arXiv:2002.09301v1 [stat.ML] 21 Feb 2020

Differentiable Likelihoods for Fast Inversion of 'Likelihood-Free' Dynamical Systems

Hans Kersting [*] [1 2 3] Nicholas Krämer [*] [1]
Martin Schiegg [3] Christian Daniel [3] Michael Tiemann [3] Philipp Hennig [1 2]

Abstract

Likelihood-free (a.k.a. simulation-based) inference problems are inverse problems with expensive, or intractable, forward models. ODE inverse problems are commonly treated as likelihood-free, as their forward map has to be numerically approximated by an ODE solver. This, however, is not a fundamental constraint but just a lack of functionality in classic ODE solvers, which do not return a likelihood but a point estimate. To address this shortcoming, we employ Gaussian ODE filtering (a probabilistic numerical method for ODEs) to construct a local Gaussian approximation to the likelihood. This approximation yields tractable estimators for the gradient and Hessian of the (log-) likelihood. Insertion of these estimators into existing gradient-based optimization and sampling methods engenders new solvers for ODE inverse problems. We demonstrate that these methods outperform standard likelihood-free approaches on three benchmark-systems.

1. Introduction

Inferring the parameters of dynamical systems that are defined by ordinary differential equations (ODEs) is of importance in almost all areas of science and engineering. Despite the wide range of available ODE inverse problem solvers, simple random-walk Metropolis methods remain the go-to solution; see e.g. Tarantola (2005, Section 2.4). That is to say that ODE inverse problems are routinely treated as if their forward problems were black boxes. The reason usually cited for this generic approach is that ODE forward solutions are highly non-linear and numerically intractable for all but the most trivial cases. Therefore, it is common to consider ODE inverse problems as 'likelihood-free' inference (read: intractable likelihood)—a.k.a. simulation-based inference or, in the Bayesian case, Approximate Bayesian

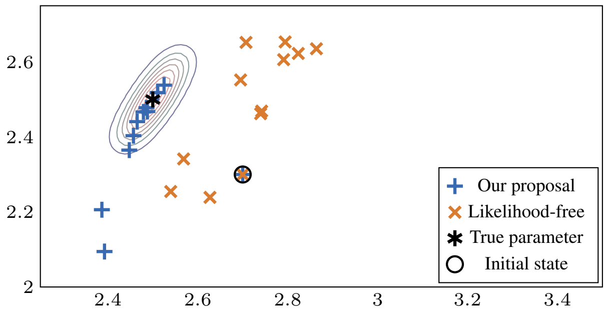

Figure 1. Inference on the logistic ODE. First twelve sampled parameter of likelihood-free inference and our proposed method. Details in text.

Computation (ABC); see Cranmer et al. (2019) for an up-to-date examination of these closely-related areas.

We here argue that, at least for ODEs, this approach is mistaken. If a dynamical system is accurately described by an ODE, its explicit mathematical definition should be exploited to design efficient algorithms—not ignored and treated as a black-box, likelihood-free inference problem. To this end, we construct a local Gaussian approximation of the likelihood by Gaussian ODE Filtering, a probabilistic numerical method (PNM) for ODE forward problems. (Appendix A provides a concise introduction to Gaussian ODE filtering; Tronarp et al. (2019) offer a more detailed presentation. See Hennig et al. (2015) or Oates & Sullivan (2019) for a broad introduction to PNMs.) The key insight of our work is that there *is* a likelihood in simulations of ODEs, and in fact it can be approximated cheaply, and analytically: The mean estimate $\boldsymbol{m}_\theta$ of the forward solution computed by Gaussian ODE filters can be linearized in the parameter θ, so that gradient, Hessian, etc. of the approximated log-likelihood can—via a cheap estimator J of the Jacobian of the map $\theta \mapsto \boldsymbol{m}_\theta$—be computed in closed form (Section 5). In this way, the probabilistic information from Gaussian ODE filtering yields a tractable, twice-differentiable likelihood for 'likelihood-free' ODE inverse problems. This enables the use of first and second-order optimization or sampling methods (cf. Figure 1).

Much thought has been devoted to improving the slow run-

[*]Primary author [1]University of Tübingen, Germany [2]Max Planck Institute for Intelligent Systems, Tübingen, Germany [3]Bosch Center for Artificial Intelligence, Renningen, Germany.

times of ODE inverse inference—which is due to the laborious explicit numerical integration per parameter. In machine learning, e.g., authors have proposed to reduce the amount of necessary parameters by active learning with Gaussian process (GP) surrogate likelihoods (Meeds & Welling, 2014), or even to avoid numerical integration altogether by gradient matching (Calderhead et al., 2008). This paper adds a new way to reduce the amount of parameters by employing gradient (and Hessian) estimates of the log-likelihood.

Contributions The main contributions are twofold: *Firstly*, we introduce tractable estimators for the gradients and Hessian matrices of the log-likelihood of ODE inverse problems by Gaussian ODE filtering. To derive these estimators, we construct a new estimator J for the Jacobian of the forward map. We theoretically support the use of J by a decomposition of the true Jacobian into J and a sensitivity term S (see Theorem 1), as well as an upper bound on its approximation error (see Theorem 2). *Secondly*, we propose a range of new solvers which require gradients and/or Hessians, by inserting these estimators into first and second-order optimization and sampling methods. The utility of these algorithms is demonstrated by experiments on three benchmark ODEs where they outperform their gradient-free counterparts.

2. Problem Setting

We consider a dynamical system defined by the ODE

$$\dot{x}(t) = f\left(x(t), \theta\right), \qquad x(0) = x_0 \in \mathbb{R}^d, \tag{1}$$

on the finite time domain $t \in [0, T]$ for some $T > 0$, with parametrized vector field $f : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^d$. We restrict our attention to choices of f satisfying the following

Assumption 1. *$f(x, \theta) = \sum_{i=1}^{n} \theta_i f_i(x)$, for some continuously differentiable $f_i : \mathbb{R}^d \to \mathbb{R}^d$, for all $i = 1, \dots, n$.*

The necessity for this assumption will become evident in Section 3.1.1. It is not very restrictive: e.g. the corresponding assumption in Gorbach et al. (2017, eq. (10)) is stronger. In fact, most standard ODEs collected in Hull et al. (1972, Appendix I), a standard set of ODE benchmarking problems, satisfy Assumption 1 either immediately or after reparametrization. If the initial value x_0 is unknown too (as is often the case in practice), it can be treated as a parameter by defining a new parameter vector $\left(x_0^\intercal, \theta^\intercal\right)^\intercal \in \mathbb{R}^{d+n}$; see eq. (10). Solving eq. (1), for a given θ, with a numerical method is known as the *forward problem*.

For the *inverse problem*, we assume the dynamical system described by eq. (1) with *unknown* true parameter θ^*. The true trajectory $x = x_{\theta^*}$ is observed under additive, zero-mean Gaussian noise at M discrete times $0 \leq t_1 < \cdots < t_M \leq T$:

$$z(t_i) \coloneqq x(t_i) + \varepsilon_i \in \mathbb{R}^d, \quad \varepsilon_i \sim \mathcal{N}(0, \Sigma_i), \tag{2}$$

for all $i \in \{1, \dots, M\}$. Below we assume, w.l.o.g., that $\Sigma_i = \Sigma$, for all $i \in \{1, \dots, M\}$. We define the stacked data across M time points and d dimensions as

$$\boldsymbol{z} \coloneqq \left[z_1^\intercal(t_1), \dots, z_1^\intercal(t_M), \dots, z_d^\intercal(t_1), \dots, z_d^\intercal(t_M)\right]^\intercal,$$

and analogously, for all $\theta \in \Theta$, the true solution at these points as $\boldsymbol{x}_\theta$. The inverse problem consists of inferring the parameter θ^* that generated the data through eq. (2). For the sake of readability, we will assume w.l.o.g. that $d = 1$. Under these conventions, eq. (2) is equivalent to

$$p(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{z}; \boldsymbol{x}, \sigma^2 I_M\right) \tag{3}$$

for some $\sigma^2 > 0$, where I_M is the $M \times M$ identity matrix.

3. Likelihoods by Gaussian ODE Filtering

The prevailing view on the uncertainty in inverse problems only considers the aleatoric uncertainty Σ_i from eq. (2) and ignores the epistemic uncertainty over the quality of the employed numerical approximation $\hat{x}_\theta$ of x_θ. In other words, the likelihood of the forward problem, $p(\boldsymbol{x}_\theta \mid \theta)$, is commonly treated as a Dirac distribution $\delta(\boldsymbol{x}_\theta - \hat{\boldsymbol{x}}_\theta)$ which yields the *uncertainty-unaware likelihood*

$$\begin{aligned} p(\boldsymbol{z} \mid \theta) &= \int p(\boldsymbol{z} \mid \boldsymbol{x}_\theta) p(\boldsymbol{x}_\theta \mid \theta) \, \mathrm{d}\boldsymbol{x}_\theta && (4) \\ &= \int p(\boldsymbol{z} \mid \boldsymbol{x}_\theta) \delta(\boldsymbol{x}_\theta - \hat{\boldsymbol{x}}_\theta) \, \mathrm{d}\boldsymbol{x}_\theta && (5) \\ &\overset{\text{eq. (3)}}{=} \mathcal{N}\left(\boldsymbol{z}; \hat{\boldsymbol{x}}_\theta, \sigma^2 I_M\right). && (6) \end{aligned}$$

as the 'true' intractable likelihood. This, however, ignores the epistemic uncertainty over the accuracy $\hat{x}_\theta$ which leads to overconfidence. This uncertainty is due to the discretization error of the numerical solver used to compute $\hat{x}_\theta$, and can only be avoided for the most trivial ODEs. This problem has previously been recognized in, e.g., Conrad et al. (2017, Section 3.2) and Abdulle & Garegnani (2018, Section 8) who, as a remedy, construct a 'cloud' of possible solutions by running a classical solver multiple times with a prespecified accuracy. This, unfortunately, requires the computational invest of several forward solves for the same θ, which could instead be used for additional θ, or higher accuracy.

