Weak-form Sparse Identification of Differential Equations from Noisy Measurements

by

Daniel Ames Messenger

B.S., B.A, University of Puget Sound, 2015
M.S., Simon Fraser University, 2019

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Applied Mathematics

at the
University of Colorado-Boulder
Year 2022

Date of Dissertation Defense: 6/2/2022
Graduate Committee:

David M. Bortz (Advisor), Professor, Applied Mathematics
Nancy Rodriguez, Assistant Professor, Applied Mathematics
Yu-Jui Huang, Assistant Professor, Applied Mathematics
Stephen Becker, Associate Professor, Applied Mathematics
Xuedong Liu, Professor, Biochemistry

Abstract

Data-driven modeling refers to the use of measurement data to infer the parameters and structure of a mathematical model. While currently an active area research, data-driven modeling is characterized by the cycle of hypothesis, observation, and conclusion, which is none other than the scientific method, and can be traced back to Aristotle. For millenia, attempts have been made to distill governing laws from the observations made on a given system, with the hope of both explaining the observations and predicting future observations. Major accomplishments in this vein include Archimedes' principle of buoyancy, Newtonian physics, and Röntgen's X-rays. In each of these cases, the observations of a peculiar phenomena, often accidental, compelled the researcher to develop models. It is this map from observations to models that is at the heart of this thesis.

The paradigm shift in recent years, driven by increased computing power, availability of large quantities of data, and the development of advanced mathematical techniques and algorithms, has been to *automate* the process of data-driven modeling. In the terminology of hypothesis, observation, and conclusion, automation can occur at the level of developing hypotheses about possible mathematical models, or the design of experiments which differentiate between the many possible models, or the map which takes experimental data and returns a mathematical model. To complete the cycle, one could also consider algorithms which generate potential *nearby* models given the model that is found to best fit the data.

In this dissertation, we explore algorithms which automate the map from *observations* to *governing equations*, specifically differential equations. Our key contribution is the development of algorithms which identify differential equations in a *weak form*, which loosely refers to integrating the differential equation against arbitrary functions. We will show that the weak form is an ideal framework for identifying models from data if the criteria are *robustness* to data corruptions, *highly accurate* model recovery when corruption levels are low, and *computational efficiency*.

We will demonstrate the superiority of our weak-form sparse identification for nonlinear dynamics algorithm (WSINDy) in the discovery of correct underlying model equations in a variety of differential equation and data corruption scenarios. We start with the simplest case, of ordinary differential equations (ODEs) depending only on time. We then move to partial differential equations (PDEs), where state variables change in both time and space. We then bridge the two previous regimes by considering interacting particle systems (IPSs), where the weak form is used to identify a mean-field PDE using data that can also be described as a large system of ODEs. Finally, we establish feasibility of weak-form identification of PDEs in an *online* context, where data is streamed in. In the online setting, we demonstrate the possibility of identifying time-varying models as well as models from data in four dimensions (3 space + 1 time).

Dedication

I dedicate this dissertation to my wife Anna Rae Schierbeek, for her continued love, patience, encouragement, and insight.

Acknowledgements

I would like to acknowledge first and foremost my advisor David Bortz. Research was exciting from our very first meeting, when David shared a cell migration dataset and directed me to the algorithm PDE-FIND to see what models might be learned, and what weaknesses existed in the "state-of-the-art". This combination led us to develop the WSINDy algorithm a few months later, through a series of experiments, many of which I would not have known to pursue without David's advice. Much of the time I felt more like a partner in crime working with David than a subordinate, except in the many instances where I needed guidance and for someone else to take the reigns. Most importantly, I've always felt that my ideas were honored and encouraged, and that my skills were valued. I'm glad to be able to finally pursue as a postdoc some of the many collaborations that David has initiated surrounding WSINDy.

My committee members are in large part responsible for my advanced knowledge of applied mathematics. Stephen Becker, Nancy Rodriguez, and Yu-Jui Huang taught me optimization, analysis, and stochastic differential equations, giving clear, student-centered lectures that I will strive to emulate in my future career. Moreover, their persepectives on success in academia, graduate student life, imposter syndrome, etc., have been crucial to my well-being, and my sense of place in APPM. I also had the pleasure of working on the challenging real-world problem of cell migration with Xuedong Liu, who was always encouraging of my efforts to make use of applied mathematics in the biological sphere. While we haven't quite cracked it yet, our progress on cell migration will continue to push me forward, if not through merely rewatching videos of the careful microscale experiments conducted in Liu's lab by Graycen Wheeler, Douglas Chapnick, and Nicholas Gray.

Outside of my PhD committee, I'd like to thank the mentors of my masters' program, my supervisors Razvan Fetecau and Ralf Wittenberg, and Professor Weiran Sun, for believing in me as I was applying for PhD programs. I would not be here without them. I also learned a ton from working alongside Hui Huang, and hope to be as kind a mentor to grad students when I'm a postdoc. I am also indebted to Anne Dougherty, Maribeth Oscamou, and William Preston Cummings for teaching me about pedagogy and the broader value of education.

My budding research career in applied mathematics has been greatly improved by Emiliano Dall'Anese, co-author of the contents of Chapter 5 of this thesis, which originated as my final project in his course on online optimization. I'm also grateful for my interactions with J. Nathan Kutz and in research group, including Kadierdan Kaheman. Thank you for the encouragement and

hospitality during my visit to Seattle.

