INTERFACE

Research

THE ROYAL SOCIETY
PUBLISHING

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

Richard Creswell[1], Katherine M. Shepherd[1], Ben Lambert[2], Gary R. Mirams[3], Chon Lok Lei[4], Simon Tavener[5], Martin Robinson[1] and David J. Gavaghan[1]

[1]Department of Computer Science, and [2]Department of Statistics, University of Oxford, Oxford, Oxfordshire, UK
[3]School of Mathematical Sciences, University of Nottingham, Nottingham, Nottinghamshire, UK
[4]Institute of Translational Medicine and Department of Biomedical Sciences, Faculty of Health Sciences, University of Macau, Taipa, Macao
[5]Department of Mathematics, Colorado State University, Fort Collins, CO, USA

RC, 0000-0002-9491-1897; KMS, 0009-0000-9064-486X; GRM, 0000-0002-4569-4312

Most ordinary differential equation (ODE) models used to describe biological or physical systems must be solved approximately using numerical methods. Perniciously, even those solvers that seem sufficiently accurate for the *forward problem*, i.e. for obtaining an accurate simulation, might 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 might 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 change point 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 run-off 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.

1. Introduction

Many scientific phenomena involve time-varying signals or outputs. These phenomena are often believed to obey some parametric model, whose parameter values are *a priori* unknown but can be inferred from observed data. In this paper, we present and analyse the key challenges that arise in parameter inference when models involve ordinary differential equations (ODEs). ODEs are used throughout the biological and physical sciences to express dynamic processes; a few examples among the myriad of their application areas include epidemiology [1], hydrology [2], cardiac electrophysiology [3] and population dynamics [4].

In general, the differential equations used in scientific applications cannot be solved analytically. However, a wide range of computational methods have been developed to obtain a numerical approximation of their solutions. (Solving the differential equation for a particular value of the parameters is known as the *forward problem*.) While numerical algorithms to solve the forward problem introduce error, the properties of this error are generally well understood and can be controlled. In solvers using a fixed time step (discussed in §3.2), the error can be reduced by decreasing the size of the time step [5]. In solvers in which the time step is set adaptively (discussed in §3.3), 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) [6].

Our focus in this paper 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. Widely used approaches to the inverse problem include optimization of an objective function that measures the quality of fit between the model and the data (e.g. maximum likelihood), or Bayesian approaches that 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. Previous work (e.g. [7]) has demonstrated that numerical error in the forward problem is liable to introduce errors in estimated parameters, and techniques have been developed to model the uncertainty in forward solutions and parameter estimates arising from the approximation errors made by the solver [8]. One approach for analysing the errors in the inverse problem that arise because of numerical approximation in the forward problem, is presented in [9]. In this approach, bounds are computed on the Hellinger distance between the true parameter posterior distribution and the posterior arising from a numerically approximate forward solution. For certain conditions on the forward solution and its numerical approximation (which our example models in this paper satisfy), it can be shown that the error in the forward solution transfers to a bound on the Hellinger distance between the true and numerically approximate posteriors [9, theorem 4.6, eqn 4.12]. This analysis, however, provides an asymptotic result, and achieving accurate posteriors in practical problems is not necessarily straightforward. In particular, as we illustrate in this paper, modest solver step sizes can produce accurate forward solutions but highly inaccurate log-likelihood surfaces which are intractable for inference algorithms such as MCMC sampling.

Our paper provides an empirical analysis, involving both synthetic and real data, demonstrating the various characteristics of errors in parameter likelihood surfaces that can appear due to solver inaccuracy. Our results demonstrate the importance of ensuring that ODE solvers used during an inference algorithm are appropriately tuned for inference, which often requires the ODE to be solved more accurately than for acceptable forward solutions.

The rest of this paper is organized as follows. In §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 that have low noise and rapid nonlinear dynamics. In §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. For both classes of solver, we illustrate the effects that solver inaccuracy can have on inference, and illustrate this using synthetic data. Additionally, in electronic supplementary material, §S3, we study how smoothing approximations can reduce the influence of numerical error on computation of the likelihood. Finally, in §4 and §5, we consider inference of ODE models for real data series. In §4, 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 §5, 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.

2. Effects of numerical error on computation of the log-likelihood

2.1. Log-likelihood function for an ordinary differential equation 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

$$\frac{\mathrm{d}x}{\mathrm{d}t} = h(t, x, \theta) \tag{2.1}$$

and

$$x(0) = x_0, \tag{2.2}$$

for some function h that is informed by the relevant scientific theory and parametrized by the (potentially unknown) values $\theta \in \Theta \subset \mathbb{R}^m$. (In some inference problems, the initial value x_0 is also an unknown parameter, or a function of unknown parameters.) Equation (2.1) has been written as a *first-order* equation (i.e. only involving the first-order derivative of x); higher order differential equations can be rewritten as systems of first-order equations.

Deterministic forward models never fully explain the variation in real observations. To make the forward model a feasible explanation of the data, an additional stochastic component representing elements not modelled as part of the differential equation model (for example, processes involved in the measurement of the signal, or a model of measurement error occurring in the sensor devices used to collect the measurements) is included. This stochastic component is often included in additive form, yielding the proposed data generating process

$$y_i = g(x(t_i; \theta)) + \epsilon_i, \tag{2.3}$$

where ϵ_i is a mean-zero random variable specifying the noise process. Many choices for ϵ_i are possible, and in applied data analysis the assumed distribution of this variable should be chosen carefully, as misspecification of ϵ_i can cause inaccurate inference results [10]. In this paper, for simplicity, we choose to generate synthetic data using the independent and identically distributed (IID) Gaussian distribution. The IID Gaussian noise distribution is given by

$$\epsilon_i \overset{\text{IID}}{\sim} N(0, \sigma^2), \tag{2.4}$$

with the parameter σ representing the standard deviation of the noise process, that is also inferred from the data together with the model parameters θ. Choosing a particular noise process enables the joint probability of the observations to be expressed as a function of the parameter values θ—this is known as a likelihood. For an IID Gaussian noise process (equation (2.4)) and one-dimensional data points y_i, the log-likelihood for time-series data $\{y_i\}_{i=1}^N$ takes the form

$$\log p(y_1, \ldots, y_N | \theta, \sigma) = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - g(x(t_i; \theta)))^2. \tag{2.5}$$

The likelihood expresses the quality of the fit between the model output and the data, with higher values of the likelihood indicating a superior fit. Thus, the values of θ most compatible with data can be found by maximizing the likelihood with respect to θ (i.e. the method of maximum likelihood). Alternatively, the likelihood can be used together with a prior distribution on the parameters ($p(\theta)$) to infer the posterior distribution according to Bayes' theorem

$$p(\theta | y_1, \ldots, y_N) = \frac{p(y_1, \ldots, y_N | \theta)p(\theta)}{p(y)}.$$

In this paper, we consider the typical case where the posterior cannot easily be expressed in closed form but can be approximated using MCMC sampling methods [11].

