The Past Does Matter: Correlation of Subsequent States in Trajectory Predictions of Gaussian Process Models

Steffen Ridderbusch[1] **Sina Ober-Blöbaum**[2] **Paul Goulart**[1]

[1]Control Group, Dept. of Engineering Science, University of Oxford, Oxford, UK
[2]Numerical Mathematics and Control, University of Paderborn, Paderborn, Germany

Abstract

Computing the distribution of trajectories from a Gaussian Process model of a dynamical system is an important challenge in utilizing such models. Motivated by the computational cost of sampling-based approaches, we consider approximations of the model's output and trajectory distribution. We show that previous work on uncertainty propagation, focussed on discrete state-space models, incorrectly included an independence assumption between subsequent states of the predicted trajectories. Expanding these ideas to continuous ordinary differential equation models, we illustrate the implications of this assumption and propose a novel piecewise linear approximation of Gaussian Processes to mitigate them.

1 INTRODUCTION

The context of this work is the combination of dynamical systems theory, with its history of widespread applications in science and engineering, and Gaussian Process (GP) models. To utilize these models, one must be able to make predictions for future trajectories, ideally in a way that incorporates the uncertainty of the model. These predictions should also be cheap to compute and qualitatively accurate.

The computation of trajectories depends on the model class. One example of *continuous* models, our main focus, are ordinary differential equation (ODE) models of the form

$$\dot{x} = f(x). \tag{1}$$

We can represent f as a GP, which means that we learn the *vector field* of a dynamical system, as seen in Ridderbusch et al., 2021 and Heinonen et al., 2018. This approach is conceptually similar to *UniversalODEs*, which use a Neural Network instead of a GP [Rackauckas et al., 2020].

We note explicitly that this setting does not result in a stochastic differential equation or random dynamical system, where *aleatory* uncertainty arises from inherent stochasticity [Banks et al., 2012]. Instead, a GP represents a distribution over a function space, conditioned on available pairs of input-output data, which we assume contains the true underlying deterministic function. This means that GP models capture *epistemic* uncertainty, as the uncertainty about the true model decreases with additional data.

The state x of a continuous model at time t is given by the *flow* $\varphi^t_f(x_0)$, which is the solution of the ODE (1) for the initial value x_0. However, the flow map is generally not available analytically and instead must be computed approximately at discrete times via numerical integration. The simplest such method is the explicit Euler's method

$$x_{n+1} = \varphi^h_f(x_n) \approx x_n + hf(x_n), \tag{2}$$

where h is the step size, which can be varied over subsequent steps. There exists a wide range of more complex numerical solvers, differing in their order of convergence, stability, and computational cost. One motivation for this work is to apply those methods to GP-based ODEs while also accounting for uncertainty. For details on dynamical systems and numerical integration see for example Guckenheimer et al., 2013.

A related class of models are discrete dynamical systems, which directly assume a model of the form

$$x_{n+1} = f(x_n), \tag{3}$$

instead of discretizing a continuous model. Representing the discrete flow map with a GP is a more popular alternative to learning the vector-field, and the resulting model is called a *state-space GP* [Kamthe et al., 2017; Buisson-Fenet et al., 2020; Girard et al., 2003; Groot et al., 2011]. This approach implicitly assumes a fixed step size h between subsequent states x_n and x_{n+1}, often the measurement interval of the available time series data. One computes a trajectory by iteratively applying f. Some approaches additionally use auto-regression, taking into account l past states [Groot et al., 2011; Kocijan, 2015; Nghiem, 2019].

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

The challenge with uncertainty propagation for both continuous and discrete models is that mapping a random variable through a nonlinear function is generally intractable, even for the generally very tractable normal distribution.

In the context of dynamical system models, this problem is compounded. To obtain the distribution of the states X_{n+1}, X_{n+2} and so forth in the trajectory of a state-space GP one has to repeatedly map the state X_n through the distribution of nonlinear functions represented by the GP. For continuous models, the difficulty increases further. To compute subsequent states, we must compute a distribution of gradients $f(X)$ for the state X, which needs to be combined with the distribution of current state X_n as seen in (2). For higher-order methods, additional gradient distributions at intermediate states are required.

To our knowledge, this work is the first to consider approximate uncertainty propagation through numerical integrators for ODEs. However, there has been some work on uncertainty propagation in state-space GPs. Girard et al. published a series of results on approximating the output distribution when mapping a normal distribution through a GP, within the context of making iterative multiple-step-ahead predictions for discrete models. In Girard et al., 2002 the authors derive an approximation for the mean and variance of the output based on Taylor series approximations of the predicted mean and variance, using derivatives for the GP kernel function. A subsequent result in Girard et al., 2003 showed that when using the squared exponential kernel specifically, the mean and the variance of the output distribution can be obtained analytically. These results are called *moment matching* in Kamthe et al., 2017. Other work has applied sampling-based approaches to estimate uncertainty in state-space GPs [Hewing et al., 2020] and for continuous GP dynamics [Hegde et al., 2022].

The combination of ODEs and GPs has also been considered in the context of *probabilistic numerics*. Well-known examples include Schober et al., 2014 and the work mentioned in Hennig et al., 2022. At a high level, ODE solvers under the probabilistic numerics umbrella consider a fully deterministic model and describe the continuous solution or trajectory as a Gaussian Process. Each discrete iteration step of the solver is then seen as a noisy observation, which makes it possible to quantify the uncertainty in the trajectory introduced by the integration algorithm. However, the setting in our work is orthogonal. The model itself is uncertain and we aim to include this uncertainty at least approximately in the trajectory prediction.

We briefly summarize the setting in this paper. We assume that the data-generating system has an unknown, but deterministic, dynamical system model. Given noisy observations, we express the different possible models and their likelihood as a distribution, represented for example by a finite GP conditioned on collected data. For simplicity, we completely omit all hyperparameter training and data conversion. Instead, we assume that model distribution is given, which allows us to focus on how to compute trajectory uncertainty from the model uncertainty.

Our main contributions are the following: We show that existing approaches to uncertainty propagation based on approximating the output distribution of the exact model assume the independence of the state and the model, and how this affects the predicted trajectory distribution. We further propose an alternative approach based on a piecewise linear model approximation that can be solved exactly, resulting in what we call the PULL (Propagating Uncertainty through Local Linearization) class of solvers. We demonstrate the effectiveness of the PULL version of the explicit Euler and include discussions on its convergence and limitations.

