Bayesian smooth-and-match inference for ordinary differential equations models linear in the parameters

Bayesian Smooth-and-Match inference for ordinary differential equations models linear in the parameters

Saverio Ranciati*

Department of Statistical Sciences,
University of Bologna, Via Belle Arti 41, Bologna, Italy

Ernst C. Wit

Institute of Computational Science,
Università della Svizzera italiana, Via Buffi 13, CH-6900 Lugano, Switzerland

Cinzia Viroli

Department of Statistical Sciences,
University of Bologna, Via Belle Arti 41, Bologna, Italy

Abstract

Dynamic processes are crucial in many empirical fields, such as in oceanography, climate science and engineering. Processes that evolve through time are often well described by systems of ordinary differential equations (ODEs). Fitting ODEs to data has long been a bottleneck, because the analytical solution of general systems of ODEs is often not explicitly available. We focus on a class of inference techniques that uses smoothing to avoid direct integration. In particular, we develop a Bayesian Smooth-and-Match strategy that approximates the ODE solution while performing Bayesian inference on the model parameters. We incorporate in the strategy two main sources of uncertainty: the noise level of the measured observations and the model approximation error. We assess the performance of the proposed approach in an extensive simulation study and on a canonical dataset of neuronal electrical activity.

keywords smoothing, penalized splines, MCMC, ridge regression

*Corresponding author: Saverio Ranciati, saverio.ranciati2@unibo.it

1 Introduction

Many processes that evolve through time can be described by systems of ordinary differential equations (ODEs). For example, the change, $dx(t)$, of the concentration level of a specific molecule in the cell follows a law that is described by some function $g_{\boldsymbol{\beta}}[x(t)]$, with a set of parameters $\boldsymbol{\beta}$ governing this law, where β_1 is the rate at which new ones are produced by the cell and some limit β_2 on the capacity to contain them. What we observe, in practice, are not the changes $dx(t)$, representing the derivative of the main process, but instead the actual concentration levels at finite sampling times, i.e., observations from the main process $x(t)$. This means that, in order to relate the data at our disposal to the parameters of interest, we need a solution for the system of differential equations. However, in most cases, no closed form solutions are available and numerical techniques are thus needed. Moreover, the observations may very well be affected by noise that perturbs the observed temporal dynamic of the process.

There are several techniques in the applied mathematics literature on this topic (see Robinson (2004) for an introduction). Most of them involve numerical integration, a straightforward approach to the problem that, however, does not take into account the uncertainty about the chosen statistical model nor the noise level of the observations. Also, methods relying on this type of solvers require the explicit computation of the solution at every step of any inference algorithm, severely hindering the procedure in practice. A way to avoid direct numerical integration or differentiation is *smoothing* the data, which, after all, are noisy draws from the true solution. An example of a *one-step* kernel-based nonparametric smooth estimator was recently proposed in Hall and Ma (2014). Another recent contribution to one-step methods to parameter estimation is given in Dattner and Gugushvili (2018), which provides both point and interval estimates for the parameters of interest of the ODE system. In general, the idea of smoothing to avoid integration falls under the class of *collocation* methods, including some *two-steps* (Liang and Wu, 2012; Gugushvili and Klaassen, 2012; Dattner and Klaassen, 2013; Vujačić, Dattner, González and Wit, 2015) and *iterative* procedures (Ramsay, Hooker, Campbell and Cao, 2007; Vujačić, Mahmoudi and Wit, 2016). Typically the first step consists of recovering a temporary solution of the system by smoothing or interpolating the data using e.g. cubic splines, nonparametric filters or local polynomial regression (Varah, 1982; Madár, Abonyi, Roubos and Szeifert, 2003; Brunel, 2008), and then applying nonlinear least squares to infer the parameters of the ODEs. The properties, such as consistency and asymptotic normality, of methods relying on nonlinear least squares with known initial conditions are discussed in Xue, Miao and Wu (2010). Other methods have involved reproducing kernel Hilbert space approaches by mapping the ODE problem into a Hilbert space (González, Vujačić and Wit, 2013, 2014).

Bayesian approaches have involved similar *two-steps* procedures Campbell and Steele (2012); Bhaumik and Ghosal (2015, 2017). Many Bayesian approaches to ODE estimation rely on assuming that the solution of the system can be described as a Gaussian Process (GP) Chkrebtii, Campbell, Calderhead and Girolami (2016). They encode naturally a source of randomness in the solution and simultaneously provide a class of flexible priors for the functions used to smooth the data coming from the ODE system. Calderhead et al. (2009) used GPs by means of adaptive gradient matching, with some drawbacks that were later addressed in Dondelinger et al. (2013). An advantage of these Bayesian methods is the complete probabilistic phrasing of the problem, allowing for a statistical quantification of the uncertainty about the solution obtained. The core of these procedures are, in fact, probabilistic solvers that can be sampled to explore the parameter space while obtaining indirectly a solution of the system (Conrad, Girolami, Särkkä, Stuart and Zygalakis, 2017). For some other implicit and explicit probabilistic solvers see Barber (2014). Applications of these methods on real life datasets are presented in Honkela, Girardot, Gustafson, Liu, Furlong, Lawrence and Rattray (2010) and Titsias, Honkela, Lawrence and Rattray (2012).

In this manuscript we propose a two-step Bayesian strategy that overcomes direct integration and that filters noise in the data, simultaneously. We achieve this by using a penalized splines approach to smooth the data and reconstruct the state variables of the ODEs. Normal priors on the ODE parameters result in adopting a ridge regression technique for regularized inference on the parameters governing the system. Our approach is entirely modular, which means that it can approach large, complex systems, as our framework allows for separate improvements, extensions or modifications.

The rest of the manuscript is organized as follows. In Section 2 we introduce the model and the distributional assumptions on the data, together with the prior distributions. In Section 3, we describe extensive simulation studies and present numerical and graphical results. In Section 4, we describe the analysis of a small dataset involving neuronal electrical activity modelled by means of a FizHugh-Nagumo system of ordinary differential equations. In Section 5, we provide a discussion of the method presented in this manuscript and we outline some future developments.

2 Dynamic model formulation

A p-dimensional dynamic process $\mathbf{x}$ is observed over n time points with noise. Each component $\mathbf{x}_k(t)$ of the process $\mathbf{x}(t)$ is sampled at time points $\{t_i\}_{i=1,\dots,n}$, with t_i a generic element from the finite set of ordered time points $t_i \in [a,b]$. Without loss of generality we will assume that $a = 0$ and $b = 1$. We furthermore assume the dynamic process $\mathbf{x}(t)$ to be well described by an ordinary differential equations (ODEs) model defined as:

$$\begin{cases} \mathbf{x}'(t) = g(\mathbf{x}(t)) \\ \mathbf{x}(0) = \boldsymbol{\xi} \end{cases} \tag{1}$$

where $\mathbf{x}'(t)$ is the first derivative with respect to time of the continuous process $\mathbf{x}(t) = (\mathbf{x}_1(t), \dots, \mathbf{x}_p(t))$, $\boldsymbol{\xi} = (\xi_1, \dots, \xi_p)$ is a vector of initial conditions for the system and $g : \mathbb{R}^p \to \mathbb{R}^p$ some integrable function of $\mathbf{x}(t)$.

We will assume that the component functions g_k can be written as

$$g_k(x) = \sum_{j=1}^{b_k} \beta_{kj} h_{kj}(x),$$

where the h_{kj} are assumed to be known integrable functions. In case the dynamical system is well-known, it could be that the functions h_{kj} are given by certain field-specific principles. In case the system is unknown, then the functions h_{kj} could be chosen as part of some convenient basis of integrable functions. We focus only on ODE models that are linear in the parameters: the choice of the functions g_k leads to parametric linear regression problems, as in Dattner and Klaassen (2015).

A solution of the system (1) can be given in integral form

$$\begin{aligned} x_{\beta,k}(t) &= \xi_k + \int_0^t g_k(\mathbf{x}(s))\, ds \\ &= \xi_k + \sum_{j=1}^{b_k} \beta_j H_{jk,\mathbf{x}_\beta}(t), \end{aligned} \tag{2}$$

where $H_{jk,\mathbf{x}_\beta}$ is the integral of $h_{kj}(x(.))$, which itself depends on the solution $\mathbf{x} = \mathbf{x}_\beta$ of the ODE. We can collect all the elements $H_{jk,\mathbf{x}_\beta}(t_i)$ in a $np \times \sum_{k=1}^{p} b_k$ regression matrix $\boldsymbol{H}_{\mathbf{x}_\beta}$, such that

$$\mathbf{x}_\beta = \xi + \boldsymbol{H}_{\mathbf{x}_\beta}\beta,$$

where $\mathbf{x}$ contains the np stacked values $x_{\beta,k}(t_i)$ for $i = 1,\dots,n$ and $k = 1,\dots,p$.

The observed process is called $\mathbf{y}$, where the collection $\{\mathbf{y}_1, \ldots, \mathbf{y}_p\}$ contains n-dimensional vectors consisting of noisy observations y_{ik} of $x_{\beta,ik} = x_{\beta,k}(t_i)$, where i indexes time and k indicates one of the p state vectors. In particular, we assume that the observations $\{y_{ik}\}$ are independent and given by

$$y_{ik}|\beta \sim \mathcal{N}\big(x_{\beta,ik}, \sigma_k^2\big) \tag{3}$$

with σ_k^2 a parameter describing the noise level in the data for component $\mathbf{x}_k(t)$.

