Fenrir: Physics-Enhanced Regression for Initial Value Problems

Filip Tronarp [*] [1] **Nathanael Bosch** [*] [1] **Philipp Hennig** [1] [2]

Abstract

We show how probabilistic numerics can be used to convert an initial value problem into a Gauss–Markov process parametrised by the dynamics of the initial value problem. Consequently, the often difficult problem of parameter estimation in ordinary differential equations is reduced to hyperparameter estimation in Gauss–Markov regression, which tends to be considerably easier. The method's relation and benefits in comparison to classical numerical integration and gradient matching approaches is elucidated. In particular, the method can, in contrast to gradient matching, handle partial observations, and has certain routes for escaping local optima not available to classical numerical integration. Experimental results demonstrate that the method is on par or moderately better than competing approaches.

1. Introduction

Consider the following initial value problem (IVP)

$$\frac{\mathrm{d}}{\mathrm{d}t}\varphi_\theta(t) = f_\theta\left(t, \varphi_\theta(t)\right), \qquad t \in [0, T], \tag{1}$$

where the vector field $f_\theta\colon [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$ and the initial condition $\varphi_\theta(0) = y_0(\theta)$ are both parametrised by θ. In this article, the concern lies in estimating θ from noisy measurements of the following form

$$u(t) = H^{\mathsf{T}}\varphi_\theta(t) + v(t), \quad v(t) \sim \mathcal{N}(0, R_\theta), \tag{2}$$

where $t \in \mathbb{T}_{\mathsf{D}} \subset [0, T]$ is the finite set of measurement nodes and H is a measurement matrix of appropriate dimension. This is a ubiquitous problem in science and engineering. Examples include ecology (Benson, 1979), pharmacokinetics (Gelman et al., 1996), process engineering (Åström & Eykhoff, 1971), and brain imaging (Friston, 2002).

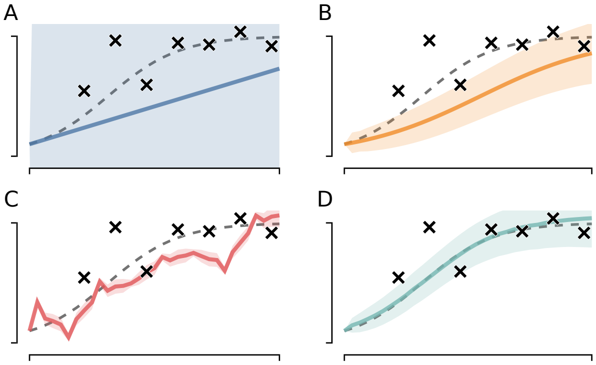


Figure 1. **From an uninformed prior to a calibrated posterior.** Starting from a standard Gauss–Markov prior (A), Fenrir first computes a physics-enhanced prior with probabilistic numerics (B), and then a posterior via Gauss–Markov regression (C). By maximizing the marginal likelihood, we obtain a calibrated posterior and parameter estimates for the underlying dynamical system (D). The data generating model is the logistic equation, which corresponds to the vector field $f(t, y) = ry(1 - y)$.

The likelihood functional $\mathcal{L}_{\mathsf{D}}$, evaluated at some function y, is given by

$$\mathcal{L}_{\mathsf{D}}(R_\theta, y) = \prod_{t \in \mathbb{T}_{\mathsf{D}}} \mathcal{N}\left(u(t); H^{\mathsf{T}}y(t), R_\theta\right),$$

and the marginal likelihood $\mathcal{M}$ of some parameter θ can be expressed by evaluating $\mathcal{L}_{\mathsf{D}}$ at the corresponding solution φ_θ according to

$$\mathcal{M}(\theta) = \mathcal{L}_{\mathsf{D}}\left(R_\theta, \varphi_\theta\right).$$

The parameter θ may be estimated by maximising $\mathcal{M}$. A persistent challenge in likelihood-based inference in initial value problems is the fact that φ_θ, and therefore the likelihood, are intractable (Bard, 1974).

A standard appproach to approximating the likelihood is based on solving the IVP numerically (Hairer et al., 1987). However, in optimisation-based inference it has been observed that this leads to many local optima (Cao et al., 2011), and can lead to divergence of the optimiser (Dass et al., 2017). On the other hand, slow convergence and poor mixing has been observed for Monte Carlo-based inference

[*] Equal contribution [1] Department of Computer Science, University of Tübingen, Tübingen, Germany [2] Max–Planck Institute for Intelligent Systems, Tübingen, Germany. Correspondence to: Filip Tronarp <filip.tronarp@uni-tuebingen.de>, Nathanael Bosch <nathanael.bosch@uni-tuebingen.de>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

(Alahmadi et al., 2020), which have led some authors to favour likelihood-free methods (Toni et al., 2009). Another alternative is gradient matching (Voit, 2000) with splines (Varah, 1982; Gugushvili & Klaassen, 2012) or Gaussian processes (Calderhead et al., 2009; Dondelinger et al., 2013; Gorbach et al., 2017; Wenk et al., 2020).

1.1. Contribution

In the present work, a probabilistic numerics approach is developed for computing the marginal likelihood. Probabilistic numerics aims at producing probability measures for solutions of numerical problems, thus giving a probabilistic description of the numerical error (Hennig et al., 2015; Oates & Sullivan, 2019).

The marginal likelihood may be viewed as $\mathcal{L}_{\mathsf{D}}$ integrated against a Dirac measure located at φ_θ according to

$$\mathcal{M}(\theta) = \int \mathcal{L}_{\mathsf{D}}(R_\theta, y)\delta(y - \varphi_\theta)\,\mathrm{d}y. \tag{3}$$

While this representation is not immediately advantageous, it is instructive for understanding the probabilistic numerics approach. Namely, it produces an approximation to the Dirac measure, giving the following approximate marginal likelihood

$$\widehat{\mathcal{M}}_N(\theta, \kappa) = \int \mathcal{L}_{\mathsf{D}}(R_\theta, y)\widehat{\delta}_N(y \mid \theta, \kappa)\,\mathrm{d}y, \tag{4}$$

where $\widehat{\delta}_N$ is the output of a suitably chosen probabilistic numerical method, which is parametrised by κ. It should be noted that $\mathcal{L}_{\mathsf{D}}$ only depends on point evaluations of y on the grid $\mathbb{T}_{\mathsf{D}}$. Therefore, it is sufficient to operate on the finite dimensional distributions of $\widehat{\delta}_N$ to compute $\widehat{\mathcal{M}}_N$.

Kersting et al. (2020a) has previously used the representation (4) and approximated its gradients in combination with low order explicit solvers, at a cost of $O(N^3)$.

The aim of this article is to show how both $\widehat{\delta}_N$ and $\widehat{\mathcal{M}}_N$ can be computed efficiently for general probabilistic solvers, at a cost of $O(N)$. The method consists of two parts:

1. Efficiently construct a Gauss–Markov representation of $\widehat{\delta}_N(y \mid \theta, \kappa)$ using probabilistic numerics.
2. Compute $\widehat{\mathcal{M}}_N(\theta, \kappa)$ and its derivatives via Gauss–Markov regression and automatic differentiation.

The first step essentially takes the initial value problem and produces a *physics-enhanced* Gauss–Markov prior. The second step utilises this prior in standard Gauss–Markov regression to estimate parameters and reconstruct the trajectory (Särkkä & Solin, 2019). Therefore, the method is called Physics-enhanced regression in initial value problems, or *Fenrir* for short. Here physics is used to refer to any mechanistic information pertaining to the dynamics of the data generating process. The method is illustrated in Figure 1.

The rest of the article is organised as follows. Probabilistic numerical solvers are reviewed in Section 2. In Section 3 it is shown how to use probablistic numerics to construct a physics-enhanced Gauss–Markov prior for initial value problems, thus reducing the marginal likelihood to Gauss–Markov regression. Related work is discussed in Section 4, which is followed by experimental results in Section 5. Finally, concluding remarks are given in Section 6.

2. Probabilistic Numerical IVP Solvers

In the Bayesian formulation, an IVP solver is completely specified by a prior and the definition of the data, on which it is conditioned. The latter is obtained by means of an information operator (Cockayne et al., 2019). For constructing a probabilistic numerical solver, we follow the account of Tronarp et al. (2019b; 2021).

2.1. Prior Specification

The probabilistic numerics prior is defined as the output of the following stochastic state-space model

$$\mathrm{d}x(t) = Ax(t)\,\mathrm{d}t + \sqrt{\kappa}B\,\mathrm{d}w(t), \quad x(0) = x_\theta^\dagger, \tag{5a}$$

$$y^{(m)}(t) = \mathrm{E}_m^{\mathsf{T}} x(t), \quad m = 0, 1, \dots, \nu, \tag{5b}$$

where $x \in \mathbb{R}^{d(\nu+1)}$ models the solution and its ν first derivatives and E_m are selection matrices for the mth derivative of the prior model for the solution of (1), which is denoted by y. Furthermore, $x_\theta^\dagger$ denotes the initial condition of x, $A \in \mathbb{R}^{d(\nu+1)\times d(\nu+1)}$ and $B \in \mathbb{R}^{d(\nu+1)}$ are model matrices and w is a standard Wiener process in $\mathbb{R}^d$ (Øksendal, 2003).

The state x is a Markov process by construction, with transition density given by (Särkkä & Solin, 2019)

$$\begin{aligned} \Phi(h) &= e^{Ah}, \\ Q(h) &= \int_0^h \Phi(h-\tau)BB^{\mathsf{T}}\Phi^{\mathsf{T}}(h-\tau)\,\mathrm{d}\tau, \\ x(t+h) \mid x(t) &\sim \mathcal{N}\big(x(t+h); \Phi(h)x(t), \kappa Q(h)\big), \end{aligned}$$

which facilitates fast computation for the probabilistic solver and our subsequent marginal likelihood approximation. Additional details on priors for probabilistic solutions of initial value problems can be found in Appendix A.1.

2.2. Data Model

In order to define a data model for probabilistic numerical solvers, a grid

$$\mathbb{T}_{\mathsf{PN}} = \{t_n\}_{n=1}^N \subset [0, T],$$

needs to be coupled with an information operator. The canonical information operator for initial value problems is given by (Tronarp et al., 2021)

$$\mathcal{Z}_\theta[x](t) = \mathrm{E}_1^\mathsf{T} x(t) - f_\theta(t, \mathrm{E}_0^\mathsf{T} x(t)), \tag{6}$$

but there are alternatives that, for instance, also take geometric invariants into account (Bosch et al., 2021b).

Note that $\mathcal{Z}$ map solutions of the initial value problem to the zero function, which is a known value. In fact, the set of functions starting at $y_0(\theta)$ which are mapped to the zero function by $\mathcal{Z}$ constitutes the set of solutions to the initial value problem (Arnol'd, 1992).[1] An appropriate data model for a probabilistic numerical solver is thus given by

$$z(t) = \mathcal{Z}_\theta[x](t) = 0, \quad t \in \mathbb{T}_{\mathsf{PN}}, \tag{7}$$

where $z(t) = 0$ is enforced only on the chosen grid as to arrive at a practical algorithm.