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, equation (2.5). 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 might 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 equation (2.5), 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 $\mathcal{L}'$, which is given by

$$\mathcal{L}' = -\frac{N}{2}\log(2\pi) - \frac{N}{2}\log(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - g(\hat{x}_i))^2. \tag{2.6}$$

Theorem 2.1. *Assuming Lipschitz continuity of the observation function g with Lipschitz constant K, the difference between $\mathcal{L}'$ and $\mathcal{L}$ (both computed at the parameter values θ, which are assumed to be those which generated the data) is bounded according to*

$$|\mathcal{L} - \mathcal{L}'| \leq \sum_{i=1}^{N}\left(\frac{K^2}{2\sigma^2}|e(t_i)|^2 + \frac{K}{\sigma^2}|e(t_i)||y_i - g(x(t_i; \theta))|\right). \tag{2.7}$$

Proof. The proof is presented in electronic supplementary material, §S1. ∎

We observe an inverse relationship between σ and the bound of $|\mathcal{L} - \mathcal{L}'|$ 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 to be more severe at smaller values of σ. Equation (2.7) also indicates that more severe biases can be expected when N is larger—i.e. there are more observations. Additionally, at a fixed level of σ, we expect the bias in the log-likelihood to decrease as the global truncation error is decreased. In electronic supplementary material, §S2, we additionally derive the distribution of the error in the likelihood, and show that $\mathbb{E}[\mathcal{L} - \mathcal{L}'] > 0$: the numerical approximation of the likelihood will, on average, underestimate the true likelihood.

## 3. Effects of ordinary differential equation 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 unknown parameters: (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 paper 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

$$\frac{d}{dt}\begin{pmatrix} x \\ \dot{x} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \frac{F(t)}{m} - \frac{c}{m}\dot{x} - \frac{k}{m}x \end{pmatrix}, \tag{3.1}$$

where $\dot{x} = dx/dt$.
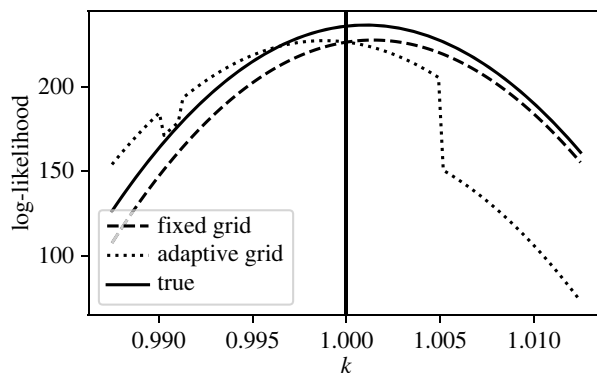
Figure 1. Comparison of log-likelihood surfaces calculated using fixed step and adaptive step solvers. Log-likelihood for the parameter k calculated from data generated from the oscillator model equation (3.1), with all other parameters held at their true values. The log-likelihood was calculated from equation (2.5) using an adaptive step RK5(4) solver with relative tolerance set to 10^{-8} (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).

3.1. Fixed step and adaptive step ordinary differential equation solvers

A wide range of numerical algorithms have been developed to obtain approximate solutions to initial value problems (IVPs) of the form given in equations (2.1) and (2.2). 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 that 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 `atol + rtol|x|` [12]. For the results presented in this paper, 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. Although the phenomena we analyse in this paper are liable to arise from a variety of adaptive step size solvers, (including, for example, the CVODE multistep method [13, fig. 3]), 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 [6]. 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, [5]). 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) [12].

3.1.1. 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 might employ different grids at different parameter values, these two classes of solvers differ in the characteristics of the error that they tend to introduce into the likelihood function.

We illustrate this by computing the likelihood surface for the k parameter in the oscillator problem, equation (3.1). In total, 75 evenly spaced data points were generated from and including $t = 0$ to $t = 50$ from the model with an accurate solver (the RK5(4) solver with relative tolerance set to 10^{-8}), using true parameter values $k = 1$, $c = 0.2$, $m = 1$, 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 RK5(4) solver with relative tolerance set to 10^{-8} 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 1.
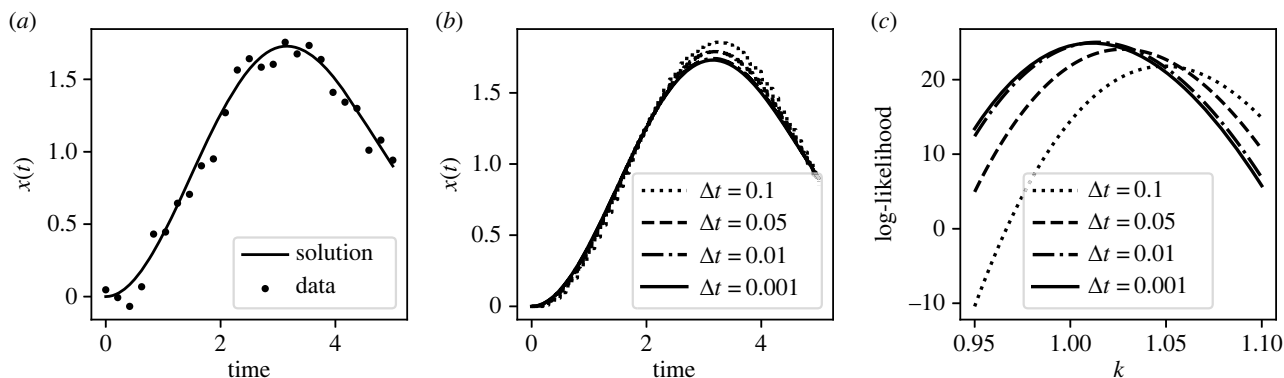
Figure 2. Damped oscillator inference using Forward Euler. (*a*) Synthetic data for the damped driven oscillator. The curved line indicates the accurate solution to the ODE with these parameters, while the points indicate the noisy data. (*b*) Solution for oscillator computed using a Forward Euler solver with four different choices for the time step Δt. (*c*) 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 equation (2.5) using a Forward Euler solver with four different choices for the time step Δt.