Assuming an underlying Bayesian framework, the unknown system parameters $\boldsymbol{\beta}_k$ are assumed to follow a conjugate normal distribution

$$\boldsymbol{\beta}_k|\lambda_{\boldsymbol{\beta}_k} \sim \mathcal{N}_{b_k}\left(\mathbf{0}, [\lambda_{\boldsymbol{\beta}_k}\mathrm{I}_{b_k}]^{-1}\right),$$

The chosen prior, effectively, reproduces Bayesian 'ridge regression', with $\lambda_{\beta,k}$ acting as a penalizing term for which we assume a prior distribution

$$\lambda_{\boldsymbol{\beta}_k}|\alpha_\beta, \gamma_\beta \sim \mathrm{Gam}(\alpha_\beta, \gamma_\beta).$$

where $(\alpha_\beta, \gamma_\beta)$ are chosen to reflect weakly informative priors. For ξ we select a flat normal distribution, whereas for each σ_k^2 an improper reference prior $p(\sigma_k^2) = 1/\sigma_k^2$.

2.1 An alternative smoothing model

Full Bayesian inference of β, which governs the dynamic system, is rather complicated. The main reason is that $H_{\mathbf{x}_\beta}$ defined in (2) is only implicitly defined. An explicit form of the posterior of β would, therefore, be only possible for the simplest ODEs with explicit solutions. Alternatively, a Metropolis-Hastings sampler would require for each proposal β' the associated solution $\mathbf{x}_\beta$ of the dynamic system in order to accept or reject the proposal.

Alternatively, we will define a Bayesian version of the smooth-and-match estimator (Gugushvili and Klaassen, 2012). It will break the dependence of the inference of β on the solution $\mathbf{x}_\beta$, by replacing it by a smoothed version of the data, which, after all, are noisy versions of the true solution $\mathbf{x}_\beta$. It has an empirical Bayes flavour, as we effectively estimate the nuisance parameter $H_{\mathbf{x}}$ "off-line".

A smooth approximation model for $\mathbf{x_k}$, as an alternative to the true model (1), can be given by a cubic spline with q_k knots of the form

$$x_{\theta,ik}(t) = x_{\theta,k}(t) = \sum_{h=1}^{q_k+2} \theta_{hk}\psi_h(t), \tag{4}$$

where ψ is a cubic splines basis on $[0, 1]$ (Wood, 2006, p. 122). This approximating spline for state variable k takes the value $x_{\theta,k}(t_i) = \theta_k^T \psi(t_i)$ at the observation times t_i. In particular, the observations y_{ik} under this model are assumed to be independent and depend on the parameters θ_k and noise level σ_k^2 as

$$y_{ik} \sim N(x_{\theta,ik}, \sigma_k^2),$$

where the smoothing parameters themselves are constrained by means of an informative, smoothness inducing hyper-distribution,

$$\boldsymbol{\theta}_k | \, \lambda_{\theta_k} \sim \mathcal{N}_{q_k+2} \left(\mathbf{0}, [\lambda_{\theta_k} S_\psi]^{-1} \right),$$

where S_ψ is the usual matrix of integrals of cross-products of the second derivatives of ψ (Gu, 2013, p. 37). Although the first two rows and columns of S_ψ are zeros, pseudo-inversion can be performed to avoid singularity of the matrix (Wood, 2006, p. 127). This ensures that non-linearity in the components is captured without, however, producing curves that would overfit the noise.

The selection of the hyper-parameter λ_{θ_k} is done by a further conjugate hyper-distribution,

$$\lambda_{\theta_k} | \, \alpha_{\theta_k}, \gamma_{\theta_k} \sim \text{Gam}(\alpha_\theta, \gamma_\theta)$$

for some weakly informative hyperparameters $(\alpha_\theta, \gamma_\theta)$. More specifically, we select values of the hyperparameters that encourage undersmoothing of the data (Gugushvili and Klaassen, 2012), with enough variance for the Gamma distribution to be able to shift to a more penalized curve if needed.

2.2 Bayesian Smooth-and-Match strategy

We have now two models for the data. The spline smoothing step (4) is a convenient approach for building an approximation of regression matrix $\boldsymbol{H}_{\mathbf{x}_\beta}$ by $\boldsymbol{H}_{\mathbf{x}_\theta}$. Borrowing a terminology from Gugushvili and Klaassen (2012), we call this a Bayesian Smooth-and-Match strategy. The procedure consists of two steps:

1. *Smoothing* step. Perform Bayesian penalized spline smoothing to sample $\{\mathbf{x}_\theta\}$ through a Gibbs sampler on $\boldsymbol{\theta}|\mathbf{y}$;

2. *Matching* step. Plug-in the sampled $\{\mathbf{x}_{\theta,k}\}$, computed with the updated sampled values for $\boldsymbol{\theta}_k$ from the previous step: this provides a way to compute the matrix $\boldsymbol{H}_{\mathbf{x}_\theta}$, required by the Bayesian regression sampler for $\boldsymbol{\beta}|\mathbf{y}$.

The two steps of the inference strategy are not completely disconnected. The vector σ^2 connects both parts and acts as a measure of uncertainty both from the smoothing and the regression.

Dropping hyper-parameters from the notation, the joint posterior distribution of the parameters in our model is:

$$\begin{aligned} p(\boldsymbol{\theta}, \xi, \boldsymbol{\beta}, \boldsymbol{\lambda}_\theta, \lambda_\beta, \boldsymbol{\sigma}^2 |\, \mathbf{y}) &\propto \prod_{k=1}^{p} p_{\mathrm{smth}}\big(\mathbf{y}_k |\, \boldsymbol{\theta}_k, \sigma_k^2\big)\, p(\boldsymbol{\theta}_k |\, \lambda_{\theta_k})\, p(\lambda_{\boldsymbol{\theta}_k}) p(\sigma_k^2) \\ &\times \prod_{k=1}^{p} p_{\mathrm{reg}}\big(\mathbf{y}_k |\, \xi_k, \boldsymbol{\beta}_k, H_{\mathbf{x}_{\boldsymbol{\theta}},k}, \sigma_k^2\big)\, p\big(\boldsymbol{\beta}_k |\, \lambda_{\boldsymbol{\beta}_k}\big)\, p\big(\lambda_{\boldsymbol{\beta}_k}\big)\, p(\xi_k), \end{aligned}$$

which is also represented in the graphical model in Figure 1.

More sophisticated smoothing could be performed (Morrissey, Juárez, Denby and Burroughs, 2011), but we consider a simpler approach because we are only interested in recovering the components $H_{\mathbf{x}}$, which are used as regressors in the second step of the procedure, since, from a frequentist point of view, a consistent estimator of the ODE parameters can be obtained under mild conditions when following this plug-in approach (Gugushvili and Klaassen, 2012). Although we rely on a simpler smoothing technique, we still operate in a Bayesian framework and thus we readily obtain measures of uncertainty about parameters in our model: in particular, after obtaining samples from the posterior distribution, we can compute quantities such as posterior standard deviations and credible intervals. The ODE parameters of the model are all updated with Gibbs samplers, using standard conjugated priors; the chains mix well and no block-sampling updates are needed. We give more details about the implementation in the Appendix, while scripts of method and code reproducing our simulations are available at the following link: `https://github.com/savranciati/ODEsnm`.

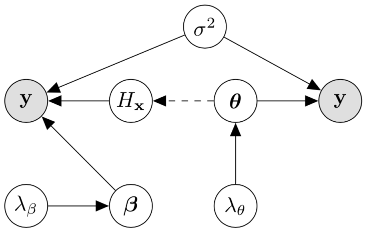

Figure 1: Graphical representation of the ODE system (left) and smoothing model (right) in relation to the data y.

2.3 Relationship with other methods

One prominent feature of our model formulation is the vector of observations $\mathbf{y}$ appears twice: once in the smoothing model and secondly as the noisy solution of the ODE system. As we mentioned, this has close resemblance to an empirical Bayes approach. However, specifically in the field of inference of ODE systems, this approach is particularly common.

The GPODE model in Barber and Wang (2014) is a probabilistic generative model and its graph representation contains two nodes for the same quantity, namely, the ODE solution $\mathbf{x}(t)$. The authors model the functions $\mathbf{x}(t)$ as coming from a Gaussian process (GP) and exploit the fact that differentiating a GP produces a derivative $\mathbf{x}'(t)$ that is still a GP with an analytically explicit kernel. Then, they marginalize over the components $\mathbf{x}(t)$ with a standard convolution integral and assume data $\mathbf{y}$ to be normally distributed. After that, they reintroduce $\mathbf{x}(t)$ and couple it with the obtained derivatives to measure the distance between the deterministic ODEs of the system and the ones estimated with the data.

This approach has some problems. As pointed out in Macdonald, Higham and Husmeier (2015), having two nodes assigned to the same quantity is methodologically inconsistent. To solve the issue, they introduce a dummy variable that mimics $\mathbf{x}(t)$, thus removing the inconsistency. However, the two nodes are still conceptually describing the very same quantity and a natural definition of this dependency would be an undirected edge between them: this addition, unfortunately, changes the graph from a directed acyclic graph (DAG) to a chain graph, which is not a probabilistic generative model anymore. To preserve its essence, they keep the two nodes separate from one another but highlight the consequences of this choice: when in the case of partially observed systems some of the noisy vectors $\mathbf{y}_k$ are not available, the model itself might not be identifiable because of the likelihood not depending anymore on the parameters of the ODEs after marginalizing over the unobserved quantities.

