Statistical Inference for Ordinary Differential Equations using Gradient Matching

Author: Benn Macdonald

Thesis submitted for the degree of Doctor of Philosophy.

School of Mathematics and Statistics

Supervisor: Professor Dirk Husmeier

February 3, 2017

Acknowledgements

Big thank you to my darling Caroline - where do I even begin? Thank you for being there with me throughout my entire PhD life, for always being so excited when I succeeded and so supportive when things were tough. You've contributed your time and energy to help me proof my work, helped me improve my oral presentation skills, listened to me moan about many (many) things and of course, taken care of the teddy bears when I was working overnight. For all of this, and so much more, thank you.

Thank you to Professor Dirk Husmeier, truly, one could not ask for a better supervisor. If someday I find myself with my own PhD student, I would consider it a massive success to be able to offer even half of the advice and encouragement that I have received.

Thank you to my PhD examiners, Professor Ernst Wit and Dr Vincent Macaulay, for helping me strengthen my thesis, and to Dr Tereza Neocleous for kindly convening my viva.

My mother, Allanna Macdonald, deserves a big mention and thank you for always supporting me. A shining example as a person and parent, your strength and struggles paved the way to my life in academia. Mum, thank you so much.

Thank you to my father, Joe Wilson, for always being so positive about my academic life and so genuinely pleased to hear my news. Your support and moral fibre has given something to our family that no one else could have given, and it means so very much to me.

My appreciation goes to my sisters, Emma and Kara. Emma, your own academic life is what introduced me to the University of Glasgow, which ultimately became my home for both my undergraduate and PhD degrees. Kara, watching your perseverance throughout our childhood was incredibly inspiring. Thank you, both of you.

Finally, I wish to mention Professor William James Oastler Michie, a gentleman, intellectual and my loving grandfather. It saddens me that you were not able to see the finishing of my PhD and you are very much missed. I am proud that we are University of Glasgow alumni.

Contents

List of Figures

List of Tables

Notation

In order to facilitate ease of reading, examples of the notational form found throughout this thesis can be found in Table 1.

Table 1: Examples of the notation used throughout this thesis.

Notation	**Meaning**	**Example**
Bold face uppercase letter or symbol	Matrix	$\mathbf{X}$
Bold face lowercase letter or symbol	Vector	$\boldsymbol{\theta}$
Vector at time t_i	Concentration for all species at time t_i	$\mathbf{y}(t_i)$ or $\mathbf{x}(t_i)$
Vector of concentrations for species "s"	Concentrations for species "s" over all timepoints	$\mathbf{y}_s$ or $\mathbf{x}_s$
Vector of concentrations	Concentrations over all timepoints for one species	$\mathbf{y}$ or $\mathbf{x}$
Lower case letter at time t_i for species "s"	Concentration for species "s" at timepoint t_i	$y_s(t_i)$ or $x_s(t_i)$

1 Introduction

A central objective of current systems biology research is explaining the interactions amongst components in biopathways. A standard approach is to view a biopathway as a network of biochemical reactions, which is modelled as a system of ordinary differential equations (ODEs).

This system can typically be expressed as:

$$\dot{x}_s = \frac{dx_s(t_i)}{dt_i} = f_s(\mathbf{x}(t_i), \boldsymbol{\theta}_s, t_i), \tag{1}$$

where $s \in \{1, \ldots, N\}$ denotes one of N components (referred to throughout as "species") in the biopathway, $x_s(t_i)$ denotes the concentration of species s at time t_i and $\mathbf{x}(t_i)$ is a vector of concentrations of all system components that influence or regulate the concentration of species s at time t_i. If, for example, species s is an mRNA, then $\mathbf{x}(t_i)$ might contain the concentrations of transcription factors (proteins), that regulate the amount of transcription from DNA for that species. The regulation is described by the regulation function f. The type of regulatory interaction depends on the species involved, e.g. f may describe mass action kinetics, Michaelis-Menten kinetics, etc. All of these interactions depend on a vector of kinetic parameters, $\boldsymbol{\theta}_s$. For many biopathways, only a small fraction of $\boldsymbol{\theta}_s$ can be measured in practice. Therefore, in order to understand the dynamics of the biopathway, the majority of these kinetic parameters need to be inferred from observed (typ-

ically noisy and sparse) time course concentration profiles.

Conventional inference methods typically rely on searching the space of $\boldsymbol{\theta}$ values, and at each candidate, numerically solving the ODEs and comparing the output with that observed. After choosing an appropriate noise model, the form of the likelihood is defined, and a measure of similarity between the data signals and the signals described by the current set of ODE parameters can be calculated. This process is repeated, as part of either an iterative optimisation scheme or sampling procedure in order to estimate the parameters. However, the computational costs involved with repeatedly numerically solving the ODEs are usually high.

Several authors have adopted approaches based on gradient matching (e.g. Calderhead et al. [8] and Liang & Wu [26]), aiming to reduce this computational complexity. These approaches are based on the following two-step procedure. At the first step, interpolation is used to smooth the time series data, in order to avoid modelling noisy observations; in a second step, the kinetic parameters $\boldsymbol{\theta}$ of the ODEs are either optimised or sampled, whilst minimising some metric measuring the difference between the slopes of the tangents to the interpolants, and the $\boldsymbol{\theta}$-dependent time derivative from the ODEs. In this fashion, the ODEs never have to be numerically integrated, and the problem of inferring the typically unknown initial conditions of the system is removed, as it is not required for matching gradients. A downside

to this two-step scheme is that the results of parameter inference are critically dependent on the quality of the initial interpolant. Alternatively, as first suggested in Ramsay et al. [40], the ODEs can be allowed to regularise the interpolant. Dondelinger et al. [11] applied this to the nonparametric Bayesian approach in Calderhead et al. [8], which uses Gaussian processes (GPs), and demonstrated that it significantly improves the parameter inference accuracy and robustness with respect to noise. Unlike in Ramsay et al. [40], all hyperparameters that control the smoothness of the interpolants are consistently inferred in the framework of nonparametric Bayesian statistics, which dispenses with the need to use heuristics and approximations in the configuration of the interpolation function.

This thesis extends and develops methods of gradient matching for parameter inference and model selection in ODE systems in a systems biology context. The layout of the thesis is as follows:

- Chapter 2 covers the literature of the current state-of-the-art methods for parameter inference using gradient matching. Details on the interpolation methods the authors adopted are also included.

- Chapter 3 contains benchmark ODE systems that are used throughout this thesis, for data generation and comparison purposes.

- Chapter 4 details the combining of the methods of Dondelinger et al. [11] and Campbell and Steele [9], creating a new gradient matching

method with a parallel tempering scheme for the gradient mismatch parameter.

- Chapter 5 has a wide-scale comparative analysis of the current state-of-the-art methods detailed in Chapter 2 and Chapter 4.

- Chapter 6 contains a discussion of gradient matching as a probabilistic generative model. Specifically, approximations that were not apparent from the original publication ([49]) are presented. It is demonstrated that they introduce large uncertainty into the parameter estimates and make the method susceptible to identifiability problems when data are systematically missing.

- Chapter 7 presents a new method for model selection using thermodynamic integration and gradient matching. This new method provides a way of performing accurate and robust model selection for ODEs using gradient matching.

- Chapter 8 is a discussion of the work covered throughout the entire thesis.

2 Literature Review

Parameter inference for systems described by ordinary differential equations is challenging and there have been many approaches developed to tackle the problem. The simplest method would be to compare the solution of the equations, for some given parameter set, to noisy observations of the signal based on some appropriate noise model. Parameter estimation would be carried out by minimising the discrepancy between the predicted solution of the ODEs and the data. However, closed-form solutions typically do not exist for many ODEs and therefore inference involving the explicit solution of the equations needs to be conducted numerically. Robinson [43] contains an introduction for obtaining explicit solutions of ordinary differential equations. Amongst many other topics, Robinson discusses the use of Euler's method and the Runge-Kutta scheme as ways for obtaining explicit solutions. Inference could be carried out on a system of ODEs, by using either of these two methods (with a reasonably small step-size) to numerically solve the equations and use least squares estimation to infer the best parameters that describe the data signal. Xue et al. [53] discuss the influence of the numerical approximation to the ODEs (employing the 4-stage Runge-Kutta algorithm in their studies). They argue that previous studies took the numerical solution as being the ground truth and only considered the measurement error when estimating the parameters. The authors show that when the maximum step size of a p-order numerical algorithm goes to zero at a rate faster

than n^{-1/p^4}, where n is the sample size, the numerical error is negligible in comparison to the measurement error. This should provide some guidance in selecting the correct step-size when numerically solving ODEs.

A different integration based approach, which aims at avoiding explicitly solving the ODEs, is to instead first smooth the data with a chosen interpolation method. This interpolant acts as a proxy for the solution of the ODEs and then non-linear least squares is used to infer the parameters. It is demonstrated in Xue et al. [53] that a sieve (a sequence of finite-dimensional models of increasing complexity) estimator is asymptotically normal and has the same asymptotic covariance as when the true solution is known, for the case of having constant parameters over time. A typical example of sieve regression is a spline [22]. Dattner and Klaassen [10] look at ODEs where the systems are linear in the parameters. Taking advantage of the linearity in the model, the authors are able to develop a two-step estimation approach that does not require repeated integration of the system. By reformulating the minimisation function in terms of integrals instead of derivatives, the authors obtain closed form estimates of the parameters of the system. These estimates are shown to be consistent estimators. Dattner and Klaassen consider two types of interpolation schemes - a local polynomial estimator and a step function estimator (which is obtained by averaging repeated measurements). The method using a local polynomial estimator was shown to outperform the two-step gradient matching approach of Liang & Wu [26], whilst it was

unable to outperform the gradient matching method by Ramsay et al. [40] (which is discussed in Chapter 2.3). The accuracy of Daatner and Klaassen's method using a step function estimator did not change much even when the number of repeated measures was quite small. Bayesian smooth-and-match is a related method, that avoids explicitly solving the ODEs and instead indirectly solves the system by numerically integrating the interpolated signals. Ranciati et al. [41] employ this approach, smoothing the data with penalised splines, and use ridge regression to infer the parameters of the ODEs. Again, this approach focuses on systems that are linear in the parameters. In order to achieve a fully probabilistic generative model, the authors take a similar approach to Wang and Barber [49] (a method that will be discussed in detail in Chapter 6) and as a consequence the vector of observations appears twice in the graphical model. The upshot of this is that the method is unable to deal with partially observed systems and the two observation vectors are coupled by a common nuisance (variance) parameter. Ranciati et al. demonstrate that the method is fast, with a built-in quantification of uncertainty about the ODE solution. The results obtained, for a fully observed system that is linear in the parameters, are accurate and robust to dataset size and noise level.

Gradient matching is the method of conducting parameter inference by minimising some metric governing the difference between gradients predicted from a set of differential equations (ordinary differential equations through-

out this thesis) with the slopes of the tangents to the interpolants. Gradient matching bypasses the need for numerical integration, making it computationally attractive. Methods can differ by choice of interpolation scheme and the chosen metric for penalising the difference between gradients. Wu et al. [52] propose a five-step approach for inference in sparse additive ordinary differential equations (SA-ODE). The SA-ODE model is denoted as

$$\dot{x}_s = \chi_s + \sum_{i=1}^{N} f_{si}(x_i(t))$$

and it is assumed that the number of significant non-linear effects, $f_{si}(\cdot)$, is small for each of the N variables even though the total number of variables in the network may be large. At step one, the data is smoothed using penalised splines. At step two, the state variables and derivatives are substituted into the aforementioned SA-ODE model, producing a pseudo-sparse additive model (PSA). A truncated series expansion with B-spline bases is used to approximate the additive components of the PSA model. The number of basis functions is chosen as large as possible with the intention to correct for this at step five. At step three, the group LASSO is used to identify significant functions in the model. The penalty parameter at this step is estimated using BIC. The group LASSO penalty treats the coefficients from each group equally, which is typically non-optimal. Hence, at step four, an adaptive group LASSO is applied to allow different levels of shrinkage to exist

for different coefficients. Finally, at step five, a regular/adaptive LASSO is applied to account for the under-smoothing from step two (due to selecting more bases than are probably necessary). Wu et al. demonstrate in their simulation studies that the method is able to obtain a high true positive rate, when the sample size is sufficiently large, and can more closely match the true underlying signal (noise free signal) than the method by Lu et al. [28] which assumes a linear ODE model and uses the smoothly clipped absolute deviation penalised likelihood method of [13] for variable selection.

The remainder of this chapter contains a literature review of the interpolation schemes and gradient mismatch metrics of the current state-of-the-art methods for parameter inference in ordinary differential equations using gradient matching.

2.1 B-Splines

Splines are used for function interpolation, where the function of interest is approximated by a weighted linear combination of basis functions. These basis functions, called "splines", are "local" polynomials, where the exact functional form depends on the particular type of spline that is used (for example, a truncated power basis). See Hastie et al. [23] for an overview of different types of splines.

The advantage of spline interpolation over global polynomial interpolation is

that the interpolation error can be made small even when using low degree polynomials for the splines. This in particular avoids the problem of Runge's phenomenon, in which oscillations can occur between datapoints when interpolating using high degree polynomials (see Runge [44]).

B-splines interpolation takes the form

$$x(t) = \sum_{i=0}^{m} \alpha_i \phi_{i,d}(t), \tag{2}$$

where $m+1$ is the number of basis functions, d is the degree of polynomial, α_i is a coefficient and $\phi_{i,d}(t)$ is the i^{th} basis function of polynomial degree d evaluated at time t. For some vector of fixed points called knots (denoted $\boldsymbol{\tau}$, where $x(t)$ is continuous at each knot), the basis functions are calculated with the following recursive formulae

$$\phi_{i,0}(t) = \begin{cases} 1 & \text{if} \quad \tau_i \leq t < \tau_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$\phi_{i,d}(t) = \frac{t - \tau_i}{\tau_{i+d} - \tau_i} \phi_{i,d-1}(t) + \frac{\tau_{i+d+1} - t}{\tau_{i+d+1} - \tau_{i+1}} \phi_{i+1,d-1}(t). \tag{4}$$

The coefficients α_i are then estimated by

$$\hat{\boldsymbol{\alpha}} = \left(\boldsymbol{\Phi}^{\mathsf{T}} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^{\mathsf{T}} \mathbf{y}, \tag{5}$$

where $\mathbf{y}$ is the data vector of observations, $\hat{\boldsymbol{\alpha}}$ is the vector containing all the

coefficients (and α_i would correspond to the $(i+1)^{th}$ position in the vector), the form of $\hat{\boldsymbol{\alpha}}$ is obtained by minimising $\sum_{s=1}^{N}(y(t_s) - x(t_s))^2$ and $\boldsymbol{\Phi}$ is the matrix containing all the basis functions

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_{0,d}(t_1) & \dots & \phi_{m,d}(t_1) \\ \vdots & \ddots & \vdots \\ \phi_{0,d}(t_T) & \dots & \phi_{m,d}(t_T) \end{bmatrix}. \tag{6}$$

One can aim to avoid over-fitting by penalising the 2^{nd} derivative of the function $x(t)$ (known as penalised splines), making the objective function

$$J(x) = \sum_{s=1}^{N}(y(t_s) - x(t_s))^2 + \lambda \int \left(\frac{d^2x}{dt^2}\right)^2 dt, \tag{7}$$

where the dependency on $\boldsymbol{\alpha}$ is via equation 2, λ controls the amount of trade-off between the data fit and penalty term. In this case, the coefficients α_i are estimated by

$$\hat{\boldsymbol{\alpha}} = \left(\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi} + \lambda\mathbf{D}\right)^{-1}\boldsymbol{\Phi}^{\mathsf{T}}\mathbf{y}, \tag{8}$$

where the form of $\hat{\boldsymbol{\alpha}}$ is obtained by minimising equation 7 i.e. $\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\, J(x)$, $\mathbf{D}$ is the solution to the penalty in equation 7 (the integral of the square of the second derivative of x). It is possible to change the penalty term in equation 7 to some other penalty form (this is known as P-splines), where the $\mathbf{D}$ in equation 8 would be updated accordingly.

2.2 Smooth Functional Tempering

This chapter details the method for parameter inference used in Campbell and Steele [9]. In the paper, the authors discuss two types of smooth functional tempering, one that needs to infer the initial conditions of the species concentrations and one that does not. Only the method that does not infer the initial conditions is considered here. If the initial conditions are unknown, then they must be inferred as an extra parameter in the inference procedure, however, the method described in this section effectively profiles over the initial conditions, dispensing with the need to infer them. This reduces the complexity of the procedure, which is more appealing. See the original publication Campbell and Steele [9] for details on the former procedure. The choice of interpolation scheme for the concentrations $\mathbf{x}_s$ is B-splines.

The posterior distribution of the parameters is

$$p_{\alpha^{(i)}}(\boldsymbol{\theta}^{(i)}, \sigma^{2(i)} | \mathbf{Y}, \mathbf{X}^{(i)}, \boldsymbol{\lambda}^{(i)})$$

$$\propto p(\boldsymbol{\theta}^{(i)}, \sigma^{2(i)}) p(\mathbf{X}^{(i)} | \boldsymbol{\theta}^{(i)}, \boldsymbol{\lambda}^{(i)}) p(\mathbf{Y} | \mathbf{X}^{(i)}, \sigma^{2(i)})^{\alpha^{(i)}}, \tag{9}$$

where $\mathbf{Y}$ is the matrix containing all of the data, $\mathbf{X}$ is the matrix containing all of the species concentrations, the superscript i denotes those variables associated with "temperature" $\alpha^{(i)}$, the likelihood, $p(\mathbf{Y}|\mathbf{X}^{(i)}, \sigma^{2(i)}) = N(\mathbf{X}^{(i)}, \sigma^{2(i)})$,

is tempered[1], $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ and $p(\mathbf{X}^{(i)}|\boldsymbol{\theta}^{(i)}, \lambda^{(i)})$ is

$$p(\mathbf{X}^{(i)}|\boldsymbol{\theta}^{(i)}, \boldsymbol{\lambda}^{(i)}) \propto \exp\left[-\sum_{s=1}^{n} \lambda_s^{(i)} ||\dot{\mathbf{x}}_s^{(i)} - f_s^{(i)}(\mathbf{X}^{(i)}, \boldsymbol{\theta}_s^{(i)}, \mathbf{t})||^2\right], \tag{10}$$

which is equivalent to

$$p(\mathbf{X}^{(i)}|\boldsymbol{\theta}^{(i)}, \boldsymbol{\lambda}^{(i)}) \propto \exp\left[-\sum_{s=1}^{n} \lambda_s^{(i)} \sum_{t=1}^{T} \left(\dot{x}_s^{(i)}(t) - f_s^{(i)}(\mathbf{x}^{(i)}(t), \boldsymbol{\theta}_s^{(i)}, t)\right)^2\right]. \tag{11}$$

In equation 10 $\lambda_s^{(j)}$ is the gradient mismatch parameter for species s corresponding to "temperature" $\alpha^{(i)}$ (similar to the mismatch parameter $\gamma_s^{(i)}$ in Chapter 4). The $\lambda_s^{(i)}$ is chosen in advance and fixed to each "temperature" $\alpha^{(i)}$ such that $0 < \lambda_s^{(1)} \leq \cdots \leq \lambda_s^{(M)} \leq \infty$, where values closer to 0 allow the gradients to be more different to one another and values closer to ∞ restrict them from being different.

Sampling from equation 9 is performed using MCMC.

[1]Note: parallel tempering is one of the main concepts for the new method proposed in Chapter 4 and therefore, to avoid repetition, the details of tempering can be found there. To summarise the concept, the likelihood is raised to a power (called a "temperature") between 0 and 1, where the posterior becomes equal to the prior when the power is 0 (up to some normalisation constant) and is recovered when the power is 1. Powers between 0 and 1 give a distribution between the prior and posterior. "Temperatures" closer to the prior tend to produce less rugged distributions, making it easier for algorithms to navigate the landscape. Different "temperatures" are randomly selected and the corresponding parameters have a probability to be exchanged. In this fashion, algorithms can avoid being trapped in local optima and more easily achieve global convergence. The likelihood here $p(\mathbf{Y}|\mathbf{X}^{(i)}, \sigma^{2(i)})$ is tempered in the same way as in equation 66.

2.3 Penalised Likelihood With Hierarchical Regularisation

Ramsay et al. [40] aim to conduct parameter inference in ODEs using a penalised likelihood approach and a hierarchical regularisation in order to tune the gradient mismatch parameter and parameters of their interpolation scheme (splines). They perform parameter inference in a hierarchical two level approach. At level 1, the gradient mismatch parameter is configured, in order to ensure the estimates of the coefficients of their interpolant are properly regularised by the mismatch to the ODEs. In their paper, they adjust the gradient mismatch parameter manually using numerical and visual heuristics, but suggest a way it could be achieved through generalised cross-validation, which is detailed in this chapter. At level 2a., the coefficients of the interpolant are optimised. Whilst optimising for the parameters, each time the ODE parameters and observational noise parameters are changed, they re-optimise the coefficients of the interpolant, by penalising the differences between the gradients, which allows the ODEs to regulate the interpolant. At level 2b., the ODE and observational noise parameters are estimated using a sum of squares criterion. This criterion is optimised directly for the ODE and observational noise parameters, but it is also optimised implicitly, since the sum of squares incorporates $\mathbf{x}_s$, which itself was optimised at level 2a. with respect to these parameters. A flow chart of these two levels can be found in Figure 1.

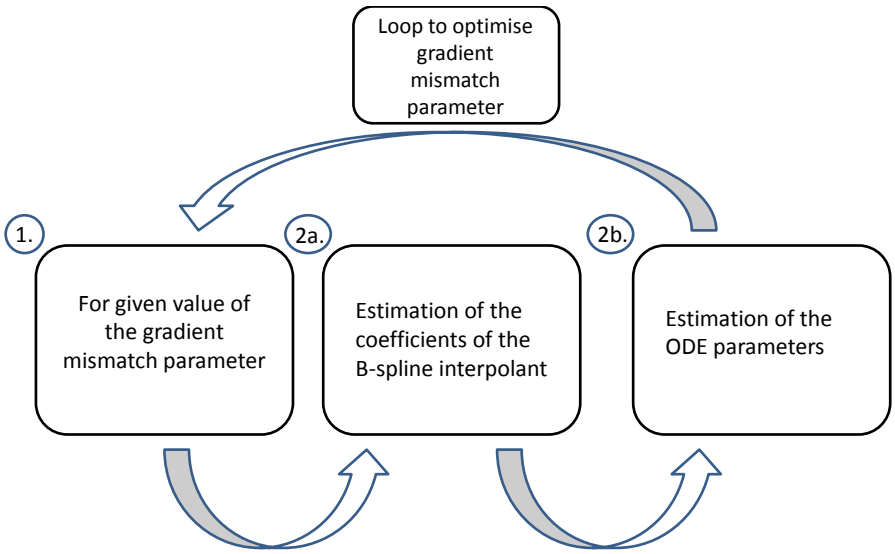


Figure 1: Flow chart of the two level approach employed by Ramsay et al. [40]. At level 1, the gradient mismatch parameter is specified. At level 2a., the coefficients of the interpolant are estimated (splines in this method) and at level 2b., the ODE parameters are estimated. Levels 1 and 2 are then iterated in order to optimise the gradient mismatch parameter and thus the model. The two levels are iterated using a pseudo-delta method (see Chapter 2.3 for details).

At level 1 of the two hierarchical levels, the gradient mismatch parameter is configured. To avoid the need for heuristics, Ramsay et al. [40] suggest the use of generalised cross-validation, since the estimation of the state variables for some gradient mismatch parameter λ is usually a non-linear problem and so standard cross-validation methods are not computationally viable. Generalised cross-validation takes the form

$$F(\boldsymbol{\lambda}) = \frac{\sum_{s=1}^{n} ||\mathbf{y}_s - \mathbf{x}_s||^2}{\left[\sum_{s=1}^{n} \left\{T - \sum_{t=1}^{T} \frac{dx_s(t)}{dy_s(t)}\right\}\right]^2}, \tag{12}$$

where $\mathbf{y}_s$ is the data for species s, $\mathbf{x}_s$ is the interpolant corresponding to species s, n is the number of species and T is the number of timepoints. A derivation of equation 12 can be found in the appendix. The derivatives in the denominator can be expressed as

$$\frac{dx_s(t)}{dy_s(t)} = \frac{\partial x_s(t)}{\partial \boldsymbol{\alpha}} \frac{d\boldsymbol{\alpha}}{dy_s(t)}, \tag{13}$$

where $\boldsymbol{\alpha}$ are the estimated coefficients of the splines interpolant (see equation 8). Calculating these derivatives takes the dependency of the data $\mathbf{y}$ and the ODE parameters $\boldsymbol{\theta}$ into account, since $\frac{d\boldsymbol{\alpha}}{d\mathbf{y}} = \frac{\partial \boldsymbol{\alpha}}{\partial \boldsymbol{\theta}} \frac{d\boldsymbol{\theta}}{d\mathbf{y}} + \frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{y}}$. The estimates of $\boldsymbol{\lambda}$ will be calculated by minimising equation 12 over values of $\boldsymbol{\lambda}$.

Level 2a. involves estimating the coefficients of the splines interpolant using the following criterion

$$J(\boldsymbol{\alpha}|\boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\lambda}) = \sum_{s=1}^{n} w_s ||\mathbf{y}_s - \mathbf{x}_s||^2 + \sum_{s=1}^{n} \lambda_s \int \left[\frac{dx_s(t)}{dt} - f_s(\mathbf{x}(t), \boldsymbol{\theta}_s, t)\right]^2 dt, \tag{14}$$

where $\frac{d\mathbf{x}_s}{dt}$ is the gradient of the interpolant for species s and w_s are weights to normalise the sum of squares of different species (so that species on varying scales of measurement do not distort the sum of squares with very large

or very small residuals that are simply a consequence of their magnitude or unit of measurement). Large values of λ_s mean that the gradients have to more closely match one another (since the difference between them will need to tend to 0, to compensate for the large penalty a large λ_s would produce), whereas small values would allow the gradients to differ more. The penalty term in equation 14 allows the mismatch between the gradients to regularise the estimates of the interpolant coefficients.

At level 2b., the ODE parameters are optimised using the sum of squares criterion

$$S(\boldsymbol{\theta}|\boldsymbol{\lambda}) = \sum_{s=1}^{n} w_s ||\mathbf{y}_s - \mathbf{x}_s||^2. \tag{15}$$

To optimise equation 15 with respect to $\boldsymbol{\theta}$, Ramsay et al. [40] find the solution of the gradient

$$\frac{dS(\boldsymbol{\theta}|\boldsymbol{\lambda})}{d\boldsymbol{\theta}} = \frac{\partial S(\boldsymbol{\theta}|\boldsymbol{\lambda})}{\partial \boldsymbol{\theta}} + \frac{\partial S(\boldsymbol{\theta}|\boldsymbol{\lambda})}{\partial \boldsymbol{\alpha}} \frac{d\boldsymbol{\alpha}}{d\boldsymbol{\theta}} = 0. \tag{16}$$

Since the function $\boldsymbol{\alpha}(\boldsymbol{\theta})$ is not explicitly available, $\dfrac{d\boldsymbol{\alpha}}{d\boldsymbol{\theta}}$ is calculated by application of the implicit function theorem of differential calculus. This gives

$$\frac{d\boldsymbol{\alpha}}{d\boldsymbol{\theta}} = -\left(\frac{\partial^2 J(\boldsymbol{\alpha}|\boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\lambda})}{\partial \boldsymbol{\alpha}^2}\right)^{-1} \frac{\partial^2 J(\boldsymbol{\alpha}|\boldsymbol{\theta}, \boldsymbol{\sigma}, \boldsymbol{\lambda})}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\theta}}. \tag{17}$$

2.4 Reproducing kernel Hilbert Space

Here a background is provided for reproducing kernel Hilbert spaces (RKHS), that are used in González et al. [19], and how they compare to Gaussian processes. RKHS interpolation is a useful tool in statistical learning, since a property of reproducing kernel Hilbert spaces, known as the representer theorem (details to follow), means that every function in an RKHS can be written as a linear combination of the kernel function evaluated at the training points. This provides a computationally fast process for interpolation, which is particularly useful in gradient matching, since the original purpose of gradient matching is to obtain a computational speed-up over methods involving calculating numerical solutions to the ODEs.

By Mercer's theorem ([35]), it is possible to represent a kernel that produces a positive definite covariance matrix in terms of eigenvalues λ_s and eigenfunctions ν_s

$$k(t_i, t_j) = \sum_{s=1}^{\infty} \lambda_s \nu_s(t_i) \nu_s(t_j). \tag{18}$$

These ν_s form an orthonormal basis for a function space

$$H = \{f : f(t) = \sum_{s=1}^{\infty} f_s \nu_s(t), \sum_{s=1}^{\infty} \frac{f_s^2}{\lambda_s} < \infty\}. \tag{19}$$

The inner product between two functions $f(t) = \sum_{s=1}^{\infty} f_s \nu_s(t)$ and $g(t) = \sum_{s=1}^{\infty} g_s \nu_s(t)$ in the space in equation 19 is defined as

$$\langle f, g \rangle_H \triangleq \sum_{s=1}^{\infty} \frac{f_s g_s}{\lambda_s}, \tag{20}$$

which Murphy [36] shows implies that

$$\langle k(t_1, \cdot), k(t_2, \cdot) \rangle_H = k(t_1, t_2). \tag{21}$$

This is known as the reproducing property and the space of functions H is called a reproducing kernel Hilbert space. Now consider the minimisation problem

$$J(f) = \frac{1}{2\sigma^2} \sum_{s=1}^{N} (y_s - f(t_s))^2 + \frac{1}{2} ||f||_H^2, \tag{22}$$

where $J(f)$ is the objective function and $||f||_H$ is the norm in Hilbert space

$$||f||_H = \langle f, f \rangle_H = \sum_{s=1}^{\infty} \frac{f_s^2}{\lambda_s}. \tag{23}$$

The desired function used for interpolation should be simple and provide a good fit to the data. Complex functions with respect to the kernel in equation 18 will produce large norms, since they will need many eigenfunctions to represent them, and therefore be more heavily penalised in equation 22. Schöelkopf and Smola [45] show that the desired function must have the following form

$$f(t) = \sum_{s=1}^{N} c_s k(t, t_s). \tag{24}$$

This follows from the representer theorem, see [35] and[45]. To solve for $\mathbf{c}$, equation 24 can be combined with equation 22, since equation 24 is of the correct form to use the reproducing property (see equation 21 and 20), giving

$$J(\mathbf{c}) = \frac{1}{2\sigma^2}|\mathbf{y} - \mathbf{K}\mathbf{c}|^2 + \frac{1}{2}\mathbf{c}^\mathsf{T}\mathbf{K}\mathbf{c}, \tag{25}$$

where $\mathbf{K}$ is a matrix of kernel elements for all combinations of observed timepoints. Minimising with respect to $\mathbf{c}$ gives

$$\hat{\mathbf{c}} = (\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}. \tag{26}$$

Hence,

$$\hat{f}(t_*) = \sum_{s=1}^{N} \hat{c}_s k(t_*, t_s) = \mathbf{k}_*^\mathsf{T}(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}, \tag{27}$$

where t_* is the timepoint at which one wants to make predictions and $\mathbf{k}_*$ is the vector of kernel elements for all combinations of t_* and t_s. This form is the same as a posterior mean of a Gaussian process predictive distribution.

2.5 Penalised Likelihood With RKHS

The aim of González et al. [19] is to create a penalised likelihood function that incorporates the information of the ODEs, then, using the properties of reproducing kernel Hilbert spaces, perform parameter estimation in a computationally fast manner. González et al. [19] consider ODEs of the form

$$\dot{\mathbf{x}}_s = g_s(\mathbf{X}, \boldsymbol{\rho}_s, \mathbf{t}) - \delta_s \mathbf{x}_s, \tag{28}$$

which can be represented in scalar form as

$$\dot{x}_s(t_i) = g_s(\mathbf{x}(t), \boldsymbol{\rho}_s, t_i) - \delta_s x_s(t_i), \tag{29}$$

where $\mathbf{x}_s$ is the vector of concentrations for species s, δ_s is the degradation rate of the concentrations for species s, $\boldsymbol{\rho}_s$ is a parameter vector for species s and $g_s(\mathbf{t}) = (g_s(t_1), \ldots, g_s(t_T))^{\mathsf{T}}$. It is important to realise the difference between equation 1 and equation 28. Whereas in equation 1, all parameter terms are included in the function $f_s()$, equation 28 considers the linear decay term separate to the rest of the ODE function $g_s(\mathbf{X}, \boldsymbol{\rho}_s, \mathbf{t})$. Now consider a differencing matrix $\mathbf{D}$, where

$$\mathbf{D} = \Delta \begin{bmatrix} -1 & 1 & 0 & \ldots & \ldots & 0 \\ -1 & 0 & 1 & 0 & \ldots & 0 \\ 0 & -1 & \ddots & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & \ldots & \ldots & -1 & 1 \end{bmatrix}, \tag{30}$$

and $\Delta = \text{diag}\left(\frac{1}{t_2 - t_1}, \frac{1}{t_3 - t_1}, \frac{1}{t_4 - t_2}, \ldots, \frac{1}{t_T - t_{T-2}}, \frac{1}{t_T - t_{T-1}}\right)$. Equation 28 can then be approximated as

$$\mathbf{D}\mathbf{x}_s = g_s(\mathbf{X}, \boldsymbol{\rho}_s, \mathbf{t}) - \delta \mathbf{x}_s. \tag{31}$$

To make it clear how $\mathbf{D}\mathbf{x}_s$ is computed, as an example, consider $\mathbf{x}_s = (x(t_1), \ldots, x(t_5))^\mathsf{T}$ and $\mathbf{t} = (3, 4, 5, 6, 7)^\mathsf{T}$.

Then

$$\mathbf{D}\mathbf{x}_s = \begin{bmatrix} \frac{1}{4-3} & & & & \\ & \frac{1}{5-3} & & & \\ & & \frac{1}{6-4} & & \\ & & & \frac{1}{7-5} & \\ & & & & \frac{1}{7-6} \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-x(3)+x(4)}{1}, & \frac{-x(3)+x(5)}{2}, & \frac{-x(4)+x(6)}{2}, & \frac{-x(5)+x(7)}{2}, & \frac{-x(6)+x(7)}{1} \end{bmatrix}^\mathsf{T}. \tag{32}$$

Now denote $\mathbf{R} = \mathbf{D} + \delta_s \mathbf{I}$ (where $\mathbf{I}$ is the identity matrix). This gives the following penalty to be incorporated into the likelihood term:

$$\boldsymbol{\Omega}(\mathbf{x}_s) = ||\mathbf{R}\mathbf{x}_s - g_s(\mathbf{X}, \boldsymbol{\rho}_s, \mathbf{t})||^2. \tag{33}$$

From equation 31, it can be seen that $\mathbf{R}\mathbf{x}_s - g_s(\mathbf{X}, \boldsymbol{\rho}_s, \mathbf{t}) = 0$. However, since $\mathbf{x}_s = \mathbf{0}$ does not necessarily imply that $\boldsymbol{\Omega}(\mathbf{x}_s) = 0$, equation 33 cannot be expressed as a norm of $\mathbf{x}_s$ within the RKHS framework. In order to make them compatible, the authors transform the state variables $\mathbf{x}_s$ (and subsequently $\mathbf{y}_s$). Instead, consider

$$\tilde{\mathbf{x}}_s = \mathbf{x}_s - \mathbf{R}^{-1} g_s(\mathbf{X}, \boldsymbol{\rho}_s, \mathbf{t}). \tag{34}$$

Multiplying both sides of equation 34 by $\mathbf{R}$ and taking squared norms gives exactly the same form as equation 33 ($||\mathbf{R}\tilde{\mathbf{x}}_s||^2 = ||\mathbf{R}\mathbf{x}_s - g_s(\mathbf{X}, \boldsymbol{\rho}_s, \mathbf{t})||^2$). Similarly, the data are transformed

$$\tilde{\mathbf{y}}_s = \mathbf{y}_s - \mathbf{R}^{-1} g_s(\mathbf{X}, \boldsymbol{\rho}_s, \mathbf{t}), \tag{35}$$

in order to correspond with the transformed states $\tilde{\mathbf{x}}_s$. The penalty function in equation 33 is now

$$\boldsymbol{\Omega}(\tilde{\mathbf{x}}_s) = ||\mathbf{R}\tilde{\mathbf{x}}_s||^2 = \langle \mathbf{R}\tilde{\mathbf{x}}_s, \mathbf{R}\tilde{\mathbf{x}}_s \rangle = \tilde{\mathbf{x}}_s^{\mathsf{T}} \mathbf{R}^{\mathsf{T}} \mathbf{R} \tilde{\mathbf{x}}_s. \tag{36}$$

Equation 36 is now a proper norm, since when $\tilde{\mathbf{x}}_s = \mathbf{0}$, this implies $\boldsymbol{\Omega}(\tilde{\mathbf{x}}_s) = 0$. Denote $\mathbf{K} = (\mathbf{R}^{\mathsf{T}}\mathbf{R})^{-1}$. $\mathbf{K}$ is a matrix of kernel elements which define a unique RKHS. Hence,

$$\boldsymbol{\Omega}(\tilde{\mathbf{x}}_s) = ||\tilde{\mathbf{x}}_s||_H^2 = \mathbf{c}^{\mathsf{T}} \mathbf{K} \mathbf{c}, \tag{37}$$

where the dependency on $\tilde{\mathbf{x}}_s$ comes via equation 24 (with $\tilde{\mathbf{x}}_s = f(\mathbf{t})$), $\mathbf{c} = \mathbf{K}^{-1}\tilde{\mathbf{x}}_s$ (since substituting this into equation 37 returns equation 36) and equation 37 is used as the term in the far right of equation 25. It is possible to obtain closed form expressions for the transformed state variables by

using equations 26 and 27 (the original expressions can be recovered using equation 34)

$$\tilde{\mathbf{x}}_s = \mathbf{K}(\mathbf{K} + 2\lambda_s \boldsymbol{\Sigma})^{-1} \tilde{\mathbf{y}}_s, \tag{38}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the data (generalising equation 26, since the observational error of the data may not be independent between species) and λ_s is a penalty parameter.

In the case of homogeneous ODEs, where $g_s() = 0$, a kernel in a Hilbert space can be constructed using the Green's function of the linear operator $\mathbf{R}$. A Green's function (G) of a linear operator ($\mathbf{R}$ in this case) is a function that satisfies $\mathbf{R}G(a, b) = \delta(a - b)$, where δ is the Dirac function [20]. $\mathbf{K}$ is the Green's function of $\mathbf{R}^\mathsf{T}\mathbf{R}$, where $\mathbf{R}^\mathsf{T}$ is the adjoint operator of $\mathbf{R}$. Aronszajin et al. [3] show $||\mathbf{R}\tilde{\mathbf{x}}_s||^2_{L^2} = ||\tilde{\mathbf{x}}_s||^2_{H_\mathbf{K}} = \Omega(\tilde{\mathbf{x}}_s)$. Since the analytical form of Green functions of $\mathbf{R}^\mathsf{T}\mathbf{R}$ is not available, the differential operator is approximated with the difference operator ($\mathbf{D}$). In the non-homogeneous ODE system, the model is linearised by feeding surrogate $\hat{\mathbf{x}}_s$ (using spline interpolation, in this case) into $g_s()$. $\Omega(\tilde{\mathbf{x}}_s)$ is still a valid RKHS norm for the transformed variable $\tilde{\mathbf{x}}_s$ defined in equation 34.

The penalised log-likelihood function is now expressed as

$$l(\boldsymbol{\rho}_s, \delta_s, \boldsymbol{\Sigma}, \boldsymbol{\alpha}_s, \mathbf{c}|\tilde{\mathbf{y}}_s) = \sum_{s=1}^{N} \left[-\frac{1}{2}(\tilde{\mathbf{y}}_s - \tilde{\mathbf{x}}_s)^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} (\tilde{\mathbf{y}}_s - \tilde{\mathbf{x}}_s) - \frac{1}{2} ln|\boldsymbol{\Sigma}| \right] - \sum_{s=1}^{N} \lambda_s \boldsymbol{\Omega}(\tilde{\mathbf{x}}_s), \tag{39}$$

where $\boldsymbol{\alpha}_s$ is the vector containing the coefficients from the spline interpolant for species s. Parameter estimation using equation 39 can be carried out with standard non-linear optimisation algorithms such as quasi-Newton or conjugate gradients.

In the original paper of González et al. [19], the penalty parameter λ_s is inferred using AIC. For a given value of λ_s, equation 39 is optimised to estimate the ODE parameters and subsequently the AIC score of the procedure is calculated. This is repeated for different λ_s values and the λ_s value corresponding to the smallest AIC score is chosen.

As well as using this approach for estimating λ_s, it was found that using 3-fold cross validation, instead of AIC, provided more robust parameter estimation. The results for both schemes are presented in Chapter 5.

3 Benchmark ODE Systems

The ODE systems used as benchmark models throughout this thesis, are detailed in this chapter. Details to the specific parameter setting used to simulate data for a particular set-up, can be found in the corresponding chapters.

3.1 The Fitz-Hugh Nagumo system

These equations originally were used to describe the voltage potential across the cell membrane of the axon of giant squid neurons ([14], [38]). There are 3 parameters; α, β and ψ and two "species"; Voltage (V) and Recovery variable (R). Species in [ ] denote the time-dependent concentration for that species and a dot over a symbol is shorthand for the temporal derivative $\frac{d}{dt}$ of that symbol:

$$[\dot{V}] = \psi([V] - \frac{[V]^3}{3} + [R]); \tag{40}$$

$$[\dot{R}] = -\frac{1}{\psi}([V] - \alpha + \beta * [R]) \tag{41}$$

The Fitz-Hugh Nagumo equations are used in Biomedical Engineering to model features such as cardiac conditions (i.e. electrical excitation-conduction in cardiac tissue [1], cardiac action potentials [12] and arrhythmias [17]) and neurodegenerative diseases (Drosophila courtship can be modelled using these equations and used to screen genes linked to memory-deficiency and

human neurodegeneration [7] and the system can also be used for diagnosing Leprosy [47]).

An example of the signals produced from these ODEs can be found in Figure 2.

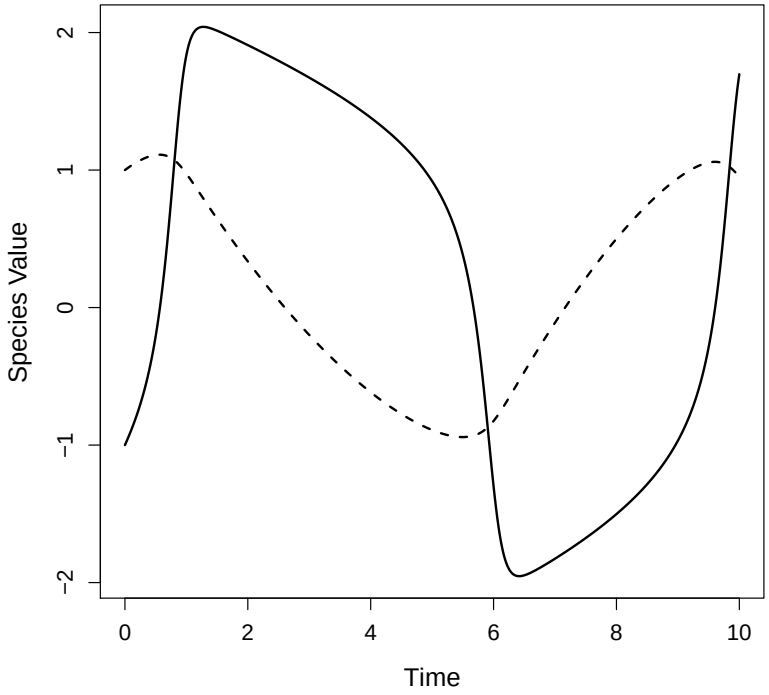


Figure 2: An example of the signals produced from the Fitz-Hugh Nagumo ODEs in equation 40. The solid line represents the signal for species V and the dashed line represents the signal for species R.

3.2 The Lotka-Volterra system

This is a simple model for prey-predator interactions in ecology [27], and autocatalysis in chemical kinetics [4]. Equations 42 - 44 are different candidate forms of the Lotka-Volterra system of equations, progressively increasing in complexity. Equation 43 has one extra parameter than the standard form (equation 42) to account for intra-species competition and the most complex version, equation 44, is described using a saturation term (similar to a Michaelis-Menten term that can appear in biological systems described by chemical kinetics).

$$[\dot{x}_1] = \theta_1 * [x_1] - \theta_2 * [x_1] * [x_2]; \quad [\dot{x}_2] = -\theta_3 * [x_2] + \theta_4 * [x_1] * [x_2] \tag{42}$$

$$[\dot{x}_1] = \theta_1 * [x_1] - \theta_2 * [x_1] * [x_2] - \theta_5 * [x_1]^2; \quad [\dot{x}_2] = -\theta_3 * [x_2] + \theta_4 * [x_1] * [x_2] \tag{43}$$

$$[\dot{x}_1] = \theta_1 * [x_1] - \frac{\theta_2 * [x_1] * [x_2]}{1 + \theta_5 * [x_1]}; \quad [\dot{x}_2] = -\theta_3 * [x_2] + \frac{\theta_4 * [x_1] * [x_2]}{1 + \theta_5 * [x_1]} \tag{44}$$

An example of the signals produced from the Lotka-Volterra model (equation 42) can be found in Figure 3.