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 that is shifted relative to the true one, but retains the smooth, unimodal shape. However, the adaptive step solver results in a log-likelihood surface that in addition to being shifted exhibits jagged, discontinuous fluctuations. In the remainder of §3, we examine these two phenomena in more detail.

3.2. Fixed time step solvers

3.2.1. Forward Euler solver

One of the simplest numerical solvers for ordinary differential equations is the Forward Euler method with a uniform step size Δt. 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. [14,15]). Whether these applications are representative of the use of Forward Euler more generally is unclear, but our results in §4 indicate that such choices of Δt can alter both forward model solutions and parameter inference results.

3.2.2. 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 equation (3.1). In total, 25 evenly spaced data points were generated from and including $t = 0$ 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 $k = 1$, $c = 0.2$, $m = 1$, 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 2 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 with the time scale of the dynamics of the system and the system with $F(t) = 1$ contains no discontinuities or other challenging features. Given that Forward Euler is a first-order method—as the step size is refined, the global truncation error introduced by the method is approximately proportional to Δt—its poor performance is unsurprising.

As the step size is refined, the log-likelihood curves converge. This suggests a diagnostic technique that 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.

3.3. 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} [16,17]. Next, we investigate the origin of the jagged likelihoods using synthetic data from the oscillator model described in equation (3.1).
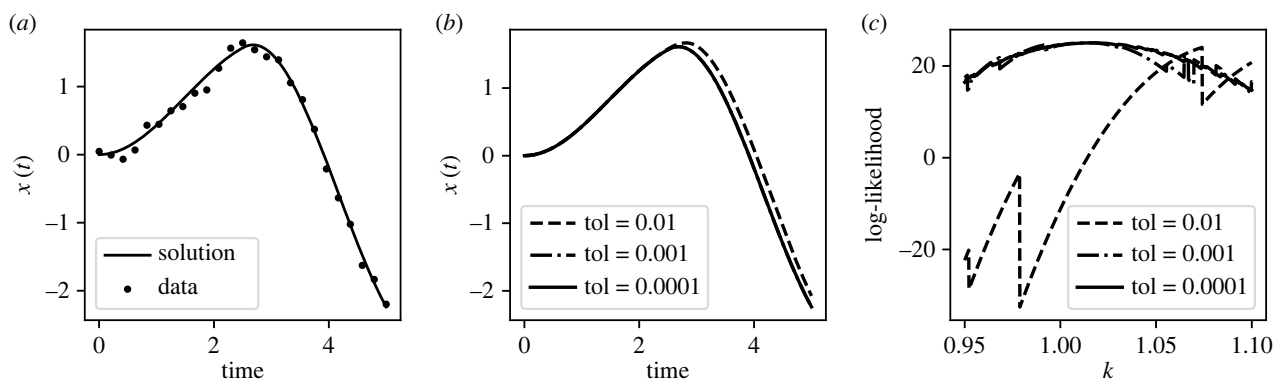
Figure 3. Damped oscillator inference using adaptive time step Runge–Kutta. (*a*) Synthetic data for the damped driven oscillator. The curved line indicates the accurate solution to the ODE with these parameters, while the points indicate the noisy data. (*b*) Solution for oscillator computed using an RK5(4) solver with three different choices for the relative tolerance (indicated by tol in the legend). (*c*) 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 equation (2.5) using an RK5(4) solver with three different choices for the tolerance.

3.3.1. 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 §3 (equation (3.1)). Here, we set the input stimulus according to

$$F(t) = \begin{cases} 1, & t < t_{\text{change}}, \\ f_1, & t \geq t_{\text{change}}. \end{cases} \qquad (3.2)$$

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

Our subsequent analyses in this section employ various discontinuities in $F(t)$ to control the difficulty of the ODE system, and we show that adaptive step solvers can introduce substantial, jagged errors into the likelihood surface. However, it is not only ODEs with discontinuous r.h.s. that are affected by similar phenomena. In electronic supplementary material, §S4, we additionally study the interplay between a continuous (but rapidly changing) $F(t)$ and an adaptive step size solver, and show similar qualitative results.