To obtain such uncertainty quantification more cheaply, we employ Gaussian ODE filtering with a once-integrated Brownian motion (IBM) prior on x; see Appendix A.2 for a short introduction. This amounts—e.g. in the notation of Tronarp et al. (2019)—to setting $q = 1$. Gaussian ODE filtering has the advantage over other numerical solvers, probabilistic or classical, that we can compute gradients of the likelihood, as demonstrated below. For a given θ, the Gaussian ODE filter computes a multivariate normal distribution over x_θ at a set of $N = T/h$, for notational simplicity, equidistant time points $\{0, h, \dots, Nh\}$ with step size $h > 0$. This set is, w.l.o.g., assumed to contain the data time points $\{t_1, \dots, t_M\}$ from eq. (2), i.e. we assume the existence of a set of integers $\{l_1, \dots, l_M\}$ such that $t_i = l_i h$. (The w.l.o.g. assumption can otherwise be satisfied by interpolating along the dynamic model; see eq. (36) in Appendix A.)

3.1. The Filtering Distribution

The Gaussian ODE filter returns the so-called (posterior) filtering distribution over the ODE solution $\boldsymbol{x}_\theta$, given by

$$p(\boldsymbol{x}_\theta \mid \theta) = \mathcal{N}(\boldsymbol{x}_\theta; \boldsymbol{m}_\theta, \boldsymbol{P}), \tag{7}$$

with $\boldsymbol{m}_\theta \in \mathbb{R}^M$ and $\boldsymbol{P} \in \mathbb{R}^{M\times M}$ given below by eq. (10) and eq. (18), respectively. This probabilistic likelihood yields the new *uncertainty-aware likelihood*

$$p(\boldsymbol{z} \mid \theta) = \int p(\boldsymbol{z} \mid \boldsymbol{x}_\theta)\mathcal{N}(\boldsymbol{x}_\theta; \boldsymbol{m}_\theta, \boldsymbol{P})\, \mathrm{d}\boldsymbol{x}_\theta \tag{8}$$

$$\overset{\text{eq. (3)}}{=} \mathcal{N}(\boldsymbol{z}; \boldsymbol{m}_\theta, \boldsymbol{P} + \sigma^2 I_M) \tag{9}$$

which has two advantages over the uncertainty-unaware likelihood from eq. (6):

1. The filtering mean $\boldsymbol{m}_\theta$ can be linearized in θ, as specified below in eq. (10). This yields an estimate J of the Jacobian matrix of $\theta \mapsto \boldsymbol{m}_\theta$ which implies estimators of gradients and Hessian matrices of the likelihood; see eqs. (26) and (27). These estimators are useful to guide samples of θ into regions of high likelihood by the gradient-based sampling and methods defined in Section 6 below.
2. The variance $\boldsymbol{P}$ captures the average-case squared (epistemic) error $\|\boldsymbol{m}_\theta - \boldsymbol{x}_\theta\|^2$, and can be added to the (aleatoric) variance Σ_i; see eq. (9). Unless $\boldsymbol{P} \ll \sigma^2 I_M$, this prevents over-confidence, as visualized in Figure 2.

In the following two subsections, we provide explicit formulas for $\boldsymbol{m}_\theta$ and $\boldsymbol{P}$. A detailed derivation of these formulas is given in Appendix B.

3.1.1. The Filtering Mean

Under Assumption 1, the filtering mean $\boldsymbol{m}_\theta = [m_\theta(t_1), \dots, m_\theta(t_M)]^\intercal$ is given by

$$\boldsymbol{m}_\theta = \begin{bmatrix} \mathbb{1}_M & J \end{bmatrix} \begin{bmatrix} x_0 \\ \theta \end{bmatrix} = x_0 \cdot \mathbb{1}_M + J\theta \in \mathbb{R}^M, \tag{10}$$

where $\mathbb{1}_M = [1, \dots, 1]^\intercal$ denotes a vector of M ones. Hence, $\boldsymbol{m}_\theta$ is linear in θ as well as in the extended parameter vector $[x_0, \theta^\intercal]^\intercal$. (A more detailed derivation of eq. (10) is provided in Appendix B.3.) Here,

$$J := KY \in \mathbb{R}^{M\times n} \tag{11}$$

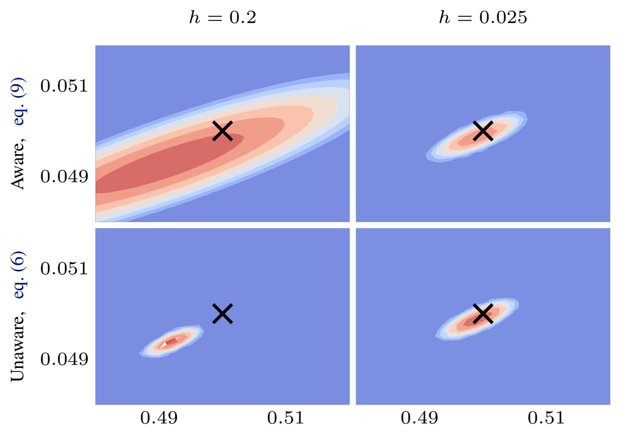

Figure 2. Uncertainty-(un)aware likelihoods, eqs. (6) and (9) w.r.t. (θ_1, θ_2) of Lotka-Volterra ODE, eq. (30), with fixed $(\theta_3, \theta_4) = (0.05, 0.5)$. θ_1 on x and θ_2 on y-axis. Black cross is true parameter. The unaware likelihood is overconfident for the large step size $h = 0.2$ (i.e. for large $\boldsymbol{P}$) while the aware likelihood has calibrated uncertainty. For the small $h = 0.025$ this effect is less pronounced as $\boldsymbol{P}$ is small.

is an estimator of the Jacobian matrix of the map $\theta \mapsto \boldsymbol{m}_\theta$, as we show in Theorem 1 below. This estimator is equal to the product of the *kernel prefactor* K and the *evaluation factor* Y. The kernel prefactor K is given by

$$K := \left[\kappa_1, \dots, \kappa_M\right]^\intercal \in \mathbb{R}^{M\times N}, \tag{12}$$

whose i-th row is

$$\kappa_i := \left[\tilde{\kappa}_i^\intercal, 0, \dots, 0\right]^\intercal \in \mathbb{R}^N, \tag{13}$$

which is defined by

$$\tilde{\kappa}_i := \left[{}^\partial K^\partial(h:t_i) + R \cdot I_{l_i}\right]^{-1} k^\partial(h:t_i, t_i) \in \mathbb{R}^{l_i}, \tag{14}$$

for some measurement variance $R \geq 0$. Here, $k^\partial = \partial k(t,t')/\partial t'$ and ${}^\partial k^\partial = \partial^2 k(t,t')/\partial t \partial t'$ are derivatives of the IBM kernel k, and, analogously, the cross-covariance w.r.t. the kernel ${}^\partial k$ and the kernel Gram matrix w.r.t. the kernel ${}^\partial k^\partial$ up to time t_i are denoted by

$$k^\partial(h:t_i, t_i) := \left[k^\partial(t_i, h), \dots, k^\partial(t_i, t_i)\right]^\intercal, \text{ and} \tag{15}$$

$${}^\partial K^\partial(h:t_i) := \begin{bmatrix} {}^\partial k^\partial(h,h) & \cdots & {}^\partial k^\partial(l_i h, l_i h) \\ \vdots & \ddots & \vdots \\ {}^\partial k^\partial(l_i h, h) & \cdots & {}^\partial k^\partial(l_i h, l_i h) \end{bmatrix}. \tag{16}$$

Now, recall Assumption 1. For a given θ, the entries of the evaluation factor $Y \in \mathbb{R}^{N\times n}$ are

$$y_{ij} := f_j(m_\theta^-(ih)) - f_j(x_0), \tag{17}$$

for all $i = 1, \dots, N$ and $j = 1, \dots, n$, where $m_\theta^-(ih)$ is the predictive mean of the ODE Filter at $t = ih$. Note that the Gaussian ODE Filter computes the $f_j(m_\theta^-(ih))$ and $f_j(x_0)$ for every forward solve as intermediate quantities, to evaluate the right-hand side of eq. (1). Hence, Y is freely accessible with every filtering distribution, eq. (7). However, as an estimate of $x_\theta(ih)$, $m_\theta^-(ih)$ depends on θ in a nonlinear and potentially sensitive way. By ignoring this dependence in the above notation, we, strictly speaking, also omit the dependence of Y and, thereby, J on θ (more in Appendix B.3). For this reason, J is not the true Jacobian of $\theta \mapsto \boldsymbol{m}_\theta$ but only an estimator (see Section 3.2).

3.1.2. The Filtering Covariance

The entries of the covariance matrix $\boldsymbol{P} := \operatorname{diag}(P(t_1), \dots, P(t_M)) \in \mathbb{R}^{M \times M}$ of the filtering distribution from eq. (7) coincide with the GP-posterior variances, i.e.

$$P(t_i) = \begin{bmatrix} k(h,h) & \cdots & k(l_i h, l_i h) \\ \vdots & \ddots & \vdots \\ k(l_i h, h) & \cdots & k(l_i h, l_i h) \end{bmatrix} - k^\partial(h : t_i, t_i)^\intercal \times \left[{}^\partial K^\partial(h : t_i) + R \cdot I_l\right]^{-1} k^\partial(h : t_i, t_i), \quad (18)$$

and are hence independent of θ. (See Appendix B.2 for a detailed derivation of eq. (18).)

3.2. Decomposition of the True Jacobian

Next, we give an explicit decomposition of the true Jacobian into the estimator J, the kernel prefactor K and a sensitivity term S.

Theorem 1. *Under Assumption 1, the true Jacobian $D\boldsymbol{m}_\theta \in \mathbb{R}^{M \times n}$ of $\theta \mapsto \boldsymbol{m}_\theta$ has the analytic form*

$$D\boldsymbol{m}_\theta := [\nabla_\theta m(t_1), \dots, \nabla_\theta m(t_M)]^\intercal = J + KS, \quad (19)$$

where the sensitivity term *S is defined by*

$$S := \left[\Lambda_1^\intercal \theta, \dots, \Lambda_N^\intercal \theta\right]^\intercal \in \mathbb{R}^{N \times n}. \quad (20)$$

Here, $\Lambda_j = \left[\lambda_{kl}(jh)\right]_{kl}$ is the $n \times n$ matrix with entries

$$\lambda_{kl}(jh) := \frac{\mathrm{d}}{\mathrm{d}x} f_l(m_\theta^-(jh)) \cdot \frac{\partial}{\partial \theta_k} m_\theta^-(jh). \quad (21)$$

Proof. See Appendix C. □

Thus, KS is the exact approximation error of J.

4. Bound on Approximation Error of J

In this section, we provide a bound on the approximation error of J under the following assumptions.

Assumption 2. *The first-order partial derivatives of f_i, $1 \leq i \leq N$, are bounded and globally L-Lipschitz, for $L > 0$.*

Assumption 2 is required to bound the global error of the ODE forward solution by Kersting et al. (2019, Thm. 6.7).