As we face the same issue with our proposed strategy, we are also limited to situations where all the components of the process $\mathbf{y}(t)$ are observed. We decide to couple the two versions of $\mathbf{y}$ only by assuming they share the same variance σ^2, as described in Figure 1. Although this seems reasonable in principle, it has a drawback, which is most evident when we try to simulate data. In fact, we start with a deterministic solution $\mathbf{x}$ of an ODE system, and we perturb it with some noise σ^2 to obtain an observed vector $\mathbf{y}$. Now, recovering the noise level in the data through smoothing is not strictly equivalent anymore to estimating σ^2 through the regression model involving design matrix $H_{\hat{\mathbf{x}}}$. However, as $\hat{\mathbf{x}}$ is consistent, asymptotically they are equivalent in a frequentist sense. Moreover, such undesired 'mismatch' effect should not present itself as a serious problem when dealing with real world observations, bearing in mind the two sources of error flow into one another.

3 Simulation study

In this section, we analyze the performance of our proposed estimation strategy by applying it to three different ODE systems, starting from a simple one component logistic growth model, then moving to a two component Lotka-Voltera model and ending with a three component epidemic model for HIV viral fitness.

For each ODE system, we simulate nine scenarios: we consider three sample sizes, $n = 25, n = 100, n = 500$, and three degrees of contamination of the data with Gaussian noise: low, medium and high. The noise level is quantified through the *signal-to-noise ratio* (SNR), that is the ratio between the standard deviation of the deterministic simulated solution and the standard deviation of the noise term we use to perturb it.

We compare the performance of our Bayesian Smooth-and-Match strategy, labelled as *BSM*, with the collocation method implemented in the `R` package *CollocInfer* (Ramsay, Hooker, Campbell and Cao, 2007). *CollocInfer* is an adaptive gradient matching method based on spline smoothing. Tuning parameters for the collocation method, such as the number of knots, order of the basis for the splines, and the penalization term, are selected with the functions provided in the `R` package. The collocation method also needs initial guesses for the regression coefficients and we provide starting points drawn randomly from uniform distributions over ranges around the true value of the parameters.

For each scenario and each ODE system, we apply both algorithms to 100 independently simulated datasets. We summarize the results as the average across these replicates. The uncertainty about the estimated parameters is quantified through the mean squared error (MSE), computed on the 100 simulated datasets. Given the sensitivity of *CollocInfer* to the provided starting values, we check if convergence is achieved by the algorithm; we discard datasets that result in degenerate estimates for the parameters and do not consider these values when computing the MSE. A measure of the number of actual datasets (NAD) used for the CollocInfer method is provided in the tables. For our BSM method, all datasets are used. As for the visual representation of the results, Figures 4 to 6 show a mosaic plot with the results for one of the 100 datasets.

3.1 Logistic population growth

We consider first the one component logistic growth model (McKendrick and Pai, 1912), frequently employed in ecology and biology to describe growth dynamics of a population. The system is defined as:

$$\frac{d\mathbf{x}(t)}{dt} = a\,\mathbf{x}(t)\left(1 - \frac{\mathbf{x}(t)}{K}\right),$$

where a is the growth rate and K the carrying capacity of the population involved. We consider another, linear representation of previous equation, that is,

$$\frac{d\mathbf{x}(t)}{dt} = \beta_1\mathbf{x}(t) + \beta_2[\mathbf{x}(t)]^2,$$

which is linear in the parameters $\beta_1 = a$ and $\beta_2 = -a/K$. Together with the initial condition ξ_1, these are the parameters of interest. The true values for ξ_1, β_1 and β_2 used in the simulations are reported in Table 2.

In our BSM procedure, we select the number of knots for the smoothing step equal to $q_1 = 2$, aiming for some undersmoothing of $\mathbf{y} = (y_i, \dots, y_n)$, as suggested by Gugushvili and Klaassen (2012). The functions used to build the regression matrix are $h_1(x) = x$ and $h_2(x) = x^2$.

In every scenario, the number of actual datasets (NAD) used to compute the results for *CollocInfer* is less than 100, meaning that a significant number of degenerate solutions were discarded in the process. As expected, Table 2 shows that increasing noise levels produces higher MSE for both methods. The average posterior mean of ξ_1 is stable throughout all the scenarios, showing some bias only when SNR is small. *CollocInfer* does not provide an estimate for the initial condition ξ_1. With *BSM* we can recover the first parameter β_1 accurately in almost every scenario, showing better results in comparison with *CollocInfer*. The *CollocInfer* algorithm seems to be more stable when retrieving the second parameter β_2. The uncertainty parameter σ^2, only estimated by *BSM*, appears to be less sensitive to changes in sample size n and more to SNR.

A visual description of the results is presented in Figure 4: in most of the plots, the solid line describing the true curve, the dotted smoother of the data $(y_1, \dots, y_n)$ and the dashed ODE regression solution are indistinguishable from each other. They start to become appreciably different in the right part of the mosaic plot, which refers to scenarios with the highest level of noise. When calculating the average L_2 difference between the smoother and the truth versus the ODE regression solution and the truth, we find that the ODE regression solution outperforms the smoother (e.g. or $n = 500$ and $SNR = 1.3$, the L_2 distances are respectively 0.022 versus 0.007). The model information helps to fit true function x more accurately.

3.2 Lotka-Volterra

In the second batch of simulations we consider the Lotka-Volterra system (Edelstein-Keshet, 1988). This system of ODEs is used to model the temporal dynamics of two competing groups, traditionally referred to as *preys* and *predators*. By setting some of the parameters of the ODEs to zero or imposing constraints, the Lotka-Volterra system can also describe systems that characterize seasonal epidemic processes. The Lotka-Volterra model is described by the following set of equations:

$$\begin{cases} \mathbf{x}_1'(t) = \beta_1 \mathbf{x}_1(t) + \beta_2 \mathbf{x}_1(t)\mathbf{x}_2(t) \\ \mathbf{x}_2'(t) = \beta_3 \mathbf{x}_2(t) + \beta_4 \mathbf{x}_1(t)\mathbf{x}_2(t) \\ \mathbf{x}_1(0) = \xi_1 \\ \mathbf{x}_2(0) = \xi_2. \end{cases}$$

We focus the inference procedure on the first ODE of the system, and thus on the subset of parameters $(\xi_1, \beta_1, \beta_2)$. The advantage of our empirical Bayes procedure is that the different equations can be inferred in an uncoupled way. We explore the same nine simulation scenarios as in the logistic growth case. We use as regressing functions the quantities $h_1(\mathbf{x}) = \mathbf{x}_1$ and $h_2(\mathbf{x}) = \mathbf{x}_1\mathbf{x}_2$. We select the same number of knots, $q_1 = q_2 = 5$, for both splines.

In Table 3, averages of the posterior means and corresponding mean square errors are reported for all the scenarios. We see both algorithms performing well when the sample size is $n = 25$, regardless of the noise levels. The number of actual datasets used to compute averages for *CollocInfer* also shows that convergence was achieved for all the first three scenarios. When evaluating the performance of the two methods, for $n = 100$, we notice *BSM* performs slightly better in terms of bias of the estimated parameters β_1 and β_2; also, *CollocInfer* shows slightly higher MSEs for the second estimated parameter with respect to *BSM*. This difference is more prominent when $n = 500$.

Figure 5 shows that the smoothing and the ODE regression curves are most different in the low information scenario, i.e., $n = 25$ and $SNR = 1.3$. As expected, the ODE regression curve is closer to the truth than the smoothing curve (L_2 difference of 0.252 vs. 0.690, respectively, which reduces to 0.084 vs. 0.094 in the high information scenario $n = 500$ and $SNR = 13$).

3.3 HIV viral fitness

The third batch of data is simulated from a set of ODEs modelling the dynamics of HIV virus (Bonhoeffer, May, Shaw and Nowak, 1997). The system is defined as:

$$\begin{cases} \mathbf{x}_1'(t) = \beta_1 + \beta_2 \mathbf{x}_1(t) + \beta_3 \mathbf{x}_1(t)\mathbf{x}_3(t) \\ \mathbf{x}_2'(t) = \beta_3 \mathbf{x}_1(t)\mathbf{x}_3(t) - 0.5\mathbf{x}_2(t) \\ \mathbf{x}_3'(t) = 0.5\beta_4 \mathbf{x}_2(t) + \beta_5 \mathbf{x}_3(t) \\ \mathbf{x}_1(0) = \xi_1 \\ \mathbf{x}_2(0) = \xi_2 \\ \mathbf{x}_3(0) = \xi_3, \end{cases}$$

where we focus on the first ODE and the subset of parameters $(\xi_1, \beta_1, \beta_2, \beta_3)$. The simulation settings are the same as before. True parameter values are chosen as the ones used in Vujačić, Dattner, González and Wit (2015). The numbers of knots are the same for all smoothing models, $q_1 = q_2 = q_3 = 20$. The regression matrix $\boldsymbol{H}_{\mathbf{x}_\theta}$ consists of the integrals of the following functions, $h_1(\mathbf{x}) = 1$, $h_2(\mathbf{x}) = \mathbf{x}_1$ and $h_3(\mathbf{x}) = \mathbf{x}_1\mathbf{x}_3$.

We report the results in Table 4 and Figure 6. The parameter β_2 proved to be difficult to estimate with *CollocInfer*: we decided to fix it to the true value for *CollocInfer*, while nevertheless estimating it via *BSM*. Except for $n = 25$ with the highest level of noise SNR $= 1.3$, *BSM* performs better than *CollocInfer* in inferring the parameters. Approximately 15% of all simulations for the first three scenarios for *CollocInfer* do not converge. If the algorithm converges, however, then the algorithm seems quite robust. The number of discarded datasets gets lower as the sample size grows, as expected.

As we can see from the mosaic plot in Figure 6, for $n = 100$ and $n = 500$ there is an appreciable difference between the two curves with respect to the true one, especially on the right-half portion of each plot. The L_2 difference for the ODE regression and the true curves versus the smoothed and true curves are, respectively, 16.179 and 11.743 for $n = 100$ and SNR$=$ 1.3, meaning that the smoothing step provides a reconstructed curve slightly closer to the true one. The opposite happens when the sample size gets to 500 and the noise level of SNR $= 1.3$: the L_2 differences are 1.397 and 5.859, respectively.

