Constraining Gaussian Processes to Systems of Linear Ordinary Differential Equations

Andreas Besginow
Department of Electrical Engineering and Computer Science
OWL University of Applied Sciences and Arts Lemgo, Germany
Institute industrial IT Lemgo, Germany
andreas.besginow@th-owl.de

Markus Lange-Hegermann
Department of Electrical Engineering and Computer Science
OWL University of Applied Sciences and Arts Lemgo, Germany
Institute industrial IT Lemgo, Germany
markus.lange-hegermann@th-owl.de

Abstract

Data in many applications follows systems of Ordinary Differential Equations (ODEs). This paper presents a novel algorithmic and symbolic construction for covariance functions of Gaussian Processes (GPs) with realizations strictly following a system of linear homogeneous ODEs with constant coefficients, which we call LODE-GPs. Introducing this strong inductive bias into a GP improves modelling of such data. Using smith normal form algorithms, a symbolic technique, we overcome two current restrictions in the state of the art: (1) the need for certain uniqueness conditions in the set of solutions, typically assumed in classical ODE solvers and their probabilistic counterparts, and (2) the restriction to controllable systems, typically assumed when encoding differential equations in covariance functions. We show the effectiveness of LODE-GPs in experiments.

1 Introduction

Many real world tasks have underlying dynamic behavior, for example chemical reactions Goeke et al. [2012], systems in bioprocess engineering Hernández Rodríguez et al. [2022], or population dynamics Wangersky [1978]. Many such systems are linear or can be decently linearized, such as in control theory Zerz [2000], biology De Hoon et al., process engineering [Adkins and Davidson, 2012, §9], or engines Bertin et al. [2000]. Including prior knowledge in the form of differential equations benefits the model fit and enhances interpretability for a model. Hence, modelling differential equations has therefore been the focus of much research in Deep Learning (DL) (e.g. Hochlehnert et al. [2021], Raissi et al. [2019], Lagaris et al. [2000], van Milligen et al. [1995], Drygala et al. [2022]). While the performance of DLs models is good, they usually lack the ability to perform uncertainy estimates as many don't have a probabilistic nature and can't guarantee to strictly satisfy the equations. Whereas the first point is inherent in Gaussian Processes (GPs) and the introduction of physical information has been focus of much research in that area as well (e.g. Lange-Hegermann [2018], Jidling et al. [2017], Särkkä [2011], Gonçalves et al. [2021], Ulaganathan et al. [2016]). In particular Lange-Hegermann [2018] showed the hidden assumption of previous works, which limited them to controllable systems, by building a mathematical foundation.

Other probabilistic models are often based on, more classical, methods like Runge-Kutta or Kalman filters Schmidt et al. [2021], Schober et al. [2019, 2014], Bosch et al. [2021] and often just estimate

2022 NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems.

the solution of Ordinary Differential Equation (ODE) initial value problems through approximation, thereby not strictly guaranteeing to yield solutions of the ODE. This class of algorithms is commonly used to solve non-linear ODEs, but typically require that systems are well-posed, in particular the solution needs to be unique once a finite number of initial conditions is known. While this second limitation is typically irrelevant in physical or biological systems, it is strongly relevant in systems in engineering, such as in control systems with their freely choosable inputs.

For GP priors for decoupled ODE systems with constant coefficients and right hand side functions see Alvarez et al. [2009], which considers the right hand side functions as latent, hence these models are called Latent Force Model (LFM). A GP prior on these latent forces is assumed and is pushed forward through differential operators and Green's operator.

This paper overcomes the necessity of approximations, the restriction to systems where initial conditions lead to unique solutions, and the restriction to controllable systems. For that purpose, we algorithmically construct LODE-GPs, a novel class of GPs whose realizations strictly follow a given system of homogenuous linear ODEs with constant coefficients.

We sketch our approach and successfully test it on systems of differential equations. The LODE-GP strictly satisfies the ODEs and outperforms GPs by several magnitudes in its precision. We also discuss a brief comparison against the LFM models.