Assumption 3. *For the computation of J we only use a maximum of $\bar{N} \leq N$ time points, for some finite $\bar{N} \in \mathbb{N}$.*

Assumption 3 precludes the condition number of the K and S from growing arbitrarily large, thereby preventing numerical instability. While this restriction is necessary for Theorem 2, it is not relevant in practice because we are computing with a non-zero step size $h > 0$ anyway so that many different parameters θ can be simulated.

Theorem 2. *If $\Theta \subset \mathbb{R}^n$ is compact and $R > 0$, then it holds true, under Assumptions 1 to 3, that*

$$\|J - D\boldsymbol{m}_\theta\| \leq C(T)\left(\|\nabla_\theta x_\theta\| + h\right) \quad (22)$$

for sufficiently small $h > 0$, where $C(T) > 0$ is a constant that depends on T.

Proof. See Appendix D. □

Intuitively, this upper bound can be thought of as a decomposition of the approximation error of the 'sensitivity-unaware' estimator J into a summand proportional to the ignored sensitivity $\|\nabla_\theta x_\theta\|$ and the global integration error of the ODE filter, which is bounded by $C(T)h$ (Kersting et al., 2019, Thm. 6.7).

5. Gradient and Hessian Estimators

We observe that the uncertainty-aware likelihood, eq. (9), can be written in the form

$$p(\boldsymbol{z} \mid \theta) = \frac{e^{-E(\boldsymbol{z})}}{Z}, \quad (23)$$

with evidence $Z > 0$ and negative log-likelihood

$$E(\boldsymbol{z}) := \frac{1}{2}\left[\boldsymbol{z} - \boldsymbol{m}_\theta\right]^\intercal \left[\boldsymbol{P} + \sigma^2 I_M\right]^{-1} \left[\boldsymbol{z} - \boldsymbol{m}_\theta\right] \quad (24)$$

$$\overset{\text{eq. (10)}}{=} \frac{1}{2}\left[\boldsymbol{z} - x_0 \cdot \mathbb{1}_M - J\theta\right]^\intercal \left[\boldsymbol{P} + \sigma^2 I_M\right]^{-1} \times \left[\boldsymbol{z} - x_0 \cdot \mathbb{1}_M - J\theta\right]. \quad (25)$$

For a given value of the Jacobian estimator J, the thereby-implied gradient and Hessian estimators are, by application of the chain rule,

$$\hat{\nabla}_\theta E(\boldsymbol{z}) := -J^\intercal \left[\boldsymbol{P} + \sigma^2 I_M\right]^{-1} \left[\boldsymbol{z} - \boldsymbol{m}_\theta\right], \quad \text{and} \quad (26)$$

$$\hat{\nabla}_\theta^2 E(\boldsymbol{z}) := J^\intercal \left[\boldsymbol{P} + \sigma^2 I_M\right]^{-1} J. \quad (27)$$

(See Figure 3 for a visualization of these estimators.) Appendix E provides versions of these estimators for Bayesian inference of θ.

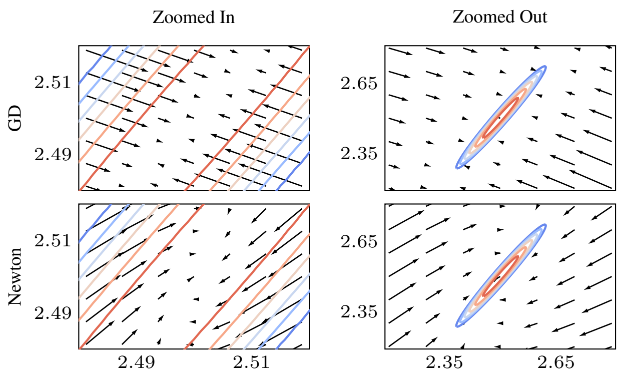

Figure 3. Directions of gradient descent (GD) and Newton using eqs. (26) and (27) on the logistic ODE; around mode (left) and globally (right). Globally, GD points more directly to the high-probability region. Within this region, however, Newton is better directed to the mode.

6. New Gradient-Based Methods

By deriving gradient and Hessian estimators of the negative log-likelihood, we have removed the need for 'likelihood-free' inference. This enables the use of two classes of inference methods for θ which could not otherwise be applied: gradient-based *optimization* and gradient-based *sampling*.

6.1. Gradient-Based Optimization

In principle, all first and second-order optimization algorithms (e.g. Bottou et al. (2018)), are now applicable by eqs. (26) and (27)—such as (stochastic) gradient descent (GD), (stochastic) Newton (NWT), Gauss-Newton and natural Gradient descent.

6.2. Gradient-Based Sampling

Likewise, all gradient-based MCMC schemes are now available. Classical gradient-based samplers include Langevin Monte Carlo (LMC) (Roberts & Tweedie, 1996) and Hamiltonian Monte Carlo (HMC) (Betancourt, 2017). They are known to be more efficient than gradient-free samplers in finding and covering regions of high probability (MacKay, 2003, Section 30.1). While their standard form only makes use of gradients, more sophisticated versions include second-order information as well: When the likelihood is ill-conditioned (i.e. it varies much more quickly in some directions than others), it is advantageous to precondition the proposal distribution with a suitable matrix (Girolami & Calderhead, 2011). A popular choice for the preconditioner is the Hessian (Qi & Minka, 2002). Hence, we can precondition LMC and HMC that use eq. (26) as a gradient with the Hessian estimator from eq. (27). For LMC, this leads to the proposal distribution

$$\pi(\theta_{i+1} \mid \theta_i) = \theta_i - \rho[\hat{\nabla}^2_\theta E(\theta_i)]^{-1}\hat{\nabla}_\theta E(\theta_i) + \xi_i, \quad (28)$$

$$\xi_i \sim \mathcal{N}(0, \hat{\nabla}^2_\theta E(\theta_i)), \quad (29)$$

where ρ is the proposal width. (Analogous formulas hold for HMC.) Below, we refer to the so-preconditioned versions of LMC and HMC as PLMC and PHMC.

6.3. Choice of Hyperparameters

Recall that the parameters σ and R stem from the data and the accuracy of the ODE model (Kersting et al., 2019, Section 2.3), and that we only consider once-integrated Brownian motion priors in this paper. Therefore, the only remaining hyperparameter is the diffusion scale σ_{dif} which controls the width of the variance $\boldsymbol{P}$; see Appendices B.1 and B.2. There are two ways to set it: either as a local (Schober et al., 2018, eq. (46)) or as a global (Tronarp et al., 2019, eq. (41)) maximum-likelihood estimate, which can both be computed from intermediate quantities of the forward solves.

6.4. Computational Cost

The additional computational cost—on top of the employed classical optimization/sampling methods—is equal to the cost of computing the inserted gradient (and Hessian) estimators. After pre-computing J and $[\boldsymbol{P}+\sigma^2 I_M]^{-1}$, this cost consists, by eqs. (26) and (27), only of a negligibly cheap matrix multiplication. The Jacobian estimator $J = KY$ is, by eq. (11), a matrix product of the precomputable kernel prefactor K and the evaluation factor Y. Y is almost free, as it is by eq. (17) only composed of terms that the Gaussian ODE filter computes anyway; see eq. (45) in Appendix A.2. The pre-computation of K requires the inversion of the M kernel Gram matrices $\{{}^\partial K^\partial(h : t_i),\ i = 1, \dots, M\}$, which can have a maximum dimension of $(N-1)\times(N-1)$. This inversion can, however, be executed in linear time since ${}^\partial k^\partial$ is a Markov kernel (Hartikainen & Särkkä, 2010). Hence, K is in $\mathcal{O}(MN)$ and, as $M \leq N$, in $\mathcal{O}(N^2)$. Finally, the cost of inverting the $M \times M$ matrix $[\boldsymbol{P} + \sigma^2 I_M]$ is in $\mathcal{O}(N^3)$, as $M \leq N$. Since K and $\boldsymbol{P}$ are independent of θ, this $\mathcal{O}(N^3)$ cost is only required once. Thus, the additional computational cost is in $\mathcal{O}(N^3)$ w.r.t. the number of time steps $N = T/h$ and linear (but almost negligible) w.r.t. the number of simulated parameters θ. As a large number of θ is usually required, the overall overhead is small.

7. Experiments

To test the hypothesis that the gradient and Hessian estimators $[\hat{\nabla}_\theta E(\boldsymbol{z}), \hat{\nabla}^2_\theta E(\boldsymbol{z})]$ of the log-likelihood are useful despite their approximate nature, we compare the new optimization and sampling methods from Section 6—which use these estimators as if exact—with the standard 'likelihood-free' approach, i.e. with random search (RS) optimization and random-walk Metropolis (RWM) sampling.

7.1. Setup and Methods

As benchmark systems, we choose the popular Lotka–Volterra (LV) predator-prey model and the more challenging biochemical dynamics of glucose uptake in yeast (GUiY). For more generality, we add the chemical protein signalling transduction (PST) dynamics which violate Assumption 1 and have to be linearized. We consider our hypothesis validated if the new gradient-based algorithms outperform the conventional 'likelihood-free' methods (RS, RWM) on these three systems. All datasets are, as in eq. (3), generated by adding Gaussian noise to the solution x_{θ^*} for some true parameter θ^*.
Out of the new family of gradient-based optimizers and samplers introduced in Section 6, we evaluate only the most basic ones: gradient descent (GD) and Newton's method (NWT) for optimization, as well as PLMC and PHMC for sampling. This isolates the impact of the gradient and Hessian estimators more clearly. The required gradient and Hessian estimators are computed as detailed above. We employ the original fixed step-size RS by Rastrigin (1963), and the RWM version from MacKay (2003, Chapter 29). For all optimizers, we picked the best the step size and, for all samplers, the best proposal width within the interval $[10^{-16}, 10^{0}]$ which is wide enough to contain all plausible values. To make these experiments an ablation study for the gradient and Hessian estimators, we use Gaussian ODE filtering as a forward solver in all methods—which is similar to classical solvers anyway (Schober et al., 2018, Section 3). Since in all below experiments $\boldsymbol{P} \gg \sigma^2 I_M$, the gradient and Hessian estimates are scale-invariant w.r.t. hyperparameter σ^2_{dif}, as can be seen from eqs. (26) and (27): In this regime, $\boldsymbol{P}$ simply scales the step-size of the gradient, and $\boldsymbol{P}$ cancels out of the Hessian, making it invariant to this scale. The same applies in the regime $\boldsymbol{P} \ll \sigma^2 I_M$; adaptation of their relative scale, by choosing σ^2_{dif} as in Section 6.3, only matters when both error-sources are of comparable scale.