3.4 Reconstructing a complex ODE system

Whereas the above simulation systems considered only a small number of equations, the method we propose is highly parallelizable and could in principle consider large ODE systems. For this purpose, we simulate data from a synthetic ODE system with $p = 10$ equations. The system is defined as:

$$\begin{cases} \mathbf{x}_1'(t) = \alpha_1 \mathbf{x}_1(t) - \beta_1 \mathbf{x}_{10}(t) \sin(\mathbf{x}_2(t)) \\ \mathbf{x}_2'(t) = -\alpha_2 \mathbf{x}_2(t) + \beta_2 \mathbf{x}_1(t) \sin(\mathbf{x}_3(t)) \\ \mathbf{x}_3'(t) = \alpha_3 \mathbf{x}_3(t) - \beta_3 \mathbf{x}_2(t) \cos(\mathbf{x}_4(t)) \\ \mathbf{x}_4'(t) = -\alpha_4 \mathbf{x}_4(t) + \beta_4 \mathbf{x}_3(t) \cos(\mathbf{x}_5(t)) \\ \mathbf{x}_5'(t) = \alpha_5 \mathbf{x}_5(t) - \beta_5 \mathbf{x}_4(t) \cos(\mathbf{x}_6(t)) \\ \mathbf{x}_6'(t) = -\alpha_6 \mathbf{x}_6(t) + \beta_6 \mathbf{x}_5(t) \cos(\mathbf{x}_7(t)) \\ \mathbf{x}_7'(t) = \alpha_7 \mathbf{x}_7(t) - \beta_7 \mathbf{x}_6(t) \cos(\mathbf{x}_8(t)) \\ \mathbf{x}_8'(t) = -\alpha_8 \mathbf{x}_8(t) + \beta_8 \mathbf{x}_7(t) \sin(\mathbf{x}_9(t)) \\ \mathbf{x}_9'(t) = \alpha_9 \mathbf{x}_9(t) - \beta_9 \mathbf{x}_8(t) \sin(\mathbf{x}_{10}(t)) \\ \mathbf{x}_{10}'(t) = -\alpha_{10} \mathbf{x}_{10}(t) + \beta_{10} \mathbf{x}_9(t) \sin(\mathbf{x}_1(t)). \end{cases}$$

The initial conditions are $\mathbf{x}_j(0) = \xi_j$ for $j = 1, \ldots, p$, and we perform inference on all the parameters appearing in the ODEs. We assume we know the structure of each equation, as in the other simulation settings, which means using the appropriate regressing functions $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$: i.e. for the first ODE, $h_1(\mathbf{x}) = \mathbf{x}_1$ and $h_2(\mathbf{x}) = \mathbf{x}_{10} \sin(\mathbf{x}_2)$. However, instead of the true one, we fit a somewhat misspecified model: for each equation in the system we consider a third regressing function $h_3(\mathbf{x})$, with an associated parameter γ_j. In principle, every γ_j should be sampled from a posterior distribution centered around zero, as the contribution from $h_3(\mathbf{x})$ is not present in the data generating process, i.e. $\gamma_j = 0 \;\; \forall j$. More details about values for the parameters used to generate data, and a list of the misspecified regressing functions $h_3(\mathbf{x})$ for each equation, are provided in the Appendix. The idea here is to use our approach to recover the structure of the ODE system, even if there is some form of misspecification.

We summarize the results by looking at the 95% credible interval of all the sampled parameters $(\alpha_1, \beta_1, \gamma_1, \ldots, \alpha_{10}, \beta_{10}, \gamma_{10})$, and considering which intervals contain or not the zero. In particular, parameters for the ODE system $\{\alpha_j, \beta_j\}_{j=1,\ldots,p}$ have credible intervals centered around their true values without including 0: this is true for all the parameters except β_3, β_5, α_6, which have posterior medians close to their true values but intervals wide enough to also contain zero. A visual example of the result is provided in Figure 2, where we report data for the fourth ODE of the system. Although the smoothing step produces a curve (dotted line) too wiggly to capture the true underlying function (solide line), the regression part of our approach (long-dashed line) better recovers the overall behavior of the component $\mathbf{x}_4(t)$.

The choice of p is mainly dictated by the ease of exposition of the system in the context of this article. From a computational point of view, inference is

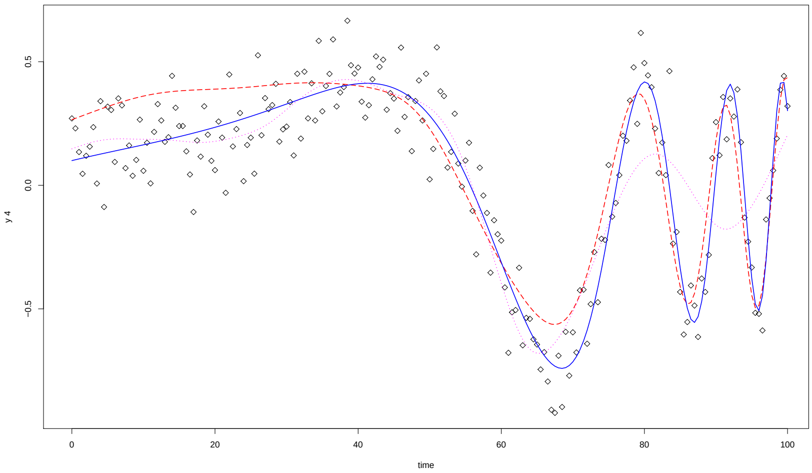

Figure 2: Fourth ODE $\mathbf{x}_4(t)$ from the synthetic system: observed noisy data (*dots*), smoothing spline (*dotted* line), true solution of the ODE system (*solid* line) and reconstructed solution (*long-dashed* line) for the first variable $\mathbf{x}_{\cdot 1}$; $n = 200$ time points.

linear in the number of equations. Moreover, parallelization could in principle result in only a sublinear growth of computational time.

4 Modelling neuron electrical activity

We analyze a synthetic neuron electrical activity example available from the package `CollocInfer` (Hooker, Xiao and Ramsay, 2010). The so-called *Fh-Ndata* data consist of 41 evenly-spaced observations in the time frame $[0, 20]$ from the following ODEs model

$$\begin{cases} \mathbf{x}_1'(t) = c(\mathbf{x}_1(t) - \mathbf{x}_1^3(t)/3 + \mathbf{x}_2(t)) \\ \mathbf{x}_2'(t) = -\frac{1}{c}(\mathbf{x}_1(t) - a + b\mathbf{x}_2(t)) \\ x_1(0) = \xi_1 \\ x_2(0) = \xi_2 \end{cases} \tag{5}$$

known as the FitzHugh-Nagumo system (FitzHugh, 1961; Nagumo, Arimoto and Yoshizawa, 1962), which describes pulse transmission for neuronal activity. The parameter values used to generate the data are $a = 0.2, d = 0.2, c = 3, \xi_1 = 0.5$. The simulated values are then perturbed with variances equal to 0.25 for both the variables $\mathbf{x}_1$ and $\mathbf{x}_2$. The system in Equation 5 is not linear in the parameters. We thus consider another representation

$$\begin{cases} \mathbf{x}_1' = \beta_4(\mathbf{x}_1(t) - \mathbf{x}_{.1}^3(t)/3 + \mathbf{x}_2) \\ \mathbf{x}_2' = \beta_1\mathbf{x}_1(t) + \beta_2 + \beta_3\mathbf{x}_2(t) \\ x_1(0) = \xi_1 \\ x_2(0) = \xi_2, \end{cases}$$

and we focus on the second differential equation and the subset of parameters $\xi_2, \beta_1 = -1/c, \beta_2 = a/c$ and $\beta_3 = -b/c$. The regression matrix $\boldsymbol{H}_{\mathbf{x}_\theta}$ consists of the integrals of the following functions, $h_1(\mathbf{x}) = \mathbf{x}_1$, $h_2(\mathbf{x}) = 1$ and $h_3(\mathbf{x}) = \mathbf{x}_2$. We compare our results with the point estimates obtained by Vujačić, Dattner, González and Wit (2015).

In Figure 3, the reconstructed curves from both *smoothing* and *matching* steps of the procedure are plotted. The dotted smoother captures some bias in the boundary, in contrast to the long-dashed ODE regression solution. As for the ODE parameters, we report their posterior means in Table 1. The BSM algorithm recovers the true value of β_1 with appreciable accuracy. For β_2 and especially β_3, the algorithm returns slightly biased posterior means. The posterior mean for σ^2 is at 0.06 lower than the one used to perturb the data, equal to 0.25. This probably suggests some level of overfitting.

Table 1: Results for the *FhNdata*: posterior means (*posterior standard deviations* within brackets) from *BSM* in the second column and point estimates from Vujačić, Dattner, González and Wit (2015) in the third column

Parameters	Post. Mean (*post. sd*)	Vujačić et al. (2015)
$\xi_2 = 0.5$	0.696 (*0.244*)	0.569
$\beta_1 = -0.33$	-0.322 (*0.041*)	-0.333
$\beta_2 = 0.067$	0.091 (*0.025*)	0.106
$\beta_3 = -0.067$	-0.028 (*0.066*)	-0.047
σ^2	0.060 (*0.019*)	

Runtime: 10,000 MCMC iterations in 29.17 seconds.