First, we consider the problem where $f_1 = -1$ and $t_{\text{change}} = 2.5$ for different choices of the RK5(4) solver tolerance. In total, 25 evenly spaced data points were generated from and including $t = 0$ 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 various tolerances. These results are shown in figure 3. At insufficient tolerances, the log-likelihood surface exhibits significant erroneous jaggedness. Notably, visual changes between the forward simulations are minor even at tolerances that 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 4, we fix $t_{\text{change}} = 25$ and consider four different values of f_1 and plot the likelihood surface for the model parameter k calculated according to an RK5(4) solver with `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 r.h.s. that could be made more tractable for inference using smoothing approximations, which we analyse in electronic supplementary material, §S3.) 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 5. For this study, the oscillator model equation (3.1) was again used. In total, 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 5. 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.
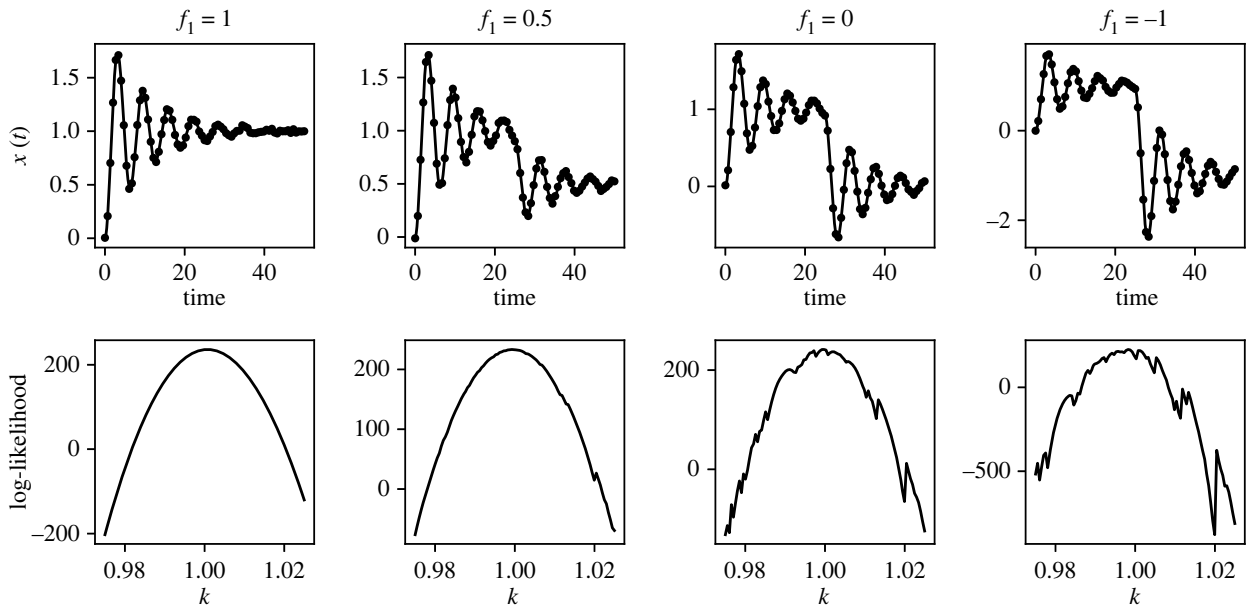
Figure 4. Damped oscillator model: forward simulations and inference using an adaptive solver. Time-series data and parameter likelihood surfaces are shown for four values of f_1 in the oscillator problem: equations (3.1) and (3.2). For each value of f_1, the top plot shows the accurate ODE solution (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 `rtol` $= 10^{-3}$.
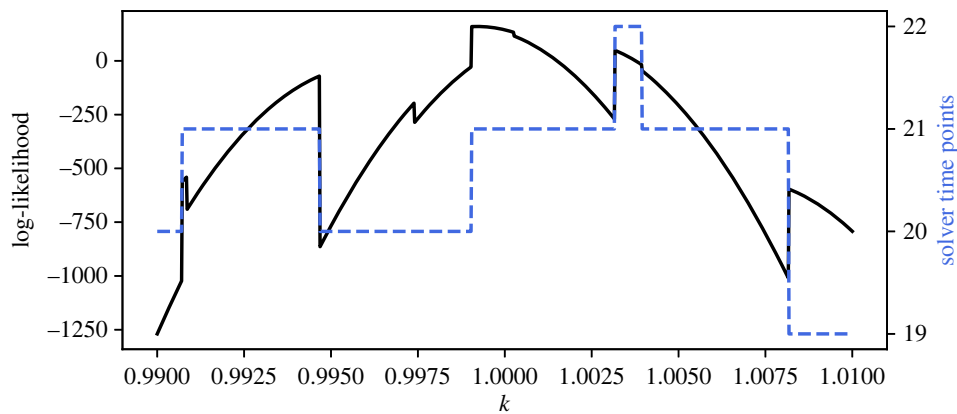
Figure 5. Damped oscillator model: likelihood discontinuities caused by variation in the number of adaptive steps. The log-likelihood surface for the parameter k in the oscillator problem (black solid line) and the number of time points used by the adaptive step size ODE solver in the calculation of each value of the likelihood (blue dashed line) are shown. An RK5(4) solver was used with `rtol` $= 10^{-3}$.

3.3.2. 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 [11]. 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

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

where $p(\theta^{\text{prop}})$ is the prior and $p(y \mid \theta^{\text{prop}})$ 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. $\log p(y \mid \theta^{\text{prop}}) = \log p(y \mid \theta^{\text{old}}) - c$, 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 4 and 5.

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 can severely restrict the ability of a Metropolis–Hastings sampler (or similar inference algorithm) to traverse the parameter space efficiently.
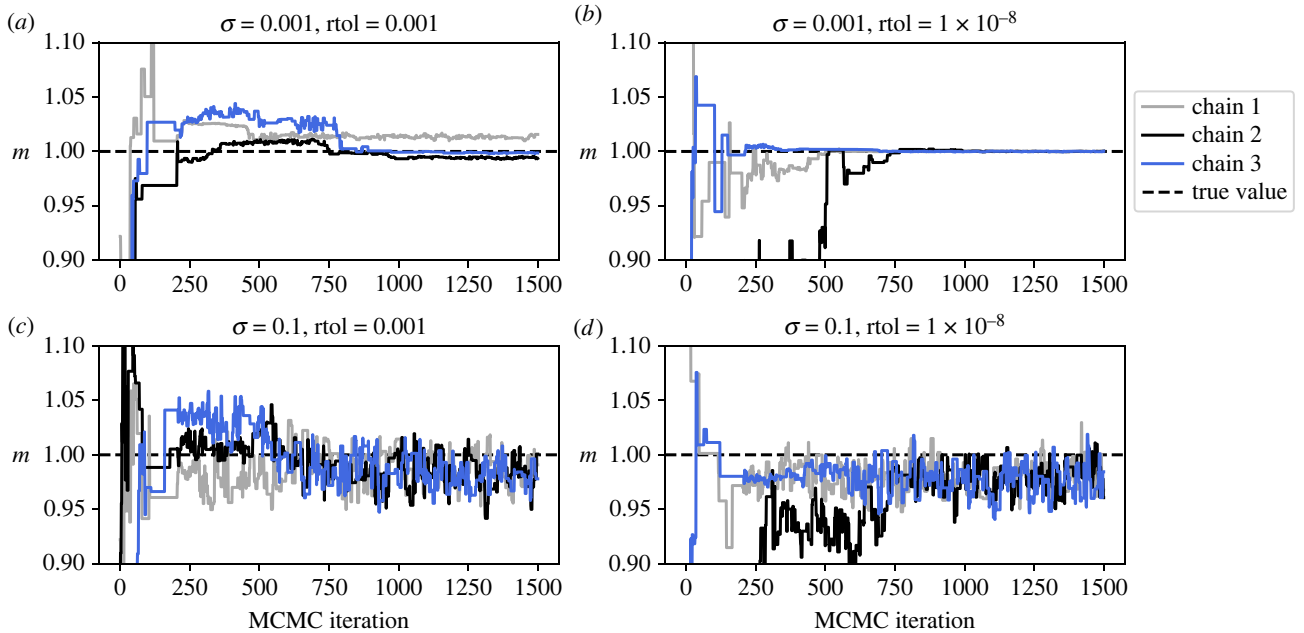
Figure 6. Effect of noise on MCMC convergence. Data were generated according to the same specifications as for figure 4, with $f_1 = -1$, and the indicated values of σ. Inference was performed for the three parameters m, c and k, as well as σ, via adaptive covariance MCMC [19] with three independent 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). In total, 1500 MCMC iterations were performed. The plots show the three chains for the m parameter. (a,c) Forward simulation was performed using the RK5(4) solver with $\mathtt{rtol} = 10^{-3}$. (b,d) Forward simulation was performed using the RK5(4) solver with $\mathtt{rtol} = 10^{-8}$.