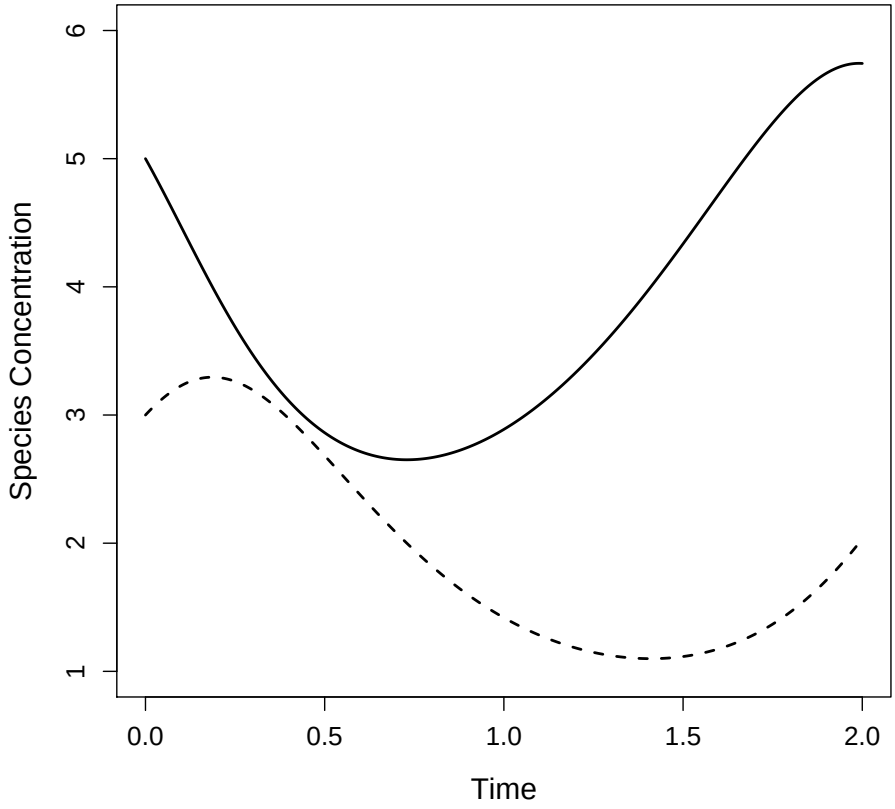


Figure 3: An example of the signals produced from the Lotka-Volterra model (equation 42). The solid line is x_1 and the dashed line is x_2.

3.3 Protein signalling transduction pathways

These equations describe protein signalling transduction pathways in a signal transduction cascade [48], where the kinetic parameters control how quickly the proteins ("species") convert to one another. There are 6 parameters $(k_1, k_2, k_3, k_4, V, K_m)$ and 5 "species" (S, dS, R, RS, Rpp). The system describes the phosphorylation of a protein, $R \rightarrow Rpp$, catalysed by an enzyme S, via an active protein complex (RS), where the enzyme is subject to degradation $(S \rightarrow dS)$. The chemical kinetics are described by a combination of mass action kinetics and Michaelis-Menten kinetics. A graphical representation of this system can be seen in Figure 4. Species in [ ] denote the time-dependent concentration for that species and a dot over a symbol is shorthand for the temporal derivative $\frac{d}{dt}$ of that symbol:

$$
\begin{aligned}
[\dot{S}] &= -k_1 * [S] - k_2 * [S] * [R] + k_3 * [RS] \\
[\dot{dS}] &= k_1 * [S] \\
[\dot{R}] &= -k_2 * [S] * [R] + k_3 * [RS] + \frac{V * [Rpp]}{K_m + [Rpp]} \\
[\dot{RS}] &= k_2 * [S] * [R] - k_3 * [RS] - k_4 * [RS] \\
[\dot{Rpp}] &= k_4 * [RS] - \frac{V * [Rpp]}{K_m + [Rpp]}
\end{aligned} \tag{45}
$$

Cell signalling is a highly relevant topic in current Biomedical Engineering and can model cancers [34] and neurodegenerative diseases that include Alzheimer's disease, Parkinson's disease and Amyotrophic Lateral Sclerosis (ALS) [25].

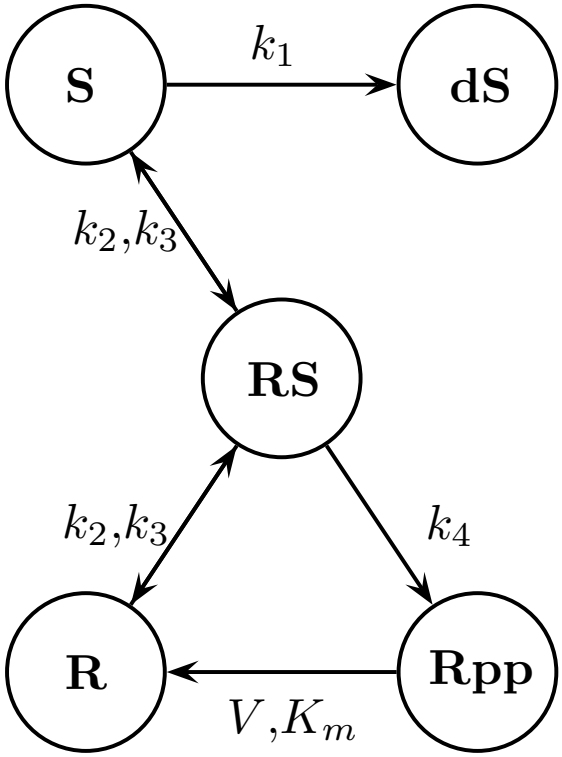


Figure 4: Graphical representation of the protein signalling transduction pathway in equation 45. There are 5 "species" (S, dS, R, RS, Rpp) and 6 parameters $(k_1, k_2, k_3, k_4, V, K_m)$. The system describes the phosphorylation of a protein, $R \rightarrow Rpp$, catalysed by an enzyme S, via an active protein complex (RS), where the enzyme is subject to degradation $(S \rightarrow dS)$. Figure adapted from [48].

An example of the signals produced from these ODEs can be found in Figure 5.

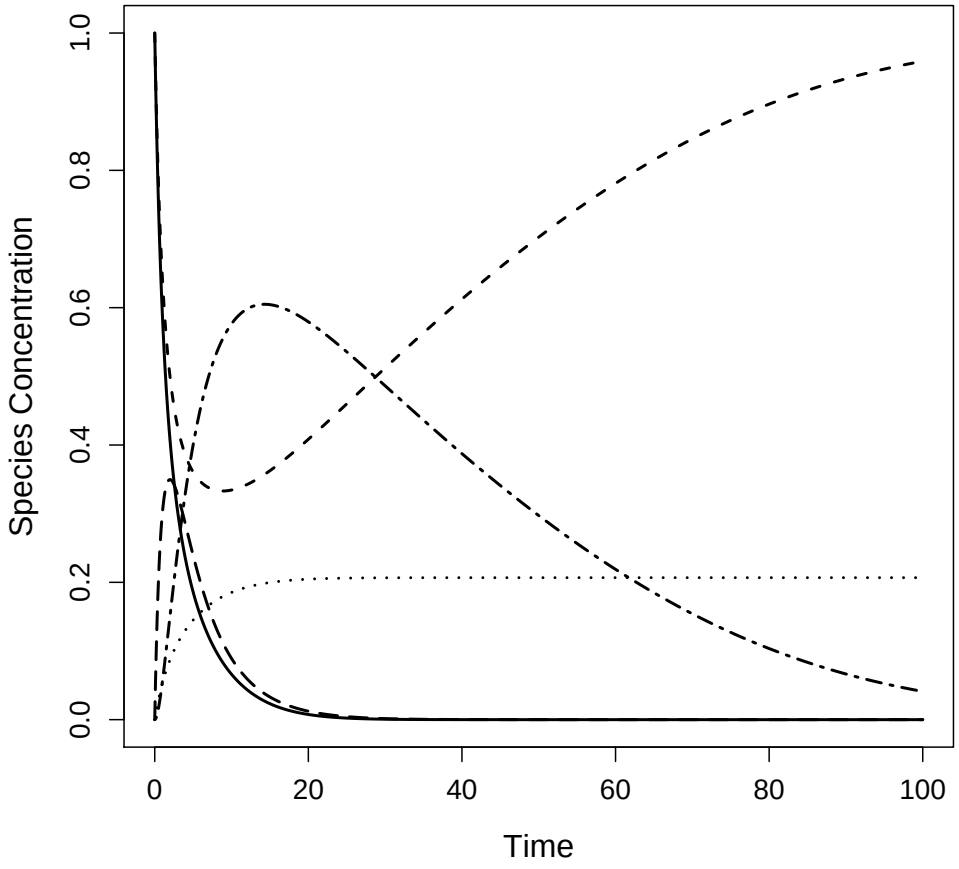


Figure 5: An example of the signals produced from the protein signalling transduction pathway in equation 45. The solid line is S, the light dotted line is dS, the dashed line near the top of the Figure is R, the longer dashed line near the bottom of the Figure is RS and the dot-dashed line is Rpp.

The following are different candidate models of the protein signalling transduction pathway, all with varying degrees of complexity. Graphical representations of the pathways can be seen in Figures 6-8.

Equation 46 is a simplified version of equation 45, where now a more general description of the activation process is considered. It is predominantly the same process as in equation 46, but now it uses Michaelis-Menten kinetics to describe the phosphorylation of protein R.

$$
\begin{aligned}
[\dot{S}] &= -k_1 * [S] \\
[\dot{dS}] &= k_1 * [S] \\
[\dot{R}] &= \frac{-V_1 * [R] * [S]}{k_2 + [R]} + \frac{V_2 * Rpp}{k_3 + Rpp} \\
[\dot{Rpp}] &= \frac{V_1 * [R] * [S]}{k_2 + [R]} - \frac{V_2 * Rpp}{k_3 + Rpp}
\end{aligned} \tag{46}
$$

Equation 47 is the least complex of the candidate models. It does not describe the degradation of protein S to dS and hence the signal of S cannot decrease.

$$[\dot{R}] = \frac{-V_1 * [R] * [S]}{k_1 + [R]} + \frac{V_2 * Rpp}{k_2 + Rpp}$$

$$[\dot{Rpp}] = \frac{V_1 * [R] * [S]}{k_1 + [R]} - \frac{V_2 * Rpp}{k_2 + Rpp} \tag{47}$$

Equation 48 is the most complex of the candidate models, it describes how the phosphatase PhA deactivates the protein Rpp. All reactions are defined by mass action kinetics.

$$[\dot{S}] = -k_1 * [S] - k_2 * [S] * [R] + k_3 * [RS]$$

$$[\dot{dS}] = k_1 * [S]$$

$$[\dot{R}] = -k_2 * [S] * [R] + k_3 * [RS] + k_7 * [RppPhA]$$

$$[\dot{RS}] = k_2 * [S] * [R] - k_3 * [RS] - k_4 * [RS]$$

$$[\dot{Rpp}] = k_4 * [RS] - k_5 * [Rpp] * [PhA] + k_6 * [RppPhA]$$

$$[\dot{PhA}] = -k_5 * [Rpp] * [PhA] + k_6 * [RppPhA] + k_7 * [RppPhA]$$

$$[\dot{RppPhA}] = k_5 * [Rpp] * [PhA] - k_6 * [RppPhA] - k_7 * [RppPhA] \tag{48}$$

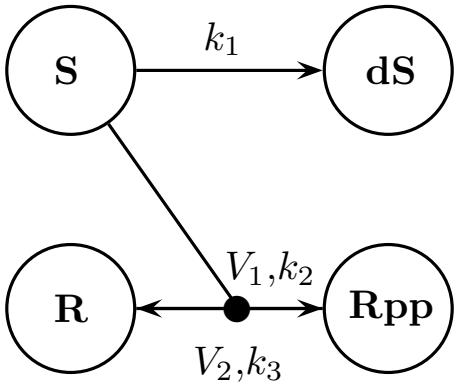


Figure 6: Graphical representation of the protein signalling transduction pathway in equation 46. A simplified version of equation 45 (and shown graphically in Figure 4), where now a more general description of the activation process is considered. There are 4 "species" (S, dS, R, Rpp) and 5 parameters $(k_1, k_2, k_3, V_1, V_2)$. Figure adapted from Vyshemirsky and Girolami [48].

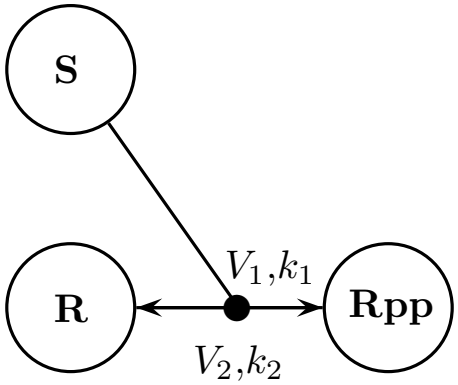


Figure 7: Graphical representation of the protein signalling transduction pathway in equation 47. The least complex of the candidate models. It does not describe the degradation of protein S to dS. There are 3 "species" (S, R, Rpp) and 4 parameters (k_1, k_2, V_1, V_2). Figure adapted from Vyshemirsky and Girolami [48].

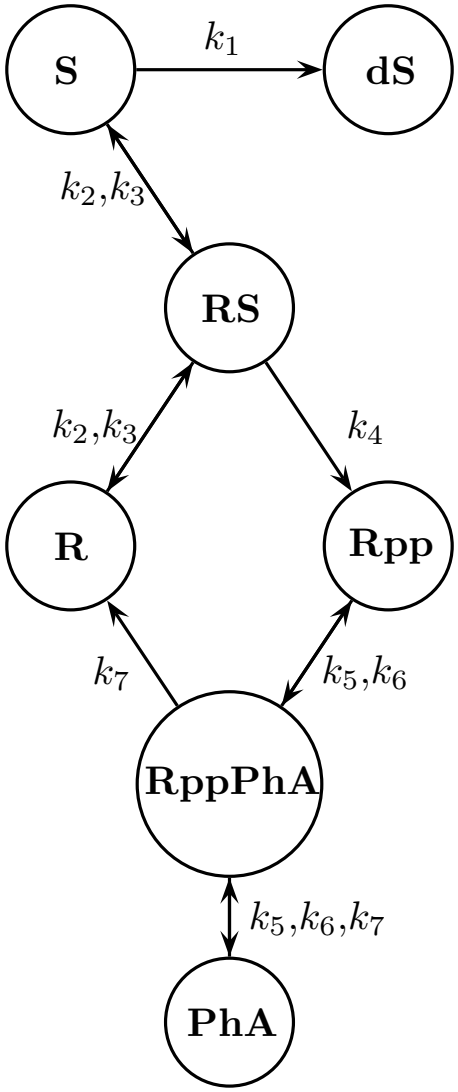


Figure 8: Graphical representation of the protein signalling transduction pathway in equation 48. The most complex of the candidate models, it describes how the phosphatase PhA deactivates the protein *Rpp*. There are 7 "species" $(S, dS, R, RS, Rpp, RppPhA, PhA)$ and 7 parameters $(k_1, k_2, k_3, k_4, k_5, k_6, k_7)$. Figure adapted from Vyshemirsky and Girolami [48].

4 Gradient Mismatch Parameter Parallel Tempering Scheme

This chapter presents work published in Macdonald et al. [31].

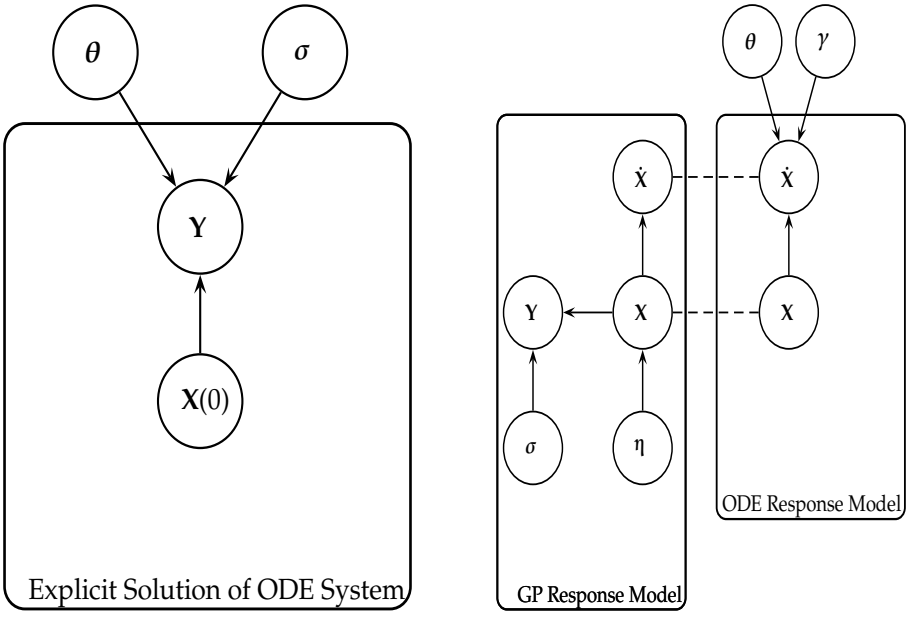
4.1 Introduction

The nature of the ODE-based model in equation 1 renders the inference problem computationally challenging in two respects. Firstly, the ODE system often does not permit closed-form solutions. One therefore has to resort to numerical integration every time the parameters $\boldsymbol{\theta}_s$ are adapted, which is computationally onerous. Secondly, the likelihood function in the space of parameters $\boldsymbol{\theta}_s$ is typically not unimodal, but suffers from multiple local optima. Hence, even if a closed-form solution of the ODEs existed, inference by maximum likelihood would not be computationally viable for many cases, and Bayesian inference would suffer from poor mixing and convergence of the Markov chain Monte Carlo (MCMC) simulations.

Conventional inference methods involve numerically integrating the system of ODEs to produce a signal, which is compared to the data by some appropriate metric defined by the chosen noise model, allowing for the calculation of a likelihood. This process is repeated as part of an iterative optimisation or sampling procedure to produce estimates of the parameters. Figure 9(a) is a graphical representation of the model for these conventional inference

methods. For a given set of initial concentrations of the entire system $\mathbf{X}(0)$ and set of ODE parameters $\boldsymbol{\theta}$, a signal can be produced by integration of the ODEs. As mentioned previously, for many ODE systems a closed-form solution does not exist, so in practice, numerical integration is implemented instead. Assuming an appropriate noise model (for example a Gaussian additive noise model) with standard deviation of the observational error $\boldsymbol{\sigma}$, the differences between the resultant signal and the data $\mathbf{Y}$ can be used to calculate the likelihood of the parameters $\boldsymbol{\theta}$. The process is repeated for different parameters $\boldsymbol{\theta}$ until the maximum likelihood of the parameters is found (in the classical approach) or until convergence to the posterior distribution is reached (in the Bayesian approach). However, the computational costs involved with repeatedly numerically solving the ODEs are large.

To reduce the computational complexity, several authors have adopted an approach based on gradient matching (e.g. Calderhead et al. [8] and Liang & Wu [26]). The idea is based on the following two-step procedure. In a preliminary smoothing step, the time series data are interpolated; then, in a second step, the parameters $\boldsymbol{\theta}$ of the ODEs are optimised so as to minimise some metric measuring the difference between the slopes of the tangents to the interpolants, and the $\boldsymbol{\theta}$-dependent time derivatives from the ODEs. In this way, the ODEs never have to be solved explicitly, and the typically unknown initial conditions are effectively profiled over. A disadvantage of this two-step scheme is that the results of parameter inference critically hinge on



(a) Explicit solution of the ODE system, as shown in [8]. The noisy data signals $\mathbf{Y}$ are described by some initial concentration $\mathbf{X}(0)$, ODE parameters $\boldsymbol{\theta}$ and observational error with standard deviation $\boldsymbol{\sigma}$. For a given set of initial concentrations $\mathbf{X}(0)$ and set of ODE parameters $\boldsymbol{\theta}$, the ODEs can be integrated to produce a signal, which is then compared to the data signal by some metric defined by the chosen noise model.

(b) Gradient matching with Gaussian processes, as proposed in [8] and [11]. The gradients $\dot{\mathbf{X}}$ are compared from two modelling approaches; the Gaussian process model and the ODEs themselves. The distribution of $\mathbf{Y}$ is given in equation 52, the Gaussian process on $\mathbf{X}$ defined in equation 53, the derivatives of the Gaussian process $\dot{\mathbf{X}}$ in equation 58, the ODE model in equation 50 and the gradient matching in equation 65. All symbols are detailed throughout Chapter 4.

Figure 9: Graphical representations of (left) the explicit solution of the ODE system, as shown in [8], and (right) gradient matching with Gaussian processes, as proposed in [8] and [11]. The nodes (depicted by circles) represent random variables and the edges represent conditional dependence from one node to another. A directed edge from node "A" to node "B" depicts that "A" is a parent of "B". The conditional probability of a node can be written as that node conditional on all of the parent nodes. The dashed lines represent that variables from the respective models are matched (see Chapter 4 for details on how they are matched).

the quality of the initial interpolant. A better approach, first suggested in Ramsay et al. [40], is to regularise the interpolants by the ODEs themselves. Dondelinger et al. [11] applied this idea to the non-parametric Bayesian approach of Calderhead et al. [8], using Gaussian processes (GPs), and demonstrated that it substantially improves the accuracy of parameter inference and robustness with respect to noise. As opposed to Ramsay et al. [40], all smoothness hyperparameters are consistently inferred in the framework of non-parametric Bayesian statistics, dispensing with the need to adopt heuristics and approximations. A graphical representation of the model is given in Figure 9(b).

This chapter furthers the work of Dondelinger et al. [11] by combining adaptive gradient matching using GPs with a parallel tempering scheme for the parameter that controls the mismatch between the gradients. This is conceptually different from the inference paradigm of the mismatch parameter that Dondelinger et al. [11] uses. Ideally, if the ODEs provide a correct mathematical description of the system, there should be no difference between the gradients of the interpolant and those predicted from the ODEs. However, in practice, forcing the gradients to be equal is likely to cause parameter inference methods to converge to a local optimum of the likelihood. Forcing the gradients to immediately be the same would restrict the inference procedure to a section of the likelihood corresponding to parameters that perfectly agree with the gradient match. However, there is no guarantee that

these parameters are suitable for the data, see Campbell and Steele [9] for details. A parallel tempering scheme is the natural way to deal with such local optima, as opposed to inferring the degree of mismatch, since different tempering levels correspond to different strengths of penalising the mismatch between the gradients. Campbell and Steele [9] explore a parallel tempering scheme, but in order to get an understanding as to how well utilising this scheme improves inference, the rest of the set-up (such as choice of interpolation scheme) should be as similar as possible. Hence, comparing the results directly to the GP approach in Dondelinger et al. [11], won't provide this understanding, since the approach in Campbell and Steele [9] uses a different methodological paradigm. This chapter describes the methodology for this new combined method and compares it with the methods by Dondelinger et al. [11] and Calderhead et al. [8]. The comparison to the method in Campbell and Steele [9], as well as a variety of other methodological paradigms, within the specific context of comparing the gradients from the interpolant to the gradients from the ODEs, is presented in Chapter 5.

4.2 Methodology

Consider a set of T arbitrary timepoints $t_1 < \cdots < t_i < \cdots < t_T$, and noisy observations $\mathbf{Y} = (\mathbf{y}(t_1), ..., \mathbf{y}(t_T))$, where

$$\mathbf{y}(t_i) = \mathbf{x}(t_i) + \boldsymbol{\epsilon}(t_i), \tag{49}$$

$N = \dim(\mathbf{x}(t_i))$, $\mathbf{X} = (\mathbf{x}(t_1), ..., \mathbf{x}(t_T))$, $\mathbf{y}(t_i)$ is the data vector of the observations of all species concentrations at time t_i, $\mathbf{x}(t_i)$ is the vector of the concentrations of all species at time t_i, $\mathbf{y}_s$ is the data vector of the observations of species concentrations s at all timepoints, $\mathbf{x}_s$ is the vector of concentrations of species s at all timepoints, $y_s(t_i)$ is the observed datapoint of the concentration of species s at time t_i, $x_s(t_i)$ is the concentration of species s at time t_i and $\boldsymbol{\epsilon}$ is multivariate Gaussian noise, $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma_s^2\mathbf{I})$.

The time-dependent signals of the system can be described by ordinary differential equations

$$\dot{\mathbf{x}}_s = \frac{d\mathbf{x}_s}{dt_i} = f_s(\mathbf{X}, \boldsymbol{\theta}_s, \mathbf{t}), \tag{50}$$

which can be represented in scalar form as

$$\dot{x}_s(t_i) = \frac{dx_s(t_i)}{dt_i} = f_s(\mathbf{x}(t_i), \boldsymbol{\theta}_s, t_i), \tag{51}$$

where $f_s(\mathbf{t}) = (f_s(t_1), \ldots, f_s(t_T))^\mathsf{T}$ and $\dot{\mathbf{x}}_s$ is the vector containing the gradients from the ODEs for species s at all timepoints.

Then,

$$p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2) = \prod_s \prod_t N(y_s(t_i)|x_s(t_i), \sigma_s^2), \tag{52}$$

where the dimension of the matrices $\mathbf{X}$ and $\mathbf{Y}$ are N by T. Following Calder-

head et al. [8], a Gaussian process (GP) prior is placed on $\mathbf{x}_s$,

$$p(\mathbf{x}_s|\boldsymbol{\phi}_s, \boldsymbol{\eta}) = N(\mathbf{x}_s|\boldsymbol{\phi}_s, \mathbf{K}_{\eta_s}), \tag{53}$$

where $\mathbf{K}_{\eta_s}$ is a positive definite matrix of covariance functions with hyperparameters η_s and $\boldsymbol{\phi}_s$ is a mean vector, which for simplicity we set as the mean of $\mathbf{Y}$ (which is possible since we assume a stationary process).

Differentiation is a linear operation, and therefore a Gaussian process is closed under differentiation ([46],[24][42]). Hence, the joint prior distribution of the concentrations of the species $\mathbf{x}_s$ and their time derivatives $\dot{\mathbf{x}}_s$ is multivariate Gaussian with mean $(\boldsymbol{\phi}_s, \mathbf{0})^\mathsf{T}$ and covariance functions

$$cov[x_s(t_i), x_s(t_j)] = K_{\eta_s}(t_i, t_j), \tag{54}$$

$$cov[\dot{x}_s(t_i), x_s(t_j)] = \frac{\partial K_{\eta_s}(t_i, t_j)}{\partial t_i} \;:=\; K'_{\eta_s}(t_i, t_j), \tag{55}$$

$$cov[x_s(t_i), \dot{x}_i(t_j)] = \frac{\partial K_{\eta_s}(t_i, t_j)}{\partial t_j} \;:=\; {}'K_{\eta_s}(t_i, t_j), \tag{56}$$

$$cov[\dot{x}_s(t_i), \dot{x}_s(t_j)] = \frac{\partial^2 K_{\eta_s}(t_i, t_j)}{\partial t_i \partial t_j} \;:=\; K''_{\eta_s}(t_i, t_j), \tag{57}$$

where $K_{\eta_s}(t_i, t_j)$ are the components of the covariance matrix $\mathbf{K}_{\eta_s}$. The conditional distribution for the state derivatives is obtained using elementary

transformations of Gaussian distributions (see page 87 of [6] for details), yielding

$$p(\dot{\mathbf{x}}_s|\mathbf{x}_s, \boldsymbol{\phi}_s, \boldsymbol{\eta}_s) = N(\boldsymbol{\mu}_s, \mathbf{A}_s), \tag{58}$$

where

$$\boldsymbol{\mu}_s = {}'\mathbf{K}_{\eta_s}{\mathbf{K}_{\eta_s}}^{-1}(\mathbf{x}_s - \boldsymbol{\phi}_s) \text{ and } \mathbf{A}_s = \mathbf{K}''_{\eta_s} - {}'\mathbf{K}_{\eta_s}{\mathbf{K}_{\eta_s}}^{-1}\mathbf{K}'_{\eta_s}. \tag{59}$$

Assuming the model for the gradients has additive Gaussian error, with a state-specific variance γ_s, using equation 50 gives

$$p(\dot{\mathbf{x}}_s|\mathbf{X}, \boldsymbol{\theta}_s, \gamma_s) = N(f_s(\mathbf{X}, \boldsymbol{\theta}_s, \mathbf{t}), \gamma_s\mathbf{I}). \tag{60}$$

Using a product of experts approach, Calderhead et al. [8] and Dondelinger et al. [11] link the interpolant in equation 58 with the ODE model in equation 60, giving the following distribution

$$\begin{aligned} p(\dot{\mathbf{x}}_s|\mathbf{X}, \boldsymbol{\theta}_s, \boldsymbol{\phi}_s, \boldsymbol{\eta}_s, \gamma_s) &\propto p(\dot{\mathbf{x}}_s|\mathbf{x}_s, \boldsymbol{\phi}_s, \boldsymbol{\eta}_s)p(\dot{\mathbf{x}}_s|\mathbf{X}, \boldsymbol{\theta}_s, \gamma_s) \\ &= N(\boldsymbol{\mu}_s, \mathbf{A}_s)N(f_s(\mathbf{X}, \boldsymbol{\theta}_s, \mathbf{t}), \gamma_s\mathbf{I}). \end{aligned} \tag{61}$$

Equation 61 can likely introduce an instability into the model and in fact, this is observed in the empirical results. The instability is discussed in Chapter 5, on page 90. The joint distribution is given by

$$p(\dot{\mathbf{X}}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma}|\boldsymbol{\phi}) = p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma}) \prod_s p(\dot{\mathbf{x}}_s|\mathbf{X}, \boldsymbol{\theta}_s, \boldsymbol{\phi}_s, \boldsymbol{\eta}_s, \gamma_s)p(\mathbf{x}_s|\boldsymbol{\eta}_s), \quad (62)$$

where $\boldsymbol{\gamma}$ is the vector which contains all the gradient mismatch parameters and $p(\boldsymbol{\theta}), p(\boldsymbol{\eta}), p(\boldsymbol{\gamma})$ are the prior distributions over the respective parameters. Dondelinger et al. [11] show that the marginalisation over the state derivatives yields a closed form solution

$$\begin{aligned} p(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma}|\boldsymbol{\phi}) &= \int p(\dot{\mathbf{X}}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma}|\boldsymbol{\phi}) d\dot{\mathbf{X}} \\ &\propto p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma}) \prod_s N(\mathbf{x}_s|\mathbf{0}, \mathbf{K}_{\eta_s}) \int N(\dot{\mathbf{x}}_s|\boldsymbol{\mu}_s, \mathbf{A}_s) N(\dot{\mathbf{x}}_s|f_s(\mathbf{X}, \boldsymbol{\theta}_s, \mathbf{t}), \gamma_s \mathbf{I}) d\dot{\mathbf{x}}_s \\ &\propto p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma}) \prod_s N(\mathbf{x}_s|\mathbf{0}, \mathbf{K}_{\eta_s}) \exp\left[-\frac{1}{2}(\mathbf{f}_s - \boldsymbol{\mu}_s)^T (\mathbf{A}_s + \gamma_s \mathbf{I})^{-1} (\mathbf{f}_s - \boldsymbol{\mu}_s)\right]. \end{aligned} \quad (63)$$

Using equation 63 and the noise model in equation 52, the full joint distribution becomes

$$p(\mathbf{Y}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\sigma}^2|\boldsymbol{\phi}) = p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2)p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\gamma})p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma})p(\boldsymbol{\sigma}^2), \quad (64)$$

where $p(\boldsymbol{\sigma}^2)$ is the prior over the variance of the observational error and

$$p(\mathbf{X}|\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\eta},\boldsymbol{\gamma}) \propto \frac{1}{C}\exp\left[-\frac{1}{2}\sum_{s}\left(\mathbf{x}_s^T\mathbf{K}_{\eta_s}^{-1}\mathbf{x}_s + (\mathbf{f}_s-\boldsymbol{\mu}_s)^T(\mathbf{A}_s+\gamma_s\mathbf{I})^{-1}(\mathbf{f}_s-\boldsymbol{\mu}_s)\right)\right], \tag{65}$$

where $C = \prod_s |2\pi(\mathbf{A}_s+\gamma_s\mathbf{I})|^{\frac{1}{2}}$ and $\mathbf{f}_s$ is the vector containing the ODE predicted gradients for species s. Sampling is conducted using MCMC and the whitening approach of Murray and Adams [37] is used to efficiently sample in the joint space of latent variables $\mathbf{X}$ and GP hyperparameters $\boldsymbol{\eta}$.

4.3 Parallel Tempering

Consider a series of "temperatures", $0 = \alpha^{(1)} < ... < \alpha^{(M)} = 1$ and a power posterior distribution of the ODE parameters ([15])

$$p_{\alpha^{(i)}}(\boldsymbol{\theta}^{(i)}|\mathbf{y}) \propto p(\boldsymbol{\theta}^{(i)})p(\mathbf{y}|\boldsymbol{\theta}^{(i)})^{\alpha^{(i)}}. \tag{66}$$

It is clear that equation 66 becomes the prior for $\alpha^{(i)} = 0$ and is the posterior when $\alpha^{(i)} = 1$. For $0 < \alpha^{(i)} < 1$ a distribution between the prior and posterior is created. The M $\alpha^{(i)}$s in equation 66 are annealed likelihoods that are used as the target densities of parallel MCMC chains ([9]). At each MCMC step, all "temperature" chains independently perform a Metropolis-Hastings step to update $\boldsymbol{\theta}^{(i)}$, the parameter vector associated with temperature $\alpha^{(i)}$

$$p_{\text{move}} = \min\left(1, \frac{p\left(\mathbf{y}|\boldsymbol{\theta}^{\text{prop}(i)}\right)^{\alpha^{(i)}} p\left(\boldsymbol{\theta}^{\text{prop}(i)}\right) q\left(\boldsymbol{\theta}^{\text{curr(i)}}|\boldsymbol{\theta}^{\text{prop(i)}}\right)}{p\left(\mathbf{y}|\boldsymbol{\theta}^{\text{curr}(i)}\right)^{\alpha^{(i)}} p\left(\boldsymbol{\theta}^{\text{curr}(i)}\right) q\left(\boldsymbol{\theta}^{\text{prop(i)}}|\boldsymbol{\theta}^{\text{curr(i)}}\right)}\right), \quad (67)$$

where $q(\ )$ represents the proposal distribution and the superscripts "prop" and "curr" indicate whether the algorithm is being evaluated at the proposed or current state, respectively. At each MCMC step, two chains are randomly selected (uniformly) and the corresponding parameters are proposed to swap between them. This proposal has acceptance probability

$$p_{\text{swap}} = \min\left(1, \frac{p_{\alpha^{(j)}}(\boldsymbol{\theta}^{(i)}|\mathbf{y}) p_{\alpha^{(i)}}(\boldsymbol{\theta}^{(j)}|\mathbf{y})}{p_{\alpha^{(i)}}(\boldsymbol{\theta}^{(i)}|\mathbf{y}) p_{\alpha^{(j)}}(\boldsymbol{\theta}^{(j)}|\mathbf{y})}\right). \quad (68)$$

The method developed in this chapter focuses on the intrinsic slack parameter γ_s (see equation 60), which theoretically should be $\gamma_s = 0$, since this corresponds to no mismatch between the gradients. In practice, to prevent the inference scheme from getting stuck in sub-optimal states, it is allowed to take on larger values $\gamma_s > 0$. However, rather than inferring γ_s like a model parameter, as Dondelinger et al. [11] do, other authors (e.g. [9]) state that γ_s should be gradually set to zero, since values closer to zero force the gradients to be more similar to one another and allow the interpolants to be informed by the ODEs. It is possible to abruptly set the values to zero, rather than gradually, however this is likely to cause the parameter inference techniques to converge to a local optimum of the likelihood. Hence, the gradient match-

ing with Gaussian processes approach in Dondelinger et al. [11] is combined with the tempering approach in Campbell & Steele [9] and this parameter is tempered to zero.

Prior to the parameter inference, values of γ_s are chosen and assigned to the variance parameter in equation 60 for each "temperature" $\alpha^{(i)}$, such that chains closer to the prior ($\alpha^{(i)}$ values closer to 0) allow the gradients from the interpolant to have more freedom to deviate from those predicted by the ODEs (which corresponds to larger γ_s values), chains closer to the posterior ($\alpha^{(i)}$ values closer to 1) more closely match the gradients (corresponding to smaller γ_s values), and for the chain corresponding to $\alpha^{(M)} = 1$, we want the mismatch to be approximately zero ($\gamma_s \approx 0$). Since γ_s corresponds to the variance of the species-specific error (see equation 60), as $\gamma_s \to 0$, there is less difference between the gradients, and as γ_s gets larger, the gradients have more freedom to deviate from one another. Hence, γ_s is tempered towards zero. Now, each $\alpha^{(i)}$ chain in equation 66 has a $\gamma_s^{(i)}$ (where the superscript (i) indicates the gradient mismatch parameter associated with "temperature" $\alpha^{(i)}$) fixed in place for the strength of the gradient mismatch. Since there is little knowledge as to optimal parameter schedules for the gradient mismatch parameter, two scheduling ladders are considered: in log_2 increments (referred throughout as LB2) and log_{10} increments (referred throughout as LB10). The specific schedules of the gradient mismatch parameter are included in Table 2.

Table 2: Ranges of the penalty parameter $\boldsymbol{\gamma}_s$ for LB2 and LB10. In this thesis $\boldsymbol{\gamma}_s = \boldsymbol{\gamma} \ \forall \ s$.

Method	**Chains**	**Range of Penalty $\boldsymbol{\gamma}$**	**Method**	**Chains**	**Range of Penalty $\boldsymbol{\gamma}$**
LB2	4	[1 , 0.125]	LB10	4	[1 , 0.001]
LB2	10	[1 , 0.00195]	LB10	10	[1 , $1e^{-9}$]

Table reproduced from [30], with permission from Springer.

4.4 Simulation

For comparison purposes, the simulations and the MCMC configuration were set up to correspond with that outlined in Dondelinger et al. [11]. For the GP, the radial basis function (rbf) kernel was used to model both systems of ODEs. The rbf kernel takes the form $k(t_i, t_j) = \sigma_{rbf}^2 \exp(-\frac{(t_i - t_j)^2}{2l^2})$, where σ_{rbf}^2 and l^2 are the hyperparameters (variance and characteristic lengthscale).

Fitz-Hugh Nagumo (FhN): The system can be found in Chapter 3, equations 40-41. The priors chosen for α and β were $N(0, 0.4^2)$, the prior for ψ, χ_2^2, true parameters, (0.2,0.2,3) and initial values for the "species", (-1,1), to correspond with Campbell and Steele [9]. 20 datapoints were evenly spaced over the time domain [0,10], since this produced one full period for each species. In Campbell and Steele [9] approximately 400 observations were simulated over 2 periods, but this felt as if it would not reflect the true sparseness of these types of datasets and so roughly 5% of this amount was used.

Lotka-Volterra (LV): The system can be found in Chapter 3, equation 42. The priors chosen for all the ODE parameters were $\Gamma(4, 0.5)$, true parameters, (2,1,4,1), initial values, (5,3) and 11 observations were evenly spaced over the time interval [0,2], to correspond with Dondelinger et al. [11].

For both the Fitz-Hugh Nagumo and Lotka-Volterra systems, Gaussian white noise, with standard deviation $\in \{0,0.1,0.5,0.8,1\}$, was added to represent observational error. These values were chosen to correspond with similar values to Dondelinger et al. [11]. For each system, method and noise level, 10 datasets were generated. By averaging over these, specific characteristics of a dataset can be removed and it is possible to observe more clearly a method's performance. The method of Dondelinger et al. [11] (referred to as INF in Chapter 4.5) was tested on both ODE models, as was the newly proposed model in this chapter. Code was not available for the Calderhead et al. [8] method and so the results obtained in the Dondelinger et al. [11] paper for the Calderhead et al. [8] method were used. This was only available for the Lotka-Volterra model and only for observational noise levels $\in \{0,0.1,0.5\}$. The results for LB2 and LB10 were similar, so only the LB10 results are shown.

4.5 Results

The posterior median was used as an estimator (since it is a robust estimator) of the parameters and the sampled parameter estimates were subtracted from

the true values

$$\text{accuracy} = \boldsymbol{\theta}_{\text{True}} - \text{median}(\boldsymbol{\theta}_{\text{Method}}), \tag{69}$$

where Method denotes the particular method used for parameter inference and the subscript True denotes the true parameter values. For the comparison of LB10 to INF (Dondelinger et al. [11]), the median was used as an estimator of the parameters and the sampled parameter estimates were subtracted from the true values for LB10 and INF and then these values were subtracted from one another

$$\text{accuracy} = |\boldsymbol{\theta}_{\text{True}} - \text{median}(\boldsymbol{\theta}_{\text{LB10}})| - |\boldsymbol{\theta}_{\text{True}} - \text{median}(\boldsymbol{\theta}_{\text{INF}})|\,, \tag{70}$$

where the subscript LB10 and INF denote the parameter estimates of the LB10 and INF methods, respectively. The distributions (of true value minus estimate) over the 10 datasets were compared. For both ODE systems, it was found that the rbf kernel provided a good fit to the data.

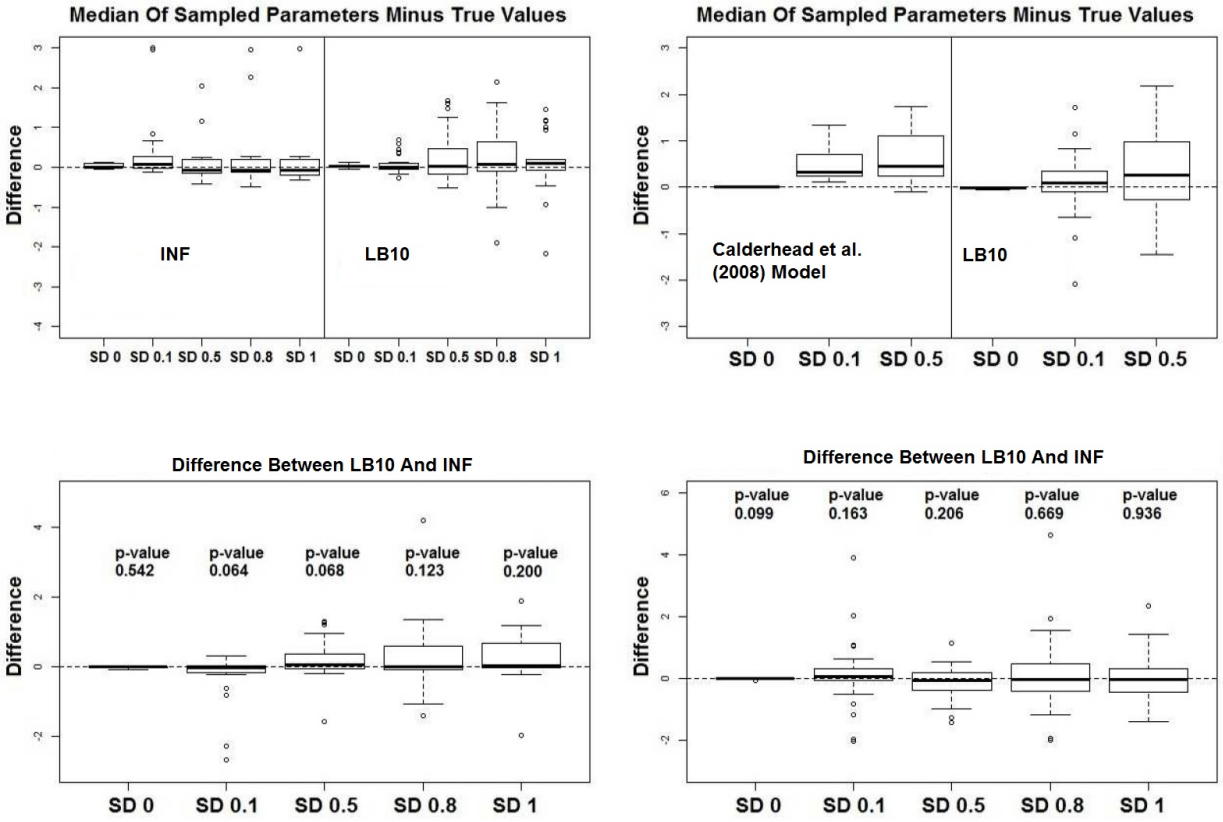


Figure 10: Parameter estimation accuracy (see equation 69) of $\boldsymbol{\theta}$ over noise instantiations, for the FhN (left) and LV (right) systems. Some outliers in the plots have been removed for scalability. Top Row: Boxplots, over the 10 datasets, of differences between the median of sampled parameters and true values. The dashed line is zero difference and the solid line splits the INF (Dondelinger et al. [11])/Calderhead et al. [8] (left) from the LB10 model (right). Bottom Row: Boxplots, over the 10 datasets, of the differences between parameter estimation accuracy for the INF and LB10, see equation 70. The dashed line is zero difference and the p-values for a paired t-test are shown above the corresponding boxplot.

The first row of Figure 10 shows the distribution of the estimate to the true parameter for the INF model, Calderhead et al. [8] and LB10 model, for

the FhN and LV systems. For zero noise, both the Calderhead et al. [8] and LB10 models have boxplots centred very close to zero, displaying good performance. However, when increasing the noise, the Calderhead et al. [8] no longer has a distribution centred around zero (no part of the distribution for noise = 0.1 and only a small part of the lower tail for noise = 0.5). For all noise instantiations, the LB10 (and INF) has most of its mass centred around zero. Therefore, if averaging over all datasets, for the LB10, the true parameters are close to the estimates. The second row of Figure 10 shows how robust the technique is. The plots show the distributions of the differences between the absolute distance of the estimator to the true parameter for the INF model and LB10 model. These distributions are centred around zero, indicating that there is no noticeable difference between the parameter estimation accuracy of these two techniques. It can therefore be seen that the new technique is robust to noise. It is worth noting that the LB10 increments were arbitrarily chosen, with the LB2 showing similar results.