2 REVIEW

2.1 GAUSSIAN PROCESSES

We review briefly the basics of GPs [Rasmussen et al., 2006]. We assume we have N output observations $y^i = f(x^i) + \epsilon^i$ with Gaussian noise $\epsilon^i \sim \mathcal{N}(0, \sigma_n^2)$ for known inputs x^i to an unknown function $f : \mathbb{R}^m \to \mathbb{R}^d$. We express f as a GP, i.e. as a distribution over functions. We specify a prior $\mathbb{P}(f)$ via a mean function $m(x)$, generally assumed to be zero, and a kernel function $k(x, x')$, which uniquely determines a Reproducing Kernel Hilbert Space containing all possible realizations of f.

For any finite subset of random variable outputs f^i corresponding to known inputs x^i the GP determines a joint Gaussian distribution, and by conditioning on the initial observations $\mathcal{D} = (x^i, y^i)$ we obtain the posterior distribution $\mathbb{P}(f|\mathcal{D})$, which allows us to predict the output $f(\hat{x})$ at a new input $\hat{x}$. Since we are considering a distribution of functions, we will obtain a distribution of outputs with the mean $\mu_f(\hat{x})$ and variance $\sigma_f^2(\hat{x})$ determined by

$$\mu_f(\hat{x}) = K_{\hat{x},\mathbf{x}}(K_{\mathbf{x},\mathbf{x}} + \sigma_n^2 I)^{-1}\mathbf{y} \tag{4a}$$

$$\sigma_f^2(\hat{x}) = K_{\hat{x},\hat{x}} - K_{\hat{x},\mathbf{x}}(K_{\mathbf{x},\mathbf{x}} + \sigma_n^2 I)^{-1}K_{\mathbf{x},\hat{x}} \tag{4b}$$

where $\mathbf{x} = [x^1, \ldots, x^N]$, $\mathbf{y} = [y^1, \ldots, y^N]$, and $K_{x,x'} = [k(x^i, x'^j)]_{i,j}$ is the kernel or covariance matrix. The most common kernel is the squared exponential kernel

$$k(x, x') = \exp\left(-\frac{1}{2}(x - x')^T W^{-1}(x - x')\right), \tag{5}$$

where $W = \text{diag}(w_1, \ldots, w_M)$ is the diagonal matrix of length scales. We also use this kernel in our work, but our results can be applied for any kernel that is at least once differentiable.

In the context of dynamical systems the input and output dimensions are equal, hence $m = d$. A number of vector-valued kernels exist Alvarez et al., 2012, but it is often assumed that each output can be treated independently with d GPs $f : \mathbb{R}^m \to \mathbb{R}$. For this work, we consider only one-dimensional dynamical systems and therefore one-dimensional GPs.

2.2 SAMPLING GAUSSIAN PROCESSES

GPs are distributions over function spaces which are generally infinite-dimensional and therefore inherently difficult to sample numerically. The standard option [Wilson et al., 2020] to compute the output of function realizations f^* at specified input locations x^* from the posterior distribution, is to generate normally distributed random variables $\zeta \sim \mathcal{N}(0, I)$, and transform them according to the GP posterior such that

$$f^*|\mathbf{y}, \mathbf{x}, x^* \; = m^* + K_{*,*}^{1/2}\zeta. \tag{6}$$

Here, $(\cdot)^{1/2}$ indicates a matrix square root such as the Cholesky factor, m^* is the GP mean $\mu_f(x^*)$ from (4a) and $K_{*,*}$ is the covariance matrix for the sample input x^* via (4b). In essence, this is a *grid-based* approach which, while numerically exact, is computationally costly as it scales cubically with the number of function values sampled. To evaluate a function sample f^* at arbitrary locations between grid points one can use standard interpolation methods.

To mitigate the cubic scaling, alternative algorithms have been developed. Wilson et al. Wilson et al., 2020 propose a combination of decoupled bases, specifically *Fourier basis functions* Rahimi et al., 2007, and kernel bases via a sparse GP approach. This has been used in other recent work Hegde et al., 2022; Ensinger et al., 2022 in the context of vector field models.

An approach of this kind was also mentioned in Hewing et al., 2020, along with a *memory-based* approach, which generates samples subsequently while conditioning each sample on at least some previous ones. It has also been used for uncertainty propagation in continuous dynamics [Ridderbusch et al., 2021].

For this work we will use samples via (6) as a source of ground truth for comparison, since it is the most accurate option to compare with.

3 A LINEAR PERSPECTIVE

We begin with the simplest possible example, a prototypical linear model. We highlight the expected behaviour and introduce the approach by Girard et al., 2002 based on approximating the output distribution of each subsequent state. We then illustrate an issue with this approach for the prediction of trajectory uncertainty.

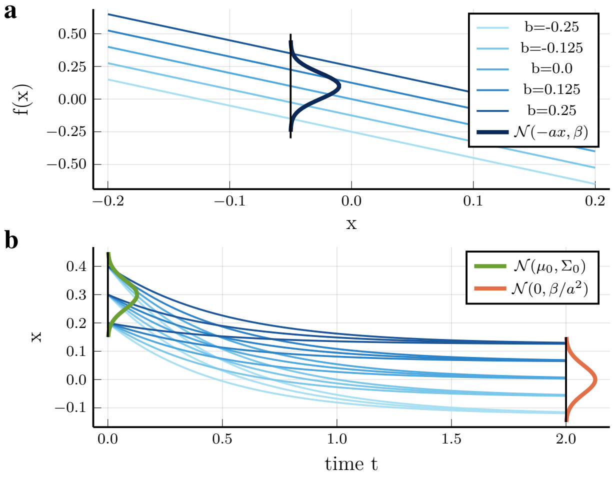

Figure 1: **Linear Prototype. a:** The distribution of linear ODEs. **b:** Solving sampled linear ODEs for a distribution of initial values $\mathcal{N}(\mu_0, \Sigma_0)$ shows trajectories converging to the model parameter-dependent fixed point distribution.

3.1 A LINEAR PROTOTYPE

Consider a distribution of stable linear ODEs of the form

$$f(x) = -ax + B, \quad \text{with } a \in \mathbb{R}^+,\, B \sim \mathcal{N}(0, \beta), \tag{7}$$

shown in Fig. 1a, whose mean and variance are

$$\mu(x) = -ax \quad \text{and} \quad \sigma^2(x) = \beta. \tag{8}$$

The fixed point distribution of this distribution of ODEs is

$$\hat{X} \sim \mathcal{N}(0, \beta/a^2), \tag{9}$$

which matches the behaviour shown in Fig. 1b. There, we sample realizations b of B to obtain deterministic linear ODEs, which we can solve for realizations x_0 of the initial value distribution $X_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$. As we will discuss in more detail in the following, the effect of the initial value is transient and trajectories converge to a fixed point only depending on the realization of the model variable B.