3.4. 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 (0.001 and 0.1) to generate the data, fixed $f_1 = -1$, and otherwise generated data exactly as described for figure 4. 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). In total, 1500 iterations of MCMC were performed using the Haario–Bardenet adaptive covariance algorithm as implemented in PINTS to sample from the posterior [18,19]. The MCMC chains for the m parameter are plotted in figure 6a,c using the RK5(4) solver with $\mathtt{rtol} = 10^{-3}$, while figure 6b,d shows the chains using the same solver but with more accurate tolerances of $\mathtt{rtol} = 10^{-8}$.

At the lowest noise level considered ($\sigma = 0.001$), 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 4. 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] this can be explained by our bound given in equation (2.7): that larger σ values lead to gentler variation in the log-likelihood surface and so easier exploration by inference algorithms.

4. Fixed step solvers applied to a susceptible–infected–recovered change point model of the spread of COVID-19 in Germany

A widely used class of differential equation models in epidemiology are compartmental models. These 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 [1]. A simple yet commonly used example is the susceptible–infected–recovered (SIR) model [20]. 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:

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\lambda \frac{SI}{N}, \tag{4.1}$$

$$\frac{\mathrm{d}I}{\mathrm{d}t} = \lambda \frac{SI}{N} - \mu I \tag{4.2}$$

and

$$\frac{\mathrm{d}R}{\mathrm{d}t} = \mu I, \tag{4.3}$$

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 $(S(0), I(0), R(0))$, where $I(0)$ 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 grow, and the epidemic will eventually infect the entire population (barring a change in λ or μ); 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 [15]. The authors used the model equations (4.1)–(4.3), discretized with a 1-day time step, equivalent to a Forward Euler solver with $\Delta t = 1$

$$S_t = S_{t-1} - \lambda(t)\Delta t \frac{S_{t-1}I_{t-1}}{N}, \tag{4.4}$$

$$I_t = I_{t-1} + \lambda(t)\Delta t \frac{S_{t-1}I_{t-1}}{N} - \mu \Delta t I_{t-1} \tag{4.5}$$

and

$$R_t = R_{t-1} + \mu \Delta t I_{t-1}. \tag{4.6}$$

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 characterized 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 characterized by two parameters f_w and Φ_w. This significant periodicity probably arises from processes involved in the reporting of COVID-19 cases and deaths [21]. Specifically, cases C_t are modelled by

$$C_t = (1 - f(t))I_{t-D}^{\text{new}}, \tag{4.7}$$

where

$$f(t) = (1 - f_w)\left(1 - \left|\sin\left(\frac{\pi}{7}t - \frac{1}{2}\Phi_w\right)\right|\right), \tag{4.8}$$

where $I_t^{\text{new}} = S_{t-1} - S_t$. Dehning et al. [15] 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 can be computed using a range of different time steps. The prior distributions for the parameters are given in electronic supplementary material, table S1.

4.1. 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. [15] used to generate their fig. 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 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 (equations (4.4)–(4.7)), with one change point, and pre-specified values of λ_0 and λ_1.

As in [15], 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
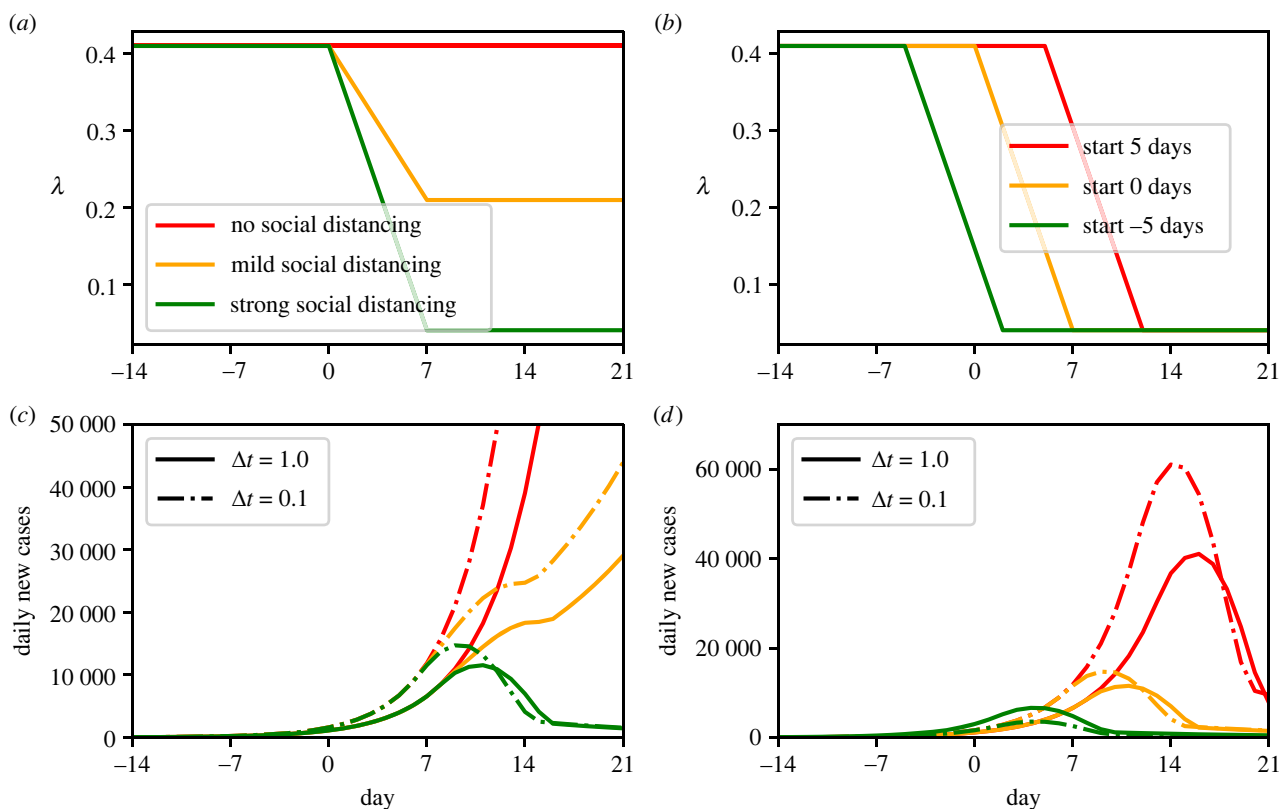
Figure 7. COVID-19 model: forward simulations using Forward Euler. In (*a*–*d*), the top panels (*a,b*) shows three different pre-specified trajectories of $\lambda(t)$, and the bottom panels (*c,d*) shows the number of daily cases resulting from these trajectories for each choice of the time step Δt.

