The Journal of Open Source Software

PySINDy: A comprehensive Python package for robust sparse system identification

Alan A. Kaptanoglu[1], Brian M. de Silva[2], Urban Fasel[3], Kadierdan Kaheman[3], Andy J. Goldschmidt[1], Jared Callaham[3], Charles B. Delahunt[2], Zachary G. Nicolaou[2], Kathleen Champion[2], Jean-Christophe Loiseau[4], J. Nathan Kutz[2], and Steven L. Brunton[3]

1 Department of Physics, University of Washington **2** Department of Applied Mathematics, University of Washington **3** Department of Mechanical Engineering, University of Washington **4** Arts et Métiers Institute of Technology, CNAM, DynFluid, HESAM Université

DOI: 10.21105/joss.03994

Software

- Review
- Repository
- Archive

Editor: Sebastian Benthall

Reviewers:

- @henrykironde
- @tuelwer

Submitted: 21 October 2021
Published: 29 January 2022

Summary

Automated data-driven modeling, the process of directly discovering the governing equations of a system from data, is increasingly being used across the scientific community. `PySINDy` is a Python package that provides tools for applying the sparse identification of nonlinear dynamics (SINDy) approach to data-driven model discovery. In this major update to `PySINDy`, we implement several advanced features that enable the discovery of more general differential equations from noisy and limited data. The library of candidate terms is extended for the identification of actuated systems, partial differential equations (PDEs), and implicit differential equations. Robust formulations, including the integral form of SINDy and ensembling techniques, are also implemented to improve performance for real-world data. Finally, we provide a range of new optimization algorithms, including several sparse regression techniques and algorithms to enforce and promote inequality constraints and stability. Together, these updates enable entirely new SINDy model discovery capabilities that have not been reported in the literature, such as constrained PDE identification and ensembling with different sparse regression optimizers.

Statement of need

Traditionally, the governing laws and equations of nature have been derived from first principles and based on rigorous experimentation and expert intuition. In the modern era, cheap and efficient sensors have resulted in an unprecedented growth in the availability of measurement data, opening up the opportunity to perform automated model discovery using data-driven modeling. These data-driven approaches are also increasingly useful for processing and interpreting the information in these large datasets. A number of such approaches have been developed in recent years, including the dynamic mode decomposition (Kutz et al., 2016; Schmid, 2010), Koopman theory (Steven L. Brunton et al., 2021), nonlinear autoregressive algorithms (Billings, 2013), neural networks (Pathak et al., 2018; M. Raissi et al., 2019; Vlachas et al., 2018), Gaussian process regression (Maziar Raissi et al., 2017), operator inference and reduced-order modeling (Benner et al., 2015; Peherstorfer & Willcox, 2016; Qian et al., 2020), genetic programming (Bongard & Lipson, 2007; Schmidt & Lipson, 2009), and sparse regression (Steven L. Brunton et al., 2016). These approaches have seen many variants and improvements over the years, so data-driven modeling software must be regularly updated to remain useful to the scientific community. The SINDy approach has experienced particularly rapid development, motivating this major update to aggregate these innovations into a single open-source tool that is transparent and easy to use for non-experts or scientists from other fields.

The original `PySINDy` code (de Silva et al., 2020) provided an implementation of the traditional SINDy method (Steven L. Brunton et al., 2016), which assumes that the dynamical evolution of a state variable $\mathbf{q}(t) \in \mathbb{R}^n$ follows an ODE described by a function $\mathbf{f}$,

$$\frac{d}{dt}\mathbf{q} = \mathbf{f}(\mathbf{q}). \tag{1}$$

SINDy approximates the dynamical system $\mathbf{f}$ in Eq. (1) as a sparse combination of terms from a library of candidate basis functions $\boldsymbol{\theta}(\mathbf{q}) = [\theta_1(\mathbf{q}), \theta_2(\mathbf{q}), \dots, \theta_p(\mathbf{q})]$

$$\mathbf{f}(\mathbf{q}) \approx \sum_{k=1}^{p} \theta_k(\mathbf{q})\boldsymbol{\xi}_k, \quad \text{or equivalently} \quad \frac{d}{dt}\mathbf{q} \approx \boldsymbol{\Theta}(\mathbf{q})\boldsymbol{\Xi}, \tag{2}$$