I would like to thank my fellow grad students at CU, including but not limited to Sam Zhang, Sage Shaw, Alex Schien, Shay Gilpin, Abe Ellison, Kate Bubar, Sachin Natech, Richard Clancy, and Zofia Stanley, for helping me through a difficult yet rewarding three years. As well, I'd like to thank my fellow grad students at Simon Fraser University, Ian May, Ray Walsh, Nicola Mulberry, Anton Iatcenko, Riley Wishart, and Mohsen Seifi, for consistently keeping in touch over Zoom and remaining great friends. I wish you all the best and hope to remain close.

Last but not least, there are many individuals outside of applied mathematics and academia who have kept me sane, and made life truly joyful over these last three years. My wife Anna, who has lovingly endured my erratic graduate student schedule while tackling the much more challenging job of teaching six year-olds how to read, write, add numbers, and not eat each other. Thanks to my friends Max, Mo, Nicolette, and Graehm, who have taken me on adventures in the snow and on bikes, to musical happenings, and many game nights. My siblings Amanda, Nora, and Peter, and my parents, who continue to respond with utter bewilderment about what I do for "work", although I will say I've gotten better at explaining things. I'm always thankful for the opportunity to interpret math in the grander scheme of life. So glad to have you all. Thanks for believing in me!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Stemming from Akaike's seminal work in the 1970's [1, 2], research into the systematic creation of accurate mathematical models from data has progressed dramatically. In the last 40 years, substantial developments have been made at the interface of applied mathematics, computer science, and statistics to design algorithms for automated selection of models from data that are both statistically rigorous and computationally efficient (see [21, 80, 85, 136, 145, 148] for both theory and applications). An important achievement in this field is the *sparse regression* approach, which formulates and subsequently discretizes the model selection problem in terms of a candidate basis of functions evaluated at the given dataset, and utilizes a sparsification measure to avoid overfitting. While similar approaches were already commonplace by 1975 for linear discrete-time system analysis (see the textbook [29]), extension to nonlinear continuous dynamical systems wasn't initiated until 1987 [35]. The subject then progressed slowly for two decades.

Notable improvements arose with the application of genetic algorithms in 2009 [124], where the authors developed a symbolic regression algorithm and employ Pareto front analysis to select sparse models for ordinary differential equations. In 2011, an algorithm was developed to select models for catastrophe prediction using ℓ_1-minimization, the hallmark of compressive sensing [140]. Finally, this approach gained widespread popularity after its generalization as the SINDy algorithm (Sparse Identification of Nonlinear Dynamics) [23], which demonstrated success in selected sparse models across the spectrum of discrete and continuous nonlinear dynamical systems.

Sparse regression approaches to data-driven modeling are promising due to their potential generalizability, computational efficiency, and interpretability. Sparse models have the best shot at avoiding overfitting, since they relax the requirement that the model must accurately approximate the training data, if the resulting model is simple enough. From a practical standpoint, sparse models have fewer operations and hence can be solved efficiently. Moreover, the output is a model with recognizable terms selected from a known dictionary, which can be analyzed and controlled using the wealth of techniques developed over the last few centuries.

On the other hand, sparse regression is only as useful as the library of terms designed by the user, and the ability of the algorithm to *select the right terms*, possibly amidst several seemingly equivalent options. An alternative view that avoids these challenges is the *approximation* approach (e.g. using deep neural networks), which seeks only to find a fast input-to-output map that approximates the underlying phenomena, with no aspirations of interpreting what happens between input and output. This is advantageous from the point of view of making fast predictions across many inputs. The algorithm WSINDy discussed here (weak-form SINDy) lies distinctly on the side of selecting sparse models. One can only expect that superior methods will arise at the intersection of *selection* and *approximation* in the future, and we argue that WSINDy should be used to tackle the selection component.

Notable variants of sparse regression-based model selection include [120], [23], and [122], where the Douglas-Rachford algorithm, sequential-thresholding least squares (STLS), and basis pursuit with denoising are used to successfully arrive at sparse models. The SINDy algorithm, upon which our weak-form variant WSINDy is based, uses sequential-thresholding least-squares (STLS) to enforce a sparse solution $x \in \mathbb{R}^n$ to a linear system $Ax = b$. While STLS provably converges to a local minimizer of the desired functional $F(x) = \|Ax - b\|_2^2 + \lambda^2 \|x\|_0$ in at-most n iterations [158], methods of enforcing sparsity remain a focus of intense study [161, 5, 27, 34]. A primary driver of continued developments in sparse regression for data-driven modeling is the highly-correlated nature of resulting linear systems, rendering theoretical guarantees from compressed sensing unhelpful. For example, there is no hope in general of guaranteeing a restricted isometry property or mutual coherence (see [52] for a comprehensive review of compressed sensing) when each column of the matrix is constructed from the same dataset.

In addition to the sparse regression approach adopted in SINDy, some of the primary techniques for data-driven modeling include Gaussian process regression [109, 113], deep neural networks [91, 89, 88, 118, 141, 80, 32, 106, 79, 152], Bayesian inference [159, 160, 144, 145], and classical methods from numerical analysis [69, 75, 150]. These approaches qualitatively differ in the interpretability of the resulting data-driven dynamical system, computational efficiency of the algorithm, scope of the algorithm, and its robustness to noise, scale separation, limited data, etc. For instance, a neural-network based data-driven dynamical system does not easily lend itself to physical interpretation[1]. Concerning the scope of an algorithm, several methods have been developed to discover models in specific contexts, such as low-degree polynomial chaotic systems, cyclic ODEs, interacting particles, Hamiltonian dynamics, and biochemical reaction networks [137, 123, 90, 149, 63]. Often these limited-scope algorithms are amenable to analysis and are presented with some form of recovery guarantee. In this dissertation, we have aimed to provide an algorithm of wide scope that is near the state-of-the-art with respect to robustness to noise, computational efficiency, and interpretability.

