

[Unwin et al., 2021] Unwin, H. J. T., Routledge, I., Flaxman, S., Rizoiu, M.-A., Lai, S., Cohen, J., Weiss, D. J., Mishra, S., and Bhatt, S. (2021). Using Hawkes Processes to model imported and local malaria cases in near-elimination settings. *PLoS Computational Biology*, 17(4):e1008830.

[van der Vegt et al., 2022] van der Vegt, S. A., Dai, L., Bouros, I., Farm, H. J., Creswell, R., Dimdore-Miles, O., Cazimoglu, I., Bajaj, S., Hopkins, L., Seiferth, D., Gavaghan, D., and Lambert, B. (2022). Learning transmission dynamics modelling of COVID-19 using comomodels. *Mathematical Biosciences*, 349:108824.

[Van Kerkhove et al., 2015] Van Kerkhove, M. D., Bento, A. I., Mills, H. L., Ferguson, N. M., and Donnelly, C. A. (2015). A review of epidemiological parameters from Ebola outbreaks to inform early public health decision-making. *Scientific Data*, 2(1):1–10.

[Verity et al., 2020] Verity, R., Okell, L. C., Dorigatti, I., Winskill, P., Whittaker, C., Imai, N., Cuomo-Dannenburg, G., Thompson, H., Walker, P. G., Fu, H., et al. (2020). Estimates of the severity of coronavirus disease 2019: a model-based analysis. *The Lancet Infectious Diseases*, 20(6):669–677.

[Virtanen et al., 2020] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

[Vrugt et al., 2009] Vrugt, J. A., Ter Braak, C., Diks, C., Robinson, B. A., Hyman, J. M., and Higdon, D. (2009). Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *International Journal of Nonlinear Sciences and Numerical Simulation*, 10(3):273–290.

[Wallinga and Teunis, 2004] Wallinga, J. and Teunis, P. (2004). Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. *American Journal of Epidemiology*, 160(6):509–516.

[Weiss, 2013] Weiss, H. H. (2013). The SIR model and the foundations of public health. *Materials Mathematics*, pages 1–17.

[Whittaker et al., 2020] Whittaker, D. G., Clerx, M., Lei, C. L., Christini, D. J., and Mirams, G. R. (2020). Calibration of ionic and cellular cardiac electrophysiology models. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 12(4):e1482.

[Wiener, 1950] Wiener, N. (1950). *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT press Cambridge, MA.

[Williams and Rasmussen, 2006] Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

[Wu et al., 2020] Wu, B., Lei, Z.-Y., Wu, K.-L., He, J.-R., Cao, H.-J., Fu, J., Chen, F., Chen, Y., Chen, B., Zhou, X.-L., et al. (2020). Compare the epidemiological and clinical features of imported and local COVID-19 cases in Hainan, China. *Infectious Diseases of Poverty*, 9(05):105–115.

[Xinhua News Agency, 2020] Xinhua News Agency (2020). Carrie Lam: The Hong Kong SAR Government will take compulsory quarantine measures to deal with the risk of foreign epidemic importation. Accessed: 13 Sept 2022.

[Zhu et al., 1997] Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J. (1997). Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560.