Uncertainty-Aware Numerical Solutions of ODEs by Bayesian Filtering

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

---

vorgelegt von

Hans Philipp Kersting

aus Frankfurt am Main

Eberhard Karls Universität Tübingen

September 2020

Tag der mündlichen Qualifikation:	17.12.2020
Dekan:	Prof. Dr. József Fortágh
1. Berichterstatter:	Prof. Dr. Philipp Hennig
2. Berichterstatter:	Jun. Prof. Dr. Anna Levina

Abstract

Numerical analysis is the branch of mathematics that studies algorithms that compute approximations of well-defined, but analytically-unknown mathematical quantities. Statistical inference, on the other hand, studies which judgments can be made on unknown parameters in a statistical model. By interpreting the unknown quantity of interest as a parameter and providing a statistical model that relates it to the available numerical information (the 'data'), we can thus recast any problem of numerical approximation as statistical inference. In this way, the field of *probabilistic numerics* introduces new 'uncertainty-aware' numerical algorithms that capture all relevant sources of uncertainty (including all numerical approximation errors) by probability distributions.

While such recasts have been a decades-long success story for global optimization and quadrature (under the names of Bayesian optimization and Bayesian quadrature), the equally important numerical task of solving ordinary differential equations (ODEs) has been, until recently, largely ignored. With this dissertation, we aim to further shed light on this area of previous ignorance in three ways: Firstly, we present a first rigorous Bayesian model for initial value problems (IVPs) as statistical inference, namely as a stochastic filtering problem, which unlocks the employment of all Bayesian filters (and smoothers) to IVPs. Secondly, we theoretically analyze the properties of these new *ODE filters*, with a special emphasis on the convergence rates of Gaussian (Kalman) ODE filters with integrated Brownian motion prior, and explore their potential for (active) uncertainty quantification. And, thirdly, we demonstrate how employing these ODE filters as a forward simulator engenders new ODE inverse problem solvers that outperform classical 'uncertainty-unaware' ('likelihood-free') approaches.

This core content is presented in Chapter 2. It is preceded by a concise introduction in Chapter 1 which conveys the necessary concepts and locates our work in the research environment of probabilistic numerics. The final Chapter 3 concludes with an in-depth discussion of our results and their implications.

Kurzfassung

Die numerische Analysis ist der Zweig der Mathematik, der sich mit Algorithmen beschäftigt, welche Approximationen von wohldefinierten, aber analytisch unbekannten mathematischen Größen berechnet. Die inferentielle Statistik studiert hingegen, welche Aussagen über die unbekannten Parameter von statistischen Modellen getroffen werden können. Indem wir die unbekannte Lösung eines numerischen Problems als einen solchen Parameter interpretieren und indem wir ein statistisches Modell konstruieren (das diesen Parameter in einen Zusammenhang mit den verfügbaren Informationen stellt), können wir jedes numerische Problem als statistische Inferenz reinterpretieren. Auf diese Art führt das Gebiet *Probabilistische Numerik* neue 'unsicherheitsbewusste' numerische Methoden ein, welche alle relevanten Quellen von Unsicherheit (inklusive der numerischen Unsicherheit) berücksichtigen.

Während solche Umformulierungen bereits seit Jahrzehnten für globale Optimierung und Quadratur (unter den Namen Bayesian Optimization und Bayesian Quadrature) eine Erfolgsgeschichte sind, wurde die ebenso wichtige Lösung von Ordinary Differential Equations (ODEs) bis vor kurzem weitgehend ignoriert. Mit dieser Dissertation wollen wir diese Wissenslücke mit drei Beiträgen füllen: Erstens präsentieren wir ein erstes rigoroses bayesianisches Modell für Anfangswertprobleme (AWPe) als statistische Inferenz - nämlich als stochastisches Filterproblem. Dies ermöglicht die Anwendung aller bayesianischer Filter (und Glätter) für AWPe. Zweitens analysieren wir die Eigenschaften von diesen neuen *ODE Filtern* - mit einer besonderen Betonung auf die Konvergenzraten von Gauß (Kalman) ODE Filtern mit einer integrierten Brownschen Bewegung als Prior - und untersuchen ihr Potential für (aktive) Unsicherheitsquantifizierung. Drittens demonstrieren wir, wie die Anwendung dieser ODE Filter als Vorwärtslöser neue Algorithmen für Inversprobleme ergibt, welche die Sampleeffizienz und Geschwindigkeit von klassischen 'unsicherheitsunbewussten' ('likelihoodfreien') Algorithmen übertreffen.

Dieser Kerninhalt wird in Kapitel 2 präsentiert. Eine konzise Einleitung ist in Kapitel 1 zu finden, welche die nötigen Konzepte vermittelt und unsere Forschung in den Kontext der probabilistischen Numerik einordnet. Das finale Kapitel 3 schließt mit einer Diskussion unserer Ergebnisse und derer Implikationen.

Acknowledgments

First and foremost, I thank Philipp Hennig who has been a great teacher to me. His guidance and example has tremendously helped me through the inevitable challenges a PhD poses. I thank Anna Levina for her valuable time examining this thesis. I thank all of my colleagues in the Method of Machine Learning group at the University of Tübingen, with particular mention to Michael Tiemann, Filip Tronarp, Nicholas Krämer, Edgar Klenske, Maren Mahsereci, Simon Bartels, Lukas Balles, Alexandra Gessner, Emilia Magnani, Motonobu Kanagawa, Filip De Roos, Frank Schneider, Matthias Werner, Felix Dangel, Susanne Zabel, Agustinus Kristiadi, Jonathan Wenger, Thomas Glässle, Katharina Ott, Nathanael Bosch, Lukas Tatzel, Julia Grosse, Marius Hobbhahn and Franziska Weiler. I thank my (additional) collaborators Tim J. Sullivan, Simo Särkkä, Martin Schiegg, and Christian Daniel. I thank my peers Jon Cockayne, Giacomo Garegnani, Toni Karvonen, Onur Teymur, FX Briol, and others„ for the good times and interesting discussions at conferences and meetings. I thank Martin Schiegg, Christian Daniel, and Michael Tiemann for my great time at the Bosch Center for Artificial Intelligence. I also thank Maren Mahsereci, Javier Gonzalez, and Neil Lawrence for my exciting internship at Amazon Research Cambridge. I thank my professors from my studies (in particular Vitali Wachtel, Peter Müller, Konstantinos Panagiotou and Frank Proske) for teaching me. I thank my family Götz, Inge and Max Kersting for everything they have helped me with. Lastly, I thank Dana Babin for her constant support throughout my thesis, in particular in challenging times.

Contents

Included Publications

This dissertation is based on the following published papers:

1. Hans Kersting, Philipp Hennig "Active Uncertainty Calibration in Bayesian ODE Solvers" *Uncertainty in Artificial Intelligence (UAI)*, 2016

2. Filip Tronarp, Hans Kersting, Simo Särkkä, Philipp Hennig "Probabilistic Solutions to Ordinary Differential Equations as Nonlinear Bayesian Filtering: A New Perspective" *Stat. Comput.*, **29**(6), 1297–1315

3. Hans Kersting, Tim J. Sullivan, Philipp Hennig "Convergence Rates of Gaussian ODE Filters" *Stat. Comput.*, **30**(6), 1791–1816

4. Hans Kersting[*1], Nicholas Krämer[*], Martin Schiegg, Christian Daniel, Michael Tiemann, Philipp Hennig "Differentiable Likelihoods for Fast Inversion of 'Likelihood-Free' Dynamical Systems", *International Conference for Machine Learning (ICML)*, 2020

5. Hans Kersting, Maren Mahsereci "A Fourier State Space Model for Bayesian ODE Filters", *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2020

The publications are contained, in full, in the Appendix.

[1] $*$ indicates primary authorship.

Preface

We shall not cease from exploration
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.

—T. S. Eliot (Four Quartets, “Little Gidding”)

Imagine, dear reader, a group of scientists at a coffee break of a conference for computational mathematics. Everybody is relaxed and, perhaps, tired from a jetlag, waiting for the coffee to kick in. One of the scientist (probably a younger one) is colloquially saying to her colleagues: "The true solution of a numerical problem is unknown, otherwise no approximation (and hence no numerical method) would be needed." Her colleagues sleepily nod along to this boring truism, but their drowsiness disappears as she goes on: "Statistics is, amongst other things, the science of inferring unknown parameters. Hence, both numerics and statics are concerned with estimating unknown quantities." As her colleagues begin to listen with interests (some of them have never thought of this clear parallel), they become more tense as she continues: "Hence, if we can provide a statistical model in which the true solution is a parameter, we can employ methods from statistical inference to compute a (posterior) probability distribution that captures our incomplete knowledge of the true solution, including the uncertainty. In many cases, classical numerical approximations can then be viewed as point estimates that are contained in this probability distribution as a mean or mode." If she (notwithstanding the increasing tension of some of her colleagues) would have the audacity to assert "Numerical analysis is but a subset of statistical inference which prioritizes worst-case bounds!", passionate push-back (at least from the senior numerical analysts in the room) would be guaranteed.

This fictional scenario is an archetype for the many (often tense) discussions my probabilistic-numerics colleagues and I have experienced with other scientists. While virtually everybody would agree with the first sentence of the above fictional quote, many very capable scientists would not follow along to the statement that statistical inference can be used for numerical problems. Even more would disagree with the claim that numerical analysis is a subset of statistical inference. Since the argumentative steps seem watertight to me, such conversations never failed to fascinate and bewilder me. In particular with very senior scientists, it made me feel a little bit like the prisoner in the

unexpected hanging paradox (Chow, 1998) who concludes that what happens to him (his hanging) is logically impossible.

Of course, people will explain why, in their view, the above arguments are wrong or misguided. To begin with, they might have the following fundamental objections (we omit some less-convincing misconceptions here): They might assert that it is, generally, absurd to use statistics for deterministic objects, and that the only true statistical model would be a Dirac measure on the true solution which, however, is unknown. Or they might claim that, since there is (a priori) no true statistical model, any model will bias the final approximation. Or they might posit that the information (read: function evaluations) used by numerical methods cannot be treated as data because "data has to be collected or measured", or point out that numerics might not be a strict subset of statistics since the solution of finite numerical problems is known if enough compute is available.

While these arguments are great starts for further thought, I believe that they ultimately do not hold up to scrutiny. This is not the right place to give a detailed refutation; but many of the relevant arguments can be found in Hennig *et al.* (2015), Oates and Sullivan (2019), and (to some extent) also in Ritter (2000).

The oppositional scientists might also point out different tendencies and emphases in statistics and numerics: They might argue that, unlike statistics, numerics tends to focus on well-posed problems, and that therefore uncertainty quantification (UQ) is not as central to numerics. They might go on to posit that, due to relying on an oftentimes-slow data collection process, computational cost is less important in statistics, leading to costly methods, and that noise in the data makes UQ in statistics more essential than in numerics. They might say that statistics uses reproducing kernel Hilbert spaces (RKHS) to classify problems, while numerics settles for more elementary function spaces. They might point out that the statistical model is an *additional* modeling assumption *on top of the inevitable function space*, and that the quality of the posterior UQ will depend on this choice (leading to complex trade-offs between computational cost and statistical expressiveness of the prior). Or, they might point out that numerical algorithms are optimized for the worst case, while statistics has the average case in mind.

These arguments, however, are not a rejection (and maybe are in fact an implicit acceptance?) of the basic arguments put forward above. They contain very important distinctions which we try to comprehend and take up in our research. In fact, all of the research presented in this thesis are informed and fueled by such considerations.

Lastly, and least productively, the opposing scientists might enumerate what probabilistic numerics (PN) has not achieved yet—enumerating algorithms and bounds for specific numerical problems. But Rome wasn't built in one day, and there is really no need to respond to this charge.

Maybe you can, dear reader, imagine how mystified these arguments have left me, who has (over the years of my PhD) become convinced that PN unifies statistics and numerics and offers (what I call) *the three promises of PN*:

1. more flexible *classification* of problems by statistical model selection,
2. comprehensive *uncertainty quantification*, and
3. *invention* of new average-case-optimal algorithms,

all of which are covered (for the example of ODEs) in this thesis.

In some communities, the jury is still out on which side of the argument prevails, but I, for one, am sure that we will eventually be acquitted of our iconoclasm of numerical analysis. This dissertation is thus written in the spirit of an enthusiastic 'Yes' to PN. If it makes the logic and the three promises of PN more visible to you, dear reader, its goal will be met.

— Hans Kersting, Tübingen, September 2020

How to read this thesis: Chapter 1 contains a concise introduction to today's research environment of probabilistic numerics for ODEs—within the larger context of modern numerical analysis and machine learning. We advise any reader unfamiliar with PN to read it. No knowledge of our published papers, which are attached in the Appendix, is required for the introduction. The summary (Chapter 2) and discussion (Chapter 3) are based on all publications. Therefore, it might be necessary for the reader to familiarize themself with the Appendix, to understand these chapters in full detail.

1 Introduction

Ordinary differential equations (ODEs) are important all over science and engineering. Originally introduced by Newton (1671), the vast list of their applications today includes: trajectories of objects in classical mechanics, action potentials in brains, radioactive decay in nuclear power plants, continuous limits of deep neural networks, shortest paths on Riemannian manifolds, predator-prey dynamics in ecosystems and the spread of pandemics. Formally, the trajectory of such a system $x : [0, T] \to \mathbb{R}^d$ is described by an ODE, written

$$\dot{x}(t) = f\left(x(t)\right), \qquad \forall t \in [0, T], \tag{1.1}$$

for some final time $T > 0$, with a vector field $f : \mathbb{R}^d \to \mathbb{R}^d$ capturing the mechanics of the system.[1] Under the assumptions of the (global) Picard–Lindelöf theorem (Kelley and Peterson, 2010, Corollary 8.35), the solution x is well-defined given any fixed initial value $x(0) \in \mathbb{R}^d$. An ODE, eq. (1.1), together with an initial condition $x(0) = a \in \mathbb{R}^d$ is therefore a well-posed problem called *initial value problem (IVP)*. The solution is often summarized in a so-called *flow map* $\Phi_t(a) = x(t)$, where $x(t)$ is the solution of eq. (1.1) with $x(0) = a$. In this dissertation, we restrict our attention to IVPs. Note, however, that our results can be applied to boundary value problems (BVPs), where a final condition $x(T) = b \in \mathbb{R}^d$ is added, via shooting methods (Press *et al.*, 2007, Section 18.1).

1.1 The paradigm of classical numerics

Almost all of numerical analysis ('numerics') is built on, what we propose to call, a *worst-case uncertainty-unaware paradigm.*

The classical paradigm is **worst-case** in precisely the sense of Ritter (2000) which we are going to concisely exemplify for ODEs next. To this end, we consider explicit RK methods as an example (but analogous arguments hold for all other methods). As presented in more detail in Hairer *et al.* (1987, Chapter II), RK methods are designed to have the fastest-shrinking convergence rates, given a certain amount of $s \in \mathbb{N}$ evaluations of f per step. For an integration step $0 \to h$ of size $h > 0$, they compute a numerical estimate $\hat{x}(h)$ as follows: First, for $i \in \{1, \ldots, s\}$, function evaluations are collected

[1] We only consider, without loss of generality, the autonomous case $f(x(t))$ but all claims below apply to the more general case $f(x(t), t)$.

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \dots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \dots & b_{s-1} & b_s \end{array}$$

Figure 1.1: Butcher tableau for explicit Runge–Kutta methods, see eqs. (1.2) and (1.3).

according to

$$y_i = f\left(x_0 + h\sum_{j=1}^{i-1} a_{ij}y_j\right), \tag{1.2}$$

which are then used to linearly predict forward in time:

$$\hat{x}(h) = x_0 + h\sum_{i=1}^{s} b_i y_i. \tag{1.3}$$

The coefficients $a_{21}, a_{31}, a_{32}, \dots, a_{s1}, a_{s2}, \dots, a_{s,s-1}, b_1, \dots, b_s, c_2, \dots, c_s \in \mathbb{R}$ are usually summarized in a *Butcher tableau*; see Figure 1.1.

But how are these coefficients chosen? To see this, let us first recursively define $f^{(i)}\colon \mathbb{R}^d \to \mathbb{R}^d$ by $f^{(0)}(a) \coloneqq a$, $f^{(1)}(a) \coloneqq f(a)$ and $f^{(i)}(a) \coloneqq [\nabla_x f^{(i-1)} \cdot f](a)$. Now, differentiating the ODE, eq. (1.1), $(i-1)$-times by the chain rule yields

$$x^{(i)}(t) = f^{(i-1)}(t)\left(x^{(0)}(t)\right), \tag{1.4}$$

as proved in Appendix C.15. Hence, the Taylor expansion of x around some time $t \in [0, T]$ is given by

$$x(t+h) = \sum_{i=0}^{\infty} \frac{h^i}{i!} f^{(i)}\left(x^{(0)}(t)\right). \tag{1.5}$$

The coefficients in the Butcher Tableau, Figure 1.1, are chosen such that $\hat{x}(h)$, matches as many summands of the Taylor expansion, eq. (1.5), with $t = 0$, as possible. In other words, they are chosen such that the maximal numerical error $\|x(h) - \hat{x}(h)\|$ is in $\mathcal{O}(h^p)$ (by Taylor's theorem, if applicable), with $p \in \mathbb{N}$ as high as possible. How p grows with s is depicted in Figure 1.2. RK methods are therefore designed to have the (asymptotically) smallest error for the worst-case f with bounded p^{th} order partial derivatives (so that Taylor's theorem is applicable).

p	1	2	3	4	5	6	7	8
$\min s$	1	2	3	4	6	7	9	11

Figure 1.2: The minimal amount of stages s for $\mathcal{O}(h^p)$ convergence in explicit RK methods; see Butcher (2008, Section 32).

To see why we also call the classical paradigm **uncertainty-unaware**, recall that Hermite interpolation (Spitzbart, 1960) fits a polynomial of least-possible order using both observations of the function and its (higher) derivative. Hence, RK methods of order p can also be thought of (iteratively) performing Hermite interpolation, with

$$\left\{\left(t, x^{(i)}(t) \overset{!}{=} f^{(i)}(\hat{x}(t))\right);\ i = 0, \ldots, p\right\} \tag{1.6}$$

as data, in every step $t \to t+h$. However, except for the first integration step $0 \to h$, the starting value $\hat{x}(t)$ is only an estimate of the true $x(t)$. Accordingly, the data of eq. (1.6) is inexact or 'noisy' as the $\overset{!}{=}$ is only a true equality if $\hat{x}(t) = x(t)$. In reality, however, the numerical error $\|\hat{x}(t) - x(t)\|$ can accumulate quickly; see e.g. Figure 7.1. in Hairer *et al.* (1987). Hence, for $t > 0$, RK methods falsely make the implicit assumption that $f^{(i)}(\hat{x}(t)) = x^{(i)}(t)$, i.e. that they extrapolate with **exact data**. In other words, they are unaware of their numerical uncertainty. Figure 1.3 shows that this effect matters by comparing *the* Runge–Kutta method (RK4) with the corresponding 4^{th} order Hermite polynomial (Hermite4) with exact data, i.e. with

$$\left\{\left(t, x^{(i)}(t) \overset{!}{=} f^{(i)}(x(t))\right);\ i = 0, \ldots, 4\right\} \tag{1.7}$$

instead of eq. (1.6). This is to say that Hermite4 receives the data that RK4 assumes to receives and does not suffer from the numerical uncertainty $\hat{x}(t) \neq x(t)$ that RK4 ignores. It does not exhibit uncertainty-unawareness because it has access to certain data. Unsurprisingly, the Hermite4 method outperforms RK4 in all cases of Figure 1.3 which shows that the underlying assumptions of classical numerics are too optimistic. As $x(t)$ and hence $f^{(i)}(x(t))$ are unknown, we cannot remove the uncertainty (like Hermite4 does), but we can model it and build more robust 'uncertainty-aware' solvers. In fact, the first paper of this thesis (Kersting and Hennig, 2016) in Appendix A introduces such a solver.

1.2 The paradigm of probabilistic numerics

As an alternative, probabilistic numerics (PN) proposes an *average-case uncertainty-aware paradigm.*

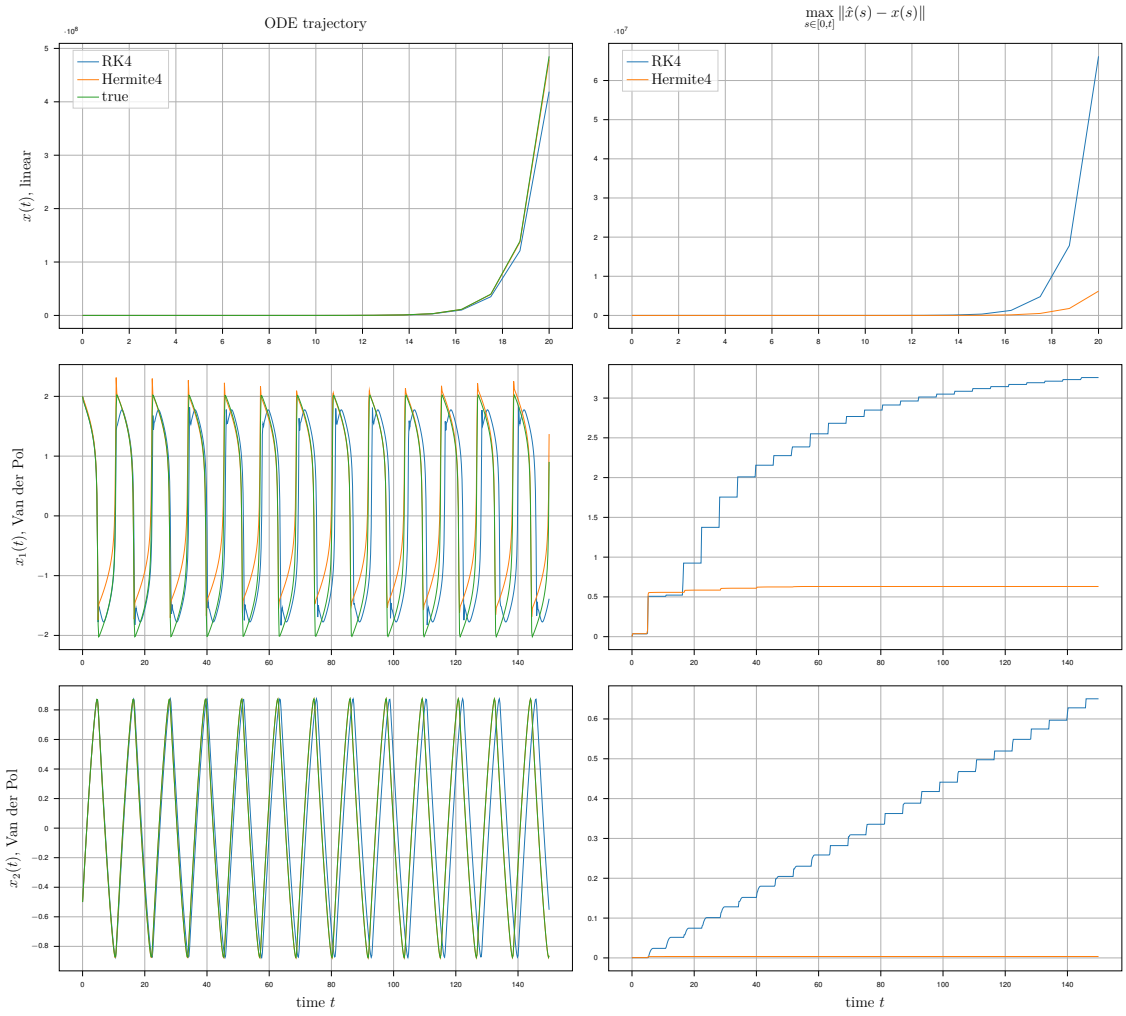


Figure 1.3: This figure shows how RK4 and Hermite4 perform on the linear and Van der Pol ODE. The linear system is given by $\dot{x}(t) = x(t)$, $x(0) = x_0$ and the Van der Pol system by $(x_1(t), x_2(t)) = (\mu(x_1(t) - \frac{1}{3}x_1(t)^3 - x_2(t), \frac{x_1(t)}{\mu})$, $x(0) = (2, -5)$, $\mu = 5$. The left column shows the trajectories. The right column shows the maximal error up to time t.

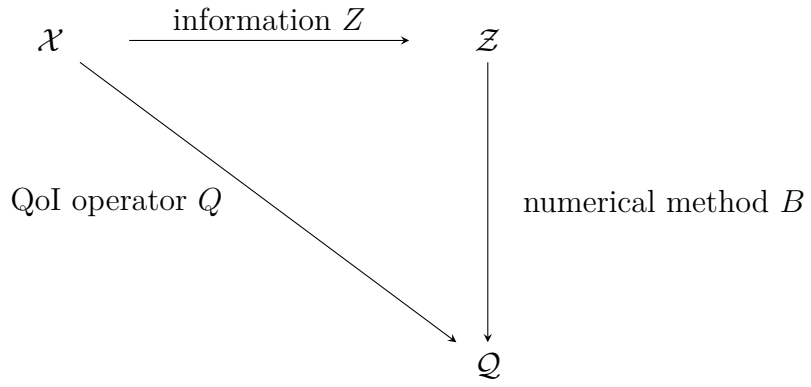


Figure 1.4: A conceptual diagram of numerical methods. Details in text.

This approach is best understood in a Bayesian framework although it can also be realized in a non-Bayesian way, e.g. by sampling-based ODE solvers as discussed in Appendix C.1.2.

The idea of *Bayesian numerical analysis*, as first specified by Diaconis (1988), was recently derived from measure-theoretic principles by Cockayne *et al.* (2019) in a rigorous way. We here give an intuitive high-level summary.

Let us first, with the help of Figure 1.4, think conceptually about what a numerical method is: Assume that our numerical task is to infer some quantity of interest (QoI) in $\mathcal{Q}$, e.g. the final point $x(T) \in \mathbb{R}^d$ of some ODE. This QoI is a feature of some unknown function $x \in \mathcal{X}$, e.g. the entire solution $x : [0, T] \to \mathbb{R}^d$. The QoI can be extracted from x by applying the information operator $Q : \mathcal{X} \to \mathcal{Q}$ to x, which is the projection $Q(x) \coloneqq x(T)$ for ODEs. We can collect information on x in the information space $\mathcal{Z}$ via the information operator Z, e.g. information on $\dot{x}(t) \in \mathbb{R}^d$ via evaluations of $f(x(t)) \in \mathbb{R}^d$ in the spirit of eq. (1.6). Now, a numerical method B is just an algorithm that receives information in $\mathcal{Z}$ as an input and constructs an approximation of the QoI in $\mathcal{Q}$ as an output. It usually does this by mapping the information back through Z to construct an approximation $\hat{x}$ of x and, then approximate the QoI by $Q(\hat{x})$; cf. above-mentioned Runge–Kutta methods.

A (Bayesian) probabilistic numerical method (PNM) performs these steps in the following principled way. It explicitly starts with a *prior* $p(x)$ over $\mathcal{X}$ and then (actively) collects information $z \in \mathcal{Z}$ according to some policy depending on $p(x)$ and Z, e.g. it queries Z at some points in x (for instance the mean of $p(x)$). Given the information z, the probability of x given z is captured by some *likelihood* $p(x \mid z)$. Application of *Bayes' rule*

$$p(x \mid z) \propto p(z \mid x)p(x) \tag{1.8}$$

now yields a *posterior* $p(x \mid z)$. The final output of the PNM is then the pushforward of $p(x \mid z)$ through Q, i.e. $Q_*(p(x \mid z))$; cf. Definition 2.5 in Cockayne *et al.* (2019) for a more rigorous version.

Computing the pushforward Q_* is trivial and straightforward. The interesting and difficult part is computing the posterior $p(x \mid z)$. However, by eq. (1.8), its tractability only depends on the prior $p(x)$ and the likelihood $p(z \mid x)$. We will now explain how the prior and likelihood are key to understanding how probabilistic numerics is both an **average-case** and **uncertainty-aware** paradigm.

The prior $p(x)$: Average-case numerical analysis (Ritter, 2000) is concerned with finding approximations to a numerical problems which are optimal when the numerical problem is sampled from a certain prior. For example, it is known that quicksort has optimal complexity of all permutations when a uniform prior over all permutations is assumed (Hoare, 1961). For non-finite problems, uniform priors do not exist; but these problems can still be equipped with (subjective) priors. Due to its excellent computational properties, Gaussian process (GP) priors are especially popular. For example, the GP-regression posterior mean is known to be average-case optimal for any GP prior (Rasmussen and Williams, 2006, Section 2.2). Consequently, Bayesian quadrature—which simply adds a QoI operator Q which integrates the estimated function—is also average-case optimal (O'Hagan, 1991) for any GP prior. For ODEs, however, the prior cannot be put on the problem definition f but on the solution x because GP regression should be performed on x and not on f. It is still average-case numerics in an important sense though: If the vector field $f(x(t), t)$ of the ODE does not depend on $x(t)$, then solving the ODE is equivalent to solving the integral $\int f(x, s)\, \mathrm{d}s$, for any $x \in \mathbb{R}^d$. It can, in fact, be shown that Bayesian ODE filters perform Bayesian quadrature for such vector fields and are therefore **average-case optimal** on this subclass of ODEs; cf. Wang *et al.* (2018).

The likelihood $p(z \mid x)$: The likelihood $p(x \mid z)$ links the unknown function $x \in \mathcal{X}$ to the data $z \in \mathcal{Z}$. In many cases, it is simply a Dirac measure at some exact evaluation of x at some input; for example in the case of GP regression (or Bayesian quadrature) with exact observations. In the case of ODEs, we do not have exact evaluations, but can only construct observations of the derivatives of $x(t)$ at some $t \in [0, T]$ through evaluations of $f(\hat{x}(t))$, as explained in Section 1.1. If we use normal distributions, this implies a likelihood of the form

$$p(z \mid x) = \mathcal{N}(z; f(\hat{x}(t)), V) \tag{1.9}$$

where z is treated as information on the derivative $\dot{x}(t)$ and the variance V is supposed to model the error $\|f(x(t)) - f(\hat{x}(t))\|$. Therefore, this non-Dirac likelihood models the import of the numerical uncertainty ($\hat{x}(t) \neq x(t)$) that classical numerical analysis ignores. Therefore, we say that probabilistic numerics is (unlike classical numerics) **uncertainty-aware**.

1.3 Numerics of ODEs: a solved topic?

ODEs are—due to the century-long relentless work of analysts of the caliber of Leibniz, Bernoulli and Richardson—analytically well-understood; see Kelley and Peterson (2010) or Teschl (2012) for a comprehensive summary of the accumulated analytical theory. The numerical analysis, however, was (absent modern computers) largely ignored until the 20^{th}: While Leonhard Euler (1768) (still mostly interested in the analysis of ODEs) invented the first numerical ODE solver now called *Euler's Method* and Bashforth and Adams (1883) introduced a generalization today known as *multi-step methods*, the interest in numerical solvers grew in the 20^{th} century alongside the relevance of computers. The most important breakthrough was provided by Runge–Kutta (RK) methods, another generalization of Euler's method by Runge (1895) and Kutta (1901), which collect multiple vector field evaluations along a single integration step; see Section 1.1. RK methods are the basis of today's *single-step methods*. As the third industrial revolution (aka digital revolution) took off in the second half of the 20^{th} and computers became omnipresent, RK methods were further developed, by the likes of NASA's Apollo program (Fehlberg, 1969), and are still the general go-to solution for most applications. In the past decades, excellent comprehensive books on the numerics of ODEs have been published; see e.g. Hairer *et al.* (1987), Hairer and Wanner (1996), Deuflhard and Bornemann (2002), and Butcher (2008). Many numerical analysts therefore consider ODEs to be a solved topic.

1.4 The new challenges and chances of machine learning

The inventors of classical methods for ODEs, however, did not foresee the advent of the Fourth Industrial Revolution (Schwab, 2017), in general, and of data-driven machine learning (ML) in particular. Therefore, we argue here, classical solvers may not be prepared for the next generation of computer science. The shortcomings are twofold: *Firstly*, classical solvers are not designed to interact with statistics and, by extension, with ML systems. Hence, we consider them suboptimal for some ML tasks; see Section 1.4.1 below. *Secondly*, classical solvers are formulated in the paradigm of classical numerics (see Section 1.1) and have not utilized the recent successes of other paradigms from ML (and statistics). Hence, we think that methods from ML can be used to solve ODEs; see Section 1.4.2 below. Some of the arguments presented next have been, in more generic form, made in Hennig *et al.* (2015) and Oates and Sullivan (2019).

1.4.1 Challenges: Why do we need novel numerical methods for ML tasks?

> *"Most of what is being called "AI" today, particularly in the public sphere, is what has been called "Machine Learning" (ML) for the past several decades. ML is an algorithmic field that blends ideas from statistics, computer science and many other disciplines to design algorithms that process data, make predictions and help make decisions."* —Michael Jordan (2018)

Machine learning is a data-driven field: without data, no predictions; without predictions, no improved decision making. The more noisy the data is, the more important is statistics in general and uncertainty quantification (Sullivan, 2015) in particular. However, this *statistical uncertainty* is not the only relevant kind of uncertainty.

Uncertainty can be subdivided into two high-level categories: aleatoric and epistemic uncertainty.[2] *Aleatoric uncertainty* signifies any unknown outcome that is truly stochastic, i.e. that is thought of as coming from as a random outcome that could be different if repeated—such as a random number, a dice role or some phenomena in stochastic quantum mechanics. On the contrary, all unknown outcomes that are thought of as non-stochastic (aka deterministic) are subsumed under *epistemic uncertainty*—such as the expected arrival time of a journey, a free model parameter or tomorrow's weather.[3]

In ML, both sources of uncertainty play an important role (Gal, 2016). While the aleatoric uncertainty is usually comprehensively treated by modeling the noise inherent in the observations and (potentially) random numbers, the epistemic uncertainty is often reduced to model uncertainty; see e.g. Kendall and Gal (2017). This, however, completely ignores the *numerical uncertainty* from imprecise numerical approximations. This uncertainty can be significant (Hennig *et al.*, 2015). For example, most of Bayesian machine learning relies on Bayesian model averaging which has to rely on quadrature in many cases (Fragoso *et al.*, 2018). The uncertainty from the quadrature is, however, largely ignored—which the PN-method Bayesian quadrature can remedy (Briol *et al.*, 2019).

Although ODEs are not as ubiquitous in ML as quadrature, they too add to the numerical uncertainty in many situations.[4] In fact, ODE inverse problems are a ML task on their own as they appear whenever the parameters of a dynamical system are inferred. This thesis contains a paper (Kersting *et al.*, 2020b) in Appendix D which

[2] Note that this dichotomy resembles the difference between frequentism and Bayesianism: aleatoric uncertainty assumes a repeatable frequentist experiment while epistemic uncertainty captures the Bayesian belief distribution of some observer.

[3] The ongoing debate whether true randomness exists is beside the point, since this definition is only concerned with how uncertainty is thought of. A random number, for example, is created by a deterministic random numbers generator, and it nevertheless makes sense to think of it as stochastic—since it would be futile to model the underlying physics.

[4] To see the import of numerical uncertainty from another angle, note that numerical uncertainty is an instance of logical uncertainty which is relevant for AI safety (Soares and Fallenstein, 2014).

shows how PN can combine aleatoric and numerical uncertainty in such problems. Other prominent cases of ODEs in ML include Hamiltonian Monte Carlo (Betancourt, 2017), neural ODEs (Chen *et al.*, 2018) and optimization (Scieur *et al.*, 2017). In all of these algorithms, statistical uncertainty interact with numerical uncertainty. Consider, for example, the case of stochastic optimization.

Example (stochastic gradient descent): The gradient descent algorithm for an objective function F has a well-known numerical interpretation as the integration of the gradient flow IVP, given by

$$\dot{x}(t) = -\nabla F(x), \qquad x(0) = x_0, \tag{1.10}$$

using Euler's method. Stochastic gradient descent algorithms, however, only receive evaluations of ∇F as information (Bottou *et al.*, 2018, Section 3.2). This means that they perform Euler steps with inaccurate information on the derivative. In other words, the aleatoric (statistical) uncertainty from the gradient estimate is fed into Euler's method which will translate it into epistemic (numerical) uncertainty. Here, too, a joint treatment of statistical and numerical uncertainty by PN might lead to better algorithms.

1.4.2 Chances: Is solving ODEs a ML task?

Judea Pearl (2018) recently caused a small scandal in the ML community by claiming that "*all the impressive achievements of deep learning amount to just curve fitting*". While we do not fully agree with this provocative statement, it nicely highlights our next point: If much (most?) of ML is a particularly efficient way of curve fitting (aka regression), then all curve fitting tasks should be ML tasks. Since ODE solvers fit the solution curve $x : [0, T] \to \mathbb{R}^d$, the application of ML regression-techniques to ODEs should engender new solvers.

This insight has, in recent years, been brought to bear not only by us: ODE solvers based on kernel ridge regression and deep learning have been introduced by Saitoh and Sawano (2016, Chapter 5) and Raissi *et al.* (2019) respectively. Our work uses Gaussian Process (GP) regression (Rasmussen and Williams, 2006, Section 2.2) instead, because the Bayesian paradigm comes with particular benefits over classical numerics (as we explained in Section 1.2) and GPs are the only Bayesian non-parametric regression technique with permissible computational cost (Ghahramani, 2013, Section 4). Maybe other regression techniques, such as regression trees (Segal, 1992), can also be applied to ODEs in this spirit.

The rest of the thesis consists of a summary and discussion of our published papers which are, in full, attached in the Appendix. In particular, we will concisely summarize how thinking about ODEs as a ML task leads to the efficient class of ODE solvers now called *ODE filters* and discuss how this research leads to the *three promises of probabilistic numerics* listed in the Preface.

2 Summary

Before we summarize our contribution to *Bayesian ODE filters*, we first set the stage by recalling Bayesian filtering for generic time series. This is didactically advantageous because ODEs are just an example of a continuous signal $x : [0, T] \to \mathbb{R}^d$ which is discretized by an ODE solver.

2.1 Gaussian filtering for generic time series

In signal processing, it is usually assumed that the signal is hidden but measurements $\{y_i; i = 1, \ldots, N\}$ of the discretized state $\{x_i;\ i = 1, \ldots, N\}$ are available. These states and measurements are modeled in a *probabilistic state space model* (SSM) consisting of

$$\text{a dynamic model} \quad x_i \sim p(x_i \mid x_{i-1}), \quad \text{and} \tag{2.1}$$

$$\text{a measurement model} \quad y_i \sim p(y_i \mid x_i). \tag{2.2}$$

The dynamic model is usually thought of stemming from a continuous prior on $x : [0, T] \to \mathbb{R}^d$. The measurement model is equivalent to a likelihood for the data provided by the measurements. Unless very specific information is available, this prior is chosen to be a linear time-invariant (LTI) stochastic differential equation (SDE), written

$$p(x) \ \sim \ X(t) = FX(t)\ \mathrm{d}t + L\ \mathrm{d}B(t), \tag{2.3}$$

with Gaussian initial distribution $\mathcal{N}(m_0, P_0)$ on $X(0)$, where the drift and diffusion matrices $F, L \in \mathbb{R}^{D\times D}$ define the deterministic and stochastic part of the dynamics respectively. The unique solution of eq. (2.3) is a GP with mean $m : [0, T] \to \mathbb{R}^D$ and covariance matrix $P : [0, T] \to \mathbb{R}^{D\times D}$ given by

$$m(t) = A(t)m(0), \quad \text{and} \tag{2.4}$$

$$P(t) = A(t)P(0)A(t)^\intercal + Q(t), \tag{2.5}$$

where (A, Q) can be derived from (F, L) in closed form; see eqs. (C.76) and (C.77). Accordingly, (F, L) parametrize the prior $p(x)$. Choosing the prior via (F, L) is fully analogous to prior selection for GP regression, as e.g. described in Rasmussen and Williams (2006, Chapter 5), but restricted to Gauss–Markov processes; see Section 2.3. If the

Algorithm 1 Generic Bayesian Filtering

1: **Input:** data $\{y_i; i = 1, \ldots, N\}$, initial distribution $\mathcal{N}(m_0, P_0)$, SSM eqs. (2.1) and (2.2), $i = 1$
2: **repeat**
3: $\quad i = i + 1$
4: $\quad$ **compute predictive distribution** $p(x_i \mid y_{1:i-1})$ by eq. (2.1) from $p(x_{i-1} \mid y_{1:i-1})$
5: $\quad$ **compute filtering distribution** $p(x_i \mid y_{1:i})$ by eq. (2.2) from $p(x_i \mid y_{1:i-1})$
6: **until** $i = N$

measurement model is Gaussian as well, i.e.

$$p(y_i \mid x_i) = \mathcal{N}(y_i; Hx_i, R), \tag{2.6}$$

for matrices H, R, posterior distributions can be computed by very efficient algorithms: The filtering distribution $p(x_i \mid y_{1:i})$ can be computed by Gaussian (Kalman) filtering in linear time $\mathcal{O}(N)$. The full posterior $p(x_i \mid y_{1:N})$ can be obtained by running Rauch—Tung—Striebel (RTS) smoothing afterwards which maintains linear cost—much faster than the cubic cost of Rasmussen and Williams (2006). If the measurement model is non-linear or non-Gaussian, non-linear filtering (and smoothing) techniques can be used to approximate the filtering and posterior distributions by Gaussian (e.g. extended/unscented Kalman filter) or sampling-based (e.g. particle filtering) approximations. A generic Bayesian Filtering algorithm is presented in Algorithm 1. More information on generic Bayesian filtering and smoothing can, e.g., be found in Anderson and Moore (1979) and Särkkä (2013).

Example (Kalman Filtering): This example is instructive to build intuition and see how fast Bayesian Filtering can be. Kalman Filtering computes *exact* filtering distributions $p(x_i \mid y_{1:i})$, $i = 1, \ldots, N$, in the framework of Algorithm 1 if the dynamic and measurement models are both linear and Gaussian. This is the case when the dynamic and measurement model can be written as $p(x_i \mid x_{i-1}) = \mathcal{N}(Ax_{i-1}, Q)$ and $p(y_i \mid x_i) = \mathcal{N}(y_i; Hx_i, R)$. In this case, the predictive and filtering distributions are Gaussians (Särkkä, 2013, Theorem 4.2), written

$$p(x_k \mid y_{1:k-1}) = \mathcal{N}(x_k; m_k^-, P_k^-), \quad \text{and} \qquad p(x_k \mid y_{1:k}) = \mathcal{N}(y_k; m_k, P_k). \tag{2.7}$$

Both the prediction step (Line 4) and the update step (Line 5) of Algorithm 1 are now cheap linear algebra computations.

- The **prediction step** is

$$m_k^- = Am_{k-1}, \qquad P_k^- = AP_{k-1}A^\intercal. \tag{2.8}$$

- The **update step** is

$$v_k = y_k - H m_k^-, \quad S_k = H P_k^- H^\intercal + R, \quad K_k = P_k^- H^\intercal S_k^{-1}, \tag{2.9}$$

$$m_k = m_k^- + K_k v_k, \qquad P_k = P_k^- - K_k S_k K_k^\intercal. \tag{2.10}$$

These steps are of cubic cost in D (due to the inversion of S_k), and (more importantly) linear cost in N.

2.2 Two new constructions of state space models for ODEs

As an efficient Bayesian model for a temporal function, a probabilistic SSM is perfect for the Bayesian paradigm of PN, as discussed in Section 1.2. In our work, we have introduced two distinct ways to define such a SSM for ODEs—published in Kersting and Hennig (2016) and Tronarp *et al.* (2019a). While both use the same dynamic model (prior), they employ a different measurement model (likelihood). While the first model from Kersting and Hennig (2016) is more intuitive (as it resembles the internal logic of classical solvers), the second model from Tronarp *et al.* (2019a) is more rigorous and general. Since the first one creates (like RK methods; recall eq. (1.2)) 'data' by evaluating the vector field f, we call it *a SSM with generated data*. The second one does not, and instead conditions 'on the ODE itself' as we will see below. Accordingly, we call it a *a SSM without data*.

2.2.1 A dynamic model for ODEs

As explained above, the dynamic model is determined by choosing the drift and diffusion matrices $F, L \in \mathbb{R}^{D\times D}$ of the SDE, eq. (2.3). This prior, the D-dimensional stochastic process $X : [0, T] \to \mathbb{R}^D$ (the solution of the SDE), is only a suitable model if the ODE solution $x : [0, T] \to \mathbb{R}^d$ can be linearly extracted from X:

$$x(t) \;\sim\; H_0 X(t), \quad \text{for some } H_0 \in \mathbb{R}^{d\times D}. \tag{2.11}$$

Moreover, to incorporate information on the derivative, $\dot{x}(t)$ also has to be linearly extractable:

$$\dot{x}(t) \;\sim\; H X(T) \quad \text{for some } H \in \mathbb{R}^{d\times D}. \tag{2.12}$$

Given (H_0, H), we can flexibly define the SDE prior, eq. (2.3), by choosing (F, L). We discuss these choices below in Section 2.3. The dynamic model for ODEs is therefore,

like in eqs. (2.4) and (2.5), given by

$$p\left(x(t+h) \mid x(t)\right) = \mathcal{N}\left(x(t+h); A(h)x(t), Q(h)\right), \tag{2.13}$$

with some initial distribution $p\left(x(0)\right) = \mathcal{N}\left(x(0); m(0), P(0)\right)$.

2.2.2 A Gaussian state space model for ODEs with generated data

In our first publication (Kersting and Hennig, 2016), the SSM is completed with generated data. This means that at any time $t \in [0, T]$, given some estimate $p(x(t)) = \mathcal{N}(x(t); m^-(t), P^-(t))$ of $x(t)$, we can 'generate' data y_t on $\dot{x}(t)$ via the equation

$$\dot{x}(t) \overset{eq.\ (1.1)}{=} f(x(t)) \approx f(m^-(t)) =: y_t, \tag{2.14}$$

where the $\approx$ holds because $m^-(t) \approx x(t)$. In other words, we choose

$$p\left(\dot{x}(t) \mid x(t)\right) = \mathcal{N}\left(\dot{x}(t); y_t, R\right), \tag{2.15}$$

where $R \geq 0$ is an additional hyperparameter for the covariance (matrix) of y_t as an estimator of $\dot{x}(t)$. (See Section 2.4 for a discussion of how R can be chosen.) The complete SSM is now given by the dynamic model in eq. (2.13) and the measurement model in eq. (2.6).

Why is this not a rigorous model?

The process of generating data resembles what classical ODE solvers do. Classical ODE solvers also construct observations ('data') of derivatives by evaluating f at some estimate $\hat{x}(t)$. The analogy becomes clear when one compares the data generation, eq. (2.14), with the interpolation data RK methods receive from eq. (1.6). But this intuitive construction, comes at a cost regarding rigor.

The goal of Bayesian inference is to approximate the posterior as uniquely defined by Bayes' rule. To apply Bayes' rule, three distinct objects are required: the prior, the likelihood, and the data. These three components are supposed to be different and independent of each other. In our first SSM, however, both the likelihood and the data depend on the prior. To see this, recall from eq. (2.8) that the predictive mean m^- is the mean of the predictive distribution $p(x_i \,|\, y_{1:i-1})$ which is a conditioned version of the prior $p(x)$ defined by the SDE, eq. (2.3). Hence, $y_t = f(m_t^-)$ (the data) and Hm_t^- (part of the likelihood) depend on the prior.

Algorithm 2 Bayesian ODE Filtering

Input: IVP(x_i, m, T), step size $h > 0$, SSM eqs. (2.13) and (2.17), $t = 0$
repeat
  compute predictive distribution $p(x(t+h) \mid z(0:t))$ by eq. (2.13) from $p(x(t) \mid z(t))$
  compute filtering distribution $p(x(t+h) \mid z(t+h))$ by eq. (2.17) from $p(x(t+h) \mid z(t))$
  $t = t + h$
until $t + h > T$

2.2.3 A flexible state space models for ODEs without data

To design a rigorous model, we thus have to remove the dependence of data and likelihood on the prior. But at any time $t > 0$, all estimates will have been computed by some extrapolation scheme which will inevitably depend on the chosen model. Hence, the only information independent of the prior is the ODE, $\dot{x}(t) = f(x(t))$, itself. Since both a model $H_0X(t)$ of $x(t)$ and $HX(t)$ of $\dot{x}(t)$ are contained in the state space by construction, eqs. (2.11) and (2.12), insertion of these models into the ODE yields the following *information*

$$Z(t) := f\left(H_0X(t)\right) - HX(t) = 0. \tag{2.16}$$

This is to say that, at any time $t \in [0, T]$, we observe that the difference between the implied derivative of our solution estimate $HX(t)$ and our derivative estimate $H_0X(t)$ ought to be zero, according to our ODE. In the diagram of Figure 1.4, the image of the information operator Z is thus $\{0\} \subset \mathbb{R}^d$. The likelihood is accordingly defined by

$$p(z(t) \mid x(t)) = \mathcal{N}(0;\ f(x(t)) - \dot{x}(t), R), \tag{2.17}$$

with data $z(t) \equiv 0$, for all $t \in [0, T]$. We introduced this SSM in more detail in Tronarp *et al.* (2019a); see Appendix B.2. Moreover, we showed that this rigorous model is more flexible and contains the earlier models by Schober *et al.* (2019) and Kersting and Hennig (2016); see Propositions B.2.3 and B.2.4 in Appendix B.2.

Equipped with this general SSM, we can now define Bayesian ODE Filtering, see Algorithm 2—which is completely analogous to Algorithm 1, except for the input (i.e. the problem and the SSM).

A unified Bayesian framework for ODE filters

In Tronarp *et al.* (2019a), we show that ODE Filtering with this more general SSM unifies all existing ODE filters and Bayesian quadrature with Markov kernels; see Appendix B. In Proposition B.2.1, we show that Gaussian ODE filtering (plus smoothing) applied

to an ODE whose vector field only depends on time t (but not on $x(t)$) reproduces the Bayesian quadrature approximation posterior; cf. Theorem 1 in Wang *et al.* (2018). In Proposition B.2.3, we show that ODE filtering in the first SSM in the sense of Schober *et al.* (2019), with $R = 0$ in eq. (2.15), can be reproduced in this framework. As shown in Proposition B.2.4, the same holds true for ODE filtering with $R > 0$ as introduced by Kersting and Hennig (2016).

2.3 New priors for flexible model selection

In all of our publications, except for Kersting and Mahsereci (2020) which models Fourier components, we restrict our attention to SSMs that model the first first q derivatives of $x(t)$. This is to say that we model the vector-valued function of x and its q first derivatives by the $(q+1)$-dimensional $\boldsymbol{X} = \left(X^{(0)}, \ldots, X^{(q)}\right)^{\intercal}$ which solves

$$\mathrm{d}\boldsymbol{X}(t) = \left(\mathrm{d}X^{(0)}(t), \ldots, \mathrm{d}X^{(q-1)}(t), \mathrm{d}X^{(q)}(t)\right)^{\intercal} \tag{2.18}$$

$$= \begin{pmatrix} 0 & 1 & 0 \ldots & 0 \\ \vdots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & 1 \\ c_0 & \ldots & \ldots & c_q \end{pmatrix} \begin{pmatrix} X^{(0)}(t) \\ \vdots \\ X^{(q-1)}(t) \\ X^{(q)}(t) \end{pmatrix} \mathrm{d}t + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sigma \end{pmatrix} \mathrm{d}B(t). \tag{2.19}$$

This class of SDEs, parametrized by $c = (c_0, \ldots, c_q)$, coincides with the GPs with Matérn kernel (Hartikainen and Särkkä, 2010) which is unsurprising since Matérn processes of this form are the standard model for Gauss–Markov processes with q continuous derivatives. Our publications do not consider all choices of c, but restrict their attention to $c = (0, \ldots, 0, -\theta)$, $\theta \geq 0$—i.e. to the q-times Integrated Ornstein Uhlenbeck processes (IOUP) and their special case, the q-times Integrated Brownian motion (IBM), if $\theta = 0$.

The IBM prior has been known to be particularly useful for ODEs, since Schober *et al.* (2014) proved that individual steps of q-stage RK methods coincide with a GP posterior mean with q-times IBM kernel for $q \in \{1, 2, 3\}$ (and a likelihood analogous to the one introduced in Section 2.2.2). The deep reason for this is that the predictive mean $m^{-}(t+h)$ is a Taylor expansion of the previous filtering mean $m(t)$, i.e.

$$m^{-}(t+h) \overset{\text{eq. (2.4)}}{=} A(h)m(t) \overset{\text{eq. (C.79)}}{=} \sum_{i=0}^{q+1} \frac{h^i}{i!} m^{(i)}(t), \tag{2.20}$$

where $m^{(i)}(t)$ is the filtering mean estimate of $x^{(i)}(t)$, which resembles the logic of RK methods from eq. (1.5).[1] Interestingly, the drifting (biased) q-times IOUP prior can

[1] Note that we here assume w.l.o.g. that $d = 1$; see Appendix C.12 for a justification for the underlying independent-dimensions assumption.

outperform the RK-like IBM prior on problems with bounded vector fields f (Magnani *et al.*, 2017), while staying close enough to Taylor predictions to maintain $\mathcal{O}(h^{q+1})$ local convergence rates. (It is currently not known under which conditions the IOUP prior has the same convergence rates as IBM; see Remark C.6.3.) The q-times IBM prior is an uninformative prior because it assumes a constant (i.e. non-drifting) q^{th} derivative while the other derivatives are fixed by the fundamental theorem of calculus (as captured by the 1s on the off-diagonal of eq. (2.19)). The IOUP prior introduces a drift on the q^{th} derivative which is more informative and therefore suboptimal in the worst-case, but potentially better in the average-case when the dynamical system is on-average expected to drift back to some equilibrium level. It remains to be seen if ODEs can be categorized such that the whole Matérn family, eq. (2.18), becomes useful.

In a more radical deviation from classical numerics, we also introduced a SSM that employs Fourier instead of Taylor expansions as predictions (Kersting and Mahsereci, 2020)—with the hope that this will be useful for periodic ODEs such as the Van-der-Pol, FitzHugh–Nagumo, and Lotka–Volterra oscillators. While the above models the summands of the Taylor expansion, this SSM accordingly models the first $J \in \mathbb{N}$ summands of the Fourier expansion (as well as the first J summands of the derivative). Its dynamic model (prior) is defined by the SDE from eq. (2.3) with

$$F = \operatorname{diag}(F_1, \ldots, F_J), \quad \text{with blocks } F_j = \begin{bmatrix} 0 & -jw_0 \\ jw_0 & 0 \end{bmatrix},\ q = 1, \ldots, J \qquad \text{and} \quad (2.21)$$

$$L = \mathbf{0} \ \in \mathbb{R}^{2(J+1)\times 2(J+1)}, \quad (2.22)$$

where $w_0 > 0$ is the angular velocity. Notably, the SDE does not have a zero diffusion L, since (unlike Taylor coefficients) Fourier coefficients are global and do not change with $t \in [0, T]$. The solution $x(t)$ and its derivative $\dot{x}(t)$ can now be extracted, as described in eqs. (2.11) and (2.12), by use of

$$H_0 = [1, 0, 1, 0, \ldots, 1, 0] \ \in \mathbb{R}^{1\times 2(J+1)}, \quad \text{and} \quad (2.23)$$

$$H = \begin{bmatrix} 0, 0, 0, -1w_0, 0, -2w_0, \ldots, 0, -Jw_0 \end{bmatrix} \in \mathbb{R}^{1\times 2(J+1)}.$$

This prior can be completed by both the intuitive and rigorous measurement model. The details of the construction are given in Appendix E. The resulting Fourier filters are, however, not practical yet and rely on support from IBM filters to learn in the beginning of the time interval; see Appendix E.4. We, however, believe that they will become functional in the future because the proposed SSM is an approximation of the periodic kernel (the standard model for periodic signals), as proved in (Solin and Särkkä, 2014).

2.4 Better probabilistic calibration by uncertainty-awareness

Recall from eq. (1.6) that classical solvers ignore the uncertainty stemming from the fact that almost always $\dot{x}(t) \neq f(\hat{x}(t))$, for $t > 0$. The significance of this uncertainty-unawareness is demonstrated above in Figure 1.2. In Kersting and Hennig (2016), we therefore rectified this shortcoming by modeling this mismatch probabilistically. In the intuitive SSM from Section 2.2.2, we first observe that, given a predictive distribution $\mathcal{N}(m_t^-, P_t^-)$ over $x(t)$ at some time $t \in [0, T]$, the true implied distribution over $\dot{x}(t)$ is given by the pushforward measure $f_* \left(\mathcal{N}(m_t^-, P_t^-)\right)$. Since f is non-linear, conditioning on it is intractable. We therefore propose to moment-match this pushforward measure to a Gaussian, i.e. use the approximation

$$\mathcal{N}\left(\int f(\xi)\, \mathrm{d}\mathcal{N}(\xi; m_t^-, P_t^-), \int [ff^\intercal](\xi)\, \mathrm{d}\mathcal{N}(\xi; m_t^-, P_t^-)\right) \;\approx\; f_*\left(\mathcal{N}(m_t^-, P_t^-)\right) \tag{2.24}$$

instead. The integrals in eq. (2.24) have to be approximated by numerical quadrature which adds numerical uncertainty on top—in particular since only few evaluations of f are affordable for each t. Hence, we propose to use Bayesian Quadrature (Briol *et al.*, 2019) for the approximation, which captures this uncertainty probabilistically. In full PN-spirit, the posterior BQ-variance estimate of the expectation integral is added to the existing variance from extrapolation. The resulting algorithm, *Bayesian Quadrature ODE Filtering*, in deed shows more adaptive and flexible uncertainty calibration, compared to the variance-less ($R = 0$; see eq. (2.15)) measurement model from Schober *et al.* (2019); see Appendix A.4.