It should be noted that the grid $\mathbb{T}_{\mathsf{PN}}$ does not need to be specified a priori but can be constructed adaptively to control the solution error (Schober et al., 2019; Bosch et al., 2021a).

2.3. Initial Value Problem Solvers as Non-linear Gauss–Markov Regression

The prior (5), data model (6), and data definition (7) define a non-linear Gauss–Markov regression problem according to Tronarp et al. (2019b)

$$\begin{aligned} x(t_n) \mid x(t_{n-1}) &\sim \mathcal{N}\big(\Phi(\Delta_n) x(t_{n-1}), \kappa Q(\Delta_n)\big), && \text{(8a)} \\ z(t_n) \mid x(t_n) &\sim \mathcal{N}\big(\mathrm{E}_1^\mathsf{T} x(t) - f_\theta(t, \mathrm{E}_0^\mathsf{T} x(t)), 0\big), && \text{(8b)} \\ z(t_n) &\coloneqq 0, && \text{(8c)} \end{aligned}$$

where $\Delta_n = t_n - t_{n-1}$ is the step-size of the nth step, $x(t_0) = x_\theta^\dagger$ by convention, and $\mathcal{N}(\cdot\,, 0)$ denotes the Dirac distribution. The probablistic numerical solver for (1) associated with the prior (5) and the data (7) is on the grid $\mathbb{T}_{\mathsf{PN}}$ given by

$$\begin{aligned} \gamma_N(t_{1:N}, x_{1:N} \mid \theta, \kappa) &= c^{-1}(\theta, \kappa) \\ &\times \prod_{n=1}^{N} \mathcal{N}\big(x_n; \Phi(\Delta_n) x_{n-1}, \kappa Q(\Delta_n)\big) \\ &\times \prod_{n=1}^{N} \delta\big(\mathrm{E}_1^\mathsf{T} x_n - f_\theta(t_n, \mathrm{E}_0^\mathsf{T} x_n)\big), \end{aligned} \tag{9}$$

where $c(\theta, \kappa)$ is a norming constant. Due to the potential non-linearity of the vector field, this object is generally intractable. However, when the vector field is linear, say

$$f_\theta(t, y) = L_\theta(t) y + b_\theta(t), \tag{10}$$

then the densities of the time marginals can be computed efficiently via Kalman filtering and Rauch–Tung–Striebel smoothing (Kalman, 1960; Rauch et al., 1965).

This fact is exploited for approximate inference when the vector field is non-linear as well. Indeed several linearisation approaches have been employed (Schober et al., 2019; Tronarp et al., 2019b; 2021), which have been demonstrated to yield accurate solvers both empirically (Schober et al., 2019; Bosch et al., 2021a; Krämer & Hennig, 2020) and theoretically (Kersting et al., 2020b; Tronarp et al., 2021).

2.4. Initial Value Problem Solvers as Kalman Filtering

The Kalman filtering recursion for (8) when the vector field is affine as in (10), recursively computes the densities

$$\pi(x(t_n) \mid z(t_{1:n})) = \mathcal{N}(\mu_\theta(t_n), \Sigma_\theta(t_n)), \tag{11}$$

which are the time marginals conditioned on all past data up to the present. The recursion is initialised by setting $\mu_\theta(t_0) = x_\theta^\dagger, \Sigma_\theta(t_0) = 0$, and then alternates between prediction and update.

- Prediction:

$$\begin{aligned} \mu_\theta(t_n^-) &= \Phi(\Delta_n) \mu_\theta(t_{n-1}), \\ \Sigma_\theta(t_n^-) &= \Phi(\Delta_n) \Sigma_\theta(t_{n-1}) \Phi^\mathsf{T}(\Delta_n) + Q(\Delta_n). \end{aligned}$$

- Update:

$$\begin{aligned} C_\theta(t_n) &= \mathrm{E}_1 - \mathrm{E}_0 L_\theta^\mathsf{T}(t_n), \\ S_\theta(t_n) &= C_\theta^\mathsf{T}(t_n) \Sigma_\theta(t_n^-) C_\theta(t_n), \\ K_\theta(t_n) &= \Sigma_\theta(t_n^-) C_\theta^\mathsf{T}(t_n) S_\theta^{-1}(t_n), \\ e_\theta(t_n) &= b_\theta(t_n) - C_\theta^\mathsf{T}(t_n) \mu_\theta(t_n^-), \\ \mu_\theta(t_n) &= \mu_\theta(t_n^-) + K_\theta(t_n) e(t_n), \\ \Sigma_\theta(t_n) &= \Sigma_\theta(t_n^-) - K_\theta(t_n) S_\theta(t_n) K_\theta^\mathsf{T}(t_n). \end{aligned}$$

The following parameters can be computed from the outputs of the Kalman filter

$$G_\theta(t_{n-1}) = \Sigma_\theta(t_{n-1}) \Phi^\mathsf{T}(\Delta_n) \Sigma_\theta^{-1}(t_n^-), \tag{12a}$$

$$P_\theta(t_{n-1}) = \Sigma_\theta(t_{n-1}) - G_\theta(t_{n-1}) \Sigma_\theta(t_n^-) G_\theta^\mathsf{T}(t_{n-1}). \tag{12b}$$

They are used for the smoothing recursion and the representation of the probabilistic numerics posterior.

3. Fenrir

In this section, it is shown that the probabilistic numerical solver yields a Gauss–Markov process approximation to (1). Consequently, inference given measurements (2) reduces to a Gauss–Markov regression problem with a *physics-enhanced prior* as determined by the probabilistic solver.

[1] Typically, we assume that the vector field is regular enough for there to be a unique solution of the initial value problem.

3.1. Probabilistic Numerical IVP Solutions as Gauss–Markov Processes

Linearising the vector field allows for approximate computation of the time marginal densities via the Rauch–Tung–Striebel smoother. These linearisations imply a Gauss–Markov representation of the approximate posterior, which in fact is used in the Bayesian derivation of the smoothing algorithm (Särkkä, 2013, c.f. proof of theorem 8.2). The following result lie at the heart of our method.

Proposition 3.1 (Gauss–Markov representation of the probabilistic solver)**.** *The restriction of the probabilistic numerics posteriors to the grid $\mathbb{T}_{\mathsf{PN}}$ admit the following representation*

$$\widehat{\gamma}_N(t_{1:N}, x_{1:N} \mid \theta, \kappa) = \mathcal{N}\big(x_N; \xi_\theta(t_N), \kappa\Lambda_\theta(t_N)\big) \prod_{n=N-1}^{1} \mathcal{N}\big(x_n; G_\theta(t_n)x_{n+1} + \zeta_\theta(t_n), \kappa P_\theta(t_n)\big), \tag{13}$$

where $\xi_\theta(t_N) = \mu_\theta(t_N)$, $\Lambda_\theta(t_N) = \Sigma_\theta(t_N)$,

$$\zeta_\theta(t_n) = \mu_\theta(t_n) - G_\theta(t_n)\mu_\theta(t_{n+1}^-),$$

and (G, P) *are given by* (12)*.*

For completeness, a detailed derivation of proposition 3.1 is given in Appendix A.2. Note that $\widehat{\gamma}_N$ is represented as a Gauss–Markov process running backwards in time. It represents a probabilistic approximation to the solution of the IVP and its derivatives, in terms of a conditional distribution given numerical data (7) and the parameter θ.

3.2. Inference in IVPs as Gauss–Markov Regression

In the previous section, the approximate Dirac $\widehat{\delta}_N(y \mid \theta)$ was implicitly defined through γ_N in (9). The purpose here is to turn this into an implementable algorithm for approximating the marginal likelihood. For ease of notation it is assumed that $\mathbb{T}_{\mathsf{D}} \subset \mathbb{T}_{\mathsf{PN}}$, in which case,

$$\begin{aligned}\widehat{\mathcal{M}}_N(\theta, \kappa) &= \int \mathcal{L}_{\mathsf{D}}(\theta, y)\widehat{\delta}_N(y \mid \theta, \kappa)\,\mathrm{d}y \\ &= \int \mathcal{L}_{\mathsf{D}}(R_\theta, \mathrm{E}_0^{\mathsf{T}}x)\gamma_N(t_{1:N}, x_{1:N} \mid \theta, \kappa)\,\mathrm{d}x_{1:N}.\end{aligned}$$

Additionally, the calibration parameter κ is also included in the marginal likelihood approximation. In practice, γ_N is replaced by its approximation $\widehat{\gamma}_N$ in (13). This results in the following approximation to the marginal likelihood

$$\begin{aligned}\widehat{\mathcal{M}}_N(\theta, \kappa) = \int \prod_{t_n \in \mathbb{T}_{\mathsf{D}}} &\mathcal{N}(u(t_n); H^{\mathsf{T}}\mathrm{E}_0^{\mathsf{T}}x_n, R_\theta) \\ &\times \widehat{\gamma}_N(t_{1:N}, x_{1:N} \mid \theta, \kappa)\,\mathrm{d}x_{1:N}.\end{aligned} \tag{14}$$

Consequently, the problem of computing the marginal likelihood and trajectory estimates is reduced to inference in the following linear state-space model

$$x(t_N) \sim \mathcal{N}(\xi(t_N), \kappa\Lambda(t_N)), \tag{15a}$$

$$x(t_n) \mid x(t_{n+1}) \sim \widehat{\gamma}_N(x(t_n) \mid x(t_{n+1}), \theta, \kappa), \tag{15b}$$

$$u(t) \mid x(t) \sim \mathcal{N}(H^{\mathsf{T}}\mathrm{E}_0^{\mathsf{T}}x(t), R_\theta), \quad t \in \mathbb{T}_{\mathsf{D}}, \tag{15c}$$

where the backwards transition densities can be read from (13). Therefore, estimating the trajectory of the solution (1) can also be done via Kalman filtering and smoothing. Furthermore, the marginal likelihood approximation can be computed via the Kalman filter through the prediction error decomposition (Schweppe, 1965). Complete details on how to compute trajectory estimates and marginal likelihoods in (15) are given in Appendix B.

Computational complexity The computation of $\widehat{\gamma}_N$ and $\widehat{\mathcal{M}}_N$ can be implemented with Gauss–Markov regression with a state dimension of $d(\nu+1)$. Therefore, assuming the measurement dimension is smaller, the computational complexity of the method is $O(Nd^3(\nu+1)^3)$. That is, it is linear in the number of data points, in contrast to cubic complexity for standard Gaussian process regresison. Further speed-ups may be obtainable by exploiting structural simplifications for certain probabilistic solvers (Krämer et al., 2021).

Hyperparameter estimation The present method provides a marginal likelihood (14); its derivatives can be computed with automatic differentiation. Consequently, Fenrir interacts with various inference methods, such as gradient-based optimisation or Markov Chain Monte Carlo, in a plug-and-play fashion. In this paper, the maximum likelihood approach is examined.

Model selection The marginal likelihood approximation (14) confers other benefits than providing a cost function for parameter inference. Namely, the possibility for a probabilistically motivated model comparisons, such as likelihood ratio testing for nested models (King, 1998), or via various information criteria (Akaike, 1974; Stoica & Selen, 2004).