As well as observing what the distributions of an estimator to the true parameter look like, it is also of interest to observe the full posterior distributions. Also, different parameters may have different properties, so it would be useful to observe the results split up by parameter. Hence, for observational noise level 0.5 (a signal to noise ratio of approximately 10), boxplots for the posterior distributions are shown in Figure 11 for the FhN system and Figure 12 for the LV system. The results across the remaining observational noise

levels are shown in Figures 44-47 for the FhN system and Figures 48-51 for the LV system, found in the appendix.

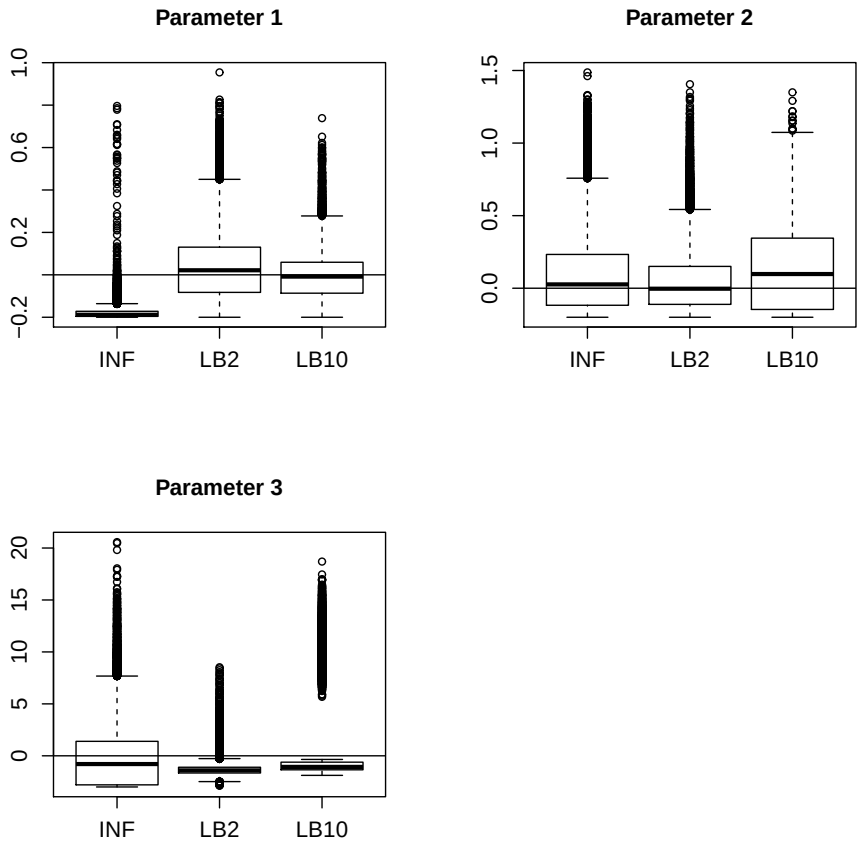


Figure 11: Posterior distributions over 10 datasets for the ODE parameters from the Fitz-Hugh Nagumo system, equations 40-41. The true parameters have been subtracted from the posterior distributions and the horizontal line shows zero difference to the true parameters. The observational noise level is 0.5 for this scenario.

By examining Figure 11, it can be seen that the LB2 and LB10 methods are slightly better at inferring parameter 1 in the Fitz-Hugh Nagumo system,

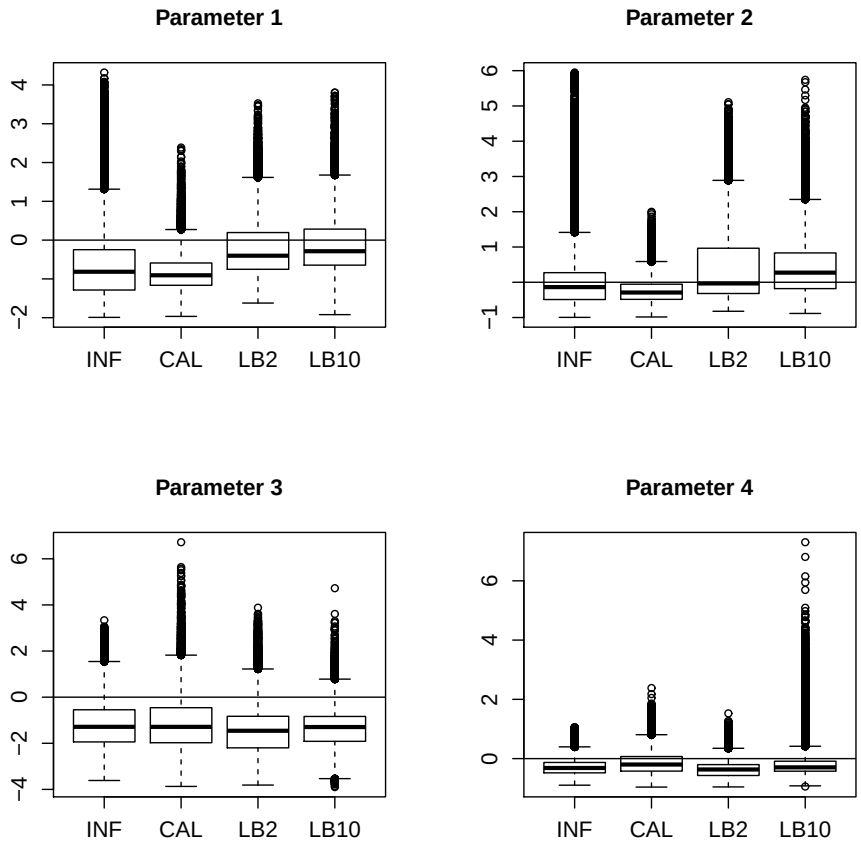


Figure 12: Posterior distributions over 10 datasets for the ODE parameters from the Lotka-Volterra system, equation 42. The true parameters have been subtracted from the posterior distributions and the horizontal line shows zero difference to the true parameters. The observational noise level is 0.5 for this scenario.

than the INF method, when the observational noise level is 0.5. The INF method is unbiased for parameter 3 and has a slightly larger variance than the LB2 and LB10 methods. The methods do equally well at inferring parameter 2. Figure 12 shows the results for the Lotka-Volterra system, for observational

noise level 0.5. LB2 and LB10 outperform the Calderhead et al. [8] and INF methods for parameter 1 (interquartile ranges include the true parameter). The methods all perform similarly for parameters 3 and 4 and have different bias/variance tradeoffs for parameter 2. The long tails for the methods INF, LB2 and LB10 methods are a consequence of the state variable concentrations flattening and is discussed in Chapter 5, page 90. The INF, LB2 and LB10 methods do not appear to be different to one another, overall, across the other noise levels (Figures 44-51 in the appendix).

4.6 Comparison with an Explicit Solution of the ODEs

Gradient matching is an approximate method to full Bayesian inference which is obtained by explicitly solving the differential equations, see Figures 9(a) and 9(b). In order to try to assess how well gradient matching approximates the full Bayesian inference approach, the results from Chapter 4.5 will be compared to results obtained by explicitly solving the ODEs.

To this end, data from Lotka-Volterra equation 42 was generated and Gaussian white noise with standard deviation 0.5 was added to represent observational noise. The priors chosen for all the ODE parameters were $\Gamma(4, 0.5)$, true parameters, (2,1,4,1), initial values, (5,3) and 11 observations were evenly spaced over the time interval [0,2]. The initial values of the system were inferred as additional model parameters.

The results presented are summaries (i.e. histograms and boxplots) of the merged samples from the posterior distributions of the replicate datasets. They are, therefore, samples of an expectation with respect to the sampling distribution of the posterior. The motivation is to free the results, to some extent, from the particular behaviour of any one dataset.

By examining Figure 13, it can be seen that distributions for the LB10 method always show slightly more variance than the explicit solution. For parameters 1 and 2, the results of both methods are similar. For parameters 3 and 4, the gradient matching results are of a similar distance away from the true parameters as with the explicit solution, but the opposite direction away from the true parameters than the explicit solution.

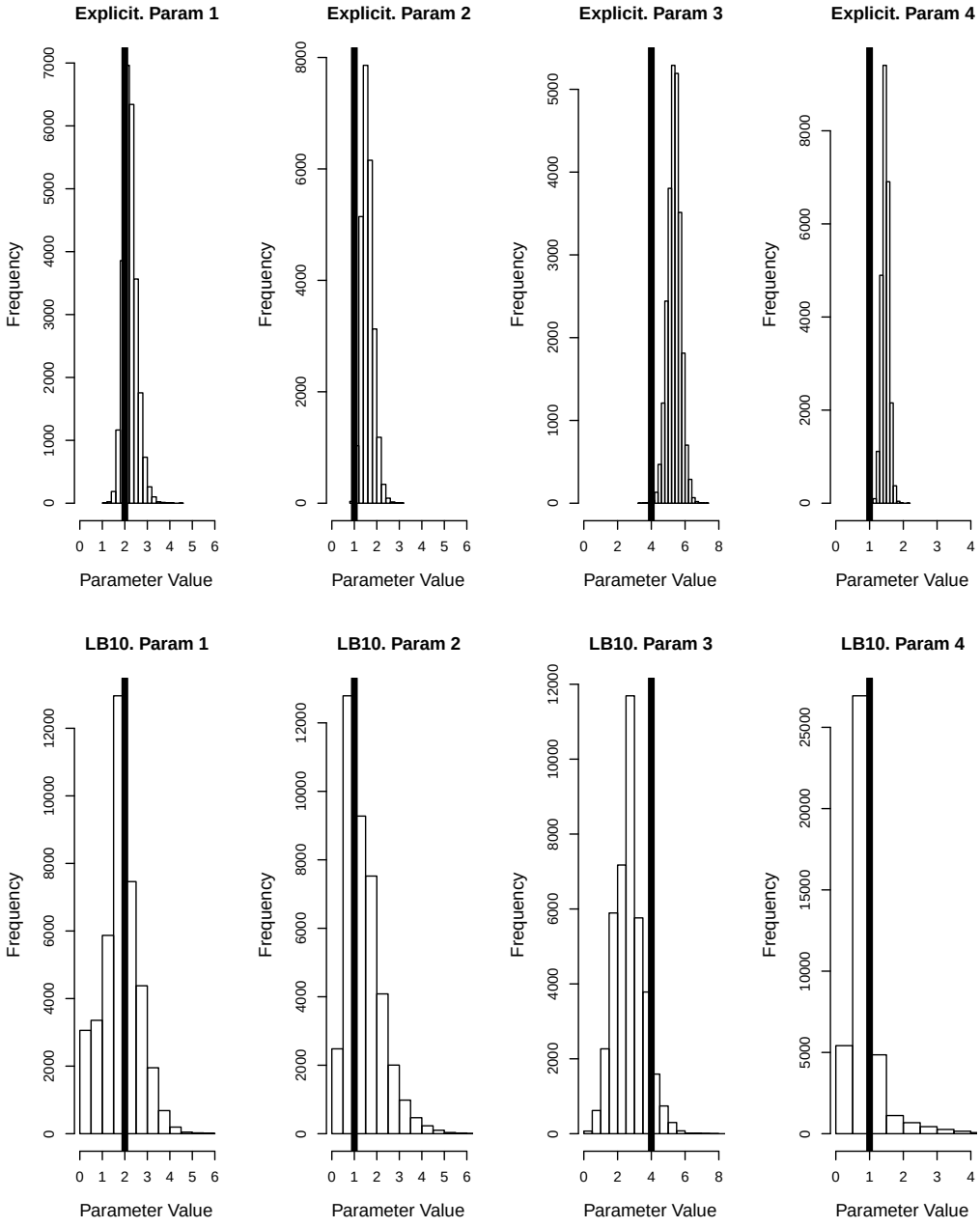


Figure 13: Posterior distributions over 10 datasets for the ODE parameters from the Lotka-Volterra system, equation 42. The top row contains the results obtained by using the explicit solution of the ODEs. The bottom row contains the results obtained from gradient matching, LB10 method. The vertical line represents the true parameter value.

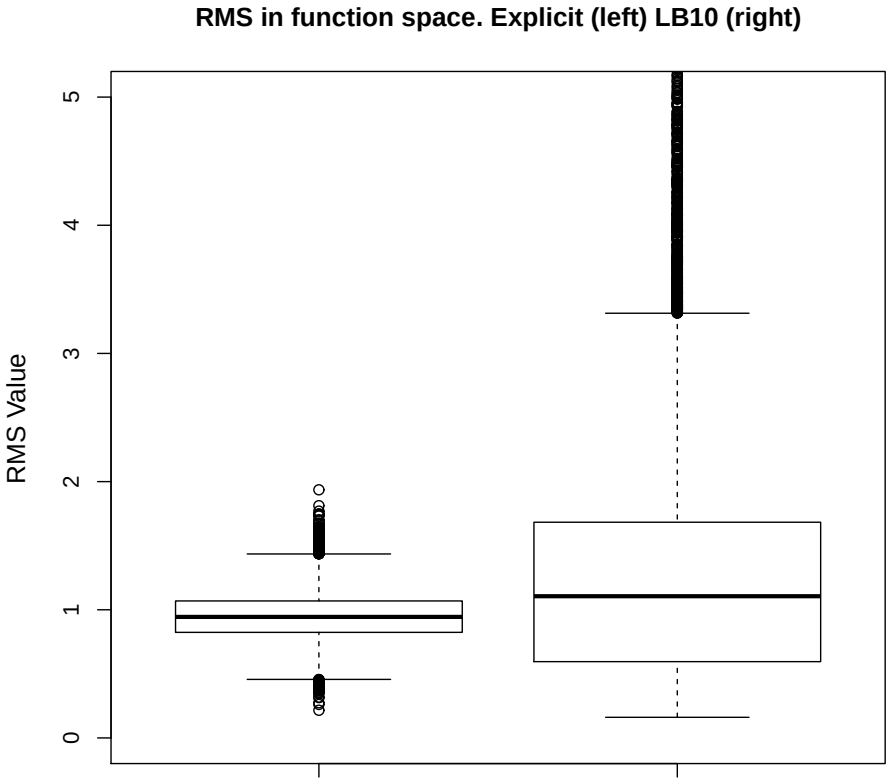


Figure 14: RMS values in function space over 10 datasets, calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced with the estimate of the parameters, for the Lotka-Volterra system, equation 42. One value in the plot represents the RMS value produced from one dataset and the parameter sample from one iteration in the MCMC. The left boxplot contains the results obtained by using the explicit solution of the ODEs. The right boxplot contains the results obtained from gradient matching, LB10 method. The LB10 gradient matching method produces a distribution that is about twice the variance of the explicit solution and has a longer tail. The boxplots for both methods show similar RMS performance (the centre of the distributions are in a similar location), indicating that although the gradient matching method is not as accurate as the explicit solution, the decrease in performance is not substantial. Some outliers for the LB10 method have been omitted for scalability.

It is difficult to see how well the results compare to one another, comparing them in parameter space only. To this end, root mean square (RMS) values in function space, calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced with the estimate of the parameters, were produced. The results from Figure 14 show that the LB10 gradient matching method produces a distribution that is about twice the variance of the explicit solution and has a longer tail. The boxplots for both methods show similar RMS performance (interquartile range covering similar ranges), indicating that although the gradient matching method is not as accurate as the explicit solution, the decrease in performance is not substantial. Some outliers for the LB10 method have been omitted for scalability. The reason for the outliers was discovered to be a consequence of the state variable concentrations flattening and is discussed in Chapter 5, page 90. The results for the INF and LB2 methods were virtually identical and therefore only the LB10 results are shown.

Comparative computational times for the explicit solution of the ODEs and the gradient matching methods are available in Tables 12-13, in the appendix.

4.7 Conclusion

An evaluation of two alternative schemes for adaptive gradient matching: posterior inference vs. parallel tempering of the gradient mismatch parameter, has been presented. The tempering scheme was originally proposed

in the context of splines-based regression, which has been adapted to non-parametric Bayesian modelling, with Gaussian processes. An application to data, generated from two different systems of ODEs, shows no overall difference between the parallel tempering and posterior inference. The simulation set-up however was not extensive, since this was an initial test of the newly proposed method and a wider comparative analysis is required to better understand the method's performance and limitations. This extensive comparative analysis is presented in Chapter 5.

When comparing the newly developed method to parameter inference with an explicit solution of the ODEs, it was found that there was reasonable consistency between the approaches. As expected, the results for the explicit solution were better, showing a narrower root mean square error in function space than the new method. The new method produces similar parameter estimates to that of the explicit method, for parameters 1 and 2 of the Lotka-Volterra system, equation 42. For parameters 3 and 4, the gradient matching results are of a similar distance away from the true parameters as with the explicit solution, but the opposite direction away from the true parameters than the explicit solution. The RMS distribution has about twice the variance than that of the explicit solution, but the decrease in performance is not substantial.

5 Comparative Analysis with the Current State-of-the-Art Gradient Matching Methods

This chapter presents work published in Macdonald and Husmeier [30], Macdonald and Husmeier [29] and Macdonald et al. [33]. Software is available at `http://researchdata.gla.ac.uk/288/`. Note: the implementation of the software for the method of González et al. [19] in this chapter was carried out by M. Niu.

5.1 Brief summary of methods

This chapter conducts a wide scale comparative analysis with the newly proposed method in Chapter 4, the method in Dondelinger et al. [11] and the methods detailed in Chapter 2.

The following is a brief summary of all the methods that are compared in this chapter. Since many methods and settings are used in this chapter for comparison purposes, abbreviations are used for ease of reading. Table 3 contains a key for those methods.

C&S [9]: Parameter inference is carried out using adaptive gradient matching and tempering of the mismatch parameter. B-splines are used as the choice of interpolation scheme. **INF** [11]: This method conducts parameter inference through adaptive gradient matching using Gaussian processes.

The penalty mismatch parameters $\boldsymbol{\gamma}_s$ are inferred. **LB2**: This method conducts parameter inference through adaptive gradient matching using Gaussian processes. The penalty mismatch parameters $\boldsymbol{\gamma}_s$ are tempered in log base 2 increments, see Table 2 for details. **LB10**: As with LB2, parameter inference is conducted through adaptive gradient matching using Gaussian processes, however, the penalty mismatch parameters $\boldsymbol{\gamma}_s$ are tempered in log base 10 increments, see Table 2 for details. **GON** [19]: Parameter inference is conducted in a non-Bayesian fashion, implementing a reproducing kernel Hilbert space (RKHS) and penalised likelihood approach. Comparisons between RKHS and GPs have been previously explored conceptually (for example, see [42], [36]), and in this chapter they are analysed empirically in the specific context of inference in ODEs. The RKHS method that incorporates the information from the ODEs in González et al. [19] obtains the ODE kernel using a differencing operator. AIC is used to estimate the penalty parameter λ. **GON Cross** [19]: The method is the same as **GON**, however, cross validation is used to estimate the penalty parameter λ, instead of AIC. **RAM** [40]: This technique uses a non-Bayesian optimisation process for parameter inference. The method penalises the difference between the gradients using splines and a hierarchical 2 level regularisation approach is used to set the tuning parameters (see [40] for details). Table 4 describes particular settings with some of the methods in Table 3. The ranges of the penalty parameters $\boldsymbol{\gamma}_s$, for the LB2 and LB10 methods are given in Table 2. The increments are linear on the log scale. The M α_ss range from 0 to 1

and are set by taking a series of M equally spaced values and raising them to the power 5 (since Friel and Pettitt [15] empirically discovered that this power yielded better results).

Table 3: Abbreviations of the methods used throughout this chapter.

Abbreviation	**Method**	**Reference**
C&S	Tempered mismatch parameter using splines-based smooth functional tempering.	Campbell & Steele [9]
INF	Inference of the gradient mismatch parameter using GPs.	Dondelinger et al. [11]
LB2	Tempered mismatch parameter using GPs in Log Base 2 increments.	New method in Chapter 4
LB10	Tempered mismatch parameter using GPs in Log Base 10 increments.	New method in Chapter 4
GON	Reproducing kernel Hilbert space and penalised likelihood. The penalty parameter is estimated using AIC.	González et al. [19]
GON Cross	Reproducing kernel Hilbert space and penalised likelihood. The penalty parameter is estimated using 3-fold cross validation.	González et al. [19]
RAM	Hierarchical 2 level regularisation approach using splines based interpolation.	Ramsay et al. [40]

Table reproduced from [30], with permission from Springer.

Table 4: Particular settings of Campbell & Steele's [9] method.

Abbreviation	Definition	Details
10C	10 Chains	When comparing methods, it was of interest to see how the performance depended on the number or parallel MCMC chains, as originally the authors used 4 chains.
Obs20	20 Observations	Originally, the authors use 401 observations. This was reduced to a dataset size more usual with these types of experiments to observe the dependency of the methods on the amount of data.
15K	15 Knots	The method in C&S uses B-splines interpolation. The original tuning parameters from the author's paper were changed to observe the sensitivity of the parameter estimation by these tuning parameters.
P3	Polynomial order 3 (Cubic Spline)	The original polynomial order is 5 and again, it was of interest to observe the sensitivity of the parameter estimation by these tuning parameters.

Table reproduced from [30], with permission from Springer.

5.2 Simulation

The proposed GP tempering scheme in Chapter 4 is compared with the alternative methods summarised in Chapter 2. For the comparison to Ramsay et al. [40], the authors' software was unavailable and so the results were compared directly with the results from the original publication. Hence, test data was generated in the same manner as described by the authors and used for the evaluation of the new method in Chapter 4. For the methods in Campbell and Steele [9], Dondelinger et al. [11] and González et al. [19], where the authors' software was obtainable, the evaluation was repeated twice, first on data equivalent to those used in the original publications, and again on new data generated with different (more realistic) parameter settings. For

comparisons using the Fitz-Hugh Nagumo model, equations 40-41, the ODE prior distributions in Campbell and Steele [9] were used and for comparisons using the protein signalling transduction pathway model, equation 45, the parameter priors from Dondelinger et al. [11] were used. This gives priors that were motivated by the current literature.

Tempered mismatch parameter using splines-based smooth functional tempering (**C&S**) [9]: The authors tested their method on the Fitz-Hugh Nagumo system, equations 40-41, with the following parameter settings: $\alpha = 0.2$, $\beta = 0.2$ and $\psi = 3$, starting from initial values of $(-1, 1)$ for the two "species". They generated 401 observations over the time course [0, 20] (producing 2 periods) and Gaussian noise with sd $\{0.5, 0.4\}$ was used to corrupt each respective "species". To infer the ODE parameters with their approach, the authors chose the following settings: B-splines of polynomial order 5 with 301 knots; 4 parallel tempering chains, gradient mismatch parameter schedules {10,100,1000,10000}; parameter prior distributions for the ODE parameters: $\alpha \sim N(0, 0.4^2)$, $\beta \sim N(0, 0.4^2)$ and $\psi \sim \chi^2_2$.

As well as comparing the new method in Chapter 4 with the results the authors had obtained with their original settings (described in the previous paragraph), the following modifications were made to test the robustness of their procedure. The number of observations were reduced from 401 to 20 over the time course [0, 10] (producing 1 period), which more closely reflects

the amount of data typically available in current systems biology. In doing so, the number of knots were also reduced for the splines to 15 (preserving the same proportionality of knots to datapoints as before), a different polynomial order was tried: 3 instead of 5. The method incurred high computational costs, (roughly $1\frac{1}{2}$ weeks for a run), and so inference could only be run on 3 independent datasets. The posterior samples were combined in order to approximately marginalise over datasets and thereby remove their potential particularities. For a fair comparison, the new method in Chapter 4 was also run with 4 rather than the 10 chains that were used as default.

Inference of the gradient mismatch parameter using GPs and adaptive gradient matching (**INF**) [11]: The method was applied in the same way as described in the original publication of Dondelinger et al. [11], using the authors' software and selecting the same kernels and parameter/hyperparameter priors for the method proposed in the present paper. Data was generated from the protein signal transduction pathway described in equation 45, with the same settings as in Dondelinger et al. [11]; initial values of the species: $(S = 1, dS = 0, R = 1, RS = 0, Rpp = 0)$; ODE parameters: $(k_1 = 0.07, k_2 = 0.6, k_3 = 0.05, k_4 = 0.3, V = 0.017, K_m = 0.3)$; 15 timepoints producing one period: $\{0, 1, 2, 4, 5, 7, 10, 15, 20, 30, 40, 50, 60, 80, 100\}$. As in Dondelinger et al. [11], multiplicative iid Gaussian noise (additive iid Gaussian noise on the log scale) of standard deviation = 0.1 was used to corrupt the signals and reflect the noisy observations obtained in experiments.

The same gamma prior on the ODE parameters was chosen, as used in Dondelinger et al. [11], namely $\Gamma(4, 0.5)$, for Bayesian inference. For the GP, the same kernel they originally used was implemented; see page 69 for details. In addition to this ODE system, this method was also applied to the rest of the described set-ups.

Reproducing kernel Hilbert space method (**GON**) [19]: The authors tested their method on the Fitz-Hugh Nagumo data (equations 40-41) with the following settings; initial values of $(-1, -1)$ and ODE parameters of $\alpha = 0.2$; $\beta = 0.2$ and $\psi = 3$. The authors generated 50 datapoints over the time domain [0, 20] (producing 2 periods), with iid Gaussian noise (sd = 0.1) added to introduce error to the observations. 50 independent datasets were created in this way.

As well as comparing to the original publication set-up, the methods were tested on a scenario with larger observational noise. They were tested on 2 scenarios, when the signal to noise ratio was on average 10 for each species and when the average signal to noise ratio was 5. The average signal to noise ratio was used so that each species had the same observational error as one another. The dataset size was reduced to 25 timepoints over the time course [0,10], producing 1 period, and the results across 10 independent datasets are shown.

To observe the variation between ODE models, the method was also run on the protein signal transduction pathway in equation 45. Data under the same settings as in Dondelinger et al. [11] were generated; ODE parameters: $(k_1 = 0.07, k_2 = 0.6, k_3 = 0.05, k_4 = 0.3, V = 0.017, K_m = 0.3)$; initial values of the species: $(S = 1, dS = 0, R = 1, RS = 0, Rpp = 0)$; 15 timepoints covering one period: $\{0, 1, 2, 4, 5, 7, 10, 15, 20, 30, 40, 50, 60, 80, 100\}$. 2 noise scenarios were examined; when the average signal to noise ratio was 10, and when the average signal to noise ratio was 5. As opposed to the set-up in Dondelinger et al. [11], additive Gaussian noise was used to corrupt the data, to correspond with the assumed noise model.

Penalised splines & 2^{nd} derivative penalty method (RAM) [40]: González et al. [19] used the method of Ramsay et al. [40] to compare to their technique. The results from the original publication of González et al. [19] are presented. For fairness of comparison, the new method in Chapter 4 was applied in the same way as with the set-up in [19].

Choice of kernel: For the Gaussian process, a suitable kernel needs to be chosen, which reflects prior knowledge in function space. Two kernels were considered in this study (to correspond with the authors' set-ups), the radial basis function (RBF) kernel

$$k(t_i, t_j) \;=\; \sigma^2_{\mathrm{RBF}} \exp(-\frac{(t_i - t_j)^2}{2l^2}) \tag{71}$$

with hyperparameters σ^2_{RBF} and l^2, and the sigmoid variance kernel

$$k(t_i, t_j) \;=\; \sigma^2_{\text{sig}} \arcsin \frac{a + (bt_i t_j)}{\sqrt{(a + (bt_i t_i) + 1)(a + (bt_j t_j) + 1)}} \tag{72}$$

with hyperparameters σ^2_{sig}, a and b [42].

To initialise the hyperparameters, a standard GP regression model (i.e. without information from the ODE) was fitted by maximum likelihood. It was then checked to see whether the interpolant adequately represented the prior knowledge. In practice, this would be available from experts involved in the experiment. From previous observations, it is possible to gain some insight into how rough or smooth a particular signal or process might be. The initialised GP would then be inspected to see whether it is over- or under-smoothed compared to what is expected.

It was found that the RBF kernel provided a good fit to the data for the data generated from the Fitz-Hugh Nagumo model. However, in confirmation of the findings in Dondelinger et al. [11], it was found that for the protein signalling transduction pathway, the non-stationary nature of the data is not represented properly with the RBF kernel, which is stationary [42]. As in Dondelinger et al. [11], the sigmoid variance kernel was used, which is non-stationary [42] and found a considerable improvement to the fit to the data.

Other settings: The values for the variance mismatch parameter of the

gradients, γ_s, need to be set. Since studies that indicate reasonable values for our technique are limited (see [8], [15]), Log_2 and Log_{10} increments with an initial start at 1 were used. All parameters were initialised with a random draw from the respective priors (apart from GON, which did not use priors).

5.3 Results

Tempered mismatch parameter using splines-based smooth functional tempering (**C&S**) [9]: By examining Figures 15-17, it can be seen that the C&S method shows good performance over all parameters in the one case where the number of observations is 401, the number of knots is 301 and the polynomial order is 3 (cubic spline), since the bulk of the distributions of the sampled parameters surround the true parameters in Figures 15 and 17 and are close to the true parameter in Figure 16. These settings, however, require a great deal of "hand-tuning" or time expensive cross-validation and would be very difficult to set when using real data. The sensitivity of the method can be observed by examining the other set-ups, where the results are noticeably worse. An important point to note is when the dataset size was reduced, the cubic spline performed very badly. This lack of robustness makes these splines based methods very difficult to apply in practice. The INF, LB2 and LB10 methods consistently outperform the C&S method with distributions being closer to or overlapping the true parameters. On the set-up with 20 observations, for both 4 and 10 chains, the INF method produced largely different estimates across the datasets, as depicted by the

wide boxplots and long tails.

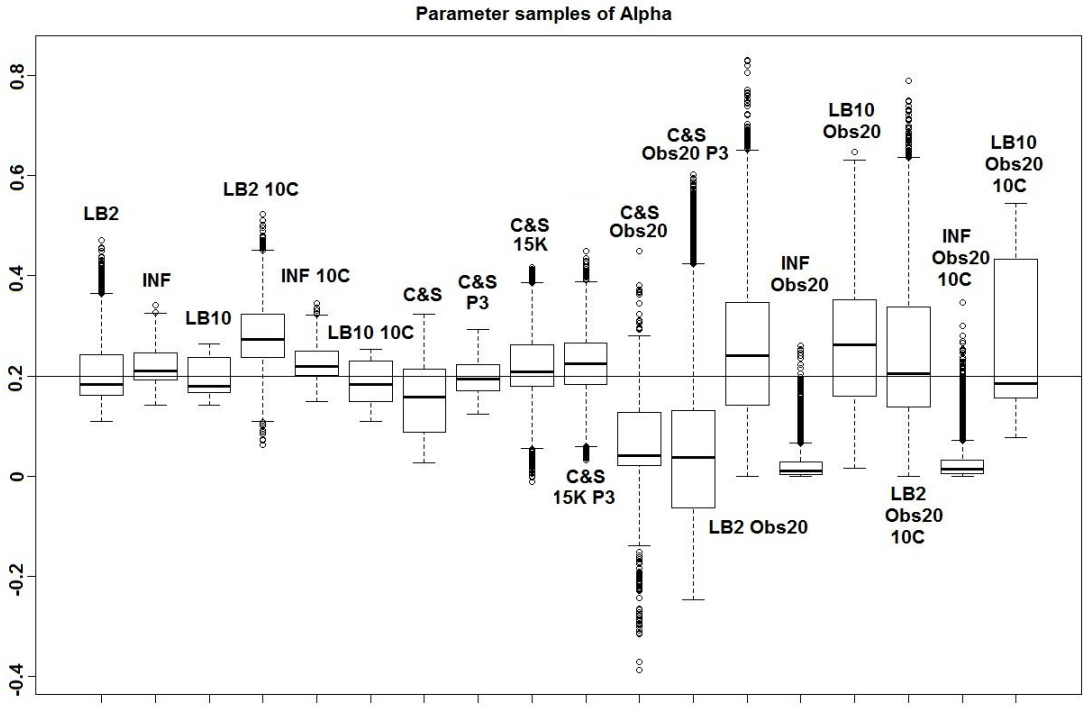


Figure 15: Average posterior distributions of parameter α from the Fitz-Hugh Nagumo model (equation 41) over 3 datasets. From left to right: LB2, INF, LB10, LB2 10C, INF 10C, LB10 10C, C&S, C&S P3, C&S 15K, C&S 15K P3, C&S Obs20, C&S Obs20 P3, LB2 Obs20, INF Obs20, LB10 Obs20, LB2 Obs20 10C, INF Obs20 10C and LB10 Obs20 10C. The solid line is the true parameter. For definitions, see Tables 3 and 4. Figure reproduced from [30], with permission from Springer.

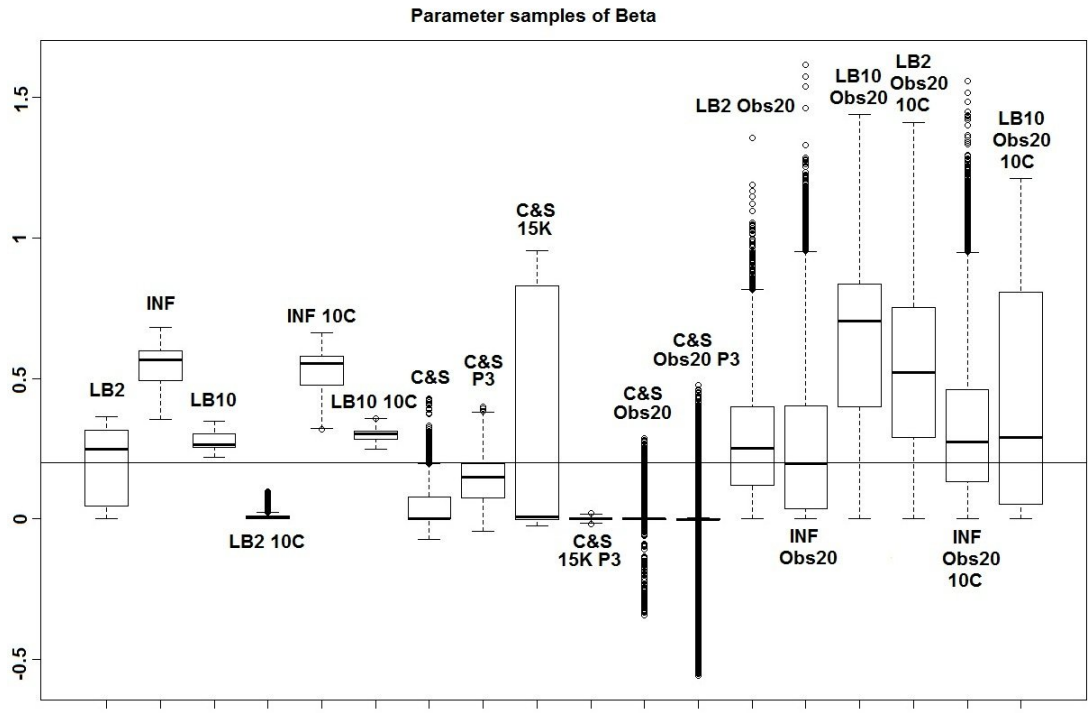


Figure 16: Average posterior distributions of parameter β from the Fitz-Hugh Nagumo model (equations 41) over 3 datasets. From left to right: LB2, INF, LB10, LB2 10C, INF 10C, LB10 10C, C&S, C&S P3, C&S 15K, C&S 15K P3, C&S Obs20, C&S Obs20 P3, LB2 Obs20, INF Obs20, LB10 Obs20, LB2 Obs20 10C, INF Obs20 10C and LB10 Obs20 10C. The solid line is the true parameter. For definitions, see Tables 3 and 4. Figure reproduced from [30], with permission from Springer.

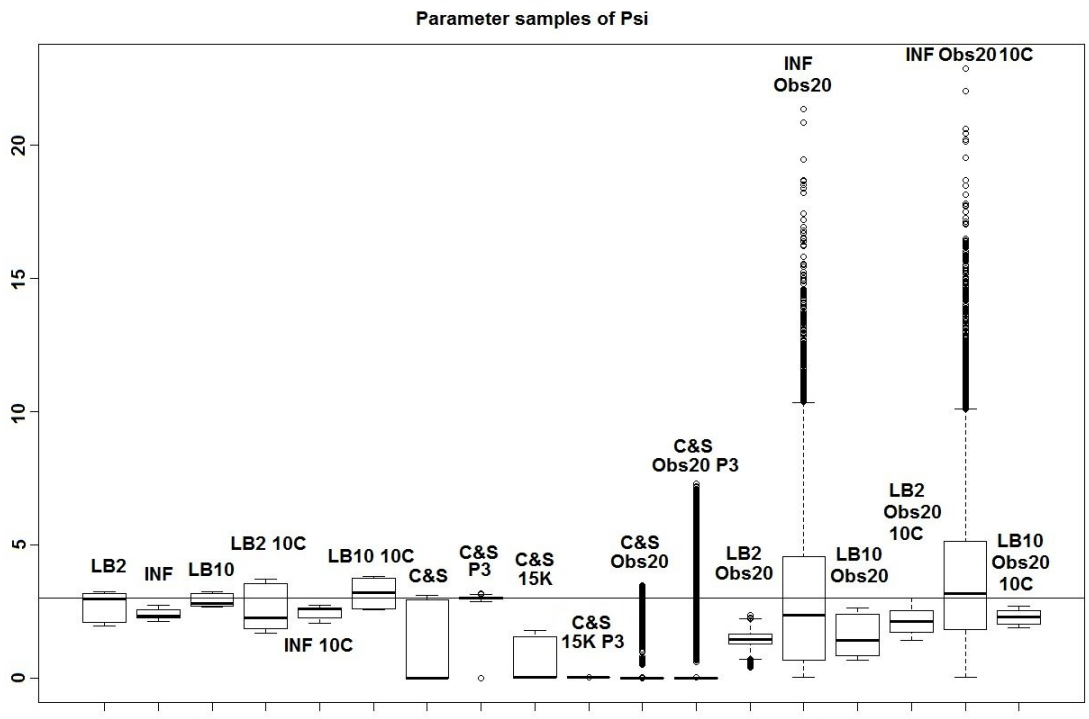


Figure 17: Average posterior distributions of parameter ψ from the Fitz-Hugh Nagumo model (equations 40-41) over 3 datasets. From left to right: LB2, INF, LB10, LB2 10C, INF 10C, LB10 10C, C&S, C&S P3, C&S 15K, C&S 15K P3, C&S Obs20, C&S Obs20 P3, LB2 Obs20, INF Obs20, LB10 Obs20, LB2 Obs20 10C, INF Obs20 10C and LB10 Obs20 10C. The solid line is the true parameter. For definitions, see Tables 3 and 4. Figure reproduced from [30], with permission from Springer.

Inference of the gradient mismatch parameter using GPs, adaptive method (**INF**) [11]: In order to see how the LB2 and LB10 tempering methods perform in comparison to the INF method, the results from the protein signalling transduction pathway (see equation 45) can be examined, as well as comparing how each method did in the other set-ups. Figure 18 shows the distributions of parameter estimates minus the true values for the protein signalling transduction pathway. After implementing the authors' code, it was noted that some of the MCMC simulations had not converged. In order to present a fair depiction of the methods' performance, the results from the dataset that produced the median performance are shown. For each dataset the root mean square was calculated on the parameter samples minus the true values. The dataset that produced the median root mean square value is given.

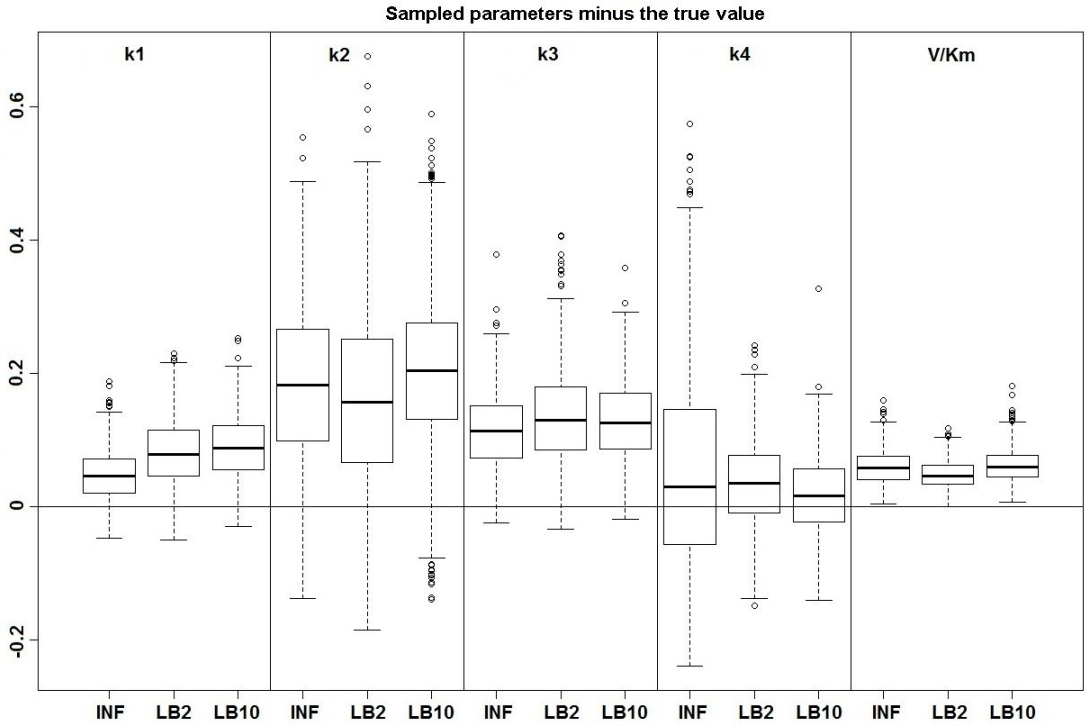


Figure 18: Results from the dataset that showed the average RMS of the posterior parameter samples minus the true values for the INF, LB2 and LB10 methods. The posterior distributions are of the sampled parameters from the protein signalling transduction pathway (equation 45) minus the true value. The horizontal line shows zero difference. For definitions, see Tables 3 and 4. Figure reproduced from [30], with permission from Springer.

By examining Figure 18, it can be seen that for each parameter the methods produce distributions that are not more than twice the interquartile range away from the true parameter. For this set-up, overall there does not appear to be a significant difference between the INF, LB2 and LB10 methods.

For the original set-up in [19], Figure 19 shows the expected cumulative distribution functions (ECDFs) of the absolute errors of the parameter samples for the tempering and inference schemes. P-values for 2-sample, 1-sided Kolmogorov-Smirnov tests are given. Since the distributions are of the average error, if a distribution's ECDF is significantly higher than another's, this constitutes better parameter estimation. A higher curve means that there are more values located in the lower range of absolute error.

By examining Figure 19 and using the standard significance level of 0.05 as a cut-off, it can be seen that the ECDFs for LB2 and LB10 are significantly higher than those for INF. This means that the parameter estimates from the LB2 and LB10 methods are closer to the true parameters than the INF method, since we are dealing with absolute error. The LB2 and LB10 method show no significant difference to each other.

As an alternative presentation, the absolute errors of the parameter estimation for the INF, LB2 and LB10 methods are depicted as boxplots in Figure 20.

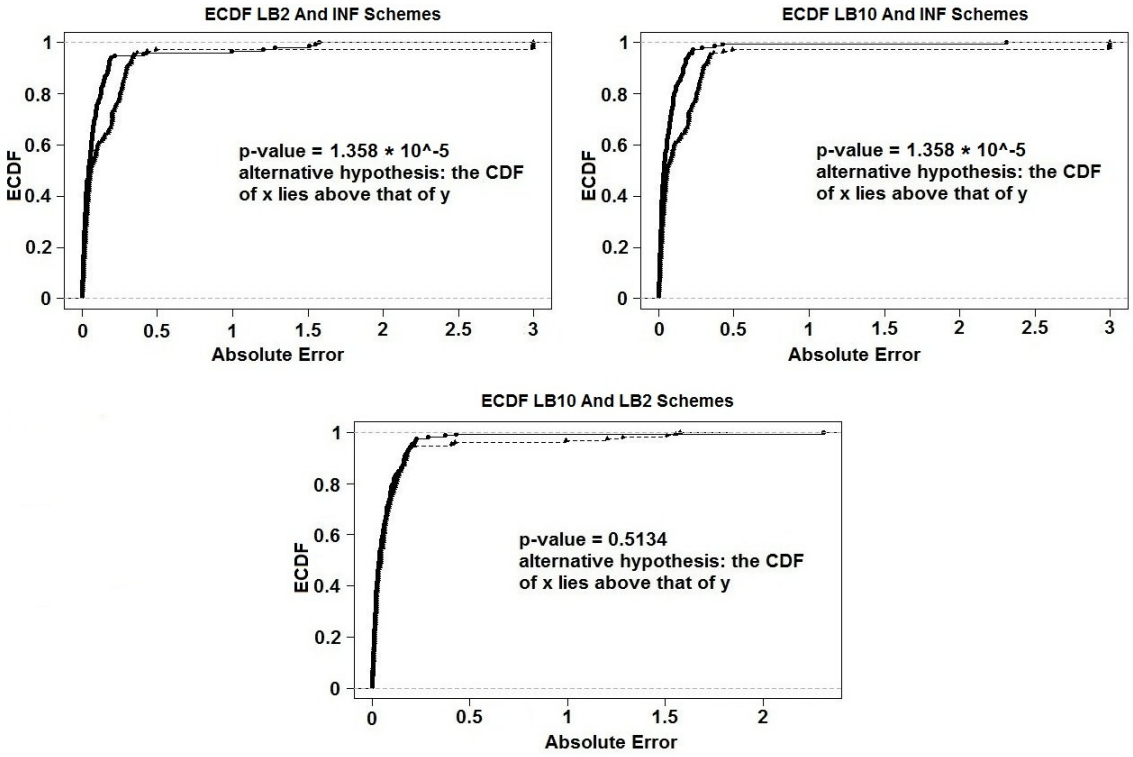


Figure 19: ECDFs of the absolute errors of the parameter estimation for the Fitz-Hugh Nagumo system (equations 40 and 41). Top left - ECDFs for LB10 and INF, top right - ECDFs for LB2 and INF and bottom - ECDFs for LB10 and LB2. Included are the p-values for 2-sample, 1-sided Kolmogorov-Smirnov tests. For definitions, see Tables 3 and 4. Figure reproduced from [30], with permission from Springer.