2.5 New algorithms for ODE inverse problems

The all-inclusive managing of uncertainty (numerical and statistical) through computational chains with multiple steps is a long-term vision of PN (Hennig *et al.*, 2015, Chapter 3(d)). The most elementary such chain involving ODEs, is an *ODE inverse problem* where the parameter $\theta \in \mathbb{R}^n$ of a parametrized ODE, written

$$\dot{x}(t) = f\left(x(t), \theta\right), \qquad x(0) = x_0 \in \mathbb{R}^d, \tag{2.25}$$

is to be inferred from noisy data

$$z(t_i) \coloneqq x(t_i) + \varepsilon_i \in \mathbb{R}^d, \quad \varepsilon_i \sim \mathcal{N}(0, \Sigma_i), \tag{2.26}$$

at times $0 \leq t_1 < \cdots < t_M \leq T$. Such problems are ill-posed which is to say that two parameters $\theta_1 \neq \theta_2$ might fit the data equally well. Therefore, they rather fall under the rubric of statistics (or machine learning) than numerical analysis. Accordingly,

most existing work on ODE inverse problems deals with the statistical variances Σ_i, $i = 1, \ldots, M$ elaborately while tacitly passing over the numerical uncertainty. In other words, they treat ODE inverse problems as 'likelihood-free inference' (Cranmer *et al.*, 2020) in that they do not model the likelihood of their numerical ODE solutions. The most popular inverse problem solvers do this by computing the likelihood for a parameter θ as if the likelihood of the forward map, $p(x_\theta \mid \theta)$, was a Dirac distribution $\delta(x_\theta - \hat{x}_\theta)$ (ignoring that $\hat{x} \neq x$). This means that it employs the *uncertainty-unaware likelihood*

$$p(z \mid \theta) = \int p(z \mid x_\theta) p(x_\theta \mid \theta) \, \mathrm{d}x_\theta = \int p(z \mid x_\theta) \delta(x_\theta - \hat{x}_\theta) \, \mathrm{d}x_\theta = \mathcal{N}\left(z; \hat{x}_\theta, \Sigma\right), \quad (2.27)$$

where x_θ denotes the solution of eq. (2.25) for some fixed θ. If we instead use the (posterior) filtering distribution $p(x_\theta \mid \theta) = \mathcal{N}(x_\theta; m_\theta, P)$, we obtain the *uncertainty-aware likelihood*

$$p(z \mid \theta) = \int p(z \mid x_\theta) \mathcal{N}(x_\theta; m_\theta, P) \, \mathrm{d}x_\theta = \mathcal{N}(z; m_\theta, P + \Sigma) \quad (2.28)$$

in which the numerical and statistical variances, P and Σ, add up *as they should*; see Figure D.2 for an illustration of how this likelihood is more suitable.

But we do not stop here, as the previous uncertainty-aware PN approaches did; see Appendix D.3.1 for a comparative discussion of the preceding work. We recall that, for every data time point t_i the filtering distribution $p(x_i \mid y_{1:i}) = \mathcal{N}(m_i, P_i)$ is the posterior of a GP with mean

$$m_\theta(t_i) = x_0 + \left[{}^{\partial}K^{\partial}(h : t_i) + R \cdot I_{l_i}\right]^{-1} k^{\partial}(h : t_i, t_i) f(m_\theta^-(0 : t_i)), \quad (2.29)$$

where $k^{\partial} = \partial k(t, t')/\partial t'$ and ${}^{\partial}k^{\partial} = \partial^2 k(t, t')/\partial t \partial t'$ are derivatives of the kernel k. Hence, if we assume that $f(x, \theta) = \sum_{i=1}^{n} x_i \theta_i$ and omit the import of ∇f under the chain rule, we can derive an estimator J of the Jacobian of the map $\theta \mapsto m_\theta$ which only involves quantities that the Gaussian ODE filter computes anyway. This estimator then gives rise to cheap estimators of the gradient and Hessian of the log-likelihood E, namely

$$\hat{\nabla}_\theta E(z) \coloneqq -J^\intercal \left[P + \Sigma\right]^{-1} \left[z - m_\theta\right], \quad \text{and} \quad (2.30)$$

$$\hat{\nabla}^2_\theta E(z) \coloneqq J^\intercal \left[P + \Sigma I_M\right]^{-1} J. \quad (2.31)$$

These two advantages (uncertainty-awareness and gradient information) can then be used to construct better sampling and optimization methods for ODE inverse problems that outperform existing 'likelihood-free' methods; see Appendix D for details of theory and experiments.

2.6 Theoretical analysis of ODE filters

Lastly, we support the above practical and conceptual advances by theory that comprises both analogues to classical theory and new PN-specific results. The former cover the two main pillars of numerics, namely convergence rates and stability; the latter covers connections between existing PN methods and the calibration of the posterior uncertainty. The following results are all contingent to modest regularity assumptions on f which are stated in the respective appendices.

2.6.1 Classical theory for ODE filters

In numerical analysis, the two main quality criteria for ODE solvers are convergence speed and numerical stability. The **convergence rates** of Gaussian (Kalman) ODE filters are analyzed in Kersting *et al.* (2020a). It turns out that both the q-times IBM and the IOUP prior (as their extrapolations do not deviate from Taylor expansions too much) have polynomial local convergence rates of order $\mathcal{O}(h^q)$, for all $q \in \mathbb{N}$ and all measurements variances $R \geq 0$ in the intuitive SSM of Section 2.2.2; see Theorem C.6.2. Moreover, if the variance R shrinks at least at rate $\mathcal{O}(h^q)$, the global convergence rate for $q = 1$ is $\mathcal{O}(h^q)$ as proved in Theorem C.7.7, and our experiments suggest that this first global result might generalize to $q \in \{2, 3, \dots\}$.

Concerning **numerical stability**, we show in Theorem B.3.5 that the Gaussian (Kalman) ODE filter is A-stable, which is to say that the numerical solution of a linear ODE with negative eigenvalues converges to zero as $t \to \infty$ (Dahlquist, 1963).

2.6.2 Uncertainty calibration in ODE filters

If the prior SDE, eq. (2.3), has a non-zero diffusion matrix L, the posterior uncertainty of any ODE filter scales with the standard deviation $\sigma > 0$ of the Brownian motion. In parallel work, a local (i.e. adaptive, step-wise) maximum-likelihood estimate for σ has been provided by (Schober *et al.*, 2019, Section 4). In Proposition B.4.1, we provide a global maximum-likelihood that does not have to be adapted in every step. Moreover, we analyze both the uncertainty calibration for the Gaussian (Kalman) ODE filter and the non-parametric particle filter. In Theorem C.8.1, we show that (in the above-detailed case where we proved that the global truncation error of the Gaussian ODE filtering mean is in $\mathcal{O}(h^q)$) the Bayesian credible intervals (aka. multiples of the posterior standard deviation) shrink at exactly the same rate as well. For particle filtering, we show (in the weak sense) that the expected error of the particle approximation of the true filtering distribution shrinks as $\mathcal{O}(1/\sqrt{J})$, where J is the amount of particles; see Theorem B.2.6.

3 Discussion and Conclusion

In our research, we set out to investigate which benefits the paradigm of probabilistic numerics brings to ODEs. We found out that ODEs can be efficiently solved by Bayesian ODE filtering. To this end, we provided a rigorous probabilistic state space model (SSM) that unlocks all Bayesian filters for ODEs (Gaussian or non-Gaussian) in Tronarp *et al.* (2019a). Via the underlying SSM (prior and likelihood) and the choice of intra-algorithmic approximations (in the measurement model), we can create both algorithms which have very similar properties to classical methods (for a small overhead) and completely new ones which defy all conventions. To advance both undertakings, we showed that the q-times integrated Brownian motion (IBM) gives convergence rates comparable with classical methods, as it uses Taylor expansion predictions (Kersting *et al.*, 2020a), and introduced a filter that employs Fourier predictions instead which might turn out to be useful for periodic systems (Kersting and Mahsereci, 2020). Moreover, we provided one of the first demonstrations for the long-term vision that passing uncertainties through computational chains by PN (Hennig *et al.*, 2015, Section 3(d)) can improve performance: In Kersting *et al.* (2020b), we showed that the uncertainty-aware likelihood provided by a Gaussian ODE filter speeds up solutions of inverse problems by providing more suitable parameters and thereby reducing the amount of necessary forward solutions.

The research field of PN in general, and of ODE filters in particular, is still in its early stages. While some foundational ground is provided by this thesis (as well as by the one of Schober (2018)), there are still many open questions. In this final chapter, we discuss which research questions immediately arise from our work in Section 3.1 and how our results advance the three promises of PN from the Preface in Section 3.2.

3.1 Future research

The presented material might pave the way to new advances in theory, the development of algorithms, and applications.

3.1.1 Theory

Our theoretical contributions (pertaining to convergence rates, stability analysis, and uncertainty quantification) lead to further research questions. Firstly, the proofs of the global convergence rates in Kersting *et al.* (2020a, Theorem 14) for Kalman ODE

filters are limited to the case of $q = 1$ and q-times IBM prior. There are, however, both conceptual (e.g., the Taylor expansion predictions of IBM priors) and experimental reasons (see Figure C.2) to believe that these rates might extend to $q \in \{2, 3, \dots\}$. Therefore, an attempt to generalize the proofs might bear fruits. We believe that the bottleneck for such results is a generalization of Proposition C.7.2. Such an analysis might also lead to a generalization of the uncertainty calibration in Kersting *et al.* (2020a, Theorem 15). Furthermore, the convergence rates of other ODE filters (e.g., the extended and unscented Kalman filter) should be examined. The latest analysis (Tronarp *et al.*, 2020), includes a convergence analysis for all $q \in \mathbb{N}$ for the maximum a posteriori (MAP) estimate, which can be computed with the more-costly iterated extended Kalman ODE smoother. It remains to be seen if such convergence rates also apply to the filtering mean, which we considered in Kersting *et al.* (2020a).

Moreover, an extension of the stability analysis of Tronarp *et al.* (2019a), which used the concept of A-stability, is called for. In particular, it would be important to investigate the stricter concepts of L-stability and B-stability (Hairer *et al.*, 1987, Chapters IV.3 and IV.12) for ODE filters.

To fully capture the competitive performance, an average-case analysis à la Ritter (2000) might be necessary. In a worst-case sense (at least for SSM that model the q first derivatives of x), one can never outperform (iterated) q^{th} Taylor expansions as performed by the Gaussian ODE filter with q-times IBM prior. However, for specific ODEs, a 'biased' deviation of Taylor expansions might be appropriate, as we postulate in Magnani *et al.* (2017) for bounded derivative fields and in Kersting and Mahsereci (2020) for oscillators. An average-case analysis might capture the utility of such approximations, but seems difficult for the following reason: Such an analysis would employ a prior $p(f)$ instead of a prior $p(x)$ which the SSM defines via eq. (2.18). Hence, we would have to match a prior $p(f)$ to a prior $p(x)$ which would require computing the pushforward of $p(f)$ through the map $f \mapsto x$. This computations is intractable, as it would require computing the ODE solution x for all f in the support of $p(f)$. It seems difficult to find good approximations here. As a first step, it might however suffice to categorize problems by $p(x)$ and match properties of f (e.g. by the amount of its continuous derivatives) to this prior on x.

Furthermore, it is conceivable that more equivalences with classical methods can be added to the known ones from (Schober *et al.*, 2019). These equivalences are usually shown for the steady state (because otherwise the Kalman gains keep changing), and the new steady states from Kersting *et al.* (2020a, Proposition 10) might therefore help to identify more such equivalences.

Lastly, it should be analyzed whether Bayesian ODE filters can be viewed as an approximation of a Bayesian probabilistic numerical method as defined in Cockayne *et al.* (2019, Definition 2.2). So far, no probabilistic ODE solver has satisfied this strict definition (because it is unclear how the information of $f(\hat{x}(t))$ relates to x since $x \neq \hat{x}$). Since the 'true' posterior which considers all possibilities of information is well-defined, it should be possible to rank probabilistic methods by how precisely they approximate

this posterior; cf. Wang *et al.* (2018).

3.1.2 Development of algorithms

For **forward problems**, our foundational work in Tronarp *et al.* (2019a) already spans a vast domain of forward problem solvers, namely all Bayesian filters and smoothers as, e.g., collected in Särkkä (2013). Hence, future research can endlessly continue to invent new ODE solvers by translating methods from the ever-expanding literature of Bayesian filters and smoothers to ODEs. Although such future research might create very well-performing methods, it is (from a theoretical viewpoint) trivial.

It is more interesting to ask if ODE filters can go beyond signal-processing filters. And, here, the answer seems to be "Yes!" for the following reason: Conventional signal-processing filters receive 'data' from an external sensor, while ODE filters construct either the data (in the intuitive SSM of Kersting and Hennig (2016); see eq. (2.14)), or the likelihood (in the rigorous SSM of Tronarp *et al.* (2019a); see eq. (2.17)), by evaluating f. The vector field f, unlike external data collections, has a known structure which can be exploited. In particular, if $f(t, x)$ is independent of time t, and if $x(t)$ passes through a time point such that $x(s) \approx x(t)$, for some $s < t$, the evaluations of f collected at time s will contain similar information on $\dot{x}(t)$ as on $\dot{x}(s)$ (for which they were collected).

Let us consider this in more detail in the intuitive SSM (see Section 2.2.2): A Gaussian ODE Filter computes sequences of predictive distributions $\{\mathcal{N}(m_i^-, P_i^-);\ i = 1, \ldots, N\}$ and filtering distributions $\{\mathcal{N}(m_i, P_i); i = 1, \ldots, N\}$. Given a predictive distribution, the implied data on $\dot{x}(ih)$ is given by the pushforward measure $p(\dot{x}(ih)) = f_*(\mathcal{N}(m_i^-, P_i^-))$. To maintain the Gaussian framework, Kersting and Hennig (2016) propose to approximate this intractable measure by a moment-matched Gaussian, as we recall from eq. (2.24). This means that the information on the sequence $\{x(ih);\ m_i, P_i\}$

$$\mathcal{N}\left(y_i \coloneqq \int f(\xi)\ \mathrm{d}\mathcal{N}(\xi; m_i^-, P_i^-), \int [ff^\intercal](\xi)\ \mathrm{d}\mathcal{N}(\xi; m_i^-, P_i^-)\right) \ \approx\ f_*\left(\mathcal{N}(m_t^-, P_t^-)\right). \tag{3.1}$$

Therefore, we have a sequence of estimators

$$\{y_i = \int f(\xi)\ \mathrm{d}\mathcal{N}(\xi; m_i^-, P_i^-);\ i = 1, \ldots, N\} \tag{3.2}$$

of $\{\dot{x}(ih);\ i = 1, \ldots, N\}$ which have to be approximated by quadrature rules, i.e.

$$y_i = \sum_{j=1}^{n} w_i^{(j)} f\left(\xi_i^{(j)}\right). \tag{3.3}$$

For each individual integral, it is known how to optimally choose the weights $\{w_i^{(j)}\}_{j=1}^n$ and states $\{\xi_i^{(j)}\}_{j=1}^n$ for both classical (Press *et al.*, 2007) and Bayesian quadrature (Briol *et al.*, 2019). Here, however, we have a series of related intervals where only the Gaussian measure with respect to which f is integrated changes. Hence, we can re-use previous evaluations $\{f(\xi_i^{(j)}); j = 1, \ldots, n_j\}$ to compute y_k, at a later time $k > i$. To this end, one could first select a subset of states $\{\xi_i^{(j)}; j = 1, \ldots, n_j,\ i = 1, \ldots, k-1\}$, and then complement it with additional states $\{\xi_k^{(j)}; j = 1, \ldots, n_j\}$. The optimal weights to compute y_k are then implied by this choice of states (Novak and Wozniakowski, 2010). To go even further, the series of integrals from eq. (3.2) could also be interpreted as a series of integrals with related integrands

$$g(\xi) \coloneqq f(\xi) \cdot \mathcal{N}(\xi; m_i^-, P_i^-) \tag{3.4}$$

which could then be jointly treated as a Bayesian quadrature problem for multiple related integrals, following Xiaoyue *et al.* (2018) and Gessner *et al.* (2019).

Maybe one could even try to predict how useful certain evaluations $f(\xi)$ are at any point $t \in [0, T]$ in the algorithm. The earlier the time t, the more useful such an evaluation would be because it can be re-used. Such a line of investigation might lead to an adaptive version of Mohammadi *et al.* (2019), where all evaluations of f are performed in the first step to learn a model of the flow map to extrapolate forward.

More ideas for this kind of active learning of ODE filtering (i.e. the designed collection of information from evaluating f) could be borrowed from the literature on active learning for GP regression (Seo *et al.*, 2000), Bayesian Optimization (Mockus and Mockus, 1991) and Bayesian Quadrature (Osborne *et al.*, 2012).

On a separate note, it should be explored which statistical estimators are most useful. Within Gaussian filters, one can either focus on the filtering mean (as we have in our publications), the smoothing mean, or the MAP estimate (Tronarp *et al.*, 2020, Section 2.3) which are all computed by different methods. In the case of particle filters, one could also consider the modes of a distribution as an estimator.

Inverse problem

All ODE filters can, of course, be part of inverse problem solvers as well—including those to be invented by the above-described means. In fact, Kersting *et al.* (2020b) provides a detailed framework in which any ODE filter can be incorporated into both (gradient-based) optimization and sampling methods. Hence, each new ODE filter implies a new ODE inverse problem solver. This vast horizon of possibilities has only been explored by Kersting *et al.* (2020b), i.e. only for Kalman ODE filters with IBM prior. While the insertion of other Gaussian filters (with different SSMs and approximation methods) might outperform this first approach, particle filtering (Tronarp *et al.*, 2019a, Section 2.7) might provide a more significant advance since it can represent arbitrary distributions on $x(t)$ by samples.

To see this, recall that both Conrad *et al.* (2017) and Abdulle and Garegnani (2020) employ sampling-based probabilistic ODE solvers, in a Bayesian inverse problem framework, to sample from a non-Gaussian distribution of possible numerical approximations of x. Since, in both cases, the target density of different sample solutions is not known, they have to use a pseudo-marginal Metropolis–Hastings algorithm (Andrieu *et al.*, 2010) to approximate the posterior distribution. The exact same framework could be used with the particle filter as a forward map which would eliminate the need to run a forward map multiple times. This would probably provide an alternative sampling-based way, to prevent numerical over-confidence in ODE inverse problems. Such a method would, thus, combine the filtering of Kersting *et al.* (2020b) and the pseudo-marginal sampling of Conrad *et al.* (2017) and Abdulle and Garegnani (2020).

3.1.3 Further applications

One of the main visions of PN is to propagate uncertainty through computational chains (Hennig *et al.*, 2015, Section 3(d)), and inverse problems are, as discussed in the preceding paragraph, an elementary example of such problems. There is a huge number of such chains, allover science and engineering, and (when no real data is involved) the passing of numerical uncertainty along such chains seems straightforward, once the error of all subroutines are quantified by (Gaussian) probability measures. Maybe a more interesting case are computational chains which link ODEs with data, such as in data-centric engineering (Girolami, 2020). It remains to be seen in which such settings a joint treatment of numerical and statistical uncertainty with the help of PN can be recommended. Due to our strong experimental results for ODE inverse problems (Kersting *et al.*, 2020b, Section 7), we expect them to be plentiful.

A promising next step could be taken by solving the ODEs of a geodesics manifold that was learned from data (Hauberg, 2018). This would go beyond our work, in a fundamental way, because the uncertainty in the data would directly translate into uncertainty over the vector field f, i.e. into model uncertainty. In the intuitive SSM of Section 2.2.3, this model uncertainty can be added to the measurement variance R—as discussed in Kersting *et al.* (2020a, Section 2.3).

On a separate note, our work on inverse problems could also spin off other applications. First, the Jacobian estimator J of the forward map, as defined in eq. (D.11), approximates the true gradient $\nabla_\theta x$ of x with respect to θ. Hence, this estimator could be used in lieu of sensitivity analysis (Rackauckas *et al.*, 2018) in all settings where such gradients are required—such as neural ODEs (Chen *et al.*, 2018). Secondly, similar inverse problem solvers could be developed, via the method of lines (Schiesser and Griffiths, 2009) for PDEs, and, via the probabilistic solver of John *et al.* (2019), for boundary value problems.

3.2 Conclusion: the three promises of PN

To close the arch of this thesis, we finally return to the punch line of the Preface: our advances of the *three promises of PN*, namely

1. more flexible *classification* of problems by statistical model selection,
2. comprehensive *uncertainty quantification*, and
3. *invention* of new average-case-optimal algorithms.

Firstly (i), our recast of ODEs as a stochastic filtering problem has introduced a complete and rigorous Bayesian model, a probabilistic state space model, as we detailed in Section 2.2. This has unlocked new ways to **classify** initial value problems. By choosing the prior $p(x)$, on the solution x (via the dynamic model), we can include our prior belief over the regularity and geometric properties of x, as we exemplified in categorizing ODEs with bounded derivatives under an integrated Ornstein–Uhlenbeck prior and oscillating ODEs under a Fourier prior (periodic kernel); see Section 2.3. For the case when nothing is known a priori, we have identified the integrated Brownian Motion prior as the go-to solution—because its Taylor-expansion predictions yield similar worst-case guarantees as classical models, as we explained in Section 2.6. Such classifications might make a paradigm shift from a worst-case to an average-case treatment of ODEs possible.

Secondly (ii), these ODE filters compute a full posterior measure over x which can be used to **quantify uncertainty**: In the case of Gaussian ODE filters, one can construct, e.g., 0.95 Bayesian credible intervals of $[m_t - 2\sqrt{P_t}, m_t + 2\sqrt{P_t}]$ for $x(t)$ from the filtering distribution $\mathcal{N}(m_t, P_t)$. For non-Gaussian filters, there is no standard way to represent the uncertainty and the right uncertainty will depend on the exact shape (e.g., number of modes) of the posterior. For both cases, we have, however, provided first theoretical results on the posterior uncertainty calibration; see Section 2.6.2. For Gaussian filters, we have, moreover, described new ways to model the uncertainty over the information on $\dot{x}(t)$ due to uncertainty over f, or where to evaluate it, by smart design of the likelihood (measurement model); see Section 2.4. This might pave the way to a joint handling of numerical (epistemic) and statistical (aleatoric) uncertainty, as we explained in Section 3.1.3.

Thirdly (iii), the interpretation of ODEs as filtering problems unlocks a simple recipe to **invent** new probabilistic ODE solvers: take a Bayesian filter (or smoother) from the signal processing literature and apply it to ODEs. We have only used this recipe for the most elementary filters so far (see Appendix B for a complete list), and expect to see new such inventions in the future. Moreover, the explicit knowledge of gathering information on $\dot{x}(t)$ via evaluations of f enables us to go beyond knowledge of signal processing by exploiting techniques from active learning. All of these inventions immediately engender new inverse problem solvers; see Section 3.1.2. In machine learning, the use of ODE filters could lead to more uncertainty-aware and robust algorithms, as discussed in Section 1.4.1.

We hope that the attentive reader will now be able to clearly see the benefits of PN and relate to the mystification of our imaginary scientist from the Preface.

Bibliography

Abdulle, A. and Garegnani, G. (2020). Random time step probabilistic methods for uncertainty quantification in chaotic and geometric numerical integration. *Stat. Comput.*, **30**(4), 907–932.

Alexander, R. (1977). Diagonally implicit Runge–Kutta methods for stiff ODEs. *SIAM Journal on Numerical Analysis*, **14**(6), 1006–1021.

Anderson, B. and Moore, J. (1979). *Optimal Filtering*. Prentice-Hall, Englewood Cliffs, NJ.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72**(3), 269–342.

Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons.

Bashforth, F. and Adams, J. C. (1883). *An attempt to test the theories of capillary action by comparing the theoretical and measured forms of drops of fluid*. Cambridge University Press.

Bell, B. M. and Cathey, F. W. (1993). The iterated Kalman filter update as a Gauss–Newton method. *IEEE Transaction on Automatic Control*, **38**(2), 294–297.

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

Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *Siam Review*, **60**(2), 223–311.

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., and Sejdinovic, D. (2019). Probabilistic integration: A role for statisticians in numerical analysis? (with discussion and rejoinder). *Statistical Sciences*, **34**(1), 1–22 (Rejoinder on p38–42).

Butcher, J. C. (2008). *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Inc., 2nd edition.

Byrne, G. D. and Hindmarsh, A. C. (1975). A polyalgorithm for the numerical solution of ordinary differential equations. *ACM Transactions on Mathematical Software*, **1**(1), 71–96.

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

Callier, F. M. and Desoer, C. A. (1991). *Linear System Theory*. Springer.

Cappé, O., Moulines, E., and Rydén, T. (2005). *Inference in Hidden Markov Models*. Springer.

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

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

Chow, T. Y. (1998). The surprise examination or unexpected hanging paradox. *American Mathematical Monthly*, **105**, 41–51.

Clark, D. S. (1987). Short proof of a discrete Gronwall inequality. *Discrete Appl. Math.*, **16**(3), 279–281.

Cockayne, J., Oates, C., Sullivan, T., and Girolami, M. (2017). Probabilistic numerical methods for PDE-constrained Bayesian inverse problems. *AIP Conference Proceedings*, **1853**(1), 060001.

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

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

Cranmer, K., Brehmer, J., and Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*.

Crisan, D. and Doucet, A. (2002). A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, **50**(3), 736–746.

Dahlquist, G. G. (1963). A special stability problem for linear multistep methods. *BIT Numerical Mathematics*, **3**(1), 27–43.

Davis, H. T. (1962). *Introduction to Nonlinear Differential and Integral Equations*. Dover Publications, New York.

Deisenroth, M. (2009). *Efficient Reinforcement Learning Using Gaussian Processes*. Ph.D. thesis, Karlsruhe Institute of Technology.

Deisenroth, M. and Rasmussen, C. (2011). PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *International Conference on Machine Learning (ICML)*.

Del Moral, P. (2004). *Feynman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer.

Deuflhard, P. and Bornemann, F. (2002). *Scientific Computing with Ordinary Differential Equations*. Springer.

Diaconis, P. (1988). Bayesian numerical analysis. *Statistical decision theory and related topics*, **IV**(1), 163–175.

Doucet, A. and Tadić, V. B. (2003). Parameter estimation in general state-space models using particle methods. *Annals of the Institute of Statistical Mathematics*, **55**(2), 409–422.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, **10**(3), 197–208.

Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer.

Euler, L. (1768). *Institutionum Calculi Integralis*, volume 1. Impensis Academiae Imperialis Scientiarum.

Fehlberg, E. (1969). Low-order classical runge-kutta formulas with stepsize control and their application to some heat transfer problems. Technical Report NASA Technical Report R-315, George C. Marshall Space Flight Center, Huntsville, Alabama.

Fragoso, T., Bertoli, W., and Louzada, F. (2018). Bayesian model averaging: A systematic review and conceptual classification. *International Statistical Review*, **86**, 1–28.

Gal, Y. (2016). *Uncertainty in Deep Learning*. Ph.D. thesis, University of Cambridge.

Garcia-Fernandez, A. F., Svensson, L., Morelande, M. R., and Särkkä, S. (2015). Posterior linearization filter: Principles and implementation using sigma points. *IEEE Transactions on Signal Processing*, **63**(20), 5561–5573.

Gessner, A., Gonzalez, J., and Mahsereci, M. (2019). Active multi-information source Bayesian quadrature. In *Uncertainty in Artificial Intelligence (UAI)*.

Ghahramani, Z. (2013). Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **371**(1984), 20110553.

Girolami, M. (2020). Introducing data-centric engineering: An open access journal dedicated to the transformation of engineering design and practice. *Data-Centric Engineering*, **1**, e1.

Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. Roy. Stat. Soc. Ser. B*, **73**(2), 123–214.

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

Golub, G. H. and Welsch, J. H. (1969). Calculation of Gauss quadrature rules. *Mathematics of computation*, **23**(106), 221–230.

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

Grathwohl, W., Chen, R. T. Q., Bettencourt, J., and Duvenaud, D. (2019). Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*.

Grewal, M. S. and Andrews, A. P. (2001). *Kalman Filtering: Theory and Practice Using MATLAB*. John Wiley & Sons, Inc.

Hairer, E. and Wanner, G. (1996). *Solving Ordinary Differential Equations II – Stiff and Differential-Algebraic Problems*. Springer, 2nd edition.

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

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

Hauberg, S. (2018). Only Bayes should learn a manifold (on the estimation of differential geometric structure from data). *arXiv:1806.04994 [stat.ML]*.

Hennig, P. and Hauberg, S. (2014). Probabilistic solutions to differential equations and their application to Riemannian statistics. In *Proc. of the 17th int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, volume 33. JMLR, W&CP.

Hennig, P., Osborne, M. A., and Girolami, M. (2015). Probabilistic numerics and uncertainty in computations. *Proc. Roy. Soc. London A*, **471**(2179), 20150142.

Hoare, C. A. R. (1961). Algorithm 64: Quicksort. *Commun. ACM*, **4**(7), 321.

Hochbruck, M., Ostermann, A., and Schweitzer, J. (2009). Exponential Rosenbrock-type methods. *SIAM Journal on Numerical Analysis*, **47**(1), 786–803.

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

Ionides, E. L., Bhadra, A., Atchadé, Y., King, A., *et al.* (2011). Iterated filtering. *The Annals of Statistics*, **39**(3), 1776–1802.

Jazwinski, A. (1970). *Stochastic Processes and Filtering Theory*. Academic Press.

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

Jordan, M. (2018). Artificial intelligence—the revolution hasn't happened yet. `https://medium.com/@mijordan3/artificial-intelligence-the-revolution-hasnt-happened-yet-5e1d5812e1e7`. Accessed: 2020-09-12.

Julier, S., Uhlmann, J., and Durrant-Whyte, H. F. (2000). A new method for the nonlinear transformation of means and covariances in filters and estimators. *IEEE Transactions on automatic control*, **45**(3), 477–482.

Kantas, N., Doucet, A., Singh, S. S., and Maciejowski, J. M. (2009). An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. *IFAC Proceedings Volumes*, **42**(10), 774–785.

Karatzas, I. and Shreve, S. (1991). *Brownian Motion and Stochastic Calculus*. Springer.

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

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kennedy, M. C. and O'Hagan, A. (2002). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B*, **63**(3).

Kersting, H. and Hennig, P. (2016). Active uncertainty calibration in Bayesian ODE solvers. *Uncertainty in Artificial Intelligence (UAI)*.

Kersting, H. and Mahsereci, M. (2020). A Fourier state space model for Bayesian ODE filters. In *Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models, ICML*.

Kersting, H., Sullivan, T. J., and Hennig, P. (2020a). Convergence rates of Gaussian ODE filters. *Stat. Comput.*, **30**(6), 1791–1816.

Kersting, H., Krämer, N., Schiegg, M., Daniel, C., Tiemann, M., and Hennig, P. (2020b). Differentiable likelihoods for fast inversion of 'likelihood-free' dynamical systems. In *International Conference on Machine Learning (ICML)*.

Kimeldorf, G. S. and Wahba, G. (1970). A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Ann. Math. Statist.*, **41**(2).

Kokkala, J., Solin, A., and Särkkä, S. (2014). Expectation maximization based parameter estimation by sigma-point and particle smoothing. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–8. IEEE.

Kutta, W. (1901). Beitrag zur näherungsweisen Integration totaler Differentialgleichungen. *Zeitschrift für Mathematik und Physik*, **46**, 435–453.

Lancaster, P. and Rodman, L. (1995). *Algebraic Riccati Equations*. Oxford Science Publications.

Law, K., Stuart, A., and Zygalakis, K. (2015). *Data Assimilation: A Mathematical Introduction*, volume 62 of *Texts in Applied Mathematics*. Springer, Cham.

Lie, H. C., Stuart, A. M., and Sullivan, T. J. (2019). Strong convergence rates of probabilistic integrators for ordinary differential equations. *Stat. Comput.*, **29**(6), 1265–1283.

Lindsten, F. (2013). An efficient stochastic approximation EM algorithm using conditional particle filters. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*, pages 6274–6278. IEEE.

Lindström, E., Ionides, E., Frydendall, J., and Madsen, H. (2012). Efficient iterated filtering. *IFAC Proceedings Volumes*, **45**(16), 1785–1790.

Lindström, E., Madsen, H., and Nielsen, J. N. (2015). *Statistics for Finance*. Chapman and Hall/CRC.

Loscalzo, F. R. and Talbot, T. D. (1967). Spline function approximations for solutions of ordinary differential equations. *SIAM J. Numer. Anal.*, **4**, 433–445.

Lotka, A. (1978). The growth of mixed populations: two species competing for a common food supply. *The Golden Age of Theoretical Ecology: 1923—1940*, **22**.

Macdonald, B. and Husmeier, D. (2015). Gradient matching methods for computational inference in mechanistic models for systems biology: A review and comparative analysis. *Frontiers in bioengineering and biotechnology*, **3**, 180.

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

Magnani, E., Kersting, H., Schober, M., and Hennig, P. (2017). Bayesian filtering for ODEs with bounded derivatives. *arXiv:1709.08471 [cs.NA]*.

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

Maybeck, P. S. (1979). *Stochastic Models, Estimation, and Control*. Academic Press.

McNamee, J. and Stenger, F. (1967). Construction of fully symmetric numerical integration formulas. *Numerische Mathematik*, **10**(4), 327–344.

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

Mockus, J. B. and Mockus, L. J. (1991). Bayesian approach to global optimization and application to multiobjective and constrained problems. *J. Optim. Theory Appl.*, **70**(1), 157—-172.

Mohammadi, H., Challenor, P., and Goodfellow, M. (2019). Emulating dynamic non-linear simulators using Gaussian processes. *Computational Statistics and Data Analysis*.

Newton, I. (1671). Methodus fluxionum et serierum infinitarum. *Opuscula mathematica*.

Nordsieck, A. (1962). On numerical integration of ordinary differential equations. *Math. Comp.*, **16**, 22–49.

Novak, E. and Wozniakowski, H. (2010). *Tractability of multivariate problems. Volume II: Standard information for functionals*. European Mathematical Society (EMS).

Oates, C., Cockayne, J., Aykroyd, R., and Girolami, M. (2019). Bayesian probabilistic numerical methods in time-dependent state estimation for industrial hydrocyclone equipment. *Journal of the American Statistical Association*, **114**(528), 1518–1531.

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

O'Hagan, A. (1991). Bayes–Hermite quadrature. *J. Statist. Plann. Inference*, **29**(3), 245–260.

O'Hagan, A. (1992). Some Bayesian numerical analysis. In *Bayesian statistics, 4 (Peñíscola, 1991)*, pages 345–363. Oxford Univ. Press, New York.

O'Hagan, A. (2006). Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering & System Safety*, **91**(10–11), 1290–1300.

Øksendal, B. (2003). *Stochastic Differential Equations: An Introduction with Applications*. Springer, 5th edition.

Osborne, M., Garnett, R., Ghahramani, Z., Duvenaud, D. K., Roberts, S. J., and Rasmussen, C. E. (2012). Active learning of model evidence using bayesian quadrature. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 46–54.

Paul, S., Chatzilygeroudis, K., Ciosek, K., Mouret, J.-B., Osborne, M. A., and Whiteson, S. (2018). Alternating optimisation and quadrature for robust control. In *AAAI Conference on Artificial Intelligence*.

Pearl, J. (2018). To build truly intelligent machines, teach them cause and effect. `https://www.quantamagazine.org/to-build-truly-intelligent-machines-teach-them-cause-and-effect-20180515/`. Accessed: 2020-09-12.

Perdikaris, P. and Karniadakis, G. E. (2016). Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond. *Journal of the Royal Society Interface*, **13**(118), 20151107.

Poincaré, H. (1896). *Calcul des probabilités*. Gauthier-Villars, Paris.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3rd edition.

Prüher, J. and Šimandl, M. (2015). Bayesian quadrature in nonlinear filtering. In *12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 01, pages 380–387.

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

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

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*.

Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT.

Rastrigin, L. A. (1963). The convergence of the random search method in the extremal control of a many parameter system. *Automation and Remote Control*, **24**(10), 1337–1342.

Reich, S. and Cotter, C. (2015). *Probabilistic Forecasting and Bayesian Data Assimilation*. Cambridge University Press, New York.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International Conference on Machine Learning (ICML)*.

Ritter, K. (2000). *Average-Case Analysis of Numerical Problems*, volume 1733 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin.

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

Rosenbrock, H. H. (1963). Some general implicit processes for the numerical solution of differential equations. *The Computer Journal*, **5**(4), 329–330.

Runge, C. (1895). Über die numerische Auflösung von Differentialgleichungen. *Mathematische Annalen*, **46**, 167–178.

Saatci, Y. (2011). *Scalable Inference for Structured Gaussian Process Models*. Ph.D. thesis, University of Cambridge.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, **4**(4), 409–423.

Saitoh, S. and Sawano, Y. (2016). *Theory of Reproducing Kernels and Applications*. Springer.

Särkkä, S. (2006). *Recursive Bayesian Inference on Stochastic Differential Equations*. Ph.D. thesis, Helsinki University of Technology.

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

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

Särkkä, S., Hartikainen, J., Svensson, L., and Sandblom, F. (2016). On the relation between gaussian process quadratures and sigma-point methods. *Journal of Advances in Information Fusion*, **11**(1), 31–46.

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

Schillings, C., Sunnaker, M., Stelling, J., and Schwab, C. (2015). Efficient characterization of parametric uncertainty of complex (bio)chemical networks. *PLOS Computational Biology*, **11**(8), 1–16.

Schober, M. (2018). *Probabilistic Ordinary Differential Equation Solvers—Theory and Applications*. Ph.D. thesis, Eberhard Karls Universität Tübingen.

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

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

Schön, T. B., Wills, A., and Ninness, B. (2011). System identification of nonlinear state-space models. *Automatica*, **47**(1), 39–49.

Schwab, K. (2017). *The Fourth Industrial Revolution*. Crown Publishing Group.

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

Scieur, D., Roulet, V., d'Aspremont, A., and Bach, F. (2017). Integration methods and accelerated optimization algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Segal, M. R. (1992). Tree-structured methods for longitudinal data. *Journal of the American Statistical Association*, **87**(418), 407–418.

Seo, S., Wallat, M., Graepel, T., and Obermayer, K. (2000). Gaussian process regression: Active data selection and test point rejection. *Mustererkennung*.

Skilling, J. (1991). Bayesian solutions of ordinary differential equations. *Maximum Entropy and Bayesian Methods, Seattle*.

Soares, N. and Fallenstein, B. (2014). Questions of reasoning under logical uncertainty. Technical report, Machine Intelligence Research Institute, Berkeley, CA.

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

Solin, A. and Särkkä, S. (2014). Explicit link between periodic covariance functions and state space models. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 904–912. PMLR.

Spitzbart, A. (1960). A generalization of Hermite's interpolation formula. *The American Mathematical Monthly*, **67**(1), 42–46.

Storvik, G. (2002). Particle filters for state-space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, **50**(2), 281–289.

Sullivan, T. J. (2015). *Introduction to Uncertainty Quantification*. Springer.

Taniguchi, A., Fujimoto, K., and Nishida, Y. (2017). On variational Bayes for identification of nonlinear state-space models with linearly dependent unknown parameters. In *56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), 2017*, pages 572–576. IEEE.

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

Teschl, G. (2012). *Ordinary Differential Equations and Dynamical Systems*. American Mathematical Society.

Teymur, O., Zygalakis, K., and Calderhead, B. (2016). Probabilistic linear multistep methods. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4314–4321. Curran Associates, Inc.

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

Tronarp, F., Garcia-Fernandez, A. F., and Särkkä, S. (2018). Iterative filtering and smoothing in non-linear and non-Gaussian systems using conditional moments. *IEEE Signal Processing Letters*, **25**(3), 408–412.

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

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

Tronarp, F., Särkkä, S., and Hennig, P. (2020). Bayesian ODE solvers: The maximum a posteriori estimate. *arXiv:2004.00623 [math.NA]*.

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

Wang, J., Cockayne, J., and Oates, C. (2018). On the Bayesian solution of differential equations. *Proceedings of the 38th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering*.

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

Xiaoyue, X., Briol, F.-X., and Girolami, M. (2018). Bayesian quadrature for multiple related integrals. In *International Conference on Machine Learning (ICML)*.

Zhang, J., Mokhtari, A., Sra, S., and Jadbabaie, A. (2018). Direct Runge–Kutta discretization achieves acceleration. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Appendix: Publications

A Active Uncertainty Calibration in Bayesian ODE Solvers (Kersting and Hennig, 2016)

Abstract: There is resurging interest, in statistics and machine learning, in solvers for ordinary differential equations (ODEs) that return probability measures instead of point estimates. Recently, Conrad et al. introduced a sampling-based class of methods that are 'well-calibrated' in a specific sense. But the computational cost of these methods is significantly above that of classic methods. On the other hand, Schober et al. pointed out a precise connection between classic Runge–Kutta ODE solvers and Gaussian filters, which gives only a rough probabilistic calibration, but at negligible cost overhead. By formulating the solution of ODEs as approximate inference in linear Gaussian SDEs, we investigate a range of probabilistic ODE solvers, that bridge the trade-off between computational cost and probabilistic calibration, and identify the inaccurate gradient measurement as a crucial source of uncertainty. We propose the novel filtering-based method Bayesian Quadrature filtering (BQF) which uses Bayesian quadrature to actively learn the imprecision in the gradient measurement by collecting multiple gradient evaluations.

A.1 Introduction

The numerical solution of an initial value problem (IVP) based on an *ordinary differential equation* (ODE)

$$u^{(n)}(t) = f\left(t, u(t), \dots, u^{(n-1)}(t)\right), \; u(0) = u_0 \in \mathbb{R}^D, \tag{A.1}$$

of order $n \in \mathbb{N}$, with $u : \mathbb{R} \to \mathbb{R}^D$, $f : [0, T] \times \mathbb{R}^{nD} \to \mathbb{R}^D$, $T > 0$, is an essential topic of numerical mathematics, because ODEs are the standard model for dynamical systems. Solving ODEs with initial values is an exceedingly well-studied problem (see Hairer *et al.*, 1987, for a comprehensive presentation) and modern solvers are designed very efficiently. Usually, the original ODE (A.1) of order n is reduced to a system of n ODEs of first order

$$u'(t) = f(t, u(t)), \; u(0) = u_0 \in \mathbb{R}^D, \tag{A.2}$$

which are solved individually. The most popular solvers in practice are based on some form of Runge–Kutta (RK) method (as first introduced in Runge (1895) and Kutta (1901)) which employ a weighted sum of a fixed amount of gradients in order to iteratively extrapolate a discretized solution. That is, these methods collect 'observations' of approximate gradients of the solved ODE, by evaluating the vector field f at an estimated solution, which is a linear combination of previously collected 'observations':

$$y_i = f\left(t + c_i h, u_0 + \sum_{j<i} w_{ij} y_j\right). \tag{A.3}$$

The final extrapolation step is a weighted sum of these gradients:

$$\hat{u}(t+h) = u(t) + \sum_{i<s} b_i y_i. \tag{A.4}$$

The weights of s-stage RK methods of p-th order are carefully chosen so that the numerical approximation $\hat{u}$ and the Taylor series of the exact solution u coincide up to the term h^p, thereby yielding a local truncation error of high polynomial order,

$$\|u(t_0 + h) - \hat{u}(t_0 + h)\| = \mathcal{O}(h^{p+1}), \tag{A.5}$$

for $h \to 0$. One can prove that $s \geq p$ in general, but for $p \leq 4$ there are RK methods with $p = s$. Hence, allowing for more function evaluations can drastically improve the speed of convergence to the exact solution.

The polynomial convergence is impressive and helpful; but it does not actually quantify the inevitable epistemic uncertainty over the accuracy of the approximate solution $\hat{u}$ for a concrete non-vanishing step-size h. One reason one may be concerned about this in machine learning is that ODEs are often one link of a chain of algorithms performing some statistical analysis. When employing classic ODE solvers and just plugging in the solution of the numerical methods in subsequent steps, the resulting uncertainty of the whole computation is ill-founded, resulting in overconfidence in a possibly wrong solution. It is thus desirable to model the epistemic uncertainty. Probability theory provides the framework to do so. Meaningful probability measures of the uncertainty about the result of deterministic computations (such as ODE solvers) can then be combined with probability measures modeling other sources of uncertainty, including 'real' aleatoric randomness (from e.g. sampling). Apart from quantifying our certainty over a computation, pinning down the main sources of uncertainty could furthermore improve the numerical solution and facilitate a more efficient allocation of the limited computational budget.

A closed framework to measure uncertainty over numerical computations was proposed by Skilling (1991) who pointed out that numerical methods can be recast as statistical inference of the latent exact solution based on the observable results of tractable computations. In this spirit, Hennig and Hauberg (2014) phrased this notion more formally, as

Gaussian process (GP) regression. Their algorithm class, however, could not guarantee the high polynomial convergence orders of Runge–Kutta methods. In parallel development, Chkrebtii *et al.* (2016) also introduced a probabilistic ODE solver of similar structure (i.e. based on a GP model), but using a Monte Carlo updating scheme. These authors showed a linear convergence rate of their solver, but again not the high-order convergence of classic solvers.
Recently, Schober *et al.* (2014) solved this problem by finding prior covariance functions which produce GP ODE solvers whose posterior means *exactly* match those of the optimal Runge–Kutta families of first, second and third order. While producing only a slight computational overhead compared to classic Runge–Kutta, this algorithm—as any GP-based algorithm—only returns Gaussian measures over the solution space.
In contrast, Conrad *et al.* (2017) recently provided a novel sampling-based class of ODE solvers which returns flexible non-Gaussian measures over the solution space, but creates significant computational overhead by running the whole classic ODE solvers multiple times over the whole time interval $[0, T]$ in order to obtain meaningful approximations for the desired measure.
For practitioners, there is a trade-off between the desire for quantified uncertainty on the one hand, and low computational cost on the other. The currently available probabilistic solvers for ODEs either provide only a roughly calibrated uncertainty (Schober *et al.*, 2014) at negligible overhead or a more fine-grained uncertainty supported by theoretical analysis (Conrad *et al.*, 2017), at a computational cost increase so high that it rules out most practical applications. In an attempt to remedy this problem, we propose an algorithm enhancing the method of Schober *et al.* (2014) by improving the gradient measurement using modern probabilistic integration methods. By modeling the uncertainty where it arises, i.e. the imprecise prediction of where to evaluate f, we hope to gain better knowledge of the propagated uncertainty and arrive at well-calibrated posterior variances as uncertainty measures.

A.2 Background

A.2.1 Sampling-based ODE solvers

The probabilistic ODE solver by Conrad *et al.* (2017) modifies a classic deterministic one-step numerical integrator Ψ_h (e.g. Runge–Kutta or multiderivative methods, cf. Hairer *et al.* (1987)) and models the discretization error of Ψ_h by adding suitably scaled i.i.d. Gaussian random variables $\{\xi_k\}_{k=0,\dots,K}$ after every step. Hence, it returns a discrete solution $\{U_k\}_{k=0,\dots,K}$ on a mesh $\{t_k = kh\}_{k=0,\dots,K}$ according to the rule

$$U_{k+1} = \Psi_h(U_k) + \xi_k. \tag{A.6}$$

This discrete solution can be extended into a continuous time approximation of the ODE, which is random by construction and can therefore be interpreted as a draw from a non-

parametric probability measure Q_h on the solution space $C^1([0,T],\mathbb{R}^n)$, the Banach space of continuously differentiable functions. This probability measure can then be interpreted as a notion of epistemic uncertainty about the solution. This is correct in so far as, under suitable assumptions, including a bound on the variance of the Gaussian noise, the method converges to the exact solution, in the sense that Q_h contracts to the Dirac measure on the exact solution δ_u with the *same* convergence rate as the original numerical integrator Ψ_h, for $h \to 0$: If $(\xi_{k,h})_{k=1}^N \sim \mathcal{N}(0, \mathrm{Var}(h))$ with $\mathrm{Var}(h) = \mathcal{O}(h^{2q+1})$, then

$$\sup_{0 \le kh \le T} \mathbb{E}^h \left\| u_k - U_k \right\|^2 \le \sigma \cdot h^{2q}. \tag{A.7}$$

This is a significant step towards a well-founded notion of uncertainty calibration for ODE solvers: It provides a probabilistic extension to classic method which does not break the convergence rate of these methods.

In practice, however, the precise shape of Q_h is *not* known and Q_h can only be interrogated by sampling, i.e. repeatedly running the entire probabilistic solver. After S samples, Q_h can be approximated by an empirical measure $Q_h(S)$. In particular, the estimated solution and uncertainty can only be expressed in terms of statistics of $Q_h(S)$, e.g. by the usual choices of the empirical mean and empirical variance respectively or alternatively by confidence intervals. For $S \to \infty$, $Q_h(S)$ converges in distribution to Q_h which again converges in distribution to δ_u for $h \to 0$:

$$Q_h(S) \overset{S \to \infty}{\to} Q_h \overset{h \to 0}{\to} \delta_u. \tag{A.8}$$

The theoretical mathematics in Conrad *et al.* (2017) only concerns the convergence of the latent probability measures $\{Q_h\}_{h>0}$. Only the empirical measures $\{Q_h(S)\}_{S \in \mathbb{N}}$, however, can be observed. Consequently, it remains unclear whether the empirical mean of $Q_h(S)$ for a fixed step-size $h > 0$ converges to the exact solution as $S \to \infty$ and whether the empirical variance of $Q_h(S)$ is directly related, in an analytical sense, to the approximation error. In order to extend the given convergence results to the practically observable measures $\{Q_h(S)\}_{S \in \mathbb{N}}$ an analysis of the first convergence in (A.8) remains missing. The deterministic algorithm proposed below avoids this problem, by instead constructing a (locally parametric) measure from prior assumptions.

The computational cost of this method also seems to mainly depend on the rate of convergence of $Q_h(S) \to Q_h$ which determines how many (possibly expensive) runs of the numerical integrator Ψ_h over $[0,T]$ have to be computed and how many samples have to be stored for a sufficient approximation of Q_h. Furthermore, we expect that in practice the mean of Q_h, as approximated by $Q_h(S)$ might not be the best possible approximation, since in one step the random perturbation of the predicted solution by Gaussian noise ξ_k worsens our solution estimate with a probability of more than $1/2$, since—due to the full support of Gaussian distributions—the numerical sample solution is as likely to be perturbed away from as towards the exact solution and—due to the tails of Gaussian distributions—it can also be perturbed way past the exact solution

with positive probability.

A.2.2 A framework for Gaussian filtering for ODEs

Describing the solution of ODEs as inference in a joint Gaussian model leverages state-space structure to achieve efficient inference. Therefore, we employ a Gauss–Markov prior on the state-space: *A priori* we model the solution function and $(q-1)$ derivatives $(u, \dot{u}, u^{(2)}, \ldots, u^{(q-1)}) : [0,T] \to \mathbb{R}^{qD}$ as a draw from a q-times integrated Wiener process $X = (X_t)_{t\in[0,T]} = (X_t^{(1)}, \ldots, X_t^{(q)})_{t\in[0,T]}^T$, i.e. the dynamics of X_t are given by the linear stochastic differential equation (Karatzas and Shreve, 1991; Øksendal, 2003):

$$dX_t = FX_t dt + LdW_t, \tag{A.9}$$

$$X_0 = \xi, \quad \xi \sim \mathcal{N}(m(0), P(0)), \tag{A.10}$$

with constant drift $F \in \mathbb{R}^{q\times q}$ and diffusion $L \in \mathbb{R}^q$ given by

$$F = \begin{pmatrix} 0 & f_1 & 0 & \ldots & 0 \\ 0 & 0 & f_2 & \ldots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & & \ldots & 0 & f_{q-1} \\ 0 & & \ldots & 0 & 0 \end{pmatrix}, \; L = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \sigma \end{pmatrix} \tag{A.11}$$

for all $t \in [0,T]$ and some $f_1, \ldots, f_{q-1} \in \mathbb{R}$, where W_t denotes a q-dimensional Wiener process ($q \geq n$). Hence, we are a priori expecting that $u^{(q)}$ behaves like a Brownian motion with variance σ^2 and that $u^{(i)}$ is modeled by $(q-1-i)$-times integrating this Brownian motion. The fact that the $(i+1)$-th component is the derivative of the i-th component in our state space is captured by a drift matrix with non-zero entries only on the first off-diagonal. The entries $f_1, \ldots, f_{q-1}$ are damping factors. A standard choice is e.g. $f_i = i$. Without additional information, it seems natural to put white noise on the q-th derivative as the first derivative which is not captured in the state space. This gives rise to Brownian noise on the $(q-1)$-th derivative which is encoded in the diffusion matrix scaled by variance σ^2. Hence, we consider the integrated Wiener process a natural prior. For notational simplicity, only the case of scalar-valued functions, i.e. $D = 1$, is presented in the following. The framework can be extended to $D \geq 2$ in a straightforward way by modeling the output dimensions of f as independent stochastic processes.

Since X is the strong solution of a linear equation (A.9) with normally distributed initial value X_0, it follows from the theory of linear SDEs (Karatzas and Shreve, 1991) that X is a uniquely-determined Gauss–Markov process. This enables Bayesian inference in a highly efficient way by Gaussian filtering (Saatci, 2011)). For time invariant linear SDEs like (A.9), the fixed matrices for Gaussian filtering can be precomputed analytically (Särkkä, 2006).

In addition, Schober *et al.* (2014) showed that for $q \leq 3$ inference in this linear SDE

yields Runge–Kutta steps.

Equipped with this advantageous prior we can perform Bayesian inference. The linearity and time-invariance of the underlying SDE permits to formulate the computation of the posterior as a Kalman filter (KF) (cf. (Särkkä, 2013) for a comprehensive introduction) with step size $h > 0$. The prediction step of the KF is given by

$$m_{t+h}^{-} = A(h)m_t, \tag{A.12}$$
$$P_{t+h}^{-} = A(h)P_t A(h)^T + Q(h), \tag{A.13}$$

with matrices $A(h), Q(h) \in \mathbb{R}^{q\times q}$ with entries

$$\begin{aligned} A(h)_{i,j} =& \exp(hF)_{i,j} = \chi_{j\geq i}\frac{h^{j-i}}{(j-i)!}\left(\prod_{k=0}^{j-i-1} f_{i+k}\right), \\ Q(h)_{i,j} =& \sigma^2 \left(\prod_{k_1=0}^{q-1-i} f_{i+k_1}\right)\cdot\left(\prod_{k_2=0}^{q-1-j} f_{j+k_2}\right)\cdot \\ & \frac{h^{2q+1-i-j}}{(q-i)!(q-j)!(2q+1-i-j)}. \end{aligned} \tag{A.14}$$

It is followed by the update step

$$z = y - Hm_{t+h}^{-}, \tag{A.15}$$
$$S = HP_{t+h}^{-}H^T + R, \tag{A.16}$$
$$K = P_{t+h}^{-}H^T S^{-1}, \tag{A.17}$$
$$m_{t+h} = m_{t+h}^{-} + Kz, \tag{A.18}$$
$$P_{t+h} = P_{t+h}^{-} - KHP_{t+h}^{-}, \tag{A.19}$$

where $H = e_n^T \in \mathbb{R}^{1\times q}$ is the n-th unit vector.

Between the prediction and update step the n-th derivative of the exact solution $\frac{\partial^n u}{\partial x^n}$ at time $t+h$ as a measurement for the n-th derivative and the noise of this measurement are estimated by the variable y and R. In order to derive precise values of y and R from the Gaussian prediction $\mathcal{N}(m_{t+h}, P_{t+h})$, we would have to compute the integrals

$$y = \int f(t+h, m_{t+h}^{-} + x)\mathcal{N}(x; 0, P_{t+h}^{-})\ \mathrm{d}x \tag{A.20}$$

and

$$\begin{aligned} R = \int & f(t+h, m_{t+h}^{-} + x)f(t+h, m_{t+h}^{-} + x)^T \cdot \\ & \mathcal{N}(x; 0, P_{t+h}^{-})\ \mathrm{d}x - yy^T, \end{aligned} \tag{A.21}$$

which are intractable for most choices of f. Below we investigate different ways to address the challenge of accurately approximating these integrals while not creating too much computational overhead.