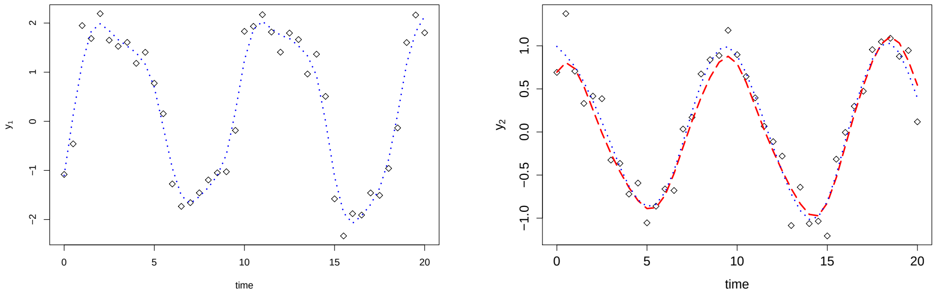

Figure 3: FitzHugh-Nagumo system for neuron electrical activity: observed noisy data (*dots*), smoothing spline (*dotted* line) and reconstructed solution (*long-dashed* line, only right plot) for the first variable (left plot) and the second variable $\mathbf{x}_2$ (right plot)

5 Conclusions

We have proposed a Bayesian approach to indirectly solve an ODE system while doing inference on its parameters. The employed strategy is suitable for ODEs that are linear in their parameters, with observations available for all the components of the system. It is compartmentalized into two main stages: first, a *smoothing* step that approximates the solutions of the ODE process through penalized spline smoothing of the noisy observations; second, a *match* step, where the smoothed curves are numerically integrated in order to construct an empirical Bayes ODE regression matrix as inputs for ridge penalized regression.

The two phases of the procedure are jointly governed by σ^2, a noise parameter common to both steps measuring the solution uncertainty. This parameter brings together the two main sources of uncertainty: measurement error and the model approximation error. We evaluated the performance and reliability of the strategy through various ODEs systems. We also tested the approach on a dataset previously analyzed by (Vujačić, Dattner, González and Wit, 2015). The Bayesian Smooth and Match (BSM) procedure that we proposed has the advantages of being fast, simple to implement and providing the ODE solution as a by-product of the inference procedure.

The 'tuning' parameters of the method are minimal: the number of knots and their placement have no substantial impact on the reliability of the smoothing step. Other spline methods, such as *B-splines*, *thin plate plates*, etc., that do not require such a choice, can also be employed in the *smoothing* step. An interesting alternative is to employ the P-splines approach proposed in Ventrucci and Rue (2016). It uses a penalized complexity prior, i.e., a prior distribution injecting information not in terms of the penalty parameter λ_θ, but about the polynomial order needed to reconstruct $\mathbf{x}$. This is quite an appealing approach because it is usually easier to elicit the information in terms of equivalent polynomial order, especially in ODE context.

As far as the integration is concerned to obtain $\boldsymbol{H}_{\mathbf{x}_\theta}$, we rely on an easy to implement, albeit 'rough', trapezoidal rule using the observed time points as the grid to evaluate the integral. A better approximation can be achieved by employing a finer grid, at the cost of increased computational time. Other types of penalization, instead of the ridge, could be explored for the regression step of the matching step, losing however the correspondence with Tikhonov regularization.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

Barber, D., 2014. On solving Ordinary Differential Equations using Gaussian Processes. arXiv preprint arXiv:1408.3807 .

Barber, D., Wang, Y., 2014. Gaussian Processes for Bayesian Estimation in Ordinary Differential Equations, in: Jebara, T., Xing, E.P. (Eds.), Proceedings of the 31st International Conference on Machine Learning (ICML-14), JMLR Workshop and Conference Proceedings. pp. 1485–1493. URL: `http://jmlr.org/proceedings/papers/v32/barber14.pdf`.

Bhaumik, P., Ghosal, S., 2015. Bayesian two-step estimation in differential equation models. Electronic Journal of Statistics 9, 3124–3154.

Bhaumik, P., Ghosal, S., 2017. Efficient Bayesian estimation and uncertainty quantification in ordinary differential equation models. Bernoulli 23, 3537–3570.

Bonhoeffer, S., May, R.M., Shaw, G.M., Nowak, M.A., 1997. Virus dynamics and drug therapy. Proceedings of the National Academy of Sciences 94, 6971–6976.

Brunel, N.J.B., 2008. Parameter estimation of ODE's via nonparametric estimators. Electronic Journal of Statistics 2, 1242–1267.

Calderhead, B., Girolami, M., Lawrence, N.D., 2009. Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes, in: Advances in neural information processing systems, pp. 217–224.

Campbell, D., Steele, R.J., 2012. Smooth functional tempering for nonlinear differential equation models. Statistics and Computing 22, 429–443.

Chkrebtii, O.A., Campbell, D.A., Calderhead, B., Girolami, M.A., 2016. Bayesian solution uncertainty quantification for differential equations. Bayesian Analysis 11, 1239–1267.

Conrad, P.R., Girolami, M., Särkkä, S., Stuart, A., Zygalakis, K., 2017. Statistical analysis of differential equations: introducing probability measures on numerical solutions. Statistics and Computing 27, 1065–1082. URL: `https://doi.org/10.1007/s11222-016-9671-0`, doi:`10.1007/s11222-016-9671-0`.

Dattner, I., Gugushvili, S., 2018. Application of one-step method to parameter estimation in ode models. Statistica Neerlandica 72, 126–156.

Dattner, I., Klaassen, C.A.J., 2013. Estimation in Systems of Ordinary Differential Equations Linear in the Parameters. arXiv preprint arXiv:1305.4126 .

Dattner, I., Klaassen, C.A.J., 2015. Optimal rate of direct estimators in systems of ordinary differential equations linear in functions of the parameters. Electronic Journal of Statistics 9, 1939–1973.

Dondelinger, F., Husmeier, D., Rogers, S., Filippone, M., 2013. Ode parameter inference using adaptive gradient matching with gaussian processes, in: Artificial Intelligence and Statistics, pp. 216–228.

Edelstein-Keshet, L., 1988. Mathematical models in biology. volume 46. Siam.

FitzHugh, R., 1961. Impulses and physiological states in theoretical models of nerve membrane. Biophysical journal 1, 445.

González, J., Vujačić, I., Wit, E., 2013. Inferring latent gene regulatory network kinetics. Statistical applications in genetics and molecular biology 12, 109–127.

González, J., Vujačić, I., Wit, E., 2014. Reproducing kernel Hilbert space based estimation of systems of ordinary differential equations. Pattern Recognition Letters 45, 26–32.

Gu, C., 2013. Smoothing spline ANOVA models. volume 297. Springer Science & Business Media.

Gugushvili, S., Klaassen, C.A.J., 2012. $\sqrt{n}$-consistent parameter estimation for systems of ordinary differential equations: bypassing numerical integration via smoothing. Bernoulli 18, 1061–1098.

Hall, P., Ma, Y., 2014. Quick and easy one-step parameter estimation in differential equations. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76, 735–748. URL: http://dx.doi.org/10.1111/rssb.12040, doi:10.1111/rssb.12040.

Honkela, A., Girardot, C., Gustafson, E.H., Liu, Y.H., Furlong, E.E., Lawrence, N.D., Rattray, M., 2010. Model-based method for transcription factor target identification with limited data. Proceedings of the National Academy of Sciences 107, 7793–7798.

Hooker, G., Xiao, L., Ramsay, J., 2010. Collocinfer: collocation inference for dynamic systems. R package version 0.1.0 .

Liang, H., Wu, H., 2012. Parameter estimation for differential equation models using a framework of measurement error in regression models. Journal of the American Statistical Association .

Macdonald, B., Higham, C., Husmeier, D., 2015. Controversy in mechanistic modelling with Gaussian processes, in: Journal of Machine Learning Research: Workshop and Conference Proceedings, Microtome Publishing. pp. 1539–1547.

Madár, J., Abonyi, J., Roubos, H., Szeifert, F., 2003. Incorporating Prior Knowledge in a Cubic Spline Approximation Application to the Identification of Reaction Kinetic Models. Industrial & engineering chemistry research 42, 4043–4049.

McKendrick, A.G., Pai, M.K., 1912. The Rate of Multiplication of Microorganisms: A Mathematical Study. Proceedings of the Royal Society of Edinburgh 31, 649–653. doi:10.1017/S0370164600025426.

Morrissey, E.R., Juárez, M.A., Denby, K.J., Burroughs, N.J., 2011. Inferring the time-invariant topology of a nonlinear sparse gene regulatory network using fully Bayesian spline autoregression. Biostatistics 12, 682–694.

Nagumo, J., Arimoto, S., Yoshizawa, S., 1962. An active pulse transmission line simulating nerve axon. Proceedings of the IRE 50, 2061–2070.

Ramsay, J.O., Hooker, G., Campbell, D., Cao, J., 2007. Parameter estimation for differential equations: a generalized smoothing approach. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69, 741–796.

Robinson, J.C., 2004. An introduction to ordinary differential equations. Cambridge University Press.

Titsias, M.K., Honkela, A., Lawrence, N.D., Rattray, M., 2012. Identifying targets of multiple co-regulating transcription factors from expression time-series by Bayesian model comparison. BMC systems biology 6, 1.

Varah, J., 1982. A spline least squares method for numerical parameter estimation in differential equations. SIAM Journal on Scientific and Statistical Computing 3, 28–46.

Ventrucci, M., Rue, H., 2016. Penalized complexity priors for degrees of freedom in bayesian p-splines. Statistical Modelling 16, 429–453.

Vujačić, I., Mahmoudi, S.M., Wit, E., 2016. Generalized tikhonov regularization in estimation of ordinary differential equations models. Stat 5, 132–143.