2 Constructing a GP for differential equations

We introduce LODE-GPs, a class of GPs with realizations dense in the space of solutions of linear ODEs with constant coefficients. This ensures that LODE-GPs produce all possible solutions to the ODEs and nothing but solutions for the ODEs. All this is guaranteed to be strictly accurate.

Our construction uses the application of an operator matrix, which formally is the pushforward operation of an operator matrix B on the GP g as $B_*g = \mathcal{GP}(B\mu(x), Bk(x,x')(B')^T)$ Berlinet and Thomas-Agnan [2011], where B' denotes the operation of B on the second argument of $k(x,x')$ [Lange-Hegermann, 2021, Lemma 2.2]. The matrix B induces the strong bias such that all realizations lie in the image of B. This pushforward is typical for applications in differential equations Jidling et al. [2017], Lange-Hegermann [2018, 2021] or geometry Hutchinson et al. [2021].

Further we use the Smith Normal Form (SNF) Smith [1862], Newman [1997], which is a normal form for a matrix $A \in \mathbb{R}[x]^{m\times n}$ over polynomial ring $\mathbb{R}[x]$, s.t. $U \cdot A \cdot V = D$. Here, $D \in \mathbb{R}[x]^{m\times n}$ is a diagonal matrix of same size as A and base change matrices $U \in \mathbb{R}[x]^{m\times m}$ and $V \in \mathbb{R}[x]^{n\times n}$ are invertible square matrices, i.e. $\det(U), \det(V) \in \mathbb{R}\backslash\{0\}$ Göllmann [2017], Cluzeau et al. [2011]. Algorithms to construct the SNF are implemented in computer algebra systems such as e.g. Maple Maplesoft, a division of Waterloo Maple Inc.. and SageMath The Sage Developers [2021], a free and open source Python library for computer algebra. Since the SNF exists for matrices over polynomial rings over any field, we can compute it over a polynomial ring over the function field $\mathbb{R}(a_1, \dots, a_k)$. Hence, we can model differential equations containing parameters $a_1, \dots, a_k$.

Consider a system of linear homogenous ordinary differential equations with constant coefficients

$$A \cdot \mathbf{f}(t) = 0 \tag{1}$$

with operator matrix $A \in \mathbb{R}[\partial_t]^{m\times n}$ determining the relationship between the smooth functions $f_i(t) \in C^\infty(\mathbb{R},\mathbb{R})$ of $\mathbf{f}(t) = \begin{pmatrix} f_1(t) & \dots & f_n(t) \end{pmatrix}^T$. For such systems our main result holds.

Theorem 1. *(LODE-GPs) For every system as in Equation* (1) *there exists a GP g, such that the set of realizations of g is dense in the set of solutions of $A \cdot \mathbf{f}(t) = 0$.* [1]

Our construction creates a GP as $g \sim V_*h = \mathcal{GP}(\mathbf{0}, V \cdot k \cdot V')$, with h a suitable GP prior, V a base change matrix from the SNF and $V' = V^T$ the operation applied on the second kernel entry.

3 Experimental evaluation

We demonstrate the effectiveness of LODE-GPs, by constraining a LODE-GP using a three tank system and train it on 25 evenly spread points in $[1, 6]$, similar to solving an initial value problem.

[1]For the proof of the theorem we refer the reader to the accepted full paper version of this work.

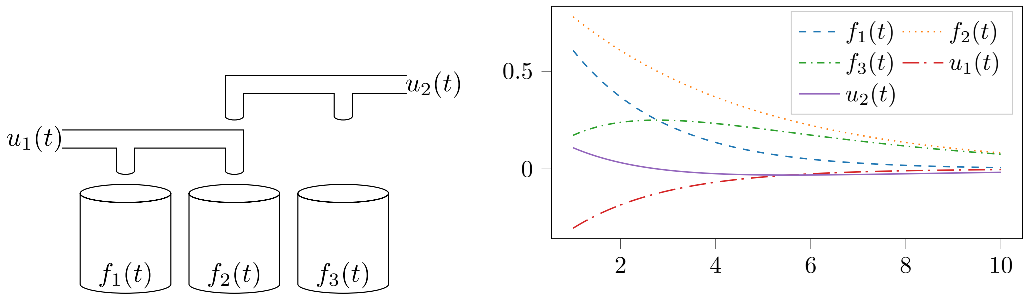

Figure 1: (left) A sketch of the three tank system and (right) a solution of the system.