3.2 APPROXIMATING THE OUTPUT DISTRIBUTION

Although the distribution of each subsequent state random variable X_n is generally not normally distributed, the idea of Girard et al., 2003 is to compute the mean and variance of each subsequent state of a discrete model (3) and approximate its unknown general distribution with normal distribution with the same moments. In this section, we extend this idea to trajectories resulting from numerically integrating continuous models.

For the example of the explicit Euler method (2), a naive approach would be to use moment-matching to approximate

$f(X_n)$ and then compute the sum of X_n and $f(X_n)$ as the sum of two independent normal random variables. However, this leads to incorrect variance growth over time, as they are clearly correlated. Instead, we extend the idea to the function $g(x) = x + hf(x)$ and compute the mean $\mathbb{E}[g(X_n)]$ and variance $\text{var}(g(X_n))$ to once again approximate the general distribution of $g(X_n)$ with a normal distribution.

Let the input X_n be

$$X_n \sim \mathcal{N}(\nu_n, \Sigma_n). \tag{10}$$

The output of the GP for a specific sample x^* of X_n is then

$$f(x^*) \sim \mathcal{N}(\mu(x^*), \sigma^2(x^*)). \tag{11}$$

The distribution of $X_{n+1} = g(X_n)$ will generally not be normal, but we can at least compute its mean $\nu_{n+1} = \mathbb{E}[X_{n+1}]$ and variance $\Sigma_{n+1} = \text{var}(X_{n+1})$.

From the law of iterated expectations it follows that the mean of the output distribution of one Euler step is given by

$$\begin{aligned} \nu_{n+1}(\nu_n, \Sigma_n) &= \mathbb{E}\left[X_n + hf(X_n)\right] \\ &= \nu_n + h\,\mathbb{E}[\mu(X_n)] \end{aligned} \tag{12}$$

The corresponding variance is given by

$$\begin{aligned} \Sigma_{n+1}(\nu_n, \Sigma_n) =& \text{var}\big(X_n + hf(X_n)\big) \\ =& \text{var}(X_n) + h^2 \text{var}\big(f(X_n)\big) \\ &+ 2h\,\text{cov}\big(X_n, f(X_n)\big) \end{aligned} \tag{13}$$

and from the law of total variance it follows that

$$\begin{aligned} &\text{var}(f(X_n)) \\ =& \,\mathbb{E}\left[\sigma^2(X_n)\right] + \mathbb{E}\left[\mu(X_n)^2\right] - \mathbb{E}\left[\mu(X_n)\right]^2. \end{aligned} \tag{14}$$

This leaves the covariance between X_n and $f(X_n)$. The law of total covariance states that

$$\begin{aligned} \text{cov}\left(X_n, f(X_n)\right) =& \,\text{cov}_{X_n}\left(\mathbb{E}_f[X_n|X_n], \mathbb{E}_f[f(X_n)|X_n]\right) \\ &+ \mathbb{E}_{X_n}[\text{cov}_f(X_n, f(X_n)|X_n)] \end{aligned} \tag{15}$$

The first term can be written as

$$\begin{aligned} &\text{cov}_{X_n}\left(\mathbb{E}_f[X_n|X_n], \mathbb{E}_f[f(X_n)|X_n]\right) \\ =& \,\mathbb{E}\left[X_n \mu(X_n)\right] - \nu_n \mathbb{E}\left[\mu(X_n)\right], \end{aligned} \tag{16}$$

and the second part resolves to

$$\begin{aligned} &\mathbb{E}_{X_n}[\text{cov}_f(X_n, f(X_n)|X_n)] \\ =& \,\mathbb{E}_{X_n}\big[\mathbb{E}_f[X_n f(X_n)|X_n]\big] - \mathbb{E}_{X_n}\big[X_n \mu(X_n)\big]. \end{aligned} \tag{17}$$

We further write

$$\begin{aligned} &\mathbb{E}_f[X_n f(X_n)|X_n = x_n] \\ =& \frac{1}{\phi_{X_n}(x_n)} \int x_n \hat{f}(x_n)\, \phi_{X_n,f}(x_n, \hat{f}) d\hat{f} \end{aligned} \tag{18}$$

where $\hat{f}$ is a realisation of f, ϕ_{X_n} is the probability density of X_n and $\phi_{X_n,f}$ the joint probability density of f and X_n.

The independence assumption If X_n and f are independent, the joint distribution simplifies to

$$\phi_{X_n,f}(x_n, \hat{f}) = \phi_{X_n}(x_n)\phi_f(\hat{f}), \tag{19}$$

and we find

$$\mathbb{E}_f[X_n f(X_n)] = X_n \mu(X_n) \tag{20}$$

Inserting (20) into (17), which is further inserted with (16) into (13), results in

$$\begin{aligned} &\Sigma_{n+1}(\nu_n, \Sigma_n) = \Sigma_n \\ &+ h^2\big(\mathbb{E}[\sigma^2(X_n)] + \mathbb{E}[\mu(X_n)^2] - \mathbb{E}[\mu(X_n)]^2\big) \\ &+ 2h\big(\mathbb{E}[X_n \mu(X_n)] - \nu_n \mathbb{E}[\mu(X_n)]\big). \end{aligned} \tag{21}$$

This is the direct extension of the original moment-matching approach in Girard et al., 2003 and Groot et al., 2011.

The dependence of X_n and f However, this independence assumption is incorrect in general for trajectory predictions. The state X_n is a function of the initial value and, critically, of the model f, resulting from telescope expression

$$\begin{aligned} &X_n = X_{n-1} + hf(X_{n-1}) \\ =& X_{n-2} + hf(X_{n-2}) + hf(X_{n-2} + hf(X_{n-2})) = \ldots \\ =& X_n(f, X_0). \end{aligned} \tag{22}$$

This means, to correctly compute the expected value over f in (18), we must incorporate (22) and somehow compute or approximate the resulting expression. Given the substantial difficulty of this, just using (21) is attractive. However, we will show in the following section that doing so leads to an incorrect behaviour that cannot be neglected, even for the simplest possible model.