4. Related Work: A Tale of Three Approaches

Three different approaches to parameter estimation in initial value problems can be discerned, namely (a) numerical integration, (b) gradient matching, and (c) probabilistic numerics. In order to get a comprehensive lay of the land of parameter estimation in ordinary differnetial equations, these approaches are reviewed in this section. Particular care is taken to highlighting similarities and differences.

4.1. Classical Numerical Integration

The traditional approach is to estimate the parameters via non-linear regression (Biegler et al., 1986), where the correct solution to (1) is replaced by a numerical approximation, say Runge–Kutta (Hairer et al., 1987). Thus the marginal likelihood approximation reads

$$\widehat{\mathcal{M}}_N(\theta) = \int \prod_{t_n \in \mathbb{T}_{\mathrm{D}}} \mathcal{N}(u(t_n); H^{\top} y_n, R_\theta) \times \prod_{t_n \in \mathbb{T}_{\mathrm{D}}} \delta(y_n - \hat{\varphi}_\theta(t_n)) \, \mathrm{d}y_{1:N}. \tag{16}$$

That is, likelihood computation via numerical integration computes the Dirac approximation $\widehat{\delta}_N$ in (4) by approximating the location of the Dirac in (3).

4.2. Gradient Matching

The main idea of gradient matching is to decompose the inference procedure into two steps:

1. Fit a curve $\hat{y}(t)$ to the data $u(t), t \in \mathbb{T}_{\mathrm{D}}$.

2. Estimate the parameter θ by minimising the deviation from the differential equation: $\dot{\hat{y}}(t) - f_\theta(t, \hat{y}(t))$.

This procedure is vaguely formulated, purposely so. Indeed, different alternatives for these steps have surfaced throughout the years.

Spline smoothing The first approach was to implement the curve fitting step with splines (Varah, 1982) or kernel regression (Gugushvili & Klaassen, 2012), whereafter the gradient matching step is posed as a non-linear least squares problem. Another variant is to couple the curve fitting step with the gradient matching step, resulting both in higher accuracy and higher computational cost (Ramsay et al., 2007).

Gaussian process regression The effort to formulate gradient matching probabilistically was spear-headed by Calderhead et al. (2009), where Gaussian process regression is combined with a product of experts approach. This method was improved upon by Dondelinger et al. (2013) via joint sampling for GP and ODE parameters. It was subsequently shown that a mean-field formulation can offer computational speed-ups (Gorbach et al., 2017).

In search for a generative model There has been effort put to formulating Gaussian process-based gradient matching as inference in a generative model. First by Barber & Wang (2014), who instead formulate a model directly linking state derivatives to measurements. However, their approach suffers from identifiability problems, as demonstrated by Macdonald et al. (2015). It was later demonstrated by Wenk et al. (2019) that identifiability issues are also present for the product of experts approach. They propose to resolve this issue by formulating an alternative model; this approach was pursued further by Wenk et al. (2020).

4.3. Probabilistic Numerics

Relation to gradient matching It might be tempting to interpret the probabilistic numerics approach as a variant of gradient matching. But gradient matching fits a curve to the data and then the differential operator to the curve, while for probabilistic numerics the order of operation is reversed:

1. Fit a curve by attempting to satisfy the differential equation at a finite set of points.

2. Fit the parameters of the differential operator by using the aforementioned curve and the data likelihood.

The first step is implemented by probabilistic numerics, resulting in a *physics-enhanced* Gaussian process prior, whereas the second step reduces to Gauss–Markov regression. By directly incorporating the physics of the problem into the prior, it is ensured that inference is done in a well-posed probability model. Consequently, issues regarding model specification and identifiability (Macdonald et al., 2015; Wenk et al., 2019), that have been recurring in gradient matching, are avoided.

Relation to numerical integration The difference between probabilistic numerics and numerical integration for computing the likelihood comes down to the Dirac approximation $\widehat{\delta}_N$. As can be seen in (16), numerical integration does so by simply approximating the locations of the Dirac. On the other hand, probabilistic numerics approximates the Dirac with a distribution of non-zero width, often Gaussian in practice. This has a smoothing effect on the likelihood and parallells can be drawn with the smoothing method in non-convex optimisation (Mobahi & Ma, 2012). But the present method is not equivalent. For example, the smoothing is not with respect to the variable of interest θ, but rather with respect to the function φ_θ.

Previous probabilistic numerics approaches The probabilistic numerics approach to approximate the marginal likelihood has been explored to some extent by Kersting et al. (2020a). However, the present approach confers certain advantages over the former, the most notable being that the Gauss–Markov representation of the probabilistic solvers ensures all computations cost at most $\mathrm{O}(N)$.

A probabilistic numerics approach has also been developed for estimating time varying parameters in the context of latent force modelling (Schmidt et al., 2021). However, for the constant parameter problem, using linearised models can cause divergence in certain situations (Ljung, 1979).

An alternative to the inference-based methods hitherto discussed is to model the error by stochastic perturbation of numerical integrators (Chkrebtii et al., 2016; Conrad et al., 2017; Matsuda & Miyatake, 2021; Teymur et al., 2018).

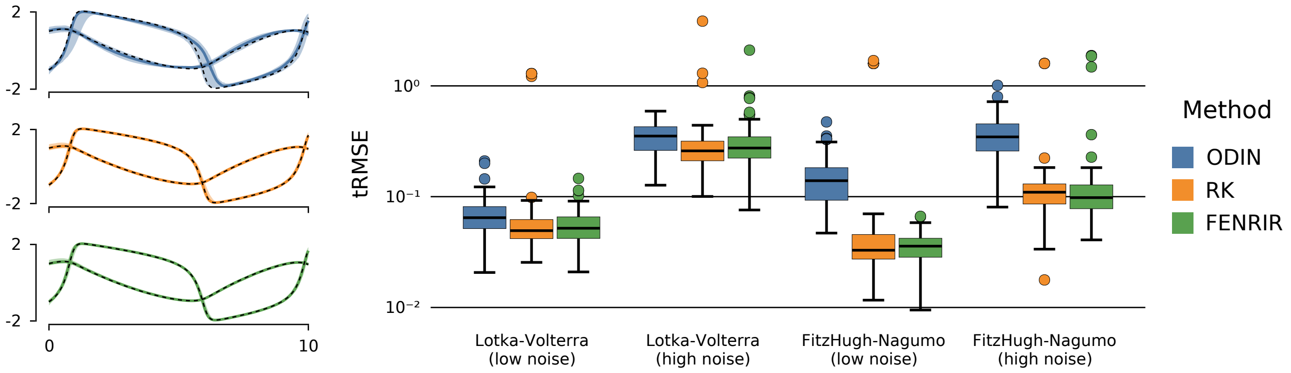


Figure 2. **Benchmarking estimation accuracy.** *Left:* Trajectory summaries of 100 experiments, obtained by numerically integrating the inferred parameters of the FitzHugh–Nagumo system from noisy observations with high noise. The solid lines show the median trajectory, the shaded areas visualize the 10% and 90% quantiles, and the black dashed line shows the ground truth. *Right:* Trajectory RMSEs (tRMSEs) on four benchmarks problems. Fenrir demonstrates performance that is competitive to ODIN and RK.

5. Experimental Results

This section investigates the utility and performance of Fenrir in a range of numerical experiments. It is structured as follows. Section 5.1 evaluates Fenrir on two standard benchmark problems. Section 5.2 demonstrates the utility of the proposed marginal likelihood for model selection. Section 5.3 considers systems with only partially observable states and shows that Fenrir, unlike most gradient matching methods, is still applicable. Finally, Section 5.4 investigates highly oscillatory systems which present a particular challenge for numerical integration-based methods.

Implementation The implementation of the probabilistic numerical IVP solvers follows a number of practices for numerically stable implementation established by Krämer & Hennig (2020). All experiments are implemented in the Julia programming language (Bezanson et al., 2017). Runge–Kutta reference solutions are computed with DifferentialEquations.jl (Rackauckas & Nie, 2017), and numerical optimizers are provided by Optim.jl (Mogensen & Riseth, 2018). All experiments run on a single, consumer-level CPU. Code is publicly available on GitHub.[2]

5.1. Parameter Inference from Fully Observed States

This experiment evaluates Fenrir on two benchmark problems that have been extensively studied in the both the gradient matching and the numerical integration literature (Calderhead et al., 2009; Wenk et al., 2020), namely the Lotka–Volterra predator-prey model and the FitzHugh–Nagumo neuronal model. Detailed system descriptions, along with the ground-truth parameters, initial values, and the chosen observation noise levels, are provided in Appendix C.2. We perform 100 experiments for each experimental setup, in which noisy observations are drawn from the numerically computed, true system trajectories. The inference task then consists in estimating initial values and parameters from noisy state observations. The quality of the resulting parameter estimates is evaluated using the trajectory RMSE (tRMSE) metric as defined in Definition C.1.

We compare Fenrir to the probabilistic gradient matching method ODIN (Wenk et al., 2020) and to a non-linear least squares regression using a Runge–Kutta solver, referred to as RK (Bard, 1974). ODIN results are computed using the code published by Wenk et al. (2020); RK is described in more detail in Appendix C.1. All methods optimise their respective objectives with the L-BFGS algorithm (Nocedal & Wright, 2006). More details are provided in Appendix C.2.

Results of the experiment are shown in Figure 2. In the median, Fenrir performs on par with ODIN and RK on Lotka–Volterra, but both RK and Fenrir outperform ODIN on FitzHugh–Nagumo and achieve more accurate state estimates as well as lower trajectory RMSEs. Both RK and Fenrir suffer from outliers, but this issue appears to be less severe for Fenrir; see also Figure 9 in Appendix C.2.

5.2. Model Selection

For a given set of noisy observations, the true parametric form of the underlying system is often not known exactly. Instead, a set of plausible models has to be evaluated against the observed data in order to find the most fitting candidate. It has been previously shown that probabilistic gradient matching can be used for model selection, by comparing estimated noise parameters which are supposed to account for model mismatch (Wenk et al., 2020). However, as Fenrir operates on a physics-informed probability model, model selection can be accomplished by statistically rigorous methods such as likelihood ratio testing (King, 1998).

[2] https://github.com/nathanaelbosch/fenrir-experiments

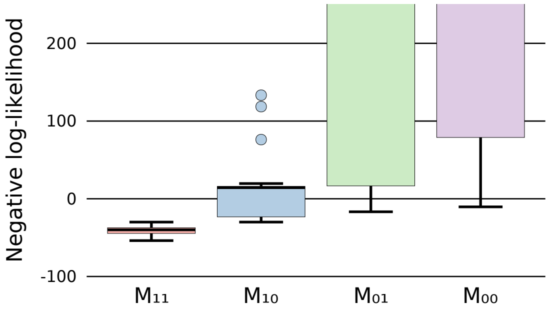


Figure 3. **Model selection results.** Fenrir correctly attributes the lowest negative log-likelihood (i.e. the highest probability) to the true M_{11} model. The figure is restricted to y-values up to 250 to show a clearer visualization, since the results vary largely in scale.