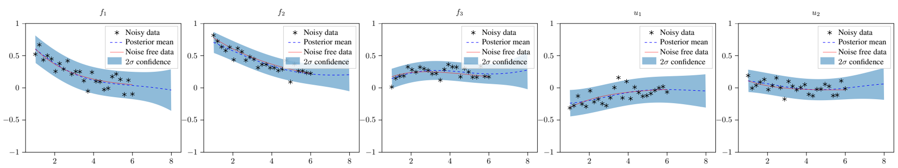

Figure 2: The posterior LODE-GP models for the three tank system, trained on noisy data. The black stars indicate the noisy datapoints, the red line is the solution to the ODEs, the blue dashed line is the LODE-GPs posterior mean, the transparent blue area is the 2σ confidence interval.

The approach of Lange-Hegermann [2018] is not applicable to this uncontrollable example. The only other method that can deal with our class of differential equations is Alvarez et al. [2009][2], and is discussed in Section 3.2, in the following we compare our model mainly to classic GPs.

Our comparison includes the error in satisfying the ODEs, specified by the median error the GPs posterior mean function has in satisfying the ODEs at evenly spread points, where we calculate derivatives through finite differences. Training and evaluation is repeated ten times in each experiment using a GPyTorch Gardner et al. [2018] implementation of our LODE-GP construction with SageMath The Sage Developers [2021] to symbolically calculate the SNF.

3.1 Three tank system - Non-controllable

We use a non-controllable fluid system where the water level in three tanks is changed by two pipes. The system is non-controllable due to pipes' overlap over the center tank, whose changes directly affect the other tanks. This system requires multiple non-zero covariance functions in the latent GP to describe both the non-controllable subsystem and the two degrees of freedom.

We use the following solution to the system of ODEs (see Figure 1) to generate 25 datapoints, to which we add white noise with standard deviation of 10% of the maximal signal.

$$\begin{bmatrix} f_1(t) \\ f_2(t) \\ f_3(t) \\ u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} \exp(-\frac{t}{2}) \\ \exp(-\frac{t}{4}) \\ \exp(-\frac{t}{4}) - \exp(-\frac{t}{2}) \\ -\frac{\exp(-\frac{t}{2})}{2} \\ -\frac{\exp(-\frac{t}{4})}{4} + \frac{\exp(-\frac{t}{2})}{2} \end{bmatrix} \tag{2}$$

Training is repeated 20 times for 300 iterations using adam Kingma and Ba [2015], which resulted in a median loss of -0.974, similarly to a standard GP which had a loss of -0.949. The main difference is the error in satisfying the ODE via finite differences to estimate derivatives, where the LODE-GP scored a median error of $1\mathrm{e}{-5}$, whereas the GP had an error of 0.040, which is several magnitudes higher. This comparison shows that the LODE-GP, as proclaimed, strictly follows the given ODE. Further, this shows that it can handle large, non-controllable systems of ODEs, despite high noise.

[2]Most probabilistic ODE solvers are not applicable to systems with free functions in their solution set, with the exception of Schmidt et al. [2021], which can only estimate free functions of parameters.

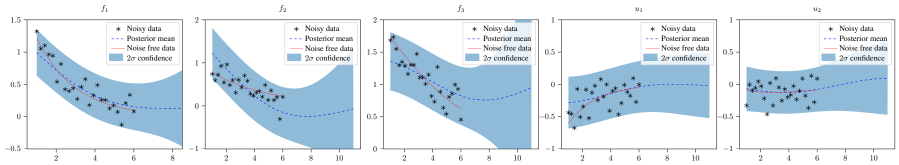

Figure 3: The posterior LODE-GP models for the three tank system, trained on noisy data. The black stars indicate the noisy datapoints, the red line is the solution to the ODEs, the blue dashed line is the LODE-GPs posterior mean, the transparent blue area is the 2σ confidence interval.