A.2.3 Measurement generation options for Gaussian filtering

Schober *et al.* (2014) as, to the best of our knowledge, the first ones to point out the connection between Gaussian filtering and probabilistic ODE solvers, presents an algorithm which simply evaluates the gradient at the predicted mean, which is equivalent to setting y to be equal to its maximum likelihood estimator:

$$y = f(t+h, m^-_{t+h}),\ R = 0. \tag{A.22}$$

While ensuring maximum speed, this is clearly not an ideal measurement. In our atomless predicted probability measure $\mathcal{N}(m^-_{t+h}, P^-_{t+h})$ the mean predictor m^-_{t+h} is different from its exact value $(u^{(0)}(t+h), \ldots, u^{(n)}(t+h))^T$ almost surely. Hence, for a non-constant f the estimate will be inaccurate most of the times. In particular this method deals poorly with 'skewed' gradient fields (a problem that leads to a phenomenon known as 'Lady Windermeres fan' (Hairer *et al.*, 1987)). To get a better estimate of the exact value of y, more evaluations of f seem necessary.

Therefore, we want to find numerical integration methods which capture y and R with sufficient precision, while using a minimal number of evaluations of f. Possible choices are:

(i) *Monte Carlo integration by sampling*:

$$y = \frac{1}{N}\sum_{i=1}^{N} f(t+h, x_i), \tag{A.23}$$

$$R = \frac{1}{N}\sum_{i=1}^{N} f(t+h, x_i) f(t+h, x_i)^T - yy^T, \tag{A.24}$$

$$x_i \sim \mathcal{N}(m^-_{t+h}, P^-_{t+h}), \tag{A.25}$$

(which is *not* the same as the sampling over the whole time axis in (Conrad *et al.*, 2017)).

(ii) *Approximation by a first-order Taylor series expansion:*

$$\begin{aligned} &f(t+h, m^-_{t+h} + x) \\ &\quad \simeq f(t+h, m^-_{t+h}) + \nabla f(t+h, m^-_{t+h} + x) \cdot x \end{aligned} \tag{A.26}$$

and thereby deriving moments of the linear transform of Gaussian distributions:

$$y = f(t+h, m^-_{t+h}), \tag{A.27}$$
$$R = \nabla f(t+h, m^-_{t+h}) P^-_{t+h} \nabla f(t+h, m^-_{t+h})^T. \tag{A.28}$$

(iii) Integration by *Bayesian quadrature* with Gaussian weight function:

$$y = \alpha^T K^{-1} \left(f(x_1), \ldots, f(x_n) \right)^T, \tag{A.29}$$
$$R = \int\int k(x, x') w(x) w(x') \, \mathrm{d}x \mathrm{d}x' - \alpha^T K^{-1} \alpha. \tag{A.30}$$

with $w(x) = \mathcal{N}(x; m^-_{t+h}, P^-_{t+h})$, kernel matrix $K \in \mathbb{R}^{N \times N}$ with $K_{i,j} = k(x_i, x_j)$ and $\alpha = (\alpha(1), \ldots, \alpha(N))^T \in \mathbb{R}^N$ with $\alpha(i) = \int k(x, x_i) w(x) \, \mathrm{d}x$ for a predefined covariance function k and evaluation points $(x_i)_{i=1,\ldots,N}$ (cf. section A.2.4).

Our experiments, presented in Section A.3, suggest that BQ is the most useful option.

Monte Carlo integration by sampling behaves poorly if the trajectory of the numerical solution passes through domain areas (as e.g. in the spikes of oscillators governed by non-stiff ODEs) where f takes highly volatile values since the random spread of samples from the domain are likely to return a skewed spread of values resulting in bad predictions of y with huge uncertainty R. Hence, the posterior variance explodes and the mean drifts back to its zero prior mean, i.e. $m_t \to 0$ and $\|P_t\| \to \infty$, for $t \to \infty$. Thus, we consider this method practically useless.

One may consider it a serious downside of Taylor-approximation based methods that the gradient only approximates the shape of f and thereby its mapping of the error on an 'infinitesimally small neighborhood' of m^-_{t+h}. Hence, it might ignore the exact value of y completely, if the mean prediction is far off. However, for a highly regular f (e.g. Lipschitz-continuous in the space variable) this gradient approximation is very good.

Moreover, the approximation by a first-order Taylor series expansion needs an approximation of the gradient, which explicit ODE solvers usually do not receive as an input. However, in many numerical algorithms (e.g. optimization) the gradient is provided anyway. Therefore the gradient might already be known in real-world applications. While we find this method promising when the gradient is known or can be efficiently computed, we exclude it from our experiments because the necessity of a gradient estimate breaks the usual framework of ODE solvers.

In contrast, Bayesian quadrature avoids the risk of a skewed distortion of the samples for Monte Carlo integration by actively spreading a grid of deterministic sigma-points. It does not need the gradient of f and still can encode prior knowledge over f by the choice of the covariance function if more is known (Briol *et al.*, 2019). The potential of using Bayesian quadrature as a part of a filter was further explored by Prüher and Šimandl (2015), however in the less structured setting of nonlinear filtering where additional

inaccuracy from the linear approximation in the prediction step arises. Moreover, Särkkä *et al.* (2016) recently pointed out that BQ can be seen as sigma-point methods and gave covariance functions and evaluation points which reproduce numerical integration methods known for their favorable behavior (for example Gauss–Hermite quadrature, which is used for a Gaussian weight function).

Due to these advantages, we propose a new class of BQ-based probabilistic ODE filters named BQ Filtering.

A.2.4 Bayesian quadrature filtering

The crucial source of error for filtering-based ODE solvers is the calculation of the gradient measurement y and its variance R (c.f. Section A.2.2). We propose the novel approach to use BQ to account for the uncertainty of the input and thereby estimate y and R. This gives rise a novel class of filtering-based solvers named *BQ Filter* (BQF). As a filtering-based method, one BQF-step consists of the KF prediction step (A.12)–(A.13), the calculation of y and R by BQ and the KF update step (A.15)–(A.19).

The KF prediction step outputs a Gaussian belief $\mathcal{N}(m^-_{t+h}, P^-_{t+h})$ over the exact solution $u(t+h)$. This input value is propagated through f yielding a distribution over the gradient at time $t+h$. In other words, our belief over $\nabla f(t+h, u(t+h))$ is equal to the distribution of $Y := f(t, X)$, with uncertain input $X \sim \mathcal{N}(m^-_{t+h}, P^-_{t+h})$. For general f the distribution of Y will be neither Gaussian nor unimodal (as e.g. in Figure A.1). But it is possible to compute the moments of this distribution under Gaussian assumptions on the input and the uncertainty over f (see for example Deisenroth (2009)). The equivalent formulation of prediction under uncertainty clarifies as numerical integration clarifies the connection to sigma-point methods, i.e. quadrature rules (Särkkä *et al.*, 2016). Quadrature is as extensively studied and well-understood as the solution of ODEs. A basic overview can be found in Press *et al.* (2007). Marginalizing over X yields an integral with Gaussian weight function

$$\mathbb{E}[Y] = \int f(t+h, x) \underbrace{\mathcal{N}(x; m^-_{t+h}, P^-_{t+h})}_{=:w(x)} \, \mathrm{d}x, \tag{A.31}$$

which is classically solved by quadrature, i.e. evaluating f at a number of evaluation points $(x_i)_{i=1,\dots,N}$ and calculating a weighted sum of these evaluations. BQ can be interpreted as a probabilistic extension of these quadrature rules in the sense that their posterior mean estimate of the integral coincides with classic quadrature rules, while adding a posterior variance estimate at low cost (Särkkä *et al.*, 2016).

By choosing a kernel k over the input space of f and evaluation points $(x_i)_{i=1,\dots,N}$, the function f is approximated by a GP regression (Rasmussen and Williams, 2006) with respect to the function evaluations $(f(x_i))_{i=1,\dots,N}$, yielding a GP posterior over f with mean m_f and covariance k_f denoted by $\mathcal{GP}(f)$. The integral is then approximated by

integrating the GP approximation, yielding the predictive distribution for $\mathcal{I}[f]$:

$$\mathcal{I}[f] \sim \int \mathcal{GP}(f)(x) \cdot \mathcal{N}(x; m^-_{t+h}, P^-_{t+h}) \, \mathrm{d}x. \tag{A.32}$$

The uncertainty arising from the probability measure over the input is now split up in two parts: the uncertainty over the input value $x \sim \mathcal{N}(0, I)$ and the uncertainty over the precise value at this uncertain input, which can only be approximately inferred by its covariance with the evaluation points $(x_i)_{i=1,\dots,N}$, i.e. by $\mathcal{GP}(f)$. These two kinds of uncertainty are depicted in Figure A.1. From the predictive distribution in (A.32), we

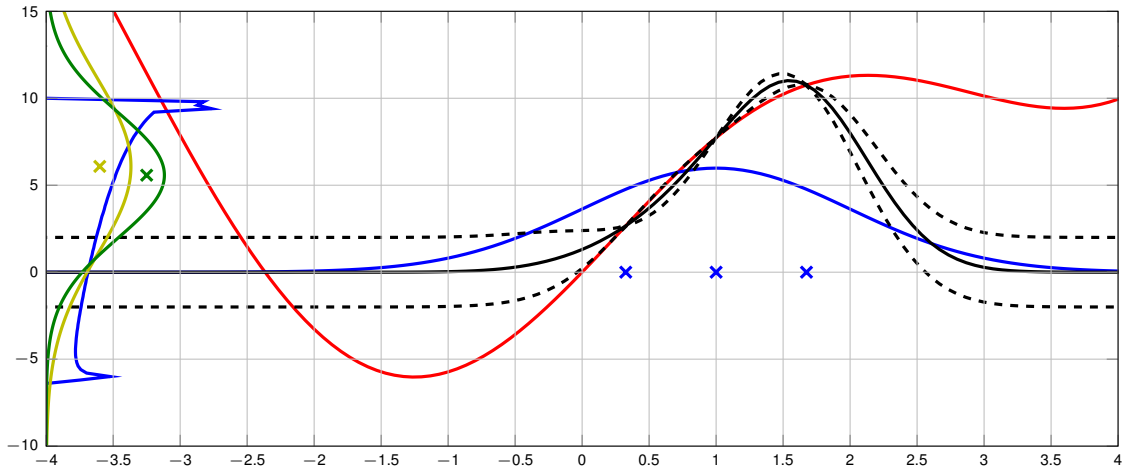


Figure A.1: Prediction of function $f(x) = 8\sin(x) + x^2$ (red) under uncertain input $x \sim \mathcal{N}(x; 1, 1)$ (density in blue). $\mathcal{GP}(f)$ (black) derived from Gaussian grid evaluation points with $N = 3$ (blue crosses) as mean $\pm$ 2 standard deviation. True distribution of prediction in blue. Gaussian fit to true distribution in yellow and predicted distribution by BQ in green with crosses at means.

can now compute a posterior mean and variance of $\mathcal{I}[f]$ which results in a weighted sum for the mean

$$y := \mathbb{E}\left[\mathcal{I}[f]\right] = \alpha^T K^{-1} \Big(f(x_1), \dots, f(x_n)\Big)^T \tag{A.33}$$

with

$$\alpha(i) = \int k(x, x_i) \mathcal{N}(x; 0, I) \, \mathrm{d}x \tag{A.34}$$

and variance

$$\begin{aligned} R &:= \mathrm{Var}\left[\mathcal{I}(f)\right] \\ &= \int\int k(x,x')w(x)w(x')\ \mathrm{d}x\mathrm{d}x' - \alpha^T K^{-1}\alpha, \end{aligned} \tag{A.35}$$

where $K \in \mathbb{R}^{N\times N}$ denotes the kernel matrix, i.e. $K_{i,j} = k(x_i, x_j)$.

The measurement generation in BQF is hence completely defined by the two free choices of BQ: the kernel k and the evaluation points $(x_i)_{i=1,\dots,n}$. By these choices, BQ and thereby the measurement generation in BQF is completely defined. For the squared exponential kernel (Rasmussen and Williams, 2006)

$$k(x,x') = \theta^2 \exp\left(-\frac{1}{2\lambda^2}\|x-x'\|^2\right), \tag{A.36}$$

with lengthscale $\lambda > 0$ and output variance $\theta^2 > 0$, it turns out that y and R can be computed in closed form and that many classic quadrature methods which are known for their favorable properties can be computed in closed form (Särkkä *et al.*, 2016), significantly speeding up computations. For the scalar case $nD = 1$, we obtain for (A.34) by straightforward computations:

$$\alpha(i) = \frac{\lambda\theta^2}{\sqrt{\lambda^2+\sigma^2}} \exp\left(-\frac{(x_i-\mu)^2}{2(\lambda^2+\sigma^2)}\right), \tag{A.37}$$

and

$$\int\int k(x,x')w(x)w(x')\ \mathrm{d}x\mathrm{d}x' = \frac{\theta^2}{\sqrt{1+2\sigma^2/\lambda^2}} \tag{A.38}$$

Hence, our BQ estimate for y is given by the sigma-point rule

$$y \approx \sum_{i=1}^{N} W_i f(t+h, x_i) \tag{A.39}$$

with easily computable weights

$$W_i = [\alpha^T K^{-1}]_i. \tag{A.40}$$

Also the variance R takes a convenient shape

$$R = \frac{\theta^2}{\sqrt{1+2\sigma^2/\lambda^2}} - \alpha^T K^{-1}\alpha. \tag{A.41}$$

For $nD > 1$, we get slightly more complicated formulas which are given in Deisenroth

(2009).

The other free choice in BQ, the evaluation points $(x_i)_{i=1,\dots,n}$, can also be chosen freely in every step of BQF. Usually, the nodes of BQ chosen are chosen so that the variance of the integral estimate is minimized (cf. Briol *et al.* (2019)). For this algorithm, the uncertainty has to be measured, not minimized though. Hence, we propose just to take a uniform grid scaled by $\mathcal{N}(m^-_{t+h}, P^-_{t+h})$ to measure the uncertainty in a comprehensive way.

Another promising choice is given by the roots of the physicists' version of the Hermite polynomials, since they yield Gauss–Hermite quadrature (GHQ), the standard numerical quadrature against Gaussian measures, as a posterior mean for a suitable covariance function (Särkkä *et al.*, 2016). For GHQ, efficient algorithms to compute the roots and the weights are readily available (Press *et al.*, 2007).

A.2.5 Computational cost

All of the presented algorithms buy their probabilistic extension to classic ODE solvers by adding computational cost, sometimes more sometimes less. In most cases, evaluation of the vector field f forms the computational bottleneck, so we will focus on it here. Of course, the internal computations of the solver adds cost as well. Since all the models discussed here have linear inference cost, though, this additional overhead is manageable. The ML-algorithm by Schober *et al.* (2014) is the fastest algorithm. By simply recasting a Runge–Kutta step as Gaussian filtering, rough probabilistic uncertainty is achieved with negligible computational overhead.
For the sampling method, the calculation of one individual sample of Q_h amounts to running the entire underlying ODE solver once, hence the overall cost is S times the original cost.
In contrast, the BQ-algorithm only has to run through $[0, T]$ once, but has to invert a $ND \times ND$ covariance matrix to perform Bayesian quadrature with N evaluation points. Usually, N will be small, since BQ performs well for a relatively small number of function evaluations (as e.g. illustrated by the experiments below). However, if the output dimension D is very large, Bayesian quadrature—like all quadrature methods—is not practical. BQ thus tends to be faster for small D, while MC tends to be faster for large D.

When considering these computational overheads, there is a nuanced point to be made about the value-to-cost trade-off of constructing a posterior uncertainty measure. If a classic numerical solver of order p is allotted a budget of M times its original one, it can use it to reduce its step-size by a factor of M, and thus reduce its approximation error by an order M^p. It may thus seem pointless to invest even such a linear cost increase into constructing an uncertainty measure around the classic estimate. But, in some practical settings, it may be more helpful to have a notion of uncertainty on a slightly less precise estimate than to produce a more precise estimate without a notion of error. In addition, classic solvers are by nature sequential algorithms, while the probabilistic

extensions (both the sampling-based and Gaussian-filtering based ones) can be easily parallelized. Where parallel hardware is available, the effective time cost of probabilistic functionality may thus be quite limited (although we do not investigate this possibility in our present experiments).

With regards to memory requirements, the MC-method needs significantly more storage, since it requires saving all sample paths, in order to statistically approximate the entire non-parametric measure Q_h on $C^1([0,T],\mathbb{R})$. The BQ-algorithm only has to save the posterior GP, i.e. a mean and a covariance function, which is arguably the minimal amount to provide a notion of uncertainty. If MC reduces the approximation of Q_h to its mean and variance, it only requires this minimal storage as well.

A.3 Experiments

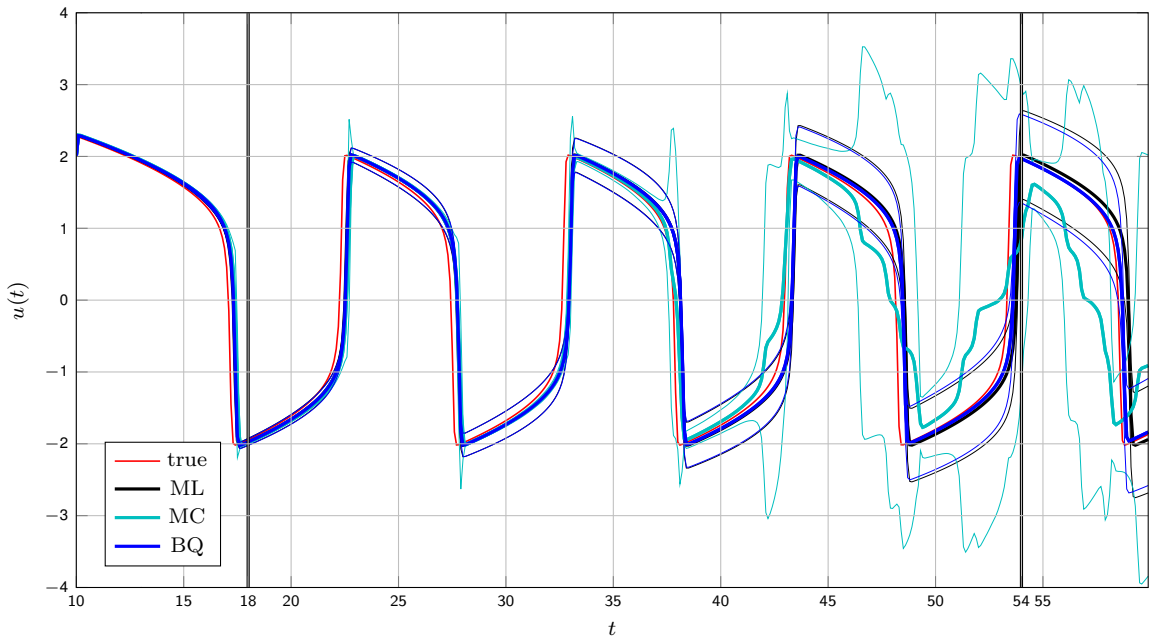


Figure A.2: Solution estimates constructed on the Van der Pol oscillator (A.42). True solution in red. Mean estimates of ML, MC and BQ in black, green, blue, respectively. Uncertainty measures (drawn at two times standard deviation) as thin lines of the same color.

This section explores applications of the probabilistic ODE solvers discussed in Section A.2. The sampling-based algorithm by (Conrad *et al.*, 2017) will be abbreviated as MC, the maximum-likelihood Gaussian filter ((Schober *et al.*, 2014)) as ML and our novel BQ-based filter (BQF) as BQ. In particular, we assess how the performance of the purely

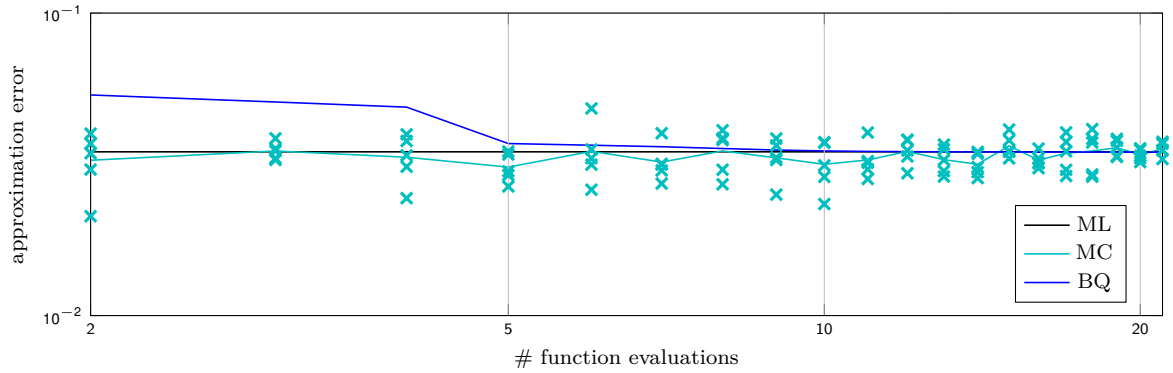


Figure A.3: Plot of errors of the mean estimates at $t = 18$ of the methods MC (green) and BQ (blue) as a function of the allowed function evaluations. Maximum likelihood error in black. Single runs of the probabilistic MC solver as green crosses. Average over all runs as green line.

deterministic class of Gaussian filtering based solvers compares to the inherently random class of sampling-based solvers.

We experiment on the Van der Pol oscillator (Hairer *et al.*, 1987), a non-conservative oscillator with non-linear damping, which is a standard example for a non-stiff dynamical system. It is governed by the equation

$$\frac{\partial^2 u}{\partial t^2} = \mu(1 - u^2)\frac{\partial u}{\partial t} - u, \tag{A.42}$$

where the parameter $\mu \in \mathbb{R}$ indicates the non-linearity and the strength of the damping. We set $\mu = 5$ on a time axis $[10, 60]$, with initial values $(u(10), \dot{u}(10)) = (2, 10)$.

All compared methods use a model of order $q = 3$, and a step size $h = 0.01$. This induces a state-space model given by a twice-integrated Wiener process prior (cf. (A.9)) which yields a version of ML close to second-order Runge–Kutta (Schober *et al.*, 2014). The same solver is used as the underlying numerical solver Ψ_h in MC. For the noise parameter, which scales the deviation of the evaluation point of f from the numerical extrapolation (i.e. the variance of the driving Wiener process for ML and BQ, and the variance of ξ_k for MC), we choose $\sigma^2 = 0.1$. The drift matrix F of the underlying integrated Wiener process is set to the default values $f_i = i$ for $i = 1, \ldots, q-1$. The covariance function used in BQ is the widely popular squared exponential (A.36), with lengthscale $\lambda = 1$ and output variance $\theta^2 = 1$. (Since all methods use the same model, this tuning does not favor one algorithm over the other. In practice all these parameters should of course be set by statistical estimation.).

For a fair comparison in all experiments, we allow MC and BQ to make the same amount of function evaluations per time step. If MC draws N samples, BQ uses N evaluation points. The first experiment presents the solutions of the presented algorithms

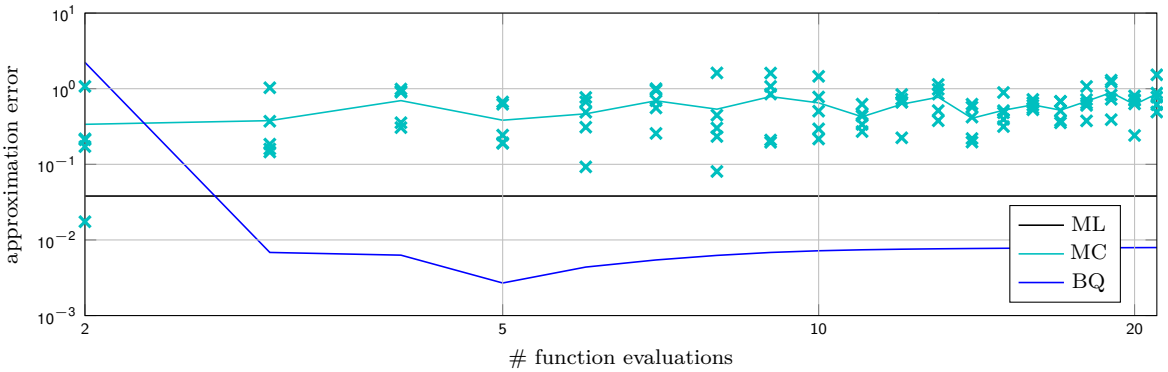


Figure A.4: Plot of errors of the mean estimates at $t = 54$ of the methods MC (green) and BQ (blue) as a function of the allowed function evaluations. Maximum likelihood error in black. Single runs of the probabilistic MC solver as green crosses. Average over all runs as green line.

on the van der Pol oscillator (A.42) on the whole time axis in one plot, when we allow BQ and MC to make five function evaluations. Then, we examine more closely how the error of each methods changes as a function of the number of evaluations of f in Figure A.3 and Figure A.4.

A.3.1 Solution measures on Van Der Pol oscillator

Figure A.2 shows the solution estimates constructed by the three solvers across the time domain. In all cases, the mean estimates roughly follow the exact solution (which e.g. Gaussian filtering with Monte Carlo integration by sampling (A.23)–(A.25) does not achieve). A fundamental difference between the filtering-based methods (ML and BQ) and the sampling-based MC algorithm is evident in both the mean and the uncertainty estimate.

While the filtering-based methods output a trajectory quite similar to the exact solution with a small time lag, the MC algorithm produces a trajectory of a more varying shape. Characteristic points of the MC mean estimate (such as local extrema) are placed further away from the exact value than for filtering-based methods.

The uncertainty estimation of MC appears more flexible as well. ML and BQ produce an uncertainty estimate which runs parallel to the mean estimate and appears to be strictly increasing. It appears to increase slightly in every step, resulting in an uncertainty estimate, which only changes very slowly. The solver accordingly appears overconfident in the spikes and underconfident in the valleys of the trajectory. The uncertainty of MC varies more, scaling up at the steep parts of the oscillator and decreasing again at the flat parts, which is a desirable feature.

Among the class of filtering-based solvers, the more refined BQ method outputs a better mean estimate with more confidence than ML.

A.3.2 Quality of estimate as a function of allowed evaluations

Figure A.3 and Figure A.4 depict the value of the error of the mean approximation as a function of the allowed function evaluations N (i.e. N evaluation points for BQ and N samples for MC) at time points $t_1 = 18$ and $t_2 = 54$. Since the desired solution measure Q_h for MC can only be statistically approximated by the N samples, the mean estimate of MC is random. For comparison, the average of five MC-runs is computed.

At the early time point $t_1 = 18$, all trajectories are still close together and the methods perform roughly the same, as we allow more evaluations. There is a slight improvement for BQ with more evaluations, but the error remains above the one of ML error.

At the later time $t_2 = 54$, BQ improves drastically when at least five evaluations are allowed, dropping much below the ML error.

The average error by MC appears to be not affected by the number of samples. The ML error is constant, because it always evaluates only once.

A.4 Discussion

The conducted experiments provide an interesting basis to discuss the differences between filtering-based methods (ML and BQ) and the sampling-based MC algorithm. We make the following observations:

(i) *Additional samples do not improve the random mean estimate of MC in expectation:*
Since the samples of MC are independent and identically distributed, the expectation of the random mean estimate of MC is the same, regardless of the amounts of samples. This property is reflected in Figure A.3 and Figure A.4, by the constant green line (up to random fluctuation). Additional samples are therefore only useful to improve the uncertainty calibration.

(ii) *The uncertainty calibration of MC appears more adaptive than of ML and BQ:*
Figure A.2 suggests that MC captures the uncertainty more flexibly: It appropriately scales up in the steep parts of the oscillator, while expressing high confidence in the flat parts of the oscillator. The exact trajectory is inside the interval between mean $\pm$ 2 standard deviations, which is not the case for BQ and ML. Moreover, MC produces a more versatile measure. The filtering-based methods appear to produce a strictly increasing uncertainty measure by adding to the posterior uncertainty in every step. MC avoids this problem by sampling multiple time over the whole time interval. We deem the resulting flexibility a highly desirable feature. BQ also outputs a meaningful uncertainty measure and we expect that adding Bayesian smoothing (Särkkä, 2013) would enable filtering-based methods to produce more adaptive measures as well.

(iii) *The expected error of MC-samples (and their mean) is higher than the error of ML:*
In the experiments, MC produced a higher error for the mean estimate, compared to both ML and BQ. We expect that this happens on all dynamical systems *by construction*: Given U_k, the next value U_{k+1} of a MC-sample is calculated by adding Gaussian noise ξ_k to the ML-extrapolation starting in U_k (cf. equation (A.6)). Due to the symmetry and full support of Gaussian distributions, the perturbed solution has a higher error than the unperturbed prediction, which coincides with the ML solution. Hence, every MC-sample accumulates with every step a positive expected error increment compared to the ML estimate. By the linearity of the average, the mean over all samples inherits the same higher error than the ML mean (and thereby also than the error of the more refined BQ mean).

Summing up, we argue that—at their current state—filtering-based methods appear to produce a 'better' mean estimate, while sampling-based methods produce in some sense a 'better' uncertainty estimate. Many applications might put emphasis on a good mean estimate, while needing a still well-calibrated uncertainty quantification. Our method BQF provides a way of combining a precise mean estimate with a meaningful uncertainty calibration. Sampling-based methods might not be able to provide this due to their less accurate mean estimate. For future work (which is beyond the scope of this paper), it could be possible to combine the advantages of both approaches in a unified method.

A.5 Conclusion

We have presented theory and methods for the probabilistic solution of ODEs which provide uncertainty measures over the solution of the ODE, contrasting the classes of (deterministic) filtering-based and (random) sampling-based solvers. We have provided a theoretical framework for Gaussian filtering as state space inference in linear Gaussian SDEs, highlighting the prediction of the gradient as the primary source of uncertainty. Of all investigated approximations of the gradient, Bayesian Quadrature (BQ) produces the best results, by actively learning the shape of the vector field f through deterministic evaluations. Hence, we propose a novel filtering-based method named *Bayesian Quadrature Filtering* (BQF), which employs BQ for the gradient measurement.

For the same amount of allowed gradient evaluations, the mean estimate of BQF appears to outperform the mean estimate of state-of-the-art sampling-based solvers on the Van der Pol oscillator, while outputting a better calibrated uncertainty than other filtering-based methods.

B Probabilistic Solutions to Ordinary Differential Equations as Non-Linear Bayesian Filtering: A New Perspective (Tronarp *et al.*, 2019a)

Abstract: We formulate probabilistic numerical approximations to solutions of ordinary differential equations (ODEs) as problems in Gaussian process (GP) regression with non-linear measurement functions. This is achieved by defining the measurement sequence to consist of the observations of the difference between the derivative of the GP and the vector field evaluated at the GP—which are all identically zero at the solution of the ODE. When the GP has a state-space representation, the problem can be reduced to a non-linear Bayesian filtering problem and all widely-used approximations to the Bayesian filtering and smoothing problems become applicable. Furthermore, all previous GP-based ODE solvers that are formulated in terms of generating synthetic measurements of the gradient field come out as specific approximations. Based on the non-linear Bayesian filtering problem posed in this paper, we develop novel Gaussian solvers for which we establish favourable stability properties. Additionally, non-Gaussian approximations to the filtering problem are derived by the particle filter approach. The resulting solvers are compared with other probabilistic solvers in illustrative experiments.

B.1 Introduction

We consider an initial value problem (IVP), that is, an ordinary differential equation (ODE)

$$\dot{y}(t) = f\left(y(t), t\right), \ \forall t \in [0, T], \quad y(0) = y_0 \in \mathbb{R}^d, \tag{B.1}$$

with initial value y_0 and vector field $f : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$. Numerical solvers for IVPs approximate $y : [0, T] \to \mathbb{R}^d$ and are of paramount importance in almost all areas of science and engineering. Extensive knowledge about this topic has been accumulated in numerical analysis literature, for example, in Hairer *et al.* (1987), Deuflhard and Bornemann (2002), and Butcher (2008). However, until recently, a probabilistic quantification

of the inevitable uncertainty–for all but the most trivial ODEs–from the numerical error over their outputs has been omitted.

Moreover, ODEs are often part of a pipeline surrounded by preceding and subsequent computations, which are themselves corrupted by uncertainty from model misspecification, measurement noise, approximate inference or, again, numerical inaccuracy (Kennedy and O'Hagan, 2002). In particular, ODEs are often integrated using estimates of its parameters rather than the correct ones. See Zhang *et al.* (2018) and Chen *et al.* (2018) for recent examples of such computational chains involving ODEs. The field of *probabilistic numerics* (PN) (Hennig *et al.*, 2015) seeks to overcome this ignorance of numerical uncertainty and the resulting overconfidence by providing *probabilistic numerical methods*. These solvers quantify numerical errors probabilistically and add them to uncertainty from other sources. Thereby, they can take decisions in a more uncertainty-aware and uncertainty-robust manner (Paul *et al.*, 2018).

In the case of ODEs, one family of probabilistic solvers (Skilling (1991), Hennig and Hauberg (2014), and Schober *et al.* (2014)) first treated IVPs as Gaussian process (GP) regression (Rasmussen and Williams, 2006, Chapter 2). Then, Kersting and Hennig (2016) and Schober *et al.* (2019) sped up these methods by regarding them as stochastic filtering problems (Øksendal, 2003). These completely deterministic filtering methods converge to the true solution with high polynomial rates (Kersting *et al.*, 2020a). In their methods data for the 'Bayesian update' is constructed by evaluating the vector field f under the GP predictive mean of $y(t)$ and linked to the model with a Gaussian likelihood (Schober *et al.*, 2019, Section 2.3). See also Wang *et al.* (2018, Section 1.2) for alternative likelihood models. This conception of data implies that it is the output of the adopted inference procedure. More specifically, one can show that with everything else being equal, two different priors may end up operating on different measurement sequences. Such a coupling between prior and measurements is not standard in statistical problem formulations, as acknowledged in Schober *et al.* (2019, Section 2.2). It makes the model and the subsequent inference difficult to interpret. For example, it is not clear how to do Bayesian model comparisons (Cockayne *et al.*, 2019, Section 2.4) when two different priors necessarily operate on two different data sets for the same inference task.

Instead of formulating the solution of Eq. (B.1) as a Bayesian GP regression problem, another line of work on probabilistic solvers for ODEs comprising the methods from Chkrebtii *et al.* (2016), Conrad *et al.* (2017), Teymur *et al.* (2016), Lie *et al.* (2019), Abdulle and Garegnani (2020), and Teymur *et al.* (2018) aims to represent the uncertainty arising from the discretization error by a set of samples. While multiplying the computational cost of classical solvers with the amount of samples, these methods can capture arbitrary (non-Gaussian) distributions over the solutions and can reduce over-confidence in inverse problems for ODEs—as demonstrated in Conrad *et al.* (2017, Section 3.2.), Abdulle and Garegnani (2020, Section 7), and Teymur *et al.* (2018). These solvers can be considered as more expensive, but statistically more expressive. This paper contributes a particle filter as a sampling-based filtering method at the intersection of both lines of work, providing a previously missing link.

The contributions of this paper are the following: Firstly, we circumvent the issue of generating synthetic data, by recasting solutions of ODEs in terms of non-linear Bayesian filtering problems in a well defined state-space model. For any fixed-time discretisation, the measurement sequence and likelihood are also fixed. That is, we avoid the coupling of prior and measurement sequence, that is for example present in Schober *et al.* (2019). This enables application of all Bayesian filtering and smoothing techniques to ODEs as described, for example, in Särkkä (2013). Secondly, we show how the application of certain inference techniques recovers the previous filtering-based methods. Thirdly, we discuss novel algorithms giving rise to both Gaussian and non-Gaussian solvers.

Fourthly, we establish a stability result for the novel Gaussian solvers. Fifthly, we discuss practical methods for uncertainty calibration, and in the case of Gaussian solvers, we give explicit expressions. Finally, we present some illustrative experiments demonstrating that these methods are practically useful both for fast inference of the unique solution of an ODE as well as for representing multi-modal distributions of trajectories.

B.2 Bayesian inference for initial value problems

Formulating an approximation of the solution to Eq. (B.1) at a discrete set of points $\{t_n\}_{n=0}^N$ as a problem of Bayesian inference requires, as always, three things: a prior measure, data, and a likelihood, which define a posterior measure through Bayes' rule.

We start with examining a continuous-time formulation in Section B.2.1, where Bayesian conditioning should, in the ideal case, give a Dirac measure at the true solution of Eq. (B.1) as the posterior. This has two issues: (1) conditioning on the entire gradient field is not feasible on a computer in finite time and (2) the conditioning operation itself is intractable. Issue (1) is present in classical Bayesian quadrature (Briol *et al.*, 2019) as well. Limited computational resources imply that only a finite number of evaluations of the integrand can be used. Issue (2) turns, what is linear GP regression in Bayesian quadrature, into non-linear GP regression. While this is unfortunate, it appears reasonable that something should be lost as the inference problem is more complex.

With this in mind, a discrete-time non-linear Bayesian filtering problem is posed in Section B.2.2, which targets the solution of Eq. (B.1) at a discrete set of points.

B.2.1 A continuous-time model

Like previous works mentioned in Section B.1, we consider priors given by a GP

$$X(t) \sim \mathrm{GP}\left(\bar{x}, k\right),$$

where $\bar{x}(t)$ is the mean function and $k(t, t')$ is the covariance function. The vector $X(t)$ is given by

$$X(t) = \left[\left(X^{(1)}(t)\right)^\intercal, \ldots, \left(X^{(q+1)}(t)\right)^\intercal\right]^\intercal, \tag{B.2}$$

where $X^{(1)}(t)$ and $X^{(2)}(t)$ model $y(t)$ and $\dot{y}(t)$, respectively. The remaining $q-1$ sub-vectors in $X(t)$ can be used to model higher order derivatives of $y(t)$ as done by Schober *et al.* (2019) and Kersting and Hennig (2016). We define such priors by a stochastic differential equation (Øksendal, 2003), that is,

$$X(0) \sim \mathcal{N}\Big(\mu^{-}(0), \Sigma^{-}(0)\Big), \tag{B.3a}$$

$$\mathrm{d}X(t) = \Big[FX(t) + u\Big]\,\mathrm{d}t + L\,\mathrm{d}B(t), \tag{B.3b}$$

where F is a state transition matrix, u is a forcing term, L is a diffusion matrix, and $B(t)$ is a vector of standard Wiener processes.

Note that for $X^{(2)}(t)$ to be the derivative of $X^{(1)}$, F, u, and L are such that

$$\mathrm{d}X^{(1)}(t) = X^{(2)}(t)\,\mathrm{d}t. \tag{B.4}$$

The use of an SDE—instead of a generic GP prior—is computationally advantageous because it restricts the priors to Markov processes due to Øksendal (2003, Theorem 7.1.2). This allows for inference with linear time-complexity in N, while the time-complexity is N^3 for GP priors in general (Hartikainen and Särkkä, 2010).

Inference requires data, and an associated likelihood. Previous authors, such as Schober *et al.* (2019) and Chkrebtii *et al.* (2016), put forth the view of the prior measure defining an *inference agent*, which cycles through extrapolating, generating measurements of the vector field, and updating. Here we argue that there is no need for generating measurements, since re-writing Eq. (B.1) yields the requirement

$$\dot{y}(t) - f(y(t), t) = 0. \tag{B.5}$$

This suggests that a measurement relating the prior defined by Eq. (B.3) to the solution of Eq. (B.1) ought to be defined as

$$Z(t) = X^{(2)}(t) - f(X^{(1)}(t), t). \tag{B.6}$$

While conditioning the process $X(t)$ on the event $Z(t) = 0$ for all $t \in [0, T]$ can be formalised using the concept of *disintegration* (Cockayne *et al.*, 2019), it is intractable in general and thus impractical for computer implementation. Therefore, we formulate a discrete-time inference problem in the sequel.

B.2.2 A discrete-time model

In order to make the inference problem tractable, we only attempt to condition the process $X(t)$ on $Z(t) = z(t) \triangleq 0$ at a set of discrete time-points, $\{t_n\}_{n=0}^{N}$. We consider a uniform grid, $t_{n+1} = t_n + h$, though extending the present methods to non-uniform grids can be done as described in Schober *et al.* (2019). In the sequel, we will denote

a function evaluated at t_n by subscript n, for example $z_n = z(t_n)$. From Eq. (B.3) an equivalent discrete-time system can be obtained (Grewal and Andrews, 2001, Chapter 3.7.3) [1]. The inference problem becomes

$$X_0 \sim \mathcal{N}(\mu_0^F, \Sigma_0^F), \tag{B.7a}$$

$$X_{n+1} \mid X_n \sim \mathcal{N}\Big(A(h)X_n + \xi(h), Q(h)\Big), \tag{B.7b}$$

$$Z_n \mid X_n \sim \mathcal{N}\Big(\dot{C}X_n - f(CX_n, t_n), R\Big), \tag{B.7c}$$

$$z_n \triangleq 0, \quad n = 1, \ldots, N, \tag{B.7d}$$

where z_n is the realisation of Z_n. The parameters $A(h)$, $\xi(h)$, and $Q(h)$ are given by

$$A(h) = \exp(Fh), \tag{B.8a}$$

$$\xi(h) = \int_0^h \exp(F(h-\tau))u \,\mathrm{d}\tau, \tag{B.8b}$$

$$Q(h) = \int_0^h \exp(F(h-\tau))LL^{\mathsf{T}} \exp(F^{\mathsf{T}}(h-\tau)) \,\mathrm{d}\tau. \tag{B.8c}$$

Furthermore, $C = [\mathrm{I}\ 0\ \ldots\ 0]$ and $\dot{C} = [0\ \mathrm{I}\ 0\ \ldots\ 0]$. That is, $CX_n = X_n^{(1)}$ and $\dot{C}X_n = X_n^{(2)}$. A measurement variance, R, has been added to $Z(t_n)$ for greater generality, which simplifies the construction of particle filter algorithms. The likelihood model in Eq. (B.7c) has previously been used in the gradient matching approach to inverse problems to avoid explicit numerical integration of the ODE (see, e.g., Calderhead *et al.* (2008)).

The inference problem posed in Eq. (B.7) is a standard problem in non-linear GP regression (Rasmussen and Williams, 2006), also known as Bayesian filtering and smoothing in stochastic signal processing (Särkkä, 2013). Furthermore, it reduces to Bayesian quadrature when the vector field does not depend on y. This is Proposition B.2.1 below.

Proposition B.2.1. *Let $X_0^{(1)} = 0$, $f(y(t), t) = g(t)$, $y(0) = 0$, and $R = 0$. Then the posteriors of $\{X_n^{(1)}\}_{n=1}^N$ are Bayesian quadrature approximations for*

$$\int_0^{nh} g(\tau) \,\mathrm{d}\tau, \quad n = 1, \ldots, N. \tag{B.9}$$

A proof of Proposition B.2.1 is given in Appendix B.7.

Remark B.2.2. *The Bayesian quadrature method described in Proposition B.2.1 conditions on function evaluations outside the domain of integration for $n < N$. This corresponds to the smoothing equations associated with Eq.* (B.7)*. If the integral on the domain $[0, nh]$ is only conditioned on evaluations of g inside the domain then the filtering estimates associated with Eq.* (B.7) *are obtained.*

[1] Here 'equivalent' is used in the sense that the probability distribution of the continuous-time process evaluated on the grid coincides with the probability distribution of the discrete-time process (Särkkä, 2006, Page 17).

B.2.3 Gaussian filtering

The inference problem posed in Eq. (B.7) is a standard problem in statistical signal processing and machine learning, and the solution is often approximated by Gaussian filters and smoothers (Särkkä, 2013). Let us define $z_{1:n} = \{z_l\}_{l=1}^n$ and the following conditional moments

$$\mu_n^F \triangleq \mathbb{E}[X_n \mid z_{1:n}], \tag{B.10a}$$
$$\Sigma_n^F \triangleq \mathbb{V}[X_n \mid z_{1:n}], \tag{B.10b}$$
$$\mu_n^P \triangleq \mathbb{E}[X_n \mid z_{1:n-1}], \tag{B.10c}$$
$$\Sigma_n^P \triangleq \mathbb{V}[X_n \mid z_{1:n-1}], \tag{B.10d}$$

where $\mathbb{E}[\cdot \mid z_{1:n}]$ and $\mathbb{V}[\cdot \mid z_{1:n}]$ are the conditional mean and covariance operators given the measurements $Z_{1:n} = z_{1:n}$. Additionally, $\mathbb{E}[\cdot \mid z_{1:0}] = \mathbb{E}[\cdot\,]$ and $\mathbb{V}[\cdot \mid z_{1:0}] = \mathbb{V}[\cdot\,]$ by convention. Furthermore, μ_n^F and Σ_n^F are referred to as the filtering mean and covariance, respectively. Similarly, μ_n^P and Σ_n^P are referred to as the predictive mean and covariance, respectively. In Gaussian filtering, the following relationships hold between μ_n^F and Σ_n^F, and μ_{n+1}^P and Σ_{n+1}^P:

$$\mu_{n+1}^P = A(h)\mu_n^F + \xi(h), \tag{B.11a}$$
$$\Sigma_{n+1}^P = A(h)\Sigma_n^F A^\intercal(h) + Q(h), \tag{B.11b}$$

which are the prediction equations (Särkkä, 2013, Eq. 6.6). The update equations, relating the predictive moments μ_n^P and Σ_n^P with the filter estimate, μ_n^F, and its covariance Σ_n^F, are given by (Särkkä, 2013, Eq. 6.7)

$$S_n = \mathbb{V}\Big[\dot{C}X_n - f(CX_n, t_n) \mid z_{1:n-1}\Big] + R, \tag{B.12a}$$
$$K_n = \mathbb{C}\Big[X_n, \dot{C}X_n - f(CX_n, t_n) \mid z_{1:n-1}\Big]S_n^{-1}, \tag{B.12b}$$
$$\hat{z}_n = \mathbb{E}\Big[\dot{C}X_n - f(CX_n, t_n) \mid z_{1:n-1}\Big], \tag{B.12c}$$
$$\mu_n^F = \mu_n^P + K_n(z_n - \hat{z}_n), \tag{B.12d}$$
$$\Sigma_n^F = \Sigma_n^P - K_n S_n K_n^\intercal, \tag{B.12e}$$

where the expectation ($\mathbb{E}$), covariance ($\mathbb{V}$) and cross-covariance ($\mathbb{C}$) operators are with respect to $X_n \sim \mathcal{N}(\mu_n^P, \Sigma_n^P)$. Evaluating these moments is intractable in general, though various approximation schemes exist in literature. Some standard approximation methods shall be examined below. In particular, the methods of Schober *et al.* (2019) and Kersting and Hennig (2016) come out as particular approximations to Eq. (B.12).

B.2.4 Taylor-series methods

A classical method in filtering literature to deal with non-linear measurements of the form in Eq. (B.7) is to make a first order Taylor-series expansion, thus turning the problem into a standard update in linear filtering. However, before going through the details of this it is instructive to interpret the method of Schober *et al.* (2019) as an even simpler Taylor-series method. This is Proposition B.2.3 below.

Proposition B.2.3. *Let $R = 0$ and approximate $f(CX_n, t_n)$ by its zeroth order Taylor expansion in X_n around the point μ_n^P*

$$f\Big(CX_n, t_n\Big) \approx f\Big(C\mu_n^P, t_n\Big). \tag{B.13}$$

Then, the approximate posterior moments are given by

$$\begin{aligned}
S_n &\approx \dot{C}\Sigma_n^P\dot{C}^\mathsf{T} + R, && \text{(B.14a)}\\
K_n &\approx \Sigma_n^P\dot{C}^\mathsf{T}S_n^{-1}, && \text{(B.14b)}\\
\hat{z}_n &\approx \dot{C}\mu_n^P - f\Big(C\mu_n^P, t_n\Big), && \text{(B.14c)}\\
\mu_n^F &\approx \mu_n^P + K_n(z_n - \hat{z}_n), && \text{(B.14d)}\\
\Sigma_n^F &\approx \Sigma_n^P - K_nS_nK_n^\mathsf{T}, && \text{(B.14e)}
\end{aligned}$$

which is precisely the update by Schober et al. *(2019).*

A First Order Approximation. The approximation in Eq. (B.14) can be refined by using a first order approximation, which is known as the extended Kalman filter (EKF) in signal processing literature (Särkkä, 2013, Algorithm 5.4). That is,

$$\begin{aligned}
f\Big(CX_n, t_n\Big) \approx\ & f\Big(C\mu_n^P, t_n\Big)\\
&+ J_f\Big(C\mu_n^P, t_n\Big)C\Big(X_n - \mu_n^P\Big),
\end{aligned} \tag{B.15}$$

where J_f is the Jacobian of $y \to f(y, t)$. The filter update is then

$$\begin{aligned}
\tilde{C}_n &= \dot{C} - J_f\Big(C\mu_n^P, t_n\Big)C, && \text{(B.16a)}\\
S_n &\approx \tilde{C}_n\Sigma_n^P\tilde{C}_n^\mathsf{T} + R, && \text{(B.16b)}\\
K_n &\approx \Sigma_n^P\tilde{C}_n^\mathsf{T}S_n^{-1}, && \text{(B.16c)}\\
\hat{z}_n &\approx \dot{C}\mu_n^P - f\Big(C\mu_n^P, t_n\Big), && \text{(B.16d)}\\
\mu_n^F &\approx \mu_n^P + K_n(z_n - \hat{z}_n), && \text{(B.16e)}\\
\Sigma_n^F &\approx \Sigma_n^P - K_nS_nK_n^\mathsf{T}. && \text{(B.16f)}
\end{aligned}$$

Hence the extended Kalman filter computes the residual, $z_n - \hat{z}_n$, in the same manner as Schober *et al.* (2019). However, as the filter gain, K_n, now depends on evaluations of the Jacobian, the resulting probabilistic ODE solver is different in general.

While Jacobians of the vector field are seldom exploited in ODE solvers, they play a central role in Rosenbrock methods, (Rosenbrock (1963) and Hochbruck *et al.* (2009)). The Jacobian of the vector field was also recently used by Teymur *et al.* (2018) for developing a probabilistic solver.

Although the extended Kalman filter goes as far back as the 1960s (Jazwinski, 1970), the update in Eq. (B.16) results in a probabilistic method for estimating the solution of (B.1) that appears to be novel. Indeed, to the best of the authors' knowledge, the only Gaussian filtering based solvers that have appeared so far are those by Kersting and Hennig (2016), Magnani *et al.* (2017), and Schober *et al.* (2019).

B.2.5 Numerical quadrature

Another method to approximate the quantities in Eq. (B.12) is by quadrature, which consists of a set of nodes $\{\mathcal{X}_{n,j}\}_{j=1}^J$ with weights $\{w_{n,j}\}_{j=1}^J$ that are associated to the distribution $\mathcal{N}(\mu_n^P, \Sigma_n^P)$. These nodes and weights can either be constructed to integrate polynomials up to some order exactly (see, e.g., McNamee and Stenger (1967) and Golub and Welsch (1969)), or by Bayesian quadrature (Briol *et al.*, 2019). In either case, the expectation of a function $\psi(X_n)$ is approximated by

$$\mathbb{E}[\psi(X_n)] \approx \sum_{j=1}^{J} w_{n,j}\psi(\mathcal{X}_{n,j}). \tag{B.17}$$

Therefore, by appropriate choices of ψ the quantities in Eq. (B.12) can be approximated. We shall refer to filters using a third degree fully symmetric rule (McNamee and Stenger, 1967) as Unscented Kalman filters (UKF), which is the name that was adopted when it was first introduced to the signal processing community (Julier *et al.*, 2000). For a suitable cross-covariance assumption and a particular choice of quadrature, the method of Kersting and Hennig (2016) is retrieved. This is Proposition B.2.4.

Proposition B.2.4. *Let $\{\mathcal{X}_{n,j}\}_{j=1}^J$ and $\{w_{n,j}\}_{j=1}^J$ be the nodes and weights, corresponding to a Bayesian quadrature rule with respect to $\mathcal{N}(\mu_n^P, \Sigma_n^P)$. Furthermore, assume $R = 0$ and that the cross-covariance between $\dot{C}X_n$ and $f(CX_n, t_n)$ is approximated as zero,*

$$\mathbb{C}\Big[\dot{C}X_n, f(CX_n, t_n) \mid z_{1:n-1}\Big] \approx 0. \tag{B.18}$$

Then the probabilistic solver proposed in Kersting and Hennig (2016) is a Bayesian quadrature approximation to Eq. (B.12)*.*

A proof of Proposition B.2.4 is given in Appendix B.8.

While a cross-covariance assumption of Proposition B.2.4 reproduces the method of Kersting and Hennig (2016), Bayesian quadrature approximations have previously been used for Gaussian filtering in signal processing applications by Prüher and Šimandl (2015), which in this context gives a new solver.

B.2.6 Affine vector fields

It is instructive to examine the particular case when the vector field in Eq. (B.1) is affine. That is,

$$f(y(t), t) = \Lambda(t)y(t) + \zeta(t). \tag{B.19}$$

In such a case, Eq. (B.7) becomes a linear Gaussian system, which is solved exactly by a Kalman filter. The equations for implementing this Kalman filter are precisely Eq. (B.11) and Eq. (B.12), although the latter set of equations can be simplified. Define $H_n = \dot{C} - \Lambda(t_n)C$, then the update equations become

$$S_n = H_n \Sigma_n^P H_n^\mathsf{T} + R, \tag{B.20a}$$
$$K_n = \Sigma_n^P H_n^\mathsf{T} S_n^{-1}, \tag{B.20b}$$
$$\mu_n^F = \mu_n^P + K_n\Big(\zeta(t_n) - H_n \mu_n^P\Big) \tag{B.20c}$$
$$\Sigma_n^F = \Sigma_n^P - K_n S_n K_n^\mathsf{T}. \tag{B.20d}$$

Lemma B.2.5. *Consider the inference problem in Eq.* (B.7) *with an affine vector field as given in Eq.* (B.19)*. Then the EKF reduces to the exact Kalman filter, which uses the update in Eq.* (B.20)*. Furthermore, the same holds for Gaussian filters using a quadrature approximation to Eq.* (B.12)*, provided that it integrates polynomials correctly up to second order with respect to the distribution* $\mathcal{N}(\mu_n^P, \Sigma_n^P)$*.*

Proof. Since the Kalman filter, the EKF, and the quadrature approach all use Eq. (B.11) for prediction, it is sufficient to make sure that the EKF and the quadrature approximation compute Eq. (B.12) exactly, just as the Kalman filter. Now the EKF approximates the vector field by an affine function for which it computes the moments in Eq. (B.12) exactly. Since this affine approximation is formed by a truncated Taylor series, it is exact for affine functions and the statement pertaining to the EKF holds. Furthermore, the Gaussian integrals in Eq. (B.12) are polynomials of degree at most two for affine vector fields and are therefore computed exactly by the quadrature rule by assumption. □ □

B.2.7 Particle filtering

The Gaussian filtering methods from Section B.2.3 may often suffice. However, there are cases where more sophisticated inference methods may be preferable, for instance, when the posterior becomes multi-modal due to chaotic behavior or 'numerical bifurcations'. That is, when it is numerically unknown whether the true solution is above or below a

certain threshold that determines the limit behaviour of its trajectory. While sampling based probabilistic solvers such as those of Chkrebtii *et al.* (2016), Conrad *et al.* (2017), Teymur *et al.* (2016), Lie *et al.* (2019), Abdulle and Garegnani (2020), and Teymur *et al.* (2018) can pick up such phenomena, the Gaussian filtering based ODE solvers discussed in Section B.2.3 cannot. However, this limitation may be overcome by approximating the filtering distribution of the inference problem in Eq. (B.7) with particle filters that are based on a sequential formulation of importance sampling (Doucet *et al.*, 2001).

A particle filter operates on a set of particles, $\{X_{n,j}\}_{j=1}^{J}$, a set of positive weights $\{w_{n,j}\}_{j=1}^{J}$ associated to the particles that sum to one and an importance density, $g(x_{n+1} \mid x_n, z_n)$. The particle filter then cycles through three steps (1) propagation, (2) re-weighting, and (3) re-sampling (Särkkä, 2013, Chapter 7.4).