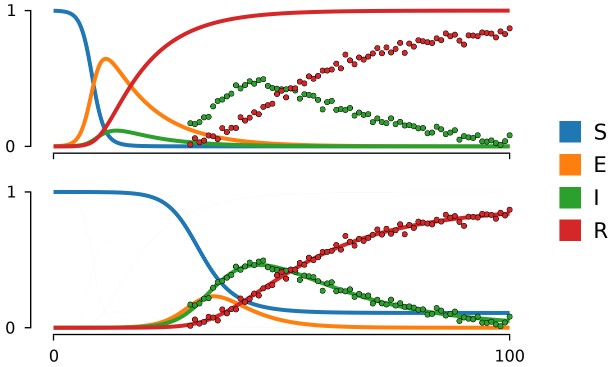


Figure 4. **Parameter inference in a SEIR model.** *Top:* Trajectory resulting from the initial, randomly chosen parameters and initial values. *Bottom:* Fenrir posterior after parameter optimization.

The experiment follows the setup proposed by Wenk et al. (2020). We consider the Lotka–Volterra system as ground truth from which we numerically simulate experimental data, and generate a set of four candidate models by combining the true ODEs with two additional, incorrect equations – all equations and parameters are provided in Appendix C.3. We obtain four models, $\{M_{11}, M_{10}, M_{01}, M_{00}\}$, where M_{11} corresponds to the true Lotka–Volterra dynamics, M_{10} and M_{01} contain one correct and one wrong equation, and M_{00} contains only incorrect equations. Thus, to succeed in this experiment, Fenrir should identify the correct model M_{11}.

We perform 100 individual model selection experiments to evaluate Fenrir's robustness regarding the observation noise. The resulting marginal likelihoods are shown in Figure 3. We observe that Fenrir consistently attributes the lowest negative log-likelihood to the correct model M_{11}, and is thus able to accurately identify the true model.

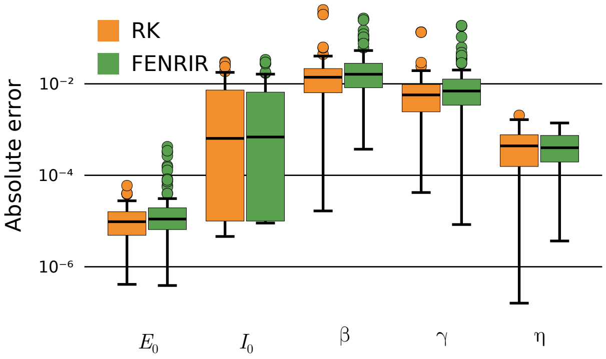


Figure 5. **Absolute parameter errors in the SEIR experiment.** Fenrir performs on par with the non-probabilistic Runge–Kutta baseline (RK) and is able to infer accurate parameter estimates from only partial observations of the SEIR system.

5.3. Partially Observed System States

Here, we evaluate Fenrir on an epidemeological model in which the system state can only be partially observed. We consider a compartmental SEIR model that describes the fractions of a population that are susceptible (S), exposed (E), infected (I; i.e. diagnosed with a positive test), and recovered (R) over time (Hethcote, 2000). Such compartmental models are commonly used to model the development of infectious diseases, and variants of the SEIR model have been used to explain COVID-19 outbreaks (Menda et al., 2021). The definition of the dynamics, ground-truth initial values, and parameters are provided in Appendix C.4.

At each point in time, only the infected and recovered population can be (approximately) observed, but the exposed and susceptible population is unknown. Since Fenrir's "dynamics-first" approach only requires the observation to be linearly dependent on the system states (see Equations (2) and (15c)), no particular adjustments are needed for this experiment. Similarly, the Runge–Kutta-based approach considered in Section 5.1 is also applicable and will be used for comparison. However, most gradient matching methods require all dimensions of the system states to be measurable in order to construct an interpolant, and are therefore not applicable to problems with partial observability.

Figure 4 visualizes an individual experiment: The initial values, parameters, and true system trajectories have to be estimated from noisy case counts of the infected and recovered population, which are furthermore given only from day 30 onwards. The results of 100 experiments are shown in Figure 5. Fenrir is able to consistently infer accurate parameter and trajectory estimates from noisy, partial observations of the dynamical system.

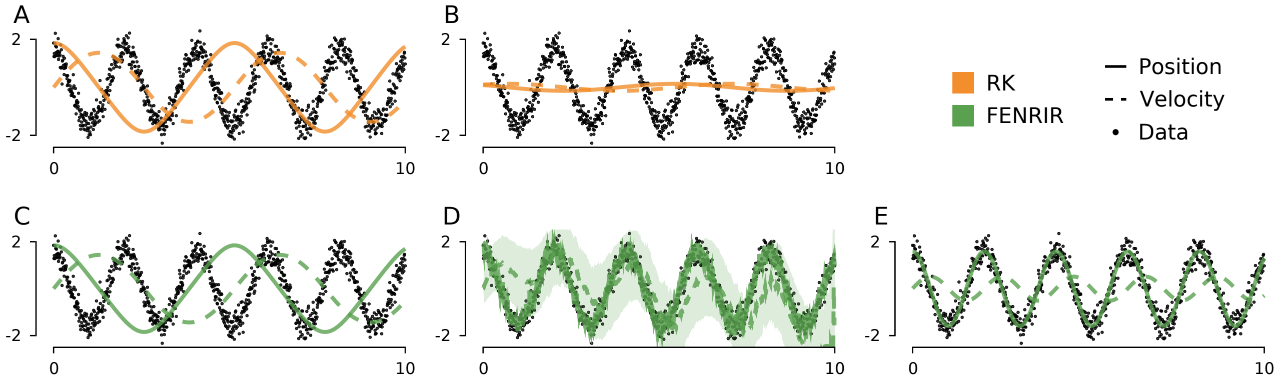


Figure 6. **Parameter inference in oscillatory systems.** Both RK and Fenrir start with an initial guess $L_0 = 5.0$ for the pendulum length parameter [A,C]. After optimization, the Runge–Kutta least-squares method RK ends up in a local minimum and is not able to recover the true parameter $L^* = 1.0$ [B]. On the other hand, Fenrir first increases its diffusion hyperparameter to interpolate the data [D] (c.f. Figure 7 below), and is then able to accurately recover the system parameter via optimization and provides accurate trajectory estimates [E].

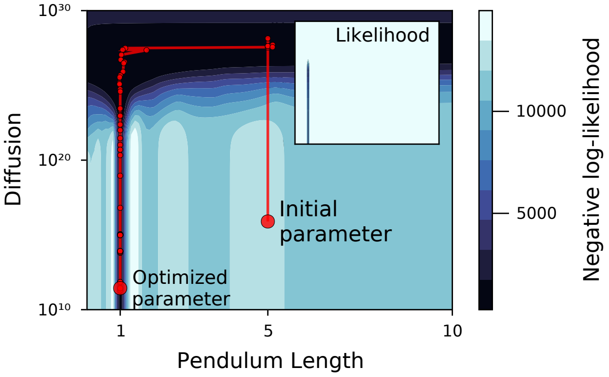


Figure 7. **Negative log-likelihood and optimization trajectory.** By first increasing its diffusion parameter, Fenrir is able to recover the true pendulum length parameter $L = 1$ by minimising the negative log-likelihood using L-BFGS. The likelihood (i.e. the negative exponential of the main plot) is shown in the inset figure.

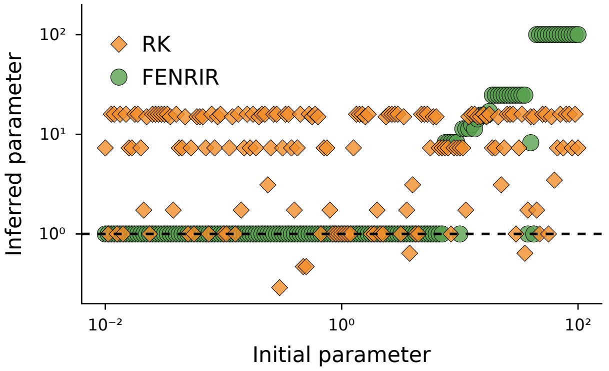


Figure 8. **Inferred parameters for various starting values.** Both RK and Fenrir are evaluated on a wide range of initial parameter estimates, from which they attempt to recover the true parameter $L = 1$ (dashed line) by optimization via L-BFGS. RK is often unable to approximate the true parameter, whereas Fenrir accurately recovers the true parameter for a wide range of starting points.

5.4. Dynamical Systems with Fast Oscillations

Finally, we evaluate Fenrir on a partially observable pendulum system that exhibits fast oscillations. Problems of this form are known to be challenging for simulation-based methods such as the previously considered Runge–Kutta least-squares approach which, with poor initialization, often fail to capture the high frequencies (Benson, 1979). While gradient-matching methods are expected to be more robust to such problems, they require fully observable states and are therefore not applicable in the present setting. Thus, we investigate Fenrir's capabilities of performing trajectory, parameter, and initial value inference under these challenges.

Figure 6 visualizes the problem setup and a single experiment; a detailed description of the dynamics and the chosen hyperparameters is provided in Appendix C.5. In the shown example, the non-linear least squares regression converges towards the constant zero function and is unable to capture the high frequencies of the data. On the other hand, by first optimizing the diffusion and observation noise parameters separately, Fenrir interpolates the experimental data and is then able to accurately approximate the true system parameters. The chosen optimization trajectory is visualised with the corresponding loss landscape in Figure 7. Figure 8 shows inferred parameters for a wider range of starting values; for simplicity, the initial value y_0 is assumed to be known here. RK often fails to converge towards the ground-truth, whereas Fenrir is able to recover the true parameter for a wide range of starting values.

6. Conclusion

It has been demonstrated that the solution of an initial value problem can be approximated by a Gauss–Markov process, reducing the inference problem to Gauss–Markov regression. The method offers advantages such as $\mathrm{O}(N)$ cost for inference, operability in the face of partial observations, regularised likelihoods, and moderate improvements in terms of estimation accuracy. But, perhaps more importantly, it has been shown that probabilistic numerics is a promising method for rigorously incorporating physics in Gaussian process regression.

Acknowledgements

The authors gratefully acknowledge financial support by the German Federal Ministry of Education and Research (BMBF) through Project ADIMEM (FKZ 01IS18052B), and financial support by the European Research Council through ERC StG Action 757275 / PANAMA; the DFG Cluster of Excellence "Machine Learning - New Perspectives for Science", EXC 2064/1, project number 390727645; the German Federal Ministry of Education and Research (BMBF) through the T¨ubingen AI Center (FKZ: 01IS18039A); and funds from the Ministry of Science, Research and Arts of the State of Baden-Württemberg. The authors also thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting N. Bosch.

References

Akaike, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

Alahmadi, A. A., Flegg, J. A., Cochrane, D. G., Drovandi, C. C., and Keith, J. M. A comparison of approximate versus exact techniques for Bayesian parameter inference in nonlinear ordinary differential equation models. *Royal Society open science*, 7(3), 2020.

Arnol'd, V. I. *Ordinary Differential Equations*. Springer-Verlag Berlin Heidelberg, 1992.