where $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_p]$ contain the sparse coefficients. In order for this strategy to be successful, a reasonably accurate approximation of $\mathbf{f}(\mathbf{q})$ should exist as a sparse expansion in the span of $\boldsymbol{\theta}$. Therefore, background scientific knowledge about expected terms in $\mathbf{f}(\mathbf{q})$ can be used to choose the library $\boldsymbol{\theta}$. To pose SINDy as a regression problem, we assume we have a set of state measurements sampled at time steps $t_1, ..., t_m$ and rearrange the data into the data matrix $\mathbf{Q} \in \mathbb{R}^{m\times n}$,

$$\mathbf{Q} = \begin{bmatrix} q_1(t_1) & q_2(t_1) & \cdots & q_n(t_1) \\ q_1(t_2) & q_2(t_2) & \cdots & q_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ q_1(t_m) & q_2(t_m) & \cdots & q_n(t_m) \end{bmatrix}. \tag{3}$$

A matrix of derivatives in time, $\mathbf{Q}_t$, is defined similarly and can be numerically computed from $\mathbf{Q}$. PySINDy defaults to second order finite differences for computing derivatives, although a host of more sophisticated methods are now available, including arbitrary order finite differences, Savitzky-Golay derivatives (i.e. polynomial-filtered derivatives), spectral derivatives with optional filters, arbitrary order spline derivatives, and total variational derivatives (Ahnert & Abel, 2007; Chartrand, 2011; Tibshirani & Taylor, 2011).

After $\mathbf{Q}_t$ is obtained, Eq. (2) becomes $\mathbf{Q}_t \approx \boldsymbol{\Theta}(\mathbf{Q})\boldsymbol{\Xi}$ and the goal of the SINDy sparse regression problem is to choose a sparse set of coefficients $\boldsymbol{\Xi}$ that accurately fits the measured data in $\mathbf{Q}_t$. We can promote sparsity in the identified coefficients via a sparse regularizer $R(\boldsymbol{\Xi})$, such as the l_0 or l_1 norm, and use a sparse regression algorithm such as SR3 (Champion et al., 2020) to solve the resulting optimization problem,

$$\text{argmin}_{\boldsymbol{\Xi}}\|\mathbf{Q}_t - \boldsymbol{\Theta}(\mathbf{Q})\boldsymbol{\Xi}\|^2 + R(\boldsymbol{\Xi}). \tag{4}$$

The original `PySINDy` package was developed to identify a particular class of systems described by Eq. (1). Recent variants of the SINDy method are available that address systems with control inputs and model predictive control (MPC) (Fasel, Kaiser, et al., 2021; Kaiser et al., 2018), systems with physical constraints (Kaptanoglu, Morgan, et al., 2021; Loiseau & Brunton, 2018), implicit ODEs (Kaheman et al., 2020; Mangan et al., 2016), PDEs (Rudy et al., 2017; Schaeffer, 2017), and weak form ODEs and PDEs (Messenger & Bortz, 2021; Reinbold et al., 2020; Schaeffer & McCalla, 2017). Other methods, such as ensembling and sub-sampling (Delahunt & Kutz, 2021; Maddu et al., 2019; Reinbold et al., 2021), are often vital for making the identification of Eq. (1) more robust. In order to incorporate these new developments and accommodate the wide variety of possible dynamical systems, we have extended `PySINDy` to a more general setting and added significant new functionality. Our code[1] is thoroughly documented, contains extensive examples, and integrates a wide range of

[1] https://github.com/dynamicslab/pysindy

functionality, some of which may be found in a number of other local SINDy implementations[2]. In contrast to some of these existing codes, PySINDy is completely open-source, professionally-maintained (for instance, providing unit tests and adhering to PEP8 stylistic standards), and minimally dependent on non-standard Python packages.