The propagation step involves sampling particles at time $n+1$ from the importance density:

$$X_{n+1,j} \sim g(x_{n+1} \mid X_{n,j}, z_n). \tag{B.21}$$

The re-weighting of the particles is done by a likelihood ratio with the product of the measurement density and the transition density of Eq. (B.7), and the importance density. That is, the updated weights are given by

$$\rho(x_{n+1}, x_n) = \frac{p(z_{n+1} \mid x_{n+1}) p(x_{n+1} \mid x_n)}{g(x_{n+1} \mid x_n, z_{n+1})}, \tag{B.22a}$$

$$w_{n+1,j} \propto \rho\Big(X_{n+1,j}, X_{n,j}\Big) w_{n,j}, \tag{B.22b}$$

where the proportionality sign indicates that the weights need to be normalised to sum to one after they have been updated according to Eq. (B.22). The weight update is then followed by an optional re-sampling step (Särkkä, 2013, Chapter 7.4). While not re-sampling in principle yields a valid algorithm, it becomes necessary in order to avoid the degeneracy problem for long time series (Doucet *et al.*, 2001, Chapter 1.3). The efficiency of particle filters depends on the choice of importance density. In terms of variance, the locally optimal importance density is given by (Doucet *et al.*, 2001)

$$g(x_n \mid x_{n-1}, z_n) \propto p(z_n \mid x_n) p(x_n \mid x_{n-1}). \tag{B.23}$$

While Eq. (B.23) is almost as intractable as the full filtering distribution, the Gaussian filtering methods from Section B.2.3 can be used to make a good approximation. For instance, the approximation to the optimal importance density using Eq. (B.14) is given

by

$$S_n = \dot{C}Q(h)\dot{C}^{\mathsf{T}} + R, \tag{B.24a}$$
$$K_n = Q(h)\dot{C}^{\mathsf{T}}S_n^{-1}, \tag{B.24b}$$
$$\hat{z}_n = \dot{C}A(h)x_{n-1} - f\Big(CA(h)x_{n-1}, t_n\Big), \tag{B.24c}$$
$$\mu_n = A(h)x_{n-1} + K_n(z_n - \hat{z}_n), \tag{B.24d}$$
$$\Sigma_n = Q(h) - K_nS_nK_n^{\mathsf{T}}, \tag{B.24e}$$
$$g(x_n \mid x_{n-1}, z_n) = \mathcal{N}(x_n; \mu_n, \Sigma_n). \tag{B.24f}$$

An importance density can be similarly constructed from Eq. (B.16), resulting in:

$$\tilde{C}_n = \dot{C} - J_f\Big(CA(h)x_{n-1}, t_n\Big)C, \tag{B.25a}$$
$$S_n = \tilde{C}_nQ(h)\tilde{C}_n^{\mathsf{T}} + R, \tag{B.25b}$$
$$K_n = Q(h)\tilde{C}_n^{\mathsf{T}}S_n^{-1}, \tag{B.25c}$$
$$\hat{z}_n = \dot{C}A(h)x_{n-1} - f\Big(CA(h)x_{n-1}, t_n\Big), \tag{B.25d}$$
$$\mu_n = A(h)x_{n-1} + K_n(z_n - \hat{z}_n), \tag{B.25e}$$
$$\Sigma_n = Q(h) - K_nS_nK_n^{\mathsf{T}}, \tag{B.25f}$$
$$g(x_n \mid x_{n-1}, z_n) = \mathcal{N}(x_n; \mu_n, \Sigma_n). \tag{B.25g}$$

Note that we have assumed $\xi(h) = 0$ in Eqs. (B.24) and (B.25), which can be extended to $\xi(h) \neq 0$ by replacing $A(h)x_{n-1}$ with $A(h)x_{n-1}+\xi(h)$. We refer the reader to Doucet *et al.* (2000, Section II.D.2) for a more thorough discussion on the use of local linearisation methods to construct importance densities.

We conclude this section with a brief discussion on the convergence of particle filters. The following theorem is given by Crisan and Doucet (2002).

Theorem B.2.6. *Let $\rho(x_{n+1}, x_n)$ in Eq.* (B.22a) *be bounded from above and denote the true filtering measure associated with Eq.* (B.7) *at time n by p_n^R and let $\hat{p}_n^{R,J}$ be its particle approximation using J particles with importance density $g(x_{n+1} \mid x_n, z_{n+1})$. Then, for all $n \in \mathbb{N}_0$, there exists a constant c_n independent of J such that for any bounded Borel function $\phi\colon \mathbb{R}^{d(q+1)} \to \mathbb{R}$ the following bound holds*

$$\mathbb{E}_{\mathrm{MC}}[(\langle \hat{p}_n^{R,J}, \phi\rangle - \langle p_n^R, \phi\rangle)^2]^{1/2} \leq c_nJ^{-1/2}\|\phi\|, \tag{B.26}$$

where $\langle p, \phi\rangle$ denotes ϕ integrated with respect to p and $\mathbb{E}_{\mathrm{MC}}$ denotes the expectation over realisations of the particle method, and $\|\cdot\|$ is the supremum norm.

Theorem B.2.6 shows that we can decrease the distance (in the weak sense) between $\hat{p}_n^{R,J}$ and p_n^R by increasing J. However, the object we want to approximate is p_n^0 (the exact filtering measure associated with Eq. (B.7) for $R = 0$) but setting $R = 0$ makes

the likelihood ratio in Eq. (B.22a) ill-defined for the proposal distributions in Eqs. (B.24) and (B.25). This is because, when $R = 0$, then $p(z_{n+1} \mid x_{n+1})p(x_{n+1} \mid x_n)$ has its support on the surface $\dot{C}x_{n+1} = f(Cx_{n+1}, t_{n+1})$ while Eqs. (B.24) or (B.25) imply that the variance of $\dot{C}X_{n+1}$ or $\tilde{C}_{n+1}X_{n+1}$ will be zero with respect to $g(x_{n+1} \mid x_n, z_{n+1})$, respectively. That is, $g(x_{n+1} \mid x_n, z_{n+1})$ is supported on a hyperplane. It follows that the null-sets of $g(x_{n+1} \mid x_n, z_{n+1})$ are not necessarily null-sets of $p(z_{n+1} \mid x_{n+1})p(x_{n+1} \mid x_n)$ and the likelihood ratio in Eq. (B.22a) can therefore be undefined. However, a straightforward application of the triangle inequality together with Theorem B.2.6 gives

$$\begin{aligned}\mathbb{E}_{\mathrm{MC}}[(\langle \hat{p}_n^{R,J}, \phi\rangle - \langle p_n^0, \phi\rangle)^2]^{1/2} &\leq \mathbb{E}_{\mathrm{MC}}[(\langle \hat{p}_n^{R,J}, \phi\rangle - \langle p_n^R, \phi\rangle)^2]^{1/2} + \mathbb{E}_{\mathrm{MC}}[(\langle p_n^R, \phi\rangle - \langle p_n^0, \phi\rangle)^2]^{1/2} \\ &= \mathbb{E}_{\mathrm{MC}}[(\langle \hat{p}_n^{R,J}, \phi\rangle - \langle p_n^R, \phi\rangle)^2]^{1/2} + \left|\langle p_n^R, \phi\rangle - \langle p_n^0, \phi\rangle\right| \\ &\leq c_n J^{-1/2}\|\phi\| + \left|\langle p_n^R, \phi\rangle - \langle p_n^0, \phi\rangle\right|. \end{aligned} \tag{B.27}$$

The last term vanishes as $R \to 0$. That is, the error can be controlled by increasing the number of particles J and decreasing R. Though a word of caution is appropriate, as particle filters can become ill-behaved in practice if the likelihoods are too narrow (too small R). However, this also depends on the quality of the proposal distribution.

Lastly, while Theorem B.2.6 is only valid if $\rho(x_{n+1}, x_n)$ is bounded, this can be ensured by either inflating the covariance of the proposal distribution or replacing the Gaussian proposal with a Student's t proposal (Cappé *et al.*, 2005, Chapter 9).

B.3 A stability result for Gaussian filters

ODE solvers are often characterised by the properties of their solution to the linear test equation

$$\dot{y}(t) = \lambda y(t),\ y(0) = 1, \tag{B.28}$$

where λ is some complex number. A numerical solver is said to be *A-stable* if the approximate solution tends to zero for any fixed step size h whenever the real part of λ resides in the left half-plane (Dahlquist, 1963). Recall that if $y_0 \in \mathbb{R}^d$ and $\Lambda \in \mathbb{R}^{d\times d}$ then the ODE $\dot{y}(t) = \Lambda y(t),\ y(0) = y_0$ is said to be asymptotically stable if $\lim_{t\to\infty} y(t) = 0$, which is precisely when the real part of eigenvalues of Λ are in the left half-plane. That is, A-stability is the notion that a numerical solver preserves asymptotic stability of linear time-invariant ODEs.

While the present solvers are not designed to solve complex valued ODEs, a real system equivalent to Eq. (B.28) is given by

$$\dot{y}(t) = \Lambda_{\mathrm{test}} y(t), \quad y^{\mathsf{T}}(0) = [1\ 0], \tag{B.29}$$

where $\lambda = \lambda_1 + i\lambda_2$ and

$$\Lambda_{\text{test}} = \begin{bmatrix} \lambda_1 & -\lambda_2 \\ \lambda_2 & \lambda_1 \end{bmatrix}. \tag{B.30}$$

However, to leverage classical stability results from the theory of Kalman filtering we investigate a slightly different test equation, namely

$$\dot{y}(t) = \Lambda y(t), \; y(0) = y_0, \tag{B.31}$$

where $\Lambda \in \mathbb{R}^{d\times d}$ is of full rank. In this case Eqs. (B.11) and (B.20) give the following recursion for μ_n^P

$$\mu_{n+1}^P = (A(h) - A(h)K_nH)\mu_n^P, \tag{B.32a}$$
$$\mu_n^F = (\mathrm{I} - K_nH)\mu_n^P, \tag{B.32b}$$

where we recall that $H = \dot{C} - C\Lambda$ and $z_n = 0$. If there exists a limit gain $\lim_{n\to\infty} K_n = K_\infty$ then asymptotic stability of the filter holds provided that the eigenvalues of $(A(h) - A(h)K_\infty H)$ are strictly within the unit circle (Anderson and Moore, 1979, Appendix C, page 341). That is, $\lim_{n\to\infty} \mu_n^P = 0$ and as a direct consequence $\lim_{n\to\infty} \mu_n^F = 0$.

We shall see that the Kalman filter using an IWP(q) prior is asymptotically stable. For the IWP(q) process on $\mathbb{R}^d$ we have $u = 0$, $L = \mathrm{e}_{q+1} \otimes \Gamma^{1/2}$, and $F = (\sum_{i=1}^q \mathrm{e}_i \mathrm{e}_{i+1}^\intercal) \otimes \mathrm{I}$, where $\mathrm{e}_i \in \mathbb{R}^d$ is the ith canonical eigenvector, $\Gamma^{1/2}$ is the symmetric square root of some positive semi-definite matrix $\Gamma \in \mathbb{R}^{d\times d}$, $\mathrm{I} \in \mathbb{R}^{d\times d}$ is the identity matrix, and $\otimes$ is Kronecker's product. By using Eq. (B.8), the properties of Kronecker products, and the definition of the matrix exponential the equivalent discrete-time system is given by

$$A(h) = A^{(1)}(h) \otimes \mathrm{I}, \tag{B.33a}$$
$$\xi(h) = 0, \tag{B.33b}$$
$$Q(h) = Q^{(1)}(h) \otimes \Gamma, \tag{B.33c}$$

where $A^{(1)}(h) \in \mathbb{R}^{(q+1)\times(q+1)}$ and $Q^{(1)}(h) \in \mathbb{R}^{(q+1)\times(q+1)}$ are given by (Kersting *et al.*, 2020a, Appendix A)[2]

$$A_{ij}^{(1)}(h) = \mathbb{I}_{i\le j} \frac{h^{j-i}}{(j-i)!}, \tag{B.34a}$$
$$Q_{ij}^{(1)}(h) = \frac{h^{2q+3-i-j}}{(2q+3-i-j)(q+1-i)!(q+1-j)!}, \tag{B.34b}$$

and $\mathbb{I}_{i\le j}$ is an indicator function. Before proceeding we need to introduce the notions of stabilisability and detectability from Kalman filtering theory. These notions can be

[2]Note that Kersting *et al.* (2020a) uses indexing $i, j = 0, \ldots, q$ while we here use $i, j = 1, \ldots, q+1$.

found in Anderson and Moore (1979, Appendix C).

Definition B.3.1 (Complete Stabilisability)**.** *The pair* $[A, G]$ *is completely stabilisable if* $w^\mathsf{T} G = 0$ *and* $w^\mathsf{T} A = \eta w^\mathsf{T}$ *for some constant* η *implies* $|\eta| < 1$ *or* $w = 0$*.*

Definition B.3.2 (Complete Detectability)**.** [3] $[A, H]$ *is completely detectable if* $[A^\mathsf{T}, H^\mathsf{T}]$ *is completely stabilisable.*

Before we state the stability result of this section the following two lemmas are useful.

Lemma B.3.3. *Consider the discretised IWP(q) prior on* $\mathbb{R}^d$ *as given by Eq.* (B.33)*. Let* $h > 0$ *and* Γ *be positive definite. Then, the* $d \times d$ *blocks of* $Q(h)$*, denoted by* $Q_{i,j}(h),\ i, j = 1, 2, \ldots, q+1$ *are of full rank.*

Proof. From Eq. (B.33c) we have $Q_{i,j}(h) = Q^{(1)}_{i,j}(h)\Gamma$. From Eq. (B.34b) and $h > 0$ we have $Q^{(1)}_{i,j}(h) > 0$, and since Γ is positive definite it is of full rank. It then follows that $Q_{i,j}(h)$ is of full rank as well. □ □

Lemma B.3.4. *Let* $A(h)$ *be the transition matrix of an IWP(q) prior as given by Eq.* (B.33a) *and* $h > 0$*, then* $A(h)$ *has a single eigenvalue given by* $\eta = 1$*. Furthermore, the right-eigenspace is given by*

$$\operatorname{span}[\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_d],$$

where $\mathrm{e}_i \in \mathbb{R}^{(q+1)d}$ *are canonical basis vectors, and the left-eigenspace is given by*

$$\operatorname{span}[\mathrm{e}_{qd+1}, \mathrm{e}_{qd+2}, \ldots, \mathrm{e}_{(q+1)d}].$$

Proof. Firstly, from Eqs. (B.33a) and (B.34a) it follows that $A(h)$ is block upper-triangular with identity matrices on the block diagonal, hence the characteristic equation is given by

$$\det(A(h) - \eta \mathrm{I}) = (1 - \eta)^{(q+1)d} = 0, \tag{B.35}$$

we conclude that the only eigenvalue is $\eta = 1$. To find the right-eigenspace let $w^\mathsf{T} = [w_1^\mathsf{T}, w_2^\mathsf{T}, \ldots, w_{q+1}^\mathsf{T}]$, $w_i \in \mathbb{R}^d,\ i = 1, 2, \ldots, q+1$ and solve $A(h)w = w$, which by using Eqs. (B.33a) and (B.34a) can be written as

$$(A(h)w)_l = \sum_{r=0}^{q+1-l} \frac{h^r}{r!} w_{r+l}, \quad l = 1, 2, \ldots, q+1, \tag{B.36}$$

where $(\cdot)_l$ is the lth sub-vector of dimension d. Starting with $l = q+1$ we trivially have $w_{q+1} = w_{q+1}$. For $l = q$ we have $w_q + w_{q+1}h = w_q$ but $h > 0$, hence $w_{q+1} = 0$. Similarly for $l = q-1$ we have $w_{q-1} = w_{q-1} + w_q h + w_{q+1}h^2/2 = w_{q-1} + w_q h + 0 \cdot h^2/2$. Again since $h > 0$ we have $w_q = 0$. By repeating this argument we have $w_1 = w_1$

[3] Anderson and Moore (1979) denotes the measurement matrix by H^T while we denote it by H. With this in mind our notion of complete detectability does not differ from Anderson and Moore (1979).

and $w_i = 0,\ i = 2, 3, \ldots, q+1$. Therefore all eigenvectors w are of the form $w^\mathsf{T} = [w_1^\mathsf{T}, 0^\mathsf{T}, \ldots, 0^\mathsf{T}] \in \text{span}[\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_d]$. Similarly, for the left eigenspace we have

$$(w^\mathsf{T} A(h))_l = \sum_{r=0}^{l-1} \frac{h^r}{r!} w_{l-r}^\mathsf{T}, \quad l = 1, 2, \ldots, q+1. \tag{B.37}$$

Starting with $l = 1$ we have trivially that $w_1^\mathsf{T} = w_1^\mathsf{T}$. For $l = 2$ we have $w_2^\mathsf{T} + w_1^\mathsf{T} h = w_2^\mathsf{T}$ but $h > 0$, hence $w_1 = 0$. For $l = 3$ we have $w_3^\mathsf{T} = w_3^\mathsf{T} + w_2^\mathsf{T} h + w_1^\mathsf{T} h^2/2 = w_3^\mathsf{T} + w_2^\mathsf{T} h + 0^\mathsf{T} \cdot h^2/2$ but $h > 0$ hence $w_2 = 0$. By repeating this argument we have $w_i = 0, i = 1, \ldots, q$ and $w_{q+1} = w_{q+1}$. Therefore, all left eigenvectors are of the form $w^\mathsf{T} = [0^\mathsf{T}, \ldots, 0^\mathsf{T}, w_{q+1}^\mathsf{T}] \in \text{span}[\mathrm{e}_{qd+1}, \mathrm{e}_{qd+2}, \ldots, \mathrm{e}_{(q+1)d}]$. □ □

We are now ready to state the main result of this section. Namely, that the Kalman filter that produces exact inference in Eq. (B.7) for linear vector fields is asymptotically stable if the linear vector field is of full rank.

Theorem B.3.5. *Let $\Lambda \in \mathbb{R}^{d \times d}$ be a matrix with full rank and consider the linear ODE*

$$\dot{y}(t) = \Lambda y(t). \tag{B.38}$$

Consider estimating the solution of Eq. (B.38) *using an IWP(q) prior with the same conditions on Γ as in Lemma B.3.3. Then the Kalman filter estimate of the solution to Eq.* (B.38) *is asymptotically stable.*

Proof. From Eq. (B.7) we have that the Kalman filter operates on the following system

$$X_{n+1} = A(h) X_n + Q^{1/2}(h) W_{n+1}, \tag{B.39a}$$
$$Z_n = H X_n, \tag{B.39b}$$

where $H = [-\Lambda, \mathrm{I}, 0, \ldots, 0]$ and W_n are i.i.d. standard Gaussian vectors. It is sufficient to show that $[A(h), H]$ is completely detectable and $[A(h), Q^{1/2}(h)]$ is completely stabilisable (Anderson and Moore, 1979, Chapter 4, page 77). We start by showing complete detectability. If we let $w^\mathsf{T} = [w_1^\mathsf{T}, \ldots, w_{q+1}^\mathsf{T}]$, $w_i \in \mathbb{R}^d,\ i = 1, 2, \ldots, q+1$, then by Lemma B.3.4 we have that $w^\mathsf{T} A^\mathsf{T}(h) = \eta w^\mathsf{T}$ for some η implies that either $w = 0$ or $w^\mathsf{T} = [w_1^\mathsf{T}, 0^\mathsf{T}, \ldots, 0^\mathsf{T}]$ for some $w_1 \in \mathbb{R}^d$ and $\eta = 1$. Furthermore, $w^\mathsf{T} H^\mathsf{T} = -w_1^\mathsf{T} \Lambda^\mathsf{T} + w_2^\mathsf{T} = 0$ implies that $w_2 = \Lambda w_1$. However, by the previous argument, we have $w_2 = 0$, therefore $0 = \Lambda w_1$ but Λ is full rank by assumption so $w_1 = 0$. Therefore, $[A^\mathsf{T}(h), H^\mathsf{T}]$ is completely detectable. As for complete stabilisability, again by Lemma B.3.4, we have $w^\mathsf{T} A(h) = \eta w^\mathsf{T}$ for some η, which implies either $w = 0$ or $w^\mathsf{T} = [0^\mathsf{T}, \ldots, 0^\mathsf{T}, w_{q+1}^\mathsf{T}]$ and $\eta = 1$. Furthermore, since the nullspace of $Q^{1/2}(h)$ is the same as the nullspace of $Q(h)$, we have that $w^\mathsf{T} Q^{1/2}(h) = 0$ is equivalent to $w^\mathsf{T} Q(h) = 0$,

which is given by

$$w^{\mathsf{T}}Q(h) = \begin{bmatrix} w_{q+1}^{\mathsf{T}}Q_{q+1,1}(h) & \dots & w_{q+1}^{\mathsf{T}}Q_{q+1,q+1}(h) \end{bmatrix} = 0,$$

but by Lemma B.3.3 the blocks $Q_{i,j}(h)$ have full rank so $w_{q+1} = 0$ and thus $w = 0$. To conclude, we have that $[A(h), Q^{1/2}(h)]$ is completely stabilisable and $[A(h), H]$ is completely detectable and therefore the Kalman filter is asymptotically stable. □ □

Corollary B.3.6. *In the same setting as Theorem B.3.5, the EKF and UKF are asymptotically stable.*

Proof. Since the vector field is linear and therefore affine Lemma B.2.5 implies that EKF and UKF reduce to the exact Kalman filter, which is asymptotically stable by Theorem B.3.5. □ □

It is worthwhile to note that Λ_{test} is of full rank for all $[\lambda_1 \ \lambda_2]^{\mathsf{T}} \in \mathbb{R}^2 \setminus \{0\}$, and consequently Theorem B.3.5 and Corollary B.3.6 guarante A-stability for the EKF and UKF in the sense of Dahlquist (1963)[4]. Lastly, a peculiar fact about Theorem B.3.5 is that it makes no reference to the eigenvalues of Λ (i.e. the stability properties of the ODE). That is, the Kalman filter will be asymptotically stable even if the underlying ODE is not, provided that, Λ is of full rank. This may seem awkward but it is rarely the case that the ODE that we want to integrate is unstable, and even in such a case most solvers will produce an error that grows without a bound as well. Though all of the aforementioned properties are at least partly consequences of using IWP(q) as a prior and they may thus be altered by changing the prior.

B.4 Uncertainty calibration

In practice the model parameters, (F, u, L), might depend on some parameters that need to be estimated for the probabilistic solver to report appropriate uncertainty in the estimated solution to Eq. (B.1). The diffusion matrix L is of particular importance as it determines the gain of the Wiener process entering the system in Eq. (B.3) and thus determines how *'diffuse'* the prior is. Herein we shall only concern ourselves with estimating L, though, one might anticipate future interest in estimating F and u as well. However, let us start with a few words on the monitoring of errors in numerical solvers in general.

B.4.1 Monitoring of errors in numerical solvers

An important aspect of numerical analysis is to monitor the error of a method. While the goal of probabilistic solvers is to do so by calibration of a probabilistic model, the

[4]Some authors require stability on the line $\lambda_1 = 0$ as well (Hairer and Wanner, 1996). Due to the exclusion of origin EKF and UKF cannot be said to be A-stable in this sense.

approach of classical numerical analysis is to examine the local and global errors. The global error can be bounded but is typically impractical for monitoring error (Hairer *et al.*, 1987, Chapter II.3). A more practical approach is to monitor (and control) the accumulation of local errors. This can be done by using two step sizes together with Richardson extrapolation (Hairer *et al.*, 1987, Theorem 4.1). Though, perhaps more commonly this is done via embedded Runge–Kutta methods (Hairer *et al.*, 1987, Chapter II.4) or the Milne device Byrne and Hindmarsh (1975).

In the context of filters, the relevant object in this regard is the scaled residual $S_n^{-1/2}(z_n - \hat{z}_n)$. Due to its role in the prediction-error decomposition, which is defined below, it directly monitors the calibration of the predictive distribution. Schober *et al.* (2019) showed how to use this quantity to effectively control step sizes in practice. It was also recently shown in (Kersting *et al.*, 2020a, Section 7), that in the case of $q = 1$, fixed σ^2 (amplitude of the Wiener process) and Integrated Wiener Process prior, the posterior standard deviation computed by the solver of Schober *et al.* (2019) contracts at the same rate as the worst-case error as the step size goes to zero—thereby preventing both under- and over-confidence.

In the following we discuss effective strategies for calibrating L when it is given by $L = \sigma\breve{L}$ for fixed $\breve{L}$ thus providing a probabilistic quantification of the error in the proposed solvers.

B.4.2 Uncertainty calibration for affine vector fields

As noted in Section B.2.6, the Kalman filter produces the exact solution to the inference problem in Eq. (B.7) when the vector field is affine. Furthermore, the marginal likelihood $p(z_{1:N})$ can be computed during the execution of the Kalman filter by the prediction error decomposition (Schweppe, 1965), which is given by:

$$\begin{aligned} p(z_{1:N}) &= p(z_1) \prod_{n=2}^{N} p(z_n \mid z_{1:n-1}) \\ &= \prod_{n=1}^{N} \mathcal{N}(z_n; \hat{z}_n, S_n). \end{aligned} \tag{B.40}$$

While the marginal likelihood in Eq. (B.40) is certainly straightforward to compute without adding much computational cost, maximising it is a different story in general. In the particular case when the diffusion matrix L and the initial covariance Σ_0 are given by re-scaling fixed matrices $L = \sigma\breve{L}$ and $\Sigma_0 = \sigma^2\breve{\Sigma}_0$ for some scalar $\sigma > 0$, then uncertainty calibration can be done by a simple post-processing step after running the Kalman filter, as is shown in Proposition B.4.1 below.

Proposition B.4.1. *Let $f(y,t) = \Lambda(t)y + \zeta(t)$, $\Sigma_0 = \sigma^2\breve{\Sigma}_0$, $L = \sigma\breve{L}$, $R = 0$ and denote the equivalent discrete-time process noise covariance for the prior model $(F, u, \breve{L})$ by*

$\breve{Q}(h)$. *Then the Kalman filter estimate to the solution of*

$$\dot{y}(t) = f(y(t), t)$$

that uses the parameters $(\mu_0^F, \Sigma_0, A(h), \xi(h), Q(h))$ *is equal to the Kalman filter estimate that uses the parameters* $(\mu_0^F, \breve{\Sigma}_0, A(h), \xi(h), \breve{Q}(h))$. *More specifically, if we denote the filter mean and covariance at time* n *using the former parameters by* (μ_n^F, Σ_n^F) *and the corresponding filter mean and covariance using the latter parameters by* $(\breve{\mu}_n^F, \breve{\Sigma}_n^F)$, *then* $(\mu_n^F, \Sigma_n^F) = (\breve{\mu}_n^F, \sigma^2 \breve{\Sigma}_n^F)$. *Additionally, denote the predicted mean and covariance of the measurement* Z_n *by* $\breve{z}_n$ *and* $\breve{S}_n$, *respectively, when using the parameters* $(\mu_0^F, \breve{\Sigma}_0, A(h), \xi(h), \breve{Q}(h))$. *Then the maximum likelihood estimate of* σ^2, *denoted by* $\widehat{\sigma_N^2}$, *is given by*

$$\widehat{\sigma_N^2} = \frac{1}{Nd} \sum_{n=1}^{N} (z_n - \breve{z}_n)^\mathsf{T} \breve{S}_n^{-1} (z_n - \breve{z}_n). \tag{B.41}$$

Proposition B.4.1 is just an amalgamation of statements from Tronarp *et al.* (2019b). Nevertheless, we provide an accessible proof in Appendix B.9.

B.4.3 Uncertainty calibration for non-affine vector fields

For non-affine vector fields the issue of parameter estimation becomes more complicated. The Bayesian filtering problem is not solved exactly and consequently any marginal likelihood will be approximate as well. Nonetheless, a common approach in the Gaussian filtering framework is to approximate the marginal likelihood in the same manner as the filtering solution is approximated (Särkkä, 2013, Chapter 12.3.3), that is:

$$p(z_{1:N}) \approx \prod_{n=1}^{N} \mathcal{N}(z_n; \hat{z}_n, S_n), \tag{B.42}$$

where $\hat{z}_n$ and S_n are the quantities in Eq. (B.12) approximated by some method (e.g. EKF). Maximising Eq. (B.42) is a common approach in signal processing (Särkkä, 2013) and referred to as *quasi maximum likelihood* in time series literature (Lindström *et al.*, 2015). Both Eq. (B.14) and Eq. (B.16) can be thought of as Kalman updates for the case where the vector field is approximated by a piece-wise affine function, without modifying Σ_0, $Q(h)$, and R. For instance the affine approximation of the vector field due to the EKF on the discretisation interval $[t_n, t_{n+1})$ is given by

$$\hat{\zeta}_n(t) = f\Big(C\mu_n^P, t_n\Big) - J_f\Big(C\mu_n^P, t_n\Big) C\mu_n^P, \tag{B.43a}$$

$$\hat{\Lambda}_n(t) = J_f\Big(C\mu_n^P, t_n\Big), \tag{B.43b}$$

$$\hat{f}_n(y, t) = \hat{\Lambda}_n(t) y + \hat{\zeta}_n(t). \tag{B.43c}$$

While the vector field is approximated by a piece-wise affine function, the discrete-time filtering problem Eq. (B.7) is still simply an affine problem, without modifications of Σ_0, $Q(h)$, and R. Therefore, the results of Proposition B.4.1 still apply and the σ^2 maximising the approximate marginal likelihood in Eq. (B.42) can be computed in the same manner as in Eq. (B.41).

On the other hand, it is clear that dependence on σ^2 in Eq. (B.12) is non-trivial in general, which is also true for the quadrature approaches of Section B.2.5. Therefore, maximising Eq. (B.42) for the quadrature approaches is not as straightforward. However, by Taylor series expanding the vector field in Eq. (B.12) one can see that the numerical integration approaches are roughly equal to the Taylor series approaches provided that $\breve{\Sigma}_n^P$ is small. Therefore, we opt for plugging in the corresponding quantities from the quadrature approximations into Eq. (B.41) in order to achieve computationally cheap calibration of these approaches.

Remark B.4.2. *A local calibration method for σ^2 is given by (Schober* et al.*, 2019, Eq. (45)), which in fact corresponds to an h-dependent prior, with the diffusion matrix in Eq.* (B.3) *$L = L(t)$ being piece-wise constant over integration steps. Moreover, Schober* et al. *(2019) had to neglect the dependence of Σ_n^P on the likelihood. Here we prefer the estimator given in Eq.* (B.41) *since it is attempting to maximise the likelihood from the globally defined probability model in Eq.* (B.7)*, and it succeeds for affine vector fields.*

More advanced methods for calibrating the parameters of the prior can be developed by combining the Gaussian smoothing equations (Särkkä, 2013, Chapter 10) with the expectation maximisation method (Kokkala *et al.*, 2014) or variational Bayes (Taniguchi *et al.*, 2017).

B.4.4 Uncertainty calibration of particle filters

If calibration of Gaussian filters was complicated by having a non-affine vector field, the situation for particle filters is even more challenging. There is, to the authors' knowledge, no simple estimator of the scale of the Wiener process (such as Proposition B.4.1) even for the case of affine vector fields. However, the literature on parameter estimation using particle methods is vast so we proceed to point the reader towards some alternatives. In the class of off-line methods, Schön *et al.* (2011) uses a particle smoother to implement an expectation maximisation algorithm, while Lindsten (2013) uses a particle Markov chain Monte Carlo methods to implement a stochastic approximation expectation maximisation algorithm. One can also use the iterated filtering method of Ionides *et al.* (2011) to get a maximum likelihood estimator, or particle Markov chain Monte Carlo (Andrieu *et al.*, 2010).

On the other hand, if on-line calibration is required then the gradient based recursive maximum likelihood estimator by Doucet and Tadić (2003) can be used, or the on-line version of iterated filtering by Lindström *et al.* (2012). Furthermore, Storvik (2002) provides an alternative for on-line calibration when sufficient statics of the parameters

are finite dimensional and can be computed recursively in n. An overview on parameter estimation using particle filters was also given by Kantas *et al.* (2009).

B.5 Experimental results

In this section we evaluate the different solvers presented in this paper in different scenarios. Though before we proceed to the experiments we define some summary metrics with which assessments of accuracy and uncertainty quantification can be made. The root mean square error (RMSE) is often used to assess accuracy of filtering algorithms and is defined by

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} \|y(nh) - C\mu_n^F\|^2}.$$

In fact $y(nh) - C\mu_n^F$ is precisely the *global error* at time t_n (Hairer *et al.*, 1987, Eq. (3.16)). As for assessing the uncertainty quantification, the χ^2-statistics is commonly used (Bar-Shalom *et al.*, 2001). That is, in a linear Gaussian model the following quantities

$$\Big(y(nh) - C\mu_n^F\Big)^{\mathsf{T}} [C\Sigma_n^F C^{\mathsf{T}}]^{-1} \Big(y(nh) - C\mu_n^F\Big),\ n = 1, \ldots, N,$$

are i.i.d. $\chi^2(d)$. For a trajectory summary we define the average χ^2-statistics as

$$\bar{\chi}^2 = \frac{1}{N}\sum_{n=1}^{N} \Big(y(nh) - C\mu_n^F\Big)^{\mathsf{T}} [C\Sigma_n^F C^{\mathsf{T}}]^{-1} \Big(y(nh) - C\mu_n^F\Big).$$

For an accurate and well calibrated model the RMSE is small and $\bar{\chi}^2 \approx d$. In the succeeding discussion we shall refer to a method producing $\bar{\chi}^2 < d$ or $\bar{\chi}^2 > d$ as *underconfident* or *overconfident*, respectively.

B.5.1 Linear systems

In this experiment we consider a linear system given by

$$\Lambda = \begin{bmatrix} \lambda_1 & -\lambda_2 \\ \lambda_2 & \lambda_1 \end{bmatrix}, \tag{B.44a}$$

$$\dot{y}(t) = \Lambda y(t), \quad y(0) = \mathrm{e}_1. \tag{B.44b}$$

This makes for a good test model as the inference problem in Eq. (B.7) can be solved exactly, and consequently its adequacy can be assessed. We compare exact inference

by the Kalman filter (KF)[5] (see Section B.2.6) with the approximation due to Schober *et al.* (2019) (SCH) (see Proposition B.2.3) and the covariance approximation due to Kersting and Hennig (2016) (KER) (see Proposition B.2.4). The integration interval is set to $[0, 10]$ and all methods use an IWP(q) prior for $q = 1, 2, \ldots, 6$, and the initial mean is set to $\mathbb{E}[X^{(j)}(0)] = \Lambda^{j-1} y(0)$ for $j = 1, \ldots, q+1$, with variance set to zero (exact initialisation). The uncertainty of the methods is calibrated by the maximum likelihood method (see Proposition B.4.1), and the methods are examined for 10 step sizes uniformly placed on the interval $[10^{-3}, 10^{-1}]$.

We examine the parameters $\lambda_1 = 0$ and $\lambda_2 = \pi$ (half a revolution per unit of time with no damping). The RMSE is plotted against step size in Figure B.1. It can be seen that SCH is a slightly better than KF and KER for $q = 1$ and small step sizes, and KF becomes slightly better than SCH for large step size while KER becomes significantly worse than both KF and SCH. For $q > 1$, it can be seen that the RMSE is significantly lower for KF than for SCH/KER in general with performance differing between one and two orders of magnitude. Particularly, the superior stability properties of KF are demonstrated (see Theorem B.3.5) for $q > 3$ where both SCH and KER produce massive errors for larger step sizes.

Furthermore, the average χ^2-statistic is shown in Figure B.2. All methods appear to be overconfident for $q = 1$ with SCH performing best, followed by KER. On the other hand, for $1 < q < 5$, SCH and KER remain overconfident for the most part, while KF is underconfident. Our experiments also show that unsurprisingly all methods perform better for smaller $|\lambda_2|$ (frequency of the oscillation). However, we omit visualising this here.

Finally, a demonstration of the error trajectory for the first component of y and the reported uncertainty of the solvers is shown in Figure B.3 for $h = 10^{-2}$ and $q = 2$. Here it can be seen that all methods produce similar errors bars, though SCH and KER produce errors that oscillate far outside their reported uncertainties.

B.5.2 The logistic equation

In this experiment the logistic equation is considered:

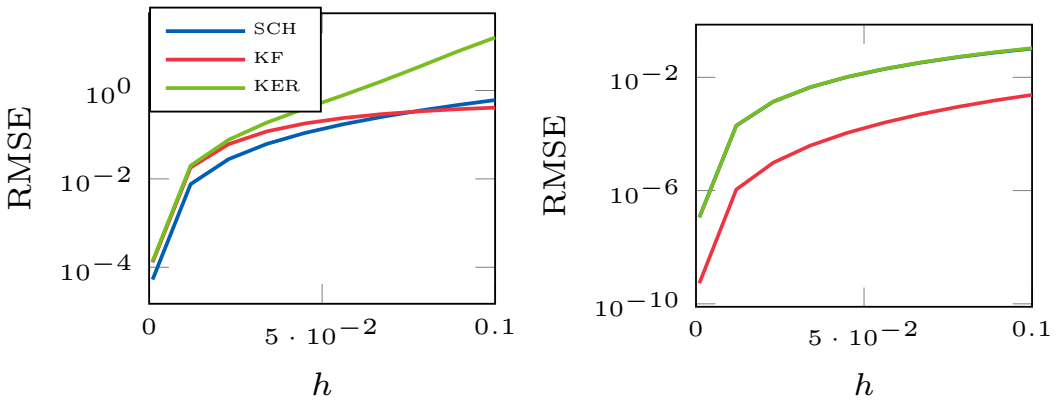
$$\dot{y}(t) = r y(t) \left(1 - y(t)\right), \quad y(0) = 1 \cdot 10^{-1}, \tag{B.45}$$

which has the solution:

$$y(t) = \frac{\exp(rt)}{1/y_0 - 1 + \exp(rt)}. \tag{B.46}$$

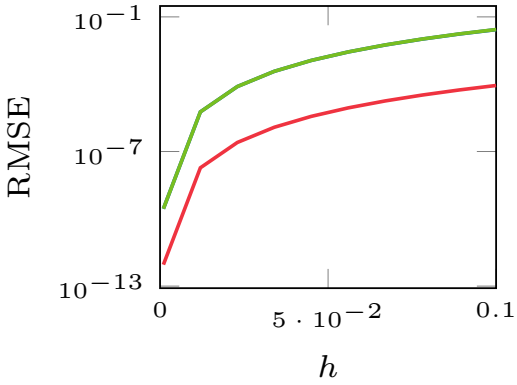
In the experiments r is set to $r = 3$. We compare the zeroth order solver (Proposition B.2.3) (Schober *et al.*, 2019) (SCH), the first order solver in Eq. (B.16) (EKF),

[5] Again note that the EKF and appropriate numerical quadrature methods are equivalent to this estimator here (see Lemma B.2.5).

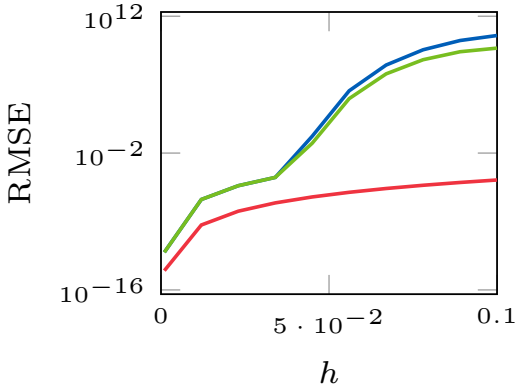


(a) $q = 1$

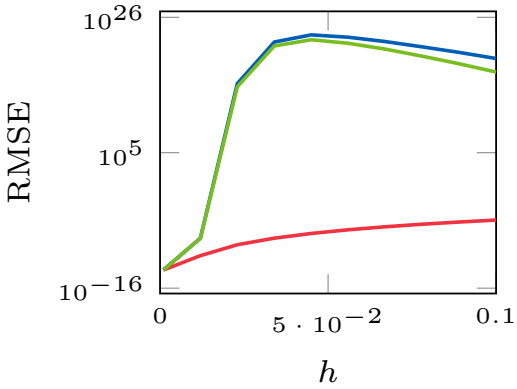
(b) $q = 2$



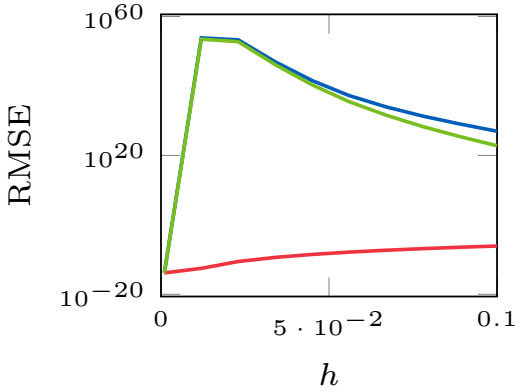
(c) $q = 3$



(d) $q = 4$



(e) $q = 5$



(f) $q = 6$

Figure B.1: RMSE of KF, SCH, and KER on the undamped oscillator using IWP(q) priors for $q = 1, \ldots, 6$ plotted against step size.

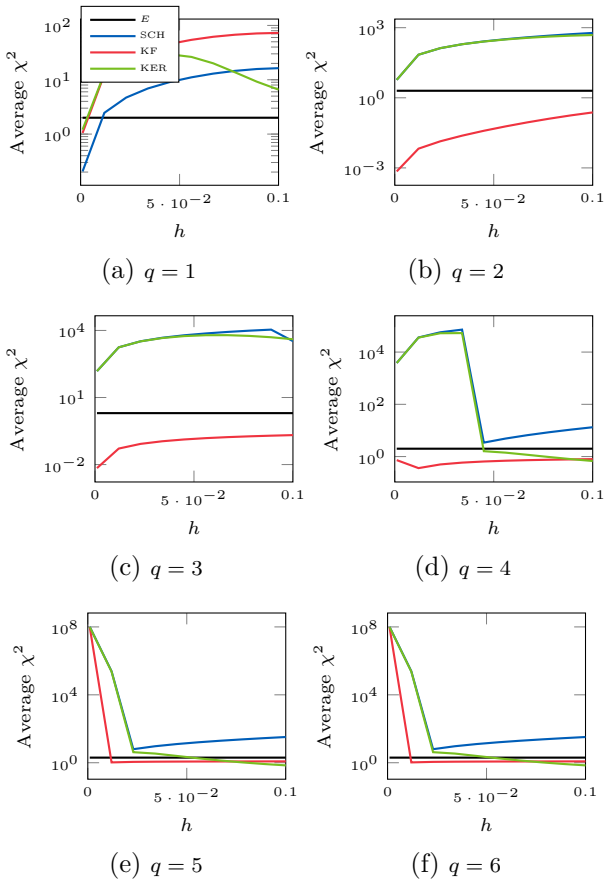


Figure B.2: Average χ^2-statistic of KF, SCH, and KER on the undamped oscillator using IWP(q) priors for $q = 1, \ldots, 6$ plotted against step size. The expected χ^2-statistic is shown in black (E).

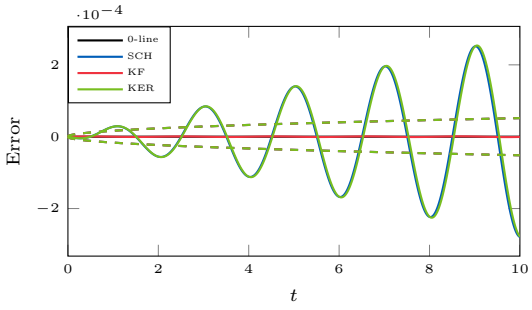


Figure B.3: The errors (solid lines) and ± 2 standard deviation bands (dashed) for KF, SCH, and KER on the undamped oscillator with $q = 2$ and $h = 10^{-2}$. A line at 0 is plotted in solid black.

a numerical integration solver based on the covariance approximation in Proposition B.2.4 (Kersting and Hennig, 2016) (KER), and a numerical integration solver based on approximating Eq. (B.12) (UKF). Both numerical integration approaches use a third degree fully symmetric rule (see McNamee and Stenger, 1967). The integration interval is set to $[0, 2.5]$ and all methods use an IWP(q) prior for $q = 1, 2, \ldots, 4$, and the initial mean of $X^{(1)}$, $X^{(2)}$, and $X^{(3)}$ are set to $y(0)$, $f(y(0))$, and $J_f(y(0))f(y(0))$, respectively (correct values), with zero covariance. The remaining state components $X^{(j)}, j > 3$ are set to zero mean with unit variance. The uncertainty of the methods is calibrated by the quasi maximum likelihood method as explained in Section B.4.3, and the methods are examined for 10 step sizes uniformly placed on the interval $[10^{-3}, 10^{-1}]$.

The RMSE is plotted against step size in Figure B.4. It can be seen that EKF and UKF tend to produce smaller errors by more than an order of magnitude than SCH and KER in general, with the notable exception of the UKF behaving badly for small step sizes and $q = 4$. This is probably due to numerical issues for generating the integration nodes, which requires the computation of matrix square roots (Julier *et al.*, 2000) that can become inaccurate for ill-conditioned matrices. Additionally, the average χ^2-statistic is plotted against step size in Figure B.5. Here it appears that all methods tend to be underconfident for $q = 1, 2$, while SCH becomes overconfident for $q = 3, 4$.

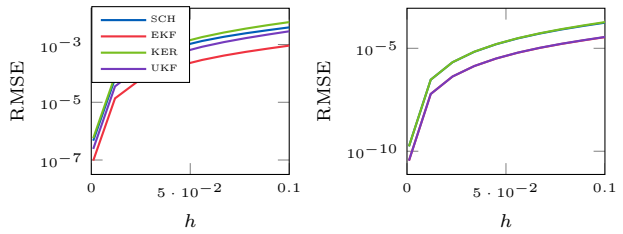
A demonstration of the error trajectory and the reported uncertainty of the solvers is shown in Figure B.3 for $h = 10^{-1}$ and $q = 2$. SCH and KER produce similar errors and they are hard to discern in the figure. The same goes for EKF and UKF. Additionally, it can be seen that the solvers produce qualitatively different uncertainty estimates. While the uncertainty of EKF and UKF first grows to then shrink as the the solution approaches the fixed point at $y(t) = 1$, the uncertainty of SCH grows over the entire interval with the uncertainty of KER growing even faster.

B.5.3 The FitzHugh—Nagumo model

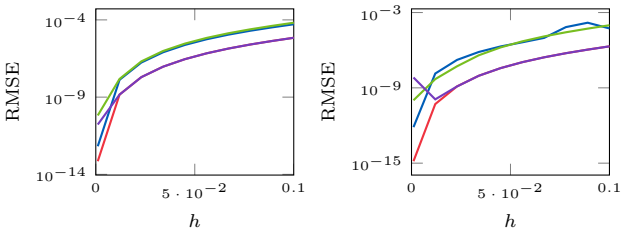
The FitzHugh–Nagumo model is given by:

$$\begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{bmatrix} = \begin{bmatrix} c\left(y_1(t) - \frac{y_1^3(t)}{3} + y_2(t)\right) \\ -\frac{1}{c}\left(y_1(t) - a + b y_2(t)\right) \end{bmatrix}, \tag{B.47}$$

where we set $(a, b, c) = (.2, .2, 3)$ and $y(0) = [-1\ 1]^\intercal$. As previous experiments showed that the behaviour of KER and UKF are similar to SCH and EKF, respectively, we opt for only comparing the latter to increase readability of the presented results. As previously, the moments of $X^{(1)}(0)$, $X^{(2)}(0)$, and $X^{(3)}(0)$ are initialised to their exact values and the remaining derivatives are initialised with zero mean and unit variance. The integration interval is set to $[0, 20]$ and all methods use an IWP(q) prior for $q = 1, \ldots, 4$ and the uncertainty is calibrated as explained in Section B.4.3. A baseline solution is computed using MATLAB's `ode45` function with an absolute tolerance of

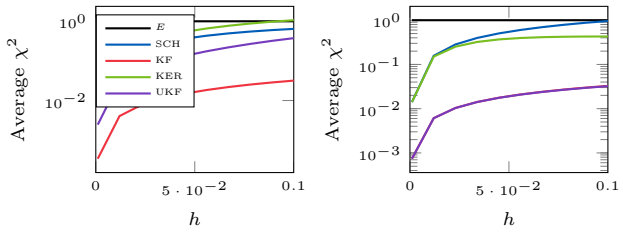


(a) $q = 1$ (b) $q = 2$

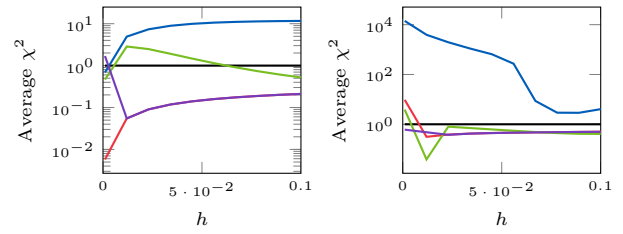


(c) $q = 3$ (d) $q = 4$

Figure B.4: RMSE of SCH, EKF, KER, and UKF on the logistic equation using IWP(q) priors for $q = 1, \ldots, 4$ plotted against step size.



(a) $q = 1$ (b) $q = 2$



(c) $q = 3$ (d) $q = 4$

Figure B.5: Average χ^2-statistic of SCH, EKF, KER, and UKF on the logistic equation using IWP(q) priors for $q = 1, \ldots, 4$ plotted against step size. The expected χ^2-statistic is shown in black (E).

10^{-15} and relative tolerance of 10^{-12}, all errors are computed under the assumption that `ode45` provides the exact solution. The methods are examined for 10 step sizes uniformly

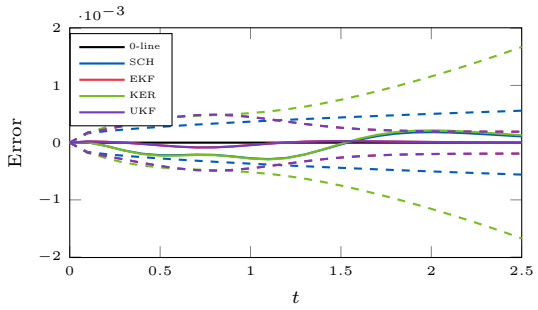


Figure B.6: The errors (solid lines) and ± 2 standard deviation bands (dashed) for KF, SCH, and KER on the logistic with $q = 2$ and $h = 10^{-1}$. A line at 0 is plotted in solid black.

placed on the interval $[10^{-3}, 10^{-1}]$.

The RMSE is shown in Figure B.7. For $q = 1$ EKF produces an error orders of magnitude larger than SCH and for $q = 2$ both methods produce similar errors until the step size grows too large, causing SCH to start producing orders of magnitude larger errors than EKF. For $q = 3, 4$ EKF is superior in producing lower errors and additionally SCH can be seen to become unstable for larger step-sizes (at $h \approx 5 \cdot 10^{-2}$ for $q = 3$ and at $h \approx 2 \cdot 10^{-2}$ for $q = 4$). Furthermore, the averaged χ^2-statistic is shown in Figure B.8. It can be seen that EKF is overconfident for $q = 1$ while SCH is underconfident. For $q = 2$ both methods are underconfident while EKF remains underconfident for $q = 3, 4$ but SCH becomes overconfident for almost all step sizes.

The error trajectory for the first component of y and the reported uncertainty of the solvers is shown in Figure B.9 for $h = 5 \cdot 10^{-2}$ and $q = 2$. It can be seen that both methods have periodically occurring spikes in their errors with EKF being larger in magnitude but also briefer. However, the uncertainty estimate of the EKF is also spiking at the same time giving an adequate assessments of its error. On the other hand, the uncertainty estimate of SCH grows slowly and monotonically over the integration interval, with the error estimate going outside the two standard deviation region at the first spike (slightly hard to see in the figure).

B.5.4 A Bernoulli equation

In this following experiment we consider a transformation of Eq. (B.45), $\eta(t) = \sqrt{y(t)}$, for $r = 2$. The resulting ODE for $\eta(t)$ now has two stable equilibrium points $\eta(t) = \pm 1$ and an unstable equilibrium point at $\eta(t) = 0$. This makes it a simple test domain for different sampling-based ODE solvers, because different types of posteriors ought to arise. We compare the proposed particle filter using both the proposal Eq. (B.24) (PF(1)) and EKF proposals (Eq. (B.25)) (PF(2)) with the method by (Chkrebtii *et al.*, 2016) (CHK) and the one by (Conrad *et al.*, 2017) (CON) for estimating $\eta(t)$ on the

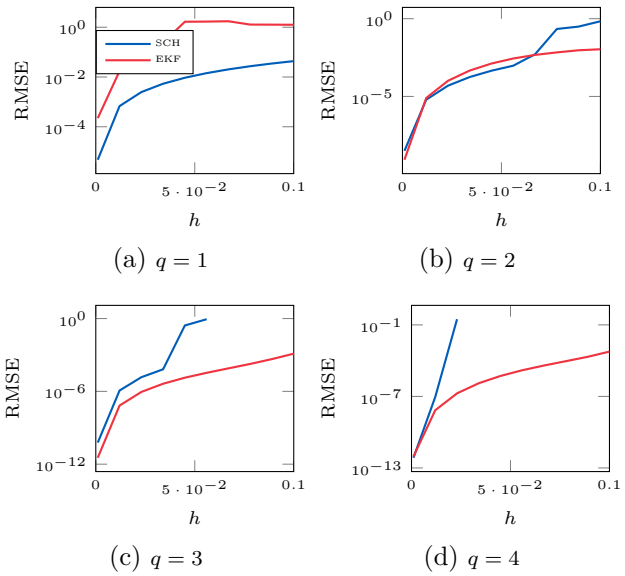


Figure B.7: RMSE of SCH and EKF on the FitzHugh–Nagumo model using IWP(q) priors for $q = 1, \ldots, 4$ plotted against step size.

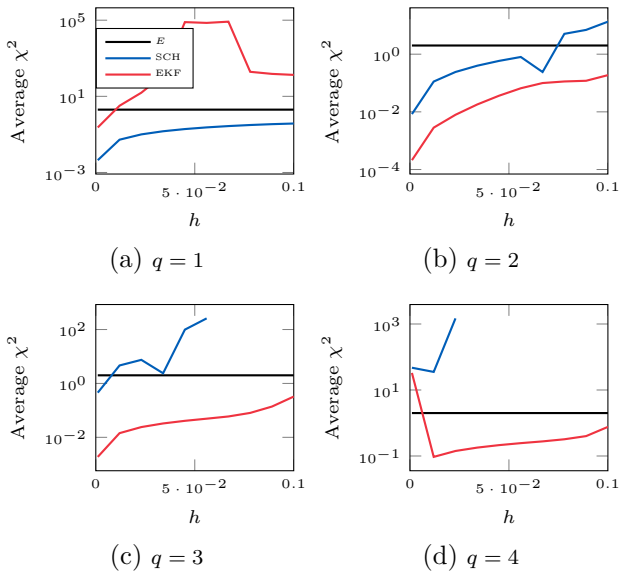


Figure B.8: Average χ^2-statistic of SCH and EKF on the FitzHugh–Nagumo model using IWP(q) priors for $q = 1, \ldots, 4$ plotted against step size.

interval $t \in [0, 5]$ with initial condition set to $\eta_0 = 0$. Both PF and CHK use and IWP(q) prior and set $R = \kappa h^{2q+1}$. CON uses a Runge–Kutta method of order q with perturbation variance $h^{2q+1}/[2q(q!)^2]$ as to roughly match the incremental variance of

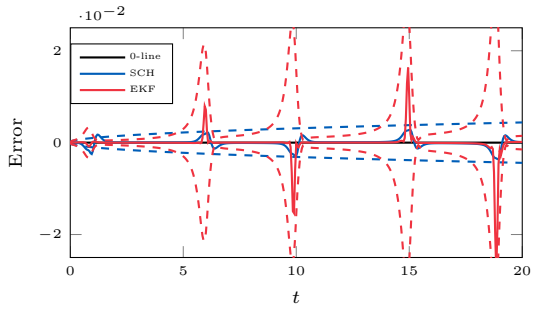


Figure B.9: The errors (solid lines) and ± 2 standard deviation bands (dashed) for KF, SCH, and KER on the FitzHugh–Nagumo model with $q = 2$ and $h = 5 \cdot 10^{-2}$. A line at 0 is plotted in solid black.

the noise entering PF(1), PF(2), and CHK, which is determined by $Q(h)$ and not R.

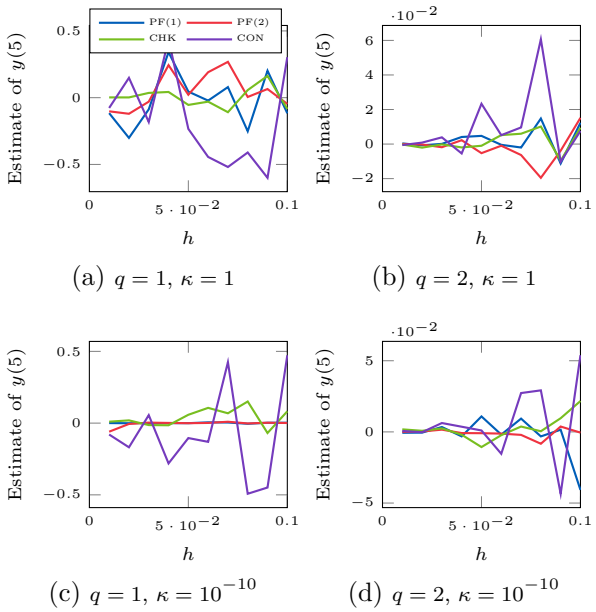
First we attempt to estimate $y(5) = 0$ for 10 step sizes uniformly placed on the interval $[10^{-3}, 10^{-1}]$ with $\kappa = 1$ and $\kappa = 10^{-10}$. All methods use 1000 samples/particles and they estimate $y(5)$ by taking the mean over samples/empirical measures. The estimate of $y(5)$ is plotted against the step size in Figure B.10. In general, the error increases with the step size for all methods, though most easily discerned in Figures B.10b and B.10d. All in all it appears that CHK, PF(1), and PF(2) behave similarly with regards to the estimation, while CON appears to produce a bit larger errors. Furthermore, the effect of κ appears to be the greatest on PF(1) and PF(2) as best illustrated in Figure B.10c.