Åström, K. J. and Eykhoff, P. System identification – a survey. *Automatica*, 7(2):123–162, 1971.

Barber, D. and Wang, Y. Gaussian processes for Bayesian estimation in ordinary differential equations. In *International Conference on Machine Learning*, pp. 1485–1493. PMLR, 2014.

Bard, Y. *Nonlinear parameter estimation*. Academic Press, 1974.

Bell, B. M. The iterated Kalman smoother as a Gauss–Newton method. *SIAM Journal on Optimization*, 4(3): 626–636, 1994.

Benson, M. Parameter fitting in dynamic models. *Ecological Modelling*, 6(2):97–115, 1979.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 9 2017. doi: 10.1137/141000671.

Bi, Q., Wu, Y., Mei, S., Ye, C., Zou, X., Zhang, Z., Liu, X., Wei, L., Truelove, S. A., Zhang, T., Gao, W., Cheng, C., Tang, X., Wu, X., Wu, Y., Sun, B., Huang, S., Sun, Y., Zhang, J., Ma, T., Lessler, J., and Feng, T. Epidemiology and transmission of COVID-19 in 391 cases and 1286 of their close contacts in Shenzhen, China: a retrospective cohort study. *The Lancet. Infectious diseases*, 20(8):911–919, Aug 2020.

Biegler, L. T., Damiano, J. J., and Blau, G. E. Nonlinear parameter estimation: a case study comparison. *AIChE Journal*, 32(1):29–45, 1986.

Bosch, N., Hennig, P., and Tronarp, F. Calibrated adaptive probabilistic ODE solvers. In *International Conference on Artificial Intelligence and Statistics*, pp. 3466–3474. PMLR, 2021.

Bosch, N., Tronarp, F., and Hennig, P. Pick-and-mix information operators for probabilistic ODE solvers. In *International Conference on Artificial Intelligence and Statistics*, pp. 10015–10027. PMLR, 2022.

Calderhead, B., Girolami, M., and Lawrence, N. D. Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 217–224, 2009.

Cao, J., Wang, L., and Xu, J. Robust estimation for ordinary differential equation models. *Biometrics*, 67(4):1305–1313, 2011.

Chkrebtii, O. A., Campbell, D. A., Calderhead, B., and Girolami, M. A. Bayesian solution uncertainty quantification for differential equations. *Bayesian Analysis*, 11 (4):1239–1267, 12 2016.

Cockayne, J., Oates, C. J., Sullivan, T. J., and Girolami, M. Bayesian probabilistic numerical methods. *SIAM Review*, 61(4):756–789, 2019.

Conrad, P. R., Girolami, M., Särkkä, S., Stuart, A., and Zygalakis, K. Statistical analysis of differential equations: introducing probability measures on numerical solutions. *Statistics and Computing*, 27(4):1065–1082, Jul 2017.

Dass, S. C., Lee, J., Lee, K., and Park, J. Laplace based approximate posterior inference for differential equation models. *Statistics and Computing*, 27(3):679–698, 2017.

Dondelinger, F., Husmeier, D., Rogers, S., and Filippone, M. ODE parameter inference using adaptive gradient matching with Gaussian processes. In *Artificial intelligence and Statistics*, pp. 216–228, 2013.

FitzHugh, R. Mathematical models of threshold phenomena in the nerve membrane. *The bulletin of mathematical biophysics*, 17(4):257–278, 1955.

Friston, K. J. Bayesian estimation of dynamical systems: an application to fMRI. *NeuroImage*, 16(2):513–530, 2002.

Gelman, A., Bois, F., and Jiang, J. Physiological pharmacokinetic analysis using population modeling and informative prior distributions. *Journal of the American Statistical Association*, 91(436):1400–1412, 1996.

Gorbach, N. S., Bauer, S., and Buhmann, J. M. Scalable variational inference for dynamical systems. In *Advances in Neural Information Processing Systems*, pp. 4806–4815, 2017.

Gugushvili, S. and Klaassen, C. A. J. $\sqrt{n}$-consistent parameter estimation for systems of ordinary differential equations: bypassing numerical integration via smoothing. *Bernoulli*, 18(3):1061–1098, 2012.

Hairer, E. and Wanner, G. Stiff differential equations solved by Radau methods. *Journal of Computational and Applied Mathematics*, 111, 1999.

Hairer, E., Nørsett, S., and Wanner, G. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, 1987.

Hennig, P., Osborne, M. A., and Girolami, M. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150142, 2015.

Hethcote, H. W. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

Kalman, R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1): 35–45, 1960.

Kersting, H., Krämer, N., Schiegg, M., Daniel, C., Tiemann, M., and Hennig, P. Differentiable likelihoods for fast inversion of 'likelihood-free' dynamical systems. In *International Conference on Machine Learning*, pp. 5198–5208. PMLR, 2020a.

Kersting, H., Sullivan, T. J., and Hennig, P. Convergence rates of Gaussian ODE filters. *Statistics and computing*, 30(6):1791–1816, 2020b.

King, G. *Unifying political methodology: The likelihood theory of statistical inference*. University of Michigan Press, 1998.

Krämer, N. and Hennig, P. Stable implementation of probabilistic ODE solvers. *arXiv preprint arXiv:2012.10106*, 2020.

Krämer, N., Bosch, N., Schmidt, J., and Hennig, P. Probabilistic ODE solutions in millions of dimensions. *arXiv preprint arXiv:2110.11812*, 2021.

Lauer, S. A., Grantz, K. H., Bi, Q., Jones, F. K., Zheng, Q., Meredith, H. R., Azman, A. S., Reich, N. G., and Lessler, J. The incubation period of coronavirus disease 2019 (COVID-19) from publicly reported confirmed cases: estimation and application. *Annals of internal medicine*, 172(9):577–582, 2020.

Ljung, L. Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, 24(1):36–50, 1979.

Lotka, A. *Elements of Physical Biology*. Williams & Wilkins, 1925.

Macdonald, B., Higham, C., and Husmeier, D. Controversy in mechanistic modelling with Gaussian processes. *Proceedings of Machine Learning Research*, 37:1539–1547, 2015.

Magnani, E., Kersting, H., Schober, M., and Hennig, P. Bayesian Filtering for ODEs with Bounded Derivatives. *arXiv:1709.08471 [cs.NA]*, September 2017.

Matsuda, T. and Miyatake, Y. Estimation of ordinary differential equation models with discretization error quantification. *SIAM/ASA Journal on Uncertainty Quantification*, 9(1):302–331, 2021.

Menda, K., Laird, L., Kochenderfer, M. J., and Caceres, R. S. Explaining COVID-19 outbreaks with reactive SEIRD models. *Scientific Reports*, 11(1):17905, Sep 2021.

Mobahi, H. and Ma, Y. Gaussian smoothing and asymptotic convexity. *Coordinated Science Laboratory Report no. UILU-ENG-12-2201, DC-254*, 2012.

Mogensen, P. K. and Riseth, A. N. Optim: A mathematical optimization package for Julia. *Journal of Open Source Software*, 3(24):615, 2018. doi: 10.21105/joss.00615.

Nagumo, J., Arimoto, S., and Yoshizawa, S. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, New York, NY, USA, 2e edition, 2006.

Oates, C. J. and Sullivan, T. J. A modern retrospective on probabilistic numerics. *Statistics and Computing*, 29(6): 1335–1351, 2019.

Øksendal, B. *Stochastic Differential Equations - An Introduction with Applications*. Springer, 2003.

Rackauckas, C. and Nie, Q. DifferentialEquations.jl–a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1), 2017.

Ramsay, J. O., Hooker, G., Campbell, D., and Cao, J. Parameter estimation for differential equations: a generalized smoothing approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(5):741–796, 2007.

Rauch, H. E., Tung, F., and Striebel, C. T. Maximum likelihood estimates of linear dynamic system. *AIAA Journal*, 3(8):1445–1450, Aug 1965.

Särkkä, S. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

Särkkä, S. and Solin, A. *Applied Stochastic Differential Equations*. Cambridge University Press, 2019.

Schmidt, J., Krämer, N., and Hennig, P. A probabilistic state space model for joint inference from differential equations and data. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Schober, M., Särkkä, S., and Hennig, P. A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing*, 29(1):99–122, January 2019.

Schweppe, F. Evaluation of likelihood functions for Gaussian signals. *IEEE Transactions on Information Theory*, 11(1):61–70, 1965.

Stoica, P. and Selen, Y. Model-order selection: a review of information criterion rules. *IEEE Signal Processing Magazine*, 21(4):36–47, 2004.

Teymur, O., Lie, H. C., Sullivan, T., and Calderhead, B. Implicit probabilistic integrators for ODEs. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. H. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31): 187–202, 2009.

Tronarp, F., Karvonen, T., and Särkkä, S. Student's t-filters for noise scale estimation. *IEEE Signal Processing Letters*, 26(2):352–356, 2019a.

Tronarp, F., Kersting, H., Särkkä, S., and Hennig, P. Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: a new perspective. *Statistics and Computing*, 29(6):1297–1315, 2019b.

Tronarp, F., Särkkä, S., and Hennig, P. Bayesian ODE solvers: The maximum a posteriori estimate. *Statistics and Computing*, 31(3):1–18, 2021.

Tsitouras, C. Runge–Kutta pairs of order 5 (4) satisfying only the first column simplifying assumption. *Computers & Mathematics with Applications*, 62, 2011.

Varah, J. M. A spline least squares method for numerical parameter estimation in differential equations. *SIAM Journal on Scientific and Statistical Computing*, 3(1):28–46, 1982.

Voit, E. O. *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. Cambridge University Press, 2000.

Volterra, V. Variations and Fluctuations of the Number of Individuals in Animal Species living together. *ICES Journal of Marine Science*, 3(1):3–51, 1928.

Wenk, P., Gotovos, A., Bauer, S., Gorbach, N. S., Krause, A., and Buhmann, J. M. Fast Gaussian process based gradient matching for parameter identification in systems of nonlinear ODEs. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1351–1360. PMLR, 2019.

Wenk, P., Abbati, G., Osborne, M. A., Schölkopf, B., Krause, A., and Bauer, S. ODIN: ODE-informed regression for parameter and state inference in time-continuous dynamical systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 6364–6371, 2020.

A. Additional Details on Probabilistic Numerics

In this appendix, the probabilistic solver is described in detail. Further details on the prior are given in Appendix A.1. In Appendix A.2, it is explained how to compute the marginal moments and the parameters of the backward Markov representation of the posterior when the vector field is linear (affine). In Appendix A.3 some linearisation methods for approximate inference when the vector field is non-linear are reviewed.