[1]There have been efforts to address the interpretability of neural networks, see e.g. [104, 135, 115].

1.2 Weak-Form Sparse Identification of Nonlinear Dynamics

A significant disadvantage of the vast majority of data-driven modeling methods, both in practice and in theory, is the requirement that pointwise derivatives of the data are either available or can be accurately computed. This severely limits an algorithm's robustness to noise and hence its applicability to real world data. This dissertation is primarily concerned with relaxing this assumption and providing rigorous justification for the weak formulation of the dynamics as a means to circumvent this ubiquitous problem in model selection. Building off of the SINDy framework, we present the robust discovery algorithm WSINDy (Weak-form SINDy), which operates under the assumption that derivatives of the data are neither available nor computable, and that the only prior knowledge of the governing equations is their inclusion in a large model library.

The use of integral equations for system identification was proposed as early as the 1980's [35] and was carried out in a sparse regression framework in [121] in the context of ODEs, however neither works utilized the full generality of the weak form. Steps to alleviate pointwise derivative approximations are also taken by the authors of [112] and [151], where neural network-based recovery schemes are combined with integral and abstract evolution equations to recover PDEs, and in [143, 142], where the finite element-based method Variational System Identification (VSI) is introduced to identify reaction-diffusion systems and uses backward Euler to approximate the time derivative.

WSINDy is a method for discovering differential equations without the use of any pointwise derivative approximations, black-box routines or conventional noise filtering. Through integration by parts in both space and time against smooth compactly-supported test functions, WSINDy is able to recover models from datasets with much higher noise levels, and PDEs from truly weak solutions (see Figure 3.5.1 in Section 3.5, as well as interacting particle system examples in 4.5). This works suprisingly well even as the signal-to-noise ratio approaches one (see Figure 3.5.3). Furthermore, WSINDy achieves high-accuracy recovery in the low-noise regime, due to spectral accuracy of the trapezoidal rule for smooth compactly supported functions (see Figure 2.2).

It should be noted that these overwhelming improvements resulting from a fully-weak identification method were also been discovered independently by another group [114, 57] shortly before WSINDy was released in May 2020. WSINDy offers several advantages over these alternative frameworks. Firstly, we use a convolutional weak form which enables efficient model identification using the Fast Fourier Transform (FFT). For measurement data with N points in each of the $D+1$ space-time dimensions (N^{D+1} total data points), the resulting algorithmic complexity of WSINDy in the PDE setting is at worst $\mathcal{O}(N^{D+1}\log(N))$, in other words $\mathcal{O}(\log(N))$ floating point operations per data-point. Subsampling then greatly reduces the cost. Furthermore, our FFT-based approach reveals a key mechanism behind the observed robustness to noise, namely that spectral decay properties of test functions can be tuned to damp noise-dominated modes in the data, and we develop a learning algorithm for test function hyperparameters based on this mechanism. WSINDy

also utilizes scale invariance of the PDE and a modified STLS algorithm with automatic threshold selection to recover models from (i) poorly-scaled data and (ii) large candidate model libraries. These properties are what has enabled our weak-form method to thrive in the challenging scenarios of interacting particle systems (Chapter 4) and online identification (Chapter 5), which were not explored in [114, 57].

1.3 Dissertation Outline

Chapters 2-5 contain previously published or submitted work, and are presented here with little modification. We include the original abstracts at the beginning of each chapter, as well as an appendix following each chapter (note there is no overall appendix at the end of the dissertation). In Chapter 2, we present WSINDy in the context of ordinary differential equations, content which was submitted to SIAM: Multiscale Modeling and Simulation in June 2020 and published September 2021 [100]. We then demonstrate the wide scope of the weak-form methodology in Chapter 3, which extends the original framework to partial differential equations. Chapter 3 was submitted to the Journal of Computational Physics in July 2020 and published October 2021 [99]. In Chapter 4 we demonstrate effectiveness of the weak-form in identifying stochastic interacting particle system dynamics from particle trajectories by using mean-field PDE representations. Chapter 4 was submitted to Physica D: Nonlinear Phenomena in November 2021 and is currently in press [98]. In Chapter 5 we provide a brief proof of concept for an online version of the algorithm, which departs from previous approaches in the way that sparsity is enforced, but retains the PDE-level implementation. Chapter 5 was recently accepted for publication in the Proceedings of Machine Learning Research [101]. Chapters 2-4 were written in collaboration with my advisor David M. Bortz, while Chapter 5 was written in collaboration with David M. Bortz and Emiliano Dall'Anese.

We now review the main findings of each chapter.

1.3.1 Chapter 2: WSINDy for ODEs

Applied to ODEs, the primary objective is demonstrate a crucial aspect of our weak-form approach, that WSINDy is capable of identifying systems across a wide range of noise levels. This is in contrast to strong-form methods, where the algorithm is required to change (e.g. the differentiation method and denoising techniques) depending on the different noise levels. With WSINDy we achieve both high-accuracy recovery for low-noise, and high-probability recovery for high-noise, without changing the method. We also show that within the myriad possibilities for weak formulations, a particular class of test functions offers suprising accuracy and robustness to noise (see Figure (2.2) for accuracy and Figures (2.A.2)-(2.A.3) for robustness compared to standard SINDy). We identify several important questions that arise in the automated selection of test-functions and weak-form based variance reduction, and provide some preliminary answers.