Vujačić, I., Dattner, I., González, J., Wit, E., 2015. Time-course window estimator for ordinary differential equations linear in the parameters. Statistics and Computing 6, 1057–1070.

Wood, S., 2006. Generalized additive models: an introduction with R. CRC Press.

Xue, H., Miao, H., Wu, H., 2010. Sieve estimation of constant and time-varying coefficients in nonlinear ordinary differential equation models by considering both numerical error and measurement error. Annals of statistics 38, 2351.

Table 2: Average posterior means (with *MSE* in brackets) for the parameters of the logistic population growth model.

Sample size	Parameters	Noise level					
		low ($SNR = 13$)		medium ($SNR = 6.5$)		high ($SNR = 1.3$)	
		BSM	CollocInfer	BSM	CollocInfer	BSM	CollocInfer
$n = 25$	$\xi_1 = 0.1$	0.098 (*0.001*)	-	0.096 (*0.002*)	-	0.078 (*0.043*)	-
	$\beta_1 = 2.5$	2.531 (*0.105*)	3.819 (*6.845*)	2.560 (*0.412*)	5.054 (*14.530*)	2.317 (*4.687*)	4.727 (*12.370*)
	$\beta_2 = -0.125$	-0.180 (*0.271*)	-0.244 (*0.091*)	-0.216 (*1.035*)	-0.420 (*0.228*)	-0.359 (*9.864*)	-0.385 (*0.215*)
	$^\diamond\sigma^2$	0.064	-	0.244	-	7.054	-
	NAD	100	98	100	96	100	93
$n = 100$	$\xi_1 = 0.1$	0.098 (*0.001*)	-	0.096 (*0.002*)	-	0.078 (*0.046*)	-
	$\beta_1 = 2.5$	2.540 (*0.092*)	5.080 (*15.480*)	2.571 (*0.364*)	5.060 (*15.267*)	2.904 (*9.403*)	5.017 (*14.886*)
	$\beta_2 = -0.125$	-0.193 (*0.213*)	-0.124 (*0.009*)	-0.238 (*0.836*)	-0.124 (*0.009*)	-0.740 (*20.655*)	-0.130 (*0.011*)
	$^\diamond\sigma^2$	0.065	-	0.245	-	6.364	-
	NAD	100	92	100	92	100	94
$n = 500$	$\xi_1 = 0.1$	0.098 (*0.001*)	-	0.095 (*0.002*)	-	0.077 (*0.048*)	-
	$\beta_1 = 2.5$	2.542 (*0.086*)	2.633 (*1.840*)	2.573 (*0.342*)	4.336 (*6.462*)	2.852 (*8.649*)	5.441 (*14.641*)
	$\beta_2 = -0.125$	-0.197 (*0.188*)	-0.091 (*0.046*)	-0.242 (*0.742*)	-0.103 (*0.003*)	-0.662 (*18.647*)	-0.095 (*0.008*)
	$^\diamond\sigma^2$	0.066	-	0.248	-	6.126	-
	NAD	100	83	100	80	100	89

$\diamond$ results reported as multiplied by 10^2

Table 3: Average posterior means (with *MSE* in brackets) for the parameters of the Lotka-Volterra ODE system.

		Noise level					
Sample	Parameters	low $(SNR = 13)$		medium $(SNR = 6.5)$		high $(SNR = 1.3)$	
size		BSM	CollocInfer	BSM	CollocInfer	BSM	CollocInfer
$n = 25$	$\xi_1 = 2.0$	1.996 (*0.001*)	-	1.965 (*0.112*)	-	1.929 (*0.448*)	-
	$\beta_1 = 0.1$	0.101 (*0.001*)	0.095 (*0.001*)	0.101 (*0.001*)	0.093 (*0.001*)	0.091 (*0.001*)	0.863 (*0.001*)
	$\beta_2 = -0.2$	-0.201 (*0.001*)	-0.197 (*0.001*)	-0.197 (*0.001*)	-0.192 (*0.002*)	-0.170 (*0.002*)	-0.183 (*0.008*)
	$^{\diamond}\sigma^2$	0.040	-	2.295	-	12.234	-
	NAD	100	100	100	100	100	100
$n = 100$	$\xi_1 = 2.0$	1.996 (*0.001*)	-	1.965 (*0.112*)	-	1.930 (*0.449*)	-
	$\beta_1 = 0.1$	0.089 (*0.001*)	0.063 (*0.001*)	0.088 (*0.001*)	0.060 (*0.002*)	0.086 (*0.001*)	0.061 (*0.002*)
	$\beta_2 = -0.2$	-0.168 (*0.001*)	-0.140 (*0.004*)	-0.167 (*0.001*)	-0.139 (*0.004*)	-0.162 (*0.002*)	-0.138 (*0.004*)
	$^{\diamond}\sigma^2$	3.650	-	5.392	-	10.507	-
	NAD	100	100	100	100	100	98
$n = 500$	$\xi_1 = 2.0$	1.996 (*0.001*)	-	1.965 (*0.112*)	-	1.930 (*0.447*)	-
	$\beta_1 = 0.1$	0.095 (*0.001*)	0.080 (*0.001*)	0.095 (*0.001*)	0.080 (*0.001*)	0.094 (*0.002*)	0.077 (*0.001*)
	$\beta_2 = -0.2$	-0.182 (*0.001*)	-0.138 (*0.004*)	-0.182 (*0.001*)	-0.138 (*0.004*)	-0.180 (*0.002*)	-0.138 (*0.004*)
	$^{\diamond}\sigma^2$	1.405	-	2.89	-	7.407	-
	NAD	100	100	100	100	100	99

$\diamond$ results reported as multiplied by 10^1

Table 4: Average posterior means (with *MSE* in brackets) for the parameters of the HIV viral fitness ODE system. For CollocInfer β_2 is not estimated, but set to the true value due to stability problems.

Sample size	Parameters	Noise level					
		low ($SNR = 13$)		medium ($SNR = 6.5$)		high ($SNR = 1.3$)	
		BSM	CollocInfer	BSM	CollocInfer	BSM	CollocInfer
$n = 25$	$\xi_1 = 60$	59.879 (*1.35*)	-	59.519 (*21.55*)	-	59.031 (*86.12*)	-
	$\beta_1 = 20$	20.808 (*2.79*)	20.933 (*1.001*)	20.294 (*21.89*)	21.142 (*7.130*)	14.609 (*175.49*)	20.849 (*8.869*)
	$\beta_2 = -0.108$	-0.113 (*0.001*)	-	-0.103 (*0.005*)	-	-0.030 (*0.032*)	-
	$^{\dagger}\beta_3 = -0.095$	-0.106 (*0.001*)	-0.106 (*0.003*)	-0.109 (*0.001*)	-0.109 (*0.001*)	-0.159 (*0.001*)	-0.106 (*0.001*)
	$^{\diamond}\sigma^2$	0.169	-	2.346	-	20.398	-
	NAD	100	84	100	85	100	85
$n = 100$	$\xi_1 = 60$	59.882 (*1.18*)	-	59.539 (*18.82*)	-	59.079 (*75.26*)	-
	$\beta_1 = 20$	21.714 (*3.13*)	19.700 (*1.936*)	21.457 (*4.98*)	19.634 (*2.188*)	20.281 (*9.94*)	19.772 (*3.728*)
	$\beta_2 = -0.108$	-0.117 (*0.001*)	-	-0.115 (*0.001*)	-	-0.106 (*0.001*)	-
	$^{\dagger}\beta_3 = -0.095$	-0.103 (*0.001*)	-0.090 (*0.001*)	-0.103 (*0.001*)	-0.089 (*0.010*)	-0.100 (*0.001*)	-0.090 (*0.009*)
	$^{\diamond}\sigma^2$	0.640	-	2.748	-	9.473	-
	NAD	100	98	100	98	100	97
$n = 500$	$\xi_1 = 60$	59.882 (*1.18*)	-	59.538 (*18.84*)	-	59.078 (*75.35*)	-
	$\beta_1 = 20$	21.012 (*1.16*)	19.612 (*1.791*)	21.040 (*3.17*)	19.648 (*1.412*)	20.936 (*8.99*)	19.499 (*2.437*)
	$\beta_2 = -0.108$	-0.114 (*0.001*)	-	-0.113 (*0.001*)	-	-0.112 (*0.001*)	-
	$^{\dagger}\beta_3 = -0.095$	-0.100 (*0.001*)	-0.088 (*0.001*)	-0.100 (*0.001*)	-0.089 (*0.001*)	-0.101 (*0.001*)	-0.087 (*0.001*)
	$^{\diamond}\sigma^2$	0.437	-	2.312	-	8.326	-
	NAD	100	99	100	99	100	100