3.3 RETURNING TO THE LINEAR PROTOTYPE

We return now to the simple linear model (7). The analytical solution for the flow of (7) is

$$\varphi^t(X_0) = \mathrm{e}^{-at}\, X_0 + \frac{B}{a}(1 - \mathrm{e}^{-at}). \tag{23}$$

We assume that the initial value and the model parameter B are independent, so $\text{cov}(X_0, B) = 0$. This results in the time-dependent mean and variance

$$\nu_t = \mathbb{E}[\varphi^t(X_0)] = \mathrm{e}^{-at}\, \mu_0 \tag{24a}$$

$$\begin{aligned} \Sigma_t &= \text{var}(\varphi^t(X_0)) \\ &= \mathrm{e}^{-2at}\, \Sigma_0 + \frac{\beta}{a^2}(1 - \mathrm{e}^{-at})^2. \end{aligned} \tag{24b}$$

The expected values from the previous sections resolve to

$$\mathbb{E}[\mu(X_n)] = -a\nu_n \tag{25a}$$

$$\mathbb{E}[\sigma^2(X_n)] = \beta \tag{25b}$$

$$\mathbb{E}[\mu(X_n)^2] = a^2(\Sigma_n + \nu_n^2) \tag{25c}$$

$$\mathbb{E}[X_n \mu(X_n)] = -a(\Sigma_n + \nu_n^2), \tag{25d}$$

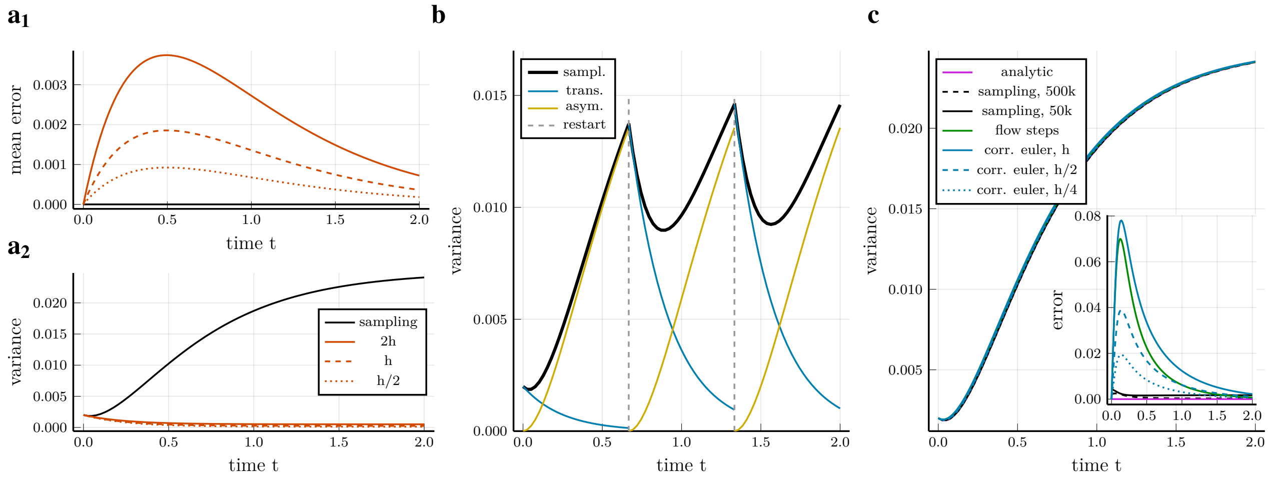

Figure 2: **Computing Trajectory Distributions a:** With the independence assumption, decreasing the Euler step size decreases the mean error. However, the variance is vastly underestimated, and the error increases with smaller step sizes due to (26b). The reason is shown in **b**, where the black line represents the sampling result, with three restarts, each time again sampling from the final distribution of the previous segment. It matches the sum of the analytic transient and asymptotic terms. **c:** Iterating the flow and taking Euler steps with the correct covariance, we obtain results that match the analytic solution and sampling. Decreasing the step size correctly reduces the variance error compared to the analytic solution.

which turns (12) and (21) into the iterations

$$\nu_{n+1} = (1 - ah)\nu_n \tag{26a}$$

$$\Sigma_{n+1} = \Sigma_n + h^2\beta + h^2a^2\Sigma_n - 2ha\Sigma_n. \tag{26b}$$

Here we get the first indication that (21) is incorrect in this context. While the exact flow (24b) results in the fixed point

$$\hat{\Sigma}^{\text{exact}} = \frac{\beta}{a^2}, \tag{27}$$

which matches the variance of (9), the variance iteration map (26b) results in

$$\hat{\Sigma}^{\text{euler}} = \frac{\beta}{a^2}\left(\frac{ah}{2 - ah}\right). \tag{28}$$

The Euler steps converge to a fixed point distribution that depends on the step size h, which is also visible in Fig. 2a. Even worse, the variance is vastly underestimated over the entire time span, and the error gets larger as the step size decreases. This is problematic – the error of a method should generally not increase with smaller step sizes.

Remarkably, we obtain a similar behaviour if we consider (24b) as an iterable function of the state X_i and step size h,

$$\Sigma^h_{X_n} = \mathrm{e}^{-2ah}\,\Sigma_n + \frac{\beta}{a^2}(1 - \mathrm{e}^{-ah})^2. \tag{29}$$

This map can be viewed as equivalent to repeatedly applying a state-space GP. It has the fixed point

$$\hat{\Sigma}^{\text{iter. flow}} = \frac{\beta}{a^2}\tanh\left(\frac{ah}{2}\right) \tag{30}$$

This is surprising, since the analytical flow map should have the semi-group property, such that $(\varphi^t \circ \varphi^s)(X_0) = \varphi^{t+s}X_0$. There should be no difference between applying the flow over multiple smaller intervals or over one large interval.

To understand this behaviour, observe that (23) is a sum of two terms. For stable systems, the first term captures the *transient* effect of the initial value, and the second term the *asymptotic* behaviour set by the model parameter B. When applying the flow to the state X_n instead of the initial value X_0, the state has already been affected by the model and at least some of its uncertainty is asymptotic. The iterative schemes above are equivalent to restarting with an independent initial value after each time step (see Fig. 2b).

The repeated transient explains the additional factor in $\hat{\Sigma}^{\text{iter. flow}}$, as smaller step sizes correspond to more "restarts" in the same time period. For $ah \to \infty$, we see either larger steps or faster dynamics, such that the transient behaviour fully decays in each step. The additional factor in $\hat{\Sigma}^{\text{euler}}$ similarly depends on ah. However, the expression fails for $ah \to 2$, the stability limit of the explicit Euler.