1.3.2 Chapter 3: WSINDy for PDEs

WSINDy truly shines in the PDE setting, where the weak-form is using to identify models from solutions that can only be understood in a weak sense. Here model identification is often robust to a staggering 50% noise (to be defined below). We find that the question of test function selection is related to the Fourier spectrum of the dataset, which is revealed by posing the weak-form as a convolution, and restricting the focus to separable test functions. (Initially, we developed the convolutional form and separability for purely computational purposes.) We also introduce a simple and interpretable way to select the sparsity threshold, leveraging the speed of the STLS algorithm, and we introduce data rescaling into the algorithm to improve conditioning, especially in the face of data with numerically disparate scales.

1.3.3 Chapter 4: WSINDy for IPSs

Harkening back to the author's time as a masters student, we next develop WSINDy for stochastic interacting particle systems (IPSs) by leveraging that the empirical measure of an IPS converges in a suitable weak sense to the solution of an associated nonlocal PDE. We incorporate nonlocal terms into the WSINDy PDE library, which are computed efficiently by performing convolutions on low-rank representations of kernels, again using the FFT. Principally, this approach bridges an important gap from trajectory data on finitely-many particles to identification of models in the *continuum limit*. We show that this approach is effective at recovering *homogenized PDEs* directly from particle data with highly-oscillatory diffusion, as well as nonlocal models with singular kernels.

1.3.4 Chapter 5: Online WSINDy

In the online setting, we restrict ourselves to the setting of limited storage capacity, streaming data, and no ability to perform least-squares solves. The goal is to iteratively solve for the correct model and track parameter values using snapshots in time of the solution to the underlying PDE. In this setting, we explore models with time-varying coefficients, finding that the algorithm can track an abruptly changing wavespeed, and large datasets that cannot be processed easily on a laptop.

Chapter 2

WSINDy for ODEs

Abstract

We present a novel weak formulation and discretization for discovering governing equations from noisy measurement data. This method of learning differential equations from data fits into a new class of algorithms that replace pointwise derivative approximations with linear transformations and variance reduction techniques. Compared to the standard SINDy algorithm presented in [23], our so-called Weak SINDy (WSINDy) algorithm allows for reliable model identification from data with large noise (often with ratios greater than 0.1) and reduces the error in the recovered coefficients to enable accurate prediction. Moreover, the coefficient error scales linearly with the noise level, leading to high-accuracy recovery in the low-noise regime. Altogether, WSINDy combines the simplicity and efficiency of the SINDy algorithm with the natural noise reduction of integration, as demonstrated in [121], to arrive at a robust and accurate method of sparse recovery.

2.1 Problem Statement

Consider a first-order dynamical system in D dimensions of the form

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(\boldsymbol{x}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^D, \quad 0 \le t \le T, \tag{2.1.1}$$

and measurement data $\mathbf{y} \in \mathbb{R}^{M\times D}$ given at M timepoints $\mathbf{t} = (t_1, \ldots, t_M)^T$ by

$$\mathbf{y}_{md} = \mathbf{x}_d(t_m) + \epsilon_{md}, \qquad m \in [M],\ d \in [D],$$

where throughout we use the bracket notation $[M] := \{1, \ldots, M\}$. The variable $\epsilon \in \mathbb{R}^{M\times D}$ represents a matrix of i.i.d. measurement noise. The focus of this chapter is the reconstruction of the dynamics (2.1.1) from the measurements $\mathbf{y}$.

The SINDy algorithm (Sparse Identification of Nonlinear Dynamics [23]) has been shown to be successful in solving this problem for sparsely represented nonlinear dynamics when noise is small and dynamic scales do not vary across multiple orders of magnitude. This framework assumes that the function $\mathbf{F} : \mathbb{R}^D \to \mathbb{R}^D$ in (2.1.1) is given component-wise by

$$\mathbf{F}_d(\mathbf{x}(t)) = \sum_{j=1}^{J} \mathbf{w}^\star_{jd}\, f_j(\mathbf{x}(t)) \tag{2.1.2}$$

for some known family of functions $(f_j)_{j\in[J]}$ and a sparse weight matrix $\mathbf{w}^\star \in \mathbb{R}^{J\times D}$. The problem is then transformed into solving for $\mathbf{w}^\star$ by building a data matrix $\Theta(\mathbf{y}) \in \mathbb{R}^{M\times J}$ given by

$$\Theta(\mathbf{y})_{mj} = f_j(\mathbf{y}_m), \qquad \mathbf{y}_m := (\mathbf{y}_{m1}, \ldots, \mathbf{y}_{mD}),$$

so that the candidate functions are directly evaluated at the noisy data. Solving (2.1.1) for $\mathbf{F}$ then reduces to identifying a sparse weight matrix $\widehat{\mathbf{w}}$ such that

$$\dot{\mathbf{y}} \approx \Theta(\mathbf{y})\, \widehat{\mathbf{w}} \tag{2.1.3}$$

where $\dot{\mathbf{y}}$ is the numerical time derivative of the data $\mathbf{y}$. Sequential-thresholding least squares is then used to arrive at a sparse solution.