New features

Given spatiotemporal data $\mathbf{Q}(\mathbf{x}, t) \in \mathbb{R}^{m\times n}$, and optional control inputs $\mathbf{u} \in \mathbb{R}^{m\times r}$ (note m has been redefined here to be the product of the number of spatial measurements and the number of time samples), PySINDy can now approximate algebraic systems of PDEs (and corresponding weak forms) in an arbitrary number of spatial dimensions. Assuming the system is described by a function $\mathbf{g}$, we have

$$\mathbf{g}(\mathbf{q}, \mathbf{q}_t, \mathbf{q}_x, \mathbf{q}_y, \mathbf{q}_{xx}, ..., \mathbf{u}) = 0. \tag{5}$$

ODEs, implicit ODEs, PDEs, and other dynamical systems are subsets of Eq. (5). We can accommodate control terms and partial derivatives in the SINDy library by adding them as columns in $\mathbf{\Theta}(\mathbf{Q})$, which becomes $\mathbf{\Theta}(\mathbf{Q}, \mathbf{Q}_t, \mathbf{Q}_x, ..., \mathbf{u})$.

In addition, we have extended PySINDy to handle more complex modeling scenarios, including trapping SINDy for provably stable ODE models for fluids (Kaptanoglu, Callaham, et al., 2021), models trained using multiple dynamic trajectories, and the generation of many models with sub-sampling and ensembling methods (Fasel, Kutz, et al., 2021) for cross-validation and probabilistic system identification. In order to solve Eq. (5), PySINDy implements several different sparse regression algorithms. Greedy sparse regression algorithms, including stepwise sparse regression (SSR) (Boninsegna et al., 2018) and forward regression orthogonal least squares (FROLS) (Billings, 2013), are now available. For maximally versatile candidate libraries, the new GeneralizedLibrary class allows for tensoring, concatenating, and otherwise combining many different candidate libraries, along with optionally specifying a subset of the inputs to use for generating each of the libraries. Figure 1 illustrates the PySINDy code structure, changes, and high-level goals for future work, and YouTube tutorials for this new functionality are available online.

PySINDy includes extensive Jupyter notebook tutorials that demonstrate the usage of various features of the package and reproduce nearly the entirety of the examples from the original SINDy paper (Steven L. Brunton et al., 2016), trapping SINDy paper (Kaptanoglu, Callaham, et al., 2021), and the PDE-FIND paper (Rudy et al., 2017). We include an extended example for the quasiperiodic shear-driven cavity flow (Callaham et al., 2021). As a simple illustration of the new functionality, we demonstrate how SINDy can be used to identify the Kuramoto-Sivashinsky (KS) PDE from data. We train the model on the first 60% of the data from Rudy et al. (Rudy et al., 2017), which in total contains 1024 spatial grid points and 251 time steps. The KS model is identified correctly and the prediction for $\dot{\mathbf{q}}$ on the remaining testing data indicates strong performance in Figure 2. Lastly, we provide a useful flow chart in Figure 3 so that users can make informed choices about which advanced methods are suitable for their datasets.

[2] https://github.com/snagcliffs/PDE-FIND, https://github.com/eurika-kaiser/SINDY-MPC, https://github.com/dynamicslab/SINDy-PI, https://github.com/SchatzLabGT/SymbolicRegression, https://github.com/dynamicslab/databook_python, https://github.com/sheadan/SINDy-BVP, https://github.com/sethhirsh/BayesianSindy, https://github.com/racdale/sindyr, https://github.com/SciML/DataDrivenDiffEq.jl, https://github.com/MathBioCU/WSINDy_PDE, https://github.com/pakreinbold/PDE_Discovery_Weak_Formulation, https://github.com/ZIB-IOL/CINDy

Conclusion

The goal of the PySINDy package is to enable anyone with access to measurement data to engage in scientific model discovery. The package is designed to be accessible to inexperienced users, adhere to scikit-learn standards, include most of the existing SINDy variations in the literature, and provide a large variety of functionality for more advanced users. We hope that researchers will use and contribute to the code in the future, pushing the boundaries of what is possible in system identification.

Acknowledgments

PySINDy is a fork of sparsereg (Quade, 2018). SLB, AAK, KK, and UF acknowledge support from the Army Research Office (ARO W911NF-19-1-0045). JLC acknowledges support from funding support from the Department of Defense (DoD) through the National Defense Science & Engineering Graduate (NDSEG) Fellowship Program. ZGN is a Washington Research Foundation Postdoctoral Fellow.