By examining Figure 20, it can be seen that the variance of absolute error to the true parameters is about half for the LB2 and LB10 methods compared to INF.

For the set-up in [9], Figures 15-17 show that the LB2 and LB10 methods perform well across dataset size and over all the parameters, since most of the

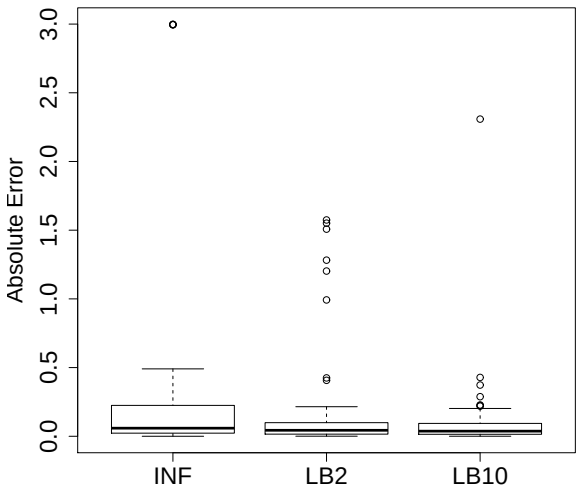


Figure 20: Boxplots of the absolute errors of the parameter estimation for the Fitz-Hugh Nagumo system (equations 40 and 41). The distributions of the absolute errors is given for the INF, LB2 and LB10 method (from left to right). For definitions, see Tables 3 and 4.

mass of the distributions surround or are situated close to the true parameters. One type of scheduling did not always outperform another. The LB2 does better than the LB10 for 4 parallel chains (distributions overlapping the true parameter for all three parameters) and the LB10 outperforms the LB2 for 10 parallel chains (distribution overlapping true parameter in Figure 15, being closer to the true parameter in Figure 16 and narrower and more symmetric around the true parameter in Figure 17). The bulks of parameter sample distributions for the INF method are located close to the true parameters for all dataset sizes. However, the method produces less uncertainty at the expense of bias. When reducing the dataset size to 20 observations, for

both 4 and 10 chains, the results deteriorate for the INF method and it is outperformed by the LB2 and LB10 methods.

Reproducing kernel Hilbert space (**GON**) [19] and **Hierarchical regularisation splines based method** (**RAM**) [40]: For these sets of results, to assess the performance of the methods, the same criterion as in GON was used. For each parameter, the absolute value of the difference between an estimator and the true parameter ($|\hat{\theta}_i - \theta_i|$) was computed and the distribution across the datasets was examined. For the LB2, LB10 and INF methods, the median of the sampled parameters was used as an estimator, since it is a robust average. Examining Figure 21, the LB2, LB10 and INF methods do as well as the GON method for 2 parameters (INF doing slightly worse for ψ) and outperform it for 1 parameter with the width of the distributions of the absolute distances to the true parameter roughly $\frac{1}{3}$ of the size. All methods outperform the RAM method.

Looking at Figure 22, the distributions of parameter 3 for LB2 and LB10 are about 5 times the absolute distance away than the other methods from the true parameter. When the noise is increased, Figure 23, the GON and GON Cross methods are slightly more robust in estimating the final parameter.

The final parameter in the Fitz-Hugh Nagumo system is the only parameter modelling Voltage, see equation 40. This species is particularly difficult for

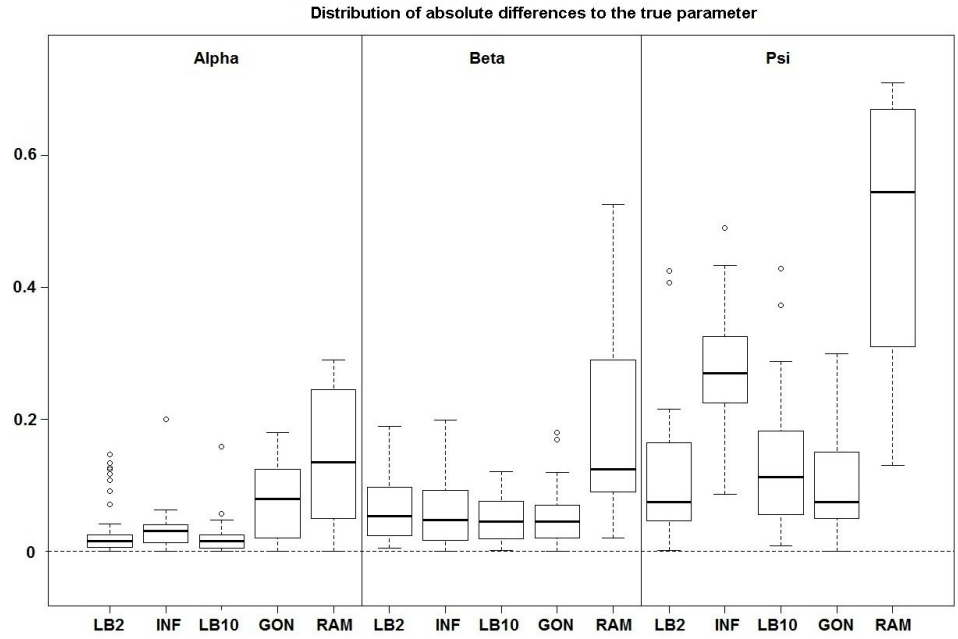


Figure 21: Boxplots of the distributions of the absolute differences of an estimate to the true parameter over 50 datasets. The three sections from left to right represent the parameters α, β and ψ from the Fitz-Hugh Nagumo model (equations 40-41). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, GON method (boxplot reconstructed from [19]) and RAM method (boxplot reconstructed from [19]). For definitions, see Tables 3 and 4. Figure reproduced from [30], with permission from Springer.

the INF, LB2 and LB10 methods, due to sharp changes in the signal (see Figure 2), as the GP currently assumes a more smooth change overall. This results in a deterioration of the parameter estimation performance.

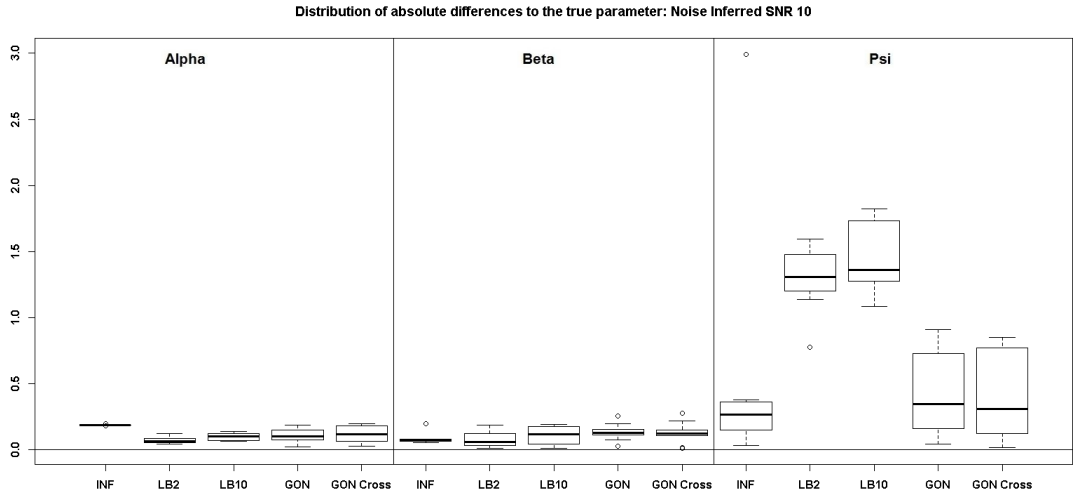


Figure 22: Boxplots of the distributions of the absolute differences of an estimate to the true parameter over 10 datasets. The three sections from left to right represent the parameters α, β and ψ from the Fitz-Hugh Nagumo model (equations 40-41). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, GON method and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 10. The standard deviation of the observational noise is inferred. For definitions, see Tables 3 and 4.

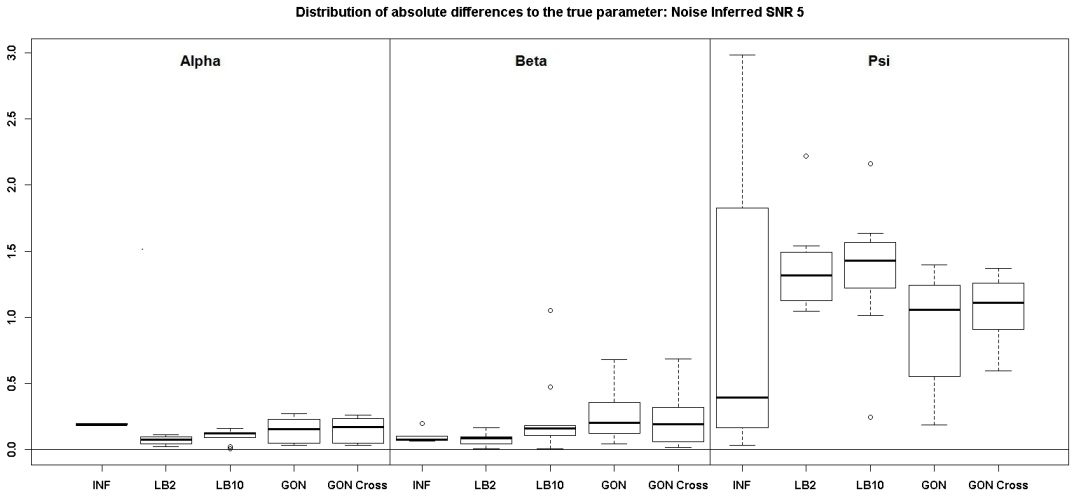


Figure 23: Boxplots of the distributions of the absolute differences of an estimate to the true parameter over 10 datasets. The three sections from left to right represent the parameters α, β and ψ from the Fitz-Hugh Nagumo model (equations 40-41). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, GON method and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 5. The standard deviation of the observational noise is inferred. For definitions, see Tables 3 and 4.

Examining the results for the protein signalling transduction pathway, equation 45, in Figures 24 and 25, it can be seen that the performance of INF, LB2 and LB10 vary in accuracy. Overall, the GON Cross method shows a more robust set of estimates. The GON method (which uses AIC to estimate the penalty parameter) was unable to optimise for this ODE system. Given certain values of λ_s, the optimiser of the log likelihood function tends to choose kernel parameters which make $(\mathbf{K} + \lambda_s \sigma_s I)$ non-invertible and computationally singular. In the cross validation scheme, all problematic λ_ss are rejected. The results for the GON Cross method are presented only, for this

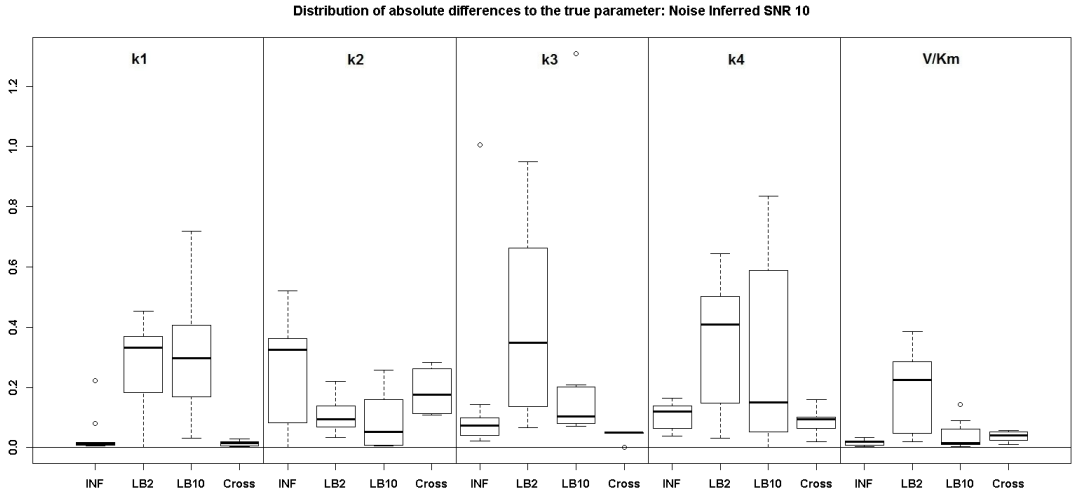


Figure 24: Boxplots of the distributions of the absolute differences of an estimate to the true parameter over 10 datasets. The 5 sections from left to right represent the parameters for the protein signalling transduction pathway, equation 45. Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method and GON method using cross validation for inferring the penalty parameter (abbreviated here to Cross, for visual clarity). The average signal to noise ratio for each "species" is 10. The standard deviation of the observational noise is inferred. For definitions, see Tables 3 and 4.

ODE model.

The root mean square (RMS) values in function space are also presented. Firstly, the signal was reconstructed using the sampled parameters and the initial conditions used to generate the simulated data, by numerically integrating the ODEs, and then the true signal was subtracted (signal created with true parameters and no observational noise added). The RMS was calculated on these residuals. It is important to assess the methods on this

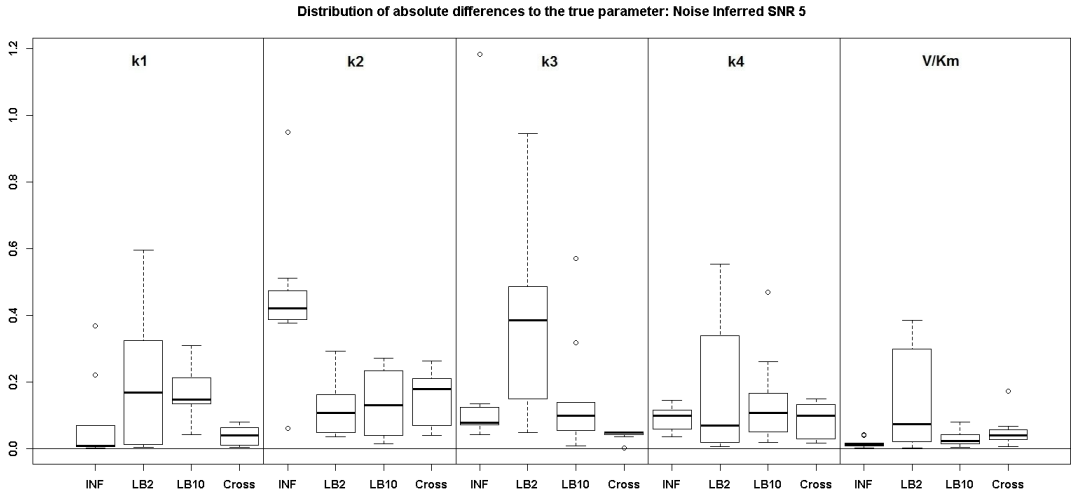


Figure 25: Boxplots of the distributions of the absolute differences of an estimate to the true parameter over 10 datasets. The 5 sections from left to right represent the parameters for the protein signalling transduction pathway, equation 45. Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method and GON method using cross validation for inferring the penalty parameter (abbreviated here to Cross, for visual clarity). The average signal to noise ratio for each "species" is 5. The standard deviation of the observational noise is inferred. For definitions, see Tables 3 and 4.

criterion as well as looking at the parameter uncertainty, as some parameters might only be weakly identifiable, corresponding to ridges in the likelihood landscape. In other words, large uncertainty in parameter estimates may not necessarily imply a poor performance by a method, if the reconstructed signals for all groups of sampled parameters were close to the truth.

By examining Figure 26 it can be seen that the LB2 and LB10 methods perform poorer than the rest, with an average RMS value roughly 0.5 larger.

In Figure 27, the increased noise scenario, it can be seen that the LB2 and LB10 methods have an average RMS value about 0.5 units larger than the other methods.

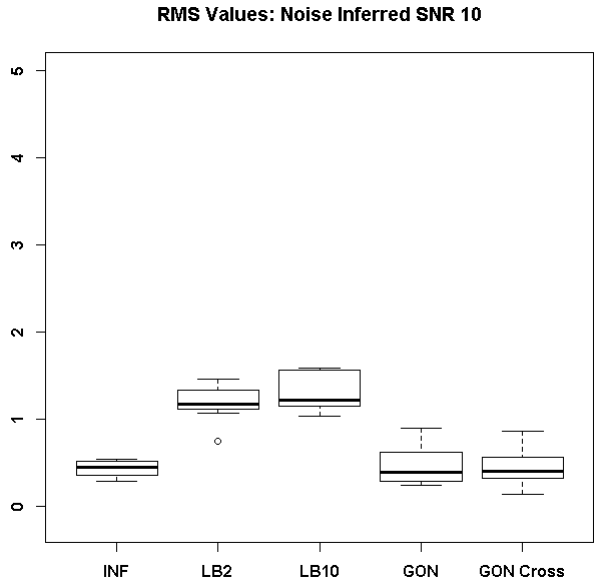


Figure 26: Distribution of RMS values in function space, calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced with the estimate of the parameters, for the Fitz-Hugh Nagumo model (equations 40-41). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, GON method and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 10. The standard deviation of the observational noise is inferred. For definitions, see Tables 3 and 4.

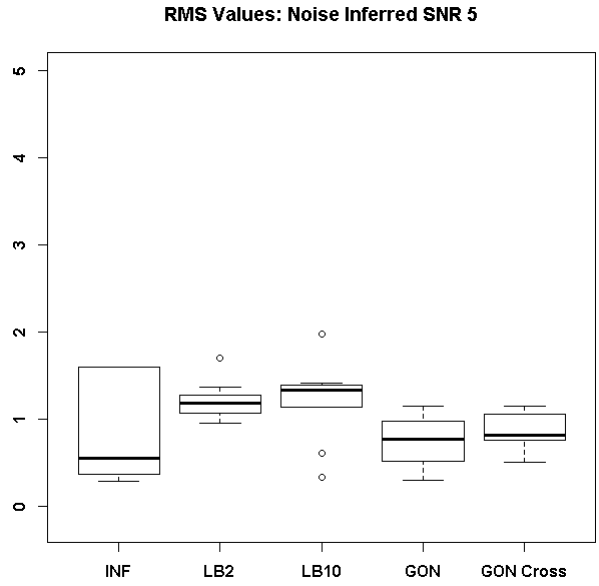


Figure 27: Distribution of RMS values in function space, calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced with the estimate of the parameters, for the Fitz-Hugh Nagumo model (equations 40-41). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, GON method and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 10. The standard deviation of the observational noise is inferred. For definitions, see Tables 3 and 4.

Figures 28-29 show that the GON Cross method is slightly outperforming the INF, LB2 and LB10 methods, with RMS distributions that are on average 0.1 units lower.

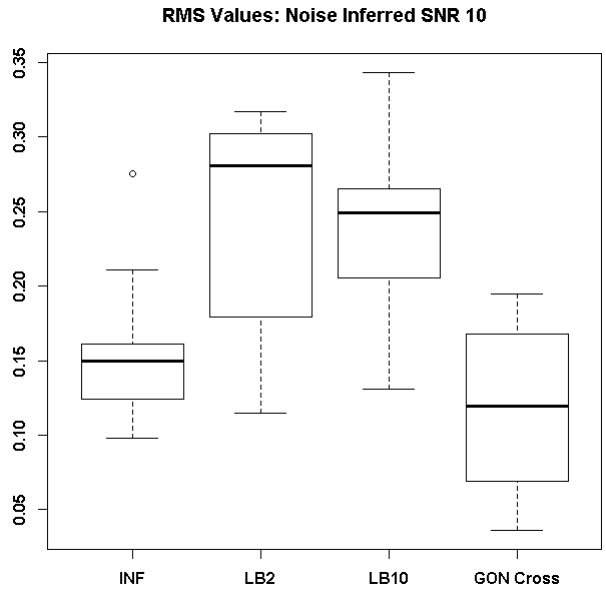


Figure 28: Distribution of RMS values in function space, calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced with the estimate of the parameters, for the protein signalling transduction pathway (equation 45). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 10. The standard deviation of the observational noise is inferred. For definitions, see Tables 3 and 4.

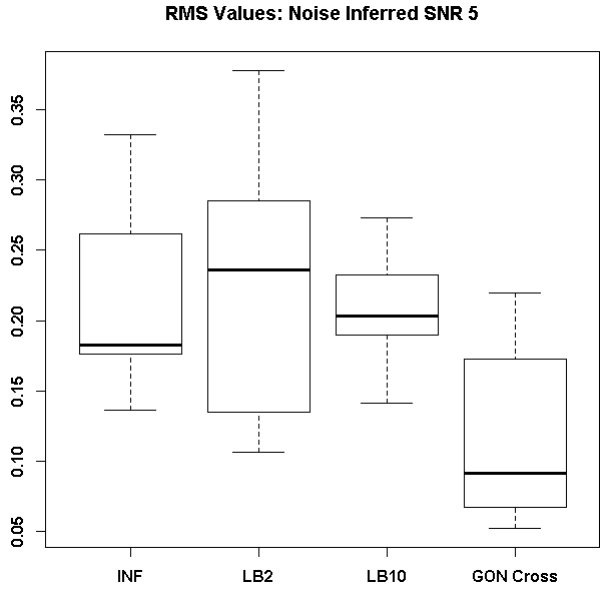


Figure 29: Distribution of RMS values in function space, calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced with the estimate of the parameters, for the protein signalling transduction pathway (equation 45). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 5. The standard deviation of the observational noise is inferred. For definitions, see Tables 3 and 4.

The wider range of estimates of the parameters (as well as the long tails in the posterior distributions in Figures 15-17), for the INF, LB2 and LB10 methods, were observed when occasionally the time course signals would flatten. An inspection of equation 65 reveals that when $f_s(\mathbf{X}, \boldsymbol{\theta}, \mathbf{t}) = \mathbf{0}\ \forall s$, then $p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\eta}, \boldsymbol{\gamma})$ is maximised at $\mathbf{x}_s = \boldsymbol{\phi}_s\ \forall s$ (see equation 53 for the definition of $\boldsymbol{\phi}_s$). This corresponds to a flattening of the true concentration profiles, and flat signals usually can be assumed to be a poor fit to the data. Hence, this flattening should be discouraged by the likelihood term $p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma})$ in equation 52. However, for $\boldsymbol{\sigma} \gg \boldsymbol{\sigma}_{\text{True}}$ (where $\boldsymbol{\sigma}_{\text{True}}$ is the unknown true standard deviation of the observational error of the signals), the likelihood term is effectively switched off, which will allow the system to converge to a high probability attractor state corresponding to $\mathbf{x}_s = \boldsymbol{\phi}_s$. In practice, this effect is observed for $\boldsymbol{\sigma}$ exceeding $\boldsymbol{\sigma}_{\text{True}}$. This attractor state is further self-enforcing by driving the length scales included in the GP hyperparameters $\boldsymbol{\eta}$ to very large values, as has been observed in the simulations. Obviously, $\mathbf{x}_s = \boldsymbol{\phi}_s$ is unrealistic. To test whether holding the standard deviation of the noise at the true value prevents the Markov chains from being driven to this unrealistic attractor state, the simulations of the comparison to GON and GON Cross were repeated, for the Fitz-Hugh Nagumo system and protein signalling transduction pathway for signal to noise ratios of 10 and 5. The standard deviation of the noise was held at the value that was used to generate the data, where in practice this could be estimated through a standard GP regression. The true value was used in order to observe whether this

approach affects the results and to what extent, under the most favourable conditions.

Examining Figure 30, where now the standard deviation of the noise is held fixed at the true value, the INF, LB2, LB10, GON and GON Cross methods perform similarly for the first 2 parameters and the GON and GON Cross are about 5 times the absolute distance closer than the other methods to the true parameter for the 3^{rd}. When the noise is increased, Figure 31, the methods produce estimates that are similar to one another for all 3 parameters.

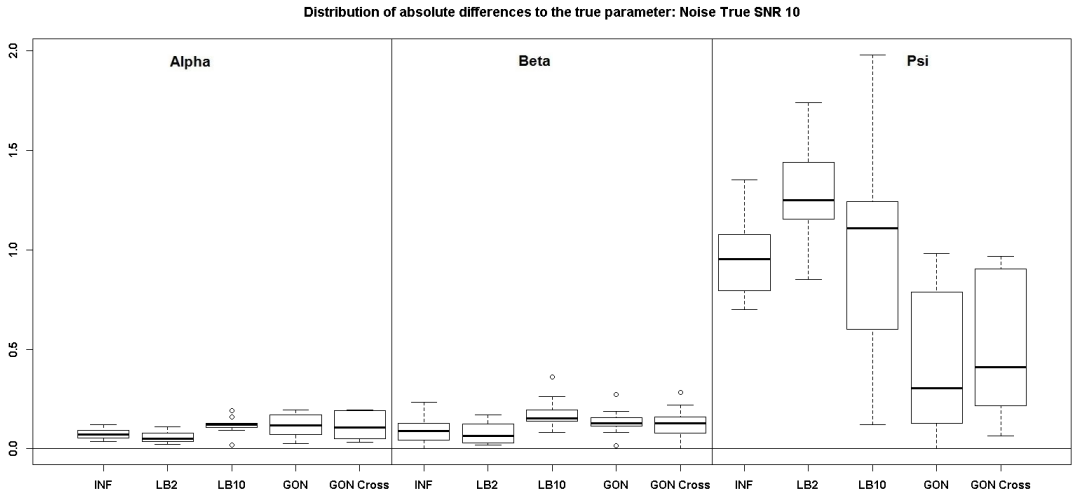


Figure 30: Boxplots of the distributions of the absolute differences of an estimate to the true parameter over 10 datasets. The three sections from left to right represent the parameters α, β and ψ from the Fitz-Hugh Nagumo model (equations 40-41). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, GON method and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 10. The standard deviation of the observational noise is held at the true value. For definitions, see Tables 3 and 4.

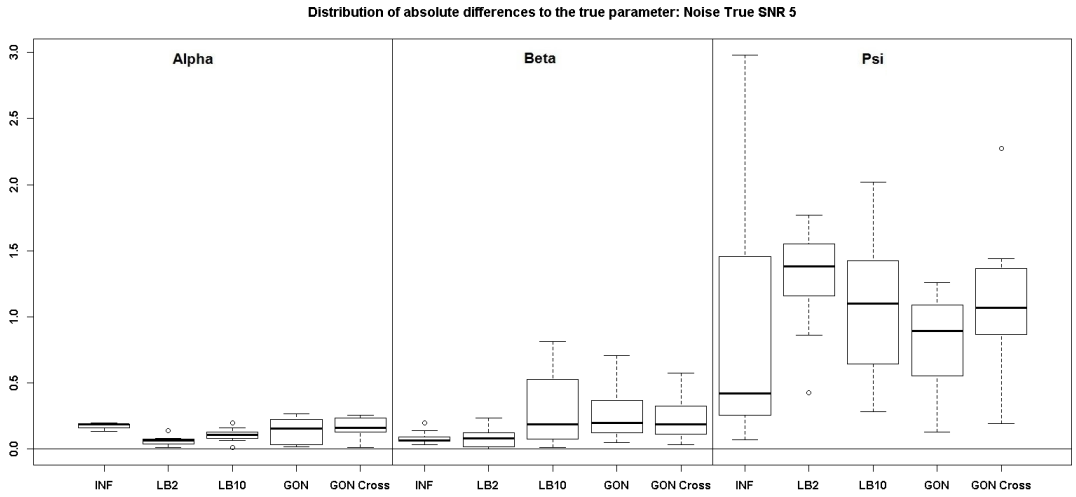


Figure 31: Boxplots of the distributions of the absolute differences of an estimate to the true parameter over 10 datasets. The three sections from left to right represent the parameters α, β and ψ from the Fitz-Hugh Nagumo model (equations 40-41). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, GON method and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 5. The standard deviation of the observational noise is held at the true value. For definitions, see Tables 3 and 4.

For the protein signalling transduction pathway, equation 45, Figure 32 shows that the INF, LB2 and LB10 methods are on average 2.5 times the absolute distance closer than GON CROSS to the true parameter, over the different parameters. Similarly, in Figure 33, INF, LB2 and LB10 perform roughly 2.5 times the absolute distance closer than GON CROSS to the true parameter, over the different parameters.

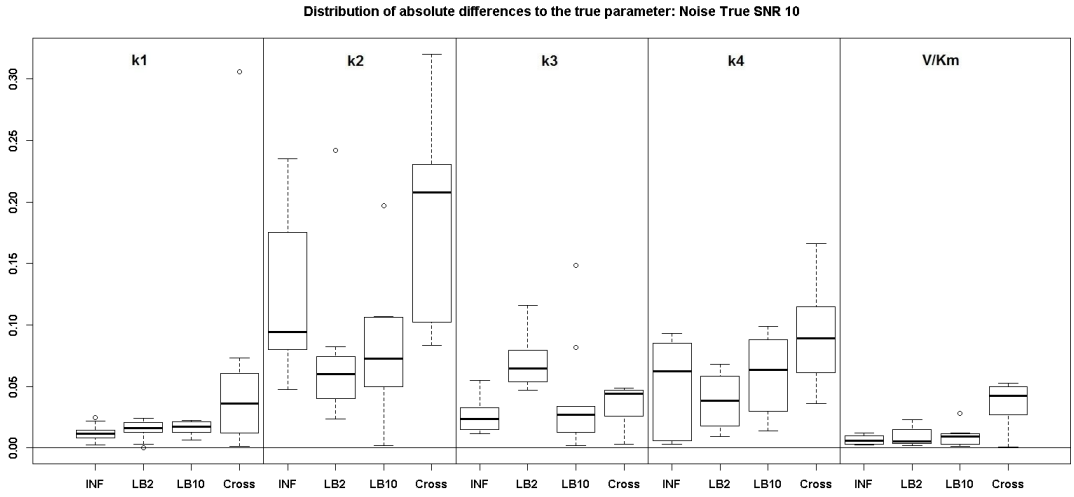


Figure 32: Boxplots of the distributions of the absolute differences of an estimate to the true parameter over 10 datasets. The 5 sections from left to right represent the parameters for the protein signalling transduction pathway, equation 45. Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method and GON method using cross validation for inferring the penalty parameter (abbreviated here to Cross, for visual clarity). The average signal to noise ratio for each "species" is 10. The standard deviation of the observational noise is held at the true value. For definitions, see Tables 3 and 4.

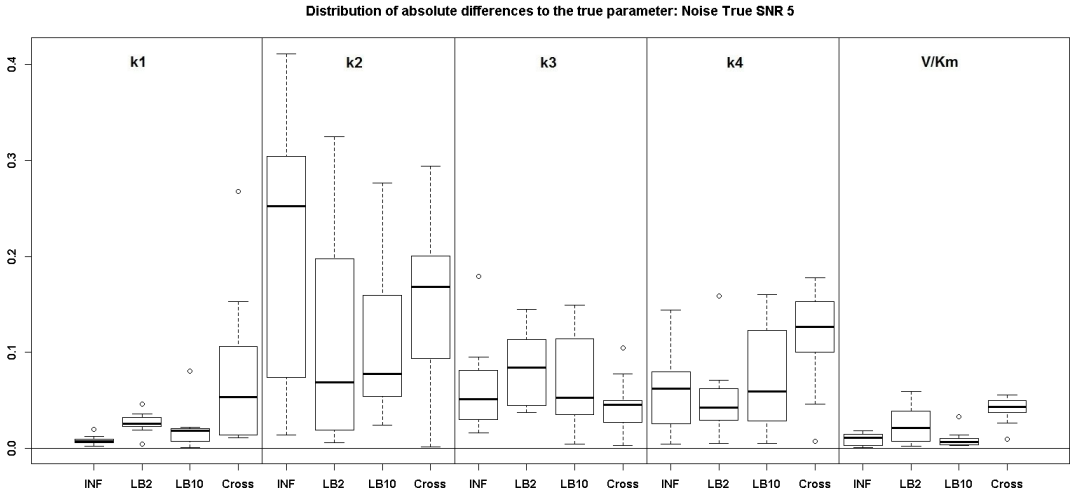


Figure 33: Boxplots of the distributions of the absolute differences of an estimate to the true parameter over 10 datasets. The 5 sections from left to right represent the parameters for the protein signalling transduction pathway, equations equation 45. Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method and GON method using cross validation for inferring the penalty parameter (abbreviated here to Cross, for visual clarity). The average signal to noise ratio for each "species" is 5. The standard deviation of the observational noise is held at the true value. For definitions, see Tables 3 and 4.

The RMS distributions in Figure 34 show that the GON and GON Cross methods are producing slightly better estimates, reflected by the distributions being around 0.5 units in RMS lower. For the increased noise scenario, Figure 35, all methods are performing similarly.

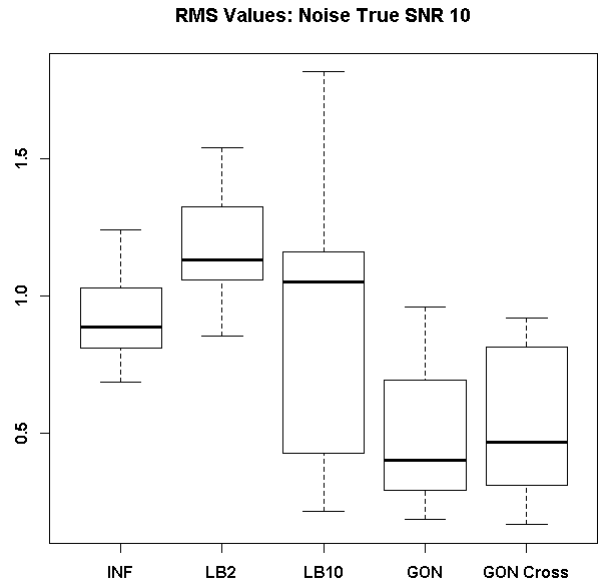


Figure 34: Distribution of RMS values in function space, calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced with the estimate of the parameters, for the Fitz-Hugh Nagumo model (equations 40-41). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, GON method and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 10. The standard deviation of the observational noise is held at the true value. For definitions, see Tables 3 and 4.

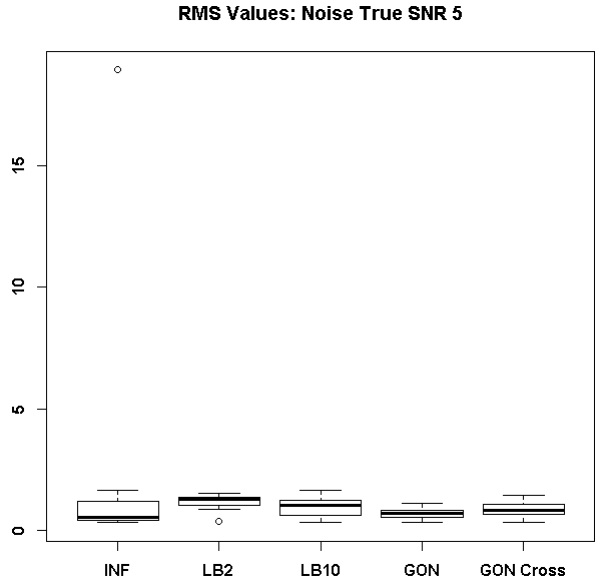


Figure 35: Distribution of RMS values in function space, calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced with the estimate of the parameters, for the Fitz-Hugh Nagumo model (equations 40-41). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, GON method and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 5. The standard deviation of the observational noise is held at the true value. For definitions, see Tables 3 and 4.

In Figure 36, it can be seen that the INF, LB2 and LB10 methods outperform the GON Cross method, shown by smaller RMS distributions that are roughly 0.05 units smaller. In Figure 37, the INF and LB10 methods do better than LB2 and GON Cross with RMS values on average 0.05 units smaller.

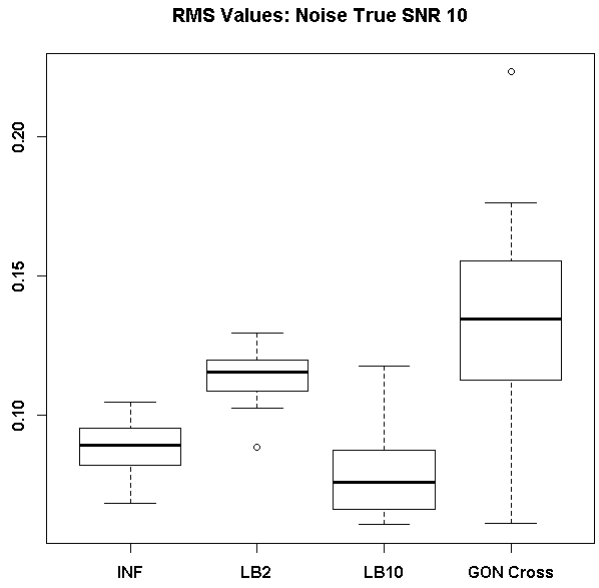


Figure 36: Distribution of RMS values in function space, calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced with the estimate of the parameters, for the protein signalling transduction pathway (equation 45). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 10. The standard deviation of the observational noise is held at the true value. For definitions, see Tables 3 and 4.

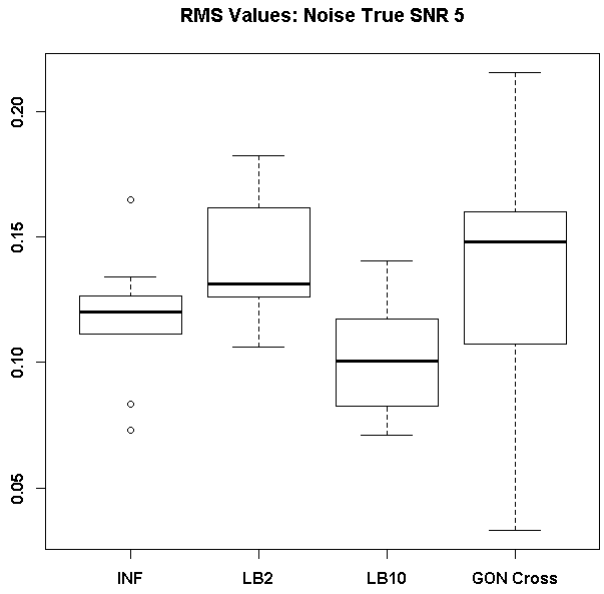


Figure 37: Distribution of RMS values in function space, calculated on the residuals of the true signal (signal produced with true parameters and no observational error) minus the signal produced with the estimate of the parameters, for the protein signalling transduction pathway (equation 45). Within each section, the boxplots from left to right are: LB2 method, INF method, LB10 method, and GON method using cross validation for inferring the penalty parameter. The average signal to noise ratio for each "species" is 5. The standard deviation of the observational noise is held at the true value. For definitions, see Tables 3 and 4.

5.4 Discussion

A recently developed gradient matching approach for systems biology has been modified (INF) by combining it with a parallel tempering scheme for the gradient mismatch parameter (C&S). A wide scale comparative evaluation of this new method from Chapter 4 with various state-of-the-art gradient matching methods has been conducted. These methods are based on different inference approaches and statistical models, namely: non-parametric Bayesian statistics using Gaussian processes (INF, LB2, LB10), splines-based smooth functional tempering (C&S), hierarchical regularisation using splines interpolation (RAM), and penalised likelihood based on reproducing kernel Hilbert spaces (GON, GON Cross). The set-ups have also allowed for the comparison of opposing paradigms of Bayesian inference (INF) versus parallel tempering (LB2, LB10) of the slack parameters controlling the amount of mismatch between the gradients.

In one case, when the number of observations was very high (higher than what would be expected in these types of experiments) and the tuning parameters were finely adjusted (which is time-consuming in practice), the C&S method does well. When the dataset size was reduced, all settings for this method deteriorated significantly, including the previous tuning setting that performed well. It is also important to note that the particular settings that were found to be optimal were different than in the original paper, which

highlights the sensitivity and lack of robustness in the splines based method.

The GON and GON Cross methods produce estimates that are close to the true parameters in terms of absolute uncertainty. For the Fitz-Hugh Nagumo ODE model, the method outperforms the other schemes for one parameter, in the case when the signal to noise ratio was 10 and 25 datapoints were generated. For the protein signalling transduction pathway, however, this method is outperformed by INF, LB2 and LB10. This method also has a drawback to practical implementation, on non-simulated data. The method, which uses a classical approach to parameter estimation (producing point estimates), cannot immediately produce confidence intervals for the parameters and so quantifying the uncertainty in the parameter estimates will be more difficult. For simulated data, this is not an issue, since it is possible to generate multiple datasets and quantify the accuracy of the method by observing the results across all datasets. In practice however, this is not available. One would need to rely on other processes, such as bootstrapping, and the effect on the accuracy and computational time is something that needs to be investigated.

The INF method performs reasonably, by producing results close to the true parameters across the scenarios that have been examined. However, this method's decrease in uncertainty is at the expense of bias.

The LB2 and LB10 methods show good performance across the set-ups. The parameter inference is accurate across the different ODE models and the different settings of those models. The parallel tempering schedule has proven to be quite robust, as the methods perform similarly across the various set-ups.

For some simulations, a flattening of the time course signals for INF, LB2 and LB10 was observed. The uncertainty in the signals reduced the accuracy in the methods. In order to achieve a robust method that provides accurate parameter estimation, we examined holding the standard deviation at the true value. In this case, the GON and GON Cross outperformed INF, LB2 and LB10 on one parameter in the Fitz-Hugh Nagumo system, when the signal to noise ratio was 10. For the signal to noise ratio setting of 5, the methods all performed similarly. The INF, LB2 and LB10 methods outperform the GON Cross method for the protein signalling transduction pathway. Holding the standard deviation of the noise at the true value, for the INF, LB2 and LB10 methods, stops the likelihood term from effectively being switched off and prevents the flattening. In practice, this parameter could be estimated by a standard GP regression, in order to fix the standard deviation of the noise when the true value is unknown. This is a somewhat heuristic fix to the problem however, and a general robust solution should be the focus for

future research.

5.5 Conclusions

The combination of adaptive gradient matching using Gaussian processes from Dondelinger et al. [11] and a parallel tempering scheme for the gradient mismatch parameter from Campbell and Steele [9], has yielded a method that provides accurate parameter estimates for ODEs when the true standard deviation of the noise is known. This method performs well across ODE models and variation of the scheduling of the tempered mismatch parameter.

It has been found that the method in Dondelinger et al. [11] provides accurate estimation, where the decrease in uncertainty is at the expense of bias. The method in Campbell and Steele [9] shows a lack of robustness, due to the difficulty in configuring the splines settings. For the method in Ramsay et al. [40], it was found that it was outperformed by the other methods that were looked at. The method in González et al. [19] is accurate and robust, but can be outperformed by Dondelinger et al. [11] and the proposed method in Chapter 4 for certain scenarios. For a signal to noise ratio of 10 on the Fitz-Hugh Nagumo system, the González et al. method is able to outperform the method in Dondelinger et al. [11] and the new method, for one parameter out of three. It was found that using cross validation as opposed to AIC for the González method, to estimate the penalty parameter, yielded results that were more robust.

In order to avoid a potential drawback to the proposed method in Chapter 4 (and the method in Dondelinger et al. [11]), the standard deviation of the noise is held at the true value, to avoid the signals deviating from the data and flattening. This remedy was found to lead to a significant improvement over the method with a flexible standard deviation of the error. In practice, the standard deviation of the noise could be estimated, for example by a standard GP regression, and general approaches to this should be the focus of future research. It is expected that this should be a fairly robust procedure, as work by Bishop [6] suggests that inferring the noise through standard GP regression is accurate, so long as the GP kernel reasonably reflects the underlying smoothness assumptions of the function being modelled.

6 Representing Gradient Matching as a Probabilistic Generative Model

This chapter presents work published in Macdonald et al. [32]. Software is available at `http://researchdata.gla.ac.uk/283/`.

6.1 Introduction

Many processes in science and engineering can be described by dynamical systems models based on ordinary differential equations (ODEs). Examples range from simple models of predator-prey interactions in ecosystems [27] or activation/deactivation dynamics of spiking neurons [38] to increasingly complex mathematical descriptions of biopathways that aim to predict the time-varying concentrations of different molecular species, like mRNAs and proteins, inside the living cell [39]. ODEs are typically constructed from well understood scientific principles and include clearly interpretable parameters that define the kinetics of the processes and the interactions between the species. However, these parameters are often unknown and not directly measurable. In principle, the task of statistically inferring them from data is not different from statistical inference in more conventional models. For given initial concentrations and under fairly mild regularity conditions, the solution of the ODEs is uniquely defined; hence, the kinetic parameters could be inferred e.g. by minimising the mismatch between the data and the ODE solutions in a maximum likelihood sense. In practice, a closed-form

solution for non-linear ODEs usually does not exist. Any variation of the kinetic parameters thus requires a numerical integration of the ODEs, which is computationally expensive and imposes severe limitations on the number of parameter adaptation steps that are practically feasible.

The present chapter focuses on a particular approach to gradient matching based on nonparametric Bayesian modelling with Gaussian processes (GPs). The key insight, first discussed in Solak et al. [46] and Graepel [21], and more recently exploited in Holsclaw et al. [24], is that for a differentiable kernel, the time derivative of a GP is also a GP. Hence a GP in data space imposes a conjugate GP in derivative space and thereby provides a natural framework for gradient matching. This idea has been exploited in recent high-profile publications, like Babtie et al. [5]. The limitation of Babtie et al. [5] is that the interpolant obtained from the GP is kept fixed, and all subsequent inference critically depends on how accurately this initial interpolant matches the unknown true process. The implication is that the noise tolerance is typically low, as seen e.g. from Fig. 4A in Babtie et al. [5], and that reliable inference requires tight prior constraints on the ODE parameters; see p.2 of the supplementary material in Babtie et al. [5]. To improve the robustness of inference, more advanced methods aim to regularise the GP by the ODEs themselves. Two alternative conceptual approaches to this end have been proposed in the recent machine learning literature. The first paradigm, originally published in Calderhead et al. [8] and more recently extended in Dondelinger et al.