2.2 Weak SINDy

We approach the problem of system identification (2.1.3) from a non-standard perspective by utilizing the weak form of the differential equation. Recall that for any smooth test function $\phi : \mathbb{R} \to \mathbb{R}$ (absolutely continuous is sufficient) and interval $(a, b) \subset [0, T]$, equation (2.1.1) admits

the weak formulation

$$\phi(b)\mathbf{x}(b) - \phi(a)\mathbf{x}(a) - \int_a^b \phi'(u)\,\mathbf{x}(u)\,du = \int_a^b \phi(u)\,\mathbf{F}(\mathbf{x}(u))\,du, \qquad 0 \le a < b \le T. \tag{2.2.1}$$

With $\phi = 1$, we arrive at the integral equation of the dynamics explored in [121]. If we instead take ϕ to be non-constant and compactly supported in (a, b), we arrive at

$$-\int_a^b \phi'(u)\,\mathbf{x}(u)\,du = \int_a^b \phi(u)\,\mathbf{F}(\mathbf{x}(u))\,du. \tag{2.2.2}$$

Assuming a representation of the form (2.1.2), we then define the generalized residual $\mathcal{R}(\mathbf{w};\phi)$ for a given test function ϕ by replacing $\mathbf{F}$ with a candidate element from the span of $(f_j)_{j\in[J]}$ and $\mathbf{x}$ with $\mathbf{y}$ as follows:

$$\mathcal{R}(\mathbf{w};\phi) := \int_a^b \left(\phi'(u)\,\mathbf{y}(u) + \phi(u)\,\left(\sum_{j=1}^{J} \mathbf{w}_j\, f_j(\mathbf{y}(u))\right)\right) du. \tag{2.2.3}$$

Clearly, with $\mathbf{w} = \mathbf{w}^\star$ and $\mathbf{y} = \mathbf{x}(t)$ we have $\mathcal{R}(\mathbf{w};\phi) = 0$ for all ϕ compactly-supported in (a, b); however, $\mathbf{y}$ is a discrete set of data, hence (2.2.3) can at best be approximated numerically. Measurement noise then presents a significant barrier to accurate indentification of $\mathbf{w}^\star$.

2.2.1 Method Overview

For analogy with traditional Galerkin methods, consider the forward problem of solving a dynamical system such as (2.1.1) for $\mathbf{x}$. The Galerkin approach is to seek a solution $\mathbf{x}$ represented in a chosen trial basis $(f_j)_{j\in[J]}$ such that the residual $\mathcal{R}$, defined by

$$\mathcal{R} = \int \phi(t)(\dot{\mathbf{x}}(t) - \mathbf{F}(\mathbf{x}(t)))\,dt,$$

is minimized over all test functions ϕ living in the span of a given test function basis $(\phi_k)_{k\in[K]}$. If the trial and test function bases are known analytically, inner products of the form $\langle f_j, \phi_k\rangle$ appearing in the residual can be computed exactly. Thus, the computational error results only from representing the solution in a finite-dimensional function space.

The method we present here can be considered a data-driven Galerkin method of solving for $\mathbf{F}$ where the trial "basis" is given by the set of gridfunctions $(f_j(\mathbf{y}))_{j\in[J]}$ evaluated at the data and only the test-function basis $(\phi_k)_{k\in[K]}$ is known analytically. In this way, inner products appearing in $\mathcal{R}(\mathbf{w};\phi)$ must be approximated numerically, implying that the accuracy of the recovered weights $\widehat{\mathbf{w}}$ is ultimately limited by the quadrature scheme used to discretize inner products. Using Lemma 1 below, we show that the correct coefficients $\mathbf{w}^\star$ may be recovered to effective machine precision accuracy (given by the tolerance of the forward ODE solver) from noise-free trajectories $\mathbf{y}$ by

discretizing (2.2.2) using the trapezoidal rule and choosing ϕ to decay smoothly to zero at the boundaries of its support. Specifically, in this chapter we demonstrate this fact by choosing test functions from a particular family of unimodal piecewise polynomials $\mathcal{S}$ defined in (2.2.6).

Having chosen a quadrature scheme, the next accuracy barrier is presented by measurement noise, introducing randomness into the residuals $\mathcal{R}(\mathbf{w};\phi)$. Numerical integration then couples residuals $\mathcal{R}(\mathbf{w};\phi_1)$ and $\mathcal{R}(\mathbf{w};\phi_2)$ whenever ϕ_1 and ϕ_2 have overlapping support. In this way, $\mathcal{R}(\mathbf{w};\phi)$ does not have an ideal error structure for least squares, but may be amenable to generalized least squares. Below we analyze the distribution of the residuals $\mathcal{R}(\mathbf{w};\phi)$ to arrive at a generalized least squares approach where an approximate covariance matrix can be computed directly from the test functions. This analysis also suggests that placing test functions near steep gradients in the dynamics may improve recovery, hence we develop a derivative-free method for adaptively clustering test functions near steep gradients.

Remark 1. The weak formulation of the dynamics introduces a wealth of information: given M timepoints $\mathbf{t} = (t_m)_{m\in[M]}$, equation (2.2.2) affords $K = M(M-1)/2$ residuals over all possible supports $(a,b) \subset \mathbf{t}\times\mathbf{t}$ with $a < b$. Of course, one could also assimilate the responses of multiple families of test functions $\big(\{\phi_k^1\}_{k\in[K_1]}, \{\phi_k^2\}_{k\in[K_2]}, \dots\big)$; however, the computational complexity of such an exhaustive approach quickly becomes intractable. We stress that even with large noise, our proposed method identifies the correct nonlinearities with accurate weight recovery while keeping the number of test functions lower than the number of timepoints ($K < M$).