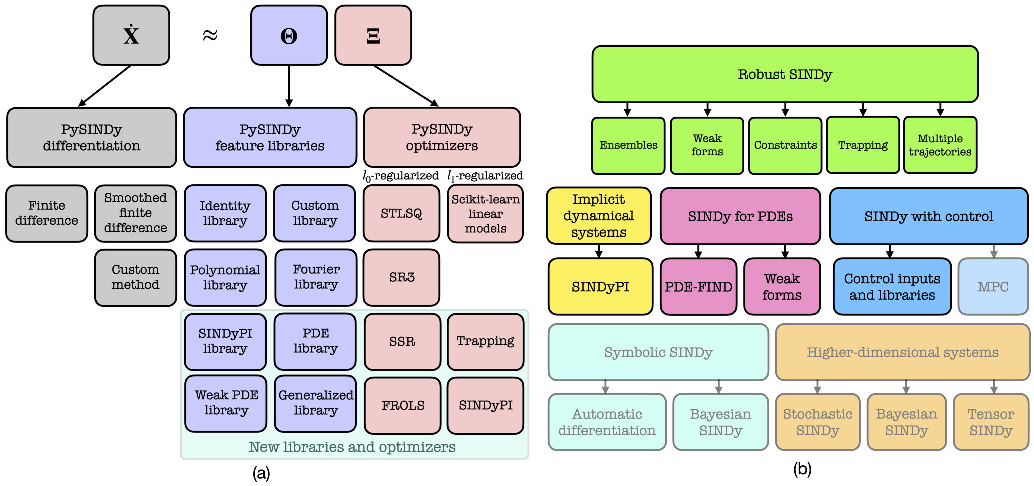

Figure 1: Summary of SINDy features organized by (a) PySINDy structure and (b) functionality. (a) Hierarchy from the sparse regression problem solved by SINDy, to the submodules of PySINDy, to the individual optimizers, libraries, and differentiation methods implemented in the code. (b) Flow chart for organizing the SINDy variants and functionality in the literature. Bright color boxes indicate the features that have been implemented through this work, roughly organized by functionality. Semi-transparent boxes indicate features that have not yet been implemented.

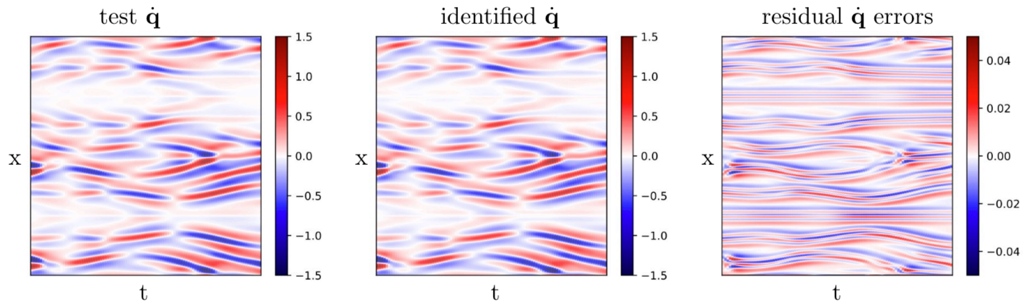

Figure 2: `PySINDy` can now be used for PDE identification; we illustrate this new capability by accurately capturing a set of testing data from the Kuramoto-Sivashinsky system, described by $q_t = -qq_x - q_{xx} - q_{xxxx}$. The identified model is $q_t = -0.98qq_x - 0.99q_{xx} - 1.0q_{xxxx}$.