Therefore, the discrepancy between (27), (28) and (30) arises from the covariance between the states and the model parameter, as mentioned at the end of the previous section. While it is valid to assume that $\operatorname{cov}(X_0, B) = 0$, all subsequent states X_i depend on the model, in this case on the random model parameter B, which means that they are not independent. Instead, we find

$$\begin{aligned}\operatorname{cov}(X_n, B) &= \operatorname{cov}\left(\mathrm{e}^{-anh}\,X_0 + \frac{B}{a}(1 - \mathrm{e}^{-anh}),\, B\right)\\ &= \frac{\beta}{a}(1 - \mathrm{e}^{-ahn})\end{aligned} \tag{31}$$

Adding this covariance to (29), we get

$$\Sigma^h_{X_n} = \mathrm{e}^{-2ah}\,\Sigma_n + \frac{\beta}{a^2}(1-\mathrm{e}^{-ah})^2 + 2\frac{\beta}{a^2}(1-\mathrm{e}^{-ah})\,\mathrm{e}^{-ah}(1-\mathrm{e}^{-ahn}) \tag{32}$$

If X_n is the result of subsequent Euler steps,

$$\begin{aligned} X_n =& X_{n-1} + h(-aX_{n-1} + B) \\ =& (1-ah)X_{n-1} + hB \end{aligned} \tag{33}$$

using a telescope sum we find $\mathrm{cov}(X_n, B)$

$$\begin{aligned} &\mathrm{cov}(X_n, B) = \mathrm{cov}\left((1-ah)X_{n-1} + hB,\, B\right) \\ &= (1-ah)\mathrm{cov}(X_{n-1}, B) + h\,\mathrm{cov}(B,B) = \dots \\ &= (1-ah)^n \underbrace{\mathrm{cov}(X_0, B)}_{0} \\ &+ h\sum_{i=0}^{n-1}(1-ah)^{n-1-i}\underbrace{\mathrm{cov}(B,B)}_{\beta}. \end{aligned} \tag{34}$$

We insert this into the corrected version of (26b)

$$\begin{aligned} \Sigma_{n+1} =& (1-ah)^2\Sigma_n + h^2\beta \\ &+ 2h(1-ah)\mathrm{cov}(X_n, B). \end{aligned} \tag{35}$$

We show the results of using (32) and (35) in Fig. 2c. The variance from the iterative solutions matches the analytic solution and the variance of the sampled solutions (see Fig. 2c), and decreasing the step size correctly decreases the variance error. We also note that despite the simplicity of the model, we need to draw a substantial number of samples to match the analytical results.

This result is noteworthy beyond continuous dynamics. Applying a flow map like (29) with some fixed step size h iteratively without the additional covariance term is equivalent to repeatedly applying a learned state-space GP, as proposed in Girard et al., 2003 and applied in Kamthe et al., 2017. This has has previously been observed by for example Ialongo et al. Ialongo et al., 2019, but attributed to the use of variational methods.

4 LOCAL LINEARIZATION

In this section, we address the general case of model distributions represented by a GP. Even discarding the correlation with past states, to use (12) and (21) to compute values for the mean $\nu_{n+1}(\nu_n, \Sigma_n)$ and the variance $\Sigma_{n+1}(\nu_n, \Sigma_n)$, we need expressions for the expected values $\mathbb{E}[\mu(X_n)]$, $\mathbb{E}[\sigma^2(X_n)]$, $\mathbb{E}[\mu(X_n)^2]$ and $\mathbb{E}[X_n\mu(X_n)]$. For the first three terms, we can find analytical expressions in Girard et al., 2002 for the squared exponential kernel, and approximate expressions using linearization for all other kernels in Girard et al., 2003. In Groot et al., 2011 we also find the previously listed expressions for the squared exponential kernel and an expression for the term $\mathbb{E}[X_n\mu(X_n)]$.

However, we demonstrated in the previous section that we must include the correlation between the state and the model to accurately compute a trajectory, which means finding an approximation of (17), which might be nearly intractable.

Instead, we propose side-stepping the issue by approximating the entire GP model. Specifically, we propose a piecewise linear approximation of the GP, defining a series of linear ODEs, which allows us to exactly propagate a normal distribution though each piece. In other words, we approximate the model as a whole with a linearized model for which we can compute exact state distributions.

We linearize around ν_i and define

$$f_i(x) = a_i x + B_i, \quad X(0) = X_i,\ t \in [t_i, t_i + h), \tag{36}$$

where

$$a_i = \mu_f'(\nu_i), \tag{37}$$

$$B_i \sim \mathcal{N}\left(\mu_f(\nu_i) - a_i\nu_i,\, \sigma_f^2(\nu_i)\right), \tag{38}$$

similar to (7). While it would be more accurate to treat a_i as a random variable, similar to the *linGP* in Nghiem, 2019, this would introduce additional complexity via the product of two random variables and additional covariances.

This approximation allows us to define the PULL (Propagation of Uncertainty through Local Linearization) class of ODE solvers for continuous GP models of ODEs. However, it is currently not clear whether a similar approximation can be found for state-space GPs in order to incorporate the correlation with past states.

4.1 EXPLICIT EULER METHOD

Taking Euler steps on each linear sections results in the iterative scheme

$$\nu_{n+1} = \nu_n + h\mu_f(\nu_n) \tag{39a}$$

$$\begin{aligned} \Sigma_{n+1} =& (1 + a_n h)^2\Sigma_n + h^2\sigma_f^2(\nu_n) \\ &+ 2h(1 + a_n h)\,\mathrm{cov}(X_n, B_n). \end{aligned} \tag{39b}$$

We use a telescope sum as in (34) for the covariance in (39b) and obtain

$$\mathrm{cov}(X_n, B_n) = h\sum_{i=0}^{n-1}\prod_{j=i+1}^{n-1}(1 + a_j h)\,\mathrm{cov}(\nu_i, \nu_n). \tag{40}$$

This expression requires storing all previous a_i, but as we store all trajectory states X_i, the storage requirements for the solver already scale linearly with the number of steps taken. The bigger challenge is computing the covariances