7.2. Results

We evaluate the performance of these methods over the first few iterations (steps). As optimizers and samplers differ in their purpose, we evaluate them in a slightly different way: Optimizers try to quickly find a local minimum and we plot the negative log-likelihood at each iteration (thus, lower is better). Samplers, on the other hand, try to find and cover regions of high probability (the typical set); see e.g. Betancourt (2017, Section 2). Thus, successful sampling tends to go into and stay in regions of high probability. Accordingly, we plot the likelihood values at each iteration and consider a steady (initially increasing) trajectory a success. Samplers that 'fall off' into regions of low likelihood, on the other hand, have failed. The details and results for each benchmark systems are presented next, in ascending order of complexity.

7.2.1. Lotka–Volterra

First, we study the Lotka–Volterra (LV) ODE (Lotka, 1978)

$$\dot{x}_1 = \theta_1 x_1 - \theta_2 x_1 x_2, \quad \dot{x}_2 = -\theta_3 x_2 + \theta_4 x_1 x_2, \quad (30)$$

the standard model for predator-prey dynamics. We used this ODE with initial value $x_0 = [20, 20]$, time interval $[0, 5]$ and true parameter $\theta^* = [1, 0.1, 0.1, 1]$. To generate data by eq. (3), we added Gaussian noise with variance $\sigma^2 = 0.01$ to the corresponding solution at time points $[0.5, 1, 1.5, 2, 2.5, 3., 3.5, 4., 4.5]$. The optimizers and samplers were initialized at $\theta = [0.8, 0.2, 0.05, 1.1]$, and the forward solutions for all likelihood evaluations were computed with step size $h = 0.05$. The results for optimization and sampling are depicted over 50 iterations in Figure 4. In

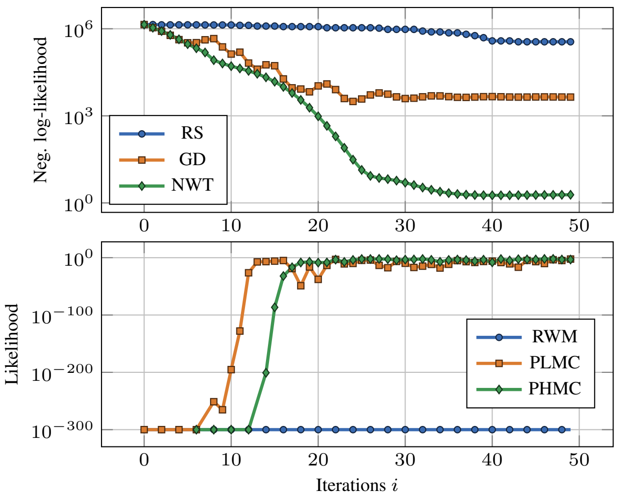

Figure 4. Comparison of **optimizers** (top) and **samplers** (bottom) on LV; see text for abbreviations. Likelihoods at 10^{-300} are placeholders for even smaller values; all RWM values are $< 10^{-300}$.

the case of optimizers, NWT outperforms GD which, in turn, outperforms RS. While PLMC and PHMC quickly reach regions of high probability, RWM does not even find likelihood values of $\geq 10^{-300}$. Thus, the gradient and Hessian estimators indeed appear to work well on LV.

7.2.2. Protein Signalling Transduction

Next, we consider the protein signalling transduction (PST) pathway. It is governed by a combination of mass-action and Michaelis–Menten kinetics:

$$\dot{S} = -\theta_1 \times S - \theta_2 \times S \times R + \theta_3 \times RS,$$
$$\dot{dS} = \theta_1 \times S,$$
$$\dot{R} = -\theta_2 \times S \times R + \theta_3 \times RS + V \times \frac{Rpp}{K_m + Rpp},$$
$$\dot{RS} = \theta_2 \times S \times R - \theta_3 \times RS - \theta_4 \times RS,$$
$$\dot{Rpp} = \theta_4 \times RS - \theta_5 \times \frac{Rpp}{K_m + Rpp}.$$

For more details, see Vyshemirsky & Girolami (2008). Due to the ratio $\frac{Rpp}{K_m+Rpp}$, Assumption 1 is violated. As a remedy, we follow Gorbach et al. (2017) in defining the latent variables $[x_1, x_2, x_3, x_4, x_5] := [S, dS, R, RS, \frac{Rpp}{K_m+Rpp}]$. This gives rise to the new linearized ODE

$$\dot{x}_1 = -\theta_1 x_1 - \theta_2 x_1 x_3 + \theta_3 x_4, \tag{31}$$
$$\dot{x}_2 = \theta_1 x_1, \tag{32}$$
$$\dot{x}_3 = -\theta_2 x_1 x_3 + \theta_3 x_4 + \theta_5 x_5, \tag{33}$$
$$\dot{x}_4 = \theta_2 x_1 x_3 - \theta_3 x_4 - \theta_4 x_4, \tag{34}$$
$$\dot{x}_5 = \theta_4 x_4 - \theta_5 x_5, \tag{35}$$

which is an approximation of the original ODE, since eq. (35) ignores the factor $(K_m + R_{pp})^{-1}$. We used this ODE with initial value $x_0 = [1, 0, 1, 0, 0]$ on time interval $[0, 100]$. Like Gorbach et al. (2017), we set the true parameter to $\theta^* = [1, 0.1, 0.1, 1]$. To generate the data by eq. (3), we added Gaussian noise with variance $\sigma^2 = 10^{-8}$ to the corresponding solution at time points $[1., 2., 4., 5., 7., 10., 15., 20., 30., 40., 50., 60., 80., 100.]$. The optimizers and samplers were initialized at $\theta = [0.24, 1.8, 0.15, 0.9, 0.05]$, and the forward solutions for all likelihood evaluations were computed with step size $h = 1.0$.

The results for optimization and sampling are depicted over 50 iterations in Figure 5. Again, the new methods outperform the conventional ones in both optimization and sampling. For optimization, NWT converges particularly fast. For sampling, both gradient-based samplers (after a fairly steep initial increase) steadily stay in regions of high probability while RWM immediately gets lost with unsuitable parameters. Hence, the gradient and Hessian estimators are beneficial on PST as well.

7.2.3. Glucose Uptake in Yeast

Last, we examine the challenging biochemical dynamics of glucose uptake in yeast (GUiY), as seen in Schillings et al. (2015). This ODE is 9-dimensional, has 10 parameters, and satisfies Assumption 1; see Appendix F for a complete mathematical definition and parameter choices. The results for optimization and sampling are depicted over 50 iterations in Figure 6. GD outperforms RS, and NWT converges even much faster than GD. Remarkably, NWT already finds very likely parameters after 5 iterations which would take RS extremely long on this 10 dimensional domain. The gradient-based samplers (PLMC, PHMC), again, stay steadily within the region of significant likelihood, while RWM first goes astray before recovering. Thus, this benchmark system reaffirms the utility of the gradient and Hessian estimators.

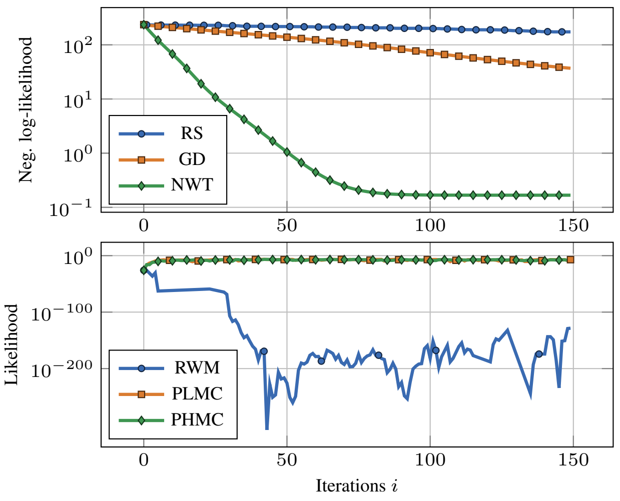

Figure 5. Comparison of **optimizers** (top) and **samplers** (bottom) on PST.

7.2.4. Summary of Experiments

On all three benchmark ODEs, the Jacobian and Hessian estimator proved useful to speed up both sampling and optimization. In the case of optimization, the new gradient-based methods consistently outperformed the classical random search. Notably, the second-order optimization was always significantly more sample-efficient than plain gradient descent—which indicates that not only the gradient but also the Hessian estimator is accurate enough to be useful. In the case of sampling, the gradient-based sampling methods, which were preconditioned by the Hessian, consistently outperformed the classical approach as well: PLMC and PHMC steadily explored a region of elevated likelihood, while the conventional random-walk Metropolis methods never reached positive likelihood (on LV), 'fell off' into regions of tiny likelihood (on PST) or wasted computational budget on less likely parameters (on GUiY). Overall, we consider these experiments first evidence for the hypothesis that the proposed gradient-based methods require drastically fewer samples than the standard 'likelihood-free' approach.

8. Related and Future Work

The following research areas are particularly closely related to this paper.

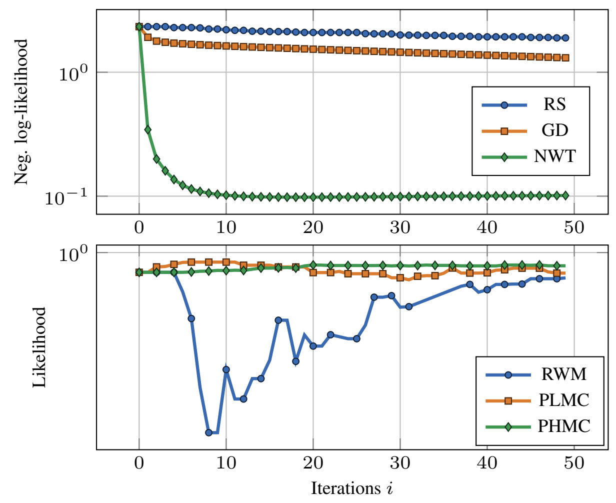

Figure 6. Comparison of **optimizers** (top) and **samplers** (bottom) on GUiY.