considered, which each yield two sets of simulations: one corresponding to Forward Euler with $\Delta t = 1$ day (as in [15]) and another with $\Delta t = 0.1$ days. We choose 0.1 days as the more accurate comparator method, as further refinement of the step size yields little change in forward solutions or inference results but is increasingly costly to run. The results of this are shown in figure 7a. Our second set of simulations, shown in figure 7b, 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 using a large time step generally leads to a substantial underestimation of case counts for a given choice of $\lambda(t)$, particularly during the (crucial) growth phase of the epidemic.

## 4.2. Effect of time step on the posterior distributions

We also studied the effect of the time step on parameter inference for the full model (equations (4.4)–(4.8)) using the German daily cases data from 2 March to 21 April 2020 as was done in [15]. Inference was performed using the PyMC3 No-U-Turn MCMC sampler (NUTS) [11,22] using the model developed in [15], modified to allow the 0.1-day step size. To initialize the chains, automatic differentiation variational inference [23] as implemented in PyMC3 [22] 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 $\hat{R} < 1.05$ [11]. These results are shown in figure 8.

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 1-day time step results in overestimation of R_0 (by approx. 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 7), 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.

5. Numerical errors arising in rainfall run-off 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 run-off models used in hydrology [24,25].
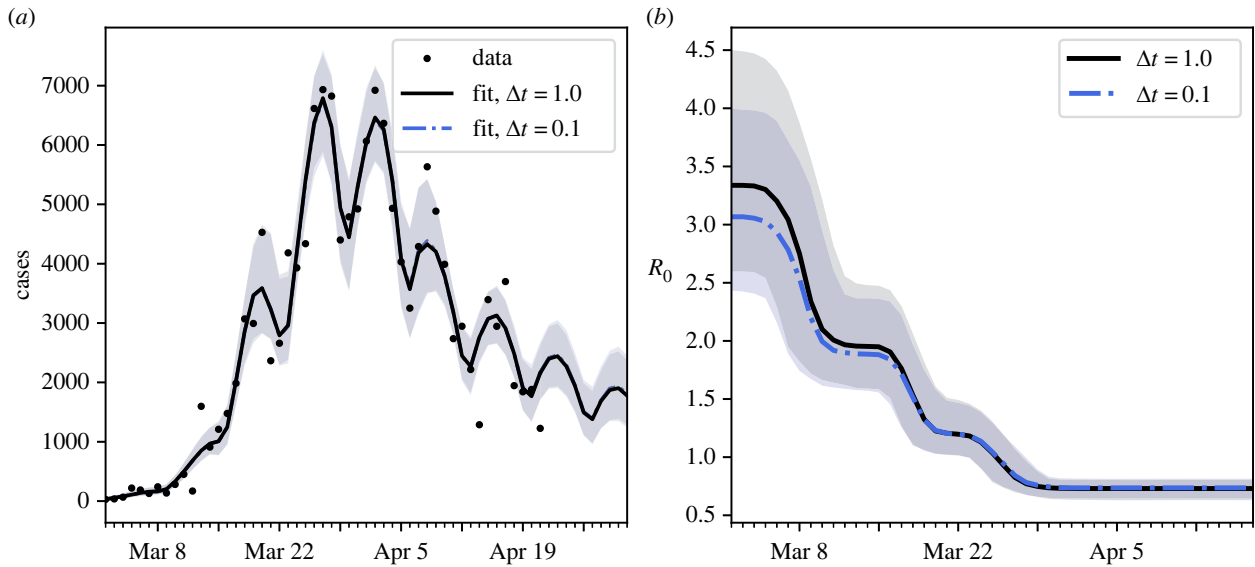
Figure 8. COVID-19 model: inference using Forward Euler. (*a*) Real data and model fits for the number of daily COVID-19 cases in Germany over the period 2 March 2020 to 21 April 2020. Note that the model fits for $\Delta t = 1$ and $\Delta t = 0.1$ overlap almost completely. (*b*) Inferred basic reproduction number over time for the Germany COVID-19 data, using the SIR model with change points in λ (equations (4.4)–(4.8)) 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.

Rainfall run-off 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:

$$\frac{\mathrm{d}S_i}{\mathrm{d}t} = \mathrm{Precip}(t) - \mathrm{InterceptEvap}(t) - \mathrm{EffectPrecip}(t), \tag{5.1}$$

$$\frac{\mathrm{d}S_u}{\mathrm{d}t} = \mathrm{EffectPrecip}(t) - \mathrm{UnsatEvap}(t) - \mathrm{Percolation}(t) - \mathrm{Runoff}(t), \tag{5.2}$$

$$\frac{\mathrm{d}S_s}{\mathrm{d}t} = \mathrm{Percolation}(t) - \mathrm{SlowStream}(t), \tag{5.3}$$

$$\frac{\mathrm{d}S_f}{\mathrm{d}t} = \mathrm{Runoff}(t) - \mathrm{FastStream}(t) \tag{5.4}$$

and

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \mathrm{SlowStream}(t) + \mathrm{FastStream}(t), \tag{5.5}$$

Each term in this equation is defined in electronic supplementary material, table S2, and the seven unknown parameters of the model and their prior distributions are defined in electronic supplementary material, table S3. The data consist of daily streamflow measurements ($\mathrm{d}z/\mathrm{d}t$), and the authors assume an IID Gaussian likelihood with unknown standard deviation σ.