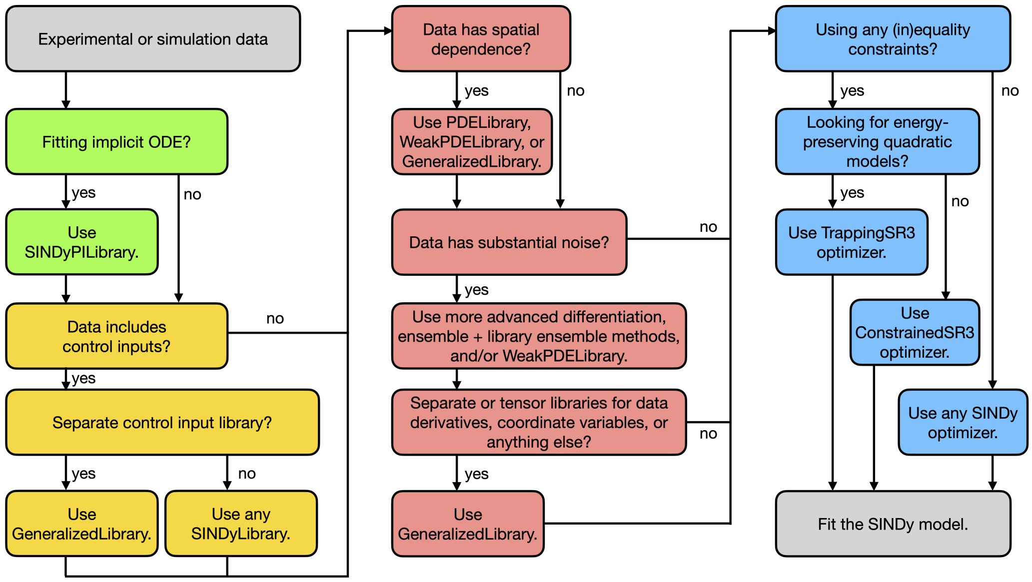

Figure 3: This flow chart summarizes how `PySINDy` users can start with a dataset and systematically choose the proper candidate library and sparse regression optimizer that are tailored for a specific scientific task.

References

Ahnert, K., & Abel, M. (2007). Numerical differentiation of experimental data: Local versus global methods. *Computer Physics Communications*, *177*(10), 764–774. https://doi.org/10.1016/j.cpc.2007.03.009

Benner, P., Gugercin, S., & Willcox, K. (2015). A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, *57*(4), 483–531. https://doi.org/10.1137/130932715

Billings, S. A. (2013). *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons.

Bongard, J., & Lipson, H. (2007). Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl. Acad. Sciences*, *104*(24), 9943–9948. https://doi.org/10.1073/pnas.0609476104

Boninsegna, L., Nüske, F., & Clementi, C. (2018). Sparse learning of stochastic dynamical equations. *The Journal of Chemical Physics*, *148*(24), 241723. https://doi.org/10.1063/1.5018409

Brunton, Steven L., Budišić, M., Kaiser, E., & Kutz, J. N. (2021). Modern Koopman theory for dynamical systems. *arXiv Preprint arXiv:2102.12086*.

Brunton, Steven L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, *113*(15), 3932–3937. https://doi.org/10.1073/pnas.1517384113

Callaham, J. L., Brunton, S. L., & Loiseau, J.-C. (2021). On the role of nonlinear correlations in reduced-order modeling. *arXiv Preprint arXiv:2106.02409*.

Champion, K., Zheng, P., Aravkin, A. Y., Brunton, S. L., & Kutz, J. N. (2020). A unified sparse optimization framework to learn parsimonious physics-informed models from data. *IEEE Access*, *8*, 169259–169271. https://doi.org/10.1109/access.2020.3023625

Chartrand, R. (2011). Numerical differentiation of noisy, nonsmooth data. *International Scholarly Research Notices*, *2011*. https://doi.org/10.5402/2011/164564

de Silva, B., Champion, K., Quade, M., Loiseau, J.-C., Kutz, J. N., & Brunton, S. (2020). PySINDy: A Python package for the sparse identification of nonlinear dynamical systems from data. *Journal of Open Source Software*, *5*(49), 1–4. https://doi.org/10.21105/joss.02104

Delahunt, C. B., & Kutz, J. N. (2021). A toolkit for data-driven discovery of governing equations in high-noise regimes. *arXiv Preprint arXiv:2111.04870*.

Fasel, U., Kaiser, E., Kutz, J. N., Brunton, B. W., & Brunton, S. L. (2021). SINDy with control: A tutorial. *arXiv Preprint arXiv:2108.13404*.

Fasel, U., Kutz, J. N., Brunton, B. W., & Brunton, S. L. (2021). Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *arXiv Preprint arXiv:2111.10992*.