Additionally, kernel density estimates for the different methods are made for time points $t = 1, 3, 5$ for $\kappa = 1$, $q = 1, 2$ and $h = 10^{-1}, 5 \cdot 10^{-2}$. In Figure B.11 kernel density estimates for $h = 10^{-1}$ are shown. At $t = 1$ all methods produce fairly concentrated unimodal densities that then disperse as time goes on, with CON being a least concentrated and dispersing quicker followed by PF(1)/PF(2) and then last CHK. Furthermore, CON goes bimodal as time goes on, which is best seen in for $q = 1$ in Figure B.11e. On the other hand, the alternatives vary between unimodal (CHK in B.11f, also to some degree PF(1) and PF(2)), bimodal (PF(1) and CHK in Figure B.11e), and even mildly trimodal (PF(2) in Figure B.11e).

Similar behaviour of the methods is observed for $h = 5 \cdot 10^{-2}$ in Figure B.11, though here all methods are generally more concentrated.

B.6 Conclusion and discussion

In this paper, we have presented a novel formulation of probabilistic numerical solution of ODEs as a standard problem in GP regression with a non-linear measurement function, and with measurements that are identically zero. The new model formulation enables the use of standard methods in signal processing to derive new solvers, such as EKF, UKF, and PF. We can also recover many of the previously proposed sequential probabilistic



(a) $q = 1$, $\kappa = 1$ (b) $q = 2$, $\kappa = 1$

(c) $q = 1$, $\kappa = 10^{-10}$ (d) $q = 2$, $\kappa = 10^{-10}$

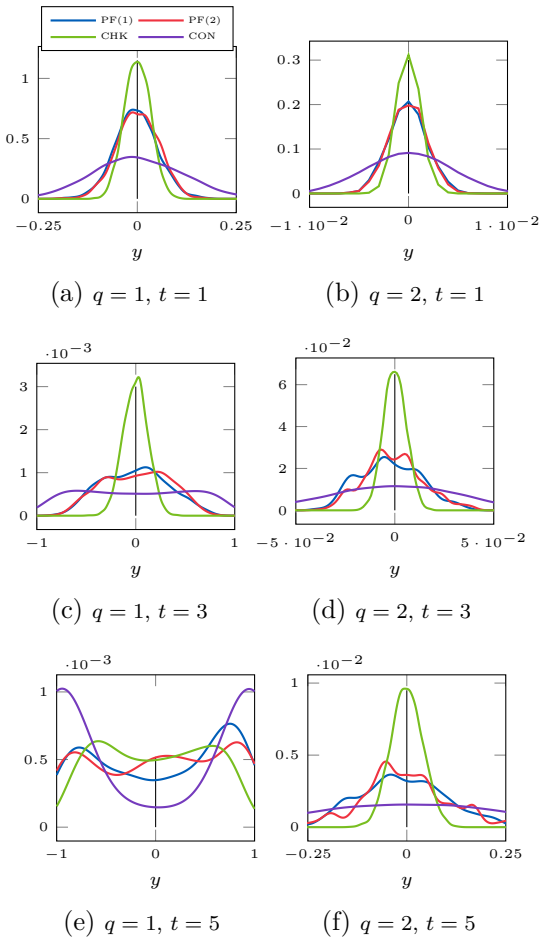
Figure B.10: Sample mean estimate of the solution at $T = 5$.

ODE solvers as special cases.

Additionally, we have demonstrated excellent stability properties of the EKF and UKF on linear test equations, that is, A-stability has been established. The notion of A-stability is closely connected with the solution of stiff equations, which is typically achieved with *implicit* or *semi-implicit* methods (Hairer and Wanner, 1996). In this respect our methods (EKF and UKF) most closely fit into the class of semi-implicit methods such as the methods of Rosenbrock type (Hairer and Wanner, 1996, Chapter IV.7). Though it does seem feasible the proposed methods can be nudged towards the class of implicit methods by means of iterative Gaussian filtering (Bell and Cathey, 1993; Garcia-Fernandez *et al.*, 2015; Tronarp *et al.*, 2018).

While the notion of A-stability has been fairly successful in discerning between methods with good and bad stability properties, it is not the whole story (Alexander, 1977, Section 3). This has lead to other notions of stability such as *L-stability* and *B-stability* (Hairer and Wanner, 1996, Chapter IV.3 and IV.12). It is certainly an interesting question whether the present framework allows for the development of methods satisfying these more strict notions of stability.

An advantage of our model formulation is the decoupling of the prior from the likelihood. Thus future work would involve investigating how well the exact posterior to our inference problem approximates the ODE and then analysing how well different approximate inference strategies behave. However, for $h \to 0$, we expect that the novel Gaussian filters (EKF,UKF) will exhibit polynomial worst-case convergence rates of the mean and its credible intervals, that is, its Bayesian uncertainty estimates, as has already



(a) $q = 1$, $t = 1$ (b) $q = 2$, $t = 1$

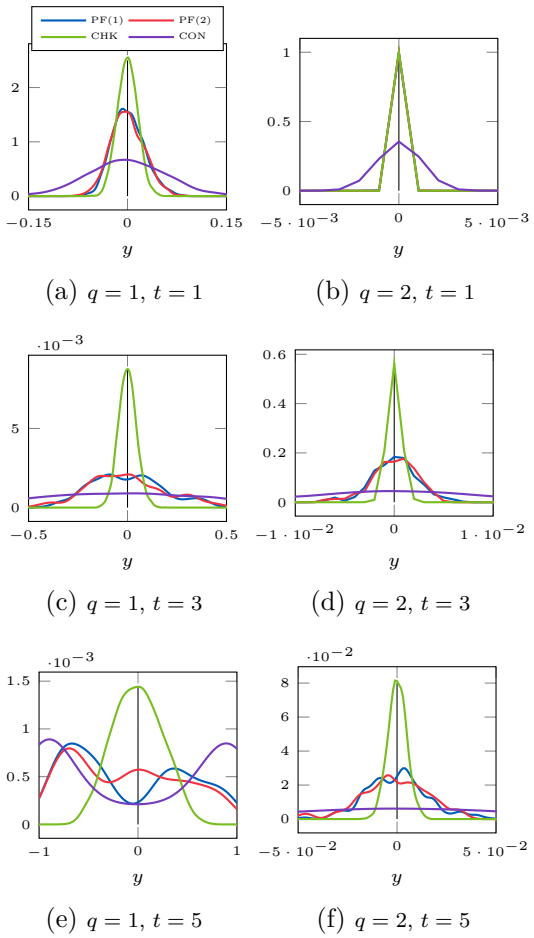
(c) $q = 1$, $t = 3$ (d) $q = 2$, $t = 3$

(e) $q = 1$, $t = 5$ (f) $q = 2$, $t = 5$

Figure B.11: Kernel density estimates of the solution of the Bernoulli equation for $h = 10^{-1}$ and $\kappa = 1$. Mind the different scale of the axes.

been proved in (Kersting *et al.*, 2020a) for 0-th order Taylor-series filters with arbitrary constant measurement variance R (see Section B.2.4).

Our Bayesian recast of ODE solvers might also pave the way toward an average-case analysis of these methods, which has already been executed in (Ritter, 2000) for the special case of Bayesian quadrature. For the PF, a thorough convergence analysis similar to Chkrebtii *et al.* (2016), Conrad *et al.* (2017), Abdulle and Garegnani (2020) and Del Moral (2004) appears feasible. However, the results on spline approximations for ODEs (see, e.g., Loscalzo and Talbot, 1967) might also apply to the present methodology via the correspondence between GP regression and spline function approximations (Kimeldorf and Wahba, 1970).



(a) $q = 1$, $t = 1$ (b) $q = 2$, $t = 1$

(c) $q = 1$, $t = 3$ (d) $q = 2$, $t = 3$

(e) $q = 1$, $t = 5$ (f) $q = 2$, $t = 5$

Figure B.12: Kernel density estimates of the solution of the Bernoulli equation for $h = 5 \cdot 10^{-2}$ and $\kappa = 1$. Mind the different scale of the axes.

Acknowledgements

This material was developed, in part, at the *Prob Num 2018* workshop hosted by the Lloyd's Register Foundation programme on Data-Centric Engineering at the Alan Turing Institute, UK, and supported by the National Science Foundation, USA, under Grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the above-named funding bodies and research institutions. Filip Tronarp gratefully acknowledge financial support by Aalto ELEC Doctoral School. Additionally, Filip Tronarp and Simo Särkkä gratefully acknowledge financial support by Academy of Finland grant #313708. Hans Kersting and Philipp Hennig gratefully acknowledge financial support by the German Federal Ministry of Education and Research through BMBF grant 01IS18052B (ADIMEM). Philipp Hennig also gratefully acknowledges support through ERC StG Action 757275

/ PANAMA. Finally, the authors would like to thank the editor and the reviewers for their help in improving the quality of this manuscript.

Supplementary Material for Tronarp *et al.* (2019a)

B.7 Supplement I: Proof of Proposition B.2.1

In this section we prove Proposition B.2.1. First note that, by Eq. (B.4), we have

$$\frac{\mathrm{d}\mathbb{C}\Big[X^{(1)}(t), X^{(2)}(s)\Big]}{\mathrm{d}t} = \mathbb{C}\Big[X^{(2)}(t), X^{(2)}(s)\Big], \tag{B.48}$$

where $\mathbb{C}$ is the cross-covariance operator. That is the cross-covariance matrix between $X^{(1)}(t)$ and $X^{(2)}(t)$ is just the integral of the covariance matrix function of $X^{(2)}$. Now define

$$(\mathbf{X}^{(i)})^\mathsf{T} = \begin{bmatrix} \left(X_1^{(i)}\right)^\mathsf{T} & \dots & \left(X_N^{(i)}\right)^\mathsf{T} \end{bmatrix}, \ i = 1, \dots, q+1, \tag{B.49a}$$

$$\mathbf{g}^\mathsf{T} = \begin{bmatrix} g^\mathsf{T}(h) & \dots & g^\mathsf{T}(Nh) \end{bmatrix}, \tag{B.49b}$$

$$\mathbf{z}^\mathsf{T} = \begin{bmatrix} z_1^\mathsf{T} & \dots & z_N^\mathsf{T} \end{bmatrix}. \tag{B.49c}$$

Since Equation (B.3) defines a Gaussian process we have that $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are jointly Gaussian distributed and from Eq. (B.48) the blocks of $\mathbb{C}[\mathbf{X}^{(1)}, \mathbf{X}^{(2)}]$ are given by

$$\mathbb{C}\Big[\mathbf{X}^{(1)}, \mathbf{X}^{(2)}\Big]_{n,m} = \int_0^{nh} \mathbb{C}\Big[X^{(2)}(t), X^{(2)}(mh)\Big] \, \mathrm{d}t$$

which is precisely the kernel mean, with respect to the Lebesgue measure on $[0, nh]$, evaluated at mh, see (Briol *et al.*, 2019, Section 2.2). Furthermore,

$$\mathbb{V}\Big[\mathbf{X}^{(2)}\Big]_{n,m} = \mathbb{C}\Big[X^{(2)}(nh), X^{(2)}(mh)\Big],$$

that is, the covariance matrix function (referred to as kernel matrix in Bayesian quadrature literature (Briol *et al.*, 2019)) evaluated at all pairs in $\{h, \dots, Nh\}$. From Gaussian conditioning rules we have for the conditional means and covariance matrices given

$\mathbf{X}^{(2)} - \mathbf{g} = 0$, denoted by $\mathbb{E}_{\mathcal{D}}[X^{(1)}(nh)]$ and $\mathbb{V}_{\mathcal{D}}[X^{(1)}(nh)]$, respectively, that

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\Big[X^{(1)}(nh)\Big] &= \mathbb{E}\Big[X^{(1)}(nh)\Big] + \mathbf{w}_n\Big(\mathbf{z} + \mathbf{g} - \mathbb{E}\Big[\mathbf{X}^{(2)}\Big]\Big) \\
&= \mathbb{E}\Big[X^{(1)}(nh)\Big] + \mathbf{w}_n\Big(\mathbf{g} - \mathbb{E}\Big[\mathbf{X}^{(2)}\Big]\Big), \\
\mathbb{V}_{\mathcal{D}}\Big[X^{(1)}(nh)\Big] &= \mathbb{V}\Big[X^{(1)}(nh)\Big] - \mathbf{w}_n \mathbb{V}\Big[\mathbf{X}^{(2)}\Big]\mathbf{w}_n^{\mathsf{T}},
\end{aligned}$$

where we used the fact that $\mathbf{z} = 0$ by definition and $\mathbf{w}_n$ are the Bayesian quadrature weights associated to the integral of g over the domain $[0, nh]$, given by (see Briol *et al.* (2019, Proposition 1))

$$\mathbf{w}_n^{\mathsf{T}} = \mathbb{V}\Big[\mathbf{X}^{(2)}\Big]^{-1} \begin{bmatrix} \mathbb{C}\Big[X^{(1)}(nh), X^{(2)}(h)\Big]^{\mathsf{T}} \\ \vdots \\ \mathbb{C}\Big[X^{(1)}(nh), X^{(2)}(Nh)\Big]^{\mathsf{T}} \end{bmatrix}.$$

□

B.8 Supplement II: Proof of Proposition B.2.4

To prove Proposition B.2.4, expand the expressions for S_n and K_n as given by Eq. (B.12):

$$\begin{aligned}
S_n &= \dot{C}\Sigma_n^P\dot{C}^{\mathsf{T}} + \mathbb{V}\Big[f(CX_n, t_n) \mid z_{1:n-1}\Big] \\
&\quad - \dot{C}\mathbb{C}\Big[X_n, f(CX_n, t_n) \mid z_{1:n-1}\Big] \\
&\quad - \mathbb{C}\Big[X_n, f(CX_n, t_n) \mid z_{1:n-1}\Big]^{\mathsf{T}}\dot{C}^{\mathsf{T}} \\
&\approx \dot{C}\Sigma_n^P\dot{C}^{\mathsf{T}} + \mathbb{V}\Big[f(CX_n, t_n) \mid z_{1:n-1}\Big] \\
K_n &= \Big(\Sigma_n^P\dot{C}^{\mathsf{T}} - \mathbb{C}\Big[X_n, f(CX_n, t_n) \mid z_{1:n-1}\Big]\Big)S_n^{-1} \\
&\approx \Sigma_n^P\dot{C}^{\mathsf{T}}S_n^{-1},
\end{aligned}$$

where in the second steps the approximation $\mathbb{C}[X_n, f(CX_n, t_n) \mid z_{1:n-1}] \approx 0$ was used. Lastly, recall that $z_n \triangleq 0$, hence the update equations become

$$S_n \approx \dot{C}\Sigma_n^P\dot{C}^{\mathsf{T}} + \mathbb{V}\Big[f(CX_n, t_n) \mid z_{1:n-1}\Big], \tag{B.52a}$$

$$K_n \approx \Sigma_n^P\dot{C}^{\mathsf{T}}S_n^{-1}, \tag{B.52b}$$

$$\mu_n^F \approx \mu_n^P + K_n\Big(\mathbb{E}\Big[f(CX_n, t_n) \mid z_{1:n-1}\Big] - \dot{C}\mu_n^P\Big), \tag{B.52c}$$

$$\Sigma_n^F \approx \Sigma_n^P - K_nS_nK_n^{\mathsf{T}}. \tag{B.52d}$$

When $\mathbb{E}[f(CX_n, t_n) \mid z_{1:n-1}]$ and $\mathbb{V}[f(CX_n, t_n) \mid z_{1:n-1}]$ are approximated by Bayesian quadrature using a squared exponential kernel and a uniform set of nodes translated and scaled by μ_n^P and Σ_n^P, respectively, the method of Kersting and Hennig (2016) is obtained. □

B.9 Supplement III: Proof of Proposition B.4.1

Note that $(\breve{\mu}_n^F, \breve{\Sigma}_n^F)$ is the output of a misspecified Kalman filter (Tronarp *et al.*, 2019b, Algorithm 1). We indicate that a quantity from Eqs. (B.11) and (B.12) is computed by the misspecified Kalman filter by $\breve{\cdot}$. For example $\breve{\mu}_n^P$ is the predictive mean of the misspecified Kalman filter. If $\Sigma_n^F = \sigma^2 \breve{\Sigma}_n^F$ and $\breve{\mu}_n^F = \mu_n^F$ holds then for the prediction step we have

$$\begin{aligned}
\mu_{n+1}^P &= A(h)\mu_n^F + \xi(h) = A(h)\breve{\mu}_n^F + \xi(h) = \breve{\mu}_{n+1}^P, \\
\Sigma_{n+1}^P &= A(h)\Sigma_n^F A^{\mathsf{T}}(h) + Q(h), \\
&= \sigma^2 \Big(A(h)\breve{\Sigma}_n^F A^{\mathsf{T}}(h) + \breve{Q}(h) \Big), \\
&= \sigma^2 \breve{\Sigma}_{n+1}^P,
\end{aligned}$$

where we used the fact that $Q(h) = \sigma^2 \breve{Q}(h)$, which follows from $L = \sigma \breve{L}$ and Eq. (B.8). Furthermore, recall that $H_{n+1} = \dot{C} - \Lambda(t_{n+1})C$, which for the update gives

$$\begin{aligned}
S_{n+1} &= H_{n+1}\Sigma^P_{n+1}H^\mathsf{T}_{n+1} \\
&= \sigma^2 H_{n+1}\breve{\Sigma}^P_{n+1}H^\mathsf{T}_{n+1} \\
&= \sigma^2 \breve{S}_{n+1}. \\
K_{n+1} &= \Sigma^P_{n+1}H^\mathsf{T}_{n+1}S^{-1}_{n+1} \\
&= \sigma^2\breve{\Sigma}^P_{n+1}H^\mathsf{T}_{n+1}[\sigma^2\breve{S}_{n+1}]^{-1} \\
&= \breve{\Sigma}^P_{n+1}H^\mathsf{T}_{n+1}\breve{S}^{-1}_{n+1} \\
&= \breve{K}_{n+1}. \\
\hat{z}_{n+1} &= H_{n+1}\mu^P_{n+1} - \zeta(t_n) \\
&= H_{n+1}\breve{\mu}^P_{n+1} - \zeta(t_n) \\
&= \breve{z}_{n+1}, \\
\mu^F_{n+1} &= \mu^P_{n+1} + K_{n+1}(z_{n+1} - \hat{z}_{n+1}) \\
&= \breve{\mu}^P_{n+1} + \breve{K}_{n+1}(z_{n+1} - \breve{z}_{n+1}) \\
&= \breve{\mu}^F_{n+1}. \\
\Sigma^F_{n+1} &= \Sigma^P_{n+1} - K_{n+1}S_{n+1}K^\mathsf{T}_{n+1} \\
&= \sigma^2\Big(\breve{\Sigma}^P_{n+1} - \breve{K}_{n+1}\breve{S}_{n+1}\breve{K}^\mathsf{T}_{n+1}\Big) \\
&= \sigma^2\breve{\Sigma}^F_{n+1}.
\end{aligned}$$

It thus follows by induction that $\mu^F_n = \breve{\mu}^F_n$, $\Sigma^F_n = \sigma^2\breve{\Sigma}^F_n$, $\hat{z}_n = \breve{z}_n$, and $S_n = \sigma^2\breve{S}_n$ for $n \geq 0$. From Eq. (B.40) we have that the log-likelihood is given by

$$\begin{aligned}
\log p(z_{1:N}) &= \log \prod_{n=1}^{N} \mathcal{N}(z_n; \hat{z}_n, S_n) \\
&= \log \prod_{n=1}^{N} \mathcal{N}(z_n; \breve{z}_n, \sigma^2\breve{S}_n) \\
&= -\frac{Nd}{2}\log\sigma^2 - \sum_{n=1}^{N}\frac{(z_n - \breve{z}_n)^\mathsf{T}\breve{S}^{-1}_n(z_n - \breve{z}_n)}{2\sigma^2}.
\end{aligned}$$

Taking the derivative of log-likelihood with respect to σ^2 and setting it to zero gives the following estimating equation

$$0 = -\frac{Nd}{2\sigma^2} + \frac{1}{2(\sigma^2)^2}\sum_{n=1}^{N}(z_n - \breve{z}_n)^\mathsf{T}\breve{S}^{-1}_n(z_n - \breve{z}_n),$$

which has the following solution

$$\sigma^2 = \frac{1}{Nd} \sum_{n=1}^{N} (z_n - \breve{z}_n)^{\mathsf{T}} \breve{S}_n^{-1} (z_n - \breve{z}_n).$$

□

C Convergence Rates of Gaussian ODE Filters (Kersting *et al.*, 2020a)

Abstract: A recently-introduced class of probabilistic (uncertainty-aware) solvers for ordinary differential equations (ODEs) applies Gaussian (Kalman) filtering to initial value problems. These methods model the true solution x and its first q derivatives *a priori* as a Gauss–Markov process $\boldsymbol{X}$, which is then iteratively conditioned on information about $\dot{x}$. This article establishes worst-case local convergence rates of order $q+1$ for a wide range of versions of this Gaussian ODE filter, as well as global convergence rates of order q in the case of $q=1$ and an integrated Brownian motion prior, and analyses how inaccurate information on $\dot{x}$ coming from approximate evaluations of f affects these rates. Moreover, we show that, in the globally convergent case, the posterior credible intervals are well calibrated in the sense that they globally contract at the same rate as the truncation error. We illustrate these theoretical results by numerical experiments which might indicate their generalizability to $q \in \{2, 3, \dots\}$.

C.1 Introduction

A solver of an initial value problem (IVP) outputs an approximate solution $\hat{x}\colon [0,T] \to \mathbb{R}^d$ of an ordinary differential equation (ODE) with initial condition:

$$\begin{aligned} x^{(1)}(t) \coloneqq \frac{\mathrm{d}x}{\mathrm{d}t}(t) = f\left(x(t)\right), \qquad & \forall t \in [0,T], \\ x(0) = x_0 \ \in \mathbb{R}^d. & \end{aligned} \tag{C.1}$$

(Without loss of generality, we simplify the presentation by restricting attention to the autonomous case.) The numerical solution $\hat{x}$ is computed by iteratively collecting information on $x^{(1)}(t)$ by evaluating $f\colon \mathbb{R}^d \to \mathbb{R}^d$ at a numerical estimate $\hat{x}(t)$ of $x(t)$ and using these approximate evaluations of the time derivative to extrapolate along the time axis. In other words, the numerical solution (or *estimator*) $\hat{x}$ of the exact solution (or *estimand*) x is calculated based on evaluations of the vector field f (or *data*). Accordingly, we treat $\hat{x}$ itself as an estimator, i.e. a statistic that translates evaluations of f

into a probability distribution over $C^1([0,T];\mathbb{R}^d)$, the space of continuously differentiable functions from $[0,T]$ to $\mathbb{R}^d$.

This probabilistic interpretation of numerical computations of tractable from intractable quantities as statistical inference of latent from observable quantities applies to all numerical problems and has been repeatedly recommended in the past (Poincaré, 1896; Diaconis, 1988; Skilling, 1991; O'Hagan, 1992; Ritter, 2000). It employs the language of probability theory to account for the epistemic uncertainty (i.e. limited knowledge) about the accuracy of intermediate and final numerical computations, thereby yielding algorithms which can be more aware of—as well as more robust against—uncertainty over intermediate computational results. Such algorithms can output probability measures, instead of point estimates, over the final quantity of interest. This approach, now called *probabilistic numerics (PN)* (Hennig *et al.*, 2015; Oates and Sullivan, 2019), has in recent years been spelled out for a wide range of numerical tasks, including linear algebra, optimization, integration and differential equations, thereby working towards the long-term goal of a coherent framework to propagate uncertainty through chained computations, as desirable, e.g., in statistical machine learning.

In this paper, we determine the convergence rates of a recent family of PN methods (Schober *et al.*, 2014; Kersting and Hennig, 2016; Magnani *et al.*, 2017; Schober *et al.*, 2019; Tronarp *et al.*, 2019a) which recast an IVP as a *stochastic filtering problem* (Øksendal, 2003, Chapter 6), an approach that has been studied in other settings (Jazwinski, 1970), but has not been applied to IVPs before. These methods assume *a priori* that the solution x and its first $q \in \mathbb{N}$ derivatives follow a Gauss–Markov process $\boldsymbol{X}$ that solves a stochastic differential equation (SDE).

The evaluations of f at numerical estimates of the true solution can then be regarded as imperfect evaluations of $\dot{x}$, which can then be used for a Bayesian update of $\boldsymbol{X}$. Such recursive updates along the time axis yield an algorithm whose structure resembles that of Gaussian (Kalman) filtering (Särkkä, 2013, Chapter 4). These methods add only slight computational overhead compared to classical methods (Schober *et al.*, 2019) and have been shown to inherit local convergence rates from equivalent classical methods in specific cases (Schober *et al.*, 2014; Schober *et al.*, 2019). These equivalences (i.e. the equality of the filtering posterior mean and the classical method) are only known to hold in the case of the integrated Brownian motion (IBM) prior and noiseless evaluations of f (in terms of our later notation, the case $R \equiv 0$), as well as under the following restrictions:

Firstly, for $q \in \{1,2,3\}$, and if the first step is divided into sub-steps resembling those of Runge–Kutta methods, an equivalence of the posterior mean of the first step of the filter and the explicit Runge–Kutta method of order q was established in Schober *et al.* (2014) (but for $q \in \{2,3\}$ only in the limit as the initial time of the IBM tends to $-\infty$). Secondly, it was shown by Schober *et al.* (2019) that, for $q = 1$, the posterior mean after each step coincides with the trapezoidal rule if it takes an additional evaluation of f at the end of each step, known as P(EC)1. The same paper shows that, for $q = 2$, the filter coincides with a third-order Nordsieck method (Nordsieck, 1962) if the filter is in the

steady state, i.e. after the sequence of error covariance matrices has converged. These results neither cover filters with the integrated Ornstein–Uhlenbeck process (IOUP) prior (Magnani *et al.*, 2017) nor non-zero noise models on evaluations of f.

In this paper, we directly prove convergence rates without first fitting the filter to existing methods, and thereby lift many of the above restrictions on the convergence rates. While the more-recent work by Tronarp *et al.* (2020) also provide convergence rates of estimators of x in the Bayesian ODE filtering/smoothing para-digm, they concern the maximum a posteriori estimator (as computed by the iterated extended Kalman ODE smoother), and therefore differ from our convergence rates of the filtering mean (as computed by the Kalman ODE filter).

C.1.1 Contribution

Our main results—Theorems C.6.2 and C.7.7—provide local and global convergence rates of the ODE filter when the step size h goes to zero. Theorem C.6.2 shows local convergence rates of h^{q+1} without the above-mentioned previous restrictions—i.e. for a generic Gaussian ODE filter for all $q \in \mathbb{N}$, both IBM and IOUP prior, flexible Gaussian initialization (see Assumptions C.2 and C.3), and arbitrary evaluation noise $R \geq 0$. As a first global convergence result, Theorem C.7.7 establishes global convergence rates of h^q in the case of $q = 1$, the IBM prior and all fixed measurement uncertainty models R of order $p \in [1, \infty]$ (see Assumption C.4). This global rate of the worst-case error is matched by the contraction rate of the posterior credible intervals, as we show in Theorem C.8.1. Moreover, we also give closed-form expressions for the steady states in the global case and illustrate our results as well as their possible generalizability to $q \geq 2$ by experiments in Appendix C.9.

C.1.2 Related work on probabilistic ODE solvers

The Gaussian ODE filter can be thought of as a self-consistent Bayesian decision agent that iteratively updates its prior belief $\boldsymbol{X}$ over $x\colon [0, T] \to \mathbb{R}^d$ (and its first q derivatives) with information on $\dot{x}$ from evaluating f.[1] For Gauss–Markov priors, it performs exact Bayesian inference and optimally (with respect to the L^2-loss) extrapolates along the time axis. Accordingly, all of its computations are deterministic and—due to its restriction to Gaussian distributions—only slightly more expensive than classical solvers. Experiments demonstrating competitive performance with classical methods are provided in Schober *et al.* (2019, Section 5).

[1] Here, the word 'Bayesian' describes the algorithm in the sense that it employs a prior over the quantity of interest and updates it by Bayes rule according to a prespecified measurement model (as also used in Skilling (1991); Chkrebtii *et al.* (2016); Kersting and Hennig (2016)). The ODE filter is not Bayesian in the stronger sense of Cockayne *et al.* (2019), and it remains an open problem to construct a Bayesian solver in this strong sense without restrictive assumptions, as discussed in Wang *et al.* (2018).

Another line of work (comprising the methods from Chkrebtii *et al.* (2016); Conrad *et al.* (2017); Teymur *et al.* (2016); Lie *et al.* (2019); Abdulle and Garegnani (2020); Teymur *et al.* (2018)) introduces probability measures to ODE solvers in a fundamentally different way—by representing the distribution of all numerically possible trajectories with a set of sample paths. To compute these sample paths, Chkrebtii *et al.* (2016) draws them from a (Bayesian) Gaussian process (GP) regression; Conrad *et al.* (2017); Teymur *et al.* (2016); Lie *et al.* (2019); Teymur *et al.* (2018) perturb classical estimates after an integration step with a suitably scaled Gaussian noise; and Abdulle and Garegnani (2020) perturbs the classical estimate instead by choosing a stochastic step-size. While Conrad *et al.* (2017); Teymur *et al.* (2016); Lie *et al.* (2019); Abdulle and Garegnani (2020); Teymur *et al.* (2018) can be thought of as (non-Bayesian) 'stochastic wrappers' around classical solvers, which produce samples with the same convergence rate, Chkrebtii *et al.* (2016) employs—like the filter—GP regression to represent the belief on x. While the Gaussian ODE filter can convergence with polynomial order (see results in this paper), However, Chkrebtii *et al.* (2016) only show first-order convergence rates and also construct a sample representation of numerical errors, from which samples are drawn iteratively. A conceptual and experimental comparison between the filter and Chkrebtii *et al.* (2016) can be found in Schober *et al.* (2019). An additional numerical test against Conrad *et al.* (2017) was given by Kersting and Hennig (2016). Moreover, Tronarp *et al.* (2019a) recently introduced a particle ODE filter, which combines a filtering-based solver with a sampling-based uncertainty quantification (UQ), and compared it numerically with Conrad *et al.* (2017) and Chkrebtii *et al.* (2016).

All of the above sampling-based methods can hence represent more expressive, non-Gaussian posteriors (as e.g. desirable for bifurcations), but multiply the computational cost of the underlying method by the number of samples. ODE filters are, in contrast, not a perturbation of known methods, but novel methods designed for computational speed and for a robust treatment of intermediate uncertain values (such as the evaluations of f at estimated points). Unless parallelization of the samples in the sampling-based solvers is possible and inexpensive, one can spend the computational budget for generating additional samples on dividing the step size h by the number of samples, and can thereby polynomially decrease the error. Its Gaussian UQ, however, should not be regarded as the true UQ—in particular for chaotic systems whose uncertainty can be better represented by sampling-based solvers, see e.g. Conrad *et al.* (2017, Figure 1) and Abdulle and Garegnani (2020, Figure 2)—but as a rough inexpensive probabilistic treatment of intermediate values and final errors which is supposed to, on average, guide the posterior mean towards the true x. Therefore, it is in a way more similar to classical non-stochastic solvers than to sampling-based stochastic solvers and, unlike sampling-based solvers, puts emphasis on computational speed over statistical accuracy. Nevertheless, its Gaussian UQ is sufficient to make the forward models in ODE inverse problems more 'uncertainty-aware'; see Kersting *et al.* (2020b, Section 3).

Accordingly, the convergence results in this paper concern the convergence rate of the posterior mean to the true solution, while the theoretical results from Teymur *et al.*

(2016); Chkrebtii *et al.* (2016); Conrad *et al.* (2017); Lie *et al.* (2019); Abdulle and Garegnani (2020); Teymur *et al.* (2018) provide convergence rates of the variance of the non-Gaussian empirical measure of samples (and not for an individual sample).

C.1.3 Relation to filtering theory

While Gaussian (Kalman) filtering was first applied to the solution of ODEs by Kersting and Hennig (2016) and Schober *et al.* (2019), it has previously been analysed in the filtering, data assimilation as well as linear system theory community. The convergence results in this paper are concerned with its asymptotics when the step size h (aka time step between data points) goes to zero. In the classical filtering setting, where the data comes from an external sensor, this quantity is not treated as a variable, as it is considered a property of the data and not, like in our case, of the algorithm. Accordingly, the standard books lack such an analysis for $h \to 0$—see Jazwinski (1970); Anderson and Moore (1979); Maybeck (1979) for filtering, Law *et al.* (2015); Reich and Cotter (2015) for data assimilation and Callier and Desoer (1991) for linear system theory—and we believe that our convergence results are completely novel. It is conceivable that, also for these communities, this paper may be of interest in settings where the data collection mechanism can be actively chosen, e.g. when the frequency of the data can be varied or sensors of different frequencies can be used.

C.1.4 Outline

The paper begins with a brief introduction to Gaussian ODE filtering in Appendix C.2. Next, Appendices C.3 and C.5 provide auxiliary bounds on the flow map of the ODE and on intermediate quantities of the filter respectively. With the help of these bounds, Appendices C.6 and C.7 establish local and global convergence rates of the filtering mean respectively. In light of these rates, Appendix C.8 analyses for which measurement noise models the posterior credible intervals are well calibrated. These theoretical results are experimentally confirmed and discussed in Appendix C.9. Appendix C.10 concludes with a high-level discussion.

C.1.5 Notation

We will use the notation $[n] \coloneqq \{0, \dots, n-1\}$. For vectors and matrices, we will use zero-based numbering, e.g. $x = (x_0, \dots, x_{d-1}) \in \mathbb{R}^d$. For a matrix $P \in \mathbb{R}^{n \times m}$ and $(i, j) \in [n] \times [m]$, we will write $P_{i,:} \in \mathbb{R}^{1 \times m}$ for the i^{th} row and $P_{:,j}$ for the j^{th} column of P. A fixed but arbitrary norm on $\mathbb{R}^d$ will be denoted by $\|\cdot\|$. The minimum and maximum of two real numbers a and b will be denoted by $a \wedge b$ and $a \vee b$ respectively. Vectors that span all q modeled derivatives will be denoted by bold symbols, such as $\boldsymbol{x}$.

C.2 Gaussian ODE filtering

This section defines how a Gaussian filter can solve the IVP eq. (C.1). In the various subsections, we first explain the choice of prior on x, then describe how the algorithm computes a posterior output from this prior (by defining a numerical integrator $\mathbf{\Psi}$), and add explanations on the measurement noise of the derivative observations. To alternatively understand how this algorithm can be derived as an extension of generic Gaussian filtering in probabilistic state space models, see the concise presentation in (Kersting *et al.*, 2020b, Supplement A).

C.2.1 Prior on x

In PN, it is common (Hennig *et al.*, 2015, Section 3(a)) to put a prior measure on the unknown solution x. Often, for fast Bayesian inference by linear algebra (Rasmussen and Williams, 2006, Chapter 2), this prior is Gaussian. To enable GP inference in linear time by Kalman filtering (Särkkä, 2013, Chapter 4.3), we further restrict the prior to Markov processes. As discussed in Särkkä and Solin (2019, Chapter 12.4), a wide class of such Gauss–Markov processes can be captured by a law of the (strong) solution (Øksendal, 2003, Chapter 5.3) of a linear SDE with Gaussian initial condition. Here—as we, by eq. (C.1), have information on at least one derivative of x—the prior also includes the first $q \in \mathbb{N}$ derivatives. Therefore, for all $j \in [d]$, we define the vector of time derivatives by $\boldsymbol{X}_j = \left(X_j^{(0)}, \ldots, X_j^{(q)}\right)^\intercal$. We define $\boldsymbol{X}_j$ as a $(q+1)$-dimensional stochastic process via the SDE

$$\begin{aligned} \mathrm{d}\boldsymbol{X}_j(t) &= \left(\mathrm{d}X_j^{(0)}(t), \ldots, \mathrm{d}X_j^{(q-1)}(t), \mathrm{d}X_j^{(q)}(t)\right)^\intercal \\ &= \begin{pmatrix} 0 & 1 & 0 \ldots & 0 \\ \vdots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & 1 \\ c_0 & \ldots & \ldots & c_q \end{pmatrix} \begin{pmatrix} X_j^{(0)}(t) \\ \vdots \\ X_j^{(q-1)}(t) \\ X_j^{(q)}(t) \end{pmatrix} \mathrm{d}t + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sigma_j \end{pmatrix} \mathrm{d}B_j(t), \end{aligned} \tag{C.2}$$

driven by mutually independent one-dimensional Brownian motions $\{B_j;\ j \in [d]\}$ (independent of $\boldsymbol{X}(0)$) scaled by $\sigma_j > 0$, with initial condition $\boldsymbol{X}_j(0) \sim \mathcal{N}(m_j(0), P_j(0)$). We assume that $\left\{X_j(0);\ j \in [d]\right\}$ are independent. In other words, we model the unknown i^{th} derivative of the j^{th} dimension of the solution x of the IVP eq. (C.1), denoted by $x_j^{(i)}$, as a draw from a real-valued, one-dimensional GP $X_j^{(i)}$, for all $i \in [q+1]$ and $j \in [d]$, such that $X_j^{(q)}$ is defined by $(c_0, \ldots, c_q)$ as well as the Brownian motion scale σ_j and $X_j^{(i-1)}$ is defined to be the integral of $X_j^{(i)}$. Note that, by the independence of the components of the d-dimensional Brownian motion, the components

$\left\{\left\{\boldsymbol{X}_j(t);\ 0 \le t \le T\right\};\ j \in [d]\right\}$ of $\{\boldsymbol{X}(t);\ 0 \le t \le T\}$ are independent[2]. The (strong) solution of eq. (C.2) is a Gauss–Markov process with mean $m_j\colon [0,T] \to \mathbb{R}^{q+1}$ and covariance matrix $P_j\colon [0,T] \to \mathbb{R}^{(q+1)\times(q+1)}$ given by

$$m_j(t) = A(t)m_j(0), \tag{C.3}$$
$$P_j(t) = A(t)P_j(0)A(t)^\intercal + Q(t), \tag{C.4}$$

where the matrices $A(t),\ Q(t) \in \mathbb{R}^{(q+1)\times(q+1)}$ yielded by the SDE eq. (C.2) are known in closed form Särkkä (2006, Theorem 2.9) (see eq. (C.77)). The precise choice of the prior stochastic process $\boldsymbol{X}$ depends on the choice of $(c_0, \ldots, c_q) \in \mathbb{R}^{q+1}$ in eq. (C.2). While the below algorithm works for all choices of c, we restrict our attention to the case of

$$(c_0, \ldots, c_q) \coloneqq (0, \ldots, 0, -\theta), \qquad \text{for some} \quad \theta \ge 0, \tag{C.5}$$

where the q-times integrated Brownian motion (IBM) and the q-times integrated Ornstein–Uhlenbeck process (IOUP) with drift parameter θ is the unique solution of eq. (C.2), in the case of $\theta = 0$ and $\theta > 0$ respectively (Karatzas and Shreve, 1991, Chapter 5: Example 6.8). In this case, the matrices A and Q from eqs. (C.3) and (C.4) are given by

$$A(t)_{ij} = \begin{cases} \mathbb{I}_{i\le j}\frac{t^{j-i}}{(j-i)!}, & \text{if } j \ne q, \\ \frac{t^{q-i}}{(q-i)!} - \theta\sum_{k=q+1-i}^{\infty}\frac{(-\theta)^{k+i-q-1}t^k}{k!}, & \text{if } j = q, \end{cases} \tag{C.6}$$

$$Q(t)_{ij} = \sigma^2\frac{t^{2q+1-i-j}}{(2q+1-i-j)(q-i)!(q-j)!} + \Theta\left(t^{2q+2-i-j}\right). \tag{C.7}$$

(Derivations of eqs. (C.6) and (C.7), as well as the precise form of Q without $\Theta(t^{2q+2-i-j})$, are presented in Appendix C.11.) Hence, for all $i \in [q+1]$, the prediction of step size h of the i^{th} derivative from any state $u \in \mathbb{R}^{q+1}$ is given by

$$\left[A(t)u\right]_i = \sum_{k=i}^{q}\frac{t^{k-i}}{(k-i)!}u_k - \theta\left[\sum_{k=q+1-i}^{\infty}\frac{(-\theta)^{k+i-q-1}}{k!}t^k\right]u_q. \tag{C.8}$$

[2]More involved correlation models of $\left\{\left\{\boldsymbol{X}_j(t);\ 0 \le t \le T\right\};\ j \in [d]\right\}$ are straightforward to incorporate into the SDE eq. (C.2), but seem complicated to analyse. Therefore, we restrict our attention to independent dimensions. See Appendix C.12 for an explanation of this restriction. Note that one can also use a state space vector $\boldsymbol{X}(t)$ which models other features of $x(t)$ than the derivatives, as demonstrated with Fourier summands in Kersting and Mahsereci (2020).

C.2.2 The algorithm

To avoid the introduction of additional indices, we will define the algorithm $\mathbf{\Psi}$ for $d = 1$; for statements on the general case of $d \in \mathbb{N}$ we will use the same symbols from eq. (C.10)–eq. (C.15) as vectors over the whole dimension—see e.g. eq. (C.31) for a statement about a general $r \in \mathbb{R}^d$. By the independence of the dimensions of $\boldsymbol{X}$, due to eq. (C.2), extension to $d \in \mathbb{N}$ amounts to applying $\mathbf{\Psi}$ to every dimension independently (recall Footnote 2). Accordingly, we may in many of the below proofs w.l.o.g. assume $d = 1$. Now, as previously spelled out in Kersting and Hennig (2016); Schober *et al.* (2019), Bayesian filtering of $\boldsymbol{X}$—i.e. iteratively conditioning $\boldsymbol{X}$ on the information on $X^{(1)}$ from evaluations of f at the mean of the current conditioned $X^{(0)}$—yields the following numerical method $\mathbf{\Psi}$. Let $\boldsymbol{m}(t) = (m^{(0)}(t), \dots, m^{(q)}(t))^\intercal \in \mathbb{R}^{q+1}$ be an arbitrary state at some point in time $t \in [0, T]$ (i.e. $m^{(i)}(t)$ is an estimate for $x^{(i)}(t)$), and let $P(t) \in \mathbb{R}^{(q+1)\times(q+1)}$ be the covariance matrix of $x^{(i)}(t)$. For $t \in [0, T]$, let the current estimate of $\boldsymbol{x}(t)$ be a normal distribution $\mathcal{N}(\boldsymbol{m}(t), P(t))$, i.e. the mean $\boldsymbol{m}(t) \in \mathbb{R}^{q+1}$ represents the best numerical estimate (given data $\{y(h), \dots, y(t)\}$, see eq. (C.12)) and the covariance matrix $P(t) \in \mathbb{R}^{(q+1)\times(q+1)}$ its uncertainty. For the time step $t \to t + h$ of size $h > 0$, the ODE filter first computes the prediction step consisting of *predictive mean*

$$\boldsymbol{m}^-(t+h) \coloneqq A(h)\boldsymbol{m}(t) \ \in \mathbb{R}^{q+1}, \tag{C.9}$$

and *predictive covariance*

$$P^-(t+h) \coloneqq A(h)P(t)A(h)^\intercal + Q(h) \ \in \mathbb{R}^{(q+1)\times(q+1)}, \tag{C.10}$$

with A and Q generally defined by eq. (C.77) and, in the considered particular case of eq. (C.5), by eqs. (C.6) and (C.7). In the subsequent step, the following quantities are computed first: the *Kalman gain*

$$\begin{aligned}\boldsymbol{\beta}(t+h) &= (\beta^{(0)}(t+h), \dots, \beta^{(q)}(t+h))^\intercal \\ &\coloneqq \frac{P^-(t+h)_{:1}}{(P^-(t+h))_{11} + R(t+h)} \in \mathbb{R}^{(q+1)\times 1},\end{aligned} \tag{C.11}$$

the *measurement/data on* $\dot{x}$

$$y(t+h) \coloneqq f\left(m^{-,(0)}(t+h)\right) \ \in \mathbb{R}, \tag{C.12}$$

and *innovation/residual*

$$r(t+h) \coloneqq y(t+h) - m^{-,(1)}(t+h) \ \in \mathbb{R}. \tag{C.13}$$

Here, R denotes the variance of y (the 'measurement noise') and captures the squared difference between the data $y(t+h) = f(m^-(t+h))$ that the algorithm actually receives and the idealised data $\dot{x}(t+h) = f(x(t+h))$ that it 'should' receive (see Appendix C.2.3). Finally, the mean and the covariance matrix are conditioned on this data, which yields the *updated mean*

$$\begin{aligned} \boldsymbol{\Psi}_{P(t),h}(\boldsymbol{m}(t)) &\coloneqq \boldsymbol{m}(t+h) \\ &= \boldsymbol{m}^-(t+h) + \boldsymbol{\beta}(t+h) r(t+h), \end{aligned} \tag{C.14}$$

and the *updated covariance*

$$P(t+h) \coloneqq P^-(t+h) - \frac{P^-(t+h)_{:,1} P^-(t+h)_{1,:}}{P^-(t+h)_{11} + R(t+h)}. \tag{C.15}$$

This concludes the step $t \to t+h$, with the Gaussian distribution $\mathcal{N}(\boldsymbol{m}(t+h), P(t+h))$ over $\boldsymbol{x}(t+h)$. The algorithm is iterated by computing $\boldsymbol{m}(t+2h) \coloneqq \boldsymbol{\Psi}_{P(t+h),h}(\boldsymbol{m}(t+h))$ as well as repeating eq. (C.10) and eq. (C.15), with $P(t+h)$ instead of $P(t)$, to obtain $P(t+2h)$. In the following, to avoid notational clutter, the dependence of the above quantities on t, h and σ will be omitted if their values are unambiguous. Parameter adaptation reminiscent of classical methods (e.g. for σ s.t. the added variance per step coincide with standard error estimates) have been explored in Schober *et al.* (2019, Section 4).

This filter is essentially an iterative application of Bayes rule (see e.g. Särkkä (2013, Chapter 4)) based on the prior $\boldsymbol{X}$ on $\boldsymbol{x}$ specified by eq. (C.2) (entering the algorithm via A and Q) and the measurement model $y \sim \mathcal{N}(\dot{x}, R)$. Since the measurement model is a likelihood by another name and therefore forms a complete Bayesian model together with the prior $\boldsymbol{X}$, it remains to detail the measurement model (recall appendix C.2.1 for the choice of prior). Concerning the data generation mechanism for y eq. (C.12), we only consider the maximum-a-posteriori point estimate of $\dot{x}(t)$ given $\mathcal{N}(m^{-,(0)}(t), P^-_{00}(t))$; a discussion of more involved statistical models for y as well as an algorithm box for the Gaussian ODE filter can be found in Schober *et al.* (2019, Subsection 2.2). Next, for lack of such a discussion for R, we will examine different choices of R—which have proved central to the UQ of the filter (Kersting and Hennig, 2016) and will turn out to affect global convergence properties in Appendix C.7.

C.2.3 Measurement noise R

Two sources of uncertainty add to $R(t)$: noise from imprecise knowledge of $x(t)$ and f. Given f, previous integration steps of the filter (as well as an imprecise initial value) inject uncertainty about how close $m^-(t)$ is to $x(t)$ and how close $y = f(m^-(t))$ is to $\dot{x}(t)) = f(x(t))$. This uncertainty stems from the discretization error $\|m^{-,(0)}(t) - x(t)\|$ and, hence, tends to increase with h. Additionally, there can be uncertainty from a

misspecified f, e.g. when f has estimated parameters, or from numerically imprecise evaluations of f, which can be added to R—a functionality which classical solvers do not possess. In this paper, since R depends on h via the numerical uncertainty on $x(t)$, we analyse the influence of noise R of order $p \in [1, \infty]$ (see Assumption C.4) on the quality of the solution to illuminate for which orders of noise we can trust the solution to which extent and when we should, instead of decreasing h, rather spend computational budget on specifying or evaluating f more precisely. The explicit dependence of the noise on its order p in h resembles, despite the fundamentally different role of R compared to additive noise in Conrad *et al.* (2017); Abdulle and Garegnani (2020), the variable p in Conrad *et al.* (2017, Assumption 1) and Abdulle and Garegnani (2020, Assumption 2.2) in the sense that the analysis highlights how uncertainty of this order can still be modeled without breaking the convergence rates. (Adaptive noise models are computationally feasible (Kersting and Hennig, 2016) but lie outside the scope of our analysis.)

C.3 Regularity of flow

Before we proceed to the analysis of $\mathbf{\Psi}$, we provide all regularity results necessary for arbitrary $q, d \in \mathbb{N}$ in this section.

Assumption C.1. *The vector field $f \in C^q(\mathbb{R}^d; \mathbb{R}^d)$ is globally Lipschitz and all its derivatives of order up to q are uniformly bounded and globally Lipschitz, i.e. there exists some $L > 0$ such that $\|D^\alpha f\|_\infty \leq L$ for all multi-indices $\alpha \in \mathbb{N}_0^d$ with $1 \leq \sum_i \alpha_i \leq q$, and $\|D^\alpha f(a) - D^\alpha f(b)\| \leq L\|a - b\|$ for all multi-indices $\alpha \in \mathbb{N}_0^d$ with $0 \leq \sum_i \alpha_i \leq q$.*

Assumption C.1 and the Picard–Lindelöf theorem imply that the solution x is a well-defined element of $C^{q+1}([0, T]; \mathbb{R}^d)$. For $i \in [q+1]$, we denote $\frac{\mathrm{d}^i x}{\mathrm{d}t^i}$ by $x^{(i)}$. Recall that, by a bold symbol, we denote the vector of these derivatives: $\boldsymbol{x} \equiv (x^{(0)}, \ldots, x^{(q)})^\intercal$. In particular, the solution x of eq. (C.1) is denoted by $x^{(0)}$. Analogously, we denote the flow of the ODE eq. (C.1) by $\Phi^{(0)}$, i.e. $\Phi_t^{(0)}(x_0) \equiv x^{(0)}(t)$, and, for all $i \in [q+1]$, its i^{th} partial derivative with respect to t by $\Phi^{(i)}$, so that $\Phi_t^{(i)}(x_0) \equiv x^{(i)}(t)$.

Lemma C.3.1. *Under Assumption C.1, for all $a \in \mathbb{R}^d$ and all $h > 0$,*

$$\left\| \Phi_h^{(i)}(a) - \sum_{k=i}^{q} \frac{h^{k-i}}{(k-i)!} \Phi_0^{(k)}(a) \right\| \leq K h^{q+1-i}. \tag{C.16}$$

Here, and in the sequel, $K > 0$ denotes a constant independent of h and θ which may change from line to line.

Proof. By Assumption C.1, $\Phi^{(q+1)}$ exists and is bounded by $\|\Phi^{(q+1)}\| \leq L$, which can be seen by applying the chain rule q times to both sides of eq. (C.1). Now, applying $\|\Phi^{(q+1)}\| \leq L$ to the term $\Phi_\tau^{(q+1)}(a)$ (for some $\tau \in (0, h)$) in the Lagrange remainder of the $(q-i)^{\text{th}}$-order Taylor expansion of $\Phi_h^{(i)}(a)$ yields eq. (C.16). □

Lemma C.3.2. *Under Assumption C.1 and for all sufficiently small $h > 0$,*

$$\sup_{a \neq b \in \mathbb{R}^d} \frac{\left\| \Phi_h^{(0)}(a) - \Phi_h^{(0)}(b) \right\|}{\|a - b\|} \leq 1 + 2Lh. \tag{C.17}$$

Proof. Immediate corollary of Teschl (2012, Theorem 2.8). □

Global convergence (Appendix C.7) will require the following generalization of Lemma C.3.2.

Lemma C.3.3. *Let $q = 1$. Then, under Assumption C.1 and for all sufficiently small $h > 0$,*

$$\sup_{a \neq b \in \mathbb{R}^d} \frac{\left\| \left\| \mathbf{\Phi}_h(a) - \mathbf{\Phi}_h(b) \right\| \right\|_h}{\|a - b\|} \leq 1 + Kh, \tag{C.18}$$

where, given the norm $\|\cdot\|$ on $\mathbb{R}^d$ and $h > 0$, the new norm $\left\| \left\| \cdot \right\| \right\|_h$ on $\mathbb{R}^{(q+1)\times d}$ is defined by

$$\left\| \left\| a \right\| \right\|_h \coloneqq \sum_{i=0}^{q} h^i \left\| a_{i,:} \right\|. \tag{C.19}$$

Remark C.3.4. *The necessity of $\left\| \left\| \cdot \right\| \right\|_h$ stems from the fact that—unlike other ODE solvers—the ODE filter $\mathbf{\Psi}$ additionally estimates and uses the first q derivatives in its state $\boldsymbol{m} \in \mathbb{R}^{(q+1)\times d}$, whose development cannot be bounded in $\|\cdot\|$, but in $\left\| \left\| \cdot \right\| \right\|_h$. The norm $\left\| \left\| \cdot \right\| \right\|_h$ is used to make rigorous the intuition that the estimates of the solution's time derivative are 'one order of h worse per derivative'.*

Proof. We bound the second summand of

$$\left\| \left\| \mathbf{\Phi}_h(a) - \mathbf{\Phi}_h(b) \right\| \right\|_h \overset{\text{eq. (C.19)}}{=} \underbrace{\left\| \Phi_h^{(0)}(a) - \Phi_h^{(0)}(b) \right\|}_{\leq (1+2Lh)\|a-b\|, \text{ by eq. (C.17)}} + h \Big\| \underbrace{\Phi_h^{(1)}(a)}_{= f\left(\Phi_h^{(0)}(a)\right)} - \underbrace{\Phi_h^{(1)}(b)}_{= f\left(\Phi_h^{(0)}(b)\right)} \Big\| \tag{C.20}$$

by

$$\begin{aligned} \left\| f\left(\Phi_h^{(0)}(a)\right) - f\left(\Phi_h^{(0)}(b)\right) \right\| &\overset{\text{Ass. } C.1}{\leq} \\ L \left\| \Phi_h^{(0)}(a) - \Phi_h^{(0)}(b) \right\| &\overset{\text{eq. (C.17)}}{\leq} L(1 + 2Lh) \|a - b\|. \end{aligned} \tag{C.21}$$

Inserting eq. (C.21) into eq. (C.20) concludes the proof. □

C.4 The role of the state misalignments δ

In Gaussian ODE filtering, the interconnection between the estimates of the ODE solution $x(t) = x^{(0)}(t)$ and its first q derivatives $\{x^{(1)}(t), \dots, x^{(q)}(t)\}$ is intricate. From a purely analytical point of view, every possible estimate $m(t)$ of $x(t)$ comes with a fixed set of derivatives, which are implied by the ODE, for the following reason: Clearly, by eq. (C.1), the estimate $m^{(1)}(t)$ of $x^{(1)}(t)$ ought to be $f(m(t))$. More generally (for $i \in [q+1]$) the estimate $m^{(i)}(t)$ of $x^{(i)}(t)$ is determined by the ODE as well. To see this, let us first recursively define $f^{(i)} \colon \mathbb{R}^d \to \mathbb{R}^d$ by $f^{(0)}(a) \coloneqq a$, $f^{(1)}(a) \coloneqq f(a)$ and $f^{(i)}(a) \coloneqq [\nabla_x f^{(i-1)} \cdot f](a)$. Now, differentiating the ODE, eq. (C.1), $(i-1)$-times by the chain rule yields

$$x^{(i)}(t) = f^{(i-1)}(t)\left(x^{(0)}(t)\right), \tag{C.22}$$

which implies that $m^{(i)}(t)$ ought to be $f^{(i-1)}(t)\left(m^{(0)}(t)\right)$ Since

$$\Phi_0^{(i)}\left(m^{(0)}(nh)\right) = f^{(i-1)}\left(m^{(0)}(nh)\right) \tag{C.23}$$

(which we prove in Appendix C.15), this amounts to requiring that

$$m^{(i)}(t) \stackrel{!}{=} \Phi_0^{(i)}\left(m^{(0)}(nh)\right). \tag{C.24}$$

Since $\Phi_0^{(i)}$ is (recall Appendix C.3) the i^{th} time derivative of the flow map $\Phi^{(0)}$ at $t = 0$, this simply means that $m^{(i)}(t)$ would be set to the 'true' derivatives in the case where the initial condition of the ODE, eq. (C.1), is $x(0) = m^{(0)}(t)$ instead of $x(0) = x_0$—or, more loosely speaking, that the derivative estimates $m^{(i)}(t)$ are forced to comply with $m^{(0)}(t)$, irrespective of our belief $x^{(i)}(t) \sim \mathcal{N}(m^{(i)}(t), P_{ii}(t))$. The Gaussian ODE filter, however, does not use this (intractable) analytical approach. Instead, it jointly models and infers $x^{(0)}(t)$ and its first q derivatives $\{x^{(1)}(t), \dots, x^{(q)}(t)\}$ in a state space $\boldsymbol{X}$, as detailed in Appendix C.2. The thus-computed filtering mean estimates $m^{(i)}(t)$ depend not only on the ODE but also on the statistical model—namely on the prior (SDE) and measurement noise R; recall Appendices C.2.1 and C.2.3. In fact, the analytically-desirable derivative estimate, eq. (C.24), is, for $i = 1$, only satisfied if $R = 0$ (which can be seen from eq. (C.14)), and generally does not hold for $i \geq 2$ since both $f^{(i-1)}$ and $\Phi^{(i)}$ are inaccessible to the algorithm. The numerical example in Appendix C.13 clarifies that $\delta^{(i)}$ is likely to be strictly positive, even after the first step $0 \to h$.