[11], where it was called AGM (for 'adaptive gradient matching'), is based on a product-of-experts approach and a marginalisation over the derivatives of the state variables. A competing approach, proposed in Wang and Barber [49] and called GPODE by the authors, formulates gradient matching with GPs in terms of a probabilistic generative model by marginalising over the state variables and conditioning on the state derivatives. Wang and Barber [49] claim that their proposed paradigm shift achieves an improvement over the first paradigm in three respects: model simplification, tractable inference, and better predictions.

In this chapter, an alternative interpretation of the GPODE model is offered, which leads to deeper insight into intrinsic approximations that were not apparent from the original publication. It is discussed that, as a consequence, the GPODE model suffers from an inherent identifiability problem, which models of the first paradigm are not affected by. The theoretical analysis is complemented with empirical demonstrations on simulated data, using the same model systems as in the original publications, Wang and Barber [49] and Dondelinger et al. [11].

6.2 Paradigm A: the AGM model

The description of paradigm A, the AGM method of Dondelinger et al. [11], has already been detailed in Chapter 4 and hence should be referred to for a description of the method. A graphical representation of the model is given

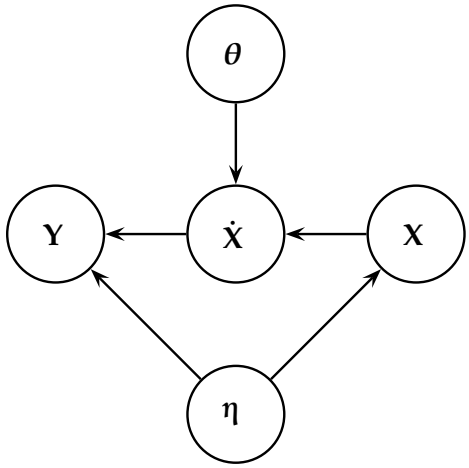


Figure 38: Graphical model of the GPODE method, as proposed in [49].

in Figure 9(b).

6.3 Paradigm B: the GPODE model

An alternative approach was proposed by Wang and Barber [49] and termed the GPODE model. As for AGM, the starting point in Wang and Barber [49] is to exploit the fact that the derivative of a Gaussian process is also a Gaussian process, and that the joint distribution of the state variables $\mathbf{X}$ and their time derivatives $\dot{\mathbf{X}}$ is multivariate Gaussian with covariance functions given by equations (54-57). Application of elementary transformations of Gaussian distributions, as shown e.g. on p. 93 in Bishop [6], leads to the following conditional distribution of the states given the state derivatives:

$$p(\mathbf{x}_s|\dot{\mathbf{x}}_s, \boldsymbol{\eta}) \;=\; \mathcal{N}(\mathbf{x}_s|\tilde{\boldsymbol{\mu}}_s, \tilde{\mathbf{A}}_s) \tag{73}$$

where

$$\tilde{\boldsymbol{\mu}}_s = \boldsymbol{\phi}_s + {}'\mathbf{K}_{\eta_s}{\mathbf{K}''_{\eta_s}}^{-1}\dot{\mathbf{x}}_s; \quad \tilde{\mathbf{A}}_s = \mathbf{K}_{\eta_s} - {}'\mathbf{K}_{\eta_s}{\mathbf{K}''_{\eta_s}}^{-1}\mathbf{K}'_{\eta_s} \tag{74}$$

Note the difference between AGM and GPODE, where the former method computes $p(\dot{\mathbf{x}}_s|\mathbf{x}_s, \boldsymbol{\eta})$, as expressed in equations (58-59), whereas the latter model computes $p(\mathbf{x}_s|\dot{\mathbf{x}}_s, \boldsymbol{\eta})$, as expressed in equations (73-74). Under the assumption that the observations $\mathbf{Y}$ are subject to additive iid Gaussian noise, equations (49,52), the marginalisation over the state variables leads to a standard Gaussian convolution integral, which is analytically tractable with solution

$$\begin{aligned} p^{\diamond}(\mathbf{y}_s|\dot{\mathbf{x}}_s) &= \int p(\mathbf{y}_s|\mathbf{x}_s)p(\mathbf{x}_s|\dot{\mathbf{x}}_s)d\mathbf{x}_s \\ &= \int \mathcal{N}(\mathbf{y}_s|\mathbf{x}_s, \sigma_s^2\mathbf{I})\mathcal{N}(\mathbf{x}_s|\tilde{\boldsymbol{\mu}}_s, \tilde{\mathbf{A}}_s)d\mathbf{x}_s \\ &= \mathcal{N}(\mathbf{y}_s|\tilde{\boldsymbol{\mu}}_s, \tilde{\mathbf{A}}_s + \sigma_s^2\mathbf{I}) \end{aligned} \tag{75}$$

The authors factorise

$$p(\mathbf{Y}, \mathbf{X}|\boldsymbol{\eta}, \boldsymbol{\theta}) \;=\; p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\eta}, \boldsymbol{\theta}) p(\mathbf{X}|\boldsymbol{\eta}) \tag{76}$$

and obtain the first term by marginalisation over the state derivatives $\dot{\mathbf{X}}$:

$$\begin{aligned} p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\eta}, \boldsymbol{\theta}) &= \int p(\mathbf{Y}, \dot{\mathbf{X}}|\mathbf{X}, \boldsymbol{\eta}, \boldsymbol{\theta}) d\dot{\mathbf{X}} \\ &= \int p^{\diamond}(\mathbf{Y}|\dot{\mathbf{X}}, \boldsymbol{\eta}) p(\dot{\mathbf{X}}|\mathbf{X}, \boldsymbol{\theta}) d\dot{\mathbf{X}} \\ &= p^{\diamond}(\mathbf{Y}|f[\mathbf{X}, \boldsymbol{\theta}], \boldsymbol{\eta}) \end{aligned} \tag{77}$$

where

$$p^{\diamond}(\mathbf{Y}|\dot{\mathbf{X}}, \boldsymbol{\eta}) = \prod_s p^{\diamond}(\mathbf{y}_s|\dot{\mathbf{x}}_s, \boldsymbol{\eta}), \tag{78}$$

with $p^{\diamond}(\mathbf{y}_s|\dot{\mathbf{x}}_s, \boldsymbol{\eta})$ given in equation (75), and the fact has been used that the state derivatives are determined by the ODEs:

$$p(\dot{\mathbf{X}}|\mathbf{X}, \boldsymbol{\theta}) \;=\; \delta(\dot{\mathbf{X}} - f[\mathbf{X}, \boldsymbol{\theta}]) \tag{79}$$

Inserting equation (77) into equation (76) gives:

$$p(\mathbf{Y}, \mathbf{X}|\boldsymbol{\eta}, \boldsymbol{\theta}) \;=\; p^{\diamond}(\mathbf{Y}|f[\mathbf{X}, \boldsymbol{\theta}], \boldsymbol{\eta}) p(\mathbf{X}|\boldsymbol{\eta}) \tag{80}$$

This is a deceptionally simple and elegant formulation, illustrated as a graphical model in Figure 38, with two advantages over the AGM model. Conceptually, the GPODE is a proper probabilistic generative model, which can be consistently represented by a directed acyclic graph (DAG). Practically, the normalisation constant of the joint distribution in equation (80) is known, which facilitates inference.

6.4 Shortcomings of the GPODE model

The Achilles heel of the GPODE model is equation (75), which includes a marginalisation over the state variables $\mathbf{x}_s$ to obtain $p(\mathbf{y}_s|\dot{\mathbf{x}}_s)$.

The derivations in equation (76) and equation (77) then treat $\mathbf{y}_s$ as independent of $\mathbf{x}_s$ given $\dot{\mathbf{x}}_s$: $p(\mathbf{y}_s|\dot{\mathbf{x}}_s, \mathbf{x}_s) = p(\mathbf{y}_s|\dot{\mathbf{x}}_s)$, or $p(\mathbf{Y}|\mathbf{X}, \dot{\mathbf{X}}) = p(\mathbf{Y}|\dot{\mathbf{X}})$; this is consistent with the graphical model in Figure 38. Having integrated the state variables $\mathbf{X}$ out in equation (75), the method subsequently conditions on them in equation (77). The fallacy of this approach is the assumption that the marginalisation over the random variables $\mathbf{x}_s$ in equation (75) is equivalent to their elimination. However, a marginalisation merely means that for the purposes of inference, the variables that have been integrated out do not need to be taken into consideration explicitly. However, these variables remain in the model conceptually. In this particular model, the data $\mathbf{Y}$ consist of noisy observations of the state variables $\mathbf{X}$, not their derivatives $\dot{\mathbf{X}}$. Consider, for instance, the tracking of a set of exoplanets with a space

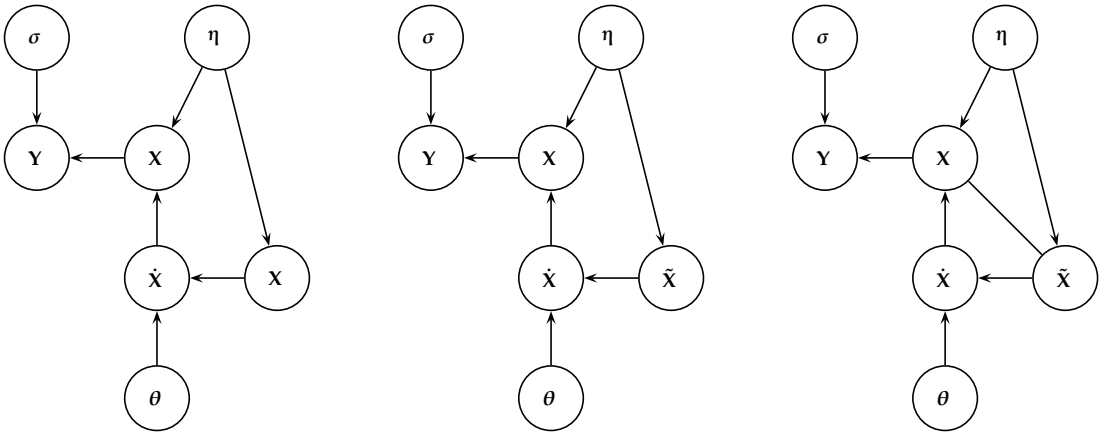


Figure 39: *Left panel:* GPODE model, as proposed in Wang and Barber [49], but explicitly presenting all random variables included in the model. The graph is inconsistent, in that the same random variables, $\mathbf{X}$, have been assigned to two different nodes. *Centre panel:* Correcting the inconsistency in the notation of Wang and Barber [49]. The model distinguishes between the unknown true state variables $\mathbf{X}$, and their model approximation $\tilde{\mathbf{X}}$. *Right panel:* In the ideal GPODE model, the true state variables $\mathbf{X}$ and their model approximation $\tilde{\mathbf{X}}$ are coupled, ideally via an identity constraint. This introduces an undirected edge between $\mathbf{X}$ and $\tilde{\mathbf{X}}$, which is no longer a consistent probabilistic graphical model represented by a DAG. To reintroduce the DAG constraint, Wang and Barber [49] have discarded this undirected edge, leading to the model shown in the centre panel. The disadvantage is that the model state variables $\tilde{\mathbf{X}}$ are no longer directly associated with the data. As discussed in the main text, this leads to an intrinsic identifiability problem.

telescope, where the state variables $\mathbf{X}$ are the positions of the planets. Given the knowledge of the initial conditions and the velocities of the planets, $\dot{\mathbf{X}}$, it is possible to compute the positions of the planets $\mathbf{X}$ using established equations from classical mechanics. This procedure might dispense with the need to keep detailed records of the planets' positions. However, it does *not* imply that the positions of the planets have disappeared.

To correct this mistake, it is necessary to reintroduce the state variables $\mathbf{X}$ into the model, as shown in Figure 39, left panel. However, this leads to the inconsistency that the same random variables, $\mathbf{X}$, are used in two different places of the graph. As a further correction, it is therefore required to introduce a set of dummy variables $\tilde{\mathbf{X}}$, as shown in Figure 39, centre panel. This is a methodologically consistent representation of the model, but leaves open the question what the difference between $\mathbf{X}$ and $\tilde{\mathbf{X}}$ is. Ideally, there is no difference, which can be represented mathematically as $p(\mathbf{X}, \tilde{\mathbf{X}}) \propto \delta(\mathbf{X} - \tilde{\mathbf{X}})$. However, in this way an edge from the node $\tilde{\mathbf{X}}$ to $\mathbf{X}$ has been introduced, as shown in Figure 39, right panel. This causes methodological problems, in whatever definition chosen for that edge. If it is treated as an undirected edge, $p(\mathbf{X}, \tilde{\mathbf{X}}) \propto \delta(\mathbf{X} - \tilde{\mathbf{X}})$, as shown in the right panel of Figure 39, based on the symmetry of the identity relation between $\tilde{\mathbf{X}}$ and $\mathbf{X}$, then a chain graph is produced. A chain graph is a probabilistic model, but not a probabilistic generative model, and the main objective of Wang and Barber [49] was to create one. If a directed edge from $\mathbf{X}$ to $\tilde{\mathbf{X}}$ is introduced, based on

$$p(\tilde{\mathbf{X}}|\mathbf{X}) = \delta(\tilde{\mathbf{X}} - \mathbf{X}), \tag{81}$$

then a directed cycle exists and this violates the DAG constraint. In order to get a valid probabilistic graphical model, a directed edge in the opposite direction must be introduced, from $\tilde{\mathbf{X}}$ to $\mathbf{X}$, based on

$$p(\mathbf{X}|\tilde{\mathbf{X}}) = \delta(\tilde{\mathbf{X}} - \mathbf{X}). \tag{82}$$

However, this structure will require the definition of the probability $p(\mathbf{X}|\dot{\mathbf{X}}, \tilde{\mathbf{X}})$, and it is not clear how to do that. For that reason, the approximation taken in Wang and Barber [49] is to discard the edge between $\mathbf{X}$ and $\tilde{\mathbf{X}}$ altogether. This simplification leads to a probabilistic generative model that can be consistently represented by a DAG. However, the disadvantage is that the true state variables $\mathbf{X}$ and their approximation $\tilde{\mathbf{X}}$ are only weakly coupled, via their common hyperparameters $\boldsymbol{\eta}$. The consequences of this will be discussed further on.

The upshot of what has been explained so far is that, by not properly distinguishing between $\mathbf{X}$ and $\tilde{\mathbf{X}}$, equation (80) introduced in Wang and Barber [49] is misleading. The correct form is

$$p(\mathbf{Y}, \tilde{\mathbf{X}}|\boldsymbol{\eta}, \boldsymbol{\theta}) \;=\; p^{\diamond}(\mathbf{Y}|f[\tilde{\mathbf{X}}, \boldsymbol{\theta}], \boldsymbol{\eta}) p(\tilde{\mathbf{X}}|\boldsymbol{\eta}) \tag{83}$$

where $\tilde{\mathbf{X}}$ are *not* the unknown true state variables $\mathbf{X}$, but some model ap-

proximation. This subtle difference has non-negligible consequences.

As an illustration, consider the simple second-order ODE (using $\ddot{x} = d^2x/dt^2$)

$$\ddot{x} + \theta^2 x \;=\; 0 \tag{84}$$

which, with the standard substitution $(x_1, x_2) := (x, \dot{x})$, leads to the linear system of first-order ODEs:

$$\dot{x}_1 \;=\; x_2; \qquad \dot{x}_2 \;=\; -\theta^2 x_1 \tag{85}$$

These ODEs have the closed-form solution:

$$x_1(t) \;=\; A\sin(\theta t + \phi); \quad x_2(t) \;=\; A\theta\cos(\theta t + \phi) \tag{86}$$

where A and ϕ are constants, which are determined by the initial conditions. Now, according to the GPODE paradigm, illustrated in the centre panel of Figure 39, x_1 and x_2 in equation (85) have to be replaced by separate variables:

$$\dot{x}_1(t) \;=\; \tilde{x}_2(t); \qquad \dot{x}_2(t) \;=\; -\theta^2 \tilde{x}_1(t) \tag{87}$$

where $\tilde{x}_1(t)$ and $\tilde{x}_2(t)$ are modelled with a GP.

Recalling that $\mathbf{x}_s = [x_s(t_1), \ldots, x_s(t_N)]^\mathsf{T}$, equation (87) is rewritten as:

$$\dot{\tilde{\mathbf{x}}}_1 = \mathbf{f}_1(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2; \theta) = \tilde{\mathbf{x}}_2; \;\; \dot{\tilde{\mathbf{x}}}_2 = \mathbf{f}_2(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2; \theta) = -\theta^2 \tilde{\mathbf{x}}_1 \tag{88}$$

Inserting these expressions into equation (83), yields:

$$\begin{aligned} & p(\mathbf{y}_1, \mathbf{y}_2, \tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2 | \boldsymbol{\eta}, \theta) = \\ & p^\diamond(\mathbf{y}_1, \mathbf{y}_2 | f_1[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \theta], f_2[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \theta], \boldsymbol{\eta}) p(\tilde{\mathbf{x}}_1 | \boldsymbol{\eta}) p(\tilde{\mathbf{x}}_2 | \boldsymbol{\eta}) = \\ & p^\diamond(\mathbf{y}_1 | f_1[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \theta], \boldsymbol{\eta}) p^\diamond(\mathbf{y}_2 | f_2[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \theta], \boldsymbol{\eta}) p(\tilde{\mathbf{x}}_1 | \boldsymbol{\eta}) \\ & p(\tilde{\mathbf{x}}_2 | \boldsymbol{\eta}) = p^\diamond(\mathbf{y}_1 | \tilde{\mathbf{x}}_2, \boldsymbol{\eta}) p^\diamond(\mathbf{y}_2 | - \theta^2 \tilde{\mathbf{x}}_1, \boldsymbol{\eta}) p(\tilde{\mathbf{x}}_1 | \boldsymbol{\eta}) p(\tilde{\mathbf{x}}_2 | \boldsymbol{\eta}) \end{aligned} \tag{89}$$

The superscript in $p^\diamond$ is used to indicate that the functional form of this probability distribution is given by equation (75). Now recall that the variable x_2 represents the time derivative of x_1 and was introduced as an auxiliary variable to transform the second-order ODE from equation (84) into a system of first-order ODEs: equation (85). In most applications, only the variables themselves rather than their derivatives can be measured or observed, i.e. y_2 is systematically missing. From equation (89), missing variables y_2 gives:

$$
\begin{aligned}
p(\mathbf{y}_1, \tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2|\boldsymbol{\eta}, \theta) &= \int p(\mathbf{y}_1, \mathbf{y}_2, \tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2|\boldsymbol{\eta}, \theta) d\mathbf{y}_2 \\
&= p^{\diamond}(\mathbf{y}_1|\tilde{\mathbf{x}}_2, \boldsymbol{\eta}) p(\tilde{\mathbf{x}}_1|\boldsymbol{\eta}) p(\tilde{\mathbf{x}}_2|\boldsymbol{\eta}) \\
&\quad \int p^{\diamond}(\mathbf{y}_2| - \theta^2 \tilde{\mathbf{x}}_1, \boldsymbol{\eta}) d\mathbf{y}_2 \\
&= p^{\diamond}(\mathbf{y}_1|\tilde{\mathbf{x}}_2, \boldsymbol{\eta}) p(\tilde{\mathbf{x}}_1|\boldsymbol{\eta}) p(\tilde{\mathbf{x}}_2|\boldsymbol{\eta}) \qquad (90)
\end{aligned}
$$

and

$$
\begin{aligned}
p(\mathbf{y}_1|\boldsymbol{\eta}, \theta) &= \int p(\mathbf{y}_1, \tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2|\boldsymbol{\eta}, \theta) d\tilde{\mathbf{x}}_1 d\tilde{\mathbf{x}}_2 \qquad (91) \\
&= \int p^{\diamond}(\mathbf{y}_1|\tilde{\mathbf{x}}_2, \boldsymbol{\eta}) p(\tilde{\mathbf{x}}_2|\boldsymbol{\eta}) d\tilde{\mathbf{x}}_2 \int p(\tilde{\mathbf{x}}_1|\boldsymbol{\eta}) d\tilde{\mathbf{x}}_1 \\
&= \int p^{\diamond}(\mathbf{y}_1|\tilde{\mathbf{x}}_2, \boldsymbol{\eta}) p(\tilde{\mathbf{x}}_2|\boldsymbol{\eta}) d\tilde{\mathbf{x}}_2 = p(\mathbf{y}_1|\boldsymbol{\eta})
\end{aligned}
$$

This implies that the likelihood, i.e. the probability of a set of observations $\mathbf{y}_1 = [y_1(t_1), \ldots, y_1(t_N)]^{\mathsf{T}}$, is independent of the ODE parameter θ. Consequently, in the GPODE model, the parameter of interest – the ODE parameter θ – is unidentifiable, i.e. it can *not* be inferred from the data. Note that this problem is intrinsic to the GPODE model, *not* the ODE it-

self. Equation (84) is a very simple ODE with a closed form solution for $x(t) = x_1(t)$, stated in equation (86). If this solution is known, the inference task reduces to inferring the frequency from noisy observations of a sine function. Hence, it is straightforward to infer θ from noisy observations $y_1(t) = x_1(t) + \varepsilon(t)$ alone, where $\varepsilon(t)$ is iid noise, and no observations of the derivative $x_2 = \frac{dx}{dt}$ are required. Even if the explicit solution were not known, it could be obtained by numerical integration of the ODEs, again rendering the inference of the ODE parameter θ a straightforward task. How do missing observations affect the AGM model? When y_2 is systematically missing, it is necessary to marginalise over $\mathbf{y}_2$ in equation (64). This will only affect the first term on the right-hand side of equation (64), which as a consequence of the marginalisation will reduce from $p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}) = p(\mathbf{y}_1, \mathbf{y}_2|\mathbf{X}, \boldsymbol{\sigma})$ to $p(\mathbf{y}_1|\mathbf{X}, \boldsymbol{\sigma})$. However, this term does not explicitly depend on the ODE parameters $\boldsymbol{\theta}$. Hence, as opposed to the GPODE model, missing observations do not systematically eliminate ODE parameters from the likelihood. In fact, an inspection of equation (85) provides an intuitive explanation of how inference in the AGM can work despite systematically missing values: noisy observations of x_1 provide information about the missing species x_2 via equation (85), left, using the very principle of gradient matching. Inference of x_2 then enables inference of the ODE parameter θ via equation (85), right. It will be demonstrated, in Chapter 6.5, that AGM indeed can successfully infer the ODE parameter θ when observations for species y_2 are missing, whereas GPODE systematically fails on this task.

6.5 Empirical findings

The empirical analysis presented in Wang and Barber [49] suggests that the GPODE model achieves very accurate parameter estimates. However, a closer inspection of the authors' study reveals that they used very informative priors with tight uncertainty intervals centred on the (known) true parameter values. In the present study, Wang and Barber's [49] simulations have been repeated, but with less informative priors, using their own software. The inference for the AGM model has also been integrated into their software, for a fair comparison between the two paradigms.

Computational inference. The objective of inference is to obtain the marginal posterior distributions of the quantities of interest, which are usually the ODE parameters. This is analytically intractable, and previous authors have used sampling methods based on MCMC. Dondelinger et al. [11] and Calderhead et al. [8] used MCMC schemes for continuous values, based on Metropolis-Hastings with appropriate proposal moves. Wang and Barber [49] used Gibbs sampling as a faster alternative, based on a discretisation of the latent variables, parameters and hyperparameters. For a fair comparison between the model paradigms (AGM versus GPODE), which is not confounded by the different convergence characteristics and potential discretisation artefacts of the two MCMC schemes (Metropolis-Hastings versus Gibbs sampling), the AGM model has been implemented in the software of Wang and Barber [49] to infer all quantities of interest with the same Gibbs

sampling scheme. The basic idea is that due to the discretisation, all quantities can be marginalised over in the joint probability density, and this allows the conditional probabilities needed for the Gibbs sampler to be easily computed. For the prior distribution over the latent variables, the software of Wang and Barber [49] fits a standard GP to the data and chooses, for each timepoint, a uniform distribution with a 3-standard-deviation width centred on the GP interpolant. For faster convergence of the MCMC simulations, the noise variance σ_s^2 was set equal to the true noise variance, and the mean $\boldsymbol{\phi}_s$ equal to the sample mean. The parameters that had to be inferred (in addition to the latent state variables) were the ODE parameters, the kernel parameters of the GP, and the slack parameter γ for the AGM. For all simulations, a squared exponential kernel was used, and a $U(5, 50)$ prior for the length scale and a $U(0.1, 1)$ prior for the amplitude hyperparameters were chosen, respectively, as in the paper by Wang and Barber [49]. Different prior distributions of the ODE parameters were tried, as specified in the figure captions; note that these priors are less informative than those used in Wang and Barber [49]. Observational noise was added in the same way as in Wang and Barber [49]. All simulations were repeated on ten independent data instantiations.

Simple ODE with missing values. As a first study, noisy data was generated from the simple ODEs of (85), with species 2 missing, using a sample size of $N = 20$ and an average signal-to-noise ratio of $SNR = 10$. The

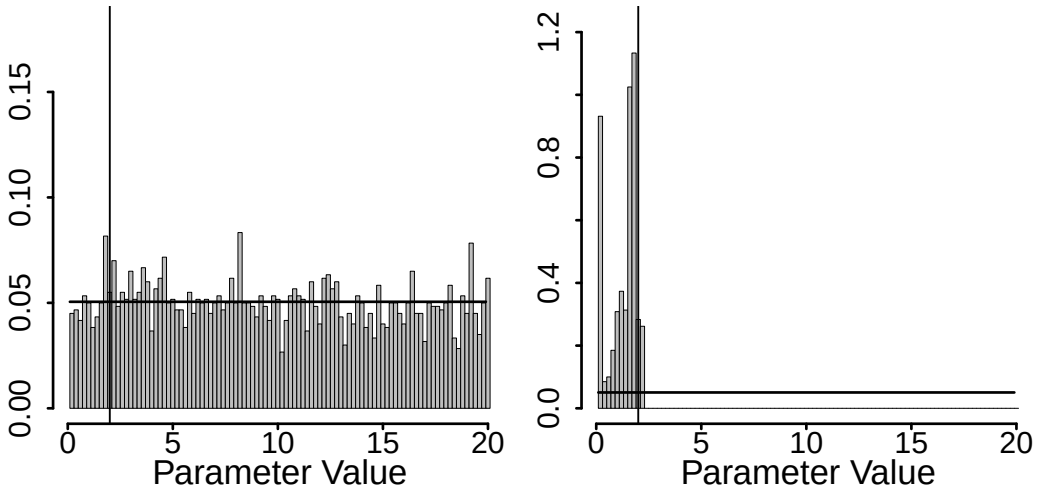


Figure 40: Inference results for the ODEs (85) with missing species. Vertical line: true parameter value. Horizontal line: uniform prior. Histogram: average posterior distribution obtained with Gibbs sampling, averaged over ten independent data instantiations. Left panel: GPODE model. Right panel: AGM model.

results are shown in Figure 40. They confirm what was discussed below equation (91): paradigm B completely fails to infer the ODE parameter; in fact, the inferred posterior distribution is indistinguishable from the prior. Paradigm A succeeds in inferring the ODE parameter: the posterior distribution is noticeably different from the prior and the 95% credible interval includes the true parameter.

The Lotka-Volterra system. This is a simple model for prey-predator interactions in ecology [27], and autocatalysis in chemical kinetics [4], see equation 42. This model was used for the evaluation of parameter inference in Dondelinger et al. [11] and Wang and Barber [49], and the simulations here were repeated with the same parameters as used in these studies. First, $N = 11$ datapoints were generated with $\theta_1 = 2, \theta_2 = 1, \theta_3 = 4, \theta_4 = 1$. Next, iid Gaussian noise with an average signal-to-noise ratio $SNR = 4$ was added, and ten independent datasets were generated this way. The results are shown in Figure 41. The AGM model (paradigm A) shows a consistent performance over both parameter priors: the Gamma $\Gamma(4, 0.5)$ prior and the uniform prior. In both cases, the inferred posterior distributions are tightly concentrated on the true parameters. The GPODE model (paradigm B) sensitively depends on the prior. The inferred posterior distributions are always more diffuse than those obtained with paradigm A, and the performance is particularly poor for the uniform prior. Here, paradigm A clearly outperforms paradigm B.

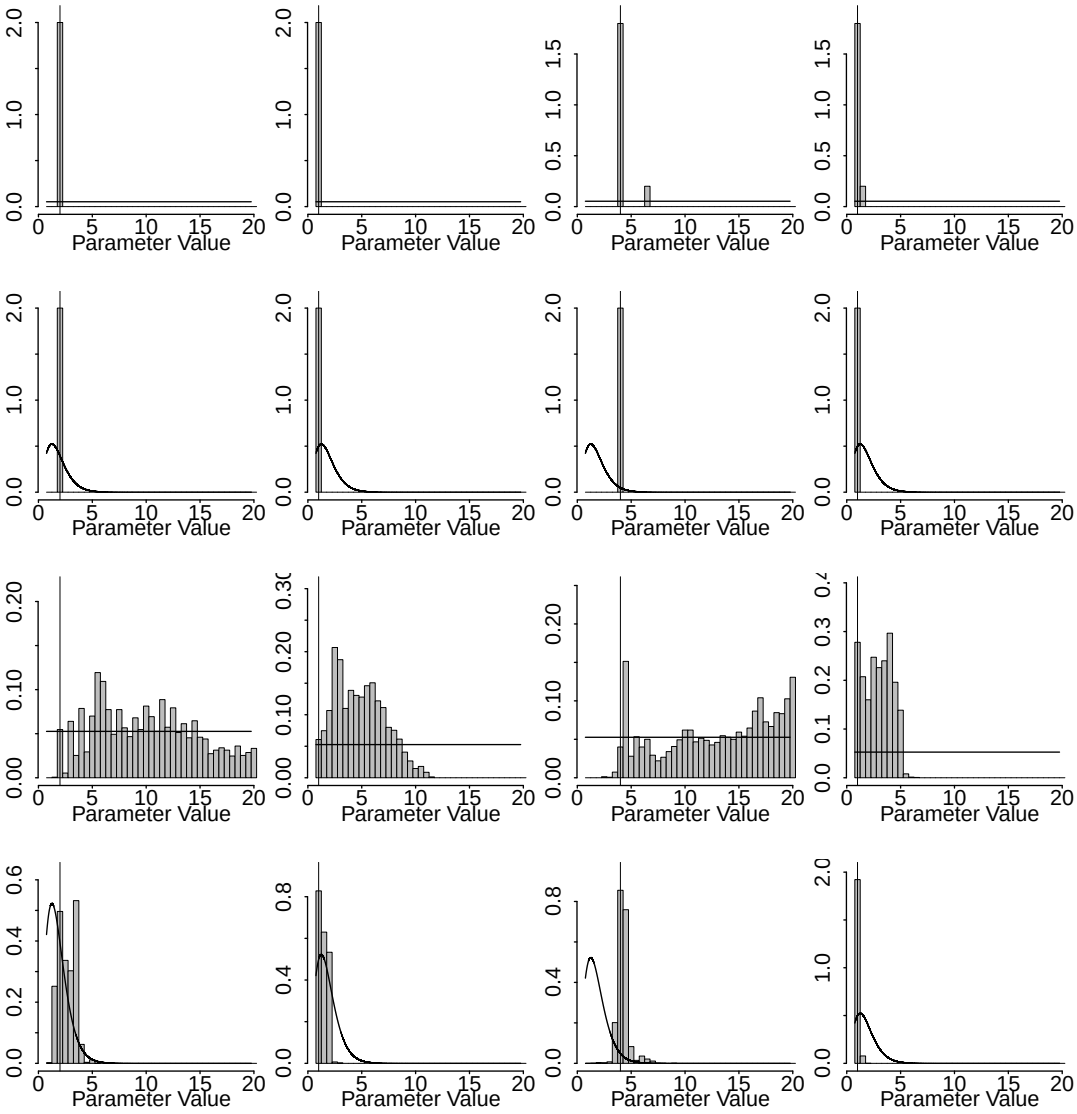


Figure 41: Inference results for the Lotka-Volterra system, equation (42). Each column represents one of the four kinetic parameters of the system, and the histograms show the average posterior distributions of the respective parameter, averaged over ten data instantiations. Vertical line: true parameter value. Black line: prior distribution - uniform or $\Gamma(4, 0.5)$. The top two rows show the results for the AGM model (paradigm A). The bottom two rows show the results for the GPODE model (paradigm B).

The Fitz-Hugh Nagumo system (equations 40-41) was introduced in Fitz-Hugh [14] and Nagumo et al. [38] to model the voltage potential across the cell membrane of the axon of giant squid neurons. The model was used in Campbell and Steele [9] to assess parameter inference in ODEs, using comparatively large sets of $N = 401$ observations. For the present study, data was generated with the same parameters, $\alpha = 0.2$, $\beta = 0.2$ and $\psi = 3$, and same initial values, $V = 1, R = -1$, but making the inference problem harder by reducing the training set size to $N = 20$, covering the time interval $[0, 10]$. Noisy measurements were emulated by adding iid Gaussian noise with an average signal-to-noise ratio $SNR = 10$, and generated ten independent data instantiations. The results are shown in Figure 42. Here, both paradigms show a similar performance. The GPODE model is slightly better than the AGM model in terms of reduced bias for the third parameter, but slightly worse in terms of increased posterior variance for the first parameter. The results are, overall, worse than for the Lotka-Volterra system. Note that the Fitz-Hugh Nagumo system poses a challenging problem, though; see Campbell and Steele [9] and recall that the dataset is considerably smaller (5%) than the one used by the authors.

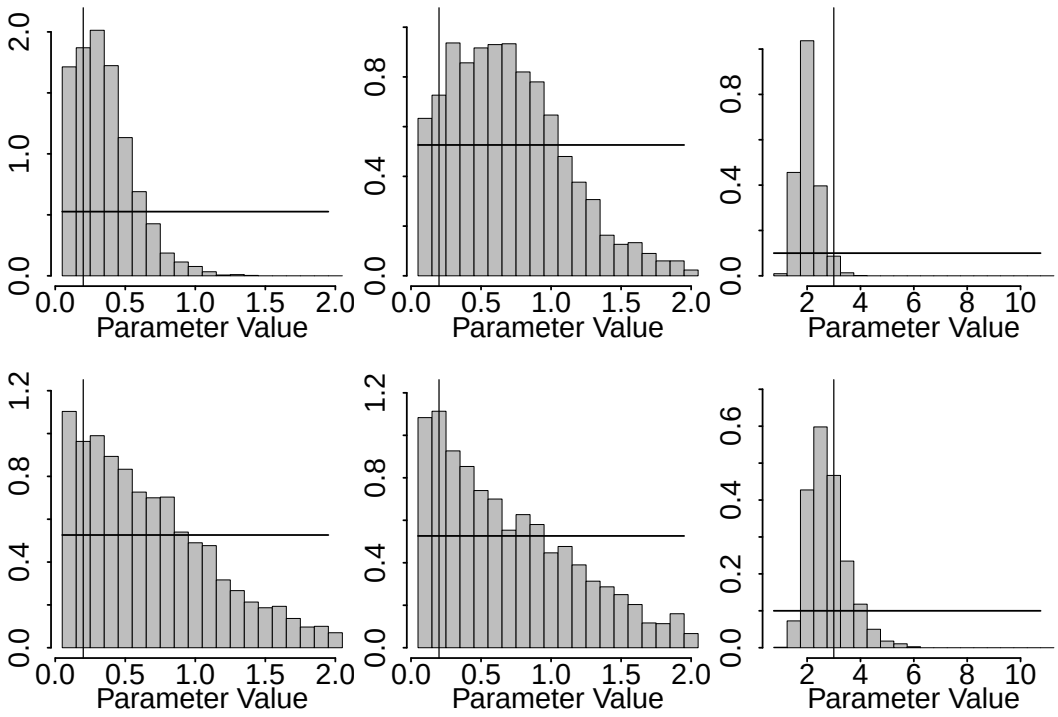


Figure 42: Inference results for the Fitz-Hugh Nagumo system, equations (40-41). Each column represents one of the three kinetic parameters of the system, and the histograms show the average posterior distributions of the respective parameter, averaged over ten data instantiations. Vertical line: true parameter value. Black line: prior distribution. The top row shows the results for the AGM model (paradigm A). The bottom row shows the results for the GPODE model (paradigm B). Since the results for the priors used in Campbell and Steele [9] – a non-negative truncated $N(0, 0.4)$ and a $\chi^2(2)$ distribution – were similar, only the results for the uniform prior are shown.

6.6 Conclusions

Inference in mechanistic models based on non-affine ODEs is challenging due to the high computational costs of the numerical integration of the ODEs, and approximate methods based on adaptive gradient matching have therefore gained much attention in the last few years. The application of nonparametric Bayesian methods based on GPs is particularly promising owing to the fact that a GP is closed under differentiation. A new paradigm termed GPODE was proposed in Wang and Barber [49] at ICML 2014, which was purported to outperform state-of-the-art GP gradient matching methods in three respects: providing a simplified mathematical description, constituting a probabilistic generative model, and achieving better inference results. The purpose of the present chapter has been to critically review these claims. It turns out that the simplicity of the model presented in Wang and Barber [49], shown in Figure 38, results from confusing the marginalisation over a random variable with its elimination from the model. A proper representation of the GPODE model leads to a more complex form, shown in Figure 39. It has been shown that the GPODE model is turned into a probabilistic generative model at the expense of certain independence assumptions, which are implausible and have not been made explicit in Wang and Barber [49]. Furthermore, it has been shown that as a consequence of these independence assumptions, the GPODE model is susceptible to identifiability problems when data are systematically missing. This problem is inherent in the GPODE model, and is avoided when gradient matching with GPs follows the product of experts

approach of Calderhead et al. [8] and Dondelinger et al. [11] (herein called paradigm A/AGM). Unlike Wang and Barber [49], the empirical comparison in this chapter has not shown any performance improvement over paradigm A. On the contrary, for two systems (simple ODE with missing values, and the Lotka-Volterra system), paradigm A achieves significantly better results. For a third system (Fitz-Hugh Nagumo system), both approaches are on a par, with different bias/variance characteristics.

The right-hand panel of Figure 39 demonstrates that gradient matching intrinsically violates the DAG constraint. This is because the function to be matched is both the output of and the input to the ODEs, leading to a directed cycle. The endeavour to model gradient matching with GPs as a probabilistic generative model based on a DAG at the expense of implausible dummy variables and independence assumptions (Figure 39, centre panel) is at the heart of the problems with the GPODE model, as previously discussed. It has been demonstrated that these problems can be avoided with gradient matching paradigm A. The study in this chapter clearly suggests that for practical applications, paradigm A is to be preferred over paradigm B. Wang and Barber [49] argue that a principled shortcoming of paradigm A is the fact that the underlying product of experts approach cannot be formulated in terms of a probabilistic generative model. However, as has just been discussed, this is of little relevance, given that gradient matching cannot be consistently conceptualised as a probabilistic generative model *per se*. This

methodological limitation is the price that has to be paid for the substantial computational advantages over the explicit solution of the ODEs that gradient matching yields.

7 Performing Model Selection via Estimation of the Marginal Likelihood by Combining Thermodynamic Integration and Gradient Matching

7.1 Introduction

Parameter inference in ODEs relates to statistically inferring the size of an effect of components in certain processes, but model selection instead aims at discerning between different hypotheses describing the structure of the systems.

Using a naive approach by choosing the model that simply has the largest likelihood, results in poor model selection performance. It is clear that a maximum of a function of a subset can never be higher than the function defined over a total set (at most, the maximum of the subset will be the same as the maximum over the total set). Therefore, for nested models (which typically exist when proposing different candidate ODE models), the maximum likelihood of the less complex model will always be equal to or less than the more complex model. Hence, performing model selection based solely on choosing the model that generates the largest likelihood, is spurious.

There are two main approaches to model selection, that aim to avoid problems occurred by solely relying on the maximum likelihood of the competing models. These are known as explanatory model selection and predictive model selection.

Explanatory model selection is the method of integrating over the parameters and focussing on the marginal likelihood of the data i.e. the probability of the data per se and not the probability of the data given some parameter set.

The posterior probability of the candidate models is given by

$$p(M|\mathbf{Y}) = \frac{p(\mathbf{Y}|M)p(M)}{p(\mathbf{Y})}, \tag{92}$$

where $\mathbf{Y}$ denotes the data and M represents different models.

Assuming a uniform prior over the models, equation 92 is maximised by the term $p(\mathbf{Y}|M)$ and therefore explanatory model selection can be conducted by focussing on this term. This term is known as the marginal likelihood and is equal to

$$p(\mathbf{Y}|M) = \int p(\mathbf{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta}|M)d\boldsymbol{\theta}. \tag{93}$$

It is then possible to assess the plausibility of the competing models by computing the Bayes factor

$$\begin{aligned} \text{Bayes factor} &= \frac{p(\mathbf{Y}|M_i)}{p(\mathbf{Y}|M_j)} \\ &= \frac{\int p(\mathbf{Y}|\boldsymbol{\theta}_i)p(\boldsymbol{\theta}_i|M_i)d\boldsymbol{\theta}_i}{\int p(\mathbf{Y}|\boldsymbol{\theta}_j)p(\boldsymbol{\theta}_j|M_j)d\boldsymbol{\theta}_j}, \end{aligned} \tag{94}$$

where the index i represents the candidate model and parameters associated with model i and j represents the candidate model and parameters associated with model j. If the ratio in equation 94 is less than 1, this is evidence in favour of model j, whereas if the ratio is greater than 1, this is evidence in favour of model i. This is equivalent to just selecting the model that produces the highest marginal likelihood.

How then does the marginal likelihood guard against overly complex models? Given that the parameters are being integrated over rather than maximised, then models that have higher likelihood do not necessarily have higher marginal likelihood. A graphical depiction of the reason behind this can be seen, for example, in Figure 5.6 of "Machine learning. A probabilistic perspective" by Kevin P. Murphy [36]. A reproduction of this plot is included in Figure 43.

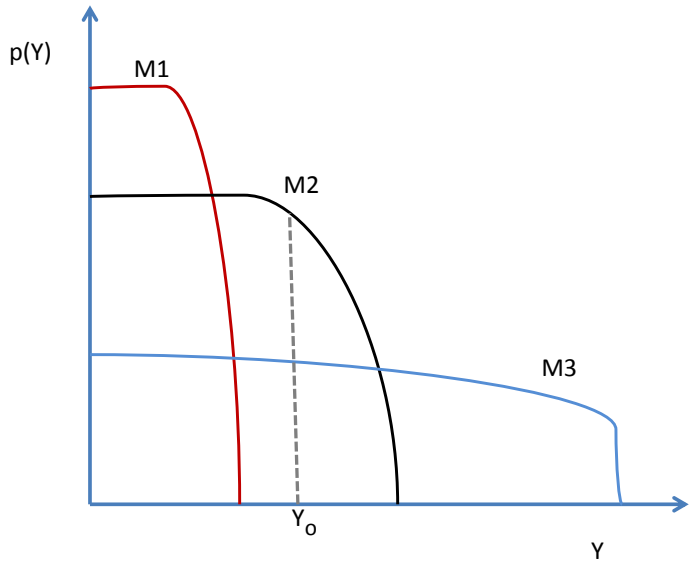


Figure 43: Reconstruction of Figure 5.6 from [36]. An illustration as to how marginal likelihoods adhere to Occam's razor. The y-axis shows the marginal likelihood, $p(\mathbf{Y})$ and the x-axis depicts different datasets that exist. $\mathbf{Y}_o$ is the observed dataset. The red line (M1) represents a model that is too simplistic and is unable to fit to the observed dataset well. The blue line (M3) represents a complex model. Although it can fit the observed dataset, it can also fit many more due to its increased complexity. Hence the marginal likelihood at the observed dataset is too low. The black line (M2) represents the true model and it achieves the highest marginal likelihood of the candidates at the observed dataset.

A more complex model can fit many different types of datasets and therefore $p(\mathbf{Y}|M)$ will be more diffused and $p(\mathbf{Y}|M)$ evaluated for the observed dataset will be smaller than a less complex model, unless the less complex model is

incapable of modelling the data.

The main difficulty, however, in computing the marginal likelihood is that usually the integral in equation 93 is not available in closed form, and the techniques used to calculate it are computationally expensive.

Thermodynamic integration, successfully used in the field of Statistical Physics and more recently introduced into the wider Statistical community by Friel and Pettitt [15], is a promising method for computing the Bayes factors. It uses the components that are already calculated in the parallel tempering scheme outlined in Chapter 4 in order to compute the log marginal likelihood. This can be done for the competing models and the exponent of the subsequent difference between each pair of candidate models is the Bayes factor.

The latter approach, predictive model selection, is a measure of out of sample performance. However, approaches such as cross validation are computationally expensive and quite often information criteria are used instead. In contrast to explanatory model selection, this approach does not integrate over the parameters. Instead, it uses the likelihood, which is the probability of the data given the parameters, and therefore model selection in this manner can be thought of as being conducted by means of predictive performance. This is true for most information criteria, but not for BIC, which instead

attempts to approximate the log marginal likelihood.