2.2.2 Algorithm: Weak SINDy

We state here the Weak SINDy algorithm in full generality. We propose a generalized least squares approach with approximate covariance matrix Σ. Below we derive a particular choice of Σ which utilizes the action of the test functions $(\phi_k)_{k\in[K]}$ on the data $\mathbf{y}$. Sequential-thresholding on the weight coefficients $\mathbf{w}$ with thresholding parameter λ is used to enforce sparsity, where $\lambda \leq \min_{\mathbf{w}^\star\neq 0}|\mathbf{w}^\star|$ is necessary for recovery. Lastly, an ℓ_2-regularization term with coefficient γ is included for problems involving rank deficiency. Methods of choosing optimal values of λ and γ directly from a given dataset do exist, for instance by selecting the optimal position in a Pareto front [34], however this is not the focus of our current study and thus we select values that work across multiple examples. Specifically, in the experiments below we set $\gamma = 0$ with the exception of the nonlinear pendulum and the five-dimensional linear system, examples which show that regularization can be used to discover dynamics from excessively large libraries. For noise-free data the algorithm is only weakly dependent on λ and so we use $\lambda = 0.001$, while for noisy data we set $\lambda = \frac{1}{4}\min_{\mathbf{w}^\star\neq 0}|\mathbf{w}^\star|$.

$\widehat{\mathbf{w}} = \mathbf{WSINDy}\big(\mathbf{y},\mathbf{t}\ ;\ (\phi_k)_{k\in[K]},\ (f_j)_{j\in[J]},\ \Sigma,\lambda,\gamma\big)$:

1. Construct matrix of trial gridfunctions $\Theta(\mathbf{y}) = \big[f_1(\mathbf{y})\,|\,\dots\,|\,f_J(\mathbf{y})\big]$

2. Construct integration matrices $\mathbf{V}$, $\mathbf{V}'$ such that

$$\mathbf{V}_{km} = \Delta t \phi_k(t_m), \quad \mathbf{V}'_{km} = \Delta t \phi'_k(t_m)$$

3. Compute Gram matrix $\mathbf{G} = \mathbf{V}\Theta(\mathbf{y})$ and right-hand side $\mathbf{b} = -\mathbf{V}'\mathbf{y}$ so that $\mathbf{G}_{kj} = \langle \phi_k, f_j(\mathbf{y}) \rangle$ and $\mathbf{b}_{kd} = -\langle \phi'_k, \mathbf{y}_d \rangle$

4. Solve the generalized least squares problem with ℓ_2-regularization

$$\widehat{\mathbf{w}} = \text{argmin}_{\mathbf{w}} \left\{ (\mathbf{G}\mathbf{w} - \mathbf{b})^T \Sigma^{-1} (\mathbf{G}\mathbf{w} - \mathbf{b}) + \gamma^2 \|\mathbf{w}\|_2^2 \right\},$$

   using sequential thresholding with parameter λ to enforce sparsity.

With this as our core algorithm, we can now consider a residual analysis (Section 2.2.3) leading to a generalized least squares framework. We can also develop theoretical results related to the test functions (Section 2.2.4), yielding a more thorough understanding of the impact of using uniform (Section 2.2.4) and adaptive (Section 2.2.4) placement of test functions along the time axis.

2.2.3 Residual Analysis

Performance of WSINDy is determined by the behavior of the residuals

$$\mathcal{R}(\mathbf{w}; \phi_k) := (\mathbf{G}\mathbf{w} - \mathbf{b})_k \ \in \ \mathbb{R}^{1 \times D},$$

denoted $\mathcal{R}(\mathbf{w}) \in \mathbb{R}^{K \times D}$ for the entire residual matrix. Here we analyze the residual for autonomous $\mathbf{F}$ to highlight key aspects for future analysis, as well as to arrive at an appropriate choice of approximate covariance Σ. We also provide a heuristic argument in favor of placing test functions near steep gradients in the dynamics.

A key difficulty in recovering the true weights $\mathbf{w}^\star$ is that for nonlinear systems the residual evaluated at the true weights $\mathbf{w}^\star$ is biased: $\mathbb{E}[\mathcal{R}(\mathbf{w}^\star)] \neq 0$. Any minimization of $\mathcal{R}$ thus introduces a bias in the recovered weights $\widehat{\mathbf{w}}$. Nevertheless, we can understand how different test functions impact the residual by linearizing around the true trajectory $\mathbf{x}(t)$ and isolating the dominant error terms:

$$\begin{aligned}
\mathcal{R}(\mathbf{w}; \phi_k) &= \langle \phi_k, \ \Theta(\mathbf{y})\mathbf{w} \rangle + \left\langle \phi'_k, \ \mathbf{y} \right\rangle \\
&= \langle \phi_k, \ \Theta(\mathbf{y})(\mathbf{w} - \mathbf{w}^\star) \rangle + \langle \phi_k, \ \Theta(\mathbf{y})\mathbf{w}^\star \rangle + \left\langle \phi'_k, \ \mathbf{y} \right\rangle \\
&= \langle \phi_k, \ \Theta(\mathbf{y})(\mathbf{w} - \mathbf{w}^\star) \rangle + \langle \phi_k, \ \mathbf{F}(\mathbf{y}) - \mathbf{F}(\mathbf{x}) \rangle + \left\langle \phi'_k, \ \epsilon \right\rangle + I_k \\
&= \underbrace{\langle \phi_k, \ \Theta(\mathbf{y})(\mathbf{w} - \mathbf{w}^\star) \rangle}_{R_1} + \underbrace{\langle \phi_k, \ \epsilon \nabla \mathbf{F}(\mathbf{x}) \rangle}_{R_2} + \underbrace{\left\langle \phi'_k, \ \epsilon \right\rangle}_{R_3} + I_k + \mathcal{O}(\epsilon^2)
\end{aligned}$$

where $\nabla F(\mathbf{x})_{dd'} = \frac{\partial \mathbf{F}_{d'}}{\partial \mathbf{x}_d}(\mathbf{x})$. The errors manifest in the following ways:

- R_1 is the misfit between $\mathbf{w}$ and $\mathbf{w}^\star$
- R_2 results from measurement error in trial gridfunctions: $f_j(\mathbf{y}) = f_j(\mathbf{x}+\epsilon) \neq f_j(\mathbf{x})$
- R_3 results from replacing $\mathbf{x}$ with $\mathbf{y} = \mathbf{x} + \epsilon$ in the left-hand side of (2.2.2)
- I_k is a deterministic integration error
- $\mathcal{O}(\epsilon^2)$ is the remainder term in the truncated Taylor expansion of $\mathbf{F}(\mathbf{y})$ around $\mathbf{x}$:

$$\mathbf{F}(\mathbf{y}_m) = \mathbf{F}(\mathbf{x}(t_m)) + \epsilon_m \nabla \mathbf{F}(\mathbf{x}(t_m)) + \mathcal{O}(|\epsilon_m|^2).$$

Clearly, recovery of $\mathbf{F}$ when $\epsilon = 0$ is straightforward: R_1 and I_k are the only error terms, thus one only needs to select a quadrature scheme that ensures that the integration error I_k is negligible and $\widehat{\mathbf{w}} = \mathbf{w}^\star$ will be the minimizer. A primary focus of this study is the use of a specific family of piecewise polynomial test functions $\mathcal{S}$ defined below for which the trapezoidal rule is highly accurate (see Lemma 1). Figure 2.2 demonstrates this fact on noise-free data.

For $\epsilon > 0$, accurate recovery of $\mathbf{F}$ requires one to choose hyperparameters that exemplify the true misfit term R_1 by enforcing that the other error terms are of lower order. We look for $(\phi_k)_{k\in[K]}$ and $\Sigma = \mathbf{C}\mathbf{C}^T$ that approximately enforce $\mathbf{C}^{-1}\mathcal{R}(\mathbf{w}^\star) \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, justifying the least squares approach. In the next subsection we address the issue of approximating the covariance matrix, providing justification for using $\Sigma = \mathbf{V}'(\mathbf{V}')^T$. The following subsection provides a heuristic argument for how to reduce corruption from the error terms R_2 and R_3 by placing test functions near steep gradients in the data.

Approximate Covariance Σ

Neglecting the deterministic integration error, which can be made small (see Lemma 1 below), and higher-order noise terms, the residual evaluated at the true weights is approximately

$$\mathcal{R}(\mathbf{w}^\star; \phi_k) \approx R_2 + R_3$$

where $\mathbb{E}[R_2] = \mathbb{E}[R_3] = (0, \ldots, 0)$ implies that $\mathbb{E}[\mathcal{R}(\mathbf{w}^\star)] = \mathbf{0}$ to leading order. Given the variances

$$\mathbb{V}[R_2] = \mathbb{V}[\langle \phi_k, \epsilon \nabla \mathbf{F}(\mathbf{x})\rangle] = \Delta t\, \sigma^2 \left(\left\| \phi_k \left| \nabla \mathbf{F}_1(\mathbf{x}) \right| \right\|_2^2, \ldots, \left\| \phi_k \left| \nabla \mathbf{F}_D(\mathbf{x}) \right| \right\|_2^2 \right)$$

and

$$\mathbb{V}[R_3] = \mathbb{V}\left[\left\langle \phi_k', \epsilon \right\rangle\right] = \Delta t\, \sigma^2 \left(\left\| \phi_k' \right\|_2^2, \ldots, \left\| \phi_k' \right\|_2^2 \right),$$

the true distribution of $\mathcal{R}(\mathbf{w}^\star)$ depends on $\mathbf{F}$, which is not known *a priori*. If it holds that $\|\phi_k'\|_2 \gg \|\phi_k\,|\nabla \mathbf{F}_d(\mathbf{x})|\|_2$, $d \in [D]$, a leading order approximation to $\mathrm{Cov}(\mathcal{R}(\mathbf{w}^\star))$ is

$$\Sigma := \mathbf{V}'\left(\mathbf{V}'\right)^T \propto \mathrm{Cov}(R_3),$$

using that $\mathrm{Cov}(R_3)_{ij} = \Delta t \sigma^2 \left\langle \phi_i', \phi_j' \right\rangle$. For this reason, we employ localized test functions and adopt the heuristic $\Sigma = \mathbf{V}'(\mathbf{V}')^T$ below.