Probabilistic numerical methods (PNMs) There are two lines of work on PNMs for ODE forward problems: sampling- and filtering-based solvers; an up-to-date comparative discussion of these two approaches is given in Kersting et al. (2019, Section 1.2.). While this paper is the first to use filtering-based PNMs for inverse problems, there are previous methods—starting with Chkrebtii et al. (2016)—that use sampling-based solvers to integrate a non-Gaussian uncertainty-aware likelihood (cf. the Gaussian eq. (9)) into a pseudo-marginal MCMC framework; see Conrad et al. (2017), Abdulle & Garegnani (2018), Teymur et al. (2018), and Lie et al. (2019). Notably, Matsuda & Miyatake (2019) recently proposed to model the numerical errors as random variables without explicitly employing PNMs. On a related note, there are also first PNMs for PDE inverse problems; see Cockayne et al. (2017) and Oates et al. (2019).

GP-surrogate methods Modelling expensive likelihoods by GP regression is a common approach in statistics; see e.g. Sacks et al. (1989) and O'Hagan (2006). Notably, Meeds & Welling (2014) incorporated this approach into an ABC framework, and Perdikaris & Karniadakis (2016), on the other hand, into a non-Bayesian setting by efficient global optimization. While these methods also compute a GP approximation to the likelihood, they are fundamentally different as they globally model the likelihood with a GP (instead of constructing a local Gaussian approximation (see eq. (9)), and do not exploit the shape of the ODE at all.

Gradient Matching This approach fits a joint GP model of the solution and its derivatives by conditioning on the ODE. Since introduced by Calderhead et al. (2008), it has received much attention in machine learning; see Macdonald & Husmeier (2015) for a detailed review, Wenk et al. (2019, Section 1) for an up-to-date overview, and Gorbach et al. (2017) for a paper that uses a slightly stronger version of our Assumption 1. As it avoids explicit numerical integration altogether, gradient matching is fundamentally different from our method (and PNMs in general).

Sensitivity analysis This field studies the derivatives of ODE solutions with respect to parameters; see, e.g., Rackauckas et al. (2018) for an overview spanning continuous (adjoint) sensitivity analysis and automatic differentiation. Therefore, the Jacobian estimator J of the map $\theta \mapsto \boldsymbol{m}_\theta \approx \boldsymbol{x}_\theta$ from eq. (11) can be interpreted as a fast, approximate sensitivity analysis. This link is particularly interesting for modern machine learning, as sensitivity analysis is the mathematical corner stone of the recent advances by, e.g., Chen et al. (2018) in training neural networks as ODEs. It should be possible to use J for neural ODEs—as well as for all other applications of sensitivity analysis.

Future Work

We hope that this is the beginning of a new line of work on ODE inverse problems by ODE filtering. Here, we only used a Gaussian ODE filtering with once-integrated Brownian motion prior. Future work could not only examine different priors (Kersting et al., 2019, Section 2.1), but also draw from the wide range of additional ODE filters (EKF, UKF, particle filter, etc.) that were unlocked by Tronarp et al. (2019). Notably, particle ODE filtering represents the belief over the ODE solution by a set of samples (particles), and could, therefore, be integrated in the above-mentioned existing framework for sampling-based PNMs.
The utility of the Jacobian estimator J is, however, not limited to inverse problems. As it constitutes a fast, approximate sensitivity analysis, it should be compared with established methods, such as automatic differentiation and continuous sensitivity analysis (Rackauckas et al., 2018). If S (eq. (20)) could also be estimated with low overhead, it is in light of eq. (19) conceivable that the approximation error of J could be further reduced.
Either way, future work should examine which optimization and sampling methods are optimal—given that they received the (approximate) gradient and Hessian estimators $[\hat{\nabla}_\theta E(\boldsymbol{z}), \hat{\nabla}^2_\theta E(\boldsymbol{z})]$. For instance, the approximation error on these estimators might—according to Bottou et al. (2018, Section 3.3)—warrant optimization by stochastic methods such as SGD. On a related note, it should be examined whether classical theorems on limit behavior of the employed optimization and MCMC methods remain true when using these estimators, and whether our approach is indeed applicable to ODEs that violate Assumption 1—as the results from Section 7.2.2 suggest. Finally, this work should be, by the methods of lines (Schiesser & Griffiths, 2009), extendable to PDEs and, by John et al. (2019), to boundary value problems.

9. Concluding Remarks

We introduced a novel Jacobian estimator for ODE solutions w.r.t. their parameters which implies approximate estimators of the gradient and Hessian of the log-likelihood. Using these estimators, we proposed new first and second-order optimization and sampling methods for ODE inverse problems which outperformed standard 'likelihood-free' approaches—namely random search optimization and random-walk Metropolis MCMC—in all conducted experiments. Moreover, the employed Jacobian estimator constitutes a new method for fast, approximate sensitivity analysis.

Acknowledgements

HK, NK and PH gratefully acknowledge 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 Tbingen AI Center (FKZ: 01IS18039A); and funds from the Ministry of Science, Research and Arts of the State of Baden-Wrttemberg.
HK, NK and PH also gratefully acknowledge financial support by the German Federal Ministry of Education and Research (BMBF) through Project ADIMEM (FKZ 01IS18052B).

References

Abdulle, A. and Garegnani, G. Random time step probabilistic methods for uncertainty quantification in chaotic and geometric numerical integration. *arXiv:1703.03680 [math.NA]*, 2018.

Betancourt, M. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv:1701.02434 [stat.ME]*, 2017.

Bottou, L., Curtis, F., and Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Review*, 2018.

Calderhead, B., Girolami, M., and Lawrence, N. Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2008.

Chen, R., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

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, 2016.

Cockayne, J., Oates, C., Sullivan, T., and Girolami, M. Probabilistic numerical methods for PDE-constrained Bayesian inverse problems. *AIP Conference Proceedings*, 2017.

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*, 2017.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. *arXiv:1911.01429 [stat.ML]*, 2019.

Girolami, M. and Calderhead, B. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 2011.

Golub, G. and Van Loan, C. *Matrix computations*. Johns Hopkins University Press, 4th edition, 1996.

Gorbach, N. S., Bauer, S., and Buhmann, J. M. Scalable variational inference for dynamical systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Hartikainen, J. and Särkkä, S. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2010.

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

Hull, T., Enright, W., Fellen, B., and Sedgwick, A. Comparing numerical methods for ordinary differential equations. *SIAM Journal on Numerical Analysis*, 9(4):603–637, 1972.

John, D., Heuveline, V., and Schober, M. GOODE: A Gaussian off-the-shelf ordinary differential equation solver. In *International Conference on Machine Learning (ICML)*, 2019.

Kelley, W. and Peterson, A. *The Theory of Differential Equations: Classical and Qualitative*. Springer, 2010.

Kersting, H., Sullivan, T. J., and Hennig, P. Convergence rates of Gaussian ODE filters. *arXiv:1807.09737v2 [math.NA]*, 2019.

Lie, H. C., Stuart, A. M., and Sullivan, T. J. Strong convergence rates of probabilistic integrators for ordinary differential equations. *Statistics and Computing*, 2019.

Lotka, A. The growth of mixed populations: two species competing for a common food supply. *The Golden Age of Theoretical Ecology: 19231940*, 1978.

Macdonald, B. and Husmeier, D. Gradient matching methods for computational inference in mechanistic models for systems biology: A review and comparative analysis. *Frontiers in bioengineering and biotechnology*, 3, 2015.

MacKay, D. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

Matsuda, T. and Miyatake, Y. Estimation of ordinary differential equation models with discretization error quantification. *arXiv:1907.10565 [stat.ME]*, 2019.

Meeds, E. and Welling, M. GPS-ABC: Gaussian process surrogate approximate Bayesian computation. In *Uncertainty in Artificial Intelligence (UAI)*, 2014.

Oates, C., Cockayne, J., Aykroyd, R., and Girolami, M. Bayesian probabilistic numerical methods in time-dependent state estimation for industrial hydrocyclone equipment. *Journal of the American Statistical Association*, 2019.

Oates, C. J. and Sullivan, T. J. A modern retrospective on probabilistic numerics. *Statistics and Computing*, 2019.

O'Hagan, A. Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering & System Safety*, 2006.

Perdikaris, P. and Karniadakis, G. E. Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond. *Journal of the Royal Society Interface*, 13, 2016.

Qi, Y. and Minka, T. Hessian-based Markov chain Monte-Carlo algorithms. *Workshop on Monte Carlo Methods*, 2002.

Rackauckas, C., Ma, Y., Dixit, V., Guo, X., Innes, M., Revels, J., Nyberg, J., and Ivaturi, V. A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions. *arXiv:1812.01892 [math.NA]*, 2018.

Rastrigin, L. A. The convergence of the random search method in the extremal control of a many parameter system. *Automation and Remote Control*, 24, 1963.

Roberts, G. and Tweedie, R. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. Design and analysis of computer experiments. *Statistical Science*, 1989.

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

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

Schiesser, W. E. and Griffiths, G. W. *A Compendium of Partial Differential Equation Models: Method of Lines Analysis with Matlab*. Cambridge University Press, 1st edition, 2009.

Schillings, C., Sunnaker, M., Stelling, J., and Schwab, C. Efficient characterization of parametric uncertainty of complex (bio)chemical networks. *PLOS Computational Biology*, 11, 2015.

Schober, M., Duvenaud, D., and Hennig, P. Probabilistic ODE solvers with Runge–Kutta means. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

Schober, M., Särkkä, S., and Hennig, P. A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing*, 2018.

Solak, E., Murray-Smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E. Derivative observations in Gaussian process models of dynamic systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2003.

Tarantola, A. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005.

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

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*, 2019.

Vyshemirsky, V. and Girolami, M. A. Bayesian ranking of biochemical system models. *Bioinformatics*, 24(6): 833–839, 2008.

Wenk, P., Abbati, G., Bauer, S., Osborne, M. A., Krause, A., and Schölkopf, B. ODIN: ODE-informed regression for parameter and state inference in time-continuous dynamical systems. In *AAAI Conference on Artificial Intelligence*, 2019.