The other information criteria tend to be estimates and approximations to some externally- or cross- validated fit [16]. Cross-validation has been demonstrated to provide an accurate way of estimating a model's predictive performance, however, these methods tend to be time-consuming. The natural step would then be to approximate the method of cross-validation to some degree, for example, AIC is asymptotically equivalent to cross-validation. WAIC on the other hand (which is an improvement over DIC, since DIC cannot deal with singular likelihood functions), is a recent method that is asymptotically equivalent to Bayesian leave-one-out cross-validation [51] and is given by

$$\mathrm{WAIC}(N) = -2 \sum_{i=1}^{N} \left(\log \mathbb{E}(p(y_i|\boldsymbol{\theta})) - \mathbb{V}(\log p(y_i|\boldsymbol{\theta})) \right), \tag{95}$$

where the expectation $\mathbb{E}$ and variance $\mathbb{V}$ are taken with respect to the posterior samples and N is the number of datapoints. Watanabe [50] shows that expectation of the Bayes generalisation loss (BgL) is asymptotically equal to

$$\mathbb{E}\left[BgL(N)\right] = \mathbb{E}\left[\mathrm{WAIC}(N)\right] + o\left(\frac{1}{N}\right). \tag{96}$$

Now writing the predictive distribution, leaving out y_i, as

$$p^{(i)}(y) = \mathbb{E}^{(i)}\left[p(y|\theta)\right], \tag{97}$$

then the log loss when y_i is used as a testing sample is

$$-\log p^{(i)}(y) = -\log \mathbb{E}^{(i)}\left[p(y|\theta)\right]. \tag{98}$$

Hence, the log loss of the Bayes cross-validation (CvL) is defined as the empirical average of them,

$$CvL(N) = -\frac{1}{N}\log \mathbb{E}^{(i)}\left[p(y|\theta)\right]. \tag{99}$$

Watanabe [51] shows that since $y_1, \ldots, y_N$ are independent training samples, it follows that

$$\mathbb{E}\left[CvL(N)\right] = \mathbb{E}\left[BgL(N-1)\right] \tag{100}$$

and therefore, by using equation 96, it follows that

$$\mathbb{E}\left[CvL(N)\right] = \mathbb{E}\left[\text{WAIC}(N-1)\right] + o\left(\frac{1}{N}\right). \tag{101}$$

Hence, $\mathbb{E}\left[CvL(N)\right]$ and $\mathbb{E}\left[\text{WAIC}(N-1)\right]$ are asymptotically equivalent to one another.

The Kullback-Leibler divergence is a measure of "distance" between any two distributions [54]. Terming $\tilde{y}$ to represent some future observation, $f(\tilde{y})$ to represent the probability density of the true model and $g(\tilde{y})$ as the probability density of the approximating model, then the Kullback-Leibler divergence is given by

$$KL(f,g) = \int f(\tilde{y}) \log \frac{f(\tilde{y})}{g(\tilde{y})} d\tilde{y} = \mathbb{E}_f\left[\log f(\tilde{y})\right] - \mathbb{E}_f\left[\log g(\tilde{y})\right]. \tag{102}$$

The Kullback-Leibler divergence, equation 102 can be interpreted as the information lost when $g(\cdot)$ is used to approximate $f(\cdot)$. The smaller $KL(f,g)$, the closer the model g is to the true distribution. In the absence of full knowledge of true distribution, only the second term of $KL(f,g)$ is relevant in comparing different possible models, since the first term is a function of f, but independent of the candidate model g. By the law of large numbers, as $n \to \infty$, the average of the log likelihood, $\frac{1}{n}\sum_{i=1}^{n} \log g(y_i|\theta)$, tends to $\mathbb{E}_f\left[\log g(\tilde{y}|\theta)\right]$. Akaike [2] showed that by assuming the fitted model with the maximum

likelihood estimate of θ as the best for the family $G = \{g(\tilde{y}|\theta), \theta \in \Theta\}$ then asymptotically

$$\frac{1}{n}\sum_{i=1}^{n} \log g(y_i|\theta_{\text{MLE}}) - \frac{p}{n} \cong \mathbb{E}_f \left[\log g(\tilde{y}|\theta_{\text{MLE}}) \right], \tag{103}$$

where p, the number of parameters, penalises over-estimating the out of sample log likelihood. AIC is the estimator of equation 103, multiplied by $-2n$.

Another information criterion that is relevant to this chapter is that of BIC. BIC is an approach that attempts to approximate the log marginal likelihood, as a work around for the difficulties aforementioned in computing the integral in equation 93. It is defined as

$$\begin{aligned} \text{BIC} &= -2\log p(\mathbf{Y}|\boldsymbol{\theta}_{\text{MLE}}) + D(\boldsymbol{\theta}_{\text{MLE}})\log N \\ &\approx -2\log p(\mathbf{Y}|M) \end{aligned} \tag{104}$$

where $D(\boldsymbol{\theta}_{\text{MLE}})$ are the number of parameters in the model and N is the number of datapoints. BIC is asymptotically equivalent to the log marginal likelihood and is derived in the following fashion. By using a Laplace approximation (see equation 8.52 in Chapter 8 of Murphy [36]), it is possible to write the log marginal likelihood as

$$\log p(\mathbf{Y}) \approx \log p(\mathbf{Y}|\boldsymbol{\theta}_{\text{Mode}}) + \log p(\boldsymbol{\theta}_{\text{Mode}}) - \frac{1}{2}\log |\mathbf{H}|, \tag{105}$$

where $\boldsymbol{\theta}_{\text{Mode}}$ is the mode of the parameters and $\mathbf{H}$ is the Hessian matrix. Assuming a uniform prior, the term $p(\boldsymbol{\theta}_{\text{Mode}})$ can be dropped and $\boldsymbol{\theta}_{\text{Mode}}$ replaced with $\boldsymbol{\theta}_{\text{MLE}}$, the maximum likelihood estimator. Now denote $\mathbf{H} = \sum_{i=1}^{N} \mathbf{H}_i$, where $\mathbf{H}_i$ is the second derivative of $\log p(y_i|\boldsymbol{\theta})$. The next step is to approximate each $\mathbf{H}_i$ by a fixed matrix $\tilde{\mathbf{H}}$. This means, assuming $\mathbf{H}$ is full rank,

$$\log |\mathbf{H}| = \log |N\tilde{\mathbf{H}}| = \log \left(N^{D(\boldsymbol{\theta}_{\text{MLE}})}|\tilde{\mathbf{H}}|\right) = D \log N + \log |\tilde{\mathbf{H}}|. \tag{106}$$

Since $\log |\tilde{\mathbf{H}}|$ is independent of N, this term can be dropped also (since the dominating term will be the likelihood for $N \to \infty$). Substituting back into equation 105 yields

$$\log p(\mathbf{Y}) \approx \log p(\mathbf{Y}|\boldsymbol{\theta}_{\text{MLE}}) - \frac{D(\boldsymbol{\theta}_{\text{MLE}})}{2}\log N$$

$$= -2\log p(\mathbf{Y}) \approx -2\log p(\mathbf{Y}|\boldsymbol{\theta}_{\text{MLE}}) + D(\boldsymbol{\theta}_{\text{MLE}})\log N. \tag{107}$$

It should be noted that it can be difficult in practice to satisfy the asymptotic assumptions of information criteria, which can often lead to poor approximations of the quantity of interest.

This chapter first combines the method of calculating the log marginal likelihood using thermodynamic integration with that of gradient matching and then demonstrates that conducting model selection using the standard computational form of thermodynamic integration (now combined with gradient matching) is suboptimal and will decrease the accuracy of the Bayes factors. An alternative form of calculating the log marginal likelihood using thermodynamic integration combined with gradient matching will be proposed and it will be discussed that this leads to more accurate estimates of the Bayes factors and a robust way of performing model selection in ordinary differential equation models with gradient matching. This new method will be compared to the results of WAIC and BIC.

This chapter combines the method of calculating the log marginal likelihood using thermodynamic integration with that of gradient matching. It demonstrates that conducting model selection using this form is suboptimal and will decrease the accuracy of the Bayes factors. An alternative form of combining thermodynamic integration with gradient matching will be proposed and it will be discussed that this leads to more accurate estimates of the Bayes factors. This new method will be compared to the results of WAIC and BIC.

7.2 Methodology

A central objective of model selection using Bayes factors is to calculate the marginal likelihood of a model. The Bayes factor is then computed by calculating the ratio of the marginal likelihoods, or the difference of the log marginal likelihoods, of the competing models. Thermodynamic integration is therefore useful, as it provides a way to compute the log marginal likelihood for a given model, using the tempered versions of the likelihood in equation 66. This gives a framework for the computation of the integral in equation 93, which is one of the main difficulties in practically performing explanatory model selection. Note that in this chapter the dependency on the particular model is not made explicit in the notation, for ease of reading, i.e. $p(\mathbf{Y}) = p(\mathbf{Y}|M)$.

Friel and Pettitt [15] show that the log marginal likelihood can be computed by taking the derivative of $\log p(\mathbf{Y}|\alpha^{(i)})$ with respect to the temperatures and then integrating over the temperatures. The starting point is

$$\frac{d}{d\alpha^{(i)}} \log p(\mathbf{Y}|\alpha^{(i)}) = \frac{1}{p(\mathbf{Y}|\alpha^{(i)})} \frac{d}{d\alpha^{(i)}} p(\mathbf{Y}|\alpha^{(i)}). \tag{108}$$

The tempered posterior distribution of the latent variables and parameters for the adaptive gradient matching method [11] described in Chapter 4 can be written as

$$p(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\sigma}^2 | \mathbf{Y}, \alpha^{(i)}) = \frac{1}{p(\mathbf{Y}|\alpha^{(i)})} p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2)^{\alpha^{(i)}}$$

$$p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma}) p(\boldsymbol{\theta}) p(\boldsymbol{\eta}) p(\boldsymbol{\gamma}) p(\boldsymbol{\sigma}^2), \tag{109}$$

where

$$p(\mathbf{Y}|\alpha^{(i)}) = \int_{\mathbf{X}} \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\eta}} \int_{\boldsymbol{\gamma}} \int_{\boldsymbol{\sigma}^2} p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2)^{\alpha^{(i)}}$$

$$p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma}) p(\boldsymbol{\theta}) p(\boldsymbol{\eta}) p(\boldsymbol{\gamma}) p(\boldsymbol{\sigma}^2) d\mathbf{X} d\boldsymbol{\theta} d\boldsymbol{\eta} d\boldsymbol{\gamma} d\boldsymbol{\sigma}^2. \tag{110}$$

Hence,

$$\frac{d}{d\alpha^{(i)}}\log p(\mathbf{Y}|\alpha^{(i)}) = \frac{1}{p(\mathbf{Y}|\alpha^{(i)})}\frac{d}{d\alpha^{(i)}}\int_{\mathbf{X}}\int_{\boldsymbol{\theta}}\int_{\boldsymbol{\eta}}\int_{\boldsymbol{\gamma}}\int_{\boldsymbol{\sigma}^2} p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)^{\alpha^{(i)}}$$

$$p(\mathbf{X}|\boldsymbol{\theta},\boldsymbol{\eta},\boldsymbol{\gamma})p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma})p(\boldsymbol{\sigma}^2)d\mathbf{X}d\boldsymbol{\theta}d\boldsymbol{\eta}d\boldsymbol{\gamma}d\boldsymbol{\sigma}^2$$

$$= \frac{1}{p(\mathbf{Y}|\alpha^{(i)})}\int_{\mathbf{X}}\int_{\boldsymbol{\theta}}\int_{\boldsymbol{\eta}}\int_{\boldsymbol{\gamma}}\int_{\boldsymbol{\sigma}^2} \log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)^{\alpha^{(i)}}$$

$$p(\mathbf{X}|\boldsymbol{\theta},\boldsymbol{\eta},\boldsymbol{\gamma})p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma})p(\boldsymbol{\sigma}^2)d\mathbf{X}d\boldsymbol{\theta}d\boldsymbol{\eta}d\boldsymbol{\gamma}d\boldsymbol{\sigma}^2$$

$$= \int_{\mathbf{X}}\int_{\boldsymbol{\theta}}\int_{\boldsymbol{\eta}}\int_{\boldsymbol{\gamma}}\int_{\boldsymbol{\sigma}^2} \log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)\frac{1}{p(\mathbf{Y}|\alpha^{(i)})}p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)^{\alpha^{(i)}}$$

$$p(\mathbf{X}|\boldsymbol{\theta},\boldsymbol{\eta},\boldsymbol{\gamma})p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma})p(\boldsymbol{\sigma}^2)d\mathbf{X}d\boldsymbol{\theta}d\boldsymbol{\eta}d\boldsymbol{\gamma}d\boldsymbol{\sigma}^2$$

$$= \int_{\mathbf{X}}\int_{\boldsymbol{\theta}}\int_{\boldsymbol{\eta}}\int_{\boldsymbol{\gamma}}\int_{\boldsymbol{\sigma}^2} \log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)p(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\eta},\boldsymbol{\gamma},\boldsymbol{\sigma}^2|\mathbf{Y},\alpha^{(i)})d\mathbf{X}d\boldsymbol{\theta}d\boldsymbol{\eta}d\boldsymbol{\gamma}d\boldsymbol{\sigma}^2$$

$$= \mathbb{E}_{\alpha^{(i)}}[\log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)]. \tag{111}$$

Note that the expectation in equation 111 is for fixed temperature $\alpha^{(i)}$, i.e. the expected value of the log likelihood uses the sampled $\mathbf{X}$ and $\boldsymbol{\sigma}^2$ from the given temperature chain. The marginal likelihood $p(\mathbf{Y})$ is simply $p(\mathbf{Y}|\alpha^{(i)} = 1)$ and the normalisation of the posterior distribution implies that $p(\mathbf{Y}|\alpha^{(i)} = 0) = 1$.

The log marginal likelihood is therefore

$$\begin{aligned}\log p(\mathbf{Y}) &= \log p(\mathbf{Y}|\alpha^{(i)}=1) - \log p(\mathbf{Y}|\alpha^{(i)}=0) \\ &= \int_{\alpha^{(i)}=0}^{\alpha^{(i)}=1} \frac{d}{d\alpha^{(i)}} \log p(\mathbf{Y}|\alpha^{(i)}) d\alpha^{(i)} \\ &= \int_{\alpha^{(i)}=0}^{\alpha^{(i)}=1} \mathbb{E}_{\alpha^{(i)}}[\log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2)] d\alpha^{(i)}. \end{aligned} \tag{112}$$

The integral in equation 112 can be solved numerically, for example, using the trapezoidal rule. It is important that due consideration is used in choosing the discretisation of the temperatures $\alpha^{(i)} = \{0, \ldots, 1\}$, as the largest contributions to this integral usually come from a small region around $\alpha^{(i)}$ close to 0. This motivates the discretisation form outlined in Friel and Pettitt [15] and the justification for the selection used throughout this thesis.

A drawback to this scheme however, is that the distribution of the data $p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2)^{\alpha^{(i)}}$ is tempered, whereas the distribution controlling the mismatch to the gradients and draws of the latent variables $p(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma})$ is not. This leads to poor mixing and convergence of the Markov chains, subsequently leading to poorer parameter estimation, which in turn negatively affects the accuracy of the Bayes factors. This observation was noted when originally testing the method and the poor results motivated the alternate scheme that is about to follow. Due to the poor mixing and convergence, the result-

ing poor parameter estimation and poor accuracy of the Bayes factors, this method was abandoned in favour of the alternative method that is to follow, and was not included for the comparisons on the simulation studies detailed in this chapter.

Due to the aforementioned issues, it would be better to also temper the distribution controlling the mismatch between the gradients, creating less disparity with the proposal distribution in the MCMC and therefore increasing the mixing and convergence of the chains. In order to do this, it is necessary to separate equation 65 into two parts: the Gaussian process part and the part that penalises the differences between the gradients.

Based on equation 65, it is possible to write the joint probability of the latent variables and parameters as

$$p(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma}) = \frac{\zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}) p(\mathbf{X}|\boldsymbol{\eta}) p(\boldsymbol{\theta}) p(\boldsymbol{\eta}) p(\boldsymbol{\gamma})}{C}, \tag{113}$$

where $\zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma})$ is a potential function (an un-normalised probability distribution), defined by equation 63 ($\zeta(\cdot)$ here is being used as shorthand for the solution to the integral of equation 63), $p(\mathbf{X}|\boldsymbol{\eta})$ is the distribution of the Gaussian process with hyperparameters $\boldsymbol{\eta}$ and the normalisation constant C is defined as

$$C = \int_{\mathbf{X}} \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\eta}} \int_{\gamma} \zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}) p(\mathbf{X}|\boldsymbol{\eta}) p(\boldsymbol{\theta}) p(\boldsymbol{\eta}) p(\boldsymbol{\gamma}) d\mathbf{X} d\boldsymbol{\theta} d\boldsymbol{\eta} d\boldsymbol{\gamma}. \tag{114}$$

Note that $\boldsymbol{\phi}$ in equation 65 is just the sample mean and not sampled as a parameter in the MCMC scheme and therefore is omitted from the notation in this chapter. The joint probability of the whole system now becomes

$$p(\mathbf{Y}, \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\sigma}^2) = p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2)p(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma})p(\boldsymbol{\sigma}^2)$$

$$= \frac{p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2)\zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma})p(\mathbf{X}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma})p(\boldsymbol{\sigma}^2)}{C}, \tag{115}$$

which therefore implies that the tempered posterior distribution of the latent variables and parameters is given by

$$p(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\sigma}^2|\mathbf{Y}, \alpha^{(i)}) = \frac{1}{\mathbb{Z}(\mathbf{Y}|\alpha^{(i)})} \left[p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2)\zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma})\right]^{\alpha^{(i)}}$$

$$p(\mathbf{X}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma})p(\boldsymbol{\sigma}^2), \tag{116}$$

and $\mathbb{Z}(\mathbf{Y}|\alpha^{(i)})$ as

$$\mathbb{Z}(\mathbf{Y}|\alpha^{(i)}) = \int_{\mathbf{X}} \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\eta}} \int_{\boldsymbol{\gamma}} \int_{\boldsymbol{\sigma}^2} \left[p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2)\zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma})\right]^{\alpha^{(i)}}$$

$$p(\mathbf{X}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma})p(\boldsymbol{\sigma}^2)d\mathbf{X}d\boldsymbol{\theta}d\boldsymbol{\eta}d\boldsymbol{\gamma}d\boldsymbol{\sigma}^2. \tag{117}$$

Taking the derivative of $\log Z(\mathbf{Y}|\alpha^{(i)})$ will yield

$$\frac{d}{d\alpha^{(i)}}\log \mathbb{Z}(\mathbf{Y}|\alpha^{(i)}) = \frac{1}{\mathbb{Z}(\mathbf{Y}|\alpha^{(i)})}\frac{d}{d\alpha^{(i)}}\mathbb{Z}(\mathbf{Y}|\alpha^{(i)})$$

$$= \frac{1}{\mathbb{Z}(\mathbf{Y}|\alpha^{(i)})}\frac{d}{d\alpha^{(i)}}\int_{\mathbf{X}}\int_{\boldsymbol{\theta}}\int_{\boldsymbol{\eta}}\int_{\boldsymbol{\gamma}}\int_{\boldsymbol{\sigma}^2}\left[p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)\zeta(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\gamma})\right]^{\alpha^{(i)}}$$

$$p(\mathbf{X}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma})p(\boldsymbol{\sigma}^2)d\mathbf{X}d\boldsymbol{\theta}d\boldsymbol{\eta}d\boldsymbol{\gamma}d\boldsymbol{\sigma}^2$$

$$= \frac{1}{\mathbb{Z}(\mathbf{Y}|\alpha^{(i)})}\int_{\mathbf{X}}\int_{\boldsymbol{\theta}}\int_{\boldsymbol{\eta}}\int_{\boldsymbol{\gamma}}\int_{\boldsymbol{\sigma}^2}\log\ \left[p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)\zeta(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\gamma})\right]$$

$$\left[p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)\zeta(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\gamma})\right]^{\alpha^{(i)}}p(\mathbf{X}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma})p(\boldsymbol{\sigma}^2)d\mathbf{X}d\boldsymbol{\theta}d\boldsymbol{\eta}d\boldsymbol{\gamma}d\boldsymbol{\sigma}^2$$

$$= \int_{\mathbf{X}}\int_{\boldsymbol{\theta}}\int_{\boldsymbol{\eta}}\int_{\boldsymbol{\gamma}}\int_{\boldsymbol{\sigma}^2}\log\ \left[p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)\zeta(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\gamma})\right]\frac{1}{\mathbb{Z}(\mathbf{Y}|\alpha^{(i)})}$$

$$\left[p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)\zeta(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\gamma})\right]^{\alpha^{(i)}}p(\mathbf{X}|\boldsymbol{\eta})p(\boldsymbol{\theta})p(\boldsymbol{\eta})p(\boldsymbol{\gamma})p(\boldsymbol{\sigma}^2)d\mathbf{X}d\boldsymbol{\theta}d\boldsymbol{\eta}d\boldsymbol{\gamma}d\boldsymbol{\sigma}^2$$

$$= \int_{\mathbf{X}}\int_{\boldsymbol{\theta}}\int_{\boldsymbol{\eta}}\int_{\boldsymbol{\gamma}}\int_{\boldsymbol{\sigma}^2}\log\ \left[p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)\zeta(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\gamma})\right]$$

$$p(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\eta},\boldsymbol{\gamma},\boldsymbol{\sigma}^2|\mathbf{Y},\alpha^{(i)})d\mathbf{X}d\boldsymbol{\theta}d\boldsymbol{\eta}d\boldsymbol{\gamma}d\boldsymbol{\sigma}^2$$

$$= \mathbb{E}_{\alpha^{(i)}}\left[\log p(\mathbf{Y}|\mathbf{X},\boldsymbol{\sigma}^2)\right] + \mathbb{E}_{\alpha^{(i)}}\left[\log \zeta(\mathbf{X},\boldsymbol{\theta},\boldsymbol{\gamma})\right]. \tag{118}$$

This in turn means that

$$\log \mathbb{Z}(\mathbf{Y}) = \log \mathbb{Z}(\mathbf{Y}|\alpha^{(i)} = 1) - \log \mathbb{Z}(\mathbf{Y}|\alpha^{(i)} = 0)$$

$$= \int_{\alpha^{(i)}=0}^{\alpha^{(i)}=1} \frac{d}{d\alpha^{(i)}} \log \mathbb{Z}(\mathbf{Y}|\alpha^{(i)}) d\alpha^{(i)}$$

$$= \int_{\alpha^{(i)}=0}^{\alpha^{(i)}=1} \mathbb{E}_{\alpha^{(i)}} \left[\log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\sigma}^2) \right] d\alpha^{(i)} + \int_{\alpha^{(i)}=0}^{\alpha^{(i)}=1} \mathbb{E}_{\alpha^{(i)}} \left[\log \zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}) \right] d\alpha^{(i)}, \tag{119}$$

where the first line in equation 119 follows from equation 117. The log marginal likelihood can now be expressed as

$$\log p(\mathbf{Y}) = \log \mathbb{Z}(\mathbf{Y}) - \log (C). \tag{120}$$

C can depend on the ODE model structure and is estimated by sampling equation 114 using MCMC. Note: since C does not depend on the data, this term can be estimated even before the data is collected, in order to speed up the whole process. Now define

$$\mathbb{Z}(C|\alpha^{(i)}) = \int_{\mathbf{X}} \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\eta}} \int_{\boldsymbol{\gamma}} \zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma})^{\alpha^{(i)}} p(\mathbf{X}|\boldsymbol{\eta}) p(\boldsymbol{\theta}) p(\boldsymbol{\eta}) p(\boldsymbol{\gamma}) d\mathbf{X} d\boldsymbol{\theta} d\boldsymbol{\eta} d\boldsymbol{\gamma}. \tag{121}$$

To approximate $\log (C)$ using thermodynamic integration, it is necessary to

compute the derivative of $\log \mathbb{Z}(C|\alpha^{(i)})$:

$$\frac{d}{d\alpha^{(i)}} \log \mathbb{Z}(C|\alpha^{(i)}) = \frac{1}{\mathbb{Z}(C|\alpha^{(i)})} \frac{d}{d\alpha^{(i)}} \mathbb{Z}(C|\alpha^{(i)})$$

$$= \frac{1}{\mathbb{Z}(C|\alpha^{(i)})} \frac{d}{d\alpha^{(i)}} \int_{\mathbf{X}} \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\eta}} \int_{\boldsymbol{\gamma}} \zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma})^{\alpha^{(i)}} p(\mathbf{X}|\boldsymbol{\eta}) p(\boldsymbol{\theta}) p(\boldsymbol{\eta}) p(\boldsymbol{\gamma}) d\mathbf{X} d\boldsymbol{\theta} d\boldsymbol{\eta} d\boldsymbol{\gamma}$$

$$= \frac{1}{\mathbb{Z}(C|\alpha^{(i)})} \int_{\mathbf{X}} \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\eta}} \int_{\boldsymbol{\gamma}} \log \zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}) \zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma})^{\alpha^{(i)}}$$

$$p(\mathbf{X}|\boldsymbol{\eta}) p(\boldsymbol{\theta}) p(\boldsymbol{\eta}) p(\boldsymbol{\gamma}) d\mathbf{X} d\boldsymbol{\theta} d\boldsymbol{\eta} d\boldsymbol{\gamma}$$

$$= \int_{\mathbf{X}} \int_{\boldsymbol{\theta}} \int_{\boldsymbol{\eta}} \int_{\boldsymbol{\gamma}} \log \zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}) \frac{1}{\mathbb{Z}(C|\alpha^{(i)})} \zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma})^{\alpha^{(i)}}$$

$$p(\mathbf{X}|\boldsymbol{\eta}) p(\boldsymbol{\theta}) p(\boldsymbol{\eta}) p(\boldsymbol{\gamma}) d\mathbf{X} d\boldsymbol{\theta} d\boldsymbol{\eta} d\boldsymbol{\gamma}$$

$$= \tilde{\mathbb{E}}_{\alpha^{(i)}} \left[\log \zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}) \right]. \tag{122}$$

Note that the $\tilde{\mathbb{E}}$ signifies that this expectation has been taken with respect to the probability distribution in equation 113 i.e. the data is not included in the MCMC sampling. Now, it is possible to compute $\log(C)$ using thermodynamic integration

$$\log(C) = \log(C|\alpha^{(i)} = 1) - \log(C|\alpha^{(i)} = 0)$$

$$= \int_{\alpha^{(i)}=0}^{\alpha^{(i)}=1} \frac{d}{d\alpha^{(i)}} \log \mathbb{Z}(C|\alpha^{(i)}) d\alpha^{(i)}$$

$$= \int_{\alpha^{(i)}=0}^{\alpha^{(i)}=1} \tilde{\mathbb{E}}_{\alpha^{(i)}} \left[\log \zeta(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}) \right] d\alpha^{(i)}. \tag{123}$$

Whilst it is possible to compute $\log(C)$ using thermodynamic integration, given that the integrand for C should be a lot smoother than for the likelihood, it is possible to instead approximate equation 114 using a simple Monte Carlo sum i.e.

$$C = \frac{1}{N_{iter}} \sum_{i=1}^{N_{iter}} \zeta(\mathbf{X}_i, \boldsymbol{\theta}_i, \boldsymbol{\gamma}_i), \tag{124}$$

where the draws required to compute $\zeta(\mathbf{X}_i, \boldsymbol{\theta}_i, \boldsymbol{\gamma}_i)$ are sampled from the priors $p(\boldsymbol{\eta})$, $p(\boldsymbol{\gamma})$, $p(\boldsymbol{\theta})$ and $p(\mathbf{X}|\boldsymbol{\eta})$, with acceptance probability 1. In the examples looked at in Chapter 7.3, the simple Monte Carlo sum was quick to converge and thus equation 124 was used to compute C.

7.3 Simulation

The proposed method was tested on data generated from each of these models in turn. For ease of reading, denote equation 42 as LV1 (for Lotka-Volterra), equation 43 as LV2 (for Lotka-Volterra intra-species competition) and equa-

tion 44 as LV3 (for Lotka-Volterra saturation term), respectively. 10 datasets were generated from each model in turn and iid Gaussian noise (SD = 0.5, average SNR for each "species" = 10) was added for the LV1 and LV3 models and iid Gaussian noise (SD = 0.2, average SNR for each "species" = 10) was added for the LV2.

Lotka-Volterra Original Model (LV1)

Data was generated with the following parameters: $\theta_1 = 2$, $\theta_2 = 1$, $\theta_3 = 4$ and $\theta_4 = 1$. Starting from initial values of (5,3) for the two "species", 11 timepoints were generated over the time course [0,2], producing one period. The priors over the parameters were $\Gamma(4, 0.5)$ prior. These settings were chosen to correspond with the set-up in Dondelinger et al. [11].

Lotka-Volterra Intra-Species Competition Model (LV2)

Data was generated with the following parameters: $\theta_1 = 4$, $\theta_2 = 1$, $\theta_3 = 4$, $\theta_4 = 2$ and $\theta_5 = 5$. Starting from initial values of (5,3) for the two "species", 11 timepoints were generated over the time course [0,2], producing one period. The parameters for this scneario were chosen so that the parameter controlling the intra-species term (θ_5) could be large enough to distinguish the model from the LV1 model (where, as $\theta_5 \to 0$, LV2 $\to$ LV1). The other parameters were chosen to ensure the signals were smooth. The priors over the parameters were $\Gamma(4, 0.5)$ prior for θ_1, θ_2, θ_3 and θ_4 and a $U(0, 9)$ for θ_5 as there was no indication from previous work what a suitable prior would

be for the parameter governing the intra-species term.

Lotka-Volterra Saturation Term Model (LV3)

Data was generated with the following parameters: $\theta_1 = 2.8$, $\theta_2 = 3.5$, $\theta_3 = 1$, $\theta_4 = 2.5$ and $\theta_5 = 1$. Starting from initial values of (5,3) for the two "species", 11 timepoints were generated over the time course [0,2], producing one period. The saturation term included in these ODEs should mean that the less complex models are unable to produce signals that match the shape of the signals produced by the LV3 model. Hence, if the model selection method is working properly, this model should be clearly favoured over the other two. The priors over the parameters were $\Gamma(4, 0.5)$ prior for θ_1, θ_2, θ_3 and θ_4 and a $U(0, 9)$ for θ_5 (reflecting the extra uncertainty surrounding the 5^{th} parameter).

Protein Signalling Transduction Pathway For ease of reading, equations 45 - 48 will be referred to as PSTP1, PSTP2, PSTP3 and PSTP4, respectively. A graphical representation of PSTP1 can be found in Figure 4 and graphical representations of PSTP2-4 can be found in Figures 6-8. Data was generated from PSTP1 as it provided a reasonable degree of complexity and was neither the least complex model nor the most complex model out of the four. This feature is very important, since otherwise it would be difficult to ascertain whether the model selection method was working properly, or whether it was biased and just happened to favour the least/most complex

model. 10 datasets were generated and iid Gaussian noise (SD = 0.0635, average SNR for each "species" = 10) was added. For the "species" that did not have data (PhA and RppPhA), the rate of change was set to zero, which implies a constant rate over time. This corresponds to a component that is disconnected from the rest of the system. For these components, given the constant rate of change and added Gaussian noise, the concentrations can be thought as very slightly fluctuating around their initial values.

Data was generated with the following parameters: $k_1 = 0.07$, $k_2 = 0.6$, $k_3 = 0.05$, $k_4 = 0.3$, $V = 0.017$ and $K_m = 0.3$. Starting from initial values of (1,0,1,0,0,1,0) for the seven "species", 15 timepoints were generated $\{0, 1, 2, 4, 5, 7, 10, 15, 20, 30, 40, 50, 60, 80, 100\}$ producing one period. These settings were chosen to correspond with the set-up in Dondelinger et al. [11].

Other Settings The RBF kernel, equation 71, was used to fit the Gaussian process for all the Lotka-Volterra models, and the sigmoid variance kernel, equation 72, was used to fit the Gaussian process for all the protein signalling transduction pathway models. This is to correspond with simulation experiments that have been set-up in the current literature e.g. see Dondelinger et al. [11]. The initial fits from the GPs using the specified kernels were plotted against the data and showed good agreement.

In order to avoid the influence from the flattening of the signals, which was

discussed in Chapter 6, the standard deviation of the noise was held at the true value and a region of 3 standard deviations around an initial interpolant was constructed for the subsequent draws of the latent variables. This was true for every scenario of every candidate model used to compute equation 119 in this chapter. MCMC was carried out in the fashion as outlined in Chapter 6.

7.4 Results

In order to assess the performance of the new scheme outlined in Chapter 7.2, the method will be tested on two ODE systems and various candidate models of each. For comparison purposes, the results of BIC and WAIC [51] will also be provided. There are two possible ways of defining successful model selection 1. How well the results match the marginal likelihood scores computed using a method that explicitly solves the ODEs, as this corresponds to full Bayesian inference. 2. How often a method selects the model the data was simulated from. This corresponds to how well a method is able to identify a particular characteristic if indeed that characteristic does exist in the process you are observing.

The results in this chapter will be assessed mainly on the second definition, as computing the marginal likelihood scores using an explicit solution of the ODEs for all simulation set-ups lies outside the scope of this thesis. However, it was possible to compute marginal likelihood scores using an explicit

solution of the ODEs for one scenario. The marginal likelihood scores were computed using thermodynamic integration i.e. using equation 112, but instead using the likelihood obtained from the explicit solution of the ODEs. These results will be used to try to ascertain how well gradient matching is approximating the marginal likelihoods and gauge the model selection performance by the methods.

A pattern was observed whereby the performance of the computation of the Bayes factors using equation 120 sometimes deteriorated, whereas the results for $\log \mathbb{Z}(\mathbf{Y})$ using equation 119 showed an improved performance. This is discussed further on. For completeness, the results of $\log \mathbb{Z}(\mathbf{Y})$ using equation 119 are presented for all simulation scenarios.

Lotka-Volterra Original Model (LV1)

Table 5: Percentage of the time, across 10 datasets, a model was favoured by a model selection method. Data generated from LV1 model.

Method	**LV1**	**LV2**	**LV3**
Bayes factor using equation 120	**100%**	0%	0%
$\log \mathbb{Z}(\mathbf{Y})$ using equation 119	**100%**	0%	0%
BIC	**80%**	20%	0%
WAIC	**70%**	30%	0%

Table 5 contains the percentages of the time a model was favoured by a particular model selection method for when data was generated using model LV1. A graphical representation can be found in Figures 54 - 57, in the appendix. The method for computing the Bayes factors using equation 120 is excellent at selecting the true model, as it does so 100% of the time. The same conclusion can be observed by looking at the results of $\log \mathbb{Z}(\mathbf{Y})$ computed using equation 119, where 100% of the time, the true model is selected. BIC and WAIC are good at selecting the true model, which they do so 80% and 70% of the time respectively.

In order to gauge how well gradient matching is approximating the marginal likelihoods, parameter inference using an explicit solution of the ODEs was conducted for this one scenario (generating data from the LV1 model and proposing the LV1, LV2 and LV3 models as candidates). Using an explicit solution of the ODEs and computing the marginal likelihood of the data should provide a benchmark gold standard for model selection in ODEs.

Table 6: Percentage of the time, across 10 datasets, a model was favoured by a model selection method, using an explicit solution of the ODEs for parameter inference. Data generated from the LV1 model. The initial values of the system were inferred as additional parameters.

Method	**LV1**	**LV2**	**LV3**
Marginal likelihood	**60%**	0%	40%

By examining Table 6, it can be seen that the marginal likelihood scores favour the true model 60% of the time and the more complex LV3 model 40% of the time. A graphical representation can be found in Figure 52, in the appendix.

These results are in contrast to the results obtained from gradient matching. It appears as if there is a performance increase in selecting the true model, for the gradient matching method. This is counterintuitive, since gradient matching is an approximation and should be less informative, rather than more. One possibility for this outcome might be that *really* the true model should not be chosen 100% of the time, and both the LV1 and LV3 models are equally supported by the data. However, the results from the explicit solution of the ODEs are not directly comparable to that of gradient matching. This is because the explicit solution of the ODEs has additional parameters that it infers: the initial conditions. Gradient matching does not need to infer the initial conditions as it effectively profiles over them. Therefore, in order to directly compare the results, the computation of the marginal likelihood scores will be repeated, but the initial conditions will not be inferred (they will be held fixed at the true initial values).

By examining Table 7, it can be seen that now the marginal likelihood scores favour the true model 100% of the time. A graphical representation can be found in Figure 53, in the appendix.

Table 7: Percentage of the time, across 10 datasets, a model was favoured by a model selection method, using an explicit solution of the ODEs for parameter inference. Data generated from the LV1 model. The initial values of the system were held fixed at the true initial values.

Method	**LV1**	**LV2**	**LV3**
Marginal likelihood	**100%**	0%	0%

The results show that gradient matching does an excellent job of approximating the marginal likelihood of the full Bayesian inference approach (at least for when data is generated from the LV1 model and proposing the LV1, LV2 and LV3 models as candidates), as the conclusion to which model is preferred in explaining the data is exactly the same between the methods. Not only this, but they also reveal that gradient matching, an approximate method, gets a performance increase over the explicit solution (which intuitively seems like it should provide an upper-bound on the level of performance). This is because the explicit solution needs to deal with initial values (which typically are unknown in practice). Estimating these in the inference procedure introduces more uncertainty, intrinsically, when explicitly solving the ODEs. Gradient matching does not deal with initial conditions. Hence, marginal likelihood estimation using gradient matching appears to be equivalent to marginal likelihood estimation using an explicit solution of the ODEs when the initial parameters are known, and therefore it has an advantage. Immediate future work will focus on seeing how consistent this is across different

ODE structure scenarios.

Lotka-Volterra Intra-Species Competition Model (LV2)

Table 8: Percentage of the time, across 10 datasets, a model was favoured by a model selection method. Data generated from LV2 model.

Method	**LV1**	**LV2**	**LV3**
Bayes factor computed using equation 120	**80%**	10%	10%
$\log \mathbb{Z}(\mathbf{Y})$ using equation 119	**100%**	0%	0%
BIC	**90%**	0%	10%
WAIC	40%	**60%**	0%

Table 8 contains the percentages of the time a model was favoured by a particular model selection method for when data was generated using model LV2. A graphical representation can be found in Figures 58 - 61, in the appendix. The new method for computing the Bayes factors using equation 120 do a poor job of selecting the true model, as 80% of the time the new method selects the LV1 model even though the data were generated from the LV2 model. Likewise the results from $\log \mathbb{Z}(\mathbf{Y})$ using equation 119 also show a similar result, as 100% of time it favours the LV1 model above the (true) LV2 model. BIC does a poor job of selecting the true model, as 0% of the

time it favours the true model. WAIC selects the true model 60% of the time.

At first glance, it would appear that model selection using the Bayes factors calculated using equation 120, $\log \mathbb{Z}(\mathbf{Y})$ using equation 119 and BIC (and arguably WAIC) are not able to select the true model for this scenario, since although the data were generated from the LV2 model, the methods favour the LV1 model. However, an inspection of the structure of LV2 can help clarify things. When θ_5 is large, the component will decrease x_1. However, θ_5 could be set to zero and θ_2 could made large and again x_1 would decrease. Hence, the LV1 model has a term that is able affect the signals in a way very much the same as the LV2 model, without the need for an extra parameter. This essentially makes the intra-species component weakly identifiable. In order to test whether this is the case and to see whether the model selection methods are able to identify the true model, the dependency of the system on θ_5 needs to be more substantial. To this end, data was generated with following parameters; $\theta_1 = 100$, $\theta_2 = 0.1$, $\theta_3 = 4$, $\theta_4 = 0.1$ and $\theta_5 = 10$. The effect this has on the system is that for x_1, this "species'" concentration rises exponentially and then plateaus, since the intra-species competition term stops the population increasing without end. The LV1 model should not be able to replicate this because concentrations for x_2 go to zero. Hence, the LV1 model should not have a way to regulate the population concentration and get good agreement with the data. For this set-up, iid Gaussian noise of SD = 0.1414 was added to each "species" (average SNR for each "species = 10).

The priors over the parameters were $\Gamma(4, 0.5)$ prior for θ_2, θ_3 and θ_4, $U(0, 110)$ for θ_1 and a $U(2, 11)$ for θ_5 (reflecting the extra uncertainty of these two parameters).

Table 9: Percentage of the time, across 10 datasets, a model was favoured by a model selection method. Data generated from LV2 with parameter settings chosen to make the intra-species component effect more substantial.

Method	**LV1**	**LV2**	**LV3**
Bayes factor computed using equation 120	30%	**70%**	0%
$\log \mathbb{Z}(\mathbf{Y})$ using equation 119	30%	**70%**	0%
BIC	**100%**	0%	0%
WAIC	**50%**	30%	20%

By examining Table 9, it is possible to observe the model selection performance when parameter settings are chosen in order to make the intra-species component effect in the LV2 model more substantial. A graphical representation can be found in Figures 62 - 65, in the appendix. Now, the new computation of the Bayes factors using equation 120 and $\log \mathbb{Z}(\mathbf{Y})$ using equation 119 select the true model 70% of the time. This is a substantial difference from before and indicates the reason the methods were unable to select the true model previously is because the intra-species component

term was effectively unidentifiable (for the particular choice of the parameter settings). It is worth noting that the parameters now chosen to make the intra-species component effect more substantial were rather arbitrary and it is likely that choosing other parameters that more clearly pronounce the effect of the intra-species term on the process, might lead to an increase in the percentage of the time the true model is selected by the new methods. BIC is unable to identify the true model for any dataset. WAIC selects the true model 30% of the time.

Lotka-Volterra Saturation Term Model (LV3)

Table 10: Percentage of the time, across 10 datasets, a model was favoured by a model selection method. Data generated from LV3 model.

Method	**LV1**	**LV2**	**LV3**
Bayes factor computed using equation 120	0%	0%	**100%**
$\log \mathbb{Z}(\mathbf{Y})$ using equation 119	0%	0%	**100%**
BIC	0%	0%	**100%**
WAIC	0%	0%	**100%**

Table 10 contains the percentages of the time a model was favoured by a particular model selection method for when data was generated using model

LV3. A graphical representation can be found in Figures 66 - 69, in the appendix. The new method for computing the Bayes factors using equation 120, $\log \mathbb{Z}(\mathbf{Y})$ using equation 119, BIC and WAIC are excellent at selecting the true model, as 100% of the time the methods select the true model.

Protein Signalling Transduction Pathway Model 1 (PSTP1)

Table 11: Percentage of the time, across 10 datasets, a model was favoured by a model selection method. Data generated from PSTP1 model.

Method	**PSTP1**	**PSTP2**	**PSTP3**	**PSTP4**
Bayes factor computed using equation 120	**70%**	0%	0%	30%
$\log \mathbb{Z}(\mathbf{Y})$ using equation 119	**100%**	0%	0%	00%
BIC	30%	0%	**60%**	10%
WAIC	40%	0%	0%	**60%**

Table 11 contains the percentages of the time a model was favoured by a particular model selection method for when data was generated using model PSTP1. A graphical representation can be found in Figures 70 - 73, in the appendix. The new method for computing the Bayes factors using equation 120 is good at selecting the true model. However, 30% of the time, the method favours the more complex model instead. $\log \mathbb{Z}(\mathbf{Y})$ using equation

119 on the other hand, selects the true model 100% of the time. BIC does a poor job of selecting the true model, where the majority of the time it favours the least complex model (PSTP3) and the true model 30% of the time. WAIC is also poor at selecting the true model, where the majority of the time it favours the most complex model (PSTP4) and the true model 40% of the time.

Based on these results, it is clear that for this case the approximation for C was deteriorating the new method's ability to select the true model. The values computed for log (C) under model PSTP4 were large and negative. This is because due to the increased complexity of the model, it can support many instances where the gradients from the ODEs match those from the interpolant. This causes the distribution for C under this model (PSTP4) to be very diffused, which in turn means that any subsequent draws from this distribution will have a very low probability density associated with them. Taking the logarithm of a small value ($\ll 0.001$) will result in a large negative value for log (C). Since the marginal likelihood in equation 120 is calculated by $\log \mathbb{Z}(\mathbf{Y}) - \log(C)$, this increases the marginal likelihood substantially. Bayes factors, however, are consistent estimators and therefore the likelihood term ($\mathbb{Z}(\mathbf{Y})$) should compensate for this occurrence. However, the usual experiments conducted in current systems biology typically yield a small number of observations. This in turn reduces the effect that the likelihood term has, in the calculation of the marginal likelihood. For some of the

scenarios that have been examined the normalisation term (C) has been subject to substantial numerical noise and therefore has been having a negative effect on the results of the model selection. Using $\log \mathbb{Z}(\mathbf{Y})$ computed using equation 119 instead of the log marginal likelihood (equation 120) improves the ability to select the true model.

7.5 Conclusions

The proposed method of calculating Bayes factors via equation 120 provides an accurate way of performing model selection in ODEs using gradient matching and thermodynamic integration, when the criterion for good performance is how often the true model is selected. For this criterion, the method outperforms BIC and WAIC over all scenarios examined, apart from when simulating data from the LV2 model with parameters; $\theta_1 = 4$, $\theta_2 = 1$, $\theta_3 = 4$, $\theta_4 = 2$, $\theta_5 = 5$. For this scenario, however, it was demonstrated that the particular parameters chosen created created weak identifiability between the true model and the less complex model. When other parameters were chosen, in order to allow the intra-species term to have a more substantial role in governing the process, the newly proposed method outperforms both BIC and WAIC in selecting the true model.

BIC was able to correctly identify the true model when data was generated from the LV1 model and the LV3 model only. BIC is asymptotically equivalent to the log marginal likelihood, but the typical data size for the

experiments in this particular area of systems biology is small (sample size 11 for the Lotka-Volterra models and a sample size of 15 for the protein signalling transduction pathway model). It is unlikely that the asymptotic assumptions of BIC have been satisfied for these scenarios.

WAIC was able to identify the true model when data was generated from the LV1 model, the LV2 model (with parameters $\theta_1 = 4$, $\theta_2 = 1$, $\theta_3 = 4$, $\theta_4 = 2$, $\theta_5 = 5$) and the LV3 model. However, when the parameters were chosen in order to pronounce the effect the intra-species term had on the system, WAIC was unable to correctly identify the true model. It was also unable to identify the true model for the protein signalling transduction pathway example. As with the case of BIC, WAIC also relies on asymptotics. It is likely that due to the sparse dataset sizes, the asymptotic properties have not been satisfied and this is deteriorating the performance of the method.