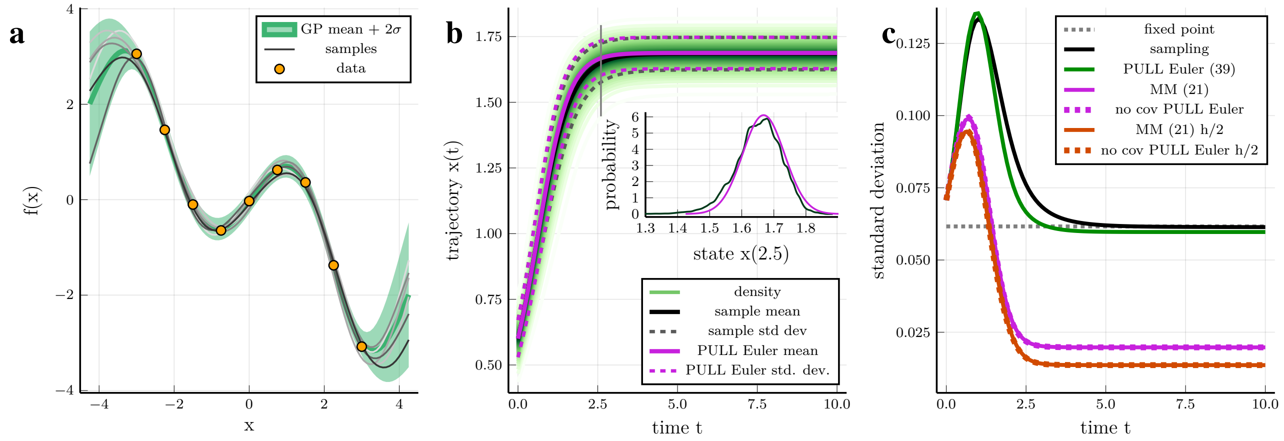

Figure 3: **Nonlinear example. a:** The example function, including the noisy data points, the mean and the variance of the GP and a few samples from the GP distribution. **b:** The distribution of the trajectories resulting from the GP samples, as well as their mean and standard deviation. We compare with the results from the approximate local linearization-based solver. **c:** The standard deviation resulting from sampling (ground truth), PULL Explicit Euler with full history, as well as the output approximation approach and PULL Euler without using any past points for two different step sizes each.

$\text{cov}(\nu_i, \nu_n)$ between the current state and all previous ones. This is done via (4b) which means that the complexity scales quadratically with the number of steps already taken.

To reduce this effort, there are two reasons to consider truncating the series in (40). Firstly, we often operate within the basin of attraction of a fixed point. Then it will generally be the case that $a_i < 0$ and $\prod_{j=i+1}^{n-1}(1+a_j h) \to 0$ for $n \to \infty$. Secondly, when using a stationary kernel for the GP, the covariance is determined by the distance of the two points, where distant points are assumed to be nearly uncorrelated, which means $\text{cov}(\nu_i, \nu_n) \approx 0$ for $i \ll n$.

Both terms have complementary behaviour. Under stable dynamics, successive x_n will be close and therefore correlated, but since we have $a_i < 0$ we can truncate based on the first term. Under unstable dynamics, we will have $a_i > 0$, but successive points will be further apart and the covariance term decreases faster.

As the terms of the series are simple to compute, it is possible to implement adaptive truncation or include adaptive step sizes. We also note that we have made no assumption about the underlying kernel beyond stationarity.

5 NUMERICAL EXPERIMENTS

In this section we illustrate the effectiveness of our PULL explicit Euler with a numerical example. [1]

5.1 A NONLINEAR EXAMPLE

We consider the ODE

$$\dot{x} = x\, cos(x), \tag{41}$$

and sample 9 data points in the interval $[-4, 4]$ and condition a GP with zero mean and the squared exponential kernel (5). Via (6), we generate 5000 samples (as seen in Fig. 3a) and integrate each one with initial values from 150 samples of the input distribution $\mathcal{N}(0.6, 0.07)$, resulting in an empirical distribution of trajectories. We also apply the PULL explicit Euler to the same GP, starting from the same initial distribution, and find in Fig. 3b that the results agree.

We also apply the moment-matching extension (21) of the output approximation by Girard et al., 2003 and Groot et al. Groot et al., 2011, using their expressions for (25) in the special case of the squared exponential kernel. The results differ substantially from the sampling-based ground truth, and match our solver when excluding past states (Fig. 3c).

We show the behaviour of moment matching and our PULL Euler in Tab. 1 with changing the step size, comparing with the prediction from sampling with a very high number of samples. A proper numerical solver should improve in accuracy with decreasing step size, which is the case for our PULL Euler (see also Fig. 4a-b), but for the moment-matching variance prediction the opposite happens.

Further, we note that our PULL explicit Euler method returns accurate results as the estimated trajectory distribution converges towards a model-dependent fixed point distribution. The approximation slightly underestimates variance compared to the ground truth sampling, likely due to linearization error and can potentiallys be improved upon.

[1] The results shown in this work were computed on an AMD Ryzen 3900, using up to 10 threads. The code and notebooksgl to reproduce these results can be found under `github.com/Crown421/the-past-does-matter-paper`

Table 1: **Convergence of the L1 error** (10^{-3})**:** We compare the L1 error of the mean and variance prediction of the PULL Euler (39) and moment matching (MM) (21) to the mean and variance obtained from a large number of samples (5000 GP samples and 200 samples of initial value distribution $\mathcal{N}(1.0, 0.07)$) for the time interval $(0.0, 3.0)$. Both approximate methods have the same error for the mean prediction, but for MM the variance prediction becomes worse with smaller step sizes. See also Fig. 2a and Fig. We also include the computation time for each step size.

Step Size	0.01	0.025	0.05	0.01	0.2
mean	**7.76**	8.80	10.53	13.99	20.85
var (PULL)	**0.63**	0.65	0.72	0.99	1.79
Time [ms]	14.4	2.8	1.1	0.53	0.35
var (MM)	25.20	24.40	18.94	13.98	**8.03**
Time [ms]	3.9	1.7	0.82	0.42	0.23

Tab. 1 also includes the computational cost of our PULL Euler, which is the low millisecond range using only a single thread as the algorithm is non-parallelizable. In Tab. 2 and 3 we study how the accuracy compared to a PULL Euler solution with a small step size and computational cost change with the number of samples used. The results are very promising, as using more samples reduces the error to the PULL Euler solution, while requires substantially more compute time, even while using 10 threads.

5.2 LIMITATIONS

The linearization-based approach detailed in this work has substantially lower computational cost than a sampling-based approach, but has a fundamental limitation. While linear models only have a single fixed point, nonlinear models introduce a variety of additional behaviours, such as multiple fixed points and complex basins of attraction.