A. Short Introduction to Gaussian ODE Filtering

A.1. Gaussian Filtering for Generic Time Series

In signal processing, a Bayesian Filter (Särkkä, 2013, Chapter 4) does Bayesian inference of the discrete state $\{x_i;\ i = 1, \dots, N\} \subset \mathbb{R}^n$ from measurements $\{y_i;\ i = 1, \dots, N\} \subset \mathbb{R}^n$ in a *probabilistic state space model* consisting of

$$\text{a dynamic model} \quad x_i \sim p(x_i \mid x_{i-1}), \quad \text{and} \quad (36)$$

$$\text{a measurement model} \quad y_i \sim p(y_i \mid x_i). \quad (37)$$

Usually, the state x_i is assumed to be the discretization of a discrete signal $x : [0, T] \to \mathbb{R}^n$ which is *a priori* modeled by a stochastic process. Absent very specific expert knowledge, this prior is usually chosen to be a linear time-invariant (LTI) stochastic differential equation (SDE):

$$p(x) \ \sim \ X(t) = FX(t)\,\mathrm{d}t + L\,\mathrm{d}B(t), \quad (38)$$

where F and L are the drift and diffusion matrix, respectively. The corresponding dynamic model (eq. (36)) can be easily constructed by discretization of the LTI SDE (eq. (38)), as described in Särkkä & Solin (2019, Chapter 6.2). If an LTI SDE prior with Gaussian initial condition is used, $p(x)$ is a GP which implies a Gaussian dynamic model

$$p(x_i \mid x_{i-1}) = \mathcal{N}(Ax_{i-1}, Q) \quad (39)$$

for matrices A, Q that are implied by F, L from eq. (38). If additionally the measurement model (eq. (37)) is Gaussian, i.e.

$$p(y_i \mid x_i) = \mathcal{N}(Hx_i, R) \quad (40)$$

for matrices H, R, the filtering distributions $p(x_i \mid y_{1:i})$, $i = 1, \dots, N$, can be computed by Gaussian filtering in linear time. Note that the filtering distribution $p(x_i \mid y_{1:i})$ is not the full posterior distribution $p(x_i \mid y_{1:N})$ which can, however, also be computed in linear time by running a smoother after the filter. See e.g. Särkkä (2013) for more information.

A.2. Gaussian ODE Filtering

A Gaussian ODE filter is simply a Gaussian filter, as defined in Appendix A.1, with a specific kind of probabilistic state space model eqs. (36) and (37), to infer the solution $x : [0, T] \to \mathbb{R}^d$ of the ODE eq. (1), at the discrete time grid $\{0 \cdot h, \dots, N \cdot h\}$ with step size $h > 0$. The dynamic model is—as usual, recall eqs. (38) and (39)—constructed from a GP defined by a LTI SDE that incorporates the available prior information on x. The measurement model, however, is specific to ODEs as we will see next: Recall that, after $i-1$ steps, the Gaussian filter has computed the $(i-1)$-th filtering distribution

$$p(x_{i-1} \mid y_{1:i-1}) = \mathcal{N}(m_{i-1}, P_{i-1}), \quad (41)$$

which is Gaussian with mean m_{i-1}^- and covariance matrix P_{i-1}^-, and computes the predictive distribution

$$p(x_i \mid x_{i-1}) = \mathcal{N}(m_i^-, P_i^-) \quad (42)$$

by inserting eq. (39) into eq. (41). Analogous to the logic

$$f(\hat{x}(t) \approx f(x(t)) = \dot{x}(t) \quad (43)$$

of classical solvers, the Gaussian ODE Filter treats evaluations at the predictive mean m_i^-—which is a numerical approximation like $\hat{x}$—as data on $\dot{x}(ih)$. This yields the measurement model

$$p(y_i \mid x_i) = \mathcal{N}(Hx_i, R), \quad (44)$$

with data

$$y_i := f(m_i^-) \approx \dot{x}(ih). \quad (45)$$

The probabilistic state space model is thereby completely defined. Gaussian ODE filtering is equivalent to running a Gaussian filter on this probabilistic state space model.
For more details on Gaussian ODE filters, see Kersting et al. (2019) or Schober et al. (2018). An extension to more Bayesian filters—such as particle filters—is provided by Tronarp et al. (2019).

B. Equivalent Form of Filtering Distribution by GP Regression

Recall from appendix A that any Gaussian filter computes a sequence of filtering distributions

$$p(x_i \mid y_{1:i}) = \mathcal{N}(m_i, P_i) \quad (46)$$

from a GP prior on x eq. (38) and a linear Gaussian measurement model (eq. (40)) with derivative data (eq. (45)). Hence, the classical framework for GP regression with derivative observations, as introduced in Solak et al. (2003), is applicable. It *a priori* models the state x and its derivative $\dot{x}$ as a multi-task GP:

$$p\left(\begin{bmatrix} x \\ \dot{x} \end{bmatrix}\right) = \mathcal{GP}\left(\begin{bmatrix} x \\ \dot{x} \end{bmatrix}; \begin{bmatrix} \mu \\ \dot{\mu} \end{bmatrix}, \begin{bmatrix} k & k^{\partial} \\ {}^{\partial}k & {}^{\partial}k^{\partial} \end{bmatrix}\right), \quad (47)$$

with

$${}^{\partial}k = \frac{\partial k(t,t')}{\partial t}, \ k^{\partial} = \frac{\partial k(t,t')}{\partial t'}, \ {}^{\partial}k^{\partial} = \frac{\partial^2 k(t,t')}{\partial t \partial t'}. \quad (48)$$

B.1. Kernels for Derivative Observations

In this paper, we model the solution x with a integrated Brownian motion kernel k or, in other words, we model $\dot{x}$ by the Brownian Motion (a.k.a. Wiener process) kernel, i.e.

$$^{\partial}k^{\partial}(t,t') = \sigma^2_{\text{dif}} \min(t,t'), \qquad \forall t,t' \in [0,T]. \tag{49}$$

Here, $\sigma_{\text{dif}} > 0$ denotes the output variance which scales the diffusion matrix L in the equivalent SDE (eq. (38)). Integration with respect to both arguments yields the integrated Brownian motion (IBM) kernel

$$k(t,t') = \sigma^2_{\text{dif}} \left(\frac{\min^3(t,t')}{3} + |t-t'| \frac{\min^2(t,t')}{2} \right) \tag{50}$$

to model x. The once-differentiated kernels in eq. (47) are given by

$$k^{\partial}(t,t') = {^{\partial}k}(t',t) = \sigma^2_{\text{dif}} \begin{cases} t \le t' : \frac{t^2}{2}, \\ t > t' : tt' - \frac{t'^2}{2} \end{cases} . \tag{51}$$

A detailed derivation of eqs. (49) to (51) can be found in Schober et al. (2014, Supplement B).

B.2. GP Form of Filtering Distribution

Now, GP regression with prior (eq. (47)), likelihood (eq. (46)) and data $y_{1:i}$ yields an equivalent form of the filtering distribution eq. (46):

$$\begin{aligned} m_i =& \mu + k^{\partial}(h:ih, ih)^{\intercal} \left[{^{\partial}K^{\partial}}(h:ih) + R \cdot I_i \right]^{-1} \\ & \times [y_1 - \dot{\mu}(h), \dots, y_i - \dot{\mu}(ih)]^{\intercal}, \qquad (52) \\ P_i =& \begin{bmatrix} k(h,h) & \dots & k(ih,ih) \\ \vdots & \ddots & \vdots \\ k(ih,h) & \dots & k(ih,ih) \end{bmatrix} - k^{\partial}(h:ih, ih)^{\intercal} \\ & \times \left[{^{\partial}K^{\partial}}(h:ih) + R \cdot I_l \right]^{-1} k^{\partial}(h:ih, ih), \qquad (53) \end{aligned}$$

with $y_{1:i} = [y_1, \dots, y_i]^{\intercal}$, where we used the notations from eqs. (15) and (16). The derivation of eq. (18) is hence concluded by eq. (53).

B.3. Derivation of Equation (10)

In this subsection, we will use the ODE-specific notation from above instead of the generic filtering notation—e.g. $m_\theta(ih)$ instead of m_i, $f(m^-(ih))$ instead of y_i etc. To derive the missing eq. (10), we first observe that, by eq. (52), $m(ih)$ is linear in the data residuals:

$$\begin{aligned} m_\theta(ih) &= \mu + \beta_{ih} \times \qquad (54) \\ & \qquad \left[f(m^-(h)) - \dot{\mu}(h), \dots, f(m^-(ih)) - \dot{\mu}(ih) \right]^{\intercal} \\ \beta_{ih} &:= k^{\partial}(h:ih, ih)^{\intercal} \left[{^{\partial}K^{\partial}}(h:ih) + R \cdot I_i \right]^{-1}. \end{aligned}$$

Now recall that, in ODE filtering, the prior mean in eq. (47) is set to be $[\mu, \dot{\mu}] \equiv [x_0; f(x_0)]$ (or $[\mu, \dot{\mu}] \equiv [m_0; f(m_0)]$ for some estimate m_0 of x_0, in the case of unknown x_0). Consequently, application of Assumption 1 to eq. (54) yields

$$\begin{aligned} m_\theta(ih) &= x_0 + J_{ih}\theta, \quad \text{with} \qquad (55) \\ J_{ih} &:= \beta_{ih} \begin{bmatrix} f_1(m^-_\theta(h)) - f_1(x_0) & \dots & f_n(m^-_\theta(h)) - f_n(x_0) \\ \vdots & \ddots & \vdots \\ f_1(m^-_\theta(ih)) - f_1(x_0) & \dots & f_n(m^-_\theta(ih)) - f_n(x_0) \end{bmatrix} \\ &= \beta_{ih} Y_{1:i} \quad , \qquad (56) \end{aligned}$$

where $Y_{1:i}$ denotes the first i rows of Y; see eq. (17). We omit the dependence of J_{ih} on θ to obtain a linear form. Recall from Section 3 that we may w.l.o.g. assume that the time points $\{t_1, \dots, t_M\}$ lie on the filter time grid, i.e. $t_i = l_i h$ from some $l_i \in \mathbb{N}$. Therefore, eq. (55) implies

$$m_\theta(t_i) \overset{\text{eq. (14)}}{=} x_0 + \tilde{\kappa}_i Y_{1:i} \overset{\text{eq. (13)}}{=} x_0 + \kappa_i Y \tag{57}$$

for all data time points t_i, $i = 1, \dots, M$. Here, we used that $\tilde{\kappa}_i$ is equal to $\beta_{l_i h}$ by eq. (14). We conclude the derivation of eq. (10) by observing that the i-th entry of eq. (10) reads eq. (57) for all $i = 1, \dots, M$.

