The Journal of Open Source Software

ProbNumDiffEq.jl: Probabilistic Numerical Solvers for Ordinary Differential Equations in Julia

Nathanael Bosch[1]

1 Tübingen AI Center, University of Tübingen, Germany

DOI: 10.21105/joss.07048

Software

- Review
- Repository
- Archive

Editor: Mehmet Hakan Satman

Reviewers:

- @PieterjanRobbe
- @ranocha

Submitted: 19 July 2024
Published: 30 September 2024

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

Probabilistic numerical solvers have emerged as an efficient framework for simulation, uncertainty quantification, and inference in dynamical systems. In comparison to traditional numerical methods, which approximate the true trajectory of a system only by a single point estimate, probabilistic numerical solvers compute a *distribution* over the true unknown solution of the given differential equation and thereby provide information about the numerical error incurred during the computation. ProbNumDiffEq.jl implements such probabilistic numerical solvers for ordinary differential equations (ODEs) and differential-algebraic equations (DAEs) in the Julia programming language (Bezanson et al., 2017) within the DifferentialEquations.jl ecosystem (Rackauckas & Nie, 2017).

More concretely, ProbNumDiffEq.jl provides a range of probabilistic numerical solvers for ordinary differential equations based on Bayesian filtering and smoothing, which have emerged as a particularly efficient and flexible class of methods for solving ODEs (Kersting, Sullivan, et al., 2020; Schober et al., 2019; Tronarp et al., 2019). These so-called "ODE filters" have known polynomial convergence rates (Kersting, Sullivan, et al., 2020; Tronarp et al., 2021) and numerical stability properties (such as A-stability or L-stability) (Bosch, Hennig, et al., 2023; Tronarp et al., 2019), their computational complexity is comparable to traditional numerical methods (Krämer, Bosch, et al., 2022), they are applicable to a range of numerical differential equation problems (Bosch et al., 2022; Krämer, Schmidt, et al., 2022; Krämer & Hennig, 2021), and they can be formulated parallel-in-time (Bosch, Corenflos, et al., 2023). ODE filters also provide a natural framework for ODE parameter inference (Beck et al., 2024; Kersting, Krämer, et al., 2020; Schmidt et al., 2021; Tronarp et al., 2022). ProbNumDiffEq.jl implements many of the methods referenced above and packages them in a software library with the aim to be easy-to-use, feature-rich, well-documented, and efficiently implemented.

Statement of need

Filtering-based probabilistic numerical ODE solvers have been an active field of research for the past decade, but their application in practical simulation and inference problems has been limited. ProbNumDiffEq.jl aims to bridge this gap. ProbNumDiffEq.jl implements probabilistic numerical methods as performant, documented, and easy-to-use ODE solvers inside the well-established DifferentialEquations.jl ecosystem (Rackauckas & Nie, 2017). Thereby, the package benefits from the extensive testing, documentation, performance optimization, and functionality that DifferentialEquations.jl provides. Users can easily find help and examples regarding many features that are not particular to ProbNumDiffEq.jl in the DifferentialEquations.jl documentation, and we provide additional examples and tutorials specific to the probabilistic solvers in the ProbNumDiffEq.jl documentation. We believe that this deep integration within DifferentialEquations.jl is a key feature to attract users to probabilistic numerics by enabling the use of probabilistic ODE solvers as drop-in replacements for traditional ODE solvers.

On the other hand, ProbNumDiffEq.jl also aims to accelerate the development of new probabilistic numerical ODE solvers by providing a solid foundation to both build on and compare against. Several publications have been developed with ProbNumDiffEq.jl, including contributions on step-size adaptation and calibration of these solvers (Bosch et al., 2021), energy-preserving solvers and DAE solvers (Bosch et al., 2022), probabilistic exponential integrators (Bosch, Hennig, et al., 2023), and novel parameter inference algorithms (Beck et al., 2024; Tronarp et al., 2022). We also hope that by providing documented and performant implementations of published algorithms, we facilitate researchers to use these methods as baselines when developing new numerical solvers.