A.1. More details on priors

Recall that the prior in state-space form is given by

$$\mathrm{d}x(t) = Ax(t)\,\mathrm{d}t + \sqrt{\kappa}B\,\mathrm{d}w(t), \quad x(0) = x_\theta^\dagger, \tag{17a}$$

$$y^{(m)}(t) = \mathrm{E}_m^\mathsf{T} x(t), \quad m = 0, 1, \dots, \nu, \tag{17b}$$

where $y^{(m)}$ models the mth derivative of the solution. By Itô's formula this implies that

$$\mathrm{d}\mathrm{E}_m^\mathsf{T} x(t) = \mathrm{E}_m^\mathsf{T} Ax(t)\,\mathrm{d}t + \sqrt{\kappa}\mathrm{E}_m^\mathsf{T} B\,\mathrm{d}w(t), \tag{18}$$

and for this to be consistent with the asserted derivative relations it must hold that

$$\mathrm{E}_m^\mathsf{T} Ax(t)\,\mathrm{d}t + \sqrt{\kappa}\mathrm{E}_m^\mathsf{T} B\,\mathrm{d}w(t) = \mathrm{E}_{m+1}^\mathsf{T} x(t)\,\mathrm{d}t, \quad m = 0, 1, \dots, \nu - 1. \tag{19}$$

This in turn implies that it must hold that

$$\mathrm{E}_m^\mathsf{T} A = \mathrm{E}_{m+1}^\mathsf{T}, \quad m = 0, 1, \dots, \nu - 1, \tag{20a}$$

$$\mathrm{E}_m^\mathsf{T} B = 0, \quad m = 0, 1, \dots, \nu - 1, \tag{20b}$$

while $\mathrm{E}_\nu^\mathsf{T} A$ and $\mathrm{E}_\nu^\mathsf{T} B$ are free parameters. Letting e_m be the mth canonical basis vector in $\mathbb{R}^{\nu+1}$, I_d be the d by d identity matrix, and fixing $\mathrm{E}_m = \mathrm{e}_m \otimes \mathrm{I}_d$ then gives the model

$$\mathrm{d}y^{(\nu)}(t) = \sum_{m=0}^{\nu} A_{\nu,m} y^{(m)}\,\mathrm{d}t + \sqrt{\kappa}B_\nu\,\mathrm{d}w(t), \tag{21}$$

where $A_{\nu,m} = \mathrm{E}_\nu^\mathsf{T} A \mathrm{E}_m$ and $B_\nu = \mathrm{E}_\nu^\mathsf{T} B$. Any other state-space model of dimension $d(\nu + 1)$ modelling a vector valued function of dimension d and its ν first derivatives must be related to this via similarity transform. The canonical model in probabilistic numerics is the ν-times integrated Wiener process (Schober et al., 2019; Tronarp et al., 2019b; Krämer & Hennig, 2020; Bosch et al., 2021a; Kersting et al., 2020b), where the parameters are given by

$$A_{\nu,m} = 0, \quad m = 0, 1, \dots, \nu - 1. \tag{22}$$

Though other priors are of course possible (Magnani et al., 2017; Tronarp et al., 2021; Kersting et al., 2020b). Usually, the diffusion matrix B_ν is set to identity as well, yielding the following prior

$$\mathrm{d}y^{(\nu)}(t) = \sqrt{\kappa}\,\mathrm{d}w(t), \tag{23}$$

which is used throughout the article.

A.2. Posterior for linear vector fields

Suppose the vector field is linear:

$$f_\theta(t, y) = L_\theta(t)y + b_\theta(t), \tag{24}$$

then the probabilistic IVP solver reduces to inference in the following model:

$$\mathrm{d}x(t) = Ax(t)\,\mathrm{d}t + \sqrt{\kappa}B\,\mathrm{d}w(t), \quad x(0) = x_\theta^\dagger, \tag{25}$$

subject to the data

$$C_\theta(t) = \mathrm{E}_1 - \mathrm{E}_0 L_\theta^\mathsf{T}(t), \tag{26a}$$

$$z(t) = 0 = \mathrm{E}_1^\mathsf{T} x(t) - L_\theta(t)\mathrm{E}_0^\mathsf{T} x(t) - b_\theta(t), \quad t \in \mathbb{T}_{\mathsf{PN}}. \tag{26b}$$

The posterior is Gaussian because the prior is Gaussian and the measurement functionals are linear, and it can be computed with the well-known forward / backward recursions (Kalman, 1960; Rauch et al., 1965).

More specifically, denote the numerics data up to time t by

$$\mathscr{Z}_{[0,t]} = \{z(t) = 0 \colon t \in \mathbb{T}_{\mathsf{PN}} \cap [0,t]\} \tag{27}$$

and up to *just before* time t by

$$\mathscr{Z}_{[0,t)} = \{z(t) = 0 \colon t \in \mathbb{T}_{\mathsf{PN}} \cap [0,t)\}. \tag{28}$$

The forward recursion then computes the filtering densities

$$p(t, x \mid \mathscr{Z}_{[0,t]}) = \mathcal{N}(x; \mu_\theta(t), \kappa\Sigma_\theta(t)), \tag{29}$$

which agree with the prediction densities

$$p(t, x \mid \mathscr{Z}_{[0,t)}) = \mathcal{N}(x; \mu_\theta(t^-), \kappa\Sigma_\theta(t^-)), \tag{30}$$

unless $t \in \mathbb{T}_{\mathsf{PN}}$. The filtering moments are then post-processed in the backwards recursion to produce the smoothing densities (time marginals of the posterior)

$$p(t, x \mid \mathscr{Z}(t_N)) = \mathcal{N}(x; \xi_\theta(t), \kappa\Lambda_\theta(t)). \tag{31}$$

For a more thorough exposition on filtering and smoothing refer to Särkkä (2013); Särkkä & Solin (2019). Furthermore, the fact that the scaling κ is retained throughout the recursion follows from the fact that the initial covariance and all transition covariances are scaled by κ (Tronarp et al., 2019a).

Forward recursion The forward recursion starts by initialising the filter mean and covariance according to

$$\mu_\theta(t_0) = x_\theta^\dagger, \tag{32a}$$
$$\Sigma_\theta(t_0) = 0, \tag{32b}$$

whereafter the algorithm alternates between prediction and update. The prediction equations are given by

$$\mu_\theta(t_n^-) = \Phi(\Delta_n)\mu_\theta(t_{n-1}), \tag{33a}$$
$$\Sigma_\theta(t_n^-) = \Phi(\Delta_n)\Sigma_\theta(t_{n-1})\Phi^\mathsf{T}(\Delta_n) + Q(\Delta_n), \tag{33b}$$
$$G_\theta(t_{n-1}) = \Sigma_\theta(t_{n-1})\Phi^\mathsf{T}(\Delta_n)\Sigma_\theta^{-1}(t_n^-), \tag{33c}$$
$$P_\theta(t_{n-1}) = \Sigma_\theta(t_{n-1}) - G_\theta(t_{n-1})\Sigma_\theta(t_n^-)G_\theta^\mathsf{T}(t_{n-1}), \tag{33d}$$

where G_θ and P_θ are parameters associated with the subsequent backward recursion. The update relations are given by

$$C_\theta(t_n) = \mathrm{E}_1 - \mathrm{E}_0 L_\theta^\mathsf{T}(t_n), \tag{34a}$$
$$S_\theta(t_n) = C_\theta^\mathsf{T}(t_n)\Sigma_\theta(t_n^-)C_\theta(t_n), \tag{34b}$$
$$K_\theta(t_n) = \Sigma_\theta(t_n^-)C_\theta^\mathsf{T}(t_n)S_\theta^{-1}(t_n), \tag{34c}$$
$$\mu_\theta(t_n) = \mu_\theta(t_n^-) + K_\theta(t_n)\big(b_\theta(t_n) - C_\theta^\mathsf{T}(t_n)\mu_\theta(t_n^-)\big), \tag{34d}$$
$$\Sigma_\theta(t_n) = \Sigma_\theta(t_n^-) - K_\theta(t_n)S_\theta(t_n)K_\theta^\mathsf{T}(t_n). \tag{34e}$$

Backward recursion The backwards recursion starts by setting the smoother mean and covariance to the filter mean and covariance at the terminal point according to

$$\xi_\theta(t_N) = \mu_\theta(t_N), \tag{35a}$$
$$\Lambda_\theta(t_N) = \Sigma_\theta(t_N). \tag{35b}$$

The backwards recursion is then given by

$$\xi_\theta(t_n) = \mu_\theta(t_n) + G_\theta(t_n)\big(\xi_\theta(t_{n+1}) - \Phi(\Delta_{n+1})\mu_\theta(t_n)\big), \tag{36a}$$
$$\Lambda_\theta(t_n) = G_\theta(t_n)\Lambda_\theta(t_{n+1})G_\theta^\mathsf{T}(t_n) + P_\theta(t_n). \tag{36b}$$

Backward Markov process representation Lastly, the posterior may be represented, on the grid, by the following backwards Markov process

$$\begin{aligned}\gamma_N(x(t_{1:N}) \mid \theta, \kappa) &= \mathcal{N}\big(x(t_N); \xi_\theta(t_N), \kappa \Lambda_\theta(t_N)\big) \\ &\prod_{n=N-1}^{1} \mathcal{N}\big(x(t_n); \mu_\theta(t_n) + G_\theta(t_n)\big(x(t_{n+1}) - \mu_\theta(t_{n+1}^-)\big), \kappa P_\theta(t_n)\big).\end{aligned} \tag{37}$$

This follows from the fact that

$$p(t, x \mid s, x', \mathscr{Z}_{[0,T]}) = p(t, x \mid s, x', \mathscr{Z}_{[0,t]}), \quad t_{n+1} \geq s > t \geq t_n,\ n = 1, \dots, N. \tag{38}$$

That is, by total probability

$$\begin{aligned} p(t_n, x_n \mid \mathscr{Z}_{[0,T]}) &= \int p(t_n, x_n \mid t_{n+1}, x_{n+1}, \mathscr{Z}_{[0,T]}) p(t_{n+1}, x_{n+1} \mid \mathscr{Z}_{[0,T]}) \, \mathrm{d}x_{n+1} \\ &= \int p(t_n, x_n \mid t_{n+1}, x_{n+1}, \mathscr{Z}_{[0,t_n]}) p(t_{n+1}, x_{n+1} \mid \mathscr{Z}_{[0,T]}) \, \mathrm{d}x_{n+1}, \end{aligned} \tag{39}$$

and by Bayes' rule

$$\begin{aligned} p(t_n, x_n \mid t_{n+1}, x_{n+1}, \mathscr{Z}_{[0,t_n]}) &\propto p(t_n, x_n \mid \mathscr{Z}_{0,t_n}) p(t_{n+1}, x_{n+1} \mid t_n, x_n, \mathscr{Z}_{[0,t_n]}) \\ &= \mathcal{N}(x_n; \mu_\theta(t_n), \kappa \Sigma_\theta(t_n)) \mathcal{N}(x_{n+1}; \Phi(\Delta_{n+1}) x_n, \kappa Q(\Delta_{n+1})) \\ &= \mathcal{N}(x_{n+1}; \mu_\theta(t_{n+1}^-), \Sigma_\theta(t_{n+1}^-)) \mathcal{N}(x_n; \mu_\theta(t_n) + G_\theta(t_n)(x_{n+1} - \mu_\theta(t_{n+1}^-)), \kappa P_\theta(t_n)) \\ &\propto \mathcal{N}(x_n; \mu_\theta(t_n) + G_\theta(t_n)(x_{n+1} - \mu_\theta(t_{n+1}^-)), \kappa P_\theta(t_n)), \end{aligned} \tag{40}$$

where the last equality follows from ordinary Gaussian conditioning and the proportionality signs are with respect to x_n. This proves the recursive structure of the posterior as asserted by proposition 3.1, and the complete result follows from the fact that the marginal filtering and smoothing densities coincide at the terminal point. That is,

$$p(t_N, x_N, \mid \mathscr{Z}_{[0,T]}) = \mathcal{N}(x_N; \mu_\theta(t_N), \kappa \Sigma_\theta(t_N)) = \mathcal{N}(x_N; \xi_\theta(t_N), \kappa \Lambda_\theta(t_N)). \tag{41}$$