In the model (41) from section 5.1, there is an unstable fixed point in 0 (see Fig. 3a). Trajectories starting to the right of it will converge towards $\pi/4$ and ones starting on the left of 0 will converge towards $-\pi/4$. Therefore, the distribution of trajectories starting from a distribution of initial values near 0 will be bi-modal, as shown in Fig. 4c. This behaviour cannot be captured by a linear approximation that assumes a uni-modal distribution for all states.

Still, Fig. 4c highlights the usefulness of our efficient approximate solver. Solving the deterministic ODE defined the GP mean would completely miss the presence and effects of the unstable fixed point. Similarly, a UniversalODE [Rackauckas et al., 2020] using a Neural Network would also be oblivious to the presence of this unstable fixed point.

Our solver converges to the right fixed point, as the initial value distribution has its mean to the right of 0, but the standard deviation shows substantially increased initial uncertainty compared to the previous result in Fig. 3b. This provides a strong hint of non-linear behaviour, which can be investigated further with a more expensive sampling-based method.

6 CONCLUSION

Accurately integrating dynamical systems expressed by GPs, either by numerically integrating a continuous vector field or subsequently applying a learned flow map, introduces a correlation between the GP distribution of functions and subsequent states. Using a linear prototype, we demonstrated that the approach in previous work incorrectly assumes independence between the distribution of the states and the GP, resulting in an underestimation of the trajectory variance.

We derived and illustrated the correct correlation for a simple linear model, and leveraged those findings to create a local linearization that can be combined with numerical integrator methods to create computationally efficient and accurate solvers, called PULL solvers.

A possible application for these solvers is incorporating them into a complete pipeline from data to predictions. We can create a likelihood-based cost function for training GP-based models with the predicted distribution of trajectories, as an alternative to multiple shooting methods [Hegde et al., 2022]. Once a model distribution is identified, we can make ahead-of-time predictions for model predictive control and take the prediction uncertainty into account when choosing a control strategy.

Our results are very promising and highlight a fundamental consideration for predicting trajectories for GP-learned dynamical systems, suggesting several extensions. To develop methods with higher convergence order, one could improve the piecewise linear approximation or combine it with higher-order integration methods.

Extending this work to higher dimensions introduces a non-zero correlation between the components of each state X_n and requires additional research. This correlation is not present in previous work due to using the independence assumption in (17).

Another interesting option is to combine our estimation of the model uncertainty with probabilistic numerics [Hennig et al., 2022]. This would quantify the total uncertainty introduced by both the model, due to insufficient data, and the solver algorithm.

Lastly, a more in-depth study of the effects of the independence assumption in the context of state-space GPs is needed, as a smaller step size of the flow map often implies a higher measurement frequency in the time series data. The resulting higher density of data already decreases the model uncertainty, and might partially mask the underestimation of the trajectory uncertainty .

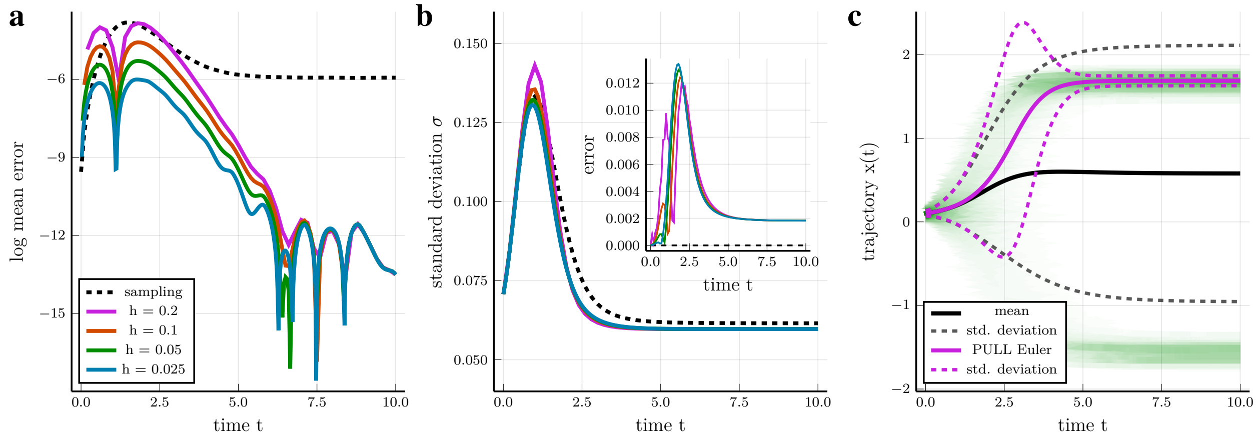

Figure 4: **PULL Euler Convergence and Limitations a:** The logarithmic error of the solver compared to the solution of the ODE defined by the GP mean. Decreasing the step size also decreases the mean error. **b:** The variance computed by our PULL solver is close to the sampling result. **c:** Near an unstable fixed point, the sampled distribution of trajectories bifurcates towards the two nearby stable fixed points, while the linear approximation can only capture a single one. However, the increased transient variance compared to Fig. 3b indicates hidden nonlinear dynamics.

Table 2: **Computation time and convergence of the L1 error** (10^{-2}) **with increased sampling for a deterministic initial value:** We compare the trajectory mean and variance predicted via sampling to the PULL Euler prediction with a small step size $h = 0.001$, for the deterministic initial value $x_0 = 1.0$ and the time interval $(0.0, 3.0)$. To improve the accuracy we increase the number of both the equidistance sampling locations (s. loc.) in (6) for a given GP sample, and the number of GP samples which produce one trajectory each. As sampling is non-deterministic, we repeat each prediction five times and also show the computation cost. We see that while the performance matches the PULL Euler, the computation cost is higher in all cases (cf. Tab. 1).