ProbNumDiffEq.jl is also the only software package in Julia, at the time of writing, that provides a comprehensive set of probabilistic numerical ODE solvers. Outside of Julia, two other software packages provide a similar functionality. ProbNum (Wenger et al., 2021) is a Python package that implements probabilistic numerical methods for various numerical problems, including linear systems, quadrature, and ODEs. ProbNum particularly aims to facilitate rapid experimentation and accelerate the development of new methods (Wenger et al., 2021). It is therefore broader in scope and provides functionality not covered by ProbNumDiffEq.jl, but it also lacks some of the specialized ODE solvers available in ProbNumDiffEq.jl. In addition, with its reliance on Python and NumPy (Harris et al., 2020) and the lack of just-in-time compilation, it is also generally less performant. ProbDiffEq (Krämer, 2023) is a probabilistic numerical ODE solver package built on JAX. At the time of writing, it provides a very similar set of ODE solvers as ProbNumDiffEq.jl with the addition of certain filtering and smoothing methods and the lack of certain specialized ODE solvers—but as both ProbDiffEq and ProbNumDiffEq.jl are under active development, this might change in the future. By building on JAX and leveraging its just-in-time compilation capabilities, ProbDiffEq provides ODE solvers with similar performance as those implemented in ProbNumDiffEq.jl (shown through benchmarks in both packages comparing to SciPy (Virtanen et al., 2020)). In summary, ProbNumDiffEq.jl provides one of the most feature-rich and performant probabilistic numerical ODE solver packages currently available and is the only one built on the Julia programming language.

Acknowledgements

The author gratefully acknowledges co-funding by the European Union (ERC, ANUBIS, 101123955. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them). The author thanks the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for their support.

I am grateful to Philipp Hennig for his support throughout the development of this package. I thank Nicholas Krämer, Filip Tronarp, and Jonathan Schmidt for many valuable discussions about probabilistic numerical ODE solvers and their correct, efficient, and elegant implementation. I also thank Christopher Rackauckas for support and feedback on how to integrate ProbNumDiffEq.jl with the DifferentialEquations.jl ecosystem and for including ProbNumDiffEq.jl into the testing pipeline of OrdinaryDiffEq.jl. I acknowledge contributions from Pietro Monticone (@pitmonticone), Vedant Puri (@vpuri3), Tim Holy (@timholy), Daniel González Arribas (@DaniGlez), David Widmann (@devmotion), Christopher Rackauckas (@ChrisRackauckas), Qingyu Qu (@ErikQQY), Cornelius Roemer (@corneliusroemer), and Jose Storopoli (@storopoli).

References

Beck, J., Bosch, N., Deistler, M., Kadhim, K. L., Macke, J. H., Hennig, P., & Berens, P. (2024). Diffusion tempering improves parameter estimation with probabilistic integrators for ordinary differential equations. *Forty-First International Conference on Machine Learning*. https://openreview.net/forum?id=43HZG9zwaj

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

Bosch, N., Corenflos, A., Yaghoobi, F., Tronarp, F., Hennig, P., & Särkkä, S. (2023). *Parallel-in-time probabilistic numerical ODE solvers*. https://doi.org/10.48550/arXiv.2310.01145

Bosch, N., Hennig, P., & Tronarp, F. (2021). Calibrated adaptive probabilistic ODE solvers. In A. Banerjee & K. Fukumizu (Eds.), *Proceedings of the 24th international conference on artificial intelligence and statistics* (Vol. 130, pp. 3466–3474). PMLR. http://proceedings.mlr.press/v130/bosch21a.html

Bosch, N., Hennig, P., & Tronarp, F. (2023). Probabilistic exponential integrators. *Thirty-Seventh Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=2dx5MNs2Ip