A discovery was made as to whether both terms in equation 120 should be used for the model selection. For the examples looked at throughout this chapter, only considering $\log \mathbb{Z}(\mathbf{Y})$ in equation 120) leads to selecting the true model as often as using the log marginal likelihood and in some cases more often. The only example where it was slightly poorer at selecting the true model than the log marginal likelihood, was when data was generated from the LV2 model with parameters $\theta_1 = 4$, $\theta_2 = 1$, $\theta_3 = 4$, $\theta_4 = 2$, $\theta_5 = 5$ (where the log marginal likelihood selects the true model 10% of the time and

log $\mathbb{Z}(\mathbf{Y})$ in equation 120) selects the true model 0% of the time). However, as detailed previously, this was due to a poor choice of parameter settings that ended up masking the effect the intra-species term had on the system. When the effect of this term is pronounced, log $\mathbb{Z}(\mathbf{Y})$ in equation 120) performs as well as the log marginal likelihood in selecting the true model. Hence, for the examples looked at, log $\mathbb{Z}(\mathbf{Y})$ computed using equation 119 outperforms all the other methods at selecting the true model. Future research should focus on whether this is a general trend and whether the normalisation term (C) in equation 120) can be dispensed with entirely.

Finally, the conclusions made so far are based on how often a model selection method identifies the true model. However, if another model is as adequate (or better) at explaining the data, then selecting the true model would not necessarily be a success. A benchmark for which models are consistent with data is necessary to judge this performance. This benchmark is the marginal likelihood computed using full Bayesian inference. Computing the marginal likelihood with gradient matching does not provide a benchmark, since gradient matching is an approximate method and therefore the marginal likelihood will approximate the true marginal likelihood. In order to see how good of an approximation gradient matching achieved, parameter inference by explicitly solving the ODEs was carried out and the marginal likelihoods were computed. The results show that gradient matching is able to exactly match the conclusion of the model selection using the marginal

likelihoods computed using an explicit solution of the ODEs, when the initial conditions of the system is known. In fact, gradient matching demonstrates a performance increase, as it does not require any initial conditions. Future work will investigate whether this is consistent across different ODE structure scenarios.

8 Discussion

The elucidation of the structure and dynamics of biopathways is a central objective of current systems biology research. A standard approach is to view a biopathway as a network of biochemical reactions, which is modelled as a system of ordinary differential equations (ODEs).

Conventional inference methods typically involve numerically integrating the system of ODEs to produce a signal. This signal is then compared to the data by some appropriate metric defined by the chosen noise model, enabling the calculation of a likelihood. This process is repeated, as part of either an iterative optimisation scheme or sampling procedure in order to estimate the parameters. However, this is onerous as the computational costs of repeatedly solving the solving the ODEs are usually high.

Aimed at reducing the computational complexity, new concepts based on gradient matching were developed. In a preliminary smoothing step, the data are interpolated; then in a second step the parameters of the ODEs are optimised or sampled so as to minimise the difference between the derivatives of the ODEs and the slopes of the tangents to the interpolant. In this way, the ODEs never have to be solved explicitly. A drawback to this approach however, is that the performance critically depends on the quality of the initial interpolant. A better approach is to have the ODEs regularise the

interpolant themselves. Not only does this have the benefit of avoiding being solely reliant on the initial interpolant, it also allows the ODE structure itself to affect the modelling of the system.

The work in Chapter 4 involved developing a new gradient matching method that combined the methodological approach of adaptive gradient matching using Gaussian processes (GPs) from Dondelinger et al. [11] with a parallel tempering scheme of the gradient mismatch parameter from Campbell and Steele [9]. The rationale behind this new approach is that if the ODEs provide a correct mathematical description of the system, there should be no difference between the gradients of the interpolant and those predicted from the ODEs. However, in practice, forcing the gradients to be equal is likely to cause parameter inference methods to converge to a local optimum of the likelihood. Forcing the gradients to immediately be the same would restrict the inference procedure to a section of the likelihood corresponding to parameters that perfectly agree with the gradient match. However, there is no guarantee that these parameters are suitable for the data, see Campbell and Steele [9] for details. A parallel tempering scheme is the natural way to deal with such local optima, as opposed to inferring the degree of mismatch, since different tempering levels correspond to different strengths of penalising the mismatch between the gradients.

A comparison between the contrasting approaches of posterior inference and

parallel tempering of the gradient mismatch parameter was carried out on two ODE models - the Fitz-Hugh Nagumo (FhN) system (equations 40-41) and the Lotka-Volterra (LV) system (equation 42). There was no significant difference between the posterior medians as estimators of the true parameters, for both ODE models and all observational noise levels. Both methods outperformed a related method by Calderhead et al. [8], in terms of the posterior medians estimating the true parameters.

When the full posterior distributions were compared, the new method outperformed the method by Dondelinger et al. [11] (denoted INF) for the first parameter of the Fitz-Hugh Nagumo system, when the observational noise level was 0.5 (signal to noise ratio of approximately 10). This was true for both parameter schedules of the new method - denoted LB2 and LB10. For the same noise level, the INF method produced an unbiased posterior distribution for the third parameter of the FhN system, whilst the LB2 and LB10 methods produced about a third of the variance than the INF method. All methods performed similarly in inferring the second parameter. The INF, LB2 and LB10 methods all perform similarly to one another, overall, across the other noise levels.

For observational noise level 0.5 (signal to noise ratio of approximately 10), the LB2 and LB10 methods outperform the Calderhead et al. [8] and INF methods for the first parameter in the Lotka-Volterra system, in terms of

having interquartile ranges that contain the true parameter. For parameters three and four, the methods perform similarly to one another. The methods have different variance/bias tradeoffs for the second parameter, and it is unclear whether a particular method is performing better than another. For the other noise levels, overall, there does not appear to be a difference between the methods.

It is important to note that due to an instability in the probability model, flattening of the concentration profiles occurs and long tails appear in the distributions. A solution to this is identified and is later discussed.

Gradient matching does not perform full Bayesian inference, as it is an approximation to conducting parameter inference using an explicit solution of the ODEs. Hence, in order to understand just how well the new gradient matching method approximates the full Bayesian inference approach, it is necessary to compare the results directly to results obtained by explicitly solving the ODEs. In a comparison on the Lotka-Volterra system, with observational noise level 0.5, the posterior samples of the explicit solution and the new gradient matching method are pretty similar. The root mean square values in function space show that the explicit solution is performing better than gradient matching (as would be expected), since the distribution is lower for the explicit solution. However, there is reasonable overall agreement between the distributions, suggesting that the approximation produced by

the new gradient matching method is not far away from the truth.

A wide scale comparative evaluation of the new method from Chapter 4 with various state-of-the-art gradient matching methods was carried out in Chapter 5. The methods are based on different inference approaches and statistical models, namely: non-parametric Bayesian statistics using Gaussian processes (INF - Dondelinger et al. [11], the new method from Chapter 4 - LB2 and LB10), splines-based smooth functional tempering (Campbell and Steele [9] - C&S), hierarchical regularisation using splines interpolation (Ramsay et al. [40] - RAM), and penalised likelihood based on reproducing kernel Hilbert spaces (González et al. [19] - GON). The set-ups have also allowed for the comparison of opposing paradigms of Bayesian inference (INF) versus parallel tempering (LB2, LB10) of the slack parameters controlling the amount of mismatch between the gradients.

The INF method by Dondelinger et al. [11] peforms well across the various set-ups as it consistently produces estimates that are close to the true parameters. The method typically produces biased estimates, which is offset for a reduction in uncertainty. The new method proposed in Chapter 5 is unbiased, producing a slightly larger variance in the parameter estimates than the INF method. The results for LB2 and LB10 are accurate and consistent across ODE models and experimental set-up.

An instability in the methods of INF, LB2 and LB10 was observed, where the time course signals were prone to flattening. This had the effect of deteriorating the performance of the methods. In order to address this issue, the standard deviation of the observation noise was held at the true value. The results are noticeably different than when the noise is inferred, indicating that flattening does have a substantial effect on the performance of the INF, LB2 and LB10 methods. Holding the standard deviation of the noise at the true value, stops the likelihood term in the model from becoming too weak and the empirical findings suggest that the flattening can be avoided in this way. In practice, the observational noise could be estimated (for example, using a standard GP regression), before conducting parameter inference using these methods. Speculating, it seems reasonable that this approach would be robust, as GP regression typically performs well so long as the GP kernel is able to model the underlying smoothness assumptions of the system. This fix would still be somewhat heuristic and future research should also involve looking to see whether a more general robust solution can be found.

The method by Campbell and Steel [9] performed well in one scenario, for the Fitz-Hugh Nagumo system. In this example, however, the dataset size was large, much larger than the dataset size typical in these experiments. It would also be time-consuming in practice to finely adjust the tuning parameters. The method's performance is critically dependent on these parameters, which can observed in the other examples on the same dataset size, where the

results deteriorate substantially. Also, the optimal results for this method were obtained using different setting for the tuning parameters than in the original publication of Campbell and Steele [9]. When the size of the dataset was reduced to something more consistent with what would be obtained in practice, none of the tested settings were able to achieve a reasonable performance.

The empirical findings show that the GON and GON Cross methods are robust, consistently estimating parameters close to the true parameters, in terms of the absolute error. Using cross validation rather than AIC to infer the penalty parameter was found to improve the robustness of the González et al. [19] method. When generating data from the Fitz-Hugh Nagumo system, with a signal to noise ratio of roughly 10 and 25 datapoints, the GON and GON Cross methods are better at inferring the 3^{rd} parameter than INF, LB2 and LB10, and when generating data from the protein signalling transduction pathway. In the study using the protein signalling transduction pathway equations, the GON Cross method (GON method was unable to optimise) was outperformed by INF, LB2 and LB10. This reflects that the approximation of GON and GON Cross, made in equation 34, is more suitable for some systems than others. The uncertainty quantification for GON and GON Cross has been obtained by examining the distribution of point estimates over multiple simulated datasets. These methods are unable produce confidence intervals and so in practice, one would need to rely on

other implementations, such as bootstrapping, to quantify the uncertainty in the parameter estimates. The knock on effect of this is something that is currently unknown and the relationship that this has on the accuracy and computational times of the methods needs to be examined. The INF, LB2 and LB10 methods use a Bayesian framework and therefore uncertainty quantification is obtained directly from the MCMC samples.

The method by Ramsay et al. [40] was examined using the Fitz-Hugh Nagumo system. It was outperformed by the GON, INF, LB2 and LB10 methods, for each parameter of the ODE model.

There was little prior knowledge as to optimal parameter schedules for the gradient mismatch parameter for the newly proposed method in Chapter 4, and so two scheduling ladders were considered: in log_2 increments (referred throughout as LB2) and log_{10} increments (referred throughout as LB10). It was found that the methods have proven to be quite robust with respect to the scheduling, however, it would be reasonable to expect the performance of the method to improve once an optimal way of specifying the increments is created. This too, should be the focus of future work.

In Chapter 6, the notion of representing gradient matching as a probabilistic generative model was investigated. In a publication by Wang and Barber [49] a gradient matching method (herein denoted as GPODE) was devel-

oped and it was asserted to outperform state-of-the-art Gaussian process gradient matching methods by: having a simplified mathematical description, constituting a probabilistic generative model and gaining an improvement in the accuracy of parameter estimation. However, as demonstrated by the work presented in Chapter 6, the mathematical simplification of the GPODE model was a consequence of confusing the marginalisation over a random variable with its elimination from the model. When the GPODE is properly represented, it is shown to have a more complex form. In order to consistently represent gradient matching as a probabilistic generative model, one needs to make independence assumptions that are implausible, were not made clear in the original publication of Wang and Barber [49] and have non-negligible repercussions. As a result of the independence assumptions of the GPODE model, the method has problems of identifiability of the ODE parameters when the data is systematically missing. This means that the method substantially struggles in inferring partially observable systems. This issue does not exist when gradient matching with Gaussian processes is followed with a product of experts approach, as with Calderhead et al. [8] and Dondelinger et al. [11] (referred to as AGM in Chapter 6).

A simulation study was carried out and the results did not agree with the assertion that GPODE was able to outperform AGM. On the Fitz-Hugh Nagumo system, both methods performed on par, producing different bias/variance characteristics. On a system with missing values and on the

Lotka-Volterra system, AGM demonstrated a substantially better performance than that of GPODE. The study in Chapter 6 shows that for practical applications, AGM is to be preferred over GPODE.

The methodological approximation that gradient matching makes to avoid explicitly solving the system of ODEs causes gradient matching not to be able to be consistently represented as a probabilistic generative model. This is due to gradient matching conceptually violating the DAG constraint.

Finally, Chapter 7 discusses the idea of performing model selection for ODEs using gradient matching. The work presented details a new method for computing the marginal likelihood, by combining gradient matching and thermodynamic integration. Since gradient matching is an approximate method, the marginal likelihood computed using the new method is not the true marginal likelihood, but an approximation also.

There is more than one way to judge model selection performance and two criteria are considered throughout this thesis. The first, is to what extent any model selection method is able to reproduce the results obtained by using the true marginal likelihood. The true marginal likelihood shows how consistent a model is with the data. The second, is to what extent a method selects the true model that the data was generated from. This corresponds to the extent a method is able to identify a particular characteristic when that

characteristic exists in the process being observed. It is the latter criterion that was mainly used to judge model selection performance in this thesis, as computing the true marginal likelihood requires explicitly solving the differential equations. To do this repeatedly for all the simulation scenarios that were examined in Chapter 7 was considered too cumbersome and hence, it lies outside the scope of this thesis. It was, however, possible to do explicitly solve the ODEs for one scenario and the corresponding marginal likelihoods will be used as a benchmark for model selection performance.

The new method proposed in Chapter 7 is able to consistently and with high accuracy select the true model. It outperforms BIC and WAIC over all scenarios that were examined, apart from one example where it was shown that the chosen ODE model parameters created weak identifiability. This study was repeated, but using parameter choices that avoided the identifiability in the ODE system. In this example, the new method is again able to outperform both BIC and WAIC in selecting the true model. BIC was only able to select the true model in two scenarios: when data was generated from the LV1 model and from the LV3 model. WAIC was able to identify the true model when data was generated from the LV1 model and the LV2 model, when the parameters were not chosen specifically to avoid the identifiability in the LV2 model. In the example where the parameters were chosen to avoid the identifiability in the LV2 model, WAIC was unable to select the true model. Both BIC and WAIC rely on asymptotics and it is unlikely that for the dataset

sizes used in the experiments (which are typical of the dataset sizes in this area of research) that the assumptions of the methods have been met. This is likely to be the reason for the poor performance in selecting the true model.

In the benchmark study, where the true marginal likelihood values were computed using an explicit solution of the ODEs and fixed initial conditions, the new method, that combines gradient matching and thermodynamic integration, is able to exactly match which model is preferred by the data. In fact, rather surprisingly, a performance increase is obtained by gradient matching over the explicit method. The empirical findings show that gradient matching performs at the same level as when using the true marginal likelihood and known (fixed) initial conditions to infer the correct model. However, gradient matching does not require any initial conditions and in practice these are often unknown quantities that need to be inferred. This finding, however, has only been observed for one ODE set-up. The natural next step would be to investigate how robust this conclusion is.

9 Appendix

Derivation of equation 12. Golub et al. [18] present generalised cross-validation in the form

$$F(\lambda) = \frac{\frac{1}{T}||(\mathbf{I} - \mathbf{A}(\lambda))\mathbf{y}||^2}{\left[\frac{1}{T}\mathrm{Tr}(\mathbf{I} - \mathbf{A}(\lambda))\right]^2}, \tag{125}$$

where T are the number of timepoints, $\mathbf{I}$ is the identity matrix, $\mathbf{y}$ is the vector of data points, Tr is the trace and $\mathbf{A}(\lambda) = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X} + T\lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}$. Hence,

$$\begin{aligned}
F(\lambda) &= \frac{\frac{1}{T}||(\mathbf{I} - \mathbf{A}(\lambda))\mathbf{y}||^2}{\left[\frac{1}{T}\mathrm{Tr}(\mathbf{I} - \mathbf{A}(\lambda))\right]^2} \\
&= \frac{\frac{1}{T}||\mathbf{y} - \mathbf{A}(\lambda)\mathbf{y}||^2}{\frac{1}{T^2}\left[\mathrm{Tr}(\mathbf{I} - \mathbf{A}(\lambda))\right]^2} \\
&= \frac{\frac{1}{T}||\mathbf{y} - \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X} + T\lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}||^2}{\frac{1}{T^2}\left[\mathrm{Tr}(\mathbf{I} - \mathbf{A}(\lambda))\right]^2} \\
&= \frac{T||\mathbf{y} - \hat{\mathbf{x}}||^2}{\left[\mathrm{Tr}(\mathbf{I} - \mathbf{A}(\lambda))\right]^2} \\
&= \frac{T||\mathbf{y} - \hat{\mathbf{x}}||^2}{\left[\mathrm{Tr}(\mathbf{I}) - \mathrm{Tr}(\mathbf{A}(\lambda))\right]^2}.
\end{aligned} \tag{126}$$

Since, $\mathbf{A}(\lambda)\mathbf{y} = \hat{\mathbf{x}}$, then $\mathbf{A}(\lambda) = \frac{d\hat{\mathbf{x}}}{d\mathbf{y}}$. Hence,

$$F(\lambda) = \frac{T||\mathbf{y} - \hat{\mathbf{x}}||^2}{\left[T - \mathrm{Tr}\left(\frac{d\hat{\mathbf{x}}}{d\mathbf{y}}\right)\right]^2}$$
$$= \frac{T||\mathbf{y} - \hat{\mathbf{x}}||^2}{\left[T - \sum_{t=1}^{T} \frac{d\hat{x}(t)}{dy(t)}\right]^2}. \tag{127}$$

Doing this for all species yields,

$$F(\boldsymbol{\lambda}) = \frac{\sum_{s=1}^{n} T_s||\mathbf{y}_s - \hat{\mathbf{x}}_s||^2}{\left[\sum_{s=1}^{n}\left\{T - \sum_{t=1}^{T} \frac{d\hat{x}(t)}{dy(t)}\right\}\right]^2}, \tag{128}$$

where the form is now the same as in equation 12. This is true because $\mathbf{x}$ in equation 12 is the interpolant and so $\hat{\mathbf{x}} = \mathbf{x}$ and the extra factor of T_s in the numerator allows the method to weight different subsamples (which the formula from Ramsay et al. [40] assumes to be equal).

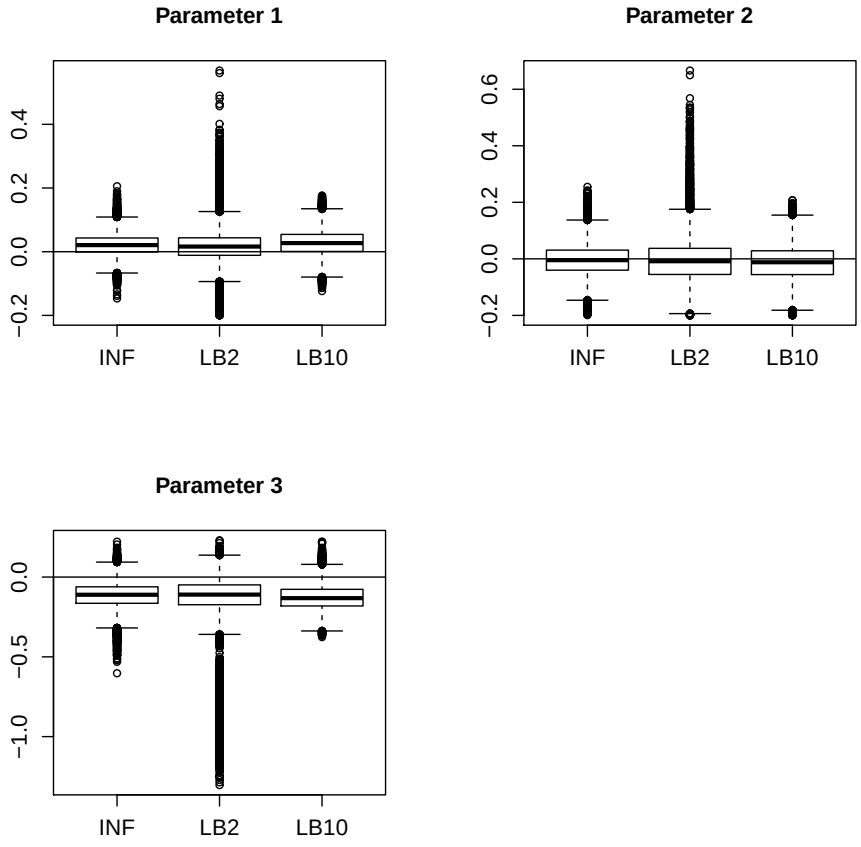


Figure 44: Posterior distributions over 10 datasets for the ODE parameters from the Fitz-Hugh Nagumo system, equations 40-41. The true parameters have been subtracted from the posterior distributions and the horizontal line shows zero difference to the true parameters. The observational noise level is 0 for this scenario.

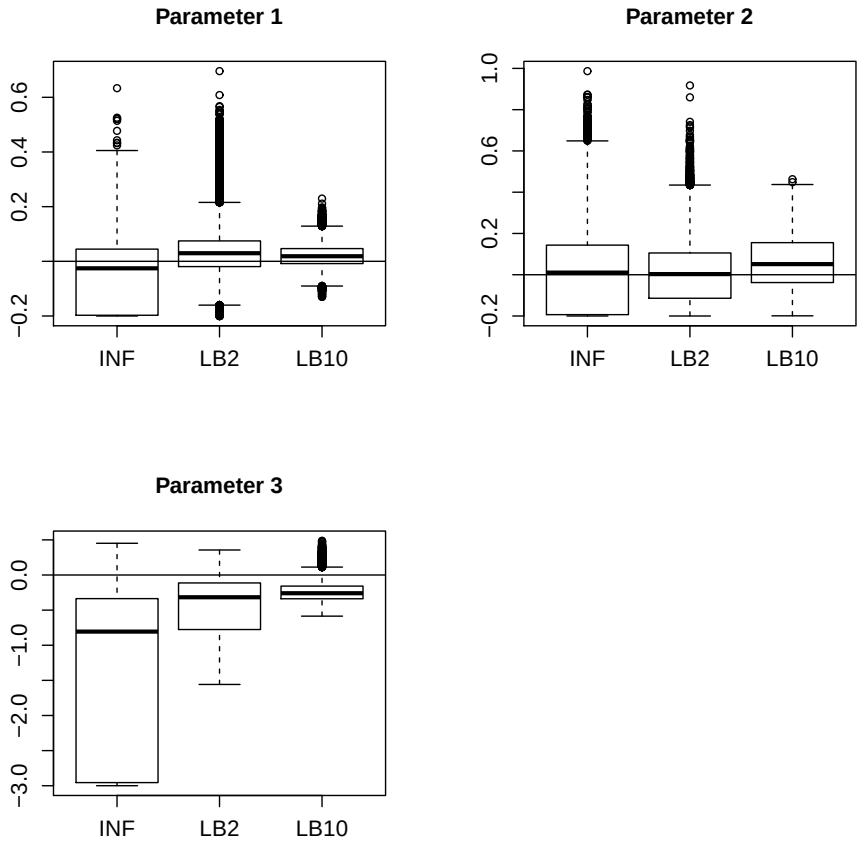


Figure 45: Posterior distributions over 10 datasets for the ODE parameters from the Fitz-Hugh Nagumo system, equations 40-41. The true parameters have been subtracted from the posterior distributions and the horizontal line shows zero difference to the true parameters. The observational noise level is 0.1 for this scenario.

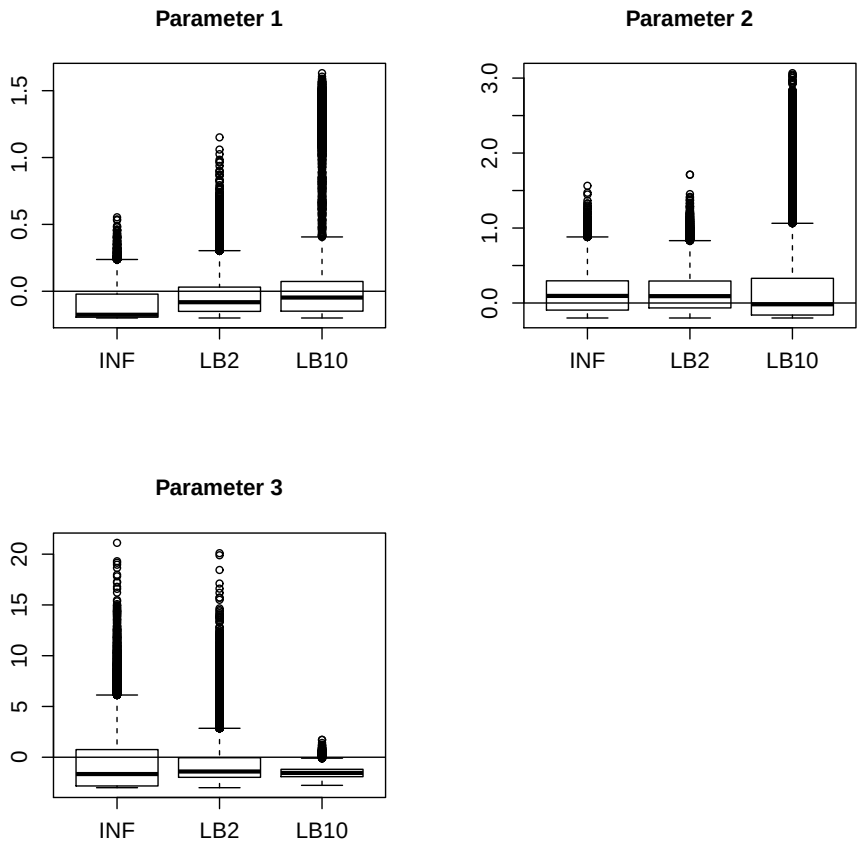


Figure 46: Posterior distributions over 10 datasets for the ODE parameters from the Fitz-Hugh Nagumo system, equations 40-41. The true parameters have been subtracted from the posterior distributions and the horizontal line shows zero difference to the true parameters. The observational noise level is 0.8 for this scenario.

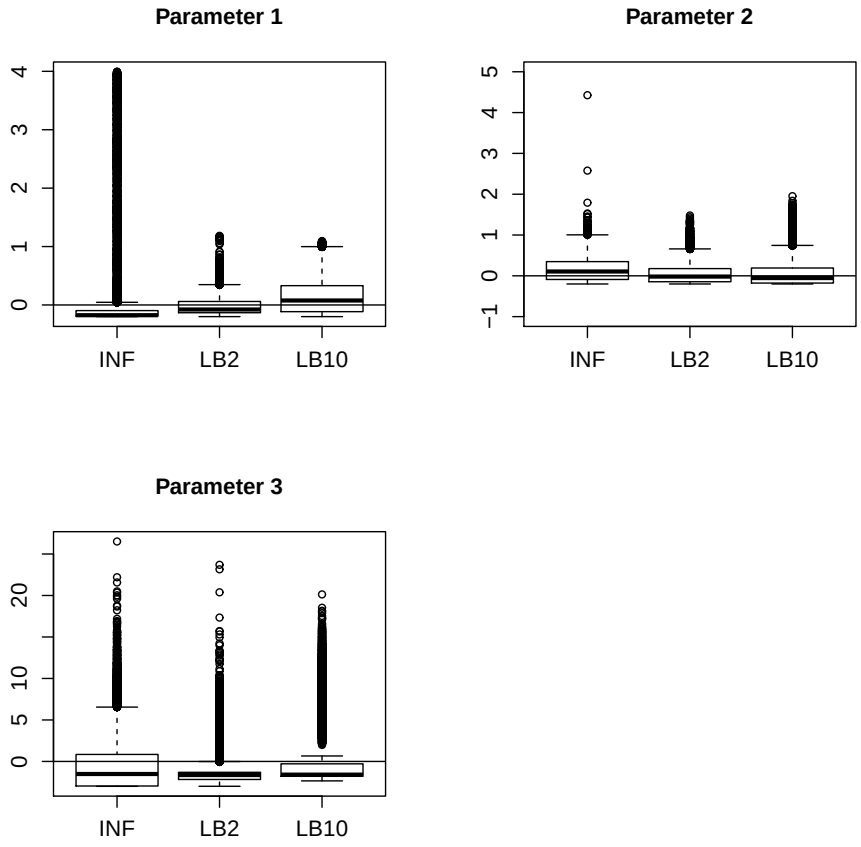


Figure 47: Posterior distributions over 10 datasets for the ODE parameters from the Fitz-Hugh Nagumo system, equations 40-41. The true parameters have been subtracted from the posterior distributions and the horizontal line shows zero difference to the true parameters. The observational noise level is 1 for this scenario. Note that for parameter 2, a long tail was removed from the INF results, for scalability purposes.

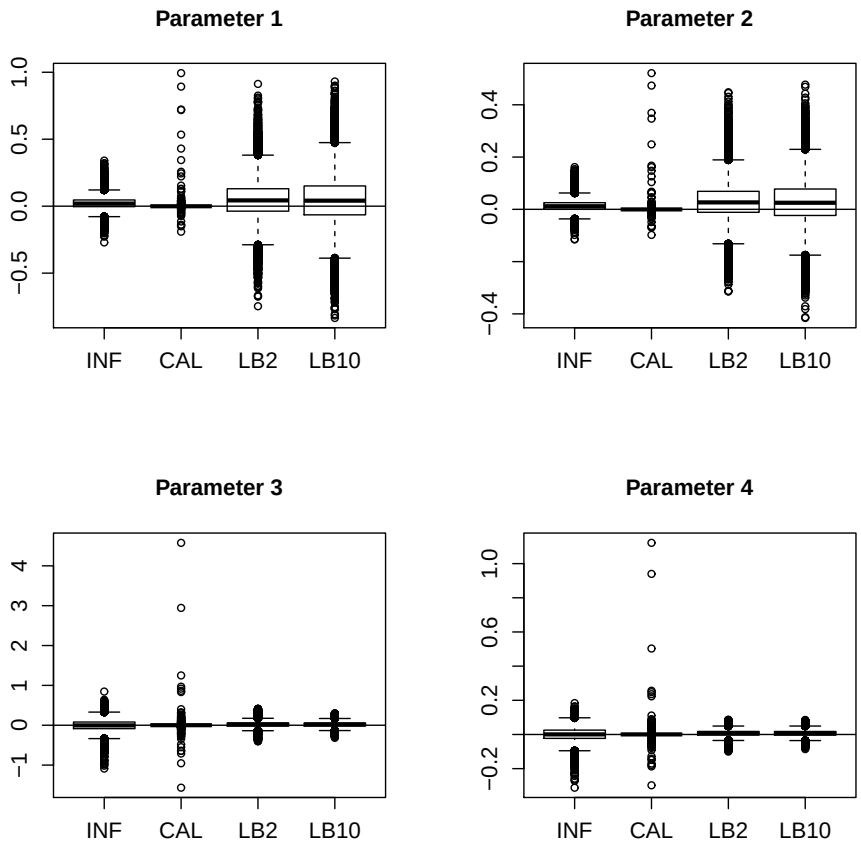


Figure 48: Posterior distributions over 10 datasets for the ODE parameters from the Lotka-Volterra system, equation 42. The true parameters have been subtracted from the posterior distributions and the horizontal line shows zero difference to the true parameters. The observational noise level is 0 for this scenario.

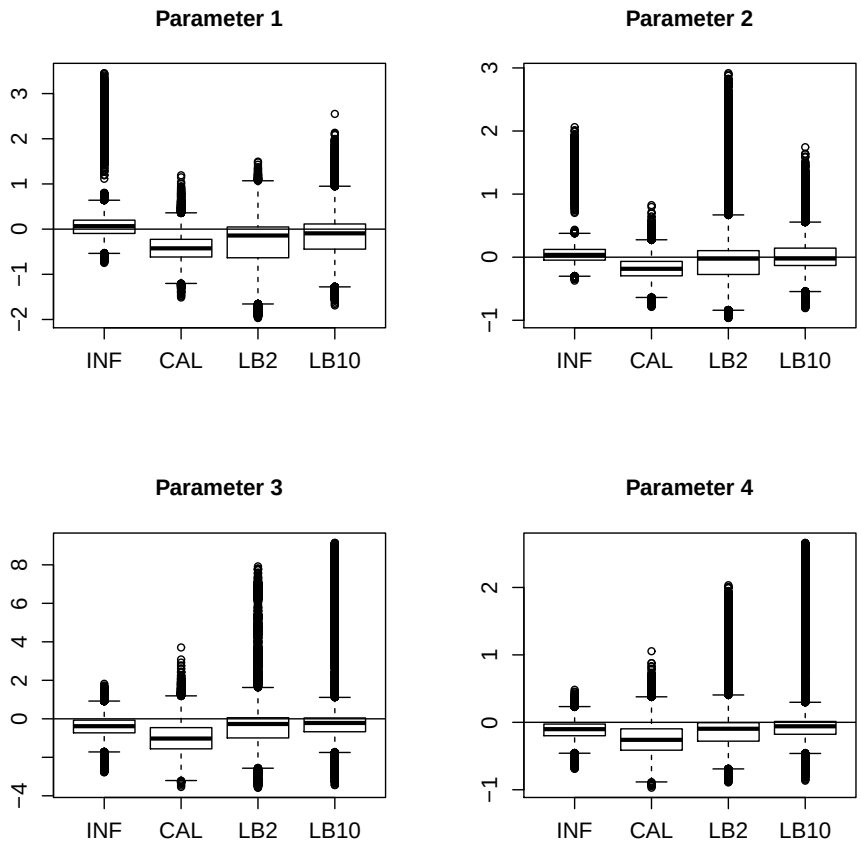


Figure 49: Posterior distributions over 10 datasets for the ODE parameters from the Lotka-Volterra system, equation 42. The true parameters have been subtracted from the posterior distributions and the horizontal line shows zero difference to the true parameters. The observational noise level is 0.1 for this scenario.

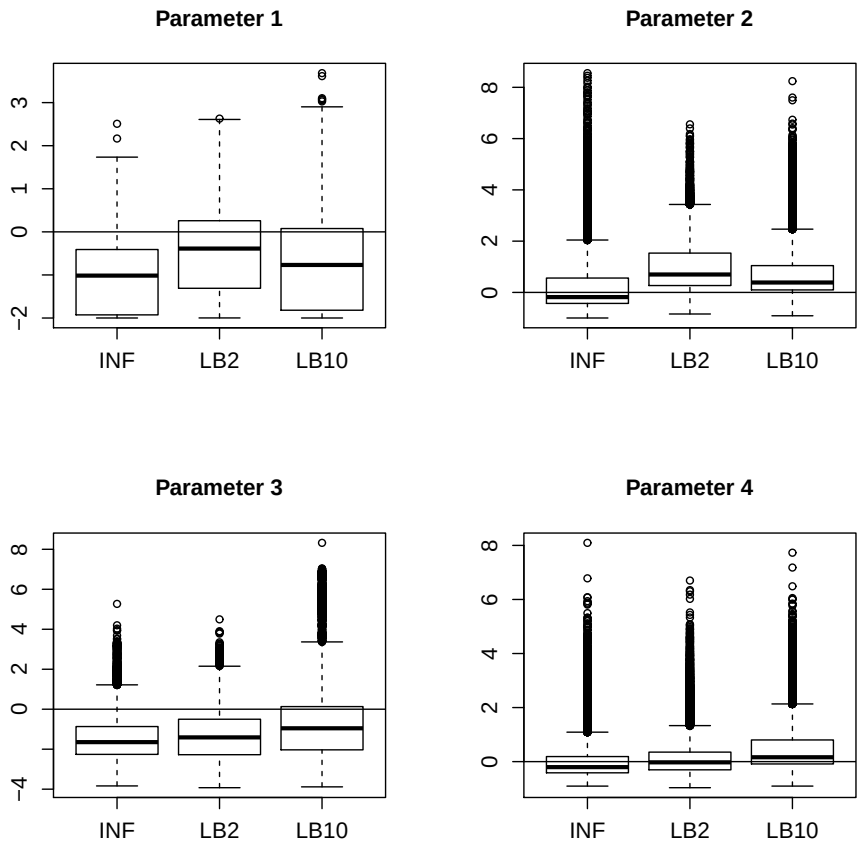


Figure 50: Posterior distributions over 10 datasets for the ODE parameters from the Lotka-Volterra system, equation 42. The true parameters have been subtracted from the posterior distributions and the horizontal line shows zero difference to the true parameters. The observational noise level is 0.8 for this scenario.

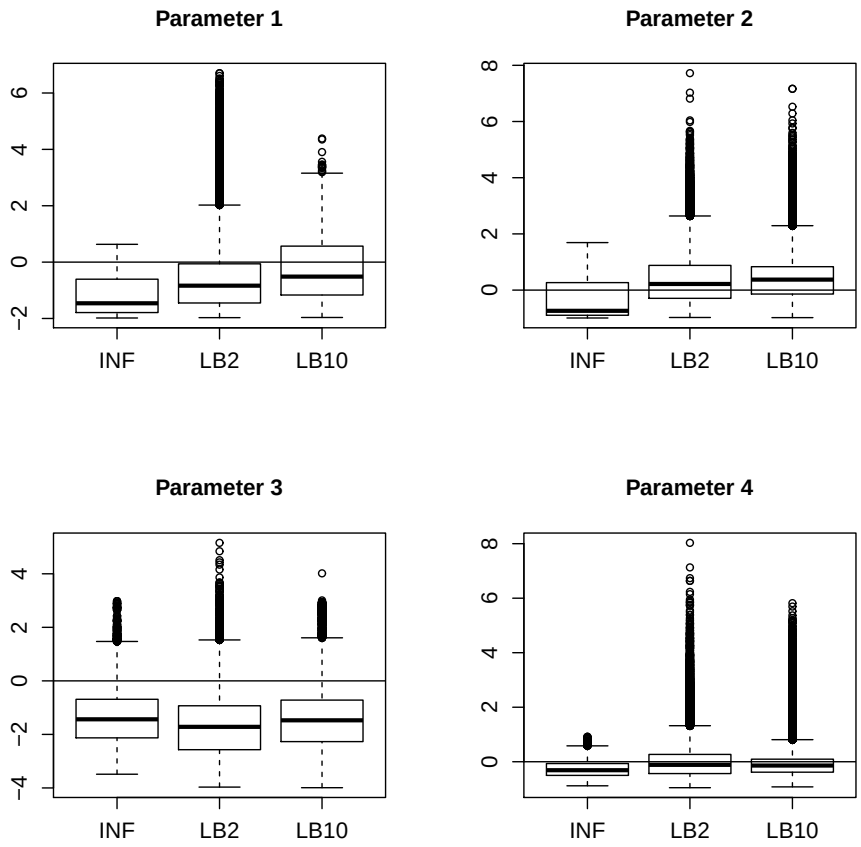


Figure 51: Posterior distributions over 10 datasets for the ODE parameters from the Lotka-Volterra system, equation 42. The true parameters have been subtracted from the posterior distributions and the horizontal line shows zero difference to the true parameters. The observational noise level is 1 for this scenario.

Table 12: Computational times for INF and a method that numerically integrates the ODEs for the protein signalling transduction pathway in equations 45. Table constructed from the boxplots in [11]. The LB2 and LB10 methods were equivalent to INF. The median time is presented alongside the interquartile range (IQR).

Method	**Time for $1 * 10^5$ MCMC steps (seconds)**
INF	2500 (Median) [2400 , 2600] (IQR)
Numerical Integration	12500 (Median) [12000 , 13000] (IQR)

Table 13: Number of steps until convergence for INF and a method that numerically integrates the ODEs for the protein signalling transduction pathway in equations 45. Table constructed from the boxplots in [11]. The LB2 and LB10 methods were equivalent to INF. The median number is presented alongside the interquartile range (IQR).

Method	**Number of steps until convergence**
INF	$3.5 * 10^4$ (Median) $[3.25 * 10^4 , 4.5 * 10^4]$ (IQR)
Numerical Integration	$7.9 * 10^4$ (Median) $[7.5 * 10^4 , 8.25 * 10^4]$ (IQR)

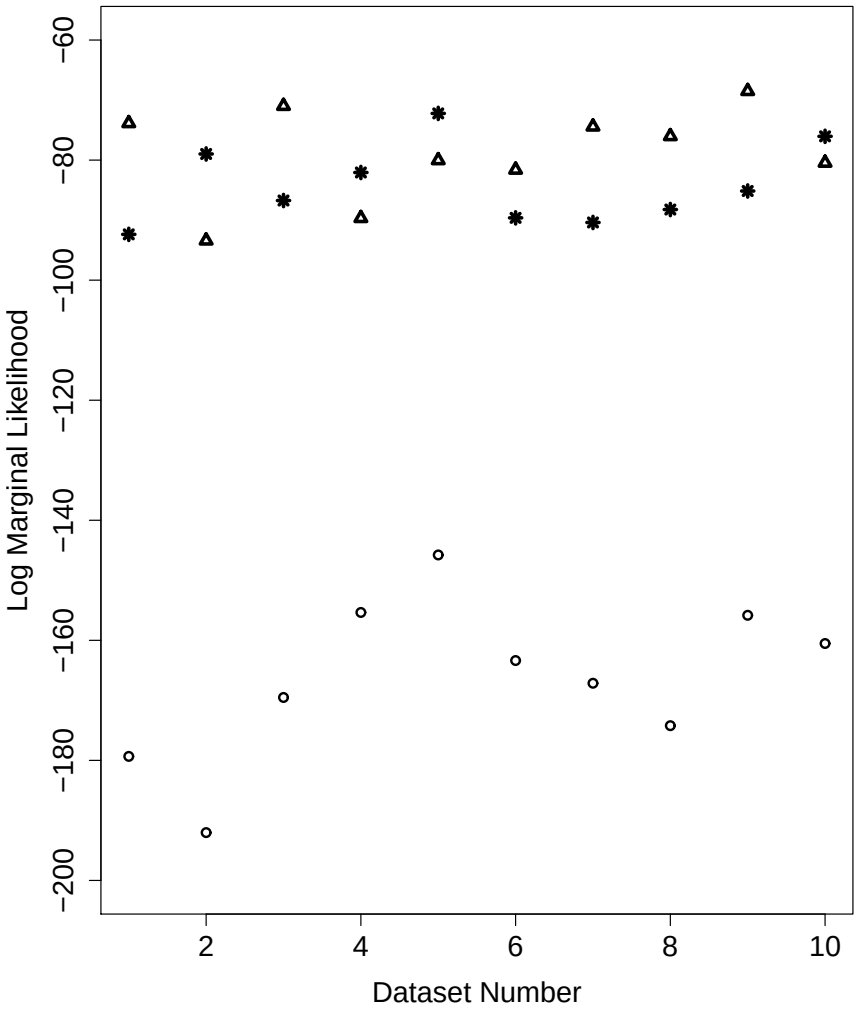


Figure 52: Log marginal likelihood scores for the set-up when data is simulated from the LV1 model and the parameters of the system were inferred using an explicit solution of the ODEs. The initial conditions of the system were inferred as additional parameters. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The higher the value on the y-axis, the more favoured a model is. Here, the log marginal likelihood scores favour the true model (LV1) 60% of the time and the LV3 model 40% of the time.

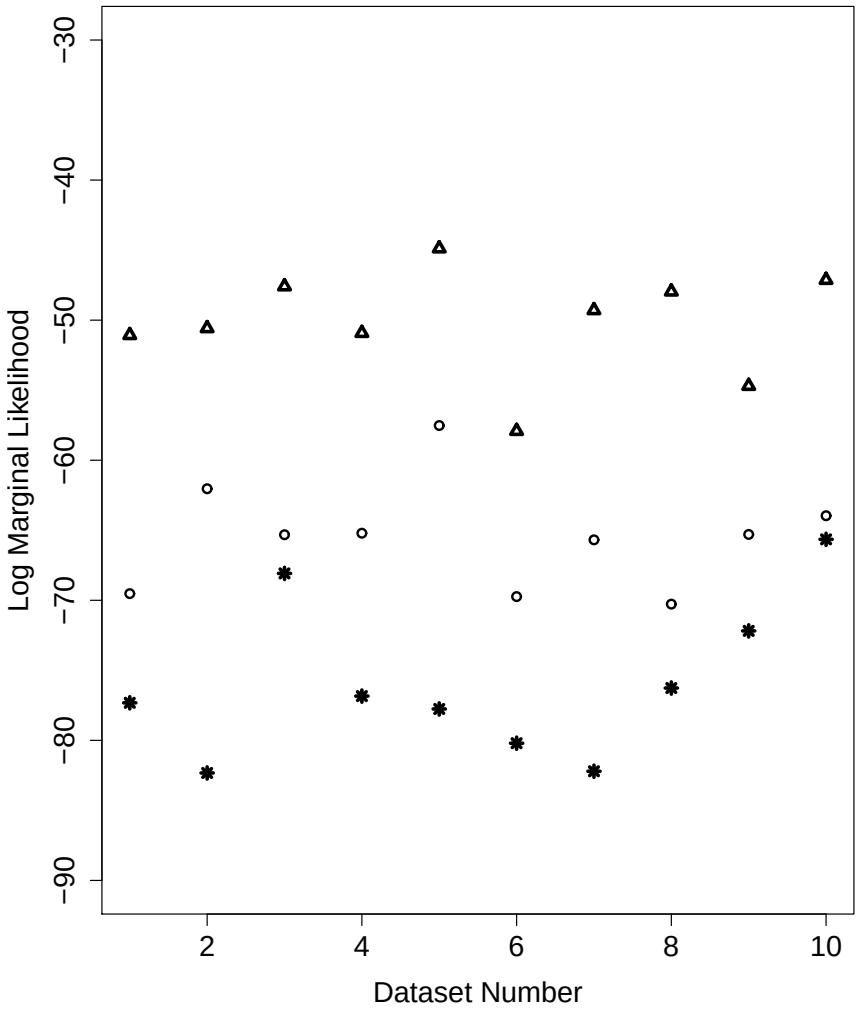


Figure 53: Log marginal likelihood scores for the set-up when data is simulated from the LV1 model and the parameters of the system were inferred using an explicit solution of the ODEs. The initial conditions of the system were held fixed at the true initial values. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The higher the value on the y-axis, the more favoured a model is. Here, the log marginal likelihood scores favour the true model (LV1) 100% of the time.