3.2 Three tank system - Comparison to LFM

We compare our LODE-GP with the LFM introduced by Alvarez et al. [2009]. To do so we set the variables from Equation 3 in Alvarez et al. [2009] as follows: $D_q = 0$, $B_q = 0$, $S_{rq} = \begin{bmatrix} -1 & -1 & 0 \\ 0 & -1 & -1 \end{bmatrix}^T$. We calculate the covariance function as the solution of the integral $k = \int_0^{t_2} \int_0^{t_1} \exp^{-(x_1 - x_2)^2} dx_1 dx_2$, effectively setting the parameters $\ell = 1$ and $\sigma = 1$. Following the steps of Alvarez et al. [2009], we get a GP that can estimate the solution of the three tank ODEs.

The resulting GP marginalizes the (in their model considered) latent function u_1 and u_2 and only considered the three data channels f_1, f_2, and f_3. To calculate the ODE error, we inserted the original data for the channels u_1 and u_2 into the calculation. For a fair comparison with our LODE-GP, we also marginalize u_1 and u_2 there. Similarly as for the LFM, we have set $\ell = 1$ and $\sigma = 1$.

The resulting ODE errors of the two models are as shown in Table 1. The performance of the marginalized LODE-GP is better but comparable to the LFM. The performance of the full LODE-GP, having learned all 5 channels and also set all lengthscales and signal variances to 1, shows a signifcant increase in performance. Thus we conclude our LODE-GP is at least as good as the LFM.

Table 1: The ODE error of the LFM, the small LODE-GP, and the full LODE-GP. Smaller is better.

	ODE 1 error	ODE 2 error	ODE 3 error
(marginalized) LFM	0.042816	0.507017	0.013084
marginalized LODE-GP	0.031331	0.025065	0.008595
full LODE-GP	1.210e-05	1.285e-05	1.081e-05

3.3 Three tank system - Non-system data

In an ablation we train on data which is not a solution to the ODEs. We take the system solution and modify the first coefficient of most equations up to factors of 3. This changes the ODE error of the generated data from $1.33\mathrm{e}{-05}$ to 0.098. As above, we also add 10% noise for training.

Despite the ODE error of the non-system training data, the LODE-GP still strictly follows the ODE with an ODE error of $9.64\mathrm{e}{-05}$. But we observe that the median loss over 20 full trainings is relatively high with 0.025. From the posterior plot (see Figure 3) we can see how much the mean of the LODE-GP deviates from the noise-free training data. When we compare this to the loss for the LODE-GP on the actual system data, we see that even a small deviation from the system's solution leads to drastically worse training results. From this we conclude that the LODE-GP is useful to recognize non-system data via the corresponding training loss, despite noise as high as 10%.

4 Conclusion

In this paper we have introduced LODE-GPs, a class of covariance functions for GPs such that their realizations strictly follow a given system of linear homogeneous ODEs, up to numerical precision, which we demonstated in examples. Additionally, we performed an ablation to show the potential to detect data that doesn't satisfy the given ODEs, with just 25 datapoints.

Acknowledgments and Disclosure of Funding

This research was supported by the research training group "Dataninja" (Trustworthy AI for Seamless Problem Solving: Next Generation Intelligence Joins Robust Data Analysis) funded by the German federal state of North Rhine-Westphalia.

References

William A Adkins and Mark G Davidson. Linear systems of differential equations. In *Ordinary Differential Equations*, pages 629–721. 2012.

Mauricio Alvarez, David Luengo, and Neil D Lawrence. Latent force models. In *Artificial Intelligence and Statistics*, pages 9–16. PMLR, 2009.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

Yves Bertin, Etienne Videcoq, Sophie Thieblin, and Daniel Petit. Thermal behavior of an electrical motor through a reduced model. *IEEE Transactions on energy conversion*, 15(2):129–134, 2000.

Nathanael Bosch, Philipp Hennig, and Filip Tronarp. Calibrated adaptive probabilistic ODE solvers. In *AISTATS*, 2021.

Thomas Cluzeau, Victorita Dolean, Frédéric Nataf, and Alban Quadrat. Symbolic preconditioning techniques for linear systems of partial differential equations. 2011. URL `https://hal.archives-ouvertes.fr/hal-00664092`.