Bosch, N., Tronarp, F., & Hennig, P. (2022). Pick-and-mix information operators for probabilistic ODE solvers. In G. Camps-Valls, F. J. R. Ruiz, & I. Valera (Eds.), *Proceedings of the 25th international conference on artificial intelligence and statistics* (Vol. 151, pp. 10015–10027). PMLR. https://proceedings.mlr.press/v151/bosch22a.html

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Kersting, H., Krämer, N., Schiegg, M., Daniel, C., Tiemann, M., & Hennig, P. (2020). Differentiable likelihoods for fast inversion of 'Likelihood-free' dynamical systems. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (Vol. 119, pp. 5198–5208). PMLR. http://proceedings.mlr.press/v119/kersting20a.html

Kersting, H., Sullivan, T. J., & Hennig, P. (2020). Convergence rates of Gaussian ODE filters. *Statistics and Computing*, *30*(6), 1791–1816. https://doi.org/10.1007/s11222-020-09972-4

Krämer, N. (2023). Probdiffeq: Probabilistic solvers for differential equations in JAX. In *GitHub repository*. GitHub. https://github.com/pnkraemer/probdiffeq

Krämer, N., Bosch, N., Schmidt, J., & Hennig, P. (2022). Probabilistic ODE solutions in millions of dimensions. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, & S. Sabato (Eds.), *Proceedings of the 39th international conference on machine learning* (Vol. 162, pp. 11634–11649). PMLR. https://proceedings.mlr.press/v162/kramer22b.html

Krämer, N., & Hennig, P. (2021). Linear-time probabilistic solution of boundary value problems. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems* (Vol. 34, pp. 11160–11171). Curran Associates, Inc. https://papers.nips.cc/paper/2021/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

Krämer, N., Schmidt, J., & Hennig, P. (2022). Probabilistic numerical method of lines for time-dependent partial differential equations. In G. Camps-Valls, F. J. R. Ruiz, & I. Valera (Eds.), *Proceedings of the 25th international conference on artificial intelligence and statistics* (Vol. 151, pp. 625–639). PMLR. https://proceedings.mlr.press/v151/kramer22a.html

Rackauckas, C., & Nie, Q. (2017). DifferentialEquations.jl – A performant and feature-rich ecosystem for solving differential equations in Julia. *Journal of Open Research Software*, *5*(1). https://doi.org/10.5334/jors.151

Schmidt, J., Krämer, N., & Hennig, P. (2021). A probabilistic state space model for joint inference from differential equations and data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems* (Vol. 34, pp. 12374–12385). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2021/file/6734fa703f6633ab896eecbdfad8953a-Paper.pdf

Schober, M., Särkkä, S., & Hennig, P. (2019). A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing*, *29*(1), 99–122. https://doi.org/10.1007/s11222-017-9798-7

Tronarp, F., Bosch, N., & Hennig, P. (2022). Fenrir: Physics-enhanced regression for initial value problems. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, & S. Sabato (Eds.), *Proceedings of the 39th international conference on machine learning* (Vol. 162, pp. 21776–21794). PMLR. https://proceedings.mlr.press/v162/tronarp22a.html

Tronarp, F., Kersting, H., Särkkä, S., & Hennig, P. (2019). Probabilistic solutions to ordinary differential equations as nonlinear Bayesian filtering: A new perspective. *Statistics and Computing*, *29*(6), 1297–1315. https://doi.org/10.1007/s11222-019-09900-1

Tronarp, F., Särkkä, S., & Hennig, P. (2021). Bayesian ODE solvers: The maximum a posteriori estimate. *Statistics and Computing*, *31*(3), 23. https://doi.org/10.1007/s11222-021-09993-7

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wenger, J., Krämer, N., Pförtner, M., Schmidt, J., Bosch, N., Effenberger, N., Zenn, J., Gessner, A., Karvonen, T., Briol, F.-X., Mahsereci, M., & Hennig, P. (2021). *ProbNum: Probabilistic numerics in Python*. https://doi.org/10.48550/arXiv.2112.02100