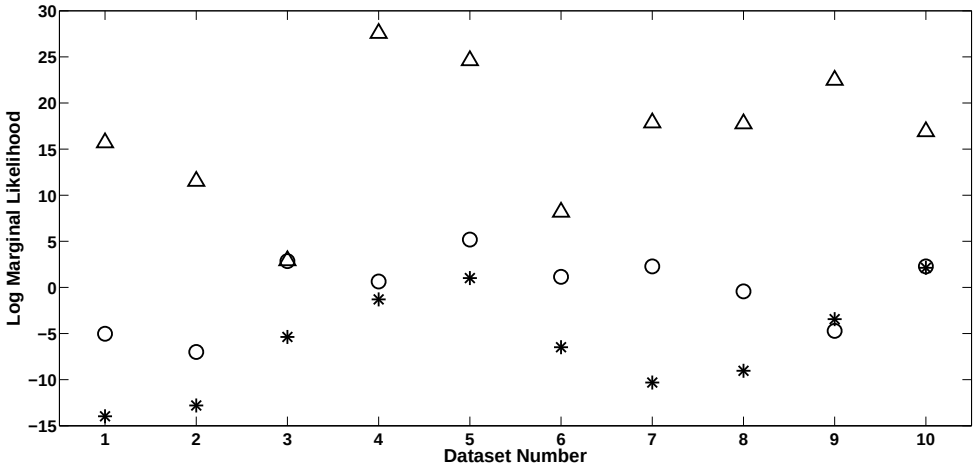


Figure 54: Log marginal likelihood scores for the set-up when data is simulated from the LV1 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The higher the value on the y-axis, the more favoured a model is. Here, the log marginal likelihood scores favour the true model (LV1) 100% of the time (the values are slightly higher for LV1 on dataset 3 than LV2).

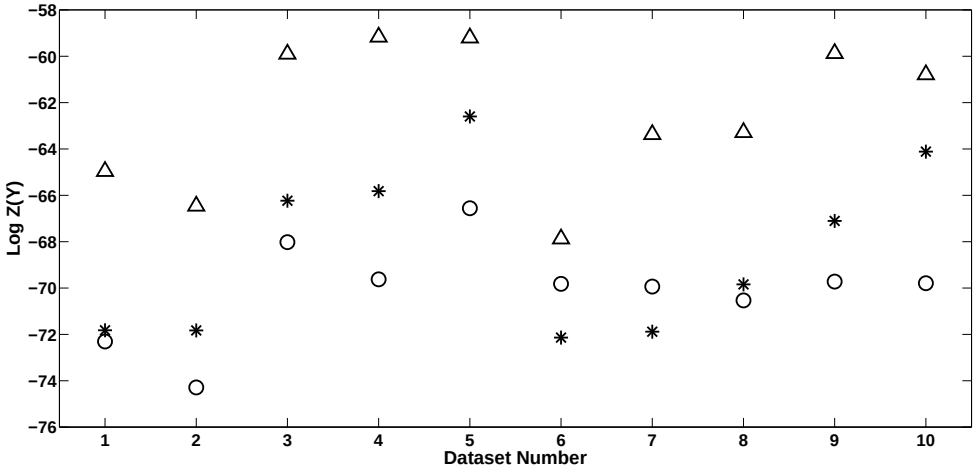


Figure 55: Log $\mathbb{Z}(\mathbf{Y})$ (equation 119) scores for the set-up when data is simulated from the LV1 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The higher the value on the y-axis the more favoured a model is. Here, the log $\mathbb{Z}(\mathbf{Y})$ scores favour the true model (LV1) 100% of the time.

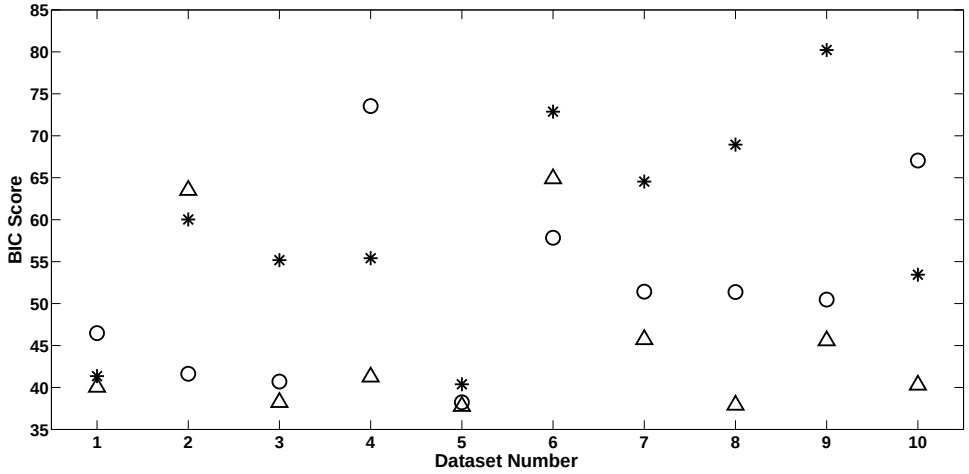


Figure 56: BIC scores for the set-up when data is simulated from the LV1 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The lower the value on the y-axis the more favoured a model is. Here, the BIC scores favour the true model (LV1) 80% of the time.

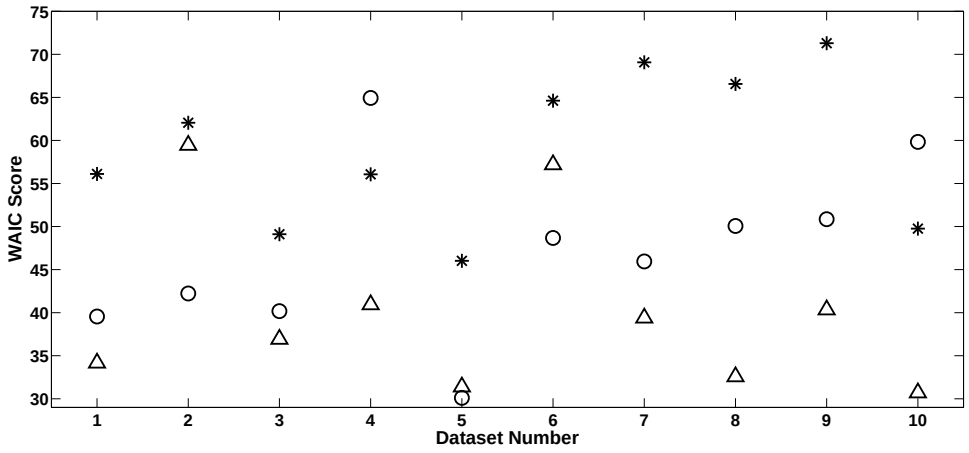


Figure 57: WAIC scores for the set-up when data is simulated from the LV1 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The lower the value on the y-axis the more favoured a model is. Here, the WAIC scores favour the true model (LV1) 70% of the time.

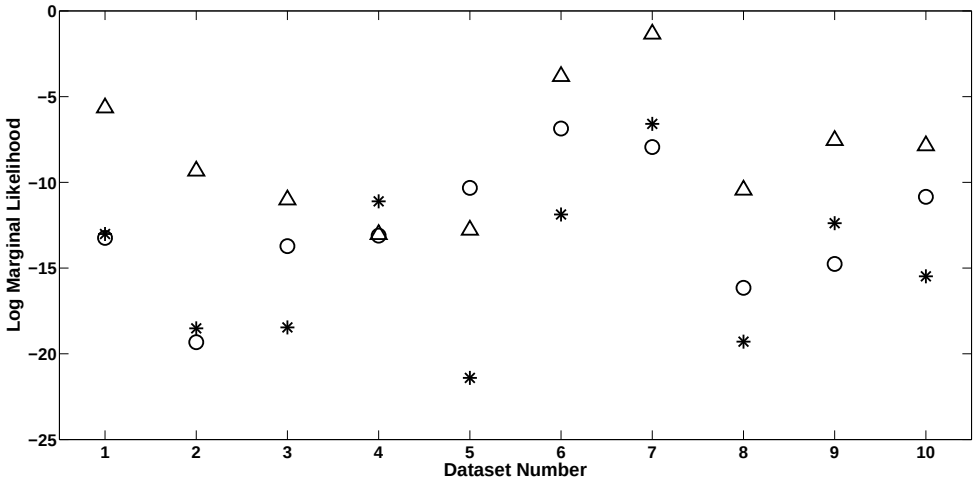


Figure 58: Log marginal likelihood scores for the set-up when data is simulated from the LV2 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The higher the value on the y-axis, the more favoured a model is. Here, the log marginal likelihood scores favour the true model (LV2) 10% of the time.

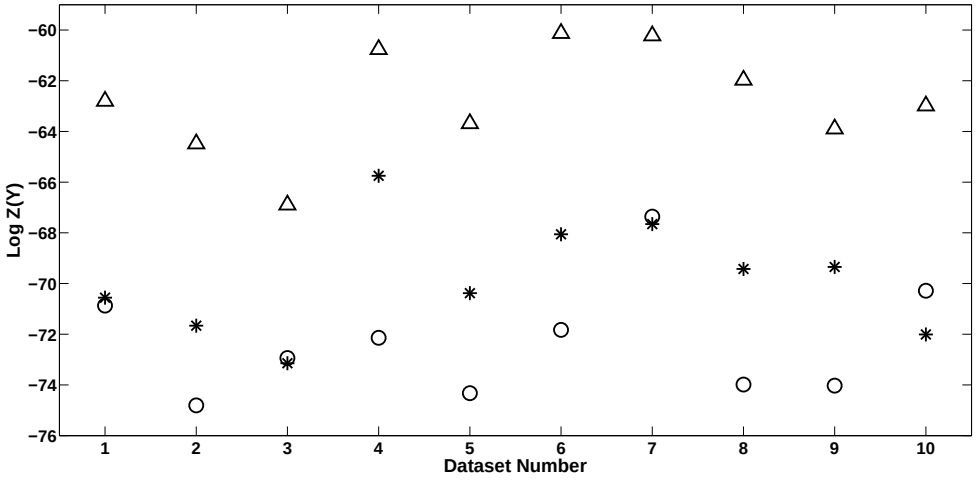


Figure 59: Log $\mathbb{Z}(\mathbf{Y})$ (equation 119) scores for the set-up when data is simulated from the LV1 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The higher the value on the y-axis, the more favoured a model is. Here, the log $\mathbb{Z}(\mathbf{Y})$ scores favour the true model (LV1) 0% of the time.

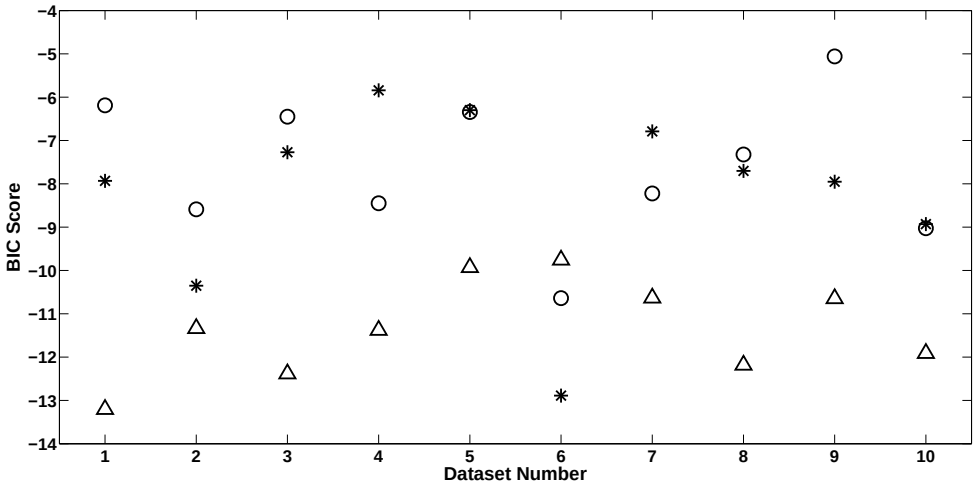


Figure 60: BIC scores for the set-up when data is simulated from the LV2 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The lower the value on the y-axis the more favoured a model is. Here, the BIC scores favour the true model (LV2) 0% of the time (the values for are slightly lower for LV2 on dataset 6 than LV1).

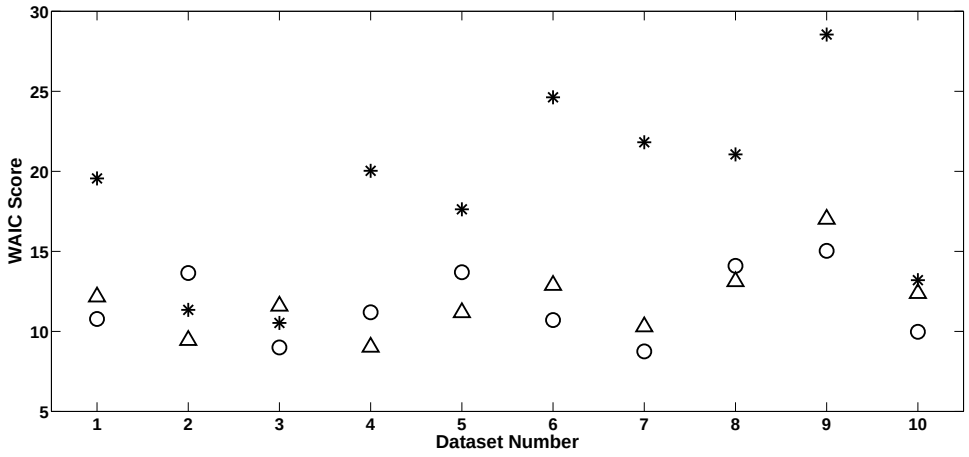


Figure 61: WAIC scores for the set-up when data is simulated from the LV2 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The lower the value on the y-axis the more favoured a model is. Here, the WAIC scores favour the true model (LV2) 60% of the time.

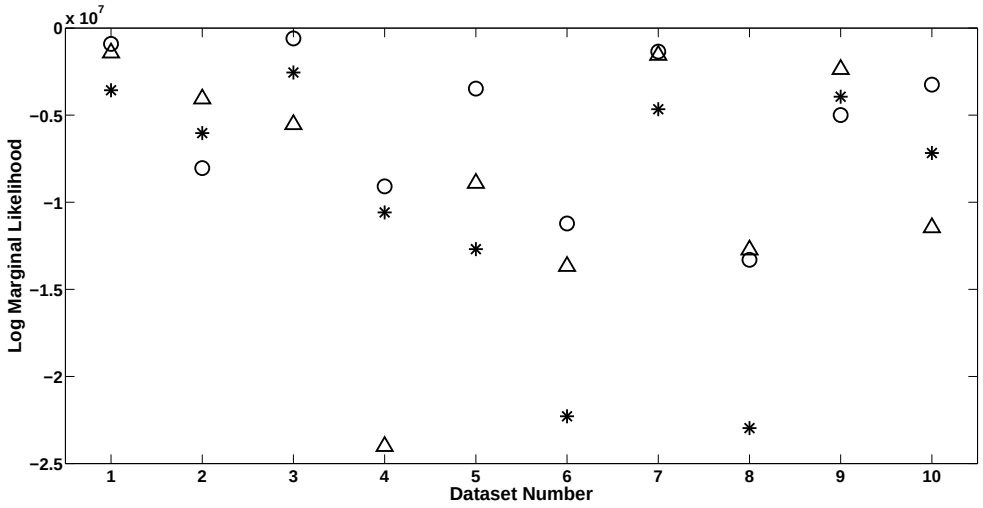


Figure 62: Log marginal likelihood scores for the set-up when data is simulated from the LV2 model, with parameter settings chosen to make the intra-species component effect more substantial. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The higher the value on the y-axis, the more favoured a model is. Here, the log marginal likelihood scores favour the true model (LV2) 70% of the time.

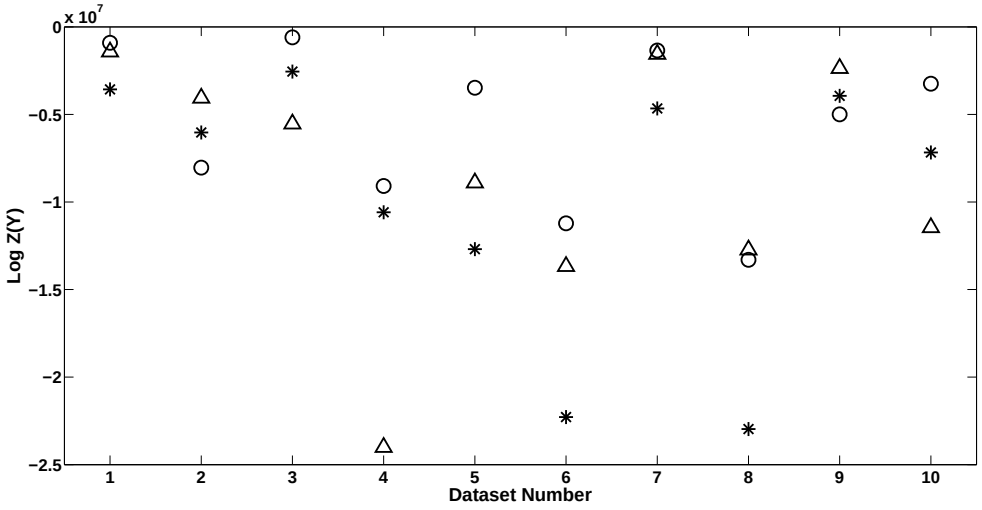


Figure 63: Log $\mathbb{Z}(\mathbf{Y})$ (equation 119) scores for the set-up when data is simulated from the LV2 model, with parameter settings chosen to make the intra-species component effect more substantial. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The higher the value on the y-axis, the more favoured a model is. Here, the log $\mathbb{Z}(\mathbf{Y})$ scores favour the true model (LV2) 70% of the time.

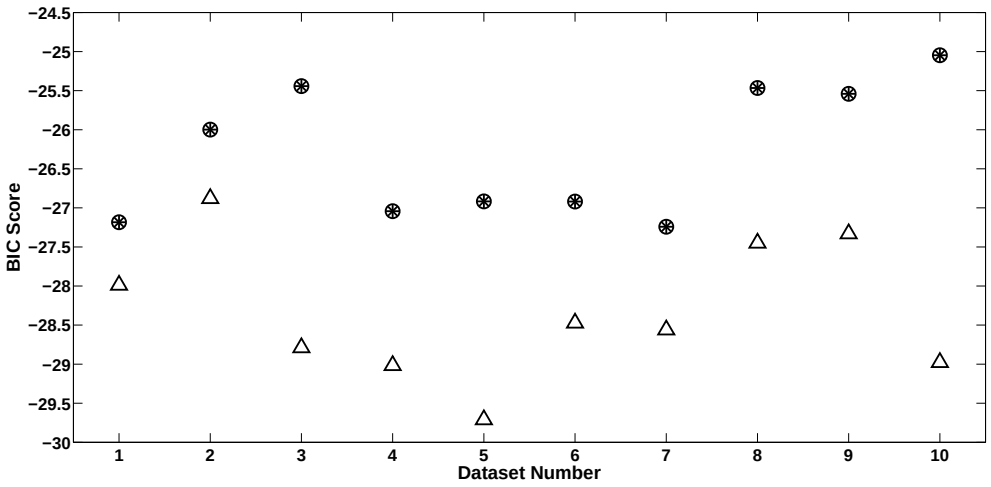


Figure 64: BIC scores for the set-up when data is simulated from the LV2 model, with parameter settings chosen to make the intra-species component effect more substantial. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The lower the value on the y-axis the more favoured a model is. Here, the BIC scores favour the true model (LV2) 0% of the time.

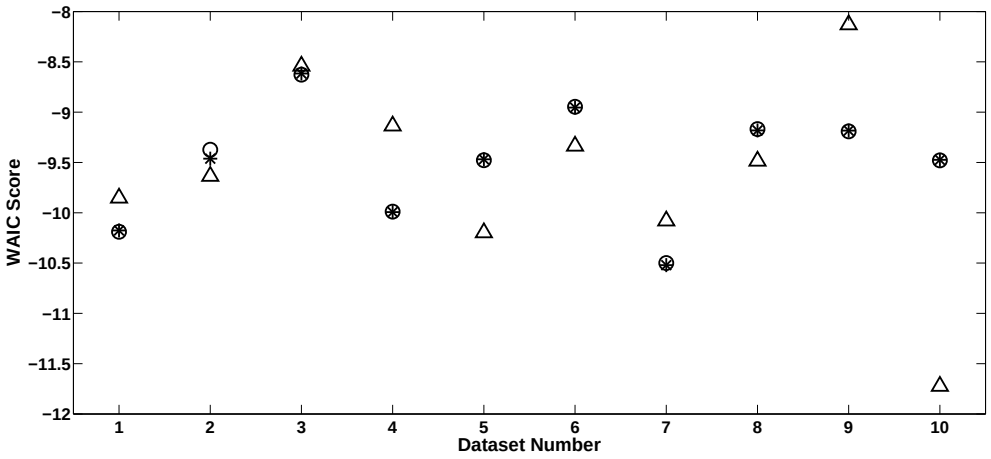


Figure 65: WAIC scores for the set-up when data is simulated from the LV2 model with, parameter settings chosen to make the intra-species component effect more substantial. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The lower the value on the y-axis the more favoured a model is. Here, the WAIC scores favour the true model (LV2) 30% of the time.

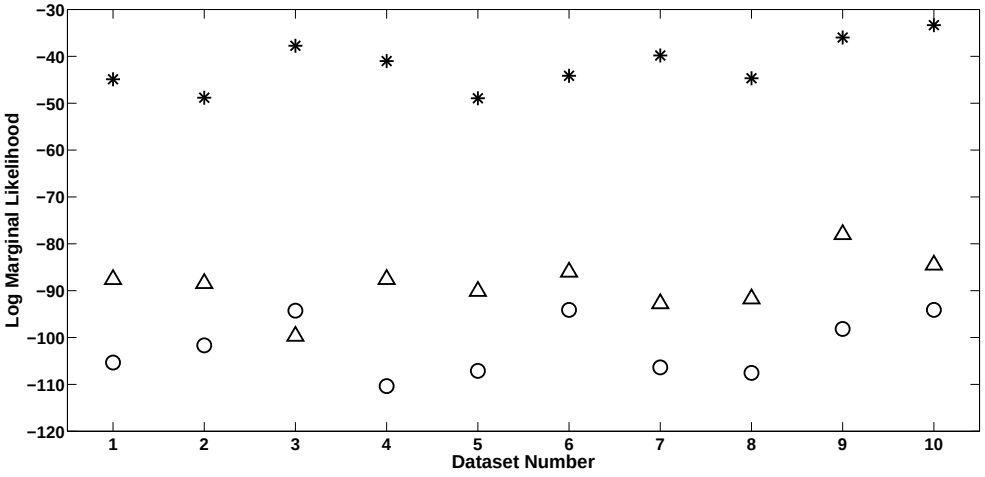


Figure 66: Log marginal likelihood scores for the set-up when data is simulated from the LV3 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The higher the value on the y-axis the more favoured a model is. Here, the log marginal likelihood scores favour the true model (LV3) 100% of the time.

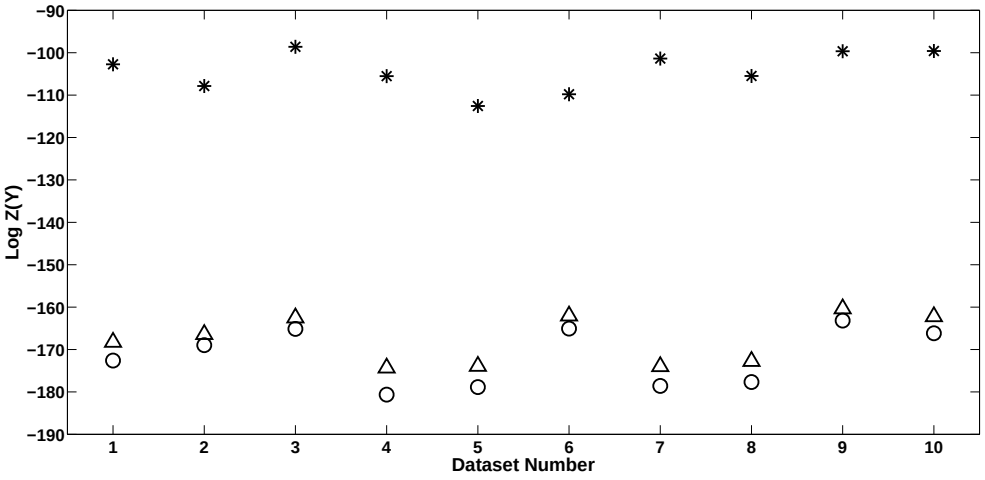


Figure 67: Log $\mathbb{Z}(\mathbf{Y})$ (equation 119) scores for the set-up when data is simulated from the LV3 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The higher the value on the y-axis the more favoured a model is. Here, the log $\mathbb{Z}(\mathbf{Y})$ scores favour the true model (LV3) 100% of the time.

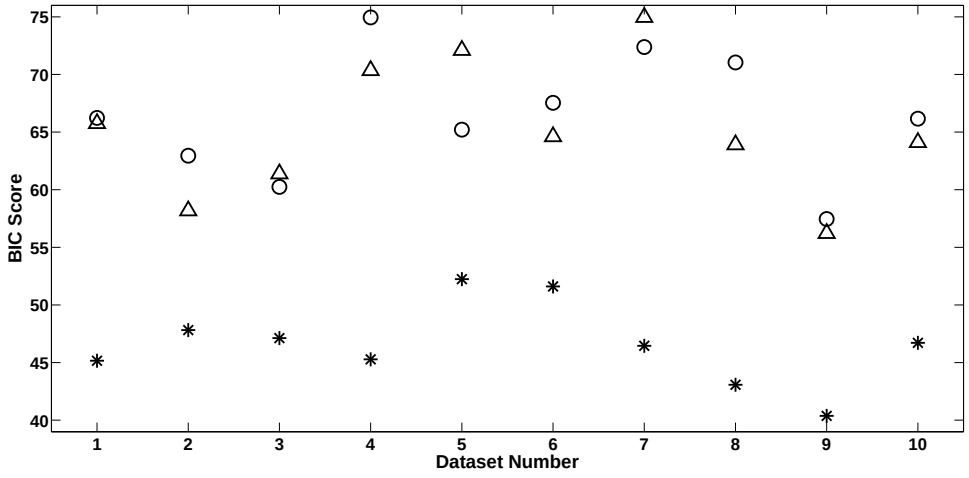


Figure 68: BIC scores for the set-up when data is simulated from the LV3 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The lower the value on the y-axis the more favoured a model is. Here, the BIC scores favour the true model (LV3) 100% of the time.

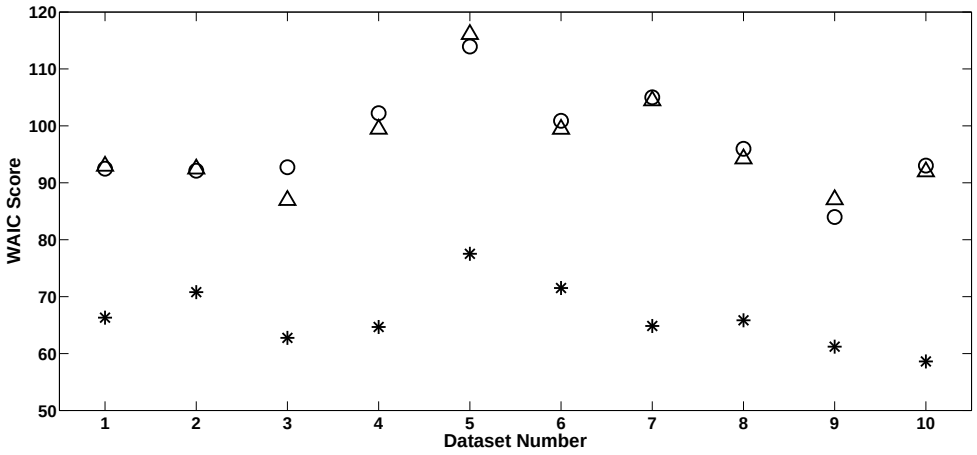


Figure 69: WAIC scores for the set-up when data is simulated from the LV3 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing LV1 as the candidate model, the circles represent the results when proposing LV2 as the candidate model and the stars represent the results when proposing LV3 as the candidate model. The lower the value on the y-axis the more favoured a model is. Here, the WAIC scores favour the true model (LV3) 100% of the time.

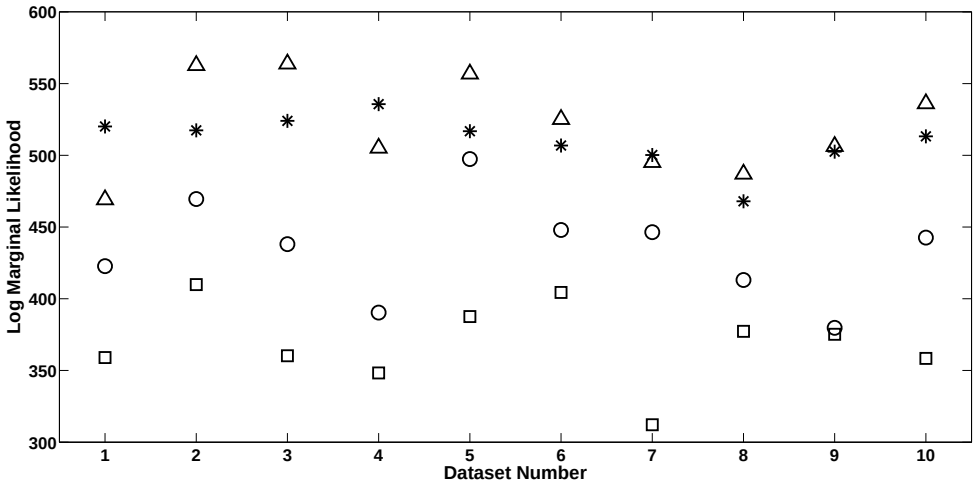


Figure 70: Log marginal likelihood scores for the set-up when data is simulated from the PSTP1 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing PSTP1 as the candidate model, the circles represent the results when proposing PSTP2 as the candidate model, the squares represent when proposing PSTP3 as the candidate model and the stars represent the results when proposing PSTP4 as the candidate model. The higher the value on the y-axis the more favoured a model is. Here, the log marginal likelihood scores favour the true model (PSTP1) 70% of the time.

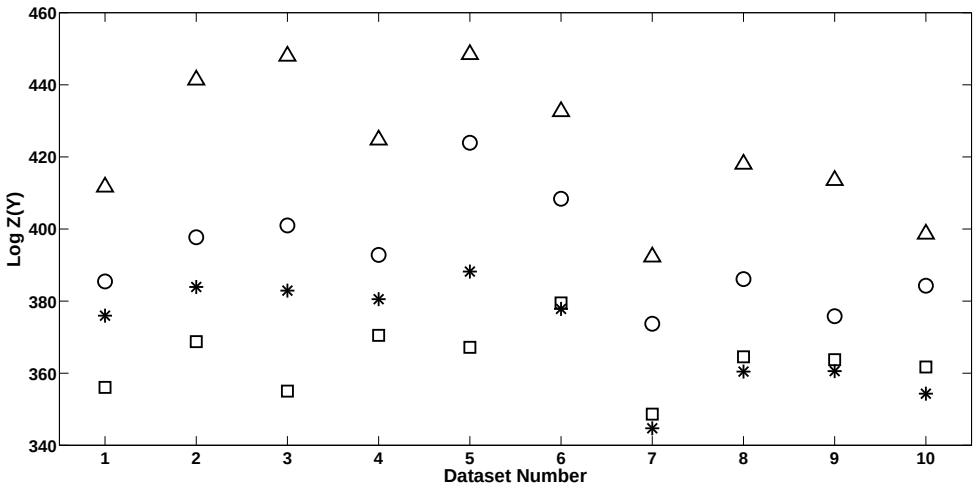


Figure 71: Log $\mathbb{Z}(\mathbf{Y})$ (equation 119) scores for the set-up when data is simulated from the PSTP1 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing PSTP1 as the candidate model, the circles represent the results when proposing PSTP2 as the candidate model, the squares represent when proposing PSTP3 as the candidate model and the stars represent the results when proposing PSTP4 as the candidate model. The higher the value on the y-axis the more favoured a model is. Here, the log $\mathbb{Z}(\mathbf{Y})$ scores favour the true model (PSTP1) 100% of the time.

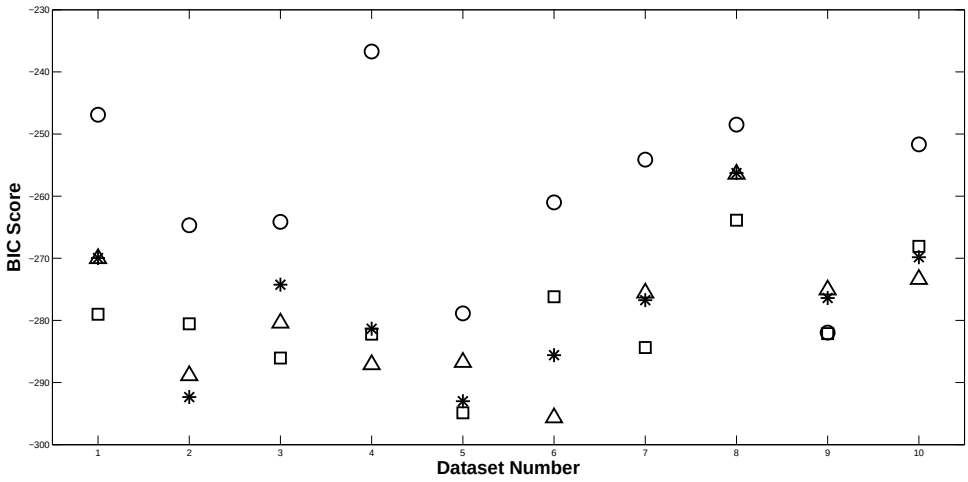


Figure 72: BIC scores for the set-up when data is simulated from the PSTP1 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing PSTP1 as the candidate model, the circles represent the results when proposing PSTP2 as the candidate model, the squares represent when proposing PSTP3 as the candidate model and the stars represent the results when proposing PSTP4 as the candidate model. The lower the value on the y-axis the more favoured a model is. Here, the BIC scores favour the true model (PSTP1) 30% of the time.

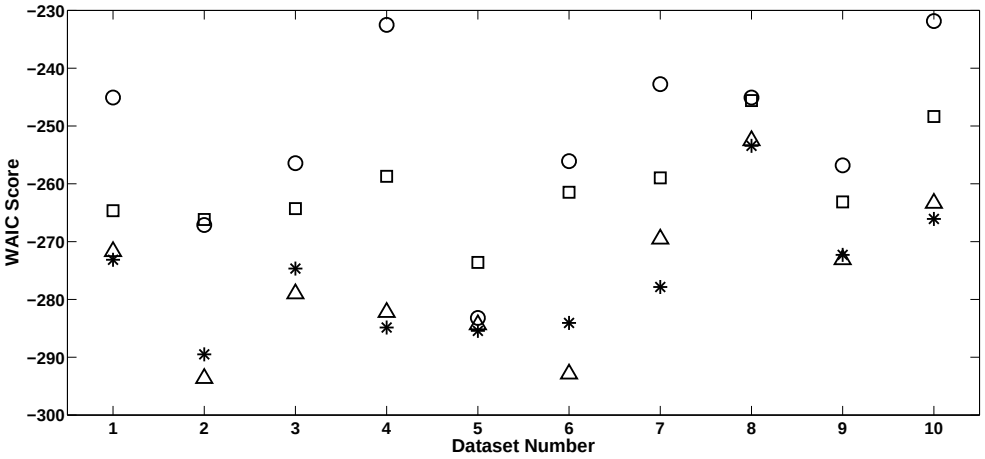


Figure 73: WAIC scores for the set-up when data is simulated from the PSTP1 model. The ticks on the x-axis represent the different datasets. The triangles represent the results when proposing PSTP1 as the candidate model, the circles represent the results when proposing PSTP2 as the candidate model, the squares represent when proposing PSTP3 as the candidate model and the stars represent the results when proposing PSTP4 as the candidate model. The lower the value on the y-axis the more favoured a model is. Here, the WAIC scores favour the true model (PSTP1) 40% of the time.

10 Bibliography

References

[1] Adon, N. A., Jabbar, M. H., and Mahmud, F. FPGA implementation for cardiac excitation-conduction simulation based on FitzHugh-Nagumo model. *5th International Conference on Biomedical Engineering in Vietnam*, 46, 2015.

[2] Akaike, H. Information theory and an extension of the maximum likelihood principle. *Proceedings of the Second International Symposium on Information Theory*, 1973.

[3] Aronszajin, N. Green's functions and reproducing kernels. *Proceedings of the Symposium on Spectral Theory and Differential Problems*, pages 355–411, 1951.

[4] Atkins, P. W. *Physical Chemistry*. Oxford University Press, Oxford, 3rd edition, 1986.

[5] Babtie, A., Kirk, P., and Stumpf, M. Topological sensitivity analysis for systems biology. *PNAS*, 111(51):18507–18512, December 2014.

[6] Bishop, C. M. Pattern recognition and machine learning. *Springer*, 2006.

[7] Bruggemeier, B., Schusterreiter, C., Pavlou, H., Jenkins, N., Lynch,

S., Bianchi, A., and Cai, X. Improving the utility of drosophila melanogaster for neurodegenerative disease research by modelling courtship behaviour patterns. *Report summarising the outcomes from the UK NC3R's and POEM's meeting*, 2014.

[8] Calderhead, B., Girolami, M. A., and Lawrence, N. D. Accelerating Bayesian inference over non-linear differential equations with Gaussian processes. *Neural Information Processing Systems (NIPS)*, 22, 2008.

[9] Campbell, D. and Steele, R. J. Smooth functional tempering for non-linear differential equation models. *Stat Comput*, 22:429–443, 2012.

[10] Dattner, I. M. and Klaassen, C. A. J. Optimal rate of direct estimators in systems of ordinary differential equations linear in functions of the parameters. *Electronic Journal of Statistics.*, 9(2):1939–1973, 2015.

[11] Dondelinger, F., Filippone, M., Rogers, S., and Husmeier, D. ODE parameter inference using adaptive gradient matching with Gaussian processes. *The 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 31 of JMLR:216–228, 2013.

[12] Duckett, G. and Barkley, D. Modeling the dynamics of cardiac action potentials. *Physical Review Letters*, 85(4), 200.

[13] Fan, J. and Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[14] FitzHugh, R. Impulses and physiological states in models of nerve membrane. *Biophys. J.*, 1:445–466, 1961.

[15] Friel, N. and Pettitt, A. N. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70:589–607, July 2008.

[16] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. *Bayesian Data Analysis*, volume 3. Chapman and Hall/CRC, 2013.

[17] Goktepe, S. and Kuhl, E. Computational modeling of cardiac electrophysiology: A novel finite element approach. *International journal for numerical methods in engineering*, 2009.

[18] Golub, G., Heath, M., and Wahba, G. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 1979.

[19] González, J., Vujačić, I., and Wit, E. Inferring latent gene regulatory network kinetics. *Statistical Applications in Genetics and Molecular Biology*, 12(1):109–127, 2013.

[20] González, J., Vujačić, I., and Wit, E. Reproducing kernel hilbert space based estimation of systems of ordinary differential equations. *Pattern Recognition Letters*, 45:26–32, 2014.

[21] Graepel, T. Solving noisy linear operator equations by Gaussian processes: Application to ordinary and partial differential equations. In

Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA, pages 234–241, 2003.

[22] Hansen, B. E. *Nonparametric Sieve Regression: Least Squares, Averaging Least Squares, and Cross-Validation.* The Oxford Handbook of Applied Nonparametric and Semiparametric Econometrics and Statistics, 2014.

[23] Hastie, T., Tibshirani, R., and Freidman, J. The elements of statistical learning. *Springer*, 2009.

[24] Holsclaw, T., Sansó, B., Lee, H. K. H., Heitmann, K., Habi, S., Higdon, D., and Alam, U. Gaussian process modeling of derivative curves. *Technometrics*, 2011.

[25] Kim, E. K. and Choi, E.-J. Pathological roles of mapk signaling pathways in human diseases. *Elsevier. Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease*, 2010.

[26] Liang, H. and Wu, H. Parameter estimation for differential equation models using a framework of measurement error in regression models. *J Am Stat Assoc.*, pages 1570–1583, December 2008.

[27] Lotka, A. The growth of mixed populations: two species competing for a common food supply. *Journal of the Washington Academy of Sciences*, 22:461–469, 1932.

[28] Lu, T., Liang, H., Li, H., and Wu, H. High dimensional odes coupled with mixed-effects modeling techniques for dynamic gene regulatory network identification. *Journal of the American Statistical Association.*, pages 1242–1258, 2011.

[29] Macdonald, B. and Husmeier, D. Gradient matching methods for computational inference in mechanistic models for systems biology: a review and comparative analysis. *Frontiers in Bioengineering and Biotechnology.*, 2015.

[30] Macdonald, B. and Husmeier, D. Computational inference in systems biology. *Bioinformatics and Biomedical Engineering: Third International Conference, IWBBIO. Proceedings, Part II. Series: Lecture Notes in Computer Science.*, pages 276–288, 2015.

[31] Macdonald, B., Dondelinger, F., and Husmeier, D. Inference in complex biological systems with gaussian processes and parallel tempering. *Proceedings of the 28th International Workshop on Statistical Modelling.*, pages 673–676, 2013.

[32] Macdonald, B., Higham, C., and Husmeier, D. Controversy in mechanistic modelling with gaussian processes. *Proceedings of the 32nd International Conference on Machine Learning.*, 37, 2015.

[33] Macdonald, B., Niu, M., Rogers, S., Filippone, M., and Husmeier, D. Approximate parameter inference in systems biology using gradient

matching: a comparative evaluation. *BioMedical Engineering Online.*, 2016.

[34] Martin, G. S. Cell signaling and cancer. *Meeting Review. Cancer.*, September 2003.

[35] Mercer, J. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209:415–446, 1909.

[36] Murphy, K. P. Machine learning. A probabilistic perspective. *The MIT Press*, 2012.

[37] Murray, I. and Adams, R. Slice sampling covariance hyperparameters of latent gaussian models. *Advances in Neural Information Processing Systems (NIPS),*, 23, 2010.

[38] Nagumo, J. S., Arimoto, S., and Yoshizawa, S. An active pulse transmission line simulating a nerve axon. *Proc. Inst. Radio Eng.*, 50:2061–2070, 1962.

[39] Pokhilko, A., Fernandez, A. P., Edwards, K. D., Southern, M. M., Halliday, K. J., and Millar, A. J. The clock gene circuit in arabidopsis includes a repressilator with additional feedback loops. *Molecular Systems Biology*, 8(574), 2012.

[40] Ramsay, J. O., Hooker, G., Campbell, D., and Cao, J. Parameter estimation for differential equations: a generalized smoothing approach. *J. R. Statist*, pages 741–796, 2007.

[41] Ranciati, S., Viroli, C., and Wit, E. Bayesian smooth-and-match estimation of ordinary differential equations parameters with quantifiable solution uncertainty. *arXiv:1604.02318v2 [stat.ME]*, 2016.

[42] Rasmussen, C. E. and Williams, C. K. I. Gaussian processes for machine learning. *The MIT Press.*, 2006.

[43] Robinson, J. C. *An introduction to ordinary differential equations.* Cambridge University Press., 2004.

[44] Runge, C. Über empirische funktionen und die interpolation zwischen aquidistanten ordinaten. *Zeitschrift fr Mathematik und Physik*, 1:224–243, 1901.

[45] Schöelkopf, B. and Smola, A. J. Learning with kernels: Support vector machines, regularization, optimisation and beyond. *MIT Press*, 2002.

[46] Solak, E., Murray-Smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E. Derivative observations in Gaussian process models of dynamic systems. *Advances in Neural Information Processing Systems*, pages 9–14, 2003.

[47] Vivekanandan, S., Emmanuel, D. S., and Kumari, R. propogation of action potential for Hansen's disease affected nerve model using FitzHugh

Nagumo like excitation. *Journal of Theoretical and Applied Information Technology*, 2013.

[48] Vyshemirsky, V. and Girolami, M. A. Bayesian ranking of biochemical system models. *Bioinformatics*, 24(6):883–839, 2008.

[49] Wang, Y. and Barber, D. Gaussian processes for bayesian estimation in ordinary differential equations. *Proceedings of the 31st International Conference on Machine Learning*, 32, 2014.

[50] Watanabe, S. *Algebraic Geometry and Statistical Learning Theory.* Cambridge University Press, 2009.

[51] Watanabe, S. Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 2010.

[52] Wu, H., Lu, T., Xue, H., and Liang, H. Sparse additive ordinary differential equations for dynamic gene regulatory network modeling. *American Statistical Association.*, 109(506), 2014.

[53] Xue, H., Miao, H., and Wu, H. Sieve estimation of constant and time-varying coefficients in nonlinear ordinary differential equation models by considering both numerical error and measurement error. *The Annals of Statistics.*, 38:2351–2387, 2010.

[54] Zhou, S. *Bayesian Model Selection in terms of Kullback-Leibler discrepancy.* PhD thesis, Columbia University, 2011.