† results reported as multiplied by 10^2

⋄ results reported as multiplied by $10^{(-1)}$

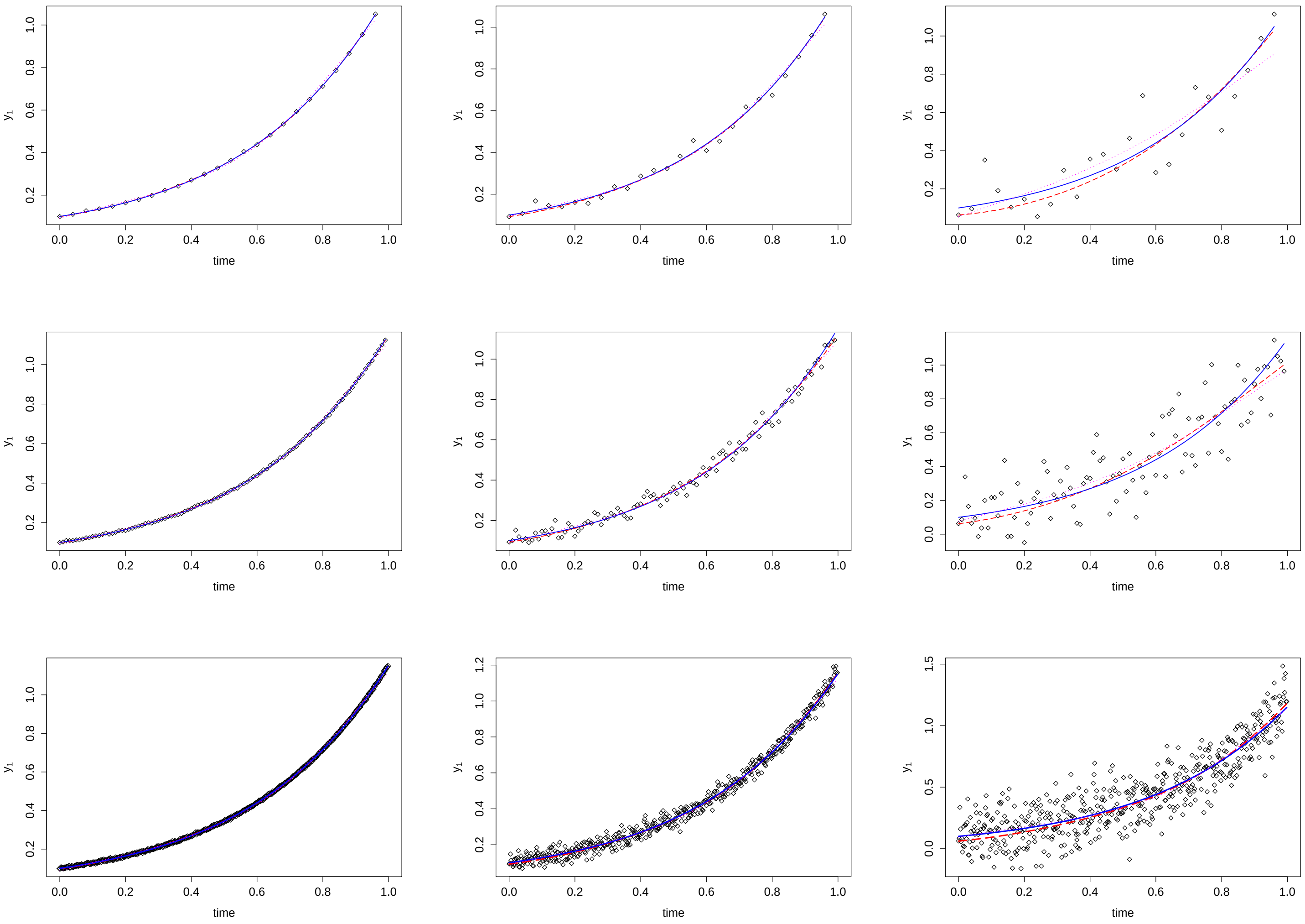

Figure 4: Logistic population growth simulations: observed noisy data (*dots*), smoothing spline (*dotted* line), true solution of the ODE system (*solid* line) and reconstructed solution (*long-dashed* line) for the variable $\mathbf{x}_{\cdot 1}$. Different sample sizes ($n = 25, 100, 500$) from the top to the bottom and noise levels (low, medium and high) from left to right

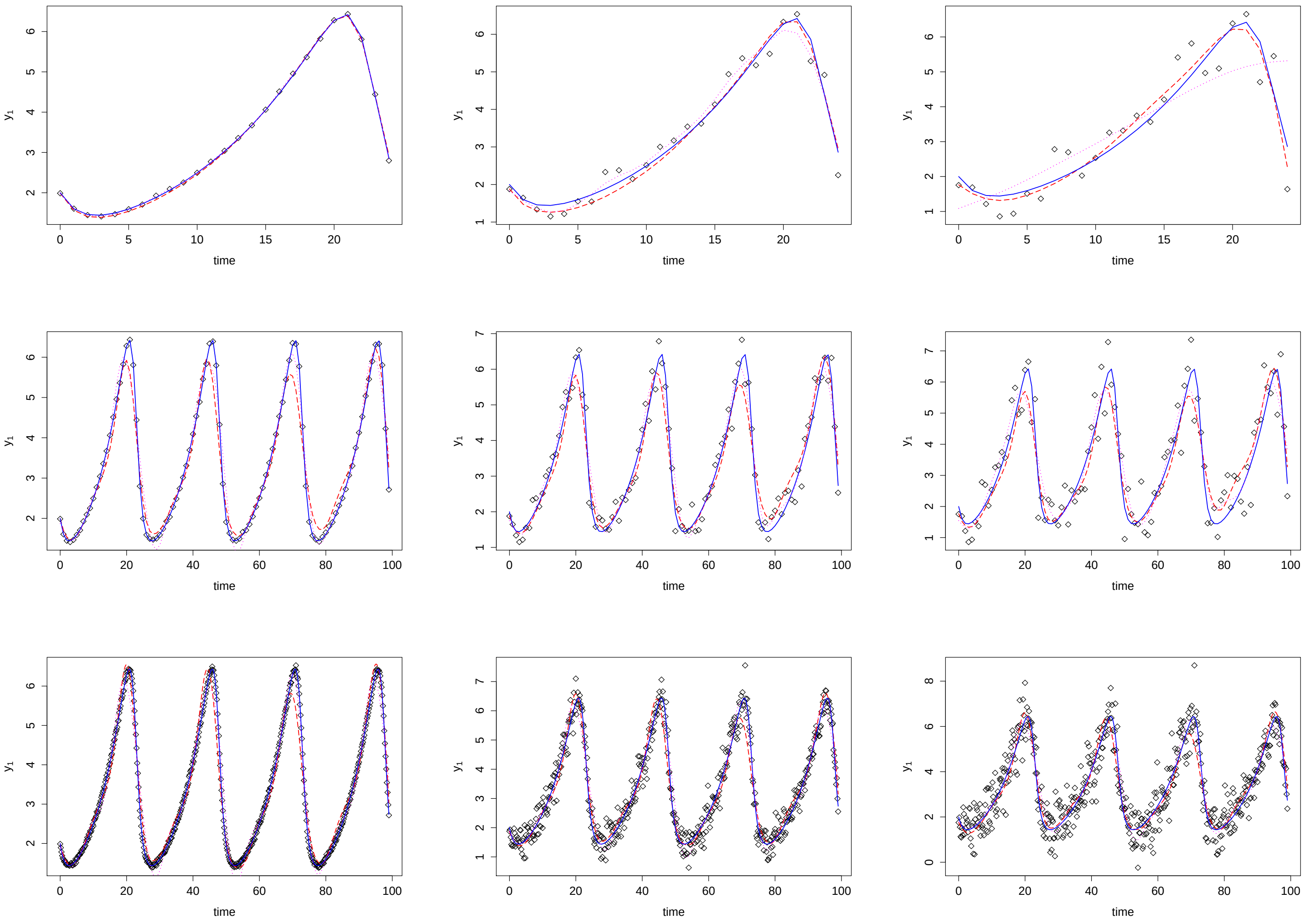

Figure 5: Lotka-Volterra simulations: observed noisy data (*dots*), smoothing spline (*dotted* line), true solution of the ODE system (*solid* line) and reconstructed solution (*long-dashed* line) for the first variable $\mathbf{x}_{\cdot 1}$. Different sample sizes ($n = 25, 100, 500$) from the top to the bottom and noise levels (low, medium and high) from left to right