C. Proof of Theorem 1

Proof. We start by computing the rows of

$$D\boldsymbol{m}_\theta = [\nabla_\theta m(t_1), \dots, \nabla_\theta m(t_M)]^{\intercal}. \tag{58}$$

By eqs. (10) and (11) and the fact that the kernel prefactor K does not depend on θ, we obtain, for all $i = 1, \dots, M$, that

$$\begin{aligned} \nabla_\theta m(t_i) &= \nabla(\tilde{\kappa}(i)^{\intercal} v(\theta)) \\ &= [Dv(\theta)]^{\intercal} \tilde{\kappa}(i) + \underbrace{[D\tilde{\kappa}(i)]^{\intercal}}_{=0} v(\theta) \qquad (59) \\ &= [Dv(\theta)]^{\intercal} \tilde{\kappa}(i), \qquad (60) \end{aligned}$$

with $v(\theta) = \tilde{Y}\theta$. Here,

$$\tilde{Y} = Y[1:l_i, :] = [Y_1(\theta), \dots, Y_{l_i}(\theta)]^{\intercal} \tag{61}$$

is defined by

$$Y_j(\theta) = [y_{j1}, \dots, y_{jn}]^{\intercal} \ \in \mathbb{R}^n, \tag{62}$$

the j-th row of $Y = Y(\theta)$ (recall eq. (17)), for $j = 1, \dots, l_i$. Next, we again compute the rows of the missing Jacobian of eq. (60)

$$Dv(\theta) = [\nabla_\theta [v(\theta)]_1, \dots, \nabla_\theta [v(\theta)]_{l_i}]^{\intercal} \tag{63}$$

by the chain rule, for all $j \in \{1, \dots, l_i\}$:

$$\nabla_\theta [v(\theta)]_j = \nabla_\theta [Y_j(\theta)^{\intercal}\theta] = [DY_j(\theta)]^{\intercal} \theta + Y_j(\theta). \tag{64}$$

Again, we compute the rows of the final missing Jacobian

$$DY_j(\theta) = [\nabla_\theta y_{j1}(\theta), \dots, \nabla y_{jn}(\theta)]^{\intercal}. \tag{65}$$

The definition of y_{ij} from eq. (17) implies, in the notation of eq. (21), that

$$[\nabla_\theta y_{jk}(\theta)]_l = \lambda_{lk}(jh), \tag{66}$$

for all $l = 1, \dots, n$. Now, we can insert backwards. First, we insert eq. (66) into eq. (65) which yields

$$DY_j(\theta) = \Lambda_j, \tag{67}$$

where $\Lambda_j = \left[\lambda_{kl}(jh)\right]_{k,l=1,\dots,n}$. Second, insertion of eq. (67) into eq. (64) provides that

$$\nabla_\theta [v(\theta)]_j = \Lambda_j^\intercal \theta + Y_j(\theta). \tag{68}$$

Third, insertion of eq. (68) into eq. (63) implies that

$$Dv(\theta) = \left[\Lambda_1^\intercal \theta, \dots, \Lambda_{l_i}^\intercal \theta\right]^\intercal + Y[: l_i, :], \tag{69}$$

where

$$Y[: l_i, :] \overset{eq.\ (68)}{=} [Y_1(\theta), \dots, Y_{l_i}(\theta)]^\intercal \overset{eq.\ (62)}{=} \begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{l_i 1} & \cdots & y_{l_i n} \end{bmatrix}.$$

Fourth, we insert eq. (69) into eq. (60) and obtain

$$\begin{aligned} \nabla_\theta m(t_i) &= \left([Y[: l_i, :]]^\intercal + \left[\Lambda_1^\intercal \theta, \dots, \Lambda_{l_i}^\intercal \theta\right]\right) \tilde{\kappa}_i \\ &= [Y[: l_i, :]]^\intercal \tilde{\kappa}_i + \left[\Lambda_1^\intercal \theta, \dots, \Lambda_{l_i}^\intercal \theta\right] \tilde{\kappa}_i. \end{aligned} \tag{70}$$

By eq. (13), it follows that

$$[Y[: l_i, :]]^\intercal \tilde{\kappa}_i \overset{eq.\ (17)}{=} Y^\intercal \kappa_i, \qquad \text{and} \tag{71}$$

$$\left[\Lambda_1^\intercal \theta, \dots, \Lambda_{l_i}^\intercal \theta\right] \tilde{\kappa}_i \overset{eq.\ (20)}{=} S^\intercal \kappa_i. \tag{72}$$

This implies via eq. (70) that

$$\nabla_\theta m(t_i) = \left(Y^\intercal + S^\intercal\right) \kappa_i, \tag{73}$$

Fifth and finally, we, by insertion of eq. (73) into eq. (58) and application of eq. (12), obtain

$$Dm_\theta = K(Y + S) \overset{eq.\ (11)}{=} J + KS. \tag{74}$$

□

D. Proof of Theorem 2

We first show some preliminary technical lemmas in Appendix D.1 which are needed to prove bounds on $\|K\|$ and $\|S\|$ in Appendix D.2 and Appendix D.3, respectively. Having proved these bounds, the core proof of Theorem 2 simply consists of combining them by Theorem 1, as executed in Appendix D.4.

D.1. Preliminary lemmas

The following lemma will be needed in Appendix D.2 to bound $\|K\|$.

Lemma 3. *Let $Q > 0$ be a symmetric positive definite and $Q' \geq 0$ a symmetric positive semi-definite matrix in $\mathbb{R}^{m\times n}$. Then, it holds true that*

$$\left\| [Q + Q']^{-1} \right\|_* \leq \left\| Q^{-1} \right\|_*, \tag{75}$$

for the nuclear norm

$$\|A\|_* = \operatorname{trace} \sqrt{A^* A} = \sum_{i=1}^{m\wedge n} \sigma_i(A), \tag{76}$$

where $\sigma_i(A)$, $i \in \{1, \dots, m \wedge n\}$, are the singular values of A.

Proof. Recall that, for all symmetric positive semi-definite matrices, the singular values are the eigenvalues. Therefore

$$\begin{aligned} \left\| [Q + Q']^{-1} \right\|_* &= \sum_{i=1}^{m\wedge n} \frac{1}{\lambda_i(Q + Q')} \\ &\leq \sum_{i=1}^{m\wedge n} \frac{1}{\lambda_i(Q)} = \left\| Q^{-1} \right\|_*. \end{aligned} \tag{77}$$

In eq. (77), we exploited the fact that $Q \leq Q + Q'$ (i.e. that $(Q + Q') - Q = Q'$ is positive semi-definite) and therefore $\lambda_i(Q) \leq \lambda_i(Q + Q')$ for ordered eigenvalues $\lambda_1(Q) \leq \dots \leq \lambda_{m\wedge n}(Q)$ counted by algebraic multiplicity. This fact is an immediate consequence of Theorem 8.1.5. in Golub & Van Loan (1996). □

The next lemma will be necessary to prove a bound on $\|S\|$ in Appendix D.3.

Lemma 4. *Let $g(x, \lambda) \in C([0, T] \times \Lambda; \mathbb{R})$ on non-empty compact $\Lambda \subset \mathbb{R}^n$ with continuous first-oder partial derivatives w.r.t. the components of λ. If*

$$\sup_{\lambda \in \Lambda} g(x, \lambda) \in \mathcal{O}(h(x)) \tag{78}$$

for some constant $C > 0$ and some strictly positive $h : [0, T] \to \mathbb{R}$, then also

$$\sup_{\lambda \in \Lambda^\circ} \left| \frac{\partial}{\partial \lambda_k} g(x, \lambda) \right| \in \mathcal{O}(h(x)), \tag{79}$$

where Λ° denotes the interior of Λ.

Proof. Assume not. Then, there is a $k \in \{1, \dots, n\}$ and a $\tilde{\lambda} \in \Lambda^\circ$ such that

$$\left| \frac{\partial}{\partial \lambda_k} g(x, \tilde{\lambda}) \right| \notin \mathcal{O}(h(x)). \tag{80}$$

Since, for all $x \in [0,T]$, $\frac{\partial}{\partial \lambda_k}(x,\cdot)$ is uniformly continuous over the bounded domain Λ°, there is a $\delta > 0$ such that

$$\left|\frac{\partial}{\partial \lambda_k} g(x,\tilde{\lambda})\right| \notin \mathcal{O}(h(x)), \quad \text{for all } \lambda \in B_{2\delta}(\tilde{\lambda}). \tag{81}$$

Let us w.l.o.g. (otherwise consider $-g$) assume that

$$\frac{\partial}{\partial \lambda_k} g(x,\tilde{\lambda}) \geq 0, \quad \text{for all } \lambda \in B_{2\delta}(\tilde{\lambda}). \tag{82}$$

Now, on the one hand, we know by the fundamental theorem of calculus that

$$\begin{aligned}\int_{-\delta}^{0} &\frac{\partial}{\partial \lambda_k} g(x_n, \tilde{\lambda} + \tilde{\delta} e_k)\,\mathrm{d}\tilde{\delta} \\ &= \underbrace{g(x,\tilde{\lambda})}_{\in \mathcal{O}(h(x))} - \underbrace{g(x,\tilde{\lambda} - \delta e_k)}_{\in \mathcal{O}(h(x))} \in \mathcal{O}(h(x)).\end{aligned} \tag{83}$$

However, on the other hand, we know from our assumption that

$$0 \overset{\text{eq. (82)}}{\leq} \int_{-\delta}^{0} \frac{\partial}{\partial \lambda_k} g(x_n, \tilde{\lambda} + \tilde{\delta} e_k)\,\mathrm{d}\tilde{\delta} \tag{84}$$

$$\leq \int_{-\delta}^{0} \underbrace{\left|\frac{\partial}{\partial \lambda_k} g(x_n, \tilde{\lambda} + \tilde{\delta} e_k)\right|}_{\notin \mathcal{O}(h(x)),\ \text{by eq. (81)}} \mathrm{d}\tilde{\delta} \notin \mathcal{O}(h(x)), \tag{85}$$

which implies

$$\int_{-\delta}^{0} \frac{\partial}{\partial \lambda_k} g(x_n, \tilde{\lambda} + \tilde{\delta} e_k)\,\mathrm{d}\tilde{\delta} \notin \mathcal{O}(h(x)). \tag{86}$$

The desired contradiction is now found between eqs. (83) and (86). □

D.2. Bound on $\|K\|$

Lemma 5. *Under Assumption 3 and for all $R > 0$, it holds true that*

$$\|K\| \leq C(T), \tag{87}$$

where $C(T) > 0$ is a constant that depends on T.