A.3. Approximate posteriors via linearisation

When the vector field is non-linear, the posterior is in most cases intractable. However, approximate posteriors may be obtained by linearising the data relation in (7). Due to the structure of the information operator, there are multiple choices for doing this, namely

1. Zeroth order linearisation (Schober et al., 2019):

$$\hat{L}_\theta(t) = 0, \tag{42a}$$

$$\hat{b}_\theta(t) = f_\theta(t, \tilde{y}(t)) \tag{42b}$$

2. First order linearisation (Tronarp et al., 2019b):

$$\hat{L}_\theta(t) = J_{f_\theta}(t, \tilde{y}(t)), \tag{43a}$$

$$\hat{b}_\theta(t) = f_\theta(t, \tilde{y}(t)) - J_{f_\theta}(t, \tilde{y}(t)) \tilde{y}(t) \tag{43b}$$

The linearisation point is typicaly chosen as the predictive mean:

$$\tilde{y}(t) = \mathrm{E}_0^\mathsf{T} \mu_\theta(t^-). \tag{44}$$

However, other choices are possible as well, such as the smoothing mean (Tronarp et al., 2021)

$$\tilde{y}(t) = \mathrm{E}_0^\mathsf{T} \xi_\theta(t), \tag{45}$$

which leads to the fixed-point equations for the Gauss–Newton algorithm (Bell, 1994).

B. Inference in IVPs as Gauss–Markov regression

Using the probabilistic numerics posterior as a surrogate for the solution of the initial value problem leads to the following inference problem

$$x(t_N) \sim \mathcal{N}(\xi_\theta(t_N), \kappa\Lambda_\theta(t_N)), \tag{46a}$$

$$x(t_n) \mid x(t_{n+1}) \sim \widehat{\gamma}_N(x(t_n) \mid x(t_{n+1}) \mid \theta, \kappa), \tag{46b}$$

$$u(t) \mid x(t) \sim \mathcal{N}(H^\mathsf{T}\mathrm{E}_0^\mathsf{T} x(t_n), R_\theta), \quad t \in \mathbb{T}_\mathsf{D}. \tag{46c}$$

This is again, a problem of Gauss–Markov regresion and can be solved by the usual forward / backward recursions. What is unusual is that the latent process is specified in terms of a terminal distribution and backward transition densities. Therefore, the equations required for implementation are given in detail.

B.1. The forward (but backward in time) recursion and the marginal likelihood

The backward recursion is implemented by a forward recursion with flipped time axis. That is, start by initialising the filter moments:

$$\breve{\mu}_\theta(t_N^+) = \xi_\theta(t_N), \tag{47a}$$

$$\breve{\Sigma}_\theta(t_N^+) = \kappa\Lambda_\theta(t_N), \tag{47b}$$

whereafter the algorithm alternates between a backward prediction and update. If $t_n \in \mathbb{T}_\mathsf{D}$, then an update is performed according to

$$\breve{H} = \mathrm{E}_0 H, \tag{48a}$$

$$\breve{S}(t_n) = \breve{H}^\mathsf{T}\breve{\Sigma}_\theta(t_n)\breve{H} + R_\theta, \tag{48b}$$

$$\breve{K}_\theta(t_n) = \breve{\Sigma}_\theta(t_n)\breve{H}\breve{S}_\theta^{-1}(t_n), \tag{48c}$$

$$\breve{\mu}_\theta(t_n) = \breve{\mu}_\theta(t_n^+) + \breve{K}_\theta(t_n)\big(u(t_n) - \breve{H}^\mathsf{T}\breve{\mu}_\theta(t_n^+)\big), \tag{48d}$$

$$\breve{\Sigma}_\theta(t_n) = \breve{\Sigma}_\theta(t_n^+) - \breve{K}_\theta(t_n)\breve{S}_\theta(t_n)\breve{K}_\theta^\mathsf{T}(t_n). \tag{48e}$$

The prediction step is given by

$$\breve{\mu}_\theta(t_{n-1}^+) = \mu_\theta(t_{n-1}) + G_\theta(t_{n-1})\big(\breve{\mu}_\theta(t_{n+1}) - \mu_\theta(t_{n+1}^-)\big), \tag{49a}$$

$$\breve{\Sigma}_\theta(t_{n-1}^+) = G_\theta(t_{n-1})\breve{\Sigma}_\theta(t_n)G_\theta^\mathsf{T}(t_{n-1}) + \kappa P_\theta(t_{n-1}). \tag{49b}$$

Finally, the marginal likelihood approximation is given by the prediction error decomposition (Schweppe, 1965)

$$\widehat{\mathcal{M}}_N(\theta, \kappa) = \prod_{t \in \mathbb{T}_\mathsf{D}} \mathcal{N}\big(u(t); \breve{H}^\mathsf{T}\breve{\mu}_\theta(t^+), \breve{S}_\theta(t)\big). \tag{50}$$

B.2. The backward (but forward in time) recursion and trajectory estimates

The smoothing parameters for the forward recursion are given by

$$\breve{G}_\theta(t_n) = \breve{\Sigma}_\theta(t_n)G_\theta^\mathsf{T}(t_{n-1})\breve{\Sigma}_\theta^{-1}(t_{n-1}^+), \tag{51a}$$

$$\breve{P}_\theta(t_n) = \breve{\Sigma}_\theta(t_n) - \breve{G}_\theta(t_n)\breve{\Sigma}_\theta(t_{n-1}^+)\breve{G}_\theta^\mathsf{T}(t_n), \tag{51b}$$

and the forward smoothing recursion is given by

$$\breve{\xi}_\theta(t_n) = \breve{\mu}_\theta(t_n) + \breve{G}_\theta(t_n)\big(\breve{\xi}_\theta(t_{n-1}) - G_\theta(t_{n-1})\breve{\mu}_\theta(t_n)\big), \tag{52a}$$

$$\breve{\Lambda}_\theta(t_n) = \breve{G}_\theta(t_n)\breve{\Lambda}_\theta(t_{n-1})\breve{G}_\theta^\mathsf{T}(t_n) + \breve{P}_\theta(t_n). \tag{52b}$$

C. Additional Details on the Experimental Evaluation

In all experiments, Fenrir uses a 5-times integrated Wiener process prior and a first-order linearisation of the vector field during the probabilistic numerical ODE solve when computing its physics-enhanced prior.

Optimization Throughout all experiments, the L-BFGS method has been used for optimization with both Fenrir and RK (Nocedal & Wright, 2006); L-BFGS is also the optimizer of choice in the official ODIN code by Wenk et al. (2020). The specific L-BFGS implementation is provided by the Optim.jl software package (Mogensen & Riseth, 2018). In all experiment, the observation noise σ^2 and the diffusion κ are optimised in log-space.

Parameter Initialization As done in ODIN, ODE parameters are initialised with a folded normal distribution, i.e. as the absolute value of a sample from standard normal Gaussian, and initial values are initialised with their noisy observation $u(t_0)$, unless specified otherwise. Observation noise is always initialised as $\sigma^2 = 1$.

C.1. Baseline: Non-linear Least Squares Regression using a Runge–Kutta Solver

Given data $\mathcal{D} = \{u(t)\}$ on the grid $t \in \mathbb{T}_{\text{D}}$, the considered "RK" baseline method minimizes the loss

$$L := \sum_{t \in \mathbb{T}_{\text{D}}} \left\| H \cdot \hat{y}(t) - u(t) \right\|_2^2, \tag{53}$$

where $\hat{y}(t)$ is computed with a classical Runge–Kutta initial value solver and H is the measurement matrix as introduced in Equation (2). In most experiments, the `Tsit5` (Tsitouras, 2011) solver is used, with adaptive step-size selection for absolute and relative tolerances $\tau_{\text{abs}} = 10^{-8}$, $\tau_{\text{rel}} = 10^{-6}$. Only on the FitzHugh-Nagumo system we use the implicit `RadauIIA5` (Hairer & Wanner, 1999) method, since we observed it to be more robust as some parameter settings can lead to stiff dynamics. Both solvers are provided by DifferentialEquations.jl (Rackauckas & Nie, 2017).

C.2. Additional Details on Section 5.1: "Parameter Inference from Fully Observed States"

Definition C.1 (Trajectory RMSE)**.** Let $\hat{\theta}$ be the parameters estimated by an inference algorithm, and let $\mathbb{T}_{\text{D}}$ be the set of measurement nodes. Then, let $\hat{y}(t)$, $t \in \mathbb{T}_{\text{D}}$, be the estimated system trajectory, computed by numerically integrating the ODE with initial values and parameters as given by the estimated $\hat{\theta}$. The trajectory RMSE (tRMSE) is then defined as

$$\text{tRMSE} := \sqrt{\frac{1}{|\mathbb{T}_{\text{D}}|} \sum_{t \in \mathbb{T}_{\text{D}}} \left\| \hat{y}(t) - y(t) \right\|_2^2}. \tag{54}$$