Previous work has shown that using large time steps with such hydrological models can bias inferences [2]. We show that using an adaptive step size method (as suggested in [24]) can also cause inaccurate inference results, unless the error is tightly controlled.

Using an accurate ODE solver (the CVODE multistep solver from the SUNDIALS library [26] with `rtol = atol = 10^{-7}`), we obtained the posterior distributions for the seven parameters of the model, using United States Geological Survey (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 [18,27], using six chains with each initialized by a sample from the prior (electronic supplementary material, table S3). In total, 25 000 MCMC iterations were performed, and convergence of the chains was assessed by requiring that $\hat{R} < 1.05$ [11]. In figure 9, 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 `rtol = 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 §3, as rapid changes in the r.h.s. cause spurious jaggedness in the computed likelihood. The likelihoods calculated using the more accurate solver have similar broad-scale shapes but are smooth enough for accurate inference to be performed.

6. 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 can manifest through the presence of phantom
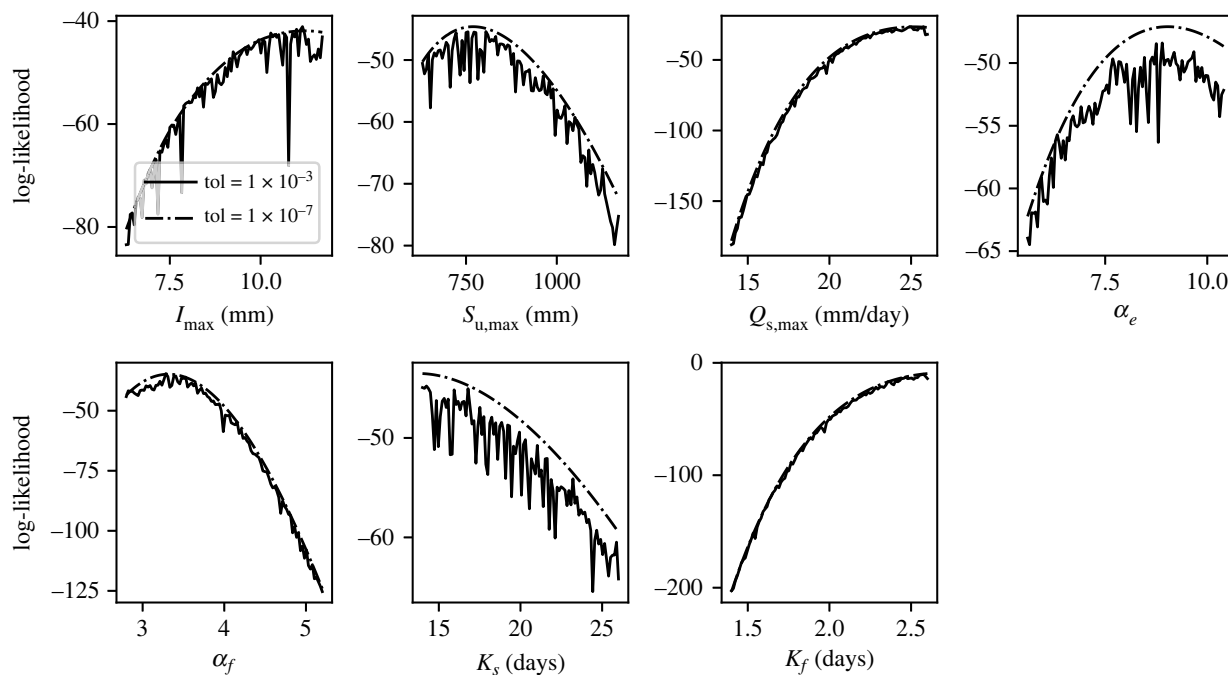
Figure 9. Rainfall run-off model: inference using accurate and inaccurate adaptive solvers. Here, we plot the likelihood surface for each parameter for the rainfall run-off model defined by equations (5.1)–(5.5). For each parameter, the solid line indicates the likelihood calculated using an RK3(2) adaptive solver with `rtol = atol = ` 10^{-3}, while the dashed line indicates the likelihood calculated using the CVODE adaptive solver with `rtol = atol = ` 10^{-7}.

jaggedness in the likelihood surface, which can wreak havoc for inference algorithms attempting to navigate the surface. They might also lead researchers to falsely conclude that a model is unidentifiable, when, for example, the chains in an MCMC run fail to mix. Researchers might then choose 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 that seem good enough for forward simulation are probably insufficient for solving inverse problems. For example, a default relative tolerance of 10^{-3} was insufficient for both the synthetic data and real data studied in §§3.3 and 5. 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 this paper, we focused our attention on two simple solvers: Forward Euler with a uniform grid and adaptive step size RK5(4). In practice, neither of these solvers may be ideal for a particular problem. More powerful solvers, such as implicit solvers which are known to perform more accurately on stiff problems, or solvers which have been augmented to better handle discontinuities (see electronic supplementary material, §S5), could help to reduce the computational runtime required to achieve sufficient accuracy for forward simulation or inference. However, regardless of the solver employed, practitioners should be aware that tuning their solver to achieve accurate forward simulations may still be insufficiently accurate for inference. For example, jagged errors in an objective function for the best fit parameters have been observed that arise when insufficient tolerances are supplied to the CVODE solver—a variable order multistep method using backwards differentiation formulae for stiff problems, which is significantly more powerful than the Forward Euler and RK5(4) solvers [13, fig. 3].

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 optimization algorithms could be augmented by monitoring for such jumps and warning the user should they occur.

ODEs involving discontinuous r.h.s. functions are known to be particularly challenging to solve accurately. Our results indicate that r.h.s. 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 electronic supplementary material, §S5 also indicate that errors in the likelihood arising from discontinuous r.h.s. 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 can 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 that must be controlled. ODE solvers that control the error on the log-likelihood directly would avoid much of the problems highlighted in this paper, and we suggest this as a fruitful research direction.