Michiel JL De Hoon, Seiya Imoto, Kazuo Kobayashi, Naotake Ogasawara, and Satoru Miyano. Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations. In *Biocomputing 2003*, pages 17–28.

Claudia Drygala, Benjamin Winhart, Francesca di Mare, and Hanno Gottschalk. Generative modeling of turbulence. *Physics of Fluids*, 34(3):035114, 2022.

Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

Alexandra Goeke, Christian Schilli, Sebastian Walcher, and Eva Zerz. Computing quasi-steady state reductions. *Journal of Mathematical Chemistry*, 50(6):1495–1513, 2012.

Laurenz Göllmann. *Lineare Algebra*. Springer, 2017.

Ítalo Gomes Gonçalves, Felipe Guadagnin, Sissa Kumaira, and Saulo Lopes Da Silva. A machine learning model for structural trend fields. *Computers & Geosciences*, 149:104715, 2021.

Tanja Hernández Rodríguez, Anton Sekulic, Markus Lange-Hegermann, and Björn Frahm. Designing robust biotechnological processes regarding variabilities using multi-objective optimization applied to a biopharmaceutical seed train design. *Processes*, 10(5):883, 2022.

Andreas Hochlehnert, Alexander Terenin, Steindor Saemundsson, and Marc Deisenroth. Learning contact dynamics using physically structured neural networks. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2152–2160. PMLR, 13–15 Apr 2021.

Michael Hutchinson, Alexander Terenin, Viacheslav Borovitskiy, So Takao, Yee Teh, and Marc Deisenroth. Vector-valued Gaussian processes on Riemannian manifolds via gauge independent projected kernels. *NeurIPS 2021*, 34, 2021.

Carl Jidling, Niklas Wahlström, Adrian Wills, and Thomas B Schön. Linearly constrained Gaussian processes. In *NeurIPS 2017*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

Isaac E Lagaris, Aristidis C Likas, and Dimitris G Papageorgiou. Neural-network methods for boundary value problems with irregular boundaries. *IEEE Transactions on Neural Networks*, 11 (5):1041–1049, 2000.

Markus Lange-Hegermann. Algorithmic linearly constrained Gaussian processes. In *NeurIPS 2018*, pages 2141–2152, 2018.

Markus Lange-Hegermann. Linearly constrained Gaussian processes with boundary conditions. In *AISTATS*, 2021.

Maplesoft, a division of Waterloo Maple Inc.. Maple.

Morris Newman. The Smith normal form. *Linear algebra and its applications*, 254(1-3):367–381, 1997.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Simo Särkkä. Linear operators and stochastic partial differential equations in Gaussian process regression. In *International Conference on Artificial Neural Networks*, pages 151–158, 2011.

Jonathan Schmidt, Nicholas Krämer, and Philipp Hennig. A probabilistic state space model for joint inference from differential equations and data. *NeurIPS 2021*, 34, 2021.

Michael Schober, David K Duvenaud, and Philipp Hennig. Probabilistic ODE solvers with Runge-Kutta means. *NeurIPS 2014*, 27, 2014.

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

Henry John Stephen Smith. I. on systems of linear indeterminate equations and congruences. *Proceedings of the Royal Society of London*, (11):86–89, 1862.

The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.4)*, 2021. available from `https://www.sagemath.org`.

Selvakumar Ulaganathan, Ivo Couckuyt, Tom Dhaene, Joris Degroote, and Eric Laermans. Performance study of gradient-enhanced Kriging. *Engineering with computers*, 32(1):15–34, 2016.

B Ph van Milligen, V Tribaldos, and JA Jiménez. Neural network differential equation and plasma equilibrium solver. *Physical review letters*, 75(20):3594, 1995.

Peter J Wangersky. Lotka-volterra population models. *Annual Review of Ecology and Systematics*, 9(1):189–218, 1978.

Eva Zerz. *Topics in Multidimensional Linear Systems Theory*, volume 256 of *Lecture Notes in Control and Information Sciences*. London, 2000.