This inevitable mismatch, between exact analysis and approximate statistics, motivates the following definition of the i^{th} state *i^{th} state misalignment at time t*:

$$\delta^{(i)}(t) \coloneqq \left\| m^{(i)}(t) - \Phi_0^{(i)}\left(m^{(0)}(t)\right) \right\| \geq 0. \tag{C.25}$$

Intuitively speaking, $\delta^{(i)}(t)$ quantifies how large this mismatch is for the i^{th} derivative

at time t. Note that $\delta^{(i)}(t) = 0$ if and only if eq. (C.24) holds—i.e. for $i = 1$ iff $R = 0$ (which can be seen from eq. (C.14)) and only by coincidence for $i \geq 2$ since both $f^{(i-1)}$ and $\Phi_0^{(i)}$ are inaccessible to the algorithm. (Since $\Phi_0^{(0)} = \mathrm{Id}$, $\delta^{(0)}(t) = 0$ for all t.)

The possibility of $\delta^{(i)} > 0$, for $i \geq 1$, is inconvenient for the below worst-case analysis since (if eq. (C.24) held true and $\delta^{(i)} \equiv 0$) the prediction step of the drift-less IBM prediction ($\theta = 0$) would coincide with a Taylor expansions of the flow map $\Phi_0^{(i)}$; see eq. (C.8). But, because $\delta^{(i)} \neq 0$ in general, we have to additionally bound the influence of $\delta \geq 0$ which complicates the below proofs further.

Fortunately, we can *locally* bound the import of $\delta^{(i)}$ by the easy Lemma C.6.1 and *globally* by the more complicated Lemma C.7.4 (see Appendix C.7.3). Intuitively, these bounds demonstrate that the order of the deviation from a Taylor expansion of the state $\boldsymbol{m} = [m^{(0)}, \ldots, m^{(q)}]$ due to δ is not smaller than the remainder of the Taylor expansion. This means, more loosely speaking, that the import of the $\delta^{(i)}$ is swallowed by the Taylor remainder. This effect is locally captured by Lemma C.5.1 and globally by Lemma C.7.5. The global convergence rates of $\delta^{(i)}(T)$, as provided by Lemma C.7.5, are experimentally demonstrated in Appendix C.14.

C.5 Auxiliary bounds on intermediate quantities

Recall from eq. (C.5) that $\theta = 0$ and $\theta > 0$ denote the cases of IBM and IOUP prior with drift coefficient θ respectively. The ODE filter Ψ iteratively computes the filtering mean $\boldsymbol{m}(nh) = (m^{(0)}(nh), \ldots, m^{(q)}(nh))^\intercal \in \mathbb{R}^{(q+1)}$ as well as error covariance matrices $P(nh) \in \mathbb{R}$ on the mesh $\{nh\}_{n=0}^{T/h}$. (Here and in the following, we assume w.l.o.g. that $T/h \in \mathbb{N}$.) Ideally, the truncation error over all derivatives

$$\boldsymbol{\varepsilon}(nh) \coloneqq (\varepsilon^{(0)}(nh), \ldots, \varepsilon^{(q)}(nh))^\intercal \coloneqq \boldsymbol{m}(nh) - \boldsymbol{x}(nh), \tag{C.26}$$

falls quickly as $h \to 0$ and is estimated by the standard deviation $\sqrt{P_{00}(nh)}$. Next, we present a classical worst-case convergence analysis over all f satisfying Assumption C.1; see Appendix C.10 for a discussion of the desirability and feasibility of an average-case analysis. To this end, we bound the added error of every step by intermediate values, defined in eqs. (C.11) and (C.13),

$$\begin{aligned}
\Delta^{(i)}((n+1)h) &\coloneqq \left\| \Psi^{(i)}_{P(nh),h}(\boldsymbol{m}(nh)) - \Phi_h^{(i)}\left(m^{(0)}(nh)\right) \right\| && \text{(C.27)} \\
&\overset{\text{eq. (C.14)}}{\leq} \underbrace{\left\| (A(h)\boldsymbol{m}(nh))_i - \Phi_h^{(i)}\left(m^{(0)}(nh)\right) \right\|}_{=:\Delta^{-(i)}((n+1)h)} \\
&\quad + \left\| \beta^{(i)}((n+1)h) \right\| \left\| r((n+1)h) \right\|, && \text{(C.28)}
\end{aligned}$$

and bound these quantities in the order $\Delta^{-(i)}$, r, $\beta^{(i)}$. These bounds will be needed for the local and global convergence analysis in Appendices C.6 and C.7 respectively. Note that, intuitively, $\Delta^{-(i)}((n+1)h)$ and $\Delta^{(i)}((n+1)h)$ denote the *additional* numerical error which is added in the $(n+1)^{\text{th}}$ step to the i^{th} derivative of the predictive mean $m^{-,(i)}(t+h)$ and the updated mean $m^{(i)}(t+h)$, respectively.

Lemma C.5.1. *Under Assumption C.1, for all $i \in [q+1]$ and all $h > 0$,*

$$\begin{aligned}\Delta^{-(i)}((n+1)h) \leq & K\left[1+\theta\left\|m^{(q)}(nh)\right\|\right] h^{q+1-i} \\ & + \sum_{k=i}^{q} \frac{h^{k-i}}{(k-i)!}\delta^{(k)}(nh).\end{aligned} \tag{C.29}$$

Proof. We may assume, as explained in Appendix C.2.2, without loss of generality that $d = 1$. We apply the triangle inequality to the definition of $\Delta^{-(i)}((n+1)h)$, as defined in eq. (C.28), which, by eq. (C.8), yields

$$\begin{aligned}&\Delta^{-(i)}((n+1)h) \quad \leq \\ &\sum_{k=i}^{q} \frac{h^{k-i}}{(k-i)!}\delta^{(k)}(nh) + K\theta\left|m^{(q)}(nh)\right| h^{q+1-i} \\ &+ \underbrace{\left|\sum_{l=i}^{q} \frac{h^{l-i}}{(l-i)!}\Phi_0^{(l)}\left(m^{(0)}(nh)\right) - \Phi_h^{(i)}\left(m^{(0)}(nh)\right)\right|}_{\leq K h^{q+1-i}, \text{ by eq. (C.16)}}.\end{aligned} \tag{C.30}$$

□

Lemma C.5.2. *Under Assumption C.1 and for all sufficiently small $h > 0$,*

$$\begin{aligned}\|r((n+1)h)\| \leq & K\left[1+\theta\left\|m^{(q)}(nh)\right\|\right] h^{q} \\ & + K\sum_{k=1}^{q} \frac{h^{k-1}}{(k-1)!}\delta^{(k)}(nh).\end{aligned} \tag{C.31}$$

Proof. See Appendix C.16. □

To bound the Kalman gains $\boldsymbol{\beta}(nh)$, we first need to assume that the orders of the initial covariance matrices are sufficiently high (matching the latter required orders of the initialization error; see Assumption C.3).

Assumption C.2. *The entries of the initial covariance matrix $P(0)$ satisfy, for all $k, l \in [q+1]$, $\|P(0)_{k,l}\| \leq K_0 h^{2q+1-k-l}$, where $K_0 > 0$ is a constant independent of h.*

We make this assumption, as well as Assumption C.3, explicit (instead of just making the stronger assumption of exact initializations with zero variance), because it highlights how statistical or numerical uncertainty on the initial value effects the accuracy

of the output of the filter—a novel functionality of PN with the potential to facilitate a management of the computational budget across a computational chain with respect to the respective perturbations from different sources of uncertainty (Hennig *et al.*, 2015, Section 3(d)).

Lemma C.5.3. *Under Assumption C.2, for all $i \in [q+1]$ and for all $h > 0$, $\|\beta^{(i)}(h)\| \leq Kh^{1-i}$.*

Proof. Again, w.l.o.g. $d = 1$. Application of the orders of A and Q from eqs. (C.6) and (C.7), the triangle inequality and Assumption C.2 to the definition of P^- in eq. (C.10) yields

$$\begin{aligned}
\left|P^-(h)_{k,l}\right| &\overset{\text{eq. (C.10)}}{\leq} \left|[A(h)P(0)A(h)^\intercal]_{k,l}\right| + \left|Q(h)_{k,l}\right| \\
&\overset{\text{eqs. (C.6),(C.7)}}{\leq} K\Bigg[\sum_{a=k}^{q}\sum_{b=l}^{q}\left|P(0)_{a,b}\right| h^{a+b-k-l} \\
&\qquad + 2\theta\sum_{b=l}^{q-1}\left|P(0)_{q,b}\right| \\
&\qquad + \theta^2\left|P(0)_{q,q}\right| + h^{2q+1-k-l}\Bigg] \\
&\overset{\text{Ass. C.2}}{\leq} K[1+\theta+\theta^2]h^{2q+1-k-l}. \qquad \text{(C.32)}
\end{aligned}$$

Recall that P and Q are (positive semi-definite) covariance matrices; hence, $P^-(h)_{1,1} \geq Kh^{2q-1}$. Inserting these orders into the definition of $\beta^{(i)}$ (eq. (C.11)), recalling that $R \geq 0$, and removing the dependence on θ by reducing the fraction conclude the proof. □

C.6 Local convergence rates

With the above bounds on intermediate algorithmic quantities (involving state misalignments $\delta^{(i)}$) in place, we only need an additional assumption to proceed—via a bound on $\delta^{(i)}(0)$—to our first main result on local convergence orders of $\mathbf{\Psi}$.

Assumption C.3. *The initial errors on the initial estimate of the i^{th} derivative $m^{(i)}(0)$ satisfy $\|\varepsilon^{(i)}(0)\| = \|m^{(i)}(0) - x^{(i)}(0)\| \leq K_0 h^{q+1-i}$. (This assumption is, like Assumption C.2, weaker than the standard assumption of exact initializations.)*

Lemma C.6.1. *Under Assumptions C.1 and C.3, for all $i \in [q+1]$ and for all $h > 0$, $\delta^{(i)}(0) \leq Kh^{q+1-i}$.*

Proof. The claim follows, using Assumptions C.1 and C.3, from

$$\delta^{(i)}(0) \leq \underbrace{\left\| m^{(i)}(0) - x^{(i)}(0) \right\|}_{=\|\varepsilon^{(i)}(0)\| \leq K_0 h^{q+1-i}} + \underbrace{\left\| f^{(i-1)}\left(x^{(0)}(0)\right) - f^{(i-1)}\left(m^{(0)}(0)\right) \right\|}_{\leq L\|\varepsilon^{(0)}(0)\| \leq L K_0 h^{q+1}}. \tag{C.33}$$

□

Now, we can bound the local truncation error $\varepsilon^{(0)}(h)$ as defined in eq. (C.26).

Theorem C.6.2 (Local Truncation Error)**.** *Under Assumptions C.1, C.2 and C.3 and for all sufficiently small $h > 0$,*

$$\left\| \varepsilon^{(0)}(h) \right\| \leq |||\boldsymbol{\varepsilon}(h)|||_h \leq K \left[1 + \theta \left\| m^{(q)}(0) \right\|\right] h^{q+1}. \tag{C.34}$$

Proof. By the triangle inequality for $|||\cdot|||_h$ and subsequent application of Lemma C.3.3 and Assumption C.3 to the second summand of the resulting inequality, we obtain

$$\begin{aligned} |||\boldsymbol{\varepsilon}(h)|||_h \leq & \underbrace{\left\| \boldsymbol{\Psi}_{P(0),h}\left(\boldsymbol{m}(0)\right) - \boldsymbol{\Phi}_h\left(x^{(0)}(0)\right) \right\|_h}_{=\sum_{i=0}^{q} h^i \Delta^{(i)}(h), \text{ by eq. (C.27)}} \\ & + \underbrace{\left\| \boldsymbol{\Phi}_h\left(x^{(0)}(0)\right) - \boldsymbol{\Phi}_h\left(m^{(0)}(0)\right) \right\|_h}_{\leq (1+Kh)\|\varepsilon^{(0)}(0)\| \leq K h^{q+1}}. \end{aligned} \tag{C.35}$$

The remaining bound on $\Delta^{(i)}(h)$, for all $i \in [q+1]$ and sufficiently small $h > 0$, is obtained by insertion of the bounds from Lemmas C.5.1 to C.5.3 (in the case of $n = 0$), into eq. (C.28):

$$\begin{aligned} \Delta^{(i)}(h) &\leq K \left[1 + \theta \left\| m^{(q)}(0) \right\|\right] h^{q+1-i} \\ &\quad + K \sum_{k=1}^{q} \frac{h^{k-1}}{(k-1)!} \delta^{(k)}(nh) && \text{(C.36)} \\ &\overset{\text{Lemma } C.6.1}{\leq} K \left[1 + \theta \left\| m^{(q)}(0) \right\|\right] h^{q+1-i}. && \text{(C.37)} \end{aligned}$$

Insertion of eq. (C.37) into eq. (C.35) and $\|\varepsilon^{(0)}(h)\| \leq |||\boldsymbol{\varepsilon}(h)|||_h$ (by eq. (C.19)) concludes the proof. □

Remark C.6.3. *Theorem C.6.2 establishes a bound of order h^{q+1} on the local truncation error $\varepsilon^{(0)}(h)$ on $x(h)$ after one step h. Moreover, by the definition eq.* (C.19) *of $|||\cdot|||_h$, this theorem also implies additional bounds of order h^{q+1-i} on the error $\varepsilon^{(i)}(h)$ on the i^{th} derivative $x^{(i)}(h)$ for all $i \in [q+1]$. Such derivative bounds are (to the best of our knowledge) not available for classical numerical solvers, since they do not explicitly model the derivatives in the first place. These bounds could be be useful for subsequent computations based on the ODE trajectory (Hennig* et al., *2015).*

Unsurprisingly, as the mean prediction (recall eq. (C.8)*) deviates from a pure q^{th} order Taylor expansion by $K\theta\|m^{(q)}(0)\|h^{q+1}$ for an IOUP prior (i.e. $\theta > 0$ in eq.* (C.5)*), the constant in front of the local h^{q+1} convergence rate depends on both θ and $m^{(q)}(0)$ in the IOUP case. A global analysis for IOUP is therefore more complicated than for IBM: Recall from eq.* (C.8) *that, for $q = 1$, the mean prediction for $x((n+1)h)$ is*

$$\begin{pmatrix} m^{-,(0)}((n+1)h) \\ m^{-,(1)}((n+1)h) \end{pmatrix} \overset{\text{eq. (C.8)}}{=} \begin{pmatrix} m^{(0)}(nh) + hm^{(1)}(nh) - \theta\left[\frac{h^2}{2!} + \mathcal{O}(h^3)\right] m^{(1)}(nh) \\ e^{-\theta h} m^{-,(1)}(nh) \end{pmatrix}, \tag{C.38}$$

which pulls both $m^{-,(0)}$ and $m^{-,(1)}$ towards zero (or some other prior mean) compared to the prediction given by its Taylor expansion for $\theta = 0$. While this is useful for ODEs converging to zero, such as $\dot{x} = -x$, it is problematic for diverging ODEs, such as $\dot{x} = x$ (Magnani et al.*, 2017). As shown in Theorem C.6.2, this effect is asymptotically negligible for local convergence, but it might matter globally and, therefore, might necessitate stronger assumptions on f than Assumption C.1, such as a bound on $\|f\|_\infty$ which would globally bound $\{y(nh);\ n = 0, \dots, T/h\}$ and thereby $\{m^{(1)}(nh);\ n = 0, \dots, T/h\}$ in eq.* (C.38)*. It is furthermore conceivable that a global bound for IOUP would depend on the relation between θ and $\|f\|_\infty$ in a nontrivial way. The inclusion of IOUP ($\theta > 0$) would hence complicate the below proofs further. Therefore, we restrict the following first global analysis to IBM ($\theta = 0$).*

C.7 Global analysis

As explained in Remark C.6.3, we only consider the case of the IBM prior, i.e. $\theta = 0$, in this section. Moreover, we restrict our analysis to $q = 1$ in this first global analysis. Although we only have definite knowledge for $q = 1$, we believe that the convergence rates might also hold for higher $q \in \mathbb{N}$—which we experimentally test in Appendix C.9.1. Moreover, we believe that proofs analogous to the below proofs might work out for higher $q \in \mathbb{N}$ and that deriving a generalized version of Proposition C.7.2 for higher q is the bottleneck for such proofs. (See Appendix C.10 for a discussion of these restrictions.)

While, for local convergence, all noise models R yield the same convergence rates in Theorem C.6.2, it is unclear how the order of R in h (as described in Appendix C.2.3) affects global convergence rates: E.g., for the limiting case $R \equiv Kh^0$, the steady-state Kalman gains $\boldsymbol{\beta}^\infty$ would converge to zero (see eqs. (C.43) and (C.44) below) for $h \to 0$, and hence the evaluation of f would not be taken into account—yielding a filter $\boldsymbol{\Psi}$ which assumes that the evaluations of f are equally off, regardless of $h > 0$, and eventually just extrapolates along the prior without global convergence of the posterior mean $\boldsymbol{m}$. For the opposite limiting case $R \equiv \lim_{p\to\infty} Kh^p \equiv 0$, it has already been shown in

Schober *et al.* (2019, Proposition 1 and Theorem 1) that—in the steady state and for $q = 1, 2$—the filter $\mathbf{\Psi}$ inherits global convergence rates from known multistep methods in Nordsieck form Nordsieck (1962). To explore a more general noise model, we assume a fixed noise model $R \equiv Kh^p$ with arbitrary order p.

In the following, we analyse how small p can be in order for $\mathbf{\Psi}$ to exhibit fast global convergence (cf. the similar role of the order p of perturbations in Conrad *et al.* (2017, Assumption 1) and Abdulle and Garegnani (2020, Assumption 2.2)). In light of Theorem C.6.2, the highest possible global convergence rate is $\mathcal{O}(h)$—which will indeed be obtained for all $p \in [1, \infty]$ in Theorem C.7.7. Since every extrapolation step of $\mathbf{\Psi}$ from t to $t+h$ depends not only on the current state, but also on the covariance matrix $P(t)$—which itself depends on all previous steps—Ψ is neither a single-step nor a multistep method. Contrary to Schober *et al.* (2019), we do not restrict our theoretical analysis to the steady-state case, but provide our results under the weaker Assumptions C.2 and C.3 that were already sufficient for local convergence in Theorem C.6.2—which is made possible by the bounds eqs. (C.48) and (C.49) in Proposition C.7.2.

C.7.1 Outline of global convergence proof

The goal of the following sequence of proofs in Appendix C.7 is Theorem C.7.7. It is proved by a special version of the discrete Grönwall inequality (Clark, 1987) whose prerequisite is provided in Lemma C.7.6. This Lemma C.7.6 follows from Lemma C.3.3 (on the regularity of the flow map $\mathbf{\Phi}_t$) as well as Lemma C.7.5 which provides a bound on the maximal increment of the numerical error stemming from local truncation errors. For the proof of Lemma C.7.5, we first have to establish

(i) global bounds on the Kalman gains $\beta^{(0)}$ and $\beta^{(1)}$ by the inequalities eqs. (C.48) and (C.49) in Proposition C.7.2, and

(ii) a global bound on the state misalignment $\delta^{(1)}$ in Lemma C.7.4.

In Appendices C.7.2 to C.7.4, we will collect these inequalities in the order of their numbering to subsequently prove global convergence in Appendix C.7.5.

C.7.2 Global bounds on Kalman gains

Since we will analyse the sequence of covariance matrices and Kalman gains using contractions in Proposition C.7.2, we first introduce the following generalization of Banach fixed-point theorem (BFT).

Lemma C.7.1. *Let $(\mathcal{X}, d)$ be a non-empty complete metric space, $T_n \colon \mathcal{X} \to \mathcal{X}$, $n \in \mathbb{N}$, a sequence of L_n-Lipschitz continuous contractions with $\sup_n L_n \leq \bar{L} < 1$. Let u_n be the fixed point of T_n, as given by BFT, and let $\lim_{n\to\infty} u_n = u^* \in \mathcal{X}$. Then, for all $x_0 \in \mathcal{X}$, the recursive sequence $x_n \coloneqq T_n(x_{n-1})$ converges to u^* as $n \to \infty$.*

Proof. See Appendix C.17. □

In the following, we will assume that T is a multiple of h.

Proposition C.7.2. *For constant $R \equiv Kh^p$ with $p \in [0, \infty]$, the unique (attractive) steady states for the following quantities are*

$$P_{11}^{-,\infty} \coloneqq \lim_{n\to\infty} P_{11}^{-}(nh) \tag{C.39}$$
$$= \frac{1}{2}\left(\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}\right),$$
$$P_{11}^{\infty} \coloneqq \lim_{n\to\infty} P_{11}(nh) \tag{C.40}$$
$$= \frac{\left(\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}\right) R}{\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2} + 2R},$$
$$P_{01}^{-,\infty} \coloneqq \lim_{n\to\infty} P_{01}^{-}(nh) \tag{C.41}$$
$$= \frac{\sigma^4 h^2 + (2R + \sigma^2 h)\sqrt{4\sigma^2 Rh + \sigma^4 h^2} + 4R\sigma^2 h}{2(\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2})} h,$$
$$P_{01}^{\infty} \coloneqq \lim_{n\to\infty} P_{01}(nh) \tag{C.42}$$
$$= \frac{R\sqrt{4R\sigma^2 h + \sigma^4 h^2}}{\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}} h,$$
$$\beta^{\infty,(0)} \coloneqq \lim_{n\to\infty} \beta^{(0)}(nh) \tag{C.43}$$
$$= \frac{\sqrt{4R\sigma^2 h + \sigma^4 h^2}}{\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}} h, \qquad \textit{and}$$
$$\beta^{\infty,(1)} \coloneqq \lim_{n\to\infty} \beta^{(1)}(nh) \tag{C.44}$$
$$= \frac{\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}}{\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2} + 2R}.$$

If furthermore Assumption C.2 holds, then, for all sufficiently small $h > 0$,

$$\max_{n\in[T/h+1]} P_{11}^{-}(nh) \leq Kh^{1\wedge\frac{p+1}{2}}, \tag{C.45}$$

$$\max_{n\in[T/h+1]} P_{11}(nh) \leq Kh^{p\vee\frac{p+1}{2}}, \tag{C.46}$$

$$\max_{n\in[T/h+1]} \|P_{01}(nh)\| \leq Kh^{p+1}, \tag{C.47}$$

$$\max_{n\in[T/h+1]} \left\|\beta^{(0)}(nh)\right\| \leq Kh, \qquad \textit{and} \tag{C.48}$$

$$\max_{n\in[T/h+1]} \left\|1-\beta^{(1)}(nh)\right\| \leq Kh^{(p-1)\vee 0}. \tag{C.49}$$

All of these bounds are sharp in the sense that they fail for any higher order in the exponent of h.

Remark C.7.3. *The recursions for $P(nh)$ and $P^{-}(nh)$ given by eqs.* (C.10) *and* (C.15) *follow a discrete algebraic Riccati equation (DARE)—a topic studied in many related settings (Lancaster and Rodman, 1995). While the asymptotic behavior eq.* (C.39) *of the completely detectable state $X^{(1)}$ can also be obtained using classical filtering theory (Anderson and Moore, 1979, Chapter 4.4), the remaining statements of Proposition C.7.2 also concern the undetectable state $X^{(0)}$ and are, to the best of our knowledge, not directly obtainable from existing theory on DAREs or filtering (which makes the following proof necessary). Note that, in the special case of no measurement noise ($R \equiv 0$), eqs.* (C.43) *and* (C.44) *yield the equivalence of the filter in the steady state with the P(EC)1 implementation of the trapezoidal rule, which was previously shown in Schober* et al. *(2019, Proposition 1). For future research, it would be interesting to examine whether insertion of positive choices of R into eqs.* (C.43) *and* (C.44) *can reproduce known methods as well.*

Proof. See Appendix C.18. □

C.7.3 Global bounds on state misalignments

For the following estimates, we restrict the choice of p to be larger than $q = 1$.

Assumption C.4. *The noise model is chosen to be $R \equiv Kh^p$, for $p \in [q, \infty] = [1, \infty]$, where $Kh^\infty := 0$.*

Before bounding the added deviation of $\mathbf{\Psi}$ from the flow $\mathbf{\Phi}$ per step, a global bound on the state misalignments defined in eq. (C.25) is necessary. The result of the following lemma is discussed in Appendix C.14.

Lemma C.7.4. *Under Assumptions C.1, C.2, C.3 and C.4, and for all sufficiently small $h > 0$,*

$$\max_{n\in[T/h+1]} \delta^{(1)}(nh) \leq Kh. \tag{C.50}$$

Proof. See Appendix C.19. □

See Lemma C.7.4 for a experimental demonstration of eq. (C.33).

C.7.4 Prerequisite for discrete Grönwall inequality

Equipped with the above bounds, we can now prove a bound on the maximal increment of the numerical error stemming from local truncation errors which is needed to prove eq. (C.56), the prerequisite for the discrete Grönwall inequality.

Lemma C.7.5. *Under Assumptions C.1, C.2, C.3, and C.4 and for all sufficiently small $h > 0$,*

$$\begin{aligned}\max_{n\in[T/h+1]} &\left\| \left\| \boldsymbol{\Psi}_{P(nh),h}\left(\boldsymbol{m}(nh)\right) - \boldsymbol{\Phi}_h\left(m^{(0)}(nh)\right) \right\| \right\|_h \\ &\leq Kh^2.\end{aligned} \tag{C.51}$$

Proof. By eq. (C.19), we have

$$\begin{aligned}&\left\| \left\| \boldsymbol{\Psi}_{P(nh),h}\left(\boldsymbol{m}(nh)\right) - \boldsymbol{\Phi}_h\left(m^{(0)}(nh)\right) \right\| \right\|_h \\ &\quad = S_1(h) + hS_2(h),\end{aligned} \tag{C.52}$$

with $S_1(h)$ and $S_2(h)$ defined and bounded by

$$\begin{aligned}S_1(h) &\coloneqq \left\| \Psi_h^{(0)}\left(\boldsymbol{m}(nh)\right) - \Phi_h^{(0)}\left(m^{(0)}(nh)\right) \right\| \\ &\overset{\text{eq. (C.28)}}{\leq} \underbrace{\Delta^{-(0)}((n+1)h)}_{\overset{\text{eq. (C.29)}}{\leq} Kh^2+\delta^{(0)}(nh)+h\delta^{(1)}(nh)} \\ &\quad + \underbrace{\left\| \beta^{(0)}((n+1)h) \right\|}_{\overset{\text{eq. (C.48)}}{\leq} Kh} \underbrace{\left\| r((n+1)h) \right\|}_{\overset{\text{eq. (C.31)}}{\leq} Kh+(1+Kh)\delta^{(1)}(nh)},\end{aligned} \tag{C.53}$$

and, analogously,

$$\begin{aligned} S_2(h) &:= \left\| \Psi_h^{(1)}\left(\boldsymbol{m}(nh)\right) - \Phi_h^{(1)}\left(m^{(0)}(nh)\right) \right\| \\ &\overset{\text{eq. (C.28)}}{\leq} \underbrace{\Delta^{-(1)}((n+1)h)}_{\overset{\text{eq. (C.29)}}{\leq} Kh+\delta^{(1)}(nh)} \\ &\quad + \underbrace{\left\| \beta^{(1)}((n+1)h) \right\|}_{\overset{\text{eq. (C.11)}}{\leq} 1} \underbrace{\left\| r((n+1)h) \right\|}_{\overset{\text{eq. (C.31)}}{\leq} Kh+(1+Kh)\delta^{(1)}(nh)} \end{aligned} \tag{C.54}$$

Insertion of eq. (C.53) and eq. (C.54) into eq. (C.52) yields

$$\begin{aligned} &\left\| \left\| \boldsymbol{\Psi}_{P(nh),h}\left(\boldsymbol{m}(nh)\right) - \boldsymbol{\Phi}_h\left(m^{(0)}(nh)\right) \right\| \right\|_h \\ &\qquad \leq Kh^2 + \delta^{(0)}(nh) + Kh\delta^{(1)}(nh), \end{aligned} \tag{C.55}$$

which—after recalling $\delta^{(0)}(nh) = 0$ and applying Lemma C.7.4 to $\delta^{(1)}(nh)$—implies eq. (C.51). □

The previous lemma now implies a suitable prerequisite for a discrete Grönwall inequality.

Lemma C.7.6. *Under Assumptions C.1, C.2, C.3, and C.4 and for all sufficiently small $h > 0$,*

$$\left\| \left\| \boldsymbol{\varepsilon}\left((n+1)h\right) \right\| \right\|_h \leq Kh^2 + (1+Kh)\left\| \varepsilon^{(0)}(nh) \right\|. \tag{C.56}$$

Proof. We observe, by the triangle inequality for the norm $\|\|\cdot\|\|_h$, that

$$\begin{aligned} &\left\| \left\| \boldsymbol{\varepsilon}\left((n+1)h\right) \right\| \right\|_h \\ &\quad = \left\| \left\| \boldsymbol{\Psi}_{P(nh),h}(\boldsymbol{m}(nh)) - \boldsymbol{\Phi}_h\left(x^{(0)}(nh)\right) \right\| \right\|_h \\ &\quad \leq \left\| \left\| \boldsymbol{\Psi}_{P(nh),h}(\boldsymbol{m}(nh)) - \boldsymbol{\Phi}_h\left(m^{(0)}(nh)\right) \right\| \right\|_h \\ &\qquad + \left\| \left\| \boldsymbol{\Phi}_h\left(m^{(0)}(nh)\right) - \boldsymbol{\Phi}_h\left(x^{(0)}(nh)\right) \right\| \right\|_h. \end{aligned} \tag{C.57}$$

The proof is concluded by applying Lemma C.7.5 to the first and Lemma C.3.3 to the second summand of this bound (as well as recalling from eq. (C.26) that $\|\varepsilon^{(0)}(nh)\| = \|m^{(0)}(nh) - x^{(0)}(nh)\|$). □

C.7.5 Global convergence rates

With the above bounds in place, we can now prove global convergence rates.

Theorem C.7.7 (Global truncation error)**.** *Under Assumptions C.1, C.2, C.3, and C.4 and for all sufficiently small $h > 0$,*

$$\max_{n\in[T/h+1]} \left\|\varepsilon^{(0)}(nh)\right\| \leq \max_{n\in[T/h+1]} |||\boldsymbol{\varepsilon}(nh)|||_h \leq K(T)h, \tag{C.58}$$

where $K(T) > 0$ is a constant that depends on T, but not on h.

Remark C.7.8. *Theorem C.7.7 not only implies that the truncation error $\|\varepsilon^{(0)}(nh)\|$ on the solution of eq.* (C.1) *has global order h, but also (by eq.* (C.19)*) that the truncation error $\|\varepsilon^{(1)}(nh)\|$ on the derivative is uniformly bounded by a constant K independent of h. The convergence rate of this theorem is sharp in the sense that it cannot be improved over all f satisfying Assumption C.1 since it is one order worse than the local convergence rate implied by Theorem C.6.2.*

Proof. Using $\left\|\varepsilon^{(0)}(nh)\right\| \leq |||\boldsymbol{\varepsilon}(nh)|||_h$ (due to eq. (C.19)), the bound eq. (C.56), a telescoping sum, and $|||\boldsymbol{\varepsilon}(0)|||_h \leq Kh^2$ (by Assumption C.3), we obtain, for all sufficiently small $h > 0$, that

$$\begin{aligned}
&|||\boldsymbol{\varepsilon}((n+1)h)|||_h - |||\boldsymbol{\varepsilon}(nh)|||_h \\
&\overset{\text{eq. (C.19)}}{\leq} |||\boldsymbol{\varepsilon}((n+1)h)|||_h - \left\|\varepsilon^{(0)}(nh)\right\| \\
&\overset{\text{eq. (C.56)}}{\leq} Kh^2 + Kh\left\|\varepsilon^{(0)}(nh)\right\| \\
&\overset{\text{eq. (C.19)}}{\leq} Kh^2 + Kh|||\boldsymbol{\varepsilon}(nh)|||_h \\
&\overset{\text{(tel. sum)}}{=} Kh^2 + |||\boldsymbol{\varepsilon}(0)|||_h \\
&\qquad + Kh\sum_{l=0}^{n-1}\left(|||\boldsymbol{\varepsilon}((l+1)h)|||_h - |||\boldsymbol{\varepsilon}(lh)|||_h\right) \\
&\overset{(|||\boldsymbol{\varepsilon}(0)|||_h \leq Kh^2)}{\leq} Kh^2 \\
&\qquad\qquad + Kh\sum_{l=0}^{n-1}\left(|||\boldsymbol{\varepsilon}((l+1)h)|||_h - |||\boldsymbol{\varepsilon}(lh)|||_h\right).
\end{aligned} \tag{C.59}$$

Now, by a special version of the discrete Grönwall inequality (Clark, 1987), if z_n and g_n are sequences of real numbers (with $g_n \geq 0$), $c \geq 0$ is a nonnegative constant, and if

$$z_n \leq c + \sum_{l=0}^{n-1} g_l z_l, \qquad \text{for all } n \in \mathbb{N}, \tag{C.60}$$

then

$$z_n \leq c \prod_{l=0}^{n-1} (1 + g_l) \leq c \exp\left(\sum_{l=0}^{n-1} g_l \right), \qquad \text{for all } n \in \mathbb{N}.$$

Application of this inequality to eq. (C.59) with $z_n \coloneqq \|\!|\boldsymbol{\varepsilon}((n+1)h)|\!\|_h - \|\!|\boldsymbol{\varepsilon}(nh)|\!\|_h$, $g_n \coloneqq Kh$, and $c \coloneqq Kh^2$ yields

$$\|\!|\boldsymbol{\varepsilon}((n+1)h)|\!\|_h - \|\!|\boldsymbol{\varepsilon}(nh)|\!\|_h \leq K(T)h^2 \exp(nKh) \tag{C.61}$$

$$\overset{n \leq T/h}{\leq} K(T)h^2. \tag{C.62}$$

By another telescoping sum argument and $\|\!|\boldsymbol{\varepsilon}(0)|\!\|_h \leq Kh^2$, we obtain

$$\|\!|\boldsymbol{\varepsilon}(nh)|\!\|_h \overset{\text{(tel. sum)}}{=} \sum_{l=0}^{n-1} (\|\!|\boldsymbol{\varepsilon}((l+1)h)|\!\|_h - \|\!|\boldsymbol{\varepsilon}(lh)|\!\|_h) + \|\!|\boldsymbol{\varepsilon}(0)|\!\|_h \tag{C.63}$$

$$\overset{\text{eq. (C.62)}}{\leq} nK(T)h^2 + Kh^2 \tag{C.64}$$

$$\overset{n \leq T/h}{\leq} K(T)h + Kh^2 \tag{C.65}$$

$$\leq K(T)h + Kh^2, \tag{C.66}$$

for all sufficiently small $h > 0$. Recalling that $\left\|\varepsilon^{(0)}(nh)\right\| \leq \|\!|\boldsymbol{\varepsilon}(nh)|\!\|_h$, by eq. (C.19), concludes the proof. □

C.8 Calibration of credible intervals

In PN, one way to judge calibration of a Gaussian output $\mathcal{N}(m, V)$ is to check whether the implied 0.95 credible interval $[m - 2\sqrt{V}, m + 2\sqrt{V}]$ contracts at the same rate as the convergence rate of the posterior mean to the true quantity of interest. For the filter, this would mean that the rate of contraction of $\max_n \sqrt{P_{00}(nh)}$ should contract at the same rate as $\max_{n \in [T/h+1]} \|\varepsilon^{(0)}(nh)\|$ (recall its rates from Theorem C.7.7). Otherwise, for a higher or lower rate of the interval it would eventually be under- or overconfident, as $h \to 0$. The following proposition shows—in light of the sharp bound eq. (C.58) on the global error—that the credible intervals are well calibrated in this sense if $p \in [1, \infty]$.

Theorem C.8.1. *Under Assumption C.2 and for $R \equiv Kh^p$, $p \in [0, \infty]$, as well as sufficiently small $h > 0$,*

$$\max_{n \in [T/h+1]} P_{00}^{-}(nh) \leq K(T) h^{(p+1) \wedge 2}, \qquad \text{and} \tag{C.67}$$

$$\max_{n \in [T/h+1]} P_{00}(nh) \leq K(T) h^{(p+1) \wedge 2}. \tag{C.68}$$

Proof. See Appendix C.20. □

C.9 Numerical experiments

In this section, we empirically assess the following hypotheses:

(i) the worst-case convergence rates from Theorem C.7.7 hold not only for $q = 1$ but also for $q \in \{2, 3\}$ (see Appendix C.9.1),

(ii) the convergence rates of the credible intervals from Theorem C.8.1 hold true (see Appendix C.9.2), and

(iii) Assumption C.4 is necessary to get these convergence rates (see Appendix C.9.3).

The three hypotheses are all supported by the experiments. These experiments are subsequently discussed in Appendix C.9.4. Appendix C.14 contains an additional experiment illustrating the convergence rates for the state misalignment δ from Lemma C.7.4.

C.9.1 Global convergence rates for $q \in \{1, 2, 3\}$

We consider the following three test IVPs: Firstly, a the following linear ODE

$$\begin{aligned} \dot{x}(t) &= \Lambda x(t), \ \forall t \in [0, 10], \\ \text{with } \Lambda = \begin{pmatrix} 0 & -\pi \\ \pi & 0 \end{pmatrix} &\text{ and } x(0) = (0, 1)^\intercal, \end{aligned} \tag{C.69}$$

and has the harmonic oscillator

$$x(t) = e^{t\Lambda} x(0) = \begin{pmatrix} -\sin(t\pi) & \cos(t\pi) \end{pmatrix}^\intercal \tag{C.70}$$

as a solution. Secondly, the logistic equation

$$\begin{aligned} \dot{x}(t) &= \lambda_0 x(t) \left(1 - x(t)/\lambda_1\right), \ \forall t \in [0, 1.5], \\ \text{with } (\lambda_0, \lambda_1) &= (3, 1) \text{ and } x(0) = 0.1, \end{aligned} \tag{C.71}$$

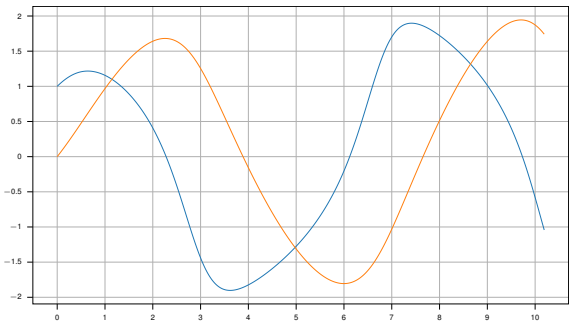


Figure C.1: True solution of the FitzHugh–Nagumo model, eq. (C.73); x_1 in blue and x_2 in orange.

which has the logistic curve

$$x(t) = \frac{\lambda_1 \exp(\lambda_0 t) x(0)}{\lambda_1 + x(0)(\exp(\lambda_0 t) - 1)}. \tag{C.72}$$

And, thirdly, the FitzHugh–Nagumo model

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} x_1(t) - \frac{x_1(t)}{3} - x_2(t) \\ \frac{1}{\tau}\left(x_1(t) + a - b x_2(t)\right), \end{pmatrix}, \forall t \in [0, 10] \tag{C.73}$$

with $(a, b, c) = (0.08, 0.07, 1.25)$ and $x(0) = (1, 0)$ which does not have a closed-form solution. Its solution, which we approximate by Euler's method with a step size of $h = 10^{-6}$ for the below experiments, is depicted in Figure C.1. We numerically solve these three IVPs with the Gaussian ODE filter for multiple step sizes $h > 0$ and with a q-times IBM prior (i.e. $\theta = 0$ in eq. (C.5)) for $q \in \{1, 2, 3\}$ and scale $\sigma = 20$. As a measurement model, we employ the minimal $R \equiv 0$ and maximal measurement variance $R \equiv K_R h^q$ (for $h \leq 1$) which are permissible under Assumption C.4 whose constant $K > 0$ is denoted explicitly by K_R in this section. The resulting convergence rates of global errors $\|m(T) - x(T)\|$ are depicted in a work-precision diagram in Figure C.2; cf. Hairer *et al.* (1987, Chapter II.1.4) for such diagrams for Runge–Kutta methods. Now, recall from Theorem C.7.7 that, for $q = 1$, the global truncation error decreases at a rate of at least h^q in the worst case. Figure C.2 shows that these convergence rates of q^{th} order hold true in the considered examples for values of up to $q = 3$ if $R \equiv 0$ and, for values of up to $q = 3$. In the case of $R \equiv 0$, even $(q+1)^{\text{th}}$ order convergence rates appear to hold true for all three ODEs and $q \in \{1, 2, 3\}$. Note that it is more difficult to validate these convergence rates for $q = 4$, for all three test problems and small $h > 0$, since numerical instability can contaminate the analytical rates.

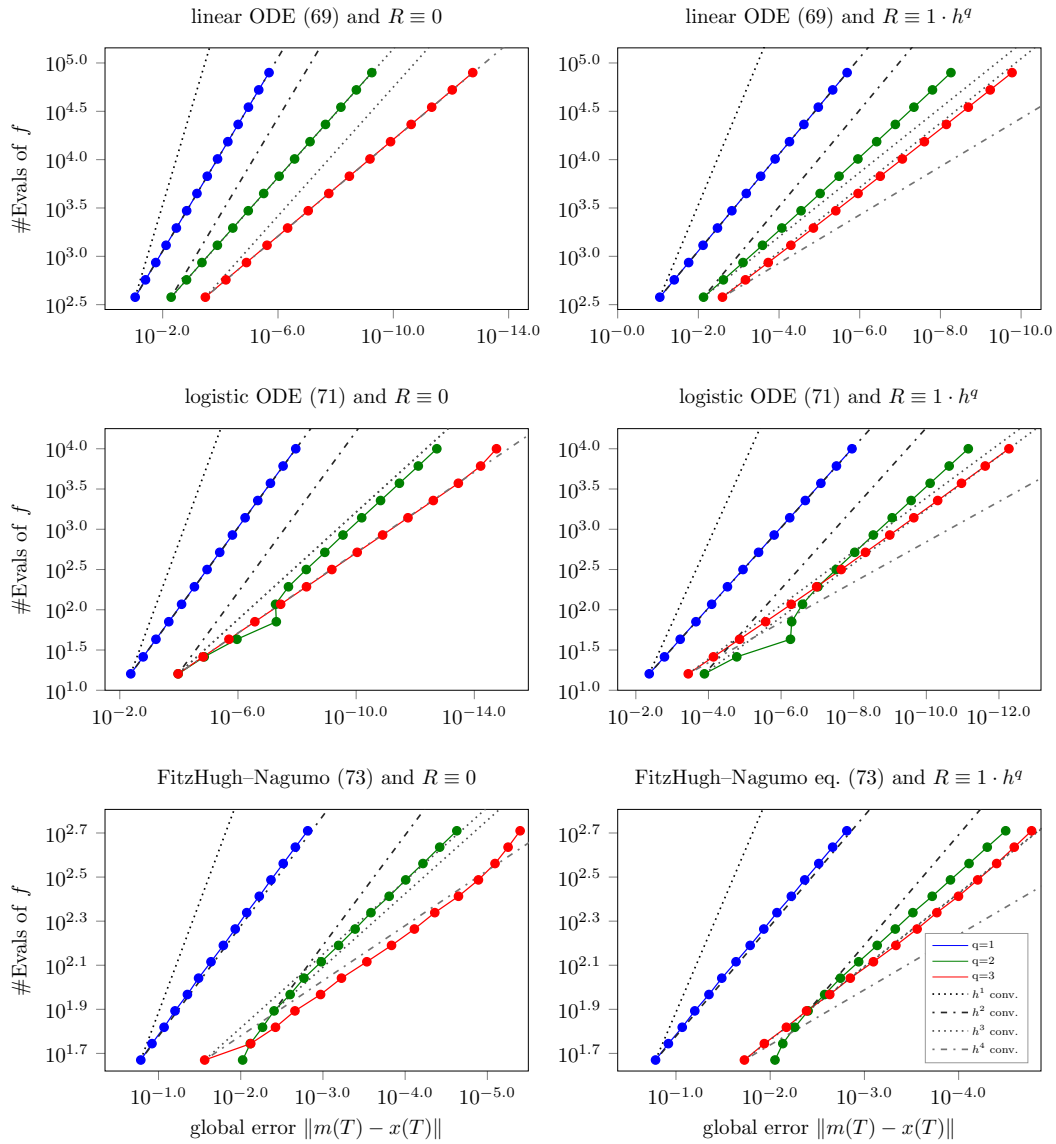


Figure C.2: Work-precision diagrams for the Gaussian ODE filter with q-times IBM prior, for $q \in \{1, 2, 3\}$, applied to the linear eq. (C.71), logistic ODE eq. (C.69) and the FitzHugh–Nagumo model. The number of function evaluations (# Evals of f), which is inversely proportional to the step size h, is plotted in color against the logarithmic global error at the final time T. The (dash-)dotted gray lines visualize idealized convergence rates of orders one to four. The left and right columns employ the minimal $R \equiv 0$ and maximal measurement variance $R \equiv K_R h^q$ ($K_R = 1$) which are permissible under Assumption C.4.

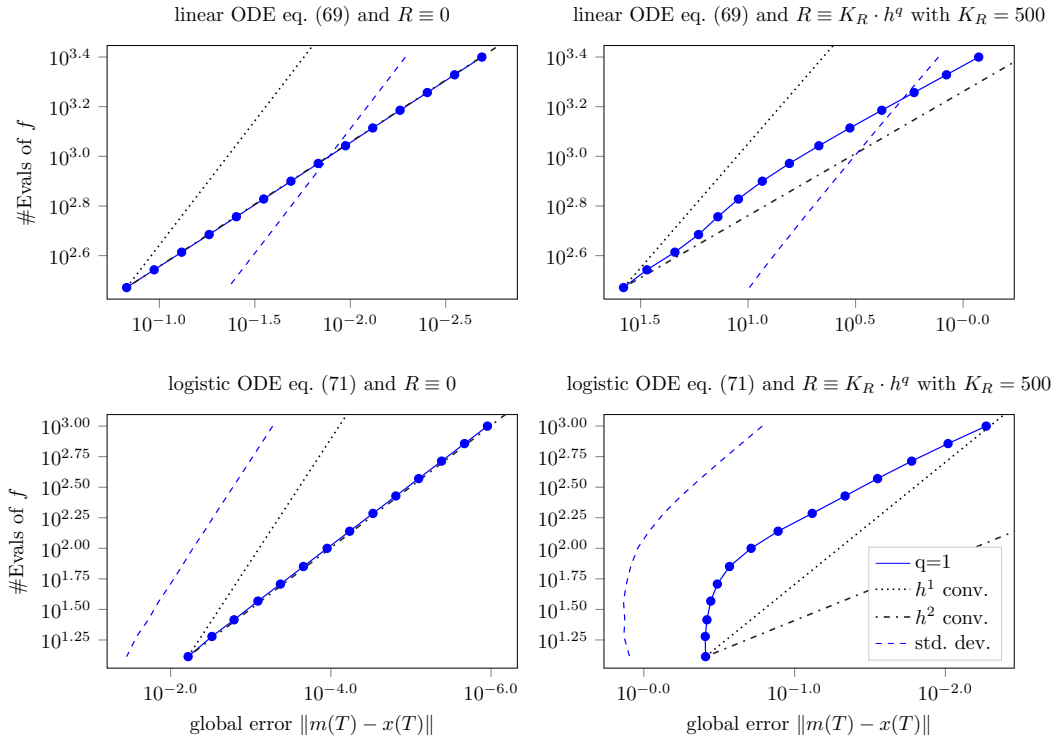


Figure C.3: Work-precision diagrams for the Gaussian ODE filter with q-times IBM prior, for $q = 1$, applied to the linear eq. (C.69) and logistic ODE eq. (C.71) in the upper and lower row, respectively. The number of function evaluations (# Evals of f), which is inversely proportional to the step size h, is plotted in color against the logarithmic global error at the final time T. The (dash-)dotted gray lines visualize idealized convergence rates of orders one and two. The dashed blue lines show the posterior standard deviations calculated by the filter. The left and right columns, respectively, employ the minimal $R \equiv 0$ and maximal measurement variance $R \equiv K_R h^q$ ($K_R = 5.00 \times 10^3$) which are permissible under Assumption C.4.

C.9.2 Calibration of credible intervals

To demonstrate the convergence rates of the posterior credible intervals proved in Theorem C.8.1, we now restrict our attention to the case of $q = 1$, that was considered therein. As in Appendix C.9.1, we numerically solve the IVPs eqs. (C.69) and (C.71) with the Gaussian ODE filter with a once IBM prior with fixed scale $\sigma = 1$. We again employ the minimal $R \equiv 0$ and maximal measurement variance $R \equiv K_R h^q$ (for $h \leq 1$) which are permissible under Assumption C.4 as a measurement model. Figure C.3 depicts the resulting convergence rates in work-precision diagrams. As the parallel standard deviation (std. dev.) and h^1 convergence curves show, the credible intervals asymptotically contract at the rate of h^1 guaranteed by Theorem C.8.1. In all four diagrams of Figure C.3, the global error shrinks at a faster rate than the width of the credible intervals.

This is unsurprising for $R \equiv 0$ as we have already observed convergence rates of h^{q+1} in this case. While this effect is less pronounced for $R \equiv K_R h^q$, it still results in underconfidence as $h \to 0$. Remarkably, the shrinking of the standard deviations seems to be 'adaptive' to the numerical error—by which we mean that, as long as the numerical error hardly decreases (up to $10^{1.75}$ evaluations of f), the standard deviation also stays almost constant, before adopting its h^1 convergence asymptotic (from $\approx 10^{2.00}$).

C.9.3 Necessity of Assumption C.4

Having explored the asymptotic properties under Assumption C.4 in Appendices C.9.1 and C.9.2, we now turn our attention to the question of whether this assumption is necessary to guarantee the convergence rates from Theorems C.7.7 and C.8.1. This question is of significance, because Assumption C.4 is weaker than the $R \equiv 0$ assumption of the previous theoretical results (i.e. Proposition 1 and Theorem 1 in Schober *et al.* (2019)) and it is not self-evident that it cannot be further relaxed. To this end, we numerically solve the logistic ODE eq. (C.71) with the Gaussian ODE filter with a once IBM prior with fixed scale $\sigma = 1$ and measurement variance $R \equiv K_R h^{1/2}$, which is impermissible under Assumption C.4, for increasing choices of K_R from 0.00×10^0 to 1.00×10^7. In the same way as in Figure C.3, the resulting work-precision diagrams are plotted in Figure C.4.

In contrast to the lower left diagram in Figure C.3, which presents the same experiment for $R \equiv K_R h^q$ (the maximal measurement variance permissible under Assumption C.4), the rate of h^2, that is again observed for $K_R = 0$ in the first diagram, is already missed for $K_R = 1.00 \times 10^0$ in the second diagram. With growing constants, the convergence rates of the actual errors as well as the expected errors (standard deviation) decrease from diagram to diagram. In the center diagram with $K_R = 3.73 \times 10^3$, the rates are already slightly worse than the h^1 convergence rates guaranteed by Theorems C.7.7 and C.8.1 under Assumption C.4, whereas, for $K_R = 5.00 \times 10^3$, the convergence rates in the lower left plot of Figure C.3 were still significantly better than h^1. For the greater constants up to $K_R = 1.00 \times 10^7$, the rates even become significantly lower. Notably, as in the lower right diagram of Figure C.3, the slope of the standard deviation curve matches the slope of the global error curve, as can be seen best in the lower right subfigure—thereby asymptotically exhibiting neither over- nor underconfidence. These experiments suggest that the convergence rates from Theorems C.7.7 and C.8.1 do not hold in general for $R \equiv K_R h^{1/2}$. Hence, it seems likely that Assumption C.4 is indeed necessary for our results and cannot be further relaxed without lowering the implied worst-case convergence rates.

C.9.4 Discussion of experiments

Before proceeding to our overall conclusions, we close this section with a comprehensive discussion of the above experiments. First and foremost, the experiments in Ap-

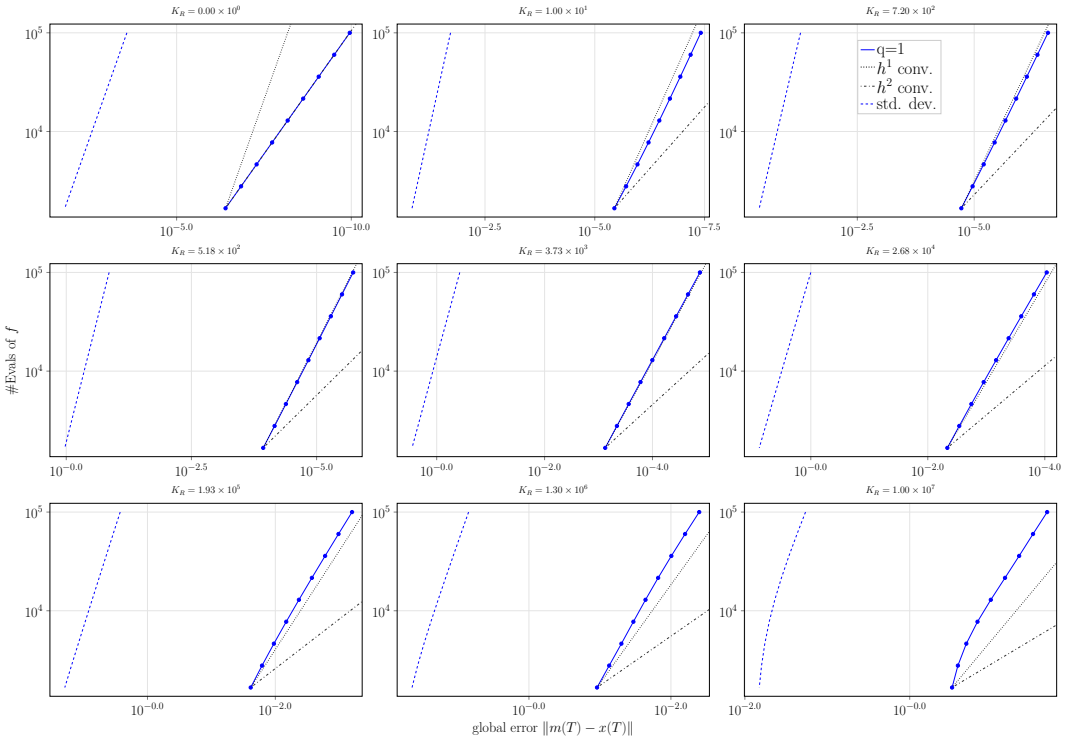


Figure C.4: Work-precision diagrams for the Gaussian ODE filter with q-times IBM prior, for $q = 1$ and $R \equiv K_R h^{1/2}$, applied to the logistic ODE eq. (C.71) for increasing values of K_R. The number of function evaluations (# Evals of f), which is inversely proportional to the step size h, is plotted in blue against the logarithmic global error at the final time T. The (dash-)dotted gray lines visualize idealized convergence rates of orders one and two. The dashed blue lines show the posterior standard deviations calculated by the filter.

pendix C.9.1 suggest that Theorem C.7.7, the main result of this paper, might be generalizable to $q \in \{2, 3\}$ and potentially even higher $q \in \mathbb{N}$—although unresolved issues with numerical instability for small step sizes prevent us from confidently asserting that these theoretical results would hold in practice for $q \geq 4$. Moreover, we demonstrated the contraction rates of the posterior credible intervals from Theorem C.8.1 and evidence for the necessity of Assumption C.4 in Appendices C.9.2 and C.9.3. The asymptotics revealed by these experiments can be divided by the employed measurement model into three cases: the zero-noise case $R \equiv 0$, the permissible non-zero case $R \leq K_R h^q$ (under Assumption C.4) and the non-permissible case $R \not\leq K_R h^q$. First, if $R \equiv 0$, the diagrams in the left column of Figure C.2 reaffirm the h^{q+1} convergence reported for $q \in \{1, 2\}$ in Schober *et al.* (2019, Figure 4) and extend them to $q = 3$ (see Appendix C.10 for a discussion on why we expect the above global convergence proofs to be extensible to $q \geq 2$)

The contraction rates of the credible intervals, for $q = 1$, appear to be asymptotically underconfident in this case as they contract faster than the error. This underconfidence is not surprising in so far as the posterior standard deviation is a worst-case bound for systems modeled by the prior, while the convergence proofs require smoothness of the solution of one order higher than sample paths from the prior. This is a typical result that highlights an aspect known to, but on the margins of classic analysis: The class of problems for which the algorithm converges is rougher than the class on which convergence order proofs operate. How to remedy such overly-cautious UQ remains an open research question in PN as well as classical numerical analysis.