Lotka–Volterra The Lotka–Volterra model describes the dynamics of biological systems in which two species interact, one as a predator and the other as prey (Lotka, 1925; Volterra, 1928). It is described by the ODEs

$$\dot{y_1} = \alpha y_1 - \beta y_1 y_2, \tag{55a}$$

$$\dot{y_2} = -\gamma y_1 + \delta y_1 y_2. \tag{55b}$$

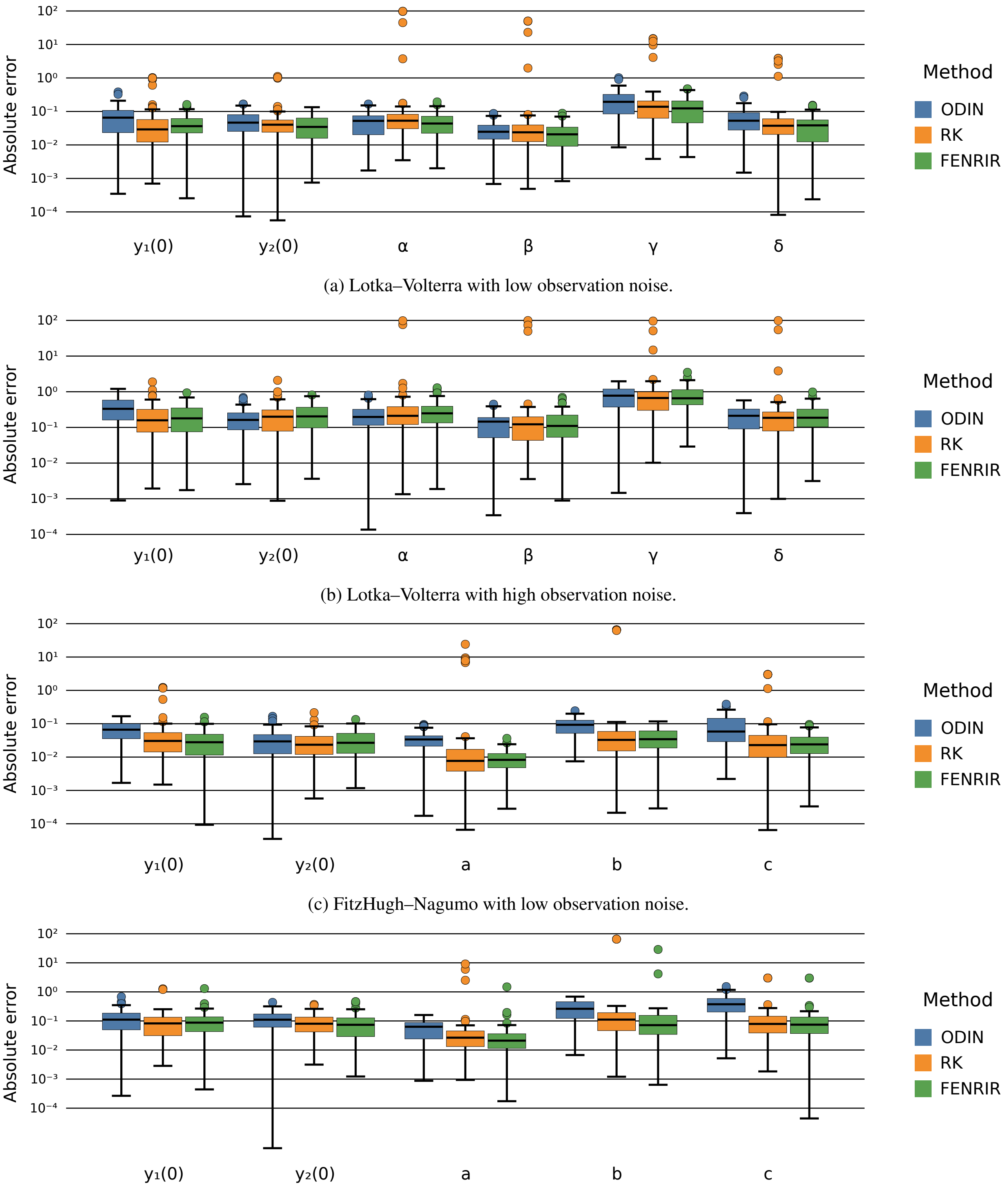
As ground truth, we assume an initial value $y_0 = [5, 3]^\intercal$ and parameters $\alpha = 2$, $\beta = 1$, $\gamma = 4$, $\delta = 1$. The experimental data is generated on the equi-spaced time grid $t_i \in \mathbb{T}_{\text{D}} = \{0.0, 0.1, \dots, 2.0\}$, as $u(t_i) = \hat{y}(t_i) + v(t_i)$, where $\hat{y}(t_i)$ is computed via accurate, numerical simulation, and with noise $v(t) \sim \mathcal{N}(0, \sigma^2 \cdot I)$. We further consider two different noise levels $\sigma^2_{\text{low}} = 0.01$ and $\sigma^2_{\text{high}} = 0.25$. Thus, the full set of parameters to be estimated is $\theta = \{y_0, \alpha, \beta, \gamma, \delta, \sigma\}$, as well as the diffusion κ. In this system, we found it helpful to first optimize the noise and diffusion parameters σ, κ individually until convergence, and only then optimize all parameters jointly; such an approach is also chosen by the gradient matching method ODIN (Wenk et al., 2020). Furthermore, as in the original experimental setup by Wenk et al. (2020), we consider bounds $y_0 \in [0, 100]^2$, $\alpha, \beta, \gamma, \delta \in [0, 100]$, $\sigma^2 \in [10^{-6}, 10^2]$, and additionally $\kappa \in [10^{-20}, 10^{50}]$. Finally, a step-size of $\Delta = 5 \cdot 10^{-3}$ is chosen for Fenrir's probabilistic numerical integration.

FitzHugh–Nagumo The FitzHugh–Nagumo neuronal model (FitzHugh, 1955; Nagumo et al., 1962) is given by the ODE

$$\dot{y}_1 = c \left(y_1 - \frac{y_1^3}{3} + y_2 \right), \tag{56a}$$

$$\dot{y}_2 = -\frac{1}{c} \left(y_1 - a - b y_2 \right). \tag{56b}$$

We consider ground-truth parameters $a = 0.2$, $b = 0.2$, $c = 3.0$, and a true initial value $y_0 = [-1, 1]^\intercal$. The experimental data is generated on the grid $t_i \in \mathbb{T}_{\text{D}} = \{0.0, 0.5, \dots, 10.0\}$, by disturbing a high-confidence numerical simulation of



(a) Lotka–Volterra with low observation noise.

(b) Lotka–Volterra with high observation noise.

(c) FitzHugh–Nagumo with low observation noise.

(d) FitzHugh–Nagumo with high observation noise.

Figure 9. **Absolute parameter errors.**

the true trajectory with Gaussian noise $v(t) \sim \mathcal{N}(0, \sigma^2 \cdot I)$, for two noise levels $\sigma^2_{\text{low}} = 0.005$ and $\sigma^2_{\text{high}} = 0.05$. The full set of (hyper)parameters to be estimated by Fenrir is then $\theta = \{y_0, \alpha, \beta, \gamma, \delta, \sigma\}$, as well as the diffusion κ. All of which are jointly optimised via L-BFGS, while assuming bounds $y_0 \in [-100, 100]^2$, $a, b, c \in [0, 100]$, $\sigma^2 \in [10^{-6}, 10^2]$, and $\kappa \in [10^{-20}, 10^{50}]$. Fenrir's physics-enhanced prior is computed with a step size $\Delta = 10^{-2}$.

C.3. Additional Details on Section 5.2: "Model Selection"

The Lotka–Volterra model with ground-truth parameters as described in Appendix C.2 is extended to a set of four candidate models, via the following additional ODEs:

$$\dot{y_1} = \alpha y_1^2 - \beta y_2, \tag{57a}$$

$$\dot{y_2} = -\gamma y_2. \tag{57b}$$

By combining these two wrong equations with the true ODEs, we obtain for models M_{ij}, with $i, j \in \{0, 1\}$ indicating if the correct (1) or incorrect equation (0) has been used; for instance, M_{01} contains Equation (57a) and Equation (55b). The experimental data is generated as described in Appendix C.2, with a "low" noise setting of $\sigma^2_{\text{low}} = 0.01$. All parameters are optimised jointly by Fenrir via L-BFGS, with bounds for parameters and initial values chosen as in Appendix C.2.

C.4. Additional Details on Section 5.3: "Partially Observed System States"

The compartmental SEIR model (Hethcote, 2000) describes the fractions of a population that are susceptible (S), exposed (E), infected (I; i.e. diagnosed by a positive test), and recovered (R). It is given in as differential equations

$$\dot{S} = -(\beta_E \cdot S \cdot E + \beta_I \cdot S \cdot I), \tag{58a}$$

$$\dot{E} = \beta_E \cdot S \cdot E + \beta_I \cdot S \cdot I - \gamma \cdot E, \tag{58b}$$

$$\dot{I} = \gamma \cdot E - \lambda \cdot I, \tag{58c}$$

$$\dot{R} = \lambda \cdot I. \tag{58d}$$

with infection rates β_E and β_I, transition rate γ from exposure to infection, and recovery / death rate λ. Following Menda et al. (2021), which used an extension of the SEIR model to explain COVID-19 outbreaks, we consider ground-truth parameters $\beta_I = 0$, $\beta_E = 0.5$, $\gamma = 1/5$, and $\lambda = 1/21$ (the latter two correspond to realistic estimates of transition and recovery rate in COVID-19, given by Lauer et al. (2020); Bi et al. (2020)). Furthermore, we generate data on the time grid $\mathbb{T}_{\mathsf{D}} = \{30, 31, \dots, 100\}$ from initial values $E_0 = 10^{-4}$, $I_0 = 10^{-5}$, $R_0 = 0$, and $S_0 = 1 - E_0 - I_0$ at time $t_0 = 0$, as $u(t_i) = H \cdot \hat{y}(t_i) + v(t_i)$, with a measurement matrix

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{59}$$

such that only the infected and recovered population is measured, and disturbed by Gaussian noise $v(t) \sim \mathcal{N}(0, \sigma^2 \cdot I)$ with $\sigma^2 = 5 \cdot 10^{-4}$.

Instead of estimating the full initial state, we parameterize it by the initial exposed and infected population count:

$$y_0(E_0, I_0) = \begin{bmatrix} 1 - E_0 - I_0, & E_0, & I_0, & 0 \end{bmatrix}^{\mathsf{T}}. \tag{60}$$

Thus, the parameters to be estimated by Fenrir in this experiment are $\theta = \{E_0, I_0, \beta_E, \gamma, \lambda, \sigma\}$, as well as the diffusion κ. All parameters are jointly optimised via L-BFGS, with bounds $E_0, I_0, \beta_E, \gamma, \lambda \in [0, 1]$, $\sigma^2 \in [10^{-6}, 10^2]$, and $\kappa \in [10^{-20}, 10^{20}]$. In each experiment, ODE parameters β_E, γ, λ are initialised as uniformly random; the starting values for E_0, I_0 are initialised as absolute values of samples from a Gaussian $\mathcal{N}(0, 10^{-2})$. Fenrir's probabilistic numerical integration is performed with a step size $\Delta = 0.2$.

C.5. Additional Details on Section 5.4: "Dynamical Systems with Fast Oscillations"

The considered pendulum system is given by a second-order ODE $\ddot{y} = -\frac{g}{L} \sin(y)$, which can be transformed to the following first-order equations

$$\dot{y_1} = y_2, \tag{61a}$$

$$\dot{y_2} = -\frac{g}{L} \sin(y_1), \tag{61b}$$

with the gravity constant $g = 9.81$. We assume a ground-truth parameter $L = 1$ and an initial value $y_0 = [0, \pi/2]$. The observation data is generated as $u(t_i) = \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \hat{y}(t_i) + v_i$, with observation noise $v(t_i) \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 0.1$, on the grid $t_i \in \mathbb{T}_{\mathsf{D}} = \{0.01 \cdot i\}_{i=0}^{1000}$. In the corresponding experiment, we found it to be beneficial to first optimize the noise σ and diffusion parameter κ, before jointly optimizing all model parameters $\theta = \{y_0, L, \sigma\}$ and the diffusion κ. while assuming bounds $y_0 \in [-100, 100]^2$, $L \in [0, 100]$, $\sigma^2 \in [10^{-8}, 10^4]$, and $\kappa \in [10^{-20}, 10^{50}]$. Finally, Fenrir's physics-enhanced prior is computed with a fixed step size $\Delta = 0.1$.