Kaheman, K., Kutz, J. N., & Brunton, S. L. (2020). SINDy-PI: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A*, *476*(2242), 20200279. https://doi.org/10.1098/rspa.2020.0279

Kaiser, E., Kutz, J. N., & Brunton, S. L. (2018). Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society of London A*, *474*(2219). https://doi.org/10.1098/rspa.2018.0335

Kaptanoglu, A. A., Callaham, J. L., Aravkin, A., Hansen, C. J., & Brunton, S. L. (2021). Promoting global stability in data-driven models of quadratic nonlinear dynamics. *Phys. Rev. Fluids*, *6*, 094401. https://doi.org/10.1103/PhysRevFluids.6.094401

Kaptanoglu, A. A., Morgan, K. D., Hansen, C. J., & Brunton, S. L. (2021). Physics-constrained, low-dimensional models for magnetohydrodynamics: First-principles and data-driven approaches. *Phys. Rev. E*, *104*, 015206. https://doi.org/10.1103/physreve.104.015206

Kutz, J. N., Brunton, S. L., Brunton, B. W., & Proctor, J. L. (2016). *Dynamic mode decomposition: Data-driven modeling of complex systems*. SIAM.

Loiseau, J.-C., & Brunton, S. L. (2018). Constrained sparse Galerkin regression. *Journal of Fluid Mechanics*, *838*, 42–67. https://doi.org/10.1017/jfm.2017.823

Maddu, S., Cheeseman, B. L., Sbalzarini, I. F., & Müller, C. L. (2019). Stability selection enables robust learning of partial differential equations from limited noisy data. *arXiv Preprint arXiv:1907.07810*.

Mangan, N. M., Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, *2*(1), 52–63. https://doi.org/10.1109/tmbmc.2016.2633265

Messenger, D. A., & Bortz, D. M. (2021). Weak SINDy for partial differential equations. *Journal of Computational Physics*, 110525. https://doi.org/10.1016/j.jcp.2021.110525

Pathak, J., Hunt, B., Girvan, M., Lu, Z., & Ott, E. (2018). Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, *120*(2), 024102. https://doi.org/10.1103/physrevlett.120.024102

Peherstorfer, B., & Willcox, K. (2016). Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, *306*, 196–215. https://doi.org/10.1016/j.cma.2016.03.025

Qian, E., Kramer, B., Peherstorfer, B., & Willcox, K. (2020). Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. *Physica D: Nonlinear Phenomena*, *406*, 132401. https://doi.org/10.1016/j.physd.2020.132401

Quade, M. (2018). *Sparsereg - collection of modern sparse regression algorithms*. https://doi.org/10.5281/zenodo.1173754

Raissi, M., Perdikaris, P., & Karniadakis, G. (2019). 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. https://doi.org/10.1016/j.jcp.2018.10.045

Raissi, Maziar, Perdikaris, P., & Karniadakis, G. E. (2017). Machine learning of linear differential equations using Gaussian processes. *Journal of Computational Physics*, *348*, 683–693. https://doi.org/10.1016/j.jcp.2017.07.050

Reinbold, P. A., Gurevich, D. R., & Grigoriev, R. O. (2020). Using noisy or incomplete data to discover models of spatiotemporal dynamics. *Physical Review E*, *101*(1), 010203. https://doi.org/10.1103/physreve.101.010203

Reinbold, P. A., Kageorge, L. M., Schatz, M. F., & Grigoriev, R. O. (2021). Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression. *Nature Communications*, *12*(1), 1–8. https://doi.org/10.1038/s41467-021-23479-0

Rudy, S. H., Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2017). Data-driven discovery of partial differential equations. *Science Advances*, *3*(e1602614). https://doi.org/10.1126/sciadv.1602614

Schaeffer, H. (2017). Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society a*, *473*, 20160446. https://doi.org/10.1098/rspa.2016.0446

Schaeffer, H., & McCalla, S. G. (2017). Sparse model selection via integral terms. *Physical Review E*, *96*(2), 023302. https://doi.org/10.1103/physreve.96.023302

Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, *656*, 5–28. https://doi.org/10.1017/s0022112010001217

Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science*, *324*(5923), 81–85. https://doi.org/10.1126/science.1165893

Tibshirani, R. J., & Taylor, J. (2011). The solution path of the generalized lasso. *The Annals of Statistics*, *39*(3), 1335–1371. https://doi.org/10.1214/11-aos878

Vlachas, P. R., Byeon, W., Wan, Z. Y., Sapsis, T. P., & Koumoutsakos, P. (2018). Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proc. R. Soc. A*, *474*(2213), 20170844. https://doi.org/10.1098/rspa.2017.0844