Secondly, in the case of $R > 0$, as permissible under Assumption C.4, the convergence rates are slightly reduced compared to the case $R \equiv 0$, exhibiting convergence between h^q and h^{q+1}. The asymptotic underconfidence of the credible intervals, however, is either reduced or completely removed as depicted in the right column of Figure C.3. Thirdly, in the final case of an impermissibly large $R > 0$, the h^q convergence speed guaranteed by Theorem C.7.7 indeed does not necessarily hold anymore—as depicted in Figure C.4. Note, however, that even then the convergence rate is only slightly worse than h^q. The asymptotic UQ matches the observed global error in this case, as the parallel standard deviation and the h^1 curves in all but the upper left $R \equiv 0$ diagram show.

Overall, the experiments suggest that, in absence of statistical noise on f, a zero-variance measurement model yields the best convergence rates of the posterior mean. Maybe this was expected as, in this case, R only models the inaccuracy from the truncation error, that ideally should be treated adaptively (Kersting and Hennig, 2016, Section 2.2). The convergence rates of adaptive noise models should be assessed in future work. As the observed convergence rates in practice sometimes outperform the proved worst-case convergence rates, we believe that an average-case analysis of the filter in the spirit of Ritter (2000) may shed more light upon the expected practical performance. Furthermore, it appears that the UQ becomes asymptotically accurate as well as adaptive to the true numerical error as soon as the $R > 0$ is large enough. This reinforces our hope that these algorithms will prove useful for IVPs when f is estimated itself (Hennig

et al., 2015, Section 3(d)), thereby introducing a $R > 0$.

C.10 Conclusions

We presented a worst-case convergence rate analysis of the Gaussian ODE filter, comprising both local and global convergence rates. While local convergence rates of h^{q+1} were shown to hold for all $q \in \mathbb{N}$, IBM and IOUP prior as well as any noise model $R \geq 0$, our global convergence results is restricted to the case of $q = 1$, IBM prior and fixed noise model $R \equiv Kh^p$ with $p \in [1, \infty]$. While a restriction of the noise model seems inevitable, we believe that the other two restrictions can be lifted: In light of Theorem C.6.2, global convergence rates for the IOUP prior might only require an additional assumption that ensures that all possible data sequences $\{y(nh); n = 1, \ldots, T/h\}$ (and thereby all possible q^{th}-state sequences $\{m^{(q)}(nh); n = 0, \ldots, T/h\}$) remain uniformly bounded (see discussion in Remark C.6.3). For the case of $q \geq 2$, it seems plausible that a proof analogous to the presented one would already yield global convergence rates of order h^q,[3] as suggested for $q \in \{2, 3\}$ by the experiments in Appendix C.9.1.

The orders of the predictive credible intervals can also help to intuitively explain the threshold of $p = 1$ (or maybe more generally: $p = q$; see Figure C.2) below which the performance of the filter is not as good, due to eqs. (C.45) to (C.49): According to Kersting and Hennig (2016, Equation (20)), the 'true' (push-forward) variance on $y(t)$ given the predictive distribution $\mathcal{N}(m^-(t), P^-(t))$ is equal to the integral of $ff^\intercal$ with respect to $\mathcal{N}(m^-(t), P^-(t))$, whose maximum over all time steps, by eq. (C.67), has order $\mathcal{O}(h^{\frac{p+1}{2} \wedge 1})$ if $ff^\intercal$ is globally Lipschitz—since $P^-(t)$ enters the argument of the integrand $ff^\intercal$, after a change of variable, only under a square root. Hence, the added 'statistical' noise R on the evaluation of f is of lower order than the accumulated 'numerical' variance $P^-(t)$ (thereby preventing numerical convergence) if and only if $p < 1$. Maybe this, in the spirit of Hennig *et al.* (2015, Subsection 3(d)), can serve as a criterion for vector fields f that are too roughly approximated for a numerical solver to output a trustworthy result, even as $h \to 0$.

Furthermore, the competitive practical performance of the filter, as numerically demonstrated in Schober *et al.* (2019, Section 5), might only be completely captured by an average-case analysis in the sense of Ritter (2000), where the average error is computed with respect to some distribution $p(f)$, i.e. over a distribution of ODEs. To comprehend this idea, recall that the posterior filtering mean is the Bayes estimator with minimum mean squared error in linear dynamical systems with Gauss–Markov prior (as defined by the SDE eq. (C.2)), i.e. when the data is not evaluations of f but real i.i.d. measurements, as well as in the special case of $\dot{x}(t) = f(t)$, when the IVP simplifies to a

[3] According to Loscalzo and Talbot (1967), the filter might, however, suffer from numerical instability for high choices of q. (See Schober *et al.* (2019, Section 3.1) for an explanation of how such results on spline-based methods concern the ODE filter.)

quadrature problem—see Solak *et al.* (2003) and O'Hagan (1991, Section 2.2) respectively. In fact, the entire purpose of the update step is to correct the prediction in the (on average) correct direction, while a worst-case analysis must assume that it corrects in the worst possible direction in every step—which we execute by the application of the triangle inequality in eq. (C.28) resulting in a worst-case upper bound that is the sum of the worst-case errors from prediction and update step. An analysis of the probabilities of 'good' vs. 'bad' updates might therefore pave the way for such an average-case analysis in the setting of this paper. Since, in practice, truncation errors of ODE solvers tend to be significantly smaller than the worst case—as mirrored by the experiments in Appendix C.9—such an analysis might be useful for applications.

Lastly, we hope that the presented convergence analysis can lay the foundations for similar results for the novel ODE filters (extended KF, unscented KF, particle filter) introduced in Tronarp *et al.* (2019a), and can advance the research on uncertainty-aware likelihoods for inverse problems by ODE filtering (Kersting *et al.*, 2020b, Section 3).

Acknowledgements

The authors are grateful to Han Cheng Lie for discussions and feedback to early versions of what is now Appendices C.3 and C.5 of this work, as well as Appendix C.7.5. The authors also thank Michael Schober for valuable discussions and helpful comments on the manuscript.

TJS's work has been partially supported by the Freie Universität Berlin within the Excellence Initiative of the German Research Foundation (DFG), by the DFG through grant CRC 1114 "Scaling Cascades in Complex Systems", and by the National Science Foundation under grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute (SAMSI) and SAMSI's QMC Working Group II "Probabilistic Numerics". HK and PH gratefully acknowledge financial support by the German Federal Ministry of Education and Research through BMBF grant 01IS18052B (ADIMEM). PH also gratefully acknowledges support through ERC StG Action 757275 / PANAMA.

Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the above-named institutions and agencies.

Conflict of interest

The authors declare that they have no conflict of interest.

Supplementary Material for Kersting *et al.* (2020a)

C.11 Supplement I: Derivation of A and Q

As derived in Särkkä (2006, Section 2.2.6) the solution of the SDE eq. (C.2), i.e.

$$\begin{aligned} \mathrm{d}\boldsymbol{X}(t) &= \begin{pmatrix} \mathrm{d}X^{(0)}(t) \\ \vdots \\ \mathrm{d}X^{(q-1)}(t) \\ \mathrm{d}X^{(q)}(t) \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 0 & 1 & 0 \dots & 0 \\ \vdots & \ddots & \ddots & 0 \\ \vdots & & \ddots & 1 \\ c_0 & \dots & \dots & c_q \end{pmatrix}}_{=:F} \underbrace{\begin{pmatrix} X^{(0)}(t) \\ \vdots \\ X^{(q-1)}(t) \\ X^{(q)}(t) \end{pmatrix}}_{=\boldsymbol{X}(t)} \mathrm{d}t + \underbrace{\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sigma \end{pmatrix}}_{=:L} \mathrm{d}B(t), \end{aligned} \tag{C.74}$$

where we omitted the index j for simplicity, is a Gauss–Markov process with mean $m(t)$ and covariance matrix $P(t)$ given by

$$m(t) = A(t)m(0), \qquad P(t) = A(t)P(0)A(t)^{\intercal} + Q(t), \tag{C.75}$$

where the matrices $A,\ Q \in \mathbb{R}^{(q+1)\times(q+1)}$ are explicitly defined by

$$A(t) = \exp(tF), \tag{C.76}$$

$$Q(t) \coloneqq \int_0^t \exp(F(t-\tau))LL^{\intercal}\exp(F(t-\tau))^{\intercal}\,\mathrm{d}\tau. \tag{C.77}$$

Parts of the following calculation can be found in Magnani *et al.* (2017). If we choose $c_0, \dots, c_{q-1} = 0$ and $c_q = -\theta$ (for $\theta \geq 0$) in eq. (C.74) the unique strong solution of the SDE is a q-times IOUP, if $\theta > 0$, and a q-times IBM, if $\theta = 0$; see e.g. Karatzas and Shreve (1991, Chapter 5: Example 6.8). By eq. (C.77) and

$$\left((tF)^k\right)_{i,j} = t^k\left[\mathbb{I}_{j-i=k} + (-\theta)^{k+i-q}\mathbb{I}_{\{j=q,\ i+k\geq q\}}\right], \tag{C.78}$$

it follows that

$$\begin{aligned} A(t)_{ij} &= \left(\sum_{k=0}^{\infty} \frac{(tF)^k}{k!}\right)_{i,j} \\ &= \begin{cases} \mathbb{I}_{i\leq j} \frac{t^{j-i}}{(j-i)!}, & \text{if } j \neq q, \\ \frac{1}{(-\theta)^{q-i}} \sum_{k=q-i}^{\infty} \frac{(-\theta t)^k}{k!}, & \text{if } j = q, \end{cases} \\ &= \begin{cases} \mathbb{I}_{i\leq j} \frac{t^{j-i}}{(j-i)!}, & \text{if } j \neq q, \\ \frac{t^{q-i}}{(q-i)!} - \theta \sum_{k=q+1-i}^{\infty} \frac{(-\theta)^{k+i-q-1} t^k}{k!}, & \text{if } j = q. \end{cases} \end{aligned} \tag{C.79}$$

Analogously, it follows that

$$\begin{aligned} &\exp(F(t-\tau)) \\ &= \begin{cases} \mathbb{I}_{i\leq j} \frac{(t-\tau)^{j-i}}{(j-i)!}, & \text{if } j \neq q, \\ \frac{(t-\tau)^{q-i}}{(q-i)!} - \theta \sum_{k=q+1-i}^{\infty} \frac{(-\theta)^{k+i-q-1} (t-\tau)^k}{k!}, & \text{if } j = q, \end{cases}. \end{aligned} \tag{C.80}$$

If we insert eq. (C.80) into eq. (C.77), then we obtain, by the sparsity of L, that

$$\begin{aligned} &Q(t)_{ij} \\ &= \frac{\sigma^2}{(-\theta)^{2q-i-j}} \int_0^t \left(\sum_{k=q-i}^{\infty} \frac{(-\theta\tau)^k}{k!}\right) \left(\sum_{l=q-j}^{\infty} \frac{(-\theta\tau)^l}{l!}\right) \, \mathrm{d}\tau, \end{aligned} \tag{C.81}$$

and the dominated convergence theorem (with dominating function $\tau \mapsto e^{2\theta\tau}$) yields

$$\begin{aligned} Q(t)_{ij} &= \frac{\sigma^2}{(-\theta)^{2q-i-j}} \sum_{k=q-i}^{\infty} \sum_{l=q-j}^{\infty} \int_0^t \frac{(-\theta\tau)^{k+l}}{k!l!} \, \mathrm{d}\tau \\ &= \frac{\sigma^2}{(-\theta)^{2q-i-j}} \sum_{k=q-i}^{\infty} \sum_{l=q-j}^{\infty} (-\theta)^{k+l} \frac{t^{k+l+1}}{(k+1+l)k!l!}. \end{aligned} \tag{C.82}$$

Now, by extracting the first term and noticing that the rest of the series is in $\Theta(t^{2q+2-i-j})$, it follows that

$$\begin{aligned} Q(t)_{ij} = \sigma^2 &\frac{t^{2q+1-i-j}}{(2q+1-i-j)(q-i)!(q-j)!} \\ &+ \Theta\left(t^{2q+2-i-j}\right). \end{aligned} \tag{C.83}$$

C.12 Supplement II: Extension to x with dependent dimensions

The algorithm in Appendix C.2.2 employs a prior $\boldsymbol{X}$ with independent dimensions $\boldsymbol{X}_j = \left(X_j^{(0)}, \ldots, X_j^{(q)}\right)^\intercal$, $j \in [d]$, by eq. (C.2). While this constitutes a loss of generality for our new theoretical results, which do not immediately carry over to the case of x with dependent dimensions, it is not a restriction to the class of models the algorithm can employ. To construct such a prior $\boldsymbol{X}$, we first stack its dimensions into the random vector $\boldsymbol{X} = (\boldsymbol{X}_0^\intercal, \ldots, \boldsymbol{X}_{d-1}^\intercal)^\intercal$, choose symmetric positive semi-definite matrices $K_x, K_\varepsilon \in \mathbb{R}^{d\times d}$, and define, using the Kronecker product $\otimes$, its law according to the SDE

$$\mathrm{d}\boldsymbol{X}(t) = [K_x \otimes F]\, \boldsymbol{X}(t)\, \mathrm{d}t + [K_\varepsilon \otimes L]\, \mathrm{d}B(t), \tag{C.84}$$

with initial condition $\boldsymbol{X}(0) \sim \mathcal{N}(m(0), P(0))$, mean $m(0) \in \mathbb{R}^{d(q+1)}$ and covariance matrix $P(0) \in \mathbb{R}^{d(q+1)\times d(q+1)}$, as well as an underlying d-dimensional Brownian motion B (independent of $\boldsymbol{X}(0)$). Now, insertion of $K_x \otimes F$ and $K_\varepsilon \otimes L$ for F and L into eq. (C.77) yields new predictive matrices $\tilde{A}$ and $\tilde{Q}$. If we now choose $K_x = I_d$ and $K_\varepsilon = I_d$, substitute $\tilde{A}$ and $\tilde{Q}$ for A and Q in eqs. (C.9) and (C.10), and use the $d(q+1)$-dimensional GP $\boldsymbol{X}$ from eq. (C.84) with $m(0) \in \mathbb{R}^{d(q+1)}$ and $P(0) \in \mathbb{R}^{d(q+1)\times d(q+1)}$ as a prior, we have equivalently defined the version of Gaussian ODE filtering with independent dimensions from Appendix C.2.2. If we, however, choose different symmetric positive semi-definite matrices for K_x and K_ε, we introduce, via $\tilde{A}$ and $\tilde{Q}$, a correlation in the development of the solution dimensions $(x_0, \ldots, x_{d-1})^\intercal$ as well as the error dimensions $(\varepsilon_0, \ldots, \varepsilon_d)^\intercal$ respectively. Note that, while K_ε plays a similar role as C^h in Conrad *et al.* (2017, Assumption 1) in correlating the numerical errors, the matrix K_x additionally introduces a correlation of the numerical estimates, that is m, along the time axis. Even more flexible correlation models (over all modeled derivatives) can be employed by inserting arbitrary matrices (of the same dimensionality) for $K_x \otimes F$ and $K_\varepsilon \otimes L$ in eq. (C.84), but such models seem hard to interpret. For future research, it would be interesting to examine whether such GP models with dependent dimensions are useful in practice. There are first publications (Xiaoyue *et al.*, 2018; Gessner *et al.*, 2019) on this topic for integrals, but not yet for ODEs.

C.13 Supplement III: Illustrative example

To illustrate the algorithm defined in Appendix C.2.2, we apply it to a special case of the Riccati equation (Davis, 1962, p. 73)

$$\frac{\mathrm{d}x}{\mathrm{d}t}(t) = f(x(t)) = -\frac{(x(t))^3}{2}, \quad x(0) = 1, \tag{C.85}$$

$$\left(\text{solution: } x(t) = (t+1)^{-1/2}\right), \tag{C.86}$$

with step size $h = 0.1$, measurement noise $R = 0.0$ (for simplicity) as well as prior hyperparameters $q = 1$, $\sigma^2 = 10.0$ and $c_i = 0$ for all $i \in [q+1]$ (recall eq. (C.2)), i.e. with a 1-times integrated Brownian motion prior whose drift and diffusion matrices are, by eq. (C.8), given by

$$A(h) = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}, \qquad Q(h) = \begin{pmatrix} 1/300 & 1/20 \\ 1/20 & 1 \end{pmatrix}. \tag{C.87}$$

As the ODE eq. (C.85) is one-dimensional (i.e. $d = 1$), the dimension index $j \in [d]$ is omitted in this section. Since the initial value and derivative are certain at $x(0) = 1$ and $\dot{x}(0) = f(x_0) = -1/2$, our prior GP is initialized with a Dirac distribution (i.e. $\boldsymbol{X}(0) = (X^{(0)}(0), X^{(1)}(0))^\intercal \sim \delta_{(x_0, f(x_0))} = \delta_{(1,-1/2)}$). Therefore, $\boldsymbol{m}(0) = (1, -1/2)^\intercal$ and $P(0) = 0 \in \mathbb{R}^{2\times 2}$ for the initial filtering mean and covariance matrix. Now, the Gaussian ODE Filter computes the first integration step by executing the prediction step eqs. (C.9) and (C.10)

$$\begin{aligned} \boldsymbol{m}^-(h) &= A(h)\boldsymbol{m}^-(0) \\ &= \left(m^{(0)}(0) + hm^{(1)}(0), m^{(1)}(0)\right)^\intercal \\ &= (19/20, -1/2)^\intercal, \qquad \text{and} \end{aligned} \tag{C.88}$$

$$P^-(h) = 0 + Q(h) = \begin{pmatrix} 1/300 & 1/20 \\ 1/20 & 1 \end{pmatrix}. \tag{C.89}$$

Note that, for all $i \in [q+1]$, $m^{-,(i)}(h)$ is obtained by a $(q-i)^{\text{th}}$-order Taylor expansion of the state $\boldsymbol{m}(0) = (x_0, f(x_0))^\intercal \in \mathbb{R}^{q+1}$. Based on this prediction, the data is then generated by

$$\begin{aligned} y(h) &= f\left(m^{-,(0)}(h)\right) \overset{\text{eq. (C.88)}}{=} f(19/20) \\ &\overset{\text{eq. (C.85)}}{=} -6859/16000 \end{aligned} \tag{C.90}$$

with variance $R = 0.0$. In the subsequent update step eqs. (C.9) and (C.11) to (C.13), a Bayesian conditioning of the predictive distribution eqs. (C.88) and (C.89) on this data

is executed:

$$\begin{aligned}
\boldsymbol{\beta}(h) &= \left(\beta^{(0)}(h), \beta^{(1)}(h)\right)^\intercal \\
&= \left(\frac{P^-(h)_{01}}{(P^-(h))_{11} + R}, \frac{P^-(h)_{11}}{(P^-(h))_{11} + R}\right)^\intercal \\
&\overset{\text{eq. (C.89)}}{=} \left(\frac{1}{20}, 1\right)^\intercal, && \text{(C.91)} \\
r(h) &= y(h) - m^{-,(1)}(h) \\
&\overset{\text{eqs. (C.88),(C.90)}}{=} -6859/16000 + 1/2 \\
&= 1141/16000, && \text{(C.92)} \\
\boldsymbol{m}(h) &\overset{\text{eq. (C.9)}}{=} \begin{pmatrix} m^{-,(0)}(h) + \beta^{(0)}(h) r(h) \\ m^{-,(1)}(h) + \beta^{(1)}(h) r(h) \end{pmatrix} \\
&\overset{\text{eqs. (C.88),(C.91),(C.92)}}{=} \begin{pmatrix} 305141/320000 \\ -6859/16000 \end{pmatrix}, && \text{(C.93)}
\end{aligned}$$

which concludes the step from 0 to h. The next step $h \to 2h$ starts with computing $m^{-,(i)}(2h)$ by a $(q-i)^{\text{th}}$-order Taylor expansion of the i^{th} state $m^{(i)}(h)$, for all $i \in [q+1]$. Note that, now, there is a non-zero *state misalignment* (recall eq. (C.25)):

$$\begin{aligned}
\delta^{(1)}(h) &\overset{eq.\ \text{(C.25)}}{=} \left| m^{(1)}(h) - f\left(m^{(0)}(h)\right) \right| && \text{(C.94)} \\
&= \left| -\frac{6859}{16000} - \frac{1}{2}\left(\frac{305141}{320000}\right)^3 \right| && \text{(C.95)} \\
&\approx 0.00485 > 0 && \text{(C.96)}
\end{aligned}$$

which confirms the exposition on the possibility of $\delta^{(i)} > 0$ from Appendix C.4. Note that δ tends to increase with R; e.g., if $R = 1.0$ in the above example, then $\delta^{(1)}(h) \approx 0.03324$.

C.14 Supplement IV: Experiment for global convergence of state misalignments δ

Figure C.5 depicts the global convergence of the state misalignment $\delta^{(1)}(T)$ in the above example eq. (C.85), as detailed in Appendix C.13, for $q \in \{1, 2, 3\}$. The plotting is analogous to Figure C.2. The resulting convergence rates of h^{q+1} confirm Lemma C.7.4 and suggest that it may also be generalizable to $q \in \{2, 3, \dots\}$.

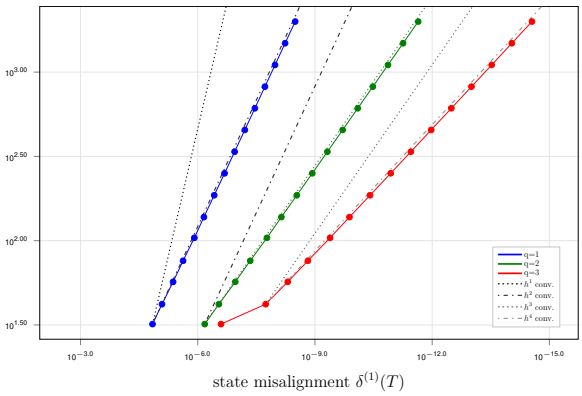


Figure C.5: Work-precision diagram plotting the number of function evaluations (# Evals of f) against the final state misalignment $\delta^{(1)}(T)$ on the Riccati equation eq. (C.85); cf. Figure C.2.

C.15 Supplement V: Proof of eq. (C.23)

We prove the stronger statement

$$\Phi_t^{(i+1)}(a) = f^{(i)}\left(\Phi_t^{(0)}(a)\right), \tag{C.97}$$

from which eq. (C.23) follows by inserting $t = 0$ and $\Phi_0^{(0)}(a) = a$. Hence, it remains to show eq. (C.97).

of eq. (C.97). By induction over $i \in \{0, \dots, q\}$. The base case ($i = 0$) is obtained using the fundamental theorem of calculus and $f^{(1)} = f$: $\Phi_t^{(1)}(a) = f\left(\Phi_t^{(0)}(a)\right) = f^{(1)}\left(\Phi_t^{(0)}(a)\right)$. For the inductive step $(i-1) \to i$, we conclude (using the inductive hypothesis (IH), the chain rule (CR), the base case (BC) and $f^{(i)} = \nabla_x f^{(i-1)} \cdot f$) that

$$\begin{aligned}
\Phi_t^{(i+1)}(a) &= \frac{\mathrm{d}}{\mathrm{d}t}\Phi_t^{(i)}(a) \\
&\overset{\text{(IH)}}{=} \frac{\mathrm{d}}{\mathrm{d}t} f^{(i-1)}\left(\Phi_t^{(0)}(a)\right) \\
&\overset{\text{(CR)}}{=} \nabla_x f^{(i-1)}\left(\Phi_t^{(0)}(a)\right) \frac{\mathrm{d}}{\mathrm{d}t}\Phi_t^{(0)}(a) \\
&= \nabla_x f^{(i-1)}\left(\Phi_t^{(0)}(a)\right) \cdot f\left(\Phi_t^{(0)}(a)\right) \\
&= \left[\nabla_x f^{(i-1)} \cdot f\right]\left(\Phi_t^{(0)}(a)\right) \\
&\overset{\text{(BC)}}{=} f^{(i)}\left(\Phi_t^{(0)}(a)\right).
\end{aligned} \tag{C.98}$$

□

C.16 Supplement VI: Proof of Lemma C.5.2

Proof. Again, w.l.o.g. $d = 1$. Recall that, by eq. (C.13), r is implied by the values of $m^{-,(0)}$ and $m^{-,(1)}$. By insertion of

$$\begin{aligned} &m^{-,(i)}((n+1)h) \\ &\quad = \sum_{k=i}^{q} \frac{h^{k-i}}{(k-i)!} m^{(k)}(nh) + K\theta \left| m^{(q)}(nh) \right| h^{q+1-i} \end{aligned} \tag{C.99}$$

(due to eqs. (C.8) and (C.14)) into the definition eq. (C.13) of $r((n+1)h)$, we obtain the following equality which we then bound by repeated application of the triangle inequality:

$$\begin{aligned} |r((n+1)h)| &= \left| f\left(\sum_{k=0}^{q} \frac{h^k}{k!} m^{(k)}(nh) + K\theta \left| m^{(q)}(nh) \right| h^{q+1} \right) \right. \\ &\qquad \left. - \left(\sum_{k=1}^{q} \frac{h^{k-1}}{(k-1)!} m^{(k)}(nh) + K\theta \left| m^{(q)}(nh) \right| h^{q} \right) \right| \\ &\leq \left| f\left(\sum_{k=0}^{q} \frac{h^k}{k!} m^{(k)}(nh) + K\theta \left| m^{(q)}(nh) \right| h^{q+1} \right) \right. \\ &\qquad \left. - \left(\sum_{k=1}^{q} \frac{h^{k-1}}{(k-1)!} m^{(k)}(nh) \right) \right| \; + \; K\theta \left| m^{(q)}(nh) \right| h^{q} \\ &\overset{\text{eq. (C.25)}}{\leq} I_1(h) + I_2(h) + I_3(h) \\ &\qquad + \sum_{k=1}^{q} \frac{h^{k-1}}{(k-1)!} \delta^{(k)}(nh) + K\theta \left| m^{(q)}(nh) \right| h^{q}, \end{aligned} \tag{C.100}$$

where I_1, I_2, and I_3 are defined and bounded as follows, using Assumption C.1 and Lemma C.3.1:

$$\begin{aligned} I_1(h) &:= \left| f\left(\sum_{k=0}^{q} \frac{h^k}{k!} m^{(k)}(nh) + K\theta \left| m^{(q)}(nh) \right| h^{q+1} \right) \right. \\ &\qquad \left. - f\left(\sum_{k=0}^{q} \frac{h^k}{k!} \Phi_0^{(k)}\left(m^{(0)}(nh) \right) \right) \right| \\ &\leq L \sum_{k=0}^{q} \frac{h^k}{k!} \delta^{(k)}(nh) + LK\theta \left| m^{(q)}(nh) \right| h^{q+1}, \end{aligned} \tag{C.101}$$

$$
\begin{aligned}
I_2(h) &\coloneqq \left| f\left(\sum_{k=0}^{q} \frac{h^k}{k!}\Phi_0^{(k)}\left(m^{(0)}(nh)\right)\right) - f\left(\Phi_h^{(0)}\left(m^{(0)}(nh)\right)\right)\right| \\
&\leq L\left|\sum_{k=0}^{q} \frac{h^k}{k!}\Phi_0^{(k)}\left(m^{(0)}(nh)\right) - \Phi_h^{(0)}\left(m^{(0)}(nh)\right)\right| \\
&\overset{\text{eq. (C.16)}}{\leq} Kh^{q+1}, \qquad \text{(C.102)}
\end{aligned}
$$

and

$$
\begin{aligned}
I_3(h) &\coloneqq \left|\Phi_h^{(1)}\left(m^{(0)}(nh)\right) - \sum_{k=1}^{q} \frac{h^{k-1}}{(k-1)!}\Phi_0^{(k)}\left(m^{(0)}(nh)\right)\right| \\
&\overset{\text{eq. (C.16)}}{\leq} Kh^q. \qquad \text{(C.103)}
\end{aligned}
$$

Inserting eq. (C.101), eq. (C.102), and (C.103) into eq. (C.100) (and recalling $\delta^{(0)} = 0$) yields eq. (C.31). □

C.17 Supplement VII: Proof of Lemma C.7.1

Proof. Let $\tilde{u}_0 = u^*$ and $\tilde{u}_n = T_n(\tilde{u}_{n-1})$, for $n \in \mathbb{N}$. Then,

$$
d(u^*, x_n) \leq \underbrace{d(u^*, u_n)}_{\to 0} + \underbrace{d(u_n, \tilde{u}_n)}_{=: a_n} + \underbrace{d(\tilde{u}_n, x_n)}_{\to 0}, \qquad \text{(C.104)}
$$

where the last summand goes to zero by

$$
\begin{aligned}
d(\tilde{u}_n, x_n) &= d\left((T_n \circ \cdots \circ T_1)(u^*), (T_n \circ \cdots \circ T_1)(x_0)\right) \\
&\leq \bar{L}^n d(u^*, x_0) \;\to\; 0, \qquad \text{as } n \to \infty.
\end{aligned}
$$

Hence, it remains to show that $\lim_{n\to\infty} a_n = 0$. The $\bar{L}$-Lipschitz continuity of T_n and the triangle inequality yield that

$$
\begin{aligned}
a_n &= d(T_n(u_n), T_n(\tilde{u}_{n-1})) \\
&\leq \bar{L}\left[d(u_n, u_{n-1}) + d(u_{n-1}, \tilde{u}_{n-1})\right] \\
&= \bar{L}a_{n-1} + b_{n-1}, \qquad \text{(C.105)}
\end{aligned}
$$

where $b_n \coloneqq \bar{L}d(u_{n+1}, u_n) \to 0$. Now, for all $m \in \mathbb{N}$, let $a_0^{(m)} \coloneqq a_0$ and $a_n^{(m)} \coloneqq \bar{L}a_{n-1}^{(m)} + b_m$. By BFT, $\lim_{n\to\infty} a_n^{(m)} = b_m/(1-\bar{L})$. Since, for all $m \in \mathbb{N}$, $a_n \leq a_n^{(m)}$ for sufficiently large

n, it follows that

$$0 \leq \limsup_{n\to\infty} a_n \leq \lim_{n\to\infty} a_n^{(m)} = \frac{b_m}{1-\bar{L}}, \qquad \forall m \in \mathbb{N}. \tag{C.106}$$

Since the convergent sequence u_n is in particular a Cauchy sequence, $\lim_{m\to\infty} b_m = 0$ and, hence, $0 \leq \lim_{n\to\infty} a_n = \limsup_{n\to\infty} a_n \leq 0$. Hence, $\lim_{n\to\infty} a_n = 0$. □

C.18 Supplement VIII: Proof of Proposition C.7.2

Proof. Again, w.l.o.g. $d = 1$. We prove the claims in the following order: eq. (C.39), eq. (C.45), eq. (C.40), eq. (C.46), eq. (C.41), eq. (C.43), eq. (C.44), eq. (C.42), eq. (C.49), eq. (C.48), eq. (C.47). The sharpness of these bounds is shown, directly after they are proved. As a start, for eq. (C.39), we show that $P_{11}^{-,\infty}$ is indeed the unique fixed point of the recursion for $\{P_{11}^{-}(nh)\}_n$ by checking that, if $P_{11}^{-}(nh) = \frac{1}{2}\left(\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}\right)$, then also $P_{11}^{-}((n+1)h) = \frac{1}{2}\left(\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}\right)$:

$$\begin{aligned} P_{11}((nh)) &\overset{\text{eq. (C.15)}}{=} P_{11}^{-}(nh)\left(1 - \frac{P_{11}^{-}(nh)}{P_{11}^{-}(nh)+R}\right) = P_{11}^{-}(nh)\left(\frac{R}{P_{11}^{-}(nh)+R}\right) \qquad \text{(C.107)} \\ &= \left[\frac{\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}}{2}\right] \cdot \left[\frac{R}{\frac{\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}}{2} + R}\right] \\ &= \frac{\left(\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}\right) R}{\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2} + 2R}. \end{aligned}$$

$$\begin{aligned} P_{11}((nh)) &\overset{\text{eq. (C.15)}}{=} P_{11}^{-}(nh)\left(1 - \frac{P_{11}^{-}(nh)}{P_{11}^{-}(nh)+R}\right) \\ &= \frac{\left(\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}\right) R}{\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2} + 2R}, \quad \text{and} \qquad \text{(C.108)} \\ P_{11}^{-}((n+1)h) &= P_{11}(nh) + \sigma^2 h \\ &\overset{\text{eq. (C.108)}}{=} \frac{1}{2}\left(\sigma^2 h + \sqrt{4\sigma^2 Rh + \sigma^4 h^2}\right) \\ &= P_{11}^{-}(nh). \qquad \text{(C.109)} \end{aligned}$$

After combining eq. (C.108) and eq. (C.109), the recursion for P_{11}^- is given by

$$P_{11}^-((n+1)h) = \underbrace{\left(\frac{R}{P_{11}^-(nh)+R}\right)}_{=:\alpha(nh)} P_{11}^-(nh) + \sigma^2 h \tag{C.110}$$

$$=: \tilde{T}\left(P_{11}^-(nh)\right). \tag{C.111}$$

Since R and $P_{11}^-(nh)$ are positive variances, we know that $\inf_{n\in[T/h+1]} P_{11}^-(nh) \geq \sigma^2 h$, and hence $\max_{n\in[T/h+1]} \alpha(nh) \leq R/(\sigma^2 h + R) < 1$. Hence, $\tilde{T}$ is a contraction. By BFT, $P_{11}^{-,\infty}$ is the unique (attractive) fixed point of $\tilde{T}$, and the sequence $\{|P_{11}^-(nh) - P_{11}^{-,\infty}|\}_n$ is strictly decreasing. Since, by eq. (C.15), eq. (C.6) with $\theta = 0$ and Assumption C.2,

$$P_{11}^-(h) = P_{11}(0) + \sigma^2 h \leq Kh, \tag{C.112}$$

we can, using the reverse triangle inequality and the (by BFT) strictly decreasing sequence $\{|P_{11}^-(nh) - P_{11}^{-,\infty}|\}_n$, derive eq. (C.45):

$$\left|P_{11}^-(nh)\right| \leq \underbrace{\left|P_{11}^-(nh) - P_{11}^{-,\infty}\right|}_{\leq \left|P_{11}^-(h) - P_{11}^{-,\infty}\right|} + \left|P_{11}^{-,\infty}\right| \tag{C.113}$$

$$\leq \underbrace{P_{11}^-(h)}_{\leq Kh} + \underbrace{2P_{11}^{-,\infty}}_{\leq Kh^{1\wedge\frac{p+1}{2}}, \text{ by } eq.\ (C.39)} \tag{C.114}$$

$$\leq Kh^{1\wedge\frac{p+1}{2}}, \tag{C.115}$$

which is sharp because it is estimated against the maximum of the initial P_{11}^- and the steady state that can both be attained. Recall that, by eq. (C.108), $P_{11}(nh)$ depends continuously on $P_{11}^-(nh)$, and, hence, inserting eq. (C.39) into eq. (C.108) yields eq. (C.40)—the necessary computation was already performed in eq. (C.108). Since $P_{11}(nh)$ monotonically increases in $P_{11}^-(nh)$ (because the derivative of $P_{11}(nh)$ with respect to $P_{11}^-(nh)$ is non-negative for all $P_{11}^-(nh)$ due to $R \geq 0$; see eq. (C.108)), we obtain

eq. (C.46):

$$P_{11}(nh) \overset{\text{eq. (C.108)}}{\leq} \frac{\left(\max_n P_{11}^-(nh)\right) R}{\max_n P_{11}^-(nh) + R} \tag{C.116}$$

$$\overset{R \sim h^p}{\leq} \frac{K h^{1 \wedge \frac{p+1}{2}} K h^p}{K h^{1 \wedge \frac{p+1}{2}} + K h^p} \tag{C.117}$$

$$\leq \frac{K h^{(p+1) \wedge \frac{3p+1}{2}}}{K h^{1 \wedge p}} \tag{C.118}$$

$$\leq \begin{cases} K h^{\frac{p+1}{2}}, & \text{if } p \leq 1, \\ K h^p, & \text{if } p \geq 1, \end{cases} \tag{C.119}$$

$$\leq K h^{p \vee \frac{p+1}{2}}, \tag{C.120}$$

which is sharp because the steady state eq. (C.45) has these rates. For eq. (C.41), we again first construct the following recursion (from eq. (C.10), eq. (C.15) and eq. (C.6) with $\theta = 0$)

$$P_{01}^-((n+1)h) = \underbrace{\frac{R}{P_{11}^-(nh) + R}}_{=\alpha(nh)} P_{01}^-(nh) + \underbrace{\left(P_{11}(nh) + \frac{\sigma^2 h}{2}\right) h}_{=:g(nh)} \tag{C.121}$$

$$= T_n\left(P_{01}^-(nh)\right), \tag{C.122}$$

where the $\alpha(nh)$-Lipschitz continuous contractions T_n satisfy the prerequisites of Lemma C.7.1, since $\sup_n \alpha(nh) \leq R/(\sigma^2 h + R) < 1$ (due to $\inf_n P_{11}^-(nh) \geq \sigma^2 h$) and the sequence of fixed points $(1 - \alpha(nh))^{-1} g(nh)$ of T_n (defined by BFT) converges. Both $\alpha(nh)$ and $g(nh)$ depend continuously on $P_{11}^-(nh)$. Hence, insertion of the limits eqs. (C.39) and (C.40) yield

$$\lim_{n \to \infty} (1 - \alpha(nh))^{-1} = \frac{\sigma^2 h + \sqrt{4\sigma^2 R h + \sigma^4 h^2} + 2R}{\sigma^2 h + \sqrt{4\sigma^2 R h + \sigma^4 h^2}}, \tag{C.123}$$

and

$$\lim_{n\to\infty} g(nh) \tag{C.124}$$
$$= \frac{(\sigma^4h^2 + (2R+\sigma^2h)\sqrt{4\sigma^2Rh+\sigma^4h^2} + 4R\sigma^2h)}{2(\sigma^2h + \sqrt{4\sigma^2Rh+\sigma^4h^2} + 2R)}h.$$

Now, application of Lemma C.7.1 implies convergence of the recursion eq. (C.122) to the product of these two limits eqs. (C.123) and (C.124), i.e. eq. (C.41):

$$\lim_{n\to\infty} P_{01}^{-}(nh) = \lim_{n\to\infty} (1-\alpha(nh))^{-1} \times \lim_{n\to\infty} g(nh)$$
$$= \frac{\sigma^4h^2 + (2R+\sigma^2h)\sqrt{4\sigma^2Rh+\sigma^4h^2} + 4R\sigma^2h}{2(\sigma^2h + \sqrt{4\sigma^2Rh+\sigma^4h^2})}h.$$

For eqs. (C.43) and (C.44), we can simply insert eqs. (C.39) and (C.41) for $P_{01}^{-}(nh)$ and $P_{11}^{-}(nh)$ respectively into their definition eq. (C.11):

$$\beta^{\infty,(0)} \overset{\text{eq. (C.11)}}{=} \frac{P_{01}^{-,\infty}}{P_{11}^{-,\infty}+R} \tag{C.125}$$
$$\overset{\text{eqs. (C.39) and (C.41)}}{=} \frac{\sqrt{4R\sigma^2h+\sigma^4h^2}}{\sigma^2h+\sqrt{4R\sigma^2h+\sigma^4h^2}}h, \tag{C.126}$$

and

$$\beta^{\infty,(1)} \overset{\text{eqs. (C.11) and (C.39)}}{=} \frac{\sigma^2h+\sqrt{4\sigma^2Rh+\sigma^4h^2}}{\sigma^2h+\sqrt{4\sigma^2Rh+\sigma^4h^2}+2R}. \tag{C.127}$$

These steady states eqs. (C.43) and (C.44) are again unique and attractive because $\beta^{(0)}(nh)$ and $\beta^{(1)}(nh)$ depend continuously on $P_{11}^{-}(nh)$ and $P_{01}^{-}(nh)$. Next, recall that

$$P_{01}(nh) \overset{\text{eq. (C.15)}}{=} \left(1 - \frac{P_{11}^{-}(nh)}{P_{11}^{-}(nh)+R}\right) P_{01}^{-}(nh) \tag{C.128}$$
$$= R\frac{P_{01}^{-}(nh)}{P_{11}^{-}(nh)+R} \overset{\text{eq. (C.11)}}{=} R\beta^{(0)}(nh), \tag{C.129}$$

which, since $P_{01}(nh)$ depends continuously on $\beta^{(0)}(nh)$, implies the unique (attractive) fixed point $P_{01}^{\infty}(nh) = R\beta^{\infty,(0)}$, which yields eq. (C.42). Now, exploiting eq. (C.11) and

$\inf_n P^-_{11}(nh) \geq \sigma^2 h$ yields eq. (C.49):

$$\left|1 - \beta^{(1)}(nh)\right| = \frac{R}{P^-_{11}(nh) + R} \tag{C.130}$$

$$\leq \frac{R}{\sigma^2 h + R} \tag{C.131}$$

$$\overset{R \sim h^p}{=} \frac{K h^p}{K h + K h^p} \tag{C.132}$$

$$\leq K h^{(p-1)\vee 0}, \tag{C.133}$$

which is sharp because $\inf_n P^-_{11}(nh) \geq Kh$ is sharp (due to eqs. (C.6) and (C.10)). And since, for $\beta^{(0)}$, maximizing over both $P^-_{01}(nh)$ and $P^-_{11}(nh)$ at the same time does not yield a sharp bound (while above in eqs. (C.120) and (C.130) the maximization over just one quantity does), we prove eq. (C.48) by inductively showing that

$$\left|\beta^{(0)}(nh)\right| \leq \hat{\beta} h, \qquad \forall n \in \mathbb{N}, \tag{C.134}$$

$$\text{with} \qquad \hat{\beta} \coloneqq \left(\frac{2K_0}{\sigma^2} + \frac{1}{2}\right) \vee 1 > 0, \tag{C.135}$$

where $K_0 > 0$ is the constant from Assumption C.2. The constant $\hat{\beta}$ is independent of n and a possible choice for K in eq. (C.48). The base case ($n = 1$) follows from

$$\left|\beta^{(0)}(h)\right| = \frac{\left|P^-_{01}(h)\right|}{P^-_{11}(h) + R} \tag{C.136}$$

$$\overset{\text{eq. (C.10)}}{\leq} \frac{|P_{01}(0)| + h P_{11}(0) + \frac{\sigma^2}{2} h^2}{\sigma^2 h} \tag{C.137}$$

$$\overset{\text{Ass. C.2}}{\leq} \left(\frac{2K_0}{\sigma^2} + \frac{1}{2}\right) h \tag{C.138}$$

$$\leq \hat{\beta} h. \tag{C.139}$$

In the following inductive step ($n-1 \to n$) we, to avoid notational clutter, simply denote $P^-((n-1)h)_{ij}$ by P^-_{ij} which leaves us—by eq. (C.11), eq. (C.10) and eq. (C.15)—with the following term to bound:

$$\left|\beta^{(0)}(nh)\right| = \frac{\left|P^-_{01}(nh)\right|}{P^-_{11}(nh) + R} \tag{C.140}$$

$$\leq \frac{\left|P^-_{01}\right| \alpha(nh) + h P^-_{11} \alpha(nh) + \frac{\sigma^2}{2} h^2}{P^-_{11} \alpha(nh) + \sigma^2 h + R}, \tag{C.141}$$

with $\alpha(nh) = \left(1 - \frac{P_{11}^-}{P_{11}^- + R}\right) = \frac{R}{P_{11}^- + R}$. Application of the inductive hypothesis (i.e. $P_{01}^- \leq \hat{\beta}(P_{11}^- + R)$) yields, after some rearrangements, that

$$\begin{aligned}
\left|\beta^{(0)}(nh)\right| &\leq \frac{\hat{\beta}\left(P_{11}^- + R\right) h\alpha(nh) + hP_{11}^-\alpha(nh) + \frac{\sigma^2}{2}h^2}{P_{11}^-\alpha(nh) + \sigma^2 h + R} \\
&= \frac{2\hat{\beta}P_{11}^- R + \sigma^2 h\left(P_{11}^- + R\right) + 2P_{11}^- R + 2\hat{\beta}R^2}{2\left(P_{11}^- R + \sigma^2 h\left(P_{11}^- + R\right) + P_{11}^- R + R^2\right)} h \\
&= \frac{2(\hat{\beta}+1)\Lambda_1 + \Lambda_2 + 2\hat{\beta}\Lambda_3}{4\Lambda_1 + 2\Lambda_2 + 2\Lambda_3} h, \qquad \text{(C.142)}
\end{aligned}$$

with $\Lambda_1 \coloneqq 2P_{11}^- R$, $\Lambda_2 \coloneqq \sigma^2 h\left(P_{11}^- + R\right)$, and $\Lambda_3 \coloneqq R^2$. Now, application of $\hat{\beta} \geq 1$ yields $|\beta^{(0)}(nh)| \leq \hat{\beta}h$, which completes the inductive proof of eq. (C.134). This implies eq. (C.48), which is sharp because it is the order of $\beta^{(0)}$ in the steady state eq. (C.43), for all $p \in [0, \infty]$. Now, insertion of eq. (C.48) into eq. (C.128) immediately yields eq. (C.47), which—by eq. (C.128)—inherits the sharpness of eq. (C.48). □

C.19 Supplement IX: Proof of Lemma C.7.4

Proof. For all $n \in [T/h + 1]$, we can estimate

$$\begin{aligned}
\delta^{(1)}(nh) &= \left\| m^{(1)}(nh) - f\left(m^{(0)}(nh)\right) \right\| && \text{(C.143)} \\
&= \left\| \Psi_h{}^{(1)}(\boldsymbol{m}((n-1)h) - f\left(m^{(0)}(nh)\right) \right\| && \text{(C.144)} \\
&\leq \underbrace{\left\| \Psi_h{}^{(1)}(\boldsymbol{m}((n-1)h) - f\left(m^{-,(0)}(nh)\right) \right\|}_{=: J_1(h)} \\
&\quad + \underbrace{\left\| f\left(m^{-,(0)}(nh)\right) - f\left(m^{(0)}(nh)\right) \right\|}_{\coloneqq J_2(h)}, && \text{(C.145)}
\end{aligned}$$

bound J_1, using the definition eq. (C.14) of $\Psi_h^{(1)}(\boldsymbol{m}((n-1)h)$ as well as the definition eq. (C.13) of $r(nh)$, by

$$J_1(h) = \Big\| m^{-,(1)}(nh) - f\left(m^{-,(0)}(nh)\right) \tag{C.146}$$

$$+ \beta^{(1)}(nh)\left[f\left(m^{-,(0)}(nh)\right) - m^{-,(1)}(nh)\right] \Big\|$$

$$\leq \left\|1 - \beta^{(1)}(nh)\right\| \|r(nh)\| \tag{C.147}$$

$$\overset{\text{eq. (C.49)}}{\leq} Kh^{(p-1)\vee 0}\|r(nh)\| \tag{C.148}$$

and bound J_2, by exploiting L-Lipschitz continuity of f, inserting the definition eq. (C.14) of $\Psi_h^{(0)}(\boldsymbol{m}((n-1)h)$ and applying eq. (C.48) to $\left\|\beta^{(0)}(nh)\right\|$,

$$J_2(h) \leq L\left\|m^{(0)}(nh) - m^{-,(0)}(nh)\right\| \tag{C.149}$$

$$\leq L\left\|\beta^{(0)}(nh)\right\| \|r(nh)\| \tag{C.150}$$

$$\overset{\text{eq. (C.48)}}{\leq} Kh\|r(nh)\|. \tag{C.151}$$

Altogether, after inserting these bounds into eq. (C.145),

$$\delta^{(1)}(nh) \leq \left(Kh^{(p-1)\vee 0} + Kh\right)\|r(nh)\| \tag{C.152}$$

$$\leq Kh^{((p-1)\vee 0)\wedge 1}\|r(nh)\| \tag{C.153}$$

$$\overset{\text{eq. (C.31)}}{\leq} Kh^{(p\vee 1)\wedge 2} \tag{C.154}$$

$$+ \left(Kh^{((p-1)\vee 0)\wedge 1} + Kh^{(p\vee 1)\wedge 2}\right)\delta^{(1)}((n-1)h)$$

$$=: \bar{T}\left(\delta^{(1)}((n-1)h)\right). \tag{C.155}$$

As $p \geq 1$ (by Assumption C.4), BFT is applicable for all sufficiently small $h > 0$ such that $Kh^{((p-1)\vee 0)\wedge 1} + Kh^{(p\vee 1)\wedge 2} < 1$ and so $\bar{T}$ is a contraction with a unique fixed point δ^∞ of order

$$\delta^\infty \leq \frac{Kh^{(p\vee 1)\wedge 2}}{1 - \left(Kh^{((p-1)\vee 0)\wedge 1} + Kh^{(p\vee 1)\wedge 2}\right)} \tag{C.156}$$

$$\leq Kh^{(p\vee 1)\wedge 2}. \tag{C.157}$$

We proceed with showing by induction that, for all $n \in [T/h]$,

$$\delta^{(1)}(nh) \leq \delta^{(1)}(0) \vee 2\delta^{\infty}. \tag{C.158}$$

The base case $n = 0$ is trivial. For the inductive step, we distinguish two cases. If $\delta^{(1)}((n-1)h) \leq \delta^{\infty}$, then $\bar{T}(\delta^{(1)}((n-1)h)) < 2\delta^{\infty}$, since

$$\begin{aligned}
\bar{T}(\delta^{(1)}((n-1)h)) - \delta^{\infty} &\leq \left|\delta^{\infty} - \bar{T}(\delta^{(1)}((n-1)h))\right| && \text{(C.159)}\\
&< \delta^{\infty} - \underbrace{\delta^{(1)}((n-1)h)}_{\geq 0} && \text{(C.160)}\\
&\leq \delta^{\infty}. && \text{(C.161)}
\end{aligned}$$

In this case,

$$\begin{aligned}
\delta^{(1)}(nh) &\overset{\text{eq. (C.155)}}{\leq} \bar{T}\left(\delta^{(1)}((n-1)h)\right) && \text{(C.162)}\\
&< 2\delta^{\infty} && \text{(C.163)}\\
&\leq \delta^{(1)}(0) \vee 2\delta^{\infty}, && \text{(C.164)}
\end{aligned}$$

where the last inequality follows from the inductive hypothesis. In the other case, namely $\delta^{(1)}((n-1)h) > \delta^{\infty}$, it follows that

$$\begin{aligned}
\delta^{(1)}(nh) - \delta^{\infty} &\overset{\text{eq. (C.155)}}{\leq} \bar{T}(\delta^{(1)}((n-1)h)) - \delta^{\infty} && \text{(C.165)}\\
&\leq \left|\bar{T}(\delta^{(1)}((n-1)h)) - \delta^{\infty}\right| && \text{(C.166)}\\
&\leq \left|\delta^{(1)}((n-1)h) - \delta^{\infty}\right| && \text{(C.167)}\\
&= \delta^{(1)}((n-1)h) - \delta^{\infty}, && \text{(C.168)}
\end{aligned}$$

which, after adding δ^{∞} and applying the inductive hypothesis, completes the inductive step. Hence, eq. (C.158) holds. Since this bound is uniform in n, inserting the orders of $\delta^{(1)}(0)$ from Lemma C.6.1 and of δ^{∞} from eq. (C.156) yields eq. (C.50). □

C.20 Supplement X: Proof of Theorem C.8.1

Proof. Again, w.l.o.g. $d = 1$. We first show that the bounds eqs. (C.67) and (C.68) hold and then argue that they are sharp. The recursion for $P_{00}^{-}(nh)$ is given by

$$\begin{aligned} P_{00}^{-}((n+1)h) &\overset{\text{eqs. (C.10),(C.6)}}{=} P_{00}(nh) + 2hP_{01}(nh) \\ &\quad + h^2 P_{11}(nh) + \frac{\sigma^2}{3}h^3 \qquad \text{(C.169)} \\ &= P_{00}^{-}(nh) - \beta^{(0)}(nh)P_{01}^{-}(nh) + \frac{\sigma^2}{3}h^3, \\ &\quad + 2hR\beta^{(0)}(nh) + h^2 R\beta^{(1)}(nh) \qquad \text{(C.170)} \end{aligned}$$

where we used $P_{00}(nh) = P_{00}^{-}(nh) - \beta^{(0)}P_{01}^{-}(nh)$ and $P_{11}(nh) = R\beta^{(1)}(nh)$ (both due to eq. (C.15) and eq. (C.11)), as well as $P_{01}(nh) = R\beta^{(0)}(nh)$ (see eq. (C.128)), for the last equality in eq. (C.170). By $P_{01}^{-}(nh) \leq P_{01}(nh)$ and $|\beta^{(1)}| \leq 1$ (due to eq. (C.11)), application of the triangle inequality to eq. (C.170) yields

$$\begin{aligned} P_{00}^{-}\left((n+1)h\right) &\leq P_{00}^{-}(nh) + \left|\beta^{(0)}(nh)\right| |P_{01}(nh)| \\ &\quad + 2hR\left|\beta^{(0)}(nh)\right| + h^2 R + \frac{\sigma^2}{3}h^3, \qquad \text{(C.171)} \end{aligned}$$

which, by eqs. (C.47) and (C.48), implies

$$P_{00}^{-}((n+1)h) \leq P_{00}^{-}(nh) + Kh^{(p+2)\wedge 3}. \qquad \text{(C.172)}$$

This, by $N = T/h$, implies eq. (C.67). Since $P_{00}(nh) \leq P_{00}^{-}(nh)$, this bound is also valid for P_{00}, i.e. eq. (C.68) holds. The bound eq. (C.67) is sharp, since, e.g. when the covariance matrices are in the steady state, the covariance matrix keeps growing by a rate of $Kh^{(p+2)\wedge 3}$ for all sufficiently small $h > 0$, since the only negative summand in eq. (C.170) is given by

$$\beta^{\infty,(0)}P_{01}^{\infty} = S_1(h) \times S_2(h) \times S_3(h) \ \in \Theta(h^{5\wedge\frac{3p+7}{2}}), \qquad \text{(C.173)}$$

where the factors have, due to $R \equiv Kh^p$, the following orders:

$$\begin{aligned} S_1(h) &= \frac{1}{2}h^2 \ \in \Theta(h^2), \qquad \text{(C.174)} \\ S_2(h) &= \sqrt{(\sigma^2 h)^2 + 4(\sigma^2 h)R}, \ \in \Theta(h^{1\wedge\frac{p+1}{2}}), \qquad \text{(C.175)} \\ S_3(h) &= ((\sigma^2 h) + 2R)\sqrt{(\sigma^2 h)^2 + 4(\sigma^2 h)R} \\ &\quad + (\sigma^2 h)^2 + 4(\sigma^2 h)R \ \in \Theta(h^{2\wedge(p+1)}). \qquad \text{(C.176)} \end{aligned}$$

The orders in eqs. (C.174) to (C.176) imply the order in eq. (C.173). Hence, the sole negative summand $-\beta^{\infty,(0)} P_{01}^{\infty}$ of eq. (C.170) is in $\Theta(h^{5 \wedge \frac{3p+7}{2}})$ and thereby of higher order than the remaining positive summands of eq. (C.170):

$$\underbrace{2hR}_{\in\Theta(h^{p+1})} \underbrace{\beta^{\infty,(0)}(nh)}_{\in\Theta(h)} \in \Theta(h^{p+2}), \tag{C.177}$$

$$\underbrace{h^2 R}_{\in\Theta(h^{p+2})} \underbrace{\beta^{\infty,(1)}(nh)}_{\in\Theta(1),\text{ by eq. (C.44)}} \in \Theta(h^{p+2}), \tag{C.178}$$

$$\frac{\sigma^2}{3} h^3 \in \Theta\left(h^3\right). \tag{C.179}$$

Hence, for all sufficiently small $h > 0$, it still holds in the steady state that $P_{00}^{-}((n+1)h) - P_{00}^{-}(nh) \geq K h^{(p+2) \wedge 3}$, and therefore eq. (C.67) is sharp. The sharpness of eq. (C.67) is inherited by eq. (C.68) since, in the steady state, by eqs. (C.11) and (C.15), $P_{00}(nh) = P_{00}^{-}(nh) - \beta^{(0),\infty} P_{01}^{-,\infty}$ and the subtracted quantity $\beta^{(0),\infty} P_{01}^{-,\infty}$ is—as shown above—only of order $\Theta(h^{5 \wedge \frac{3p+7}{2}})$. □

D Differentiable Likelihoods for Fast Inversion of 'Likelihood-Free' Dynamical Systems (Kersting *et al.*, 2020b)

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.