Proof. First, recall eqs. (12) to (16) and observe that

$$\left\|k^{\partial}(h : t_i, t_i)\right\| \leq C\frac{\sigma^2}{2}\left\|\left[h^2, \ldots, T^2\right]\right\|_{\infty} = C\left(2^{-\frac{1}{2}}\sigma T\right)^2,$$

for all $i = 1, \ldots, M$. Second, Lemma 3 implies that

$$\begin{aligned}\left\|\left[{}^{\partial}K^{\partial}(h : t_i) + R \cdot I_{l_i}\right]^{-1}\right\| &\overset{\text{eq. (75)}}{\leq} C\left\|R^{-1} \cdot I_{l_i - 1}\right\|_{*} \\ &\leq C\left\|R^{-1} \cdot I_{\bar{N}-1}\right\|_{*} \leq CR\bar{N}.\end{aligned}$$

Now, by eq. (13), we observe

$$\begin{aligned}\|\kappa_i\|_1 &= \|\tilde{\kappa}_i\|_1 \\ &\leq \left\|\left[{}^{\partial}K^{\partial}(h : t_i) + R \cdot I_{l_i}\right]^{-1}\right\| \cdot \left\|k^{\partial}(h : t_i, t_i)\right\| \\ &\leq C(T),\end{aligned} \tag{88}$$

where we inserted the above inequalities in the last step. Finally, we obtain eq. (87) by plugging eq. (88) into

$$\|K\| \leq C\|K\|_{\infty} \overset{\text{eq. (12)}}{=} \max_{1 \leq i \leq M} \|\kappa_i\|_1. \tag{89}$$

□

D.3. Bound on $\|S\|$

Before estimating $\|S\|$, we need to bound how far the entries of S (recall eq. (20)) deviate from the true sensitivities $\frac{\partial}{\partial \theta_k} x_\theta(T)$.

Lemma 6. *If $\Theta \subset \mathbb{R}^n$ is compact, then it holds true, under Assumptions 1 and 2, that*

$$\sup_{\theta \in \Theta^{\circ}} \left\|\frac{\partial}{\partial \theta_k} m_\theta^{-}(T) - \frac{\partial}{\partial \theta_k} x_\theta(T)\right\| \in \mathcal{O}(h). \tag{90}$$

Proof. First, recall that the convergence rates of $\mathcal{O}(h)$ provided by Theorem 6.7 in Kersting et al. (2019) only depend on f through the dependence of the constant $K(T) > 0$ on the Lipschitz constant L of f. But this L is independent of θ by Assumption 1. Hence, Theorem 6.7 from Kersting et al. (2019) yields under Assumption 2 that

$$\sup_{\theta \in \Theta^{\circ}} m_\theta^{-}(T) - x_\theta(T) \in \mathcal{O}(h). \tag{91}$$

Moreover, Theorem 8.49 in Kelley & Peterson (2010) is applicable under Assumption 1 and implies that $x_\theta(t)$ is continuous and has continuous first-order partial derivatives with respect to of θ_k. By construction—recall eq. (10)—the filtering mean $m_\theta(t)$ has the same regularity too. Hence, application of Lemma 4 with $x = h$, $\Lambda = \Theta$, $\lambda = \theta$, $g(x,\lambda) = m_\theta^{-}(T) - x_\theta(T)$ is possible, which yields eq. (90) from eq. (91). □

Lemma 7. *If $\Theta \subset \mathbb{R}^n$ is compact, then it holds true, under Assumptions 1 to 3, that*

$$\|S\| \leq C\left(\|\nabla_\theta x_\theta\| + h\right), \tag{92}$$

for sufficiently small $h > 0$.

Proof. By Assumption 3 and the equivalence of all matrix norms, we observe

$$\|S\| \leq C\|S\|_2 = C\|S^\intercal\|_2 \leq C\|S^\intercal\|_{2,1} \tag{93}$$
$$\overset{\text{eq. (20)}}{=} C\sum_{j=1}^{\bar{N}} \left\|\Lambda_j^\intercal \theta\right\|_2 \tag{94}$$
$$\leq C\sum_{j=1}^{\bar{N}} \|\Lambda_j^\intercal\|_2 \underbrace{\|\theta\|_2}_{\leq C,\text{ since } \Theta \text{ bounded}}, \tag{95}$$

where $\|\cdot\|_{2,1}$ denotes the $L_{2,1}$ norm. We conclude, using Assumption 2 and Lemma 6, that

$$\|\Lambda_j^\intercal\|_2 \overset{\text{eq. (21)}}{\leq} L \max_{jk}\left[\frac{\partial}{\partial\theta_k} m_\theta^-(jh)\right] \tag{96}$$
$$\overset{\text{eq. (90)}}{\leq} C\left(\|\nabla_\theta x_\theta\| + h\right). \tag{97}$$

□

D.4. Proof of Theorem 2

Proof. By Theorem 1 and the sub-multiplicativity of the induced p-norm $\|\cdot\|_p$, we observe that

$$\|J - \boldsymbol{D}\boldsymbol{m}_\theta\| = \|KS\| \leq C\|KS\|_p \leq \|K\|_p\|S\|_q$$
$$\leq C\|K\|\|S\|, \tag{98}$$

for some $p, q \geq 1$. Application of Lemmas 5 and 7 concludes the proof. □

E. Gradient and Hessian Estimators for the Bayesian Case

In the main paper, we only consider the maximum likelihood objective; see eq. (23). Nonetheless, the extension to the Bayesian objective, with a prior $\pi(\theta)$, is straightforward:

$$-\log\left(p(\boldsymbol{z} \mid \theta)\pi(\theta)\right) = -\log\left(p(\boldsymbol{z} \mid \theta)\right) - \log\left(\pi(\theta)\right)$$

Accordingly, the gradients and Hessian of this objective are

$$\nabla_\theta\left[-\log\left(p(\boldsymbol{z} \mid \theta)\pi(\theta)\right)\right] \overset{\text{eq. (26)}}{=} \hat{\nabla}_\theta E(\boldsymbol{z}) - \nabla_\theta \log\left(\pi(\theta)\right),$$
$$\nabla_\theta^2\left[-\log\left(p(\boldsymbol{z} \mid \theta)\pi(\theta)\right)\right] \overset{\text{eq. (27)}}{=} \hat{\nabla}_\theta^2 E(\boldsymbol{z}) - \nabla_\theta^2 \log\left(\pi(\theta)\right).$$

Hence, for a Gaussian prior $\pi(\theta) = \mathcal{N}(\theta; \mu_\theta, V_\theta)$, the Bayesian version of the gradients and Hessian estimators in eqs. (26) and (27) are hence given by

$$\hat{\nabla}_\theta E(\boldsymbol{z})_{\text{Bayes}} := -J^\intercal\left[\boldsymbol{P} + \sigma^2 I_M\right]^{-1}\left[\boldsymbol{z} - \boldsymbol{m}_\theta\right]$$
$$- V_\theta^{-1}\left[\theta - \mu_\theta\right], \quad \text{and} \tag{99}$$
$$\hat{\nabla}_\theta^2 E(\boldsymbol{z})_{\text{Bayes}} := J^\intercal\left[\boldsymbol{P} + \sigma^2 I_M\right]^{-1} J + V_\theta^{-1}. \tag{100}$$

F. Glucose Uptake in Yeast

The Glucose uptake in yeast (GUiY) is described by mass-action kinetics. In the notation of Schillings et al. (2015), the underlying ODE is given by:

$$\begin{aligned}
\dot{x}^e_{\text{Glc}} &= -k_1 x^e_E x^e_{\text{Glc}} + k_{-1} x^e_{\text{E–Glc}} \\
\dot{x}^i_{\text{Glc}} &= -k_2 x^i_E x^i_{\text{Glc}} + k_{-2} x^i_{\text{E–Glc}} \\
\dot{x}^i_{\text{E–G6P}} &= k_4 x^i_E x^i_{\text{G6P}} + k_{-4} x^i_{\text{E–G6P}} \\
\dot{x}^i_{\text{E–Glc–G6P}} &= k_3 x^i_{\text{E–Glc}} x^i_{\text{G6P}} - k_{-3} x^i_{\text{E–Glc–G6P}} \\
\dot{x}^i_{\text{G6P}} &= -k_3 x^i_{\text{E–Glc}} x^i_{\text{G6P}} + k_{-3} x^i_{\text{E–Glc–G6P}} \\
&\quad - k_4 x^i_E x^i_{\text{G6P}} + k_{-4} x^i_{\text{E–Glc}} \\
\dot{x}^e_{\text{E–Glc}} &= \alpha\left(x^i_{\text{E–Glc}} - \dot{x}^e_{\text{E–Glc}}\right) + k_1 x^e_E x^e_{\text{Glc}} \\
&\quad - k_{-1} x^e_{\text{E–Glc}} \\
\dot{x}^i_{\text{E–Glc}} &= \alpha\left(x^e_{\text{E–Glc}} - \dot{x}^i_{\text{E–Glc}}\right) - k_3 x^i_{\text{E–Glc}} x^i_{\text{G6P}} \\
&\quad + k_{-3} x^i_{\text{E–Glc–G6P}} + k_2 x^i_{\text{E}} x^i_{\text{Glc}} - k_{-2} x^i_{\text{E–Glc}} \\
\dot{x}^e_{\text{E}} &= \beta\left(x^i_{\text{E}} - x^e_{\text{E}}\right) - k_1 x^e_{\text{E}} x^e_{\text{Glc}} + k_{-1} x^e_{\text{E–Glc}} \\
\dot{x}^i_{\text{E}} &= \beta\left(x^e_{\text{E}} - x^i_{\text{E}}\right) - k_4 x^i_{\text{E}} x^i_{\text{G6P}} + k_{-4} x^i_{\text{E–G6P}} \\
&\quad - k_2 x^i_{\text{E}} x^i_{\text{Glc}} + k_{-2} x^i_{\text{E–Glc}},
\end{aligned}$$

where k_1, k_{-1}, k_2, k_{-2}, k_3, k_{-3}, k_4, k_{-4}, α, and β are the 10 parameters. Note that this system satisfies Assumption 1. Following Schillings et al. (2015) and Gorbach et al. (2017), we used this ODE with initial value $x_0 = \mathbb{1}_M$, time interval $[0., 100.]$ and true parameter $\theta^* = [0.1, 0.0, 0.4, 0.0, 0.3, 0.0, 0.7, 0.0, 0.1, 0.2]$. To generate data by eq. (3), we added Gaussian noise with variance $\sigma^2 = 10^{-5}$ to the corresponding solution at time points $[1., 2., 4., 5., 7., 10., 15., 20., 30., 40., 50., 60., 80., 100.]$. The optimizers and samplers were initialized at $\theta = 1.2 \cdot \theta^* = [0.12, 0, 0.48, 0, 0.36, 0, 0.84, 0, 0.12, 0.24]$, and the forward solutions for all likelihood evaluations were computed with step size $h = 1.0$.