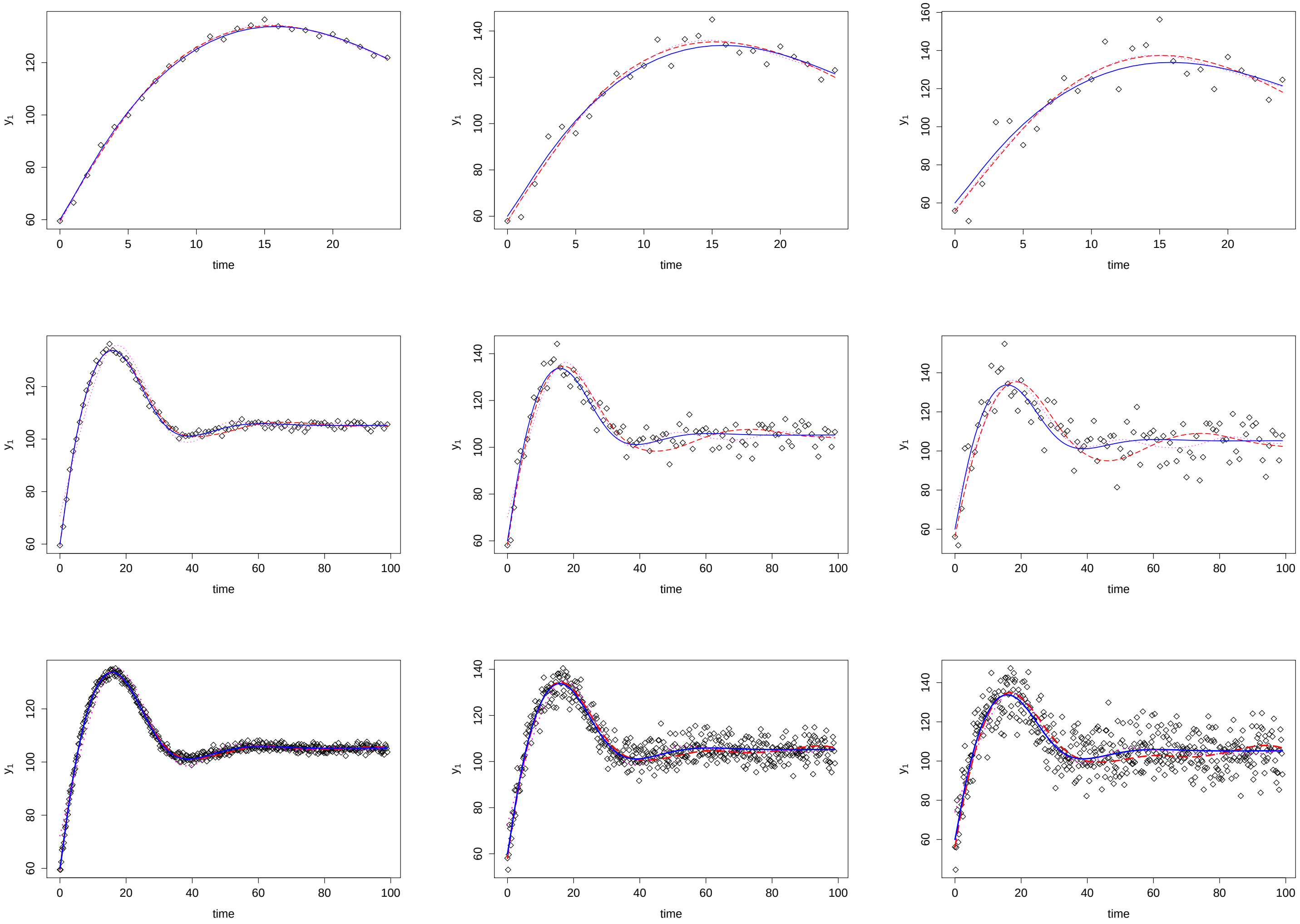

Figure 6: HIV viral fitness simulations: observed noisy data (*dots*), smoothing spline (*dotted* line), true solution of the ODE system (*solid* line) and reconstructed solution (*long-dashed* line) for the first variable $\mathbf{x}_{\cdot 1}$. Different sample sizes ($n = 25, 100, 500$) from the top to the bottom and noise levels (low, medium and high) from left to right

A Appendix: Full conditionals for MCMC implementation

A.1 Gibbs samplers for the first step (*smooth*)

At iteration $(l+1)$, for each $k = 1, \dots, p$, we sample new values for $\boldsymbol{\theta}_k$ from its full conditional distribution

$$p(\boldsymbol{\theta}_k^{(l+1)} | \dots) \propto p_{\text{smth}}\Big(\mathbf{y}_k | \, \boldsymbol{\theta}_k, \sigma_k^{2^{(l)}}\Big) \, p\Big(\boldsymbol{\theta}_k | \, \lambda_{\theta_k}^{(l)}\Big) \,,$$

which is a multivariate Normal distribution with the following mean vector and variance-covariance matrix

$$m_{\theta_k} = V_{\theta_k} \Psi_k \frac{\mathbf{y}_k}{\sigma_k^{2^{(l)}}}$$

$$V_{\theta_k} = \left[S_{\theta_k} \lambda_{\theta_k}^{(l)} + \frac{\Psi_k^\top \Psi_k}{\sigma_k^{2^{(l)}}} \right]^{-1}$$

where Ψ_k is the basis matrix of the corresponding spline for $\mathbf{x}_k$.

A.2 Gibbs samplers for the second step (*match*)

At iteration $(l+1)$, we sample new values for $\boldsymbol{\beta}$ from its full conditional distribution

$$p(\boldsymbol{\beta}^{(l+1)} | \dots) \propto p_{\text{regr}}\Big(\mathbf{y}_k | \, \xi_k^{(l)}, \boldsymbol{\beta}, \boldsymbol{\Theta}^{(l)}, \sigma_k^{2^{(l)}}\Big) \, p\Big(\boldsymbol{\beta} | \, \lambda_\beta^{(l)}\Big)$$

which is a multivariate Normal distribution with the following mean vector and variance-covariance matrix

$$m_\beta = V_\beta \boldsymbol{H}_{\mathbf{x}_\theta}^{(l)} \frac{\tilde{\mathbf{y}}_k}{\sigma_k^{2^{(l)}}}$$

$$V_\beta = \left[S_\beta \lambda_\beta^{(l)} + \frac{\boldsymbol{H}_{\mathbf{x}_\theta}^{(l)^\top} \boldsymbol{H}_{\mathbf{x}_\theta}^{(l)}}{\sigma_k^{2^{(l)}}} \right]^{-1}$$

where $\tilde{\mathbf{y}}_k = \mathbf{y}_k - \xi_1^{(l)} \mathbf{1}_n$ and $\boldsymbol{H}_{\mathbf{x}_\theta}^{(l)}$ needs to be computed at each iteration by numerically integrating the updated vectors $\mathbf{x}_{\cdot k}^{(l)}$ using the trapezoidal rule. The initial condition $\xi_k^{(l+1)}$ is sampled from a Normal distribution with expected value $m_\xi = y_{1k} - \boldsymbol{H}_{1,\mathbf{x}_\theta}^{(l)} \boldsymbol{\beta}^{(l)}$ and variance $V_\xi = \sigma_k^{2^{(l)}}$. As for the penalizing term λ_β its full conditional distribution is a Gamma distribution with shape and rate parameters equal to

$$s_{\lambda_\beta} = \frac{b}{2} + \alpha_\beta$$

$$r_{\lambda_\beta} = \frac{\boldsymbol{\beta}_k^{(l)^\top} S_\beta \boldsymbol{\beta}_k^{(l)}}{2} + \gamma_\beta,$$

where b is the number of elements in $\boldsymbol{\beta}$ and $(\alpha_\beta, \gamma_\beta)$ the hyperparameters governing its prior distribution. Finally, we sample $\sigma_k^{2^{(l+1)}}$ from an Inverse Gamma distribution with parameters

$$s_{\sigma^2} = n$$

$$r_{\sigma^2} \;=\; \frac{\left(\tilde{\mathbf{y}}_k - \boldsymbol{H}_{\mathbf{x}_\theta}^{(l)}\boldsymbol{\beta}^{(l)}\right)^\top \left(\tilde{\mathbf{y}}_k - \boldsymbol{H}_{\mathbf{x}_\theta}^{(l)}\boldsymbol{\beta}^{(l)}\right)}{2} + \frac{\left(\mathbf{y}_k - \mathbf{x}_k^{(l)}\right)^\top \left(\mathbf{y}_k - \mathbf{x}_k^{(l)}\right)}{2}.$$

A.3 Synthetic ODE system specifications

The model we fit, which is the true ODE system augmented with misspecified regressing functions $h_3(\mathbf{x})$, is given by

$$\begin{cases}
\mathbf{x}_1'(t) = \alpha_1\mathbf{x}_1(t) - \beta_1\mathbf{x}_{10}(t)\sin(\mathbf{x}_2(t)) + \gamma_1\mathbf{x}_5(t) \\
\mathbf{x}_2'(t) = -\alpha_2\mathbf{x}_2(t) + \beta_2\mathbf{x}_1(t)\sin(\mathbf{x}_3(t)) + \gamma_2\mathbf{x}_4(t) \\
\mathbf{x}_3'(t) = \alpha_3\mathbf{x}_3(t) - \beta_3\mathbf{x}_2(t)\cos(\mathbf{x}_4(t)) + \gamma_3\mathbf{x}_6(t) \\
\mathbf{x}_4'(t) = -\alpha_4\mathbf{x}_4(t) + \beta_4\mathbf{x}_3(t)\cos(\mathbf{x}_5(t)) + \gamma_4\mathbf{x}_8(t) \\
\mathbf{x}_5'(t) = \alpha_5\mathbf{x}_5(t) - \beta_5\mathbf{x}_4(t)\cos(\mathbf{x}_6(t)) + \gamma_5\mathbf{x}_1(t) \\
\mathbf{x}_6'(t) = -\alpha_6\mathbf{x}_6(t) + \beta_6\mathbf{x}_5(t)\cos(\mathbf{x}_7(t)) + \gamma_6\mathbf{x}_4(t) \\
\mathbf{x}_7'(t) = \alpha_7\mathbf{x}_7(t) - \beta_7\mathbf{x}_6(t)\cos(\mathbf{x}_8(t)) + \gamma_7\mathbf{x}_2(t) \\
\mathbf{x}_8'(t) = -\alpha_8\mathbf{x}_8(t) + \beta_8\mathbf{x}_7(t)\sin(\mathbf{x}_9(t)) + \gamma_8\mathbf{x}_{10}(t) \\
\mathbf{x}_9'(t) = \alpha_9\mathbf{x}_9(t) - \beta_9\mathbf{x}_8(t)\sin(\mathbf{x}_{10}(t)) + \gamma_9\mathbf{x}_4(t) \\
\mathbf{x}_{10}'(t) = -\alpha_{10}\mathbf{x}_{10}(t) + \beta_{10}\mathbf{x}_9(t)\sin(\mathbf{x}_1(t)) + \gamma_{10}\mathbf{x}_3(t).
\end{cases}$$

If $\gamma_j = 0\ \forall j$ the fitted model reverts to the true data generating process. Values for the parameters of the system are: $\beta_1 = \beta_4 = \beta_6 = \beta_7 = \beta_{10} = 0.08$; $\beta_2 = \beta_5 = 0.07$; $\beta_3 = \beta_8 = \beta_9 = 0.06$ $\alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = \alpha_9 = \alpha_{10} = 0.05$; $\alpha_8 = 0.04$; $\alpha_1 = 0.03$; $\alpha_7 = 0.02$.. The initial conditions are $\xi_j = 0.10$ for all $j = 1, \ldots, p$. In the simulation study, we run our MCMC algorithm for 10000 iterations with a 5000 burn-in window.