Endnote

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

References

1. van der Vegt SA *et al.* 2022 Learning transmission dynamics modelling of COVID-19 using comomodels. *Math. Biosci.* **349**, 108824. (doi:10.1016/j.mbs.2022.108824)

2. Kavetski D, Kuczera G, Franks SW. 2003 Semidistributed hydrological modeling: a 'saturation path' perspective on TOPMODEL and VIC. *Water Resour. Res.* **39**, 1246. (doi:10.1029/2003WR002122)

3. Whittaker DG, Clerx M, Lei CL, Christini DJ, Mirams GR. 2020 Calibration of ionic and cellular cardiac electrophysiology models. *Wiley Interdiscip. Rev. Dev. Biol.* **12**, e1482. (doi:10.1002/wsbm.1482)

4. Shertzer KW, Ellner SP, Fussmann GF, Hairston Jr NG. 2002 Predator-prey cycles in an aquatic microcosm: testing hypotheses of mechanism. *J. Anim. Ecol.* **71**, 802–815. (doi:10.1046/j.1365-2656.2002.00645.x)

5. Gautschi W. 1997 *Numerical analysis*. Boston, MA: Springer Science & Business Media.

6. Dormand JR, Prince PJ. 1980 A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.* **6**, 19–26. (doi:10.1016/0771-050X(80)90013-3)

7. Kaipio J, Somersalo E. 2006 *Statistical and computational inverse problems*, vol. 160. New York, NY: Springer Science & Business Media.

8. Conrad PR, Girolami M, Särkkä S, Stuart A, Zygalakis K. 2017 Statistical analysis of differential equations: introducing probability measures on numerical solutions. *Stat. Comput.* **27**, 1065–1082. (doi:10.1007/s11222-016-9671-0)

9. Stuart AM. 2010 Inverse problems: a Bayesian perspective. *Acta Numer.* **19**, 451–559. (doi:10.1017/S0962492910000061)

10. Lambert B, Lei CL, Robinson M, Clerx M, Creswell R, Ghosh S, Tavener S, Gavaghan D. 2023 Autocorrelated measurement processes and inference for ordinary differential equation models of biological systems. *J. R. Soc. Interface* **20**, 20220725. (doi:10.1098/rsif.2022.0725)

11. Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB. 2013 *Bayesian data analysis*. New York, NY: CRC Press.

12. Virtanen P *et al.* SciPy 1.0 Contributors. 2020 SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272. (doi:10.1038/s41592-019-0686-2)

13. Clerx M, Beattie KA, Gavaghan DJ, Mirams GR. 2019 Four ways to fit an ion channel model. *Biophys. J.* **117**, 2420–2437. (doi:10.1016/j.bpj.2019.08.001)

14. Birrell P, Blake J, Van Leeuwen E, Gent N, De Angelis D. 2021 Real-time nowcasting and forecasting of COVID-19 dynamics in England: the first wave. *Phil. Trans. R. Soc. B* **376**, 20200279. (doi:10.1098/rstb.2020.0279)

15. Dehning J, Zierenberg J, Spitzner FP, Wibral M, Neto JP, Wilczek M, Priesemann V. 2020 Inferring change points in the spread of COVID-19 reveals the effectiveness of interventions. *Science* **369**, eabb9789. (doi:10.1126/science.abb9789)

16. Johnstone RH. 2018 Uncertainty characterisation in action potential modelling for cardiac drug safety. PhD thesis, University of Oxford, Oxford, UK.

17. Mirams G. 2018 Numerical errors from ODE solvers can mess up optimisation and inference very easily. *Math. Matters Heart*. See https://mirams.wordpress.com/2018/10/17/ode-errors-and-optimisation/.

18. Clerx M, Robinson M, Lambert B, Lei CL, Ghosh S, Mirams GR, Gavaghan DJ. 2019 Probabilistic inference on noisy time series (PINTS). *J. Open Res. Softw.* **7**, 23. (doi:10.5334/jors.252)

19. Haario H, Saksman E, Tamminen J. 2001 An adaptive metropolis algorithm. *Bernoulli* **7**, 223–242. (doi:10.2307/3318737)

20. Weiss HH. 2013 The SIR model and the foundations of public health. *Mater. Math.* **3**, 1–17.

21. Gallagher K, Creswell R, Gavaghan D, Lambert B. 2023 Identification and attribution of weekly periodic biases in epidemiological time series data. *medRxiv*. (doi:10.1101/2023.06.13.23290903)

22. Salvatier J, Wiecki TV, Fonnesbeck C. 2016 Probabilistic programming in Python using PyMC3. *PeerJ Comput. Sci.* **2**, e55. (doi:10.7717/peerj-cs.55)

23. Kucukelbir A, Tran D, Ranganath R, Gelman A, Blei DM. 2017 Automatic differentiation variational inference. *J. Mach. Learn. Res.* **18**, 1–45.

24. Schoups G, Vrugt J, Fenicia F, 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 Resour. Res.* **46**, W10530. (doi:10.1029/2009WR008648)

25. Schoups G, Vrugt JA. 2010 A formal likelihood function for parameter and predictive inference of hydrologic models with correlated, heteroscedastic, and non-Gaussian errors. *Water Resour. Res.* **46**, W10531. (doi:10.1029/2009WR008933)

26. Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, Woodward CS. 2005 SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw.* **31**, 363–396. (doi:10.1145/1089014.1089020)

27. Vrugt JA, Ter Braak C, Diks C, Robinson BA, Hyman JM, Higdon D. 2009 Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *Int. J. Nonlinear Sci. Numer. Simul.* **10**, 273–290. (doi:10.1515/IJNSNS.2009.10.3.273)

28. rccreswell. 2024 rccreswell/ode_inference: v1.0 (v1.0). *Zenodo*. (doi:10.5281/zenodo.10578920)

29. Onas Dehning Sebastian, Mohr Paul, Spitzner Joao Pinheiro, Neto Johannes, Vp Michael, Osthege Lode. 2024 rccreswell/covid19_inference_forecast: v0.1.2 (v0.1.2). *Zenodo*. (doi:10.5281/zenodo.10578938)

30. Creswell R, Shepherd KM, Lambert B, Mirams GR, Lei CL, Tavener S, Robinson M, Gavaghan DJ. 2024 Understanding the impact of numerical solvers on inference for differential equation models. Figshare. (doi:10.6084/m9.figshare.c.7075553)