Mean	Number of GP samples		
s. loc.	500	1000	2000
4	1.39±3.9e-3	1.41±2.9e-3	1.40±1.6e-3
6	11.28±3.5e-3	11.30±1.6e-3	11.27±2.9e-3
8	1.32±2.1e-3	1.21±3.8e-3	1.25±1.0e-3
10	0.49±3.4e-3	0.45±2.3e-3	**0.44±2.3e-3**
Var	Number of GP samples		
s. loc.	500	1000	2000
4	1.45±2.9e-3	1.48±1.7e-3	1.51±2.7e-4
6	2.04±3.2e-3	2.09±3.2e-3	2.10±1.2e-3
8	1.59±2.9e-3	1.60±1.9e-3	1.57±5.8e-4
10	1.24±1.2e-3	**1.21±1.3e-3**	1.22±8.5e-4
Time [ms]	Number of GP samples		
s. loc.	500	1000	2000
4	63±54	93±42	186±59
6	65±50	113±58	196±58
8	61±46	93±47	212±5
10	40±0.7	94±48	175±53

Table 3: **Computation time and convergence of the L1 error** (10^{-3}) **with increased sampling for a initial value distribution:** We use the same setup as in Tab. 2, with the difference that we have an initial value distribution $x \sim \mathcal{N}(1.0, 0.7)$ that we sample an increasing number of times (in. s.) and the equidistance sampling locations are fixed to 11. Instead, we need to sample The error decreases with additional samples, apart from some outlier which are most likely caused by insufficient sampling. However, the computational cost is substantially higher than for the PULL Euler (cf. Tab. 1).

Mean	Number of GP samples		
in. s.	500	1000	2000
50	4.76±4.7e-2	1.1e-1	6.07±9.6e-2
100	8.1±7.5e-2	6.25±4.8e-2	**5.12±6.0e-2**
150	6.06±9.0e-2	8.08±5.4e-2	5.81±7.8e-2
Var	Number of GP samples		
in. s.	500	1000	2000
50	2.05±4.5e-2	1.4±5.7e-2	0.44±2.8e-2
100	0.55±5.5e-2	1.24±3.0e-2	0.51±4.7e-2
150	0.68±2.8e-2	1.45±3.8e-2	**0.39±3.4e-2**
Time [s]	Number of GP samples		
in. s.	500	1000	2000
50	1.8±0.14	3.6±0.03	7.1±0.09
100	3.6±0.05	7.2±0.08	14.4±0.11
150	5.3±0.09	10.8±0.13	21.7±0.08

Acknowledgements

S.R. acknowledges funding by the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines & Systems EP/L015897/1.

REFERENCES

Ridderbusch, S., Offen, C., Ober-Blöbaum, S., Goulart, P. (Dec. 2021). 'Learning ODE Models with Qualitative Structure Using Gaussian Processes'. 2021 60th IEEE Conference on Decision and Control (CDC).

Heinonen, M., Yildiz, C., Mannerström, H., Intosalmi, J., Lähdesmäki, H. (12th Mar. 2018). 'Learning Unknown ODE Models with Gaussian Processes'. arXiv: 1803.04303.

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A. (13th Jan. 2020). 'Universal Differential Equations for Scientific Machine Learning'. arXiv: 2001.04385.

Banks, H. T., Hu, S. (2012). *Uncertainty Propagation and Quantification in a Continuous Time Dynamical System*.

Guckenheimer, J., Holmes, P., Guckenheimer, J., Sirovich, L. (2013). *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. New York, NY: Springer.

Kamthe, S., Deisenroth, M. P. (20th June 2017). 'Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control'. arXiv: 1706.06491.

Buisson-Fenet, M., Solowjow, F., Trimpe, S. (10th–11th June 2020). 'Actively Learning Gaussian Process Dynamics'. *Proceedings of the 2nd Conference on Learning for Dynamics and Control*. Proceedings of Machine Learning Research. The Cloud: PMLR.

Girard, A., Rasmussen, C. E., Murray-Smith, R. (2003). 'Multiple-Step Ahead Prediction for Non Linear Dynamic Systems – A Gaussian Process Treatment with Propagation of the Uncertainty'. *Advances in Neural Information Processing Systems*.

Groot, P., Lucas, P., Bosch, P. (2011). 'Multiple-Step Time Series Forecasting with Sparse Gaussian Processes'. *http://allserv.kahosl.be/bnaic2011/*.

Kocijan, J. (2015). *Modelling and Control of Dynamic Systems Using Gaussian Process Models*. Advances in Industrial Control. Springer International Publishing.

Nghiem, T. X. (1st Oct. 2019). 'Linearized Gaussian Processes for Fast Data-driven Model Predictive Control'. arXiv: 1812.10579 [cs].

Girard, A., Rasmussen, C. E., Candela, J. Q., Murray-Smith, R. (2002). 'Gaussian Process Priors with Uncertain Inputs Application to Multiple-Step Ahead Time Series Forecasting'. *Advances in Neural Information Processing Systems*.

Hewing, L., Arcari, E., Fröhlich, L. P., Zeilinger, M. N. (10th–11th June 2020). 'On Simulation and Trajectory Prediction with Gaussian Process Dynamics'. *Proceedings of the 2nd Conference on Learning for Dynamics and Control*. Proceedings of Machine Learning Research. The Cloud: PMLR.

Hegde, P., Yıldız, Ç., Lähdesmäki, H., Kaski, S., Heinonen, M. (17th Aug. 2022). 'Variational Multiple Shooting for Bayesian ODEs with Gaussian Processes'. *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. Uncertainty in Artificial Intelligence. PMLR.

Schober, M., Duvenaud, D., Hennig, P. (24th Oct. 2014). 'Probabilistic ODE Solvers with Runge-Kutta Means'. arXiv: 1406.2582 [cs, math, stat].

Hennig, P., Osborne, M. A., Kersting, H. P. (2022). *Probabilistic Numerics: Computation as Machine Learning*. Cambridge: Cambridge University Press.

Rasmussen, C. E., Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press. 248 pp.

Alvarez, M. A., Rosasco, L., Lawrence, N. D. (16th Apr. 2012). 'Kernels for Vector-Valued Functions: A Review'. arXiv: 1106.6251 [cs, math, stat].

Wilson, J. T., Borovitskiy, V., Terenin, A., Mostowsky, P., Deisenroth, M. P. (16th Aug. 2020). *Efficiently Sampling Functions from Gaussian Process Posteriors*. arXiv: 2002.09309 [cs, stat]. preprint.

Rahimi, A., Recht, B. (2007). 'Random Features for Large-Scale Kernel Machines'. *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Ensinger, K., Solowjow, F., Ziesche, S., Tiemann, M., Trimpe, S. (9th Jan. 2022). 'Structure-Preserving Gaussian Process Dynamics'. arXiv: 2102.01606 [cs].

Ialongo, A. D., van der Wilk, M., Hensman, J., Rasmussen, C. E. (13th June 2019). 'Overcoming Mean-Field Approximations in Recurrent Gaussian Process Models'. arXiv: 1906.05828 [cs, stat].