D.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 Computation (ABC); see Cranmer *et al.* (2020) 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

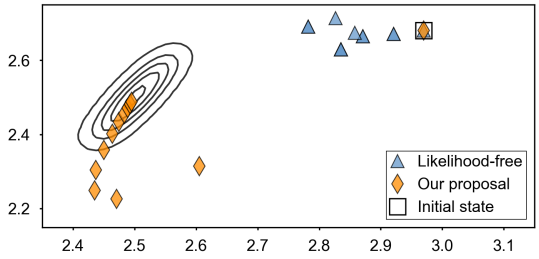


Figure D.1: Inference on the logistic ODE. First twelve sampled parameters of likelihood-free inference and our proposed method. Details in text.

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. (Supplement D.10 provides a concise introduction to Gaussian ODE filtering; Tronarp *et al.* (2019a) offer a more detailed presentation. See Hennig *et al.* (2015) or Oates and 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 (Appendix D.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 (see Figure D.1).

Much thought has been devoted to improving the slow run-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 and 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 D.3.1), as well as an upper bound on its approximation error (see Theorem D.4.1). *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.

D.2 Problem setting

We consider a dynamical system defined by the ODE

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

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 D.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 $i = 1, \ldots, n$.*

The necessity for this assumption will become evident in Appendix D.3.1. 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 D.1 either immediately or after reparametrization. Otherwise, we can still transform a non-conforming ODE into a system that obeys Assumption D.1, as exemplified for the protein signalling transduction pathway in Appendix D.7.2. While this adds an additional layer of imprecision, the experiments appear to be equally good—which suggests a wider applicability of our methods than Assumption D.1.

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. (D.10). Solving eq. (D.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. (D.1) with *unknown* true parameter θ^*. The true trajectory $x = x_{\theta^*}$ is observed under additive, zero-mean Gaussian noise at M discrete times $0 \leq t_1 < \cdots < t_M \leq T$:

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

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

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

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. (D.2). For the sake of readability, we will assume w.l.o.g. that $d = 1$; this restriction is purely

notational as can be seen from the multi-dimensional experiments below. Under these conventions, eq. (D.2) is equivalent to

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

for some $\sigma^2 > 0$, where I_M is the $M \times M$ identity matrix. Heteroscedastic noise can be modelled by replacing $\sigma^2 I_M$ with a diagonal matrix with varying diagonal entries.

D.3 Likelihoods by Gaussian ODE filtering

The prevailing view on the uncertainty in inverse problems only considers the aleatoric uncertainty Σ_i from eq. (D.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, $p(\boldsymbol{x}_\theta \,|\, \theta)$, is commonly treated as a Dirac distribution $\delta(\boldsymbol{x}_\theta - \hat{\boldsymbol{x}}_\theta)$ which yields the *uncertainty-unaware likelihood*

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

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 and Garegnani (2020, 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 Supplement D.10.2 for a short introduction. This amounts—e.g. in the notation of Tronarp *et al.* (2019a)—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 $\{0, h, \ldots, Nh\}$ with step size $h > 0$. This set is, w.l.o.g., assumed to contain the data time points $\{t_1, \ldots, t_M\}$ from eq. (D.2), i.e. we assume the existence of a set of integers $\{l_1, \ldots, l_M\}$ 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. (D.36) in Supplement D.10.)

D.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

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

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

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

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

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

1. The filtering mean $\boldsymbol{m}_\theta$ can be linearized in θ, as specified below in eq. (D.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. (D.26) and (D.27). These estimators are useful to guide samples of θ into regions of high likelihood by the gradient-based sampling and methods defined in Appendix D.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. (D.9). Unless $\boldsymbol{P} \ll \sigma^2 I_M$, this prevents over-confidence, as visualized in Figure D.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 Supplement D.11.

The filtering mean

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

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

where $1_M = [1, \ldots, 1]^\intercal$ 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. (D.10) is provided in Supplement D.11.3.) Here,

$$J \coloneqq KY \ \in \mathbb{R}^{M\times n} \tag{D.11}$$

is an estimator of the Jacobian matrix of the map $\theta \mapsto \boldsymbol{m}_\theta$, as we show in Theorem D.3.1 below. This estimator is equal to the product of the *kernel prefactor* K and the *evalu-*

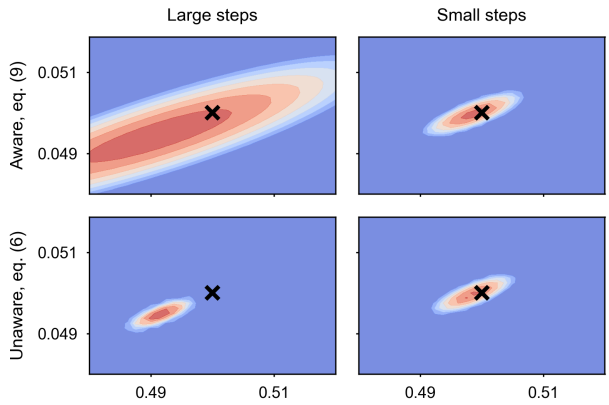


Figure D.2: Uncertainty-(un)aware likelihoods, eqs. (D.6) and (D.9) w.r.t. (θ_1, θ_2) of Lotka-Volterra ODE, eq. (D.30), with fixed $(\theta_3, \theta_4) = (0.05, 0.5)$. θ_1 on x and θ_2 on y-axis. Black cross is true parameter. The unaware likelihood is overconfident for the large step size ($h = 0.2$),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

ation factor Y. The kernel prefactor K is given by

$$K \coloneqq \left[\kappa_1, \ldots, \kappa_M\right]^\intercal \in \mathbb{R}^{M \times N}, \tag{D.12}$$

whose i-th row is

$$\kappa_i \coloneqq \left[\tilde{\kappa}_i^\intercal, 0, \ldots, 0\right]^\intercal \in \mathbb{R}^N, \tag{D.13}$$

which is defined by

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

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

$$k^{\partial}(h : t_i, t_i) \coloneqq \left[k^{\partial}(t_i, h), \ldots, k^{\partial}(t_i, t_i)\right]^\intercal, \text{ and} \tag{D.15}$$

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

Now, recall Assumption D.1. For a given θ, the entries of the evaluation factor $Y \in$

$\mathbb{R}^{N\times n}$ are

$$y_{ij} \coloneqq f_j(m_\theta^-(ih)) - f_j(x_0), \tag{D.17}$$

for all $i = 1, \ldots, N$ and $j = 1, \ldots, n$, 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. (D.1). Hence, Y is freely accessible with every filtering distribution, eq. (D.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 Supplement D.11.3). For this reason, J is not the true Jacobian of $\theta \mapsto \boldsymbol{m}_\theta$ but only an estimator (see Appendix D.3.2).

The filtering covariance

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

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

and are hence independent of θ. (See Supplement D.11.2 for a detailed derivation of eq. (D.18).)

D.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 D.3.1. *Under Assumption D.1, the true Jacobian $D\boldsymbol{m}_\theta \in \mathbb{R}^{M\times n}$ of $\theta \mapsto \boldsymbol{m}_\theta$ has the analytic form*

$$D\boldsymbol{m}_\theta \coloneqq [\nabla_\theta m(t_1), \ldots, \nabla_\theta m(t_M)]^\intercal = J + KS, \tag{D.19}$$

where the sensitivity term *S is defined by*

$$S \coloneqq \left[\Lambda_1^\intercal \theta, \ldots, \Lambda_N^\intercal \theta\right]^\intercal \in \mathbb{R}^{N\times n}. \tag{D.20}$$

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

$$\lambda_{kl}(jh) \coloneqq \frac{\mathrm{d}}{\mathrm{d}x} f_l(m_\theta^-(jh)) \cdot \frac{\partial}{\partial \theta_k} m_\theta^-(jh). \tag{D.21}$$

Proof. See Supplement D.12. □

Thus, KS is the exact approximation error of J.

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

Assumption D.2 is required to bound the global error of the ODE forward solution by Kersting *et al.* (2020a, Thm. 6.7).

Assumption D.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 D.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 D.4.1, 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 D.4.1. *If* $\Theta \subset \mathbb{R}^n$ *is compact and* $R > 0$, *then it holds true, under Assumptions D.1 to D.3, that*

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

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

Proof. See Supplement D.13. □

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.*, 2020a, Thm. 6.7).

D.5 Gradient and Hessian estimators

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

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

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

$$\begin{aligned} E(\boldsymbol{z}) &\coloneqq \frac{1}{2} [\boldsymbol{z} - \boldsymbol{m}_\theta]^\intercal \left[\boldsymbol{P} + \sigma^2 I_M\right]^{-1} [\boldsymbol{z} - \boldsymbol{m}_\theta] && \text{(D.24)} \\ &\overset{eq.\ (D.10)}{=} \frac{1}{2} [\boldsymbol{z} - x_0 \cdot 1_M - J\theta]^\intercal \left[\boldsymbol{P} + \sigma^2 I_M\right]^{-1} \\ &\quad \times [\boldsymbol{z} - x_0 \cdot 1_M - J\theta] \,. && \text{(D.25)} \end{aligned}$$

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

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

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

(See Figure D.3 for a visualization of these estimators.) Supplement D.14 provides versions of these estimators for Bayesian inference of θ.

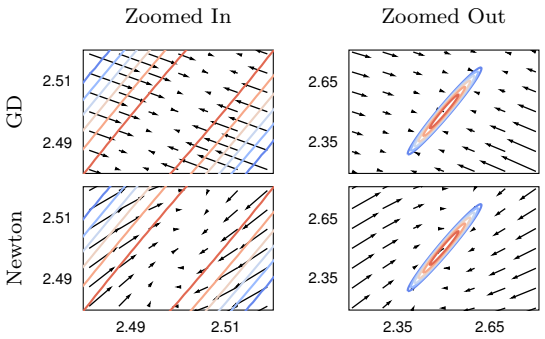


Figure D.3: Directions of gradient descent (GD) and Newton using eqs. (D.26) and (D.27); around mode (left) and globally (right) of the likelihood, based on the logistic ODE. Globally, GD points more directly to the high-probability region. Within this region, however, Newton is better directed to the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

D.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*.

D.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. (D.26) and (D.27)—such as (stochastic) gradient descent (GD), (stochastic) Newton (NWT), Gauss-Newton and natural Gradient descent. This application of the estimators (D.26) and (D.27) unlocks fast computation of single parameter estimates by maximum-likelihood estimation, as we demonstrate in the experiments (see Appendix D.7).

D.6.2 Gradient-based sampling

Likewise, all gradient-based MCMC schemes are now available. Classical gradient-based samplers include Langevin Monte Carlo (LMC) (Roberts and 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 and Calderhead, 2011). A popular choice for the preconditioner is the Hessian (Qi and Minka, 2002). Hence, we can precondition LMC and HMC that use eq. (D.26) as a gradient with the Hessian estimator from eq. (D.27). For LMC, this leads to the proposal distribution

$$\pi(\theta^{i+1} \mid \theta^i) = \theta^i - \rho[\hat{\nabla}^2_\theta E_{\theta^i}(\boldsymbol{z}))]^{-1}\hat{\nabla}_\theta E_{\theta^i}(\boldsymbol{z})) + \xi^i, \tag{D.28}$$

$$\xi^i \sim \mathcal{N}(0, 2\rho[\hat{\nabla}^2_\theta E_{\theta^i}(\boldsymbol{z}))]^{-1}), \tag{D.29}$$

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. In Appendix D.7, we show that the gradient-based versions more aptly explore regions of high likelihood than their gradient-free counterparts.

D.6.3 Algorithm

The generic method that we propose is outlined in Algorithm 3. It includes all above-

Algorithm 3 Gradient-based sampling/optimization

1: Precompute K and $(P + \sigma^2 I_M)^{-1}$ (see eqs. (D.12), (D.6))
2: Initialize $\theta = \theta^0$
3: **repeat**
4: Solve ODE with θ (this generates Y; see eq. (D.17))
5: Compute $J = KY$ (see eq. (D.11))
6: Compute $[\hat{\nabla}_\theta E, \hat{\nabla}^2_\theta E]$ (see eqs. (D.26), (D.27))
7: Update θ with gradient-based sampler/optimizer
8: **until** convergence/mixing

mentioned classical optimization and sampling methods (by a corresponding choice in Line 7). The only difference, compared to all of these existing gradient-based methods, are the additional Lines 5 and 6 where we compute our gradient and Hessian estimators from eqs. (D.26) and (D.27).

D.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: precomputation of K (Line 1 in Algorithm 3) requires the inversion of the M kernel Gram matrices $\{{}^{\partial}K^{\partial}(h : t_i),\ i = 1, \ldots, M\}$, 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 and Särkkä, 2010). Hence, K is in $\mathcal{O}(MN)$ and, as $M \leq N$, in $\mathcal{O}(N^2)$. 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. The Jacobian estimator $J = KY$ (Line 5 in Algorithm 3) is, by eq. (D.11), a matrix product of the precomputed kernel prefactor K and the evaluation factor Y. Y is almost free, as it is by eq. (D.17) only composed of terms that the Gaussian ODE filter computes anyway; see eq. (D.45) in Supplement D.10.2. Given J and $[\boldsymbol{P} + \sigma^2 I_M]^{-1}$, computing the gradient and Hessian estimators (Line 6 in Algorithm 3) is of the same complexity as computing J. Thus, the additional computational cost is in $\mathcal{O}(N^3)$ w.r.t. the number of time steps $N = T/h$ executed once and otherwise 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.

$$ {}^{\partial}K^{\partial}(h : T)^{-1} = \frac{1}{h} \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & 2 & -1 \\ & & -1 & 1 \end{bmatrix} $$

. And then use matrix inversion lemma (Woodbury) to compute $({}^{\partial}K^{\partial}(h:T) + R \cdot I_{N-1})^{-1}$.

D.6.5 Choice of hyperparameters

Recall that the parameters σ and R stem from the data and the accuracy of the ODE model (Kersting *et al.*, 2020a, 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 Supplements D.11.1 and D.11.2. There are two ways to set it: either as a local (Schober *et al.*, 2019, eq. (46)) or as a global (Tronarp *et al.*, 2019a, eq. (41)) maximum-likelihood estimate, which can both be computed from intermediate quantities of the forward solves.

D.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 Appendix D.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.

D.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 D.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. (D.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 Appendix D.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.*, 2019, 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. (D.26) and (D.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 Appendix D.6.5, only matters when both error-sources are of comparable scale.

D.7.2 Results

We evaluate the performance of these methods over the first few iterations (steps), comparing the values of the negative log-likelihood E as well as the relative error in the parameter space, $\|\theta^i - \theta^*\|/\|\theta^*\|$. For optimizers, low values in both metrics indicate success and, in fact, both are important: ODE inverse problems are inherently ill-posed and can have parameters with high likelihood and large inference error that fit the data as well as the true parameter. Finding these parameters would not be a failure of the algorithms, but a success, as they are a mode of the true posterior.

Samplers, on the other hand, try to identify and explore regions of high probability (the typical set); see e.g. Betancourt (2017, Section 2). We opt for plotting the relative error in the parameter space additionally to the negative log-likelihood values to emphasize that, once a sampler creates samples near the typical set, MCMC methods keep exploring suitable values instead of relying on a single estimate with high likelihood. Despite maintaining a low near-constant negative log-likelihood, the error in the parameter space of a sampler may have (some) variation.

The details and results for each benchmark systems are presented next, in ascending order of complexity.

Lotka–Volterra

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

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

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. (D.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 $\theta^0 = [0.8, 0.2, 0.05, 1.1]$, and the forward solutions for all likelihood evaluations were computed with step size $h = 0.05$. In order to turn this θ^0 into a useful initialization for the Markov chains, we accepted the first 45 states generated by PHMC and PLMC—the same would be counterproductive for RWM since a proposed sample may be further away from the region of nonzero probability. The results for optimization and sampling are depicted in Figure D.4. In the case of optimizers, NWT outperforms

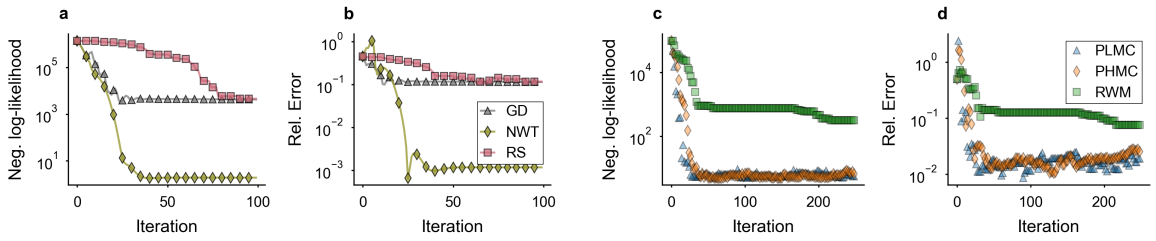


Figure D.4: **Results for optimization (a, b) and sampling (c, d) on Lotka–Volterra**. Comparison of negative log-likelihood $E(\boldsymbol{z}) = E_{\theta^i}(\boldsymbol{z})$ (a and c, resp.) and relative error $\|\theta^i - \theta^*\|/\|\theta^*\|$ (b and d, resp.). 100 iterations of optimization (only every fifth iteration has a marker) and 250 Metropolis-Hastings samples (only every other sample has a marker).

GD which, in turn, outperforms RS. After roughly 25 samples, NWT generates iterations with relative error of less than 10^{-3}. While PLMC and PHMC quickly reach and explore regions of high probability, RWM does not find likelihood values within the first 250 samples. Thus, the gradient and Hessian estimators indeed appear to work well on LV.

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:

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

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

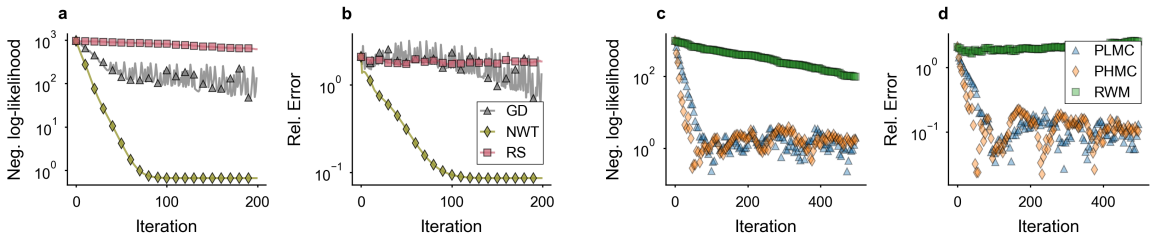


Figure D.5: **Results for optimization (a, b) and sampling (c, d) on PST**. Comparison of negative log-likelihood $E(\boldsymbol{z}) = E_{\theta^i}(\boldsymbol{z})$ (a and c, resp.) and relative error $\|\theta^i - \theta^*\|/\|\theta^*\|$ (b and d, resp.). 200 iterations of optimization (only every tenth iteration has a marker) and 500 Metropolis-Hastings samples (only every fourth sample has a marker).

new linearized ODE

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

which is an approximation of the original ODE, since eq. (D.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 $[0, 100]$. We set the true parameter to $\theta^* = [0.07, 0.6, 0.05, 0.3, 0.017]$. To generate the data by eq. (D.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 $\theta^0 = [0.24, 1.8, 0.15, 0.9, 0.05]$, and the forward solutions for all likelihood evaluations were computed with step size $h = 0.05$. We use the same burn-in procedure as on the Lotka–Volterra example, accepting the first 100 samples. The results for optimization and sampling are depicted in Figure D.5.

Again, the new methods outperform the conventional ones in both optimization and sampling. For optimization, NWT converges particularly fast. The final estimate that is returned by NWT is, rounded to two digits, $\theta^{200} = (0.07, 0.60, 0.05, 0.30, 0.02)$, and hence recovers four out of five parameters exactly. For sampling, both gradient-based samplers (after a fairly steep initial improvement) steadily stay in regions of high likelihood, while RWM only increases the likelihood in a much slower pace. Hence, the gradient and Hessian estimators are beneficial on PST as well—although we had to linearize the ODE first.

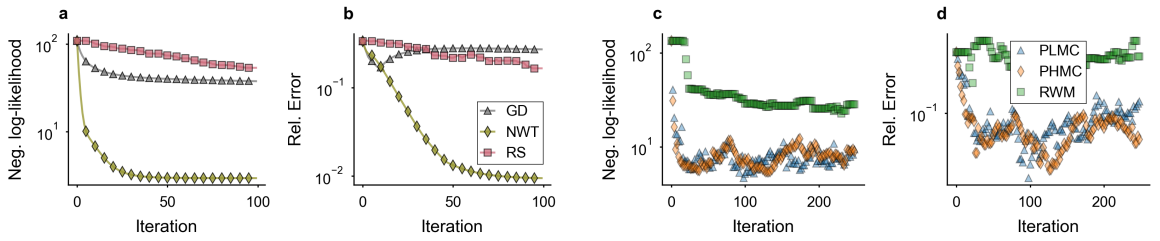


Figure D.6: **Results for optimization (a, b) and sampling (c, d) on GUiY**. Comparison of negative log-likelihood $E(\boldsymbol{z}) = E_{\theta^i}(\boldsymbol{z})$ (a and c, resp.) and relative error $\|\theta^i - \theta^*\|/\|\theta^*\|$ (b and d, resp.). 100 iterations of optimization (only every fifth iteration has a marker) and 250 Metropolis-Hastings samples (only every other sample has a marker).

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 D.1; see Supplement D.15 for a complete mathematical definition and parameter choices. The results for optimization and sampling are depicted in Figure D.6.

GD outperforms RS, and NWT converges even much faster than GD. Remarkably, NWT already finds parameters that are exact up to two relative digits after only five 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 has difficulties sampling from this high dimensional problem in an efficient manner. Thus, this benchmark system also reaffirms the utility of the gradient and Hessian estimators.

D.7.3 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 regions of elevated likelihood, while the conventional random-walk Metropolis methods hardly ever reached regions of nonzero probability and wasted computational budget on less likely parameters. 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.

D.8 Related and future work

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

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.* (2020a, 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. (D.9)) into a pseudo-marginal MCMC framework; see Conrad *et al.* (2017), Teymur *et al.* (2018), Lie *et al.* (2019), and Abdulle and Garegnani (2020). Notably, Matsuda and 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 and Welling (2014) incorporated this approach into an ABC framework, and Perdikaris and 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. (D.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 and 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 D.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. (D.11) can be interpreted as 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 Gaussian ODE filtering with once-integrated Brownian motion prior. Future work could not only examine different priors (Kersting *et al.*, 2020a, 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.* (2019a). 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 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. (D.20)) could also be estimated with low overhead, it is in light of eq. (D.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 D.1—as the results from Appendix D.7.2 suggest. Finally, this work should be, by the methods of lines (Schiesser and Griffiths, 2009), extendable to PDEs and, by John *et al.* (2019), to boundary value problems.

D.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

We thank the anonymous reviewers for their careful, constructive comments. We thank Filip Tronarp and Katharina Ott for helpful feedback.

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

Supplementary Material for Kersting *et al.* (2020b)

D.10 Supplement I: Short introduction to Gaussian ODE filtering

D.10.1 Gaussian filtering for generic time series

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

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

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

Usually, the state x_i is assumed to be the discretization of a continuous 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):

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

where F and L are the drift and diffusion matrix, respectively. The corresponding dynamic model (eq. (D.36)) can be easily constructed by discretization of the LTI SDE (eq. (D.38)), as described in Särkkä and 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

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

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

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

for matrices H, R, the filtering distributions $p(x_i \mid y_{1:i})$, $i = 1, \ldots, N$, 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.

D.10.2 Gaussian ODE filtering

A Gaussian ODE filter is simply a Gaussian filter, as defined in Appendix D.10.1, with a specific kind of probabilistic state space model eqs. (D.36) and (D.37), to infer the solution $x : [0, T] \to \mathbb{R}^d$ of the ODE eq. (D.1), at the discrete time grid $\{0 \cdot h, \ldots, N \cdot h\}$ with step size $h > 0$. The dynamic model is—as usual, recall eqs. (D.38) and (D.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

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

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

$$p(x_i \mid y_{1:i-1}) = \mathcal{N}(m_i^-, P_i^-) \tag{D.42}$$

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

$$f(\hat{x}(t)) \approx f(x(t)) = \dot{x}(t) \tag{D.43}$$

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

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

with data

$$y_i \coloneqq f(m_i^-) \approx \dot{x}(ih). \tag{D.45}$$

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.* (2020a) or Schober *et al.* (2019). An extension to more Bayesian filters—such as particle filters—is provided by Tronarp *et al.* (2019a).

D.11 Supplement II: Equivalent form of filtering distribution by GP regression

Recall from Appendix D.10 that any Gaussian filter computes a sequence of filtering distributions

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

from a GP prior on x eq. (D.38) and a linear Gaussian measurement model (eq. (D.40)) with derivative data (eq. (D.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:

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

with

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

D.11.1 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.

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

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

integrated Brownian motion (IBM) kernel

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

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

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

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

D.11.2 GP form of filtering distribution

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

$$\begin{aligned} m_i =& \mu + k^{\partial}(h:ih, ih)^{\intercal} \left[{}^{\partial}K^{\partial}(h:ih) + R \cdot I_i \right]^{-1} \\ & \times \left[y_1 - \dot{\mu}(h), \ldots, y_i - \dot{\mu}(ih) \right]^{\intercal}, \end{aligned} \tag{D.52}$$

$$\begin{aligned} P_i =& \begin{bmatrix} k(h,h) & \ldots & k(ih,ih) \\ \vdots & \ddots & \vdots \\ k(ih,h) & \ldots & k(ih,ih) \end{bmatrix} - k^{\partial}(h:ih,ih)^{\intercal} \\ & \times \left[{}^{\partial}K^{\partial}(h:ih) + R \cdot I_l \right]^{-1} k^{\partial}(h:ih,ih), \end{aligned} \tag{D.53}$$

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

D.11.3 Derivation of eq. (D.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. (D.10), we first observe that, by eq. (D.52), $m(ih)$ is linear in the data residuals:

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

Now recall that, in ODE filtering, the prior mean in eq. (D.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 D.1 to eq. (D.54) yields

$$m_\theta(ih) = x_0 + J_{ih}\theta, \quad \text{with} \tag{D.55}$$

$$\begin{aligned} J_{ih} &:= \beta_{ih} \begin{bmatrix} f_1(m_\theta^-(h)) - f_1(x_0) & \dots & f_n(m_\theta^-(h)) - f_n(x_0) \\ \vdots & \ddots & \vdots \\ f_1(m_\theta^-(ih)) - f_1(x_0) & \dots & f_n(m_\theta^-(ih)) - f_n(x_0) \end{bmatrix} \\ &= \beta_{ih} Y_{1:i} \quad , \end{aligned} \tag{D.56}$$

where $Y_{1:i}$ denotes the first i rows of Y; see eq. (D.17). We omit the dependence of J_{ih} on θ to obtain a linear form. Recall from Appendix D.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. (D.55) implies

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

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. (D.14). We conclude the derivation of eq. (D.10) by observing that the i-th entry of eq. (D.10) reads eq. (D.57) for all $i = 1, \dots, M$.

D.12 Supplement III: Proof of Theorem D.3.1

Proof. We start by computing the rows of

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

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

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

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

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

is defined by

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

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

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

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

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

Again, we compute the rows of the final missing Jacobian

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

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

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

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

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

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

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

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

$$Dv(\theta) = \Big[\Lambda_1^\intercal \theta, \ldots, \Lambda_{l_i}^\intercal \theta \Big]^\intercal + Y[:l_i, :], \tag{D.69}$$

where

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

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

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

By eq. (D.13), it follows that

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

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

This implies via eq. (D.70) that

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

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

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

□

D.13 Supplement IV: Proof of Theorem D.4.1

We first show some preliminary technical lemmas in Appendix D.13.1 which are needed to prove bounds on $\|K\|$ and $\|S\|$ in Appendix D.13.2 and Appendix D.13.3, respectively. Having proved these bounds, the core proof of Theorem D.4.1 simply consists of combining them by Theorem D.3.1, as executed in Appendix D.13.4.

D.13.1 Preliminary lemmas

The following lemma will be needed in Appendix D.13.2 to bound $\|K\|$.

Lemma D.13.1. *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*

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

for the nuclear norm

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

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

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

are the eigenvalues. Therefore

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

In eq. (D.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 $\lambda_1(Q) \leq \cdots \leq \lambda_{m\wedge n}(Q)$ counted by algebraic multiplicity. This fact is an immediate consequence of Theorem 8.1.5. in Golub and Van Loan (1996). □

The next lemma will be necessary to prove a bound on $\|S\|$ in Appendix D.13.3.

Lemma D.13.2. *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*

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

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

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

where Λ^{o} denotes the interior of Λ.

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

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

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

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

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

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

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

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

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

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

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

which implies

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

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

D.13.2 Bound on $\|K\|$

Lemma D.13.3. *Under Assumption D.3 and for all $R > 0$, it holds true that*

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

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

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

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

for all $i = 1, \ldots, M$. Second, Lemma D.13.1 implies that

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

Now, by eq. (D.13), we observe

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

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

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

□

D.13.3 Bound on $\|S\|$

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

Lemma D.13.4. *If $\Theta \subset \mathbb{R}^n$ is compact, then it holds true, under Assumptions D.1 and D.2, that*

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

Proof. First, recall that the convergence rates of $\mathcal{O}(h)$ provided by Theorem 6.7 in Kersting *et al.* (2020a) 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 D.1. Hence, Theorem 6.7 from Kersting *et al.* (2020a) yields under Assumption D.2 that

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

Moreover, Theorem 8.49 in Kelley and Peterson (2010) is applicable under Assumption D.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. (D.10)—the filtering mean $m_\theta(t)$ has the same regularity too. Hence, application of Lemma D.13.2 with $x = h$, $\Lambda = \Theta$, $\lambda = \theta$, $g(x,\lambda) = m_\theta^-(T) - x_\theta(T)$ is possible, which yields eq. (D.90) from eq. (D.91). □

Lemma D.13.5. *If $\Theta \subset \mathbb{R}^n$ is compact, then it holds true, under Assumptions D.1 to D.3, that*

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

for sufficiently small $h > 0$.

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

$$\|S\| \leq C\|S\|_2 = C\|S^\intercal\|_2 \leq C\|S^\intercal\|_{2,1} \tag{D.93}$$

$$\overset{eq.\ (D.20)}{=} C\sum_{j=1}^{\bar{N}}\|\Lambda_j^\intercal\theta\|_2 \tag{D.94}$$

$$\leq C\sum_{j=1}^{\bar{N}}\|\Lambda_j^\intercal\|_2 \underbrace{\|\theta\|_2}_{\leq C,\text{ since }\Theta\text{ bounded}}, \tag{D.95}$$

where $\|\cdot\|_{2,1}$ denotes the $L_{2,1}$ norm. We conclude, using Assumption D.2 and Lemma D.13.4, that

$$\|\Lambda_j^\intercal\|_2 \overset{eq.\ (D.21)}{\leq} L\max_{jk}\left[\frac{\partial}{\partial\theta_k}m_\theta^-(jh)\right] \tag{D.96}$$

$$\overset{eq.\ (D.90)}{\leq} C\left(\|\nabla_\theta x_\theta\| + h\right). \tag{D.97}$$

□

D.13.4 Proof of Theorem D.4.1

Proof. By Theorem D.3.1 and the sub-multiplicativity of the induced p-norm $\|\cdot\|_p$, we observe that

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

for some $p, q \geq 1$. Application of Lemmas D.13.3 and D.13.5 concludes the proof. □

D.14 Supplement V: Gradient and Hessian estimators for the Bayesian case

In the main paper, we only consider the maximum likelihood objective; see eq. (D.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

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

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. (D.26) and (D.27) are hence given by

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

D.15 Supplement VI: 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 D.1. Following Schillings *et al.* (2015) and Gorbach *et al.* (2017), we used this ODE with initial value $x_0 = 1_M$, 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. (D.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 $\theta^0 = 1.2 \cdot \theta^* = [0.12, 0, 0.48, 0, 0.36, 0, 0.84, 0, 0.12, 0.24]$, and the forward solutions for all likelihood evaluations were computed with step size $h = 0.05$. To create a good initialization, we accepted the first 30 proposals for PHMC and PLMC.

E A Fourier State Space Model for Bayesian ODE Filters (Kersting and Mahsereci, 2020)

Abstact: Gaussian ODE filtering is a probabilistic numerical method to solve ordinary differential equations (ODEs). It computes a Bayesian posterior over the solution from evaluations of the vector field defining the ODE. Its most popular version, which employs an integrated Brownian motion prior, uses Taylor expansions of the mean to extrapolate forward and has the same convergence rates as classical numerical methods. As the solution of many important ODEs are periodic functions (oscillators), we raise the question whether Fourier expansions can also be brought to bear within the framework of Gaussian ODE filtering. To this end, we construct a Fourier state space model for ODEs and a 'hybrid' model that combines a Taylor (Brownian motion) and Fourier state space model. We show by experiments how the hybrid model might become useful in cheaply predicting until the end of the time domain.

E.1 Introduction

Ordinary differential equations (ODEs) appear in many machine learning algorithms. In recent years, there has been a particular surge of interest in ODEs for normalizing flows (Rezende and Mohamed, 2015). This development is driven by neural ODEs (Chen *et al.*, 2018), which allow for maximum-likelihood estimation and variational inference. Neural ODEs replace learning by gradient descent with learning by ODE sensitivity analysis (Rackauckas *et al.*, 2018).

A recent recast of ODEs as a stochastic filtering problems has made it possible to solve initial value problems (IVPs) by all available Bayesian filtering methods (Tronarp *et al.*, 2019a, 2020). The resulting class of methods, called *ODE filters*, has not only fulfilled the goal of probabilistic numerics (PN) (Hennig *et al.*, 2015; Oates and Sullivan, 2019) to quantify numerical uncertainty in a Bayesian way, but has also identified the dynamic model as the fundamental internal modeling assumption of ODE solvers. This dynamic model determines how the solver extrapolates forward in time and is equivalent to a prior over the ODE solution (Kersting *et al.*, 2020b, Appendix A).

Early PN research has, to show that its new methods are indeed practical, focused mostly on creating probabilistic analogues of classical methods. This line of inquiry has

discovered that the integrated Brownian Motion (IBM) prior gives rise to ODE filters whose mean coincides with standard classical methods; see Schober *et al.* (2019). This is due to the fact that—like e.g. Runge–Kutta method—ODE filters with the IBM prior use Taylor expansions to locally predict forward; see Equation (6) in Kersting *et al.* (2020a). In the meantime, other local expansions based on the Matérn covariance function have been studied; see Tronarp *et al.* (2020). Fourier expansions, however, have not been investigated in the context of ODE filters although it is known how to incorporate them in a dynamic model; see (Solin and Särkkä, 2014). With this paper, we aim to begin filling in this gap. Since so many important ODEs are oscillators with periodic solutions, we consider Fourier expansions a promising research direction in the context of ODE filtering.

E.2 ODE Filtering for initial value problems

We consider the following IVP

$$\dot{x}(t) = f\left(x(t)\right), \ \forall t \in [0, T], \qquad x(0) = x_0 \ \in \mathbb{R}^d, \tag{E.1}$$

with vector field $f : \mathbb{R}^d \to \mathbb{R}^d$. For notational convenience, we restrict w.l.o.g. the below presentation to $d = 1$. As the solution $x : [0, T] \to \mathbb{R}^d$ in general lies in an infinite dimensional function space such as $C^2([0, T]; \mathbb{R}^d)$, a finite dimensional representation is needed to extrapolate from $x(t)$ to $x(t + h)$. Runge–Kutta methods, for example, use Taylor expansions (i.e. projections to polynomial spaces) as finite dimensional approximations of x. While classical methods only do so implicitly, Gaussian ODE filtering represents $x(t)$ explicitly in a D-dimensional *state vector*, i.e. in a stochastic process $X(t)$ from which a model of $x(t)$ can be linearly extracted:

$$x(t) \ \sim \ H_0 X(t), \quad \text{for some } H_0 \in \mathbb{R}^{d \times D}. \tag{E.2}$$

Moreover, for ODEs, the derivative has also to be linearly extractable

$$\dot{x}(t) \ \sim \ H X(T) \quad \text{for some } H \in \mathbb{R}^{d \times D}. \tag{E.3}$$

As ODE filtering is a Bayesian method, the state $X(t)$ is modeled by a stochastic process, which is usually represented by a linear time-invariant stochastic differential equation (SDE)

$$\mathrm{d}X(t) = F X(t) \ \mathrm{d}t + L \ \mathrm{d}B(t) \tag{E.4}$$

with Gaussian initial condition on $X(0)$, where the drift and diffusion matrices $F, L \in \mathbb{R}^{D \times D}$ detail the deterministic and stochastic part of the dynamics respectively. This SDE prior can be thought of as a localized definition of a Gauss–Markov process. In its

Algorithm 4 Gaussian ODE Filtering

Input: IVP(x_i, m, T), step size $h > 0$
Initialize, $t = 0$, $H_0 X(0) = x_0$ and $HX(0) = f(x_0)$
repeat
  predict state X, $t \to t + h$, along Equation (E.5)
  $t = t + h$
  update $X(t)$ by Equations (E.8) and (E.9)
until $t + h > T$

discretized form, it defines a *dynamic model*

$$p\left(X(t+h) \mid X(t)\right) = \mathcal{N}(A(h)X(t), Q(h)), \tag{E.5}$$

with matrices $A(h), Q(h) \in \mathbb{R}^{D \times D}$ which are implied in closed form by F and L. To update this model, a *measurement model* is added

$$p(Z(t) \mid X(t)) = \mathcal{N}\left(f(H_0 X(t)) - HX(t), R\right), \tag{E.6}$$

$R \geq 0$, which is conditioned on the data

$$Z(t) \coloneqq 0. \tag{E.7}$$

As f is non-linear, the above measurement model is intractable (Tronarp *et al.*, 2019a, Section 2). By substituting $f(H_0 \mathbb{E}[X(t)])$ for $f(H_0 X(t))$, we obtain the following tractable measurement model

$$p(Z(t) \mid X(t)) = \mathcal{N}\left(HX(t), R\right) \tag{E.8}$$

$$Z(t) \coloneqq f(H_0 \mathbb{E}[X(t)]). \tag{E.9}$$

We will employ this measurement model in this paper. The dynamic and measurement model together are called a *probabilistic state space model* (SSM), which Gaussian ODE filtering uses to infer x as detailed in Algorithm 4.

The classical Taylor SSM

In previous research, the most commonly recommended model uses an integrated Brownian motion as a dynamic model (prior). It is defined by inserting the following matrices into Equation (E.5):

$$A(h)_{ij} = \mathbb{I}_{i \leq j} \frac{h^{j-i}}{(j-i)!}, \tag{E.10}$$

$$Q(h)_{ij} = \sigma^2 \frac{h^{2q+1-i-j}}{(2q+1-i-j)(q-i)!(q-j)!}, \tag{E.11}$$

where σ^2 is the variance scale of the underlying Brownian motion; see Kersting *et al.* (2020a, Appendix A). Here, the state vector

$$\left[x^{(0)}(t), \ldots, x^{(q)}(t)\right] \;\sim\; X(t) \tag{E.12}$$

models the first $q \in \mathbb{N}$ derivatives of $x(t)$. Therefore, the mean prediction $X(t+h) = A(h)X(t)$ is a *Taylor expansion* of the numerical estimates of these derivatives. Consequently, since Runge–Kutta methods also use Taylor approximations of $x(t)$ (Hairer *et al.*, 1987, Section II.2), the posterior mean is similar to Runge–Kutta methods with local convergence rates of $q+1$ and global convergence rates of q. It is hence a very good method to solve generic ODEs; see Schober *et al.* (2019) and Kersting *et al.* (2020a).

E.3 Fourier models for ODEs

In this paper, we are concerned with oscillating ODEs. Hence, let the ODE be such that its solution $x(t)$ is a periodic function. Let us denote the period of x by $p > 0$, and its angular velocity by $w_0 = 2\pi/p$. For such periodic functions, a J-th order *Fourier series*, $J \in \mathbb{N}$, of x is the standard approximation:

$$x^J(t) = x_0 + \sum_{j=1}^{J} x_j(t) \;\overset{J\to\infty}{\to}\; x(t), \tag{E.13}$$

almost everywhere, where $x_0 = a_0/2$ and

$$x_j(t) = a_j \cos(w_0 j t) + b_j \sin(w_0 j t). \tag{E.14}$$

The derivative of Equation (E.13) is

$$\dot{x}^J(t) = y_0 + \sum_{j=1}^{J} -w_0 j y_j(t) \;\overset{J\to\infty}{\to}\; x(t), \tag{E.15}$$

almost everywhere, where $y_0 = 0$ and

$$y_j(t) = a_j \sin(w_0 j t) - b_j \cos(w_0 j t). \tag{E.16}$$

The exact *Fourier coefficients* are given by the integrals

$$a_j = \frac{2}{p} \int_0^p x(t) \cos(j w_0 t)\, \mathrm{d}t, \tag{E.17}$$

$$b_j = \frac{2}{p} \int_0^p x(t) \sin(j x_0 t)\, \mathrm{d}t, \tag{E.18}$$

which are, in general, intractable. Thus, learning a periodic approximation of x amounts to inferring the coefficients (a_j, b_j). To reproduce the model from Solin and Särkkä (2014, Section 3.2), we will however run inference on the corresponding harmonic oscillators $(x_j(t), y_j(t))$ instead. To this end, we first observe that, for each $j = 0, \ldots, J$, $[x_j(t), y_j(t)]$ satisfies the following differential equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_j(t) \\ y_j(t) \end{bmatrix} = \begin{bmatrix} 0 & -jw_0 \\ jw_0 & 0 \end{bmatrix}\begin{bmatrix} x_j(t) \\ y_j(t) \end{bmatrix}. \tag{E.19}$$

Note that, if we set $[x_0(0), y_0(0)] = [a_0/2, 0]$ and $[x_j(0), y_j(0)] = [a_j, -b_j]$, then the only solution of Equation (E.19) is indeed $[x_j(t), y_j(t)]$ as defined in Equations (E.14) and (E.16). Since we (in general) do not know the Fourier coefficients (a_j, b_j), we model the initial values with a Gaussian probability distribution: $[x_j(0), y_j(0)] \sim \mathcal{N}(\mathbf{0}, q_j^2 \boldsymbol{I})$, for some $q_j^2 > 0$. We then model these oscillators by a stochastic process $X(t)$, i.e.

$$[x_0(t), y_0(t), x_1(t), y_1(t), \ldots, x_J(t), y_J(t)] \sim X(t) \tag{E.20}$$

which (according to Equation (E.19)) follows the SDE, Equation (E.4), if and only if

$$F = \mathrm{diag}(F_1, \ldots, F_J), \qquad \text{with blocks} \tag{E.21}$$

$$F_j = \begin{bmatrix} 0 & -jw_0 \\ jw_0 & 0 \end{bmatrix}, \qquad \text{and} \tag{E.22}$$

$$L = \mathbf{0} \ \in \mathbb{R}^{2(J+1)\times 2(J+1)}, \tag{E.23}$$

and if the initial condition is $[X(0)_{2j}, X(0)_{2j+1}] \sim \mathcal{N}(\mathbf{0}, q_j^2 \boldsymbol{I})$. It is natural that there is no diffusion ($L = 0$), as the Fourier coefficients (unlike Taylor coefficients) of a periodic signal $x(t)$ do not change in t. Since we want the prior defined by this SDE to be a zero-mean Gaussian Process with the canonical periodic covariance function

$$k_p(t, t') = \sigma^2 \exp\left(-\frac{2\sin^2\left(w_0 \frac{t-t'}{2}\right)}{l^2}\right), \tag{E.24}$$

we have to set

$$q_j^2 = \frac{2I_j(l^{-2})}{\exp(l^{-2})}, \qquad \text{for } j = 1, \ldots, J, \tag{E.25}$$

where $I_j(z)$ is the modified Bessel function of the first kind of order j; see Solin and Särkkä (2014, Eq. 27).

Dynamic model The implied matrices for the dynamic model, Equation (E.5), are

now given by

$$A = \operatorname{diag}(A_0, \ldots, A_J), \qquad \text{with blocks} \tag{E.26}$$

$$A_j = \begin{bmatrix} \cos(w_0 jt) & -\sin(w_0 jt) \\ \sin(w_0 jt) & \cos(w_0 jt) \end{bmatrix}, \qquad \text{and} \tag{E.27}$$

$$Q = \mathbf{0} \ \in \mathbb{R}^{2(J+1)\times 2(J+1)}. \tag{E.28}$$

Measurement Model This dynamic model is, like all dynamic models, combinable with all measurement models. For ODEs, we need, by Equation (E.8), a model H_0 that extracts $x(t)$ and a model H that extracts the derivative from the state $X(t)$ of Equation (E.20). By Equations (E.13) and (E.15), this is satisfied by

$$H_0 = [1, 0, 1, 0, \ldots, 1, 0] \ \in \mathbb{R}^{1\times 2(J+1)}, \quad \text{and} \tag{E.29}$$

$$H = \Big[0, 0, 0, -1w_0, 0, -2w_0, \ldots, 0, -Jw_0\Big] \in \mathbb{R}^{1\times 2(J+1)}.$$

E.3.1 Discussion of the Fourier model

As their coefficients do not depend on a support point, Fourier models are (unlike Taylor models) global expansions. Hence, they are best at extrapolating globally, while Taylor methods excel at extrapolating locally. As ODE methods are usually designed for a small step size $h > 0$, the Taylor approximation is the standard approximation in ODE solvers, such as Runge–Kutta methods. Hence, we expect the Fourier SSM to be more useful for global extrapolation with larger step sizes. Accordingly, we suggest a hybrid ODE solver which combines the Taylor and the Fourier state space model in the next section. This model could be used to extrapolate from a certain time, after learning the Fourier coefficients with data from the Taylor model.

E.4 The hybrid Taylor-Fourier model

As Taylor approximations excel at local approximations and Fourier approximations at global approximations of periodic signals, we combine both to the hybrid Taylor-Fourier model. The idea is that one can learn the Fourier coefficients (a_j, b_j) from Equations (E.17) and (E.18) with data from the Taylor SSM and then extrapolate with the Fourier SSM using the learned coefficients. Let us denote the Taylor SSM by $(A^{\text{Tay}}, Q^{\text{Tay}}, H^{\text{Tay}})$ and the Fourier SSM $(A^{\text{Four}}, Q^{\text{Four}}, H^{\text{Four}})$. The hybrid Filter works now as follows: It splits the time domain $[0, T]$ of the ODE into two parts $[0, T_p]$ and $[T_p, T]$ for some *prediction time point* $T_p \in (0, T)$. On the first interval $[0, T_p]$, we solve the ODE with the classical Gaussian ODE filter with the Taylor SSM, and we, simultaneously, train the Fourier SSM with the data from the Taylor model. On the second interval $[T_p, T]$, we just predict along the Fourier dynamical model defined by

$(A^{\text{Four}}, Q^{\text{Four}})$.
As the computation on the time interval $[T_p, T]$ does not require additional evaluations of f and is therefore almost free, we hope that that this model turns out useful to reduce the computational time of solving periodic ODEs. In the next section we present some experiments which, while not practical yet, highlight that this hybrid model in principle works.

E.5 Experiments

We try a Gaussian ODE filter with hybrid Taylor-Fourier SSM on two standard oscillating ODEs: the Van der Pol oscillator

$$\begin{cases} \dot{x}_1(t) &= \mu(x_1(t) - \frac{1}{3}x_1(t)^3 - x_2(t)), \\ \dot{x}_2(t) &= \frac{1}{\mu}x_1(t), \end{cases} \tag{E.30}$$

with $\mu = 5$, $x(0) = [1, -1]$, $T = 50$, and the FitzHugh–Nagumo model

$$\begin{cases} \dot{x}_1(t) &= x_1(t) - \frac{x_1(t)^3}{3} - x_2(t) + I, \\ \dot{x}_2(t) &= \frac{1}{\tau}\left(x_1(t) + a - x_2(t)\right), \end{cases} \tag{E.31}$$

with parameters $(I, a, b, \tau) = (0.5, 0.7, 0.8, 10.0)$, $x(0) = [1., 0.1]$ and $T = 50$.

E.5.1 Experimental set-up

We set the parameters of the Taylor model as follows: $q = 1$ and $\sigma^2 = 1$. Moreover, we choose the following parameters of the Fourier model: $l = 3$, $w_0 = 1$, $\sigma^2 = 1$, $J = 3$ $p = 2\pi$ and $R = 0$. Note that these parameters are not fine-tuned. We define the prediction time T_p for both ODEs to be $T_p = \frac{3}{4}T = 37.5$.

E.5.2 Results

The plots in Figures E.1 and E.2 show that the hybrid ODE filter works in principle. It picks up some structure from the trajectories on $[0, T_p]$ and can extrapolate forward by a sum of harmonic oscillators. The quality of the extrapolation is, however, not good enough yet. We suspect that this is due to our ad-hoc choice of parameters. Since our state space model is by Solin and Särkkä (2014) an approximation of GP regression with periodic kernel and derivative observations, it should be possible to make the extrapolation as accurate as periodic GP regression (at least for $J \to \infty$).
In particular, it should be possible to choose the angular velocity w_0 in a more principled way and thereby match the period of the oscillator more precisely. Our choice of $w_0 = 1$ is particularly off in Figure E.2. We also believe that a more accurate extrapolation

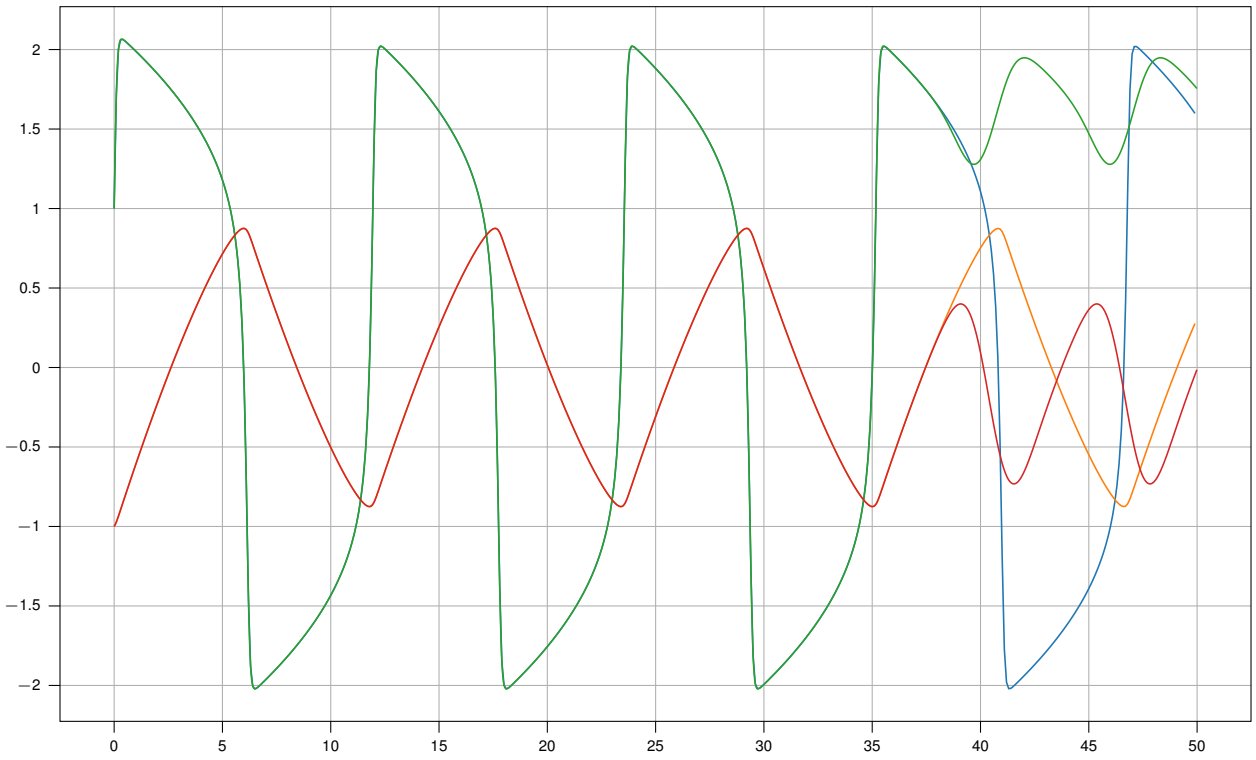


Figure E.1: Hybrid Filter on Van der Pol ODE, Equation (E.30). Blue and yellow line are true curves x_1 and x_2. Red and green line are hybrid filter mean with prediction from $T_p = 37.5$.

could be achieved if a larger J is chosen. This will probably only work well once we have found a suitable way to choose w_0. Moreover, future research should examine which $J + 1$ data points from the Taylor model should be used for the Fourier model—which is an active learning task with Gaussian processes (Seo *et al.*, 2000).

E.6 Conclusion

We examined how Fourier state space models can be employed in Gaussian ODE filtering, to solve oscillating ODEs. To this end, we developed a novel Fourier sate space model that is applicable to ODEs. We reasoned that it might outperform Taylor methods on global extrapolation tasks. Since Fourier expansions are not locally accurate enough to serve as a practical ODE solver on its own, we have developed the hybrid ODE Solver which combines Taylor and Fourier expansions. It first uses a Taylor SSM to compute up to a certain time T_p while training a Fourier SSM 'on the fly', and then uses the so-trained Fourier SSM to predict forward. We demonstrated that, in principle, this can work—even if we are not yet satisfied with the quality of the Fourier prediction.

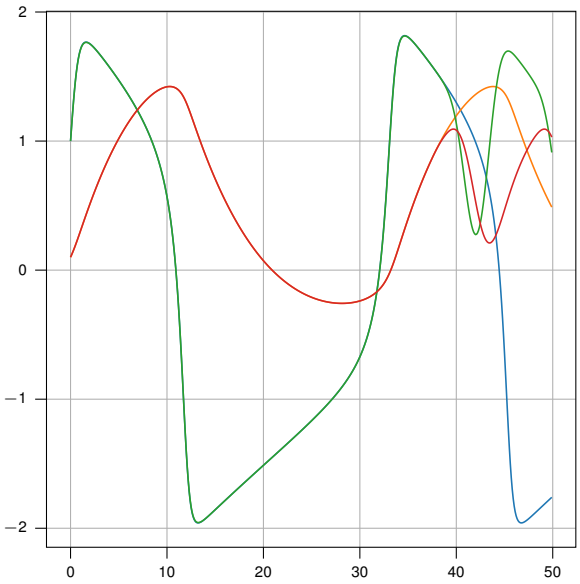


Figure E.2: Hybrid Filter on FitzHugh–Nagumo ODE, Equation (E.31). Blue and yellow line are true curves x_1 and x_2. Red and green line are hybrid filter mean with prediction from $T_p = 37.5$.

Future research should examine how the Fourier coefficients can be learned better—e.g. by finding ways to choose the Fourier parameters (w_0, J) better or to employ smart active learning (Seo *et al.*, 2000) for the Fourier coefficients (a_j, b_j). Since the desired Fourier coefficients are integrals, maybe ideas from Bayesian quadrature (Briol *et al.*, 2019) can be borrowed for this purpose; see Equations (E.17) and (E.18). We hope that this might pave the way to almost cost-free predictions of oscillating systems which could come in useful in settings where ODEs have to be solved over a long time horizon with very limited budget or where a (reinforcement learning) system has to make sudden decisions in the context of ODE dynamics (Deisenroth and Rasmussen, 2011).
If so, then exciting new ideas unknown to classical numerical analysis—such as quasi-periodic extrapolations (Solin and Särkkä, 2014, Section 3.5)—could be introduced to ODE solvers. Such a development would also benefit probabilistic numerical methods for boundary value problems (John *et al.*, 2019), PDEs (Oates *et al.*, 2019), and ODE inverse problems (Kersting *et al.*, 2020b).
In machine learning, such advances could provide better uncertainty quantification (of the numerical error) for continuous normalizing flows with ODEs, see (Chen *et al.*, 2018, Section 4)—where a free-form ODE, potentially an oscillator (Grathwohl *et al.*, 2019), has to be numerically solved to approximate the transformed state and the numerical error is, to date, not accounted for.