Adaptive Refinement

Next we show that by localizing ϕ_k around large $|\dot{\mathbf{x}}|$, we get an approximate cancellation of the error terms R_2 and R_3. Consider the one-dimensional case ($D = 1$) where m is an arbitrary time index and $\mathbf{y}_m = \mathbf{x}(t_m) + \epsilon$ is an observation. When $|\dot{\mathbf{x}}(t_m)|$ is large compared to ϵ, we approximately have

$$\mathbf{y}_m = \mathbf{x}(t_m) + \epsilon_m \approx \mathbf{x}(t_m + \delta t) \approx \mathbf{x}(t_m) + \delta t \mathbf{F}(\mathbf{x}(t_m)) \tag{2.2.4}$$

for some small δt, i.e. the perturbed value $\mathbf{y}_m$ lands close to the true trajectory $\mathbf{x}$ at the time $t_m + \delta t$. To understand the heuristic behind this approximation, let $t_m + \delta t$ be the point of intersection between the tangent line to $\mathbf{x}(t)$ at t_m and $\mathbf{x}(t_m) + \epsilon$. Then

$$\delta t = \frac{\epsilon}{\dot{\mathbf{x}}(t_m)},$$

hence $|\dot{\mathbf{x}}(t_m)| \gg \epsilon$ implies that $\mathbf{x}(t_m) + \epsilon$ will approximately lie on the true trajectory. As well, regions where $|\dot{\mathbf{x}}(t_m)|$ is small will not yield accurate recovery in the case of noisy data, since perturbations are more likely to exit the relevant region of phase space. If we linearize $\mathbf{F}$ using the approximation (2.2.4) we get

$$\mathbf{F}(\mathbf{y}_m) \approx \mathbf{F}(\mathbf{x}(t_m)) + \delta t \mathbf{F}'(\mathbf{x}(t_m))\mathbf{F}(\mathbf{x}(t_m)) = \mathbf{F}(\mathbf{x}(t_m)) + \delta t \ddot{\mathbf{x}}(t_m). \tag{2.2.5}$$

Assuming ϕ_k is sufficiently localized around t_m, (2.2.4) also implies that

$$\left\langle \phi_k', \mathbf{x} \right\rangle + \underbrace{\left\langle \phi_k', \epsilon \right\rangle}_{R_3} = \left\langle \phi_k', \mathbf{y} \right\rangle \approx \left\langle \phi_k', \mathbf{x} \right\rangle + \delta t \left\langle \phi_k', \mathbf{F}(\mathbf{x}) \right\rangle,$$

hence $R_3 \approx \delta t \left\langle \phi_k', \mathbf{F}(\mathbf{x}) \right\rangle$, while (2.2.5) implies

$$\begin{aligned}
\langle \phi_k, \Theta(\mathbf{y})\mathbf{w} \rangle &= \underbrace{\langle \phi_k, \Theta(\mathbf{y})(\mathbf{w} - \mathbf{w}^\star) \rangle}_{=R_1} + \langle \phi_k, \mathbf{F}(\mathbf{y}) \rangle \\
&\approx \langle \phi_k, \Theta(\mathbf{y})(\mathbf{w} - \mathbf{w}^\star) \rangle + \langle \phi_k, \mathbf{F}(\mathbf{x}) \rangle + \underbrace{\delta t \left\langle \phi_k, \ddot{\mathbf{x}} \right\rangle}_{\approx R_2} \\
&= \langle \phi_k, \Theta(\mathbf{y})(\mathbf{w} - \mathbf{w}^\star) \rangle + \langle \phi_k, \mathbf{F}(\mathbf{x}) \rangle - \delta t \left\langle \phi_k', \mathbf{F}(\mathbf{x}) \right\rangle
\end{aligned}$$

having integrated by parts. Collecting the terms together yields that the residual takes the form

$$\mathcal{R}(\mathbf{w};\phi_k) = \langle \phi_k', \mathbf{y} \rangle + \langle \phi_k, \Theta(\mathbf{y})\mathbf{w} \rangle \approx R_1,$$

and we see that R_2 and R_3 have effectively cancelled. In higher dimensions this interpretation does not appear to be as illuminating, but nevertheless, for any given coordinate $\mathbf{x}_d$, it does hold that terms in the error expansion vanish around points t_m where $|\dot{\mathbf{x}}_d|$ is large, precisely because $\mathbf{x}_d(t_m) + \epsilon \approx \mathbf{x}_d(t_m + \delta t)$.

2.2.4 Test Function Basis $(\phi_k)_{k\in[K]}$

Here we introduce a test function space $\mathcal{S}$ and quadrature scheme to minimize integration errors and enact the heuristic arguments above, which rely on ϕ_k having fast decay to its support boundaries and sufficiently localized to ensure $\|\phi_k'\|_2^2 \gg \|\phi_k\|_2^2$. We define the space $\mathcal{S}$ of unimodal piecewise polynomials of the form

$$\phi(t) = \begin{cases} C(t-a)^p(b-t)^q & t \in [a,b], \\ 0 & \text{otherwise}, \end{cases} \tag{2.2.6}$$

where $(a,b) \subset \mathbf{t} \times \mathbf{t}$ satisfies $a < b$ and $p, q \geq 1$. The normalization

$$C = \frac{1}{p^p q^q} \left(\frac{p+q}{b-a} \right)^{p+q}$$

ensures that $\|\phi\|_\infty = 1$. Functions $\phi \in \mathcal{S}$ are non-negative, unimodal, and compactly supported in $[0,T]$ with $\lfloor \min\{p,q\} \rfloor - 1$ continuous derivatives. Larger p and q imply faster decay towards the endpoints of the support. For $p = q$, we refer to p as the degree of ϕ.

To ensure the integration error in approximating inner products $\langle f_j, \phi_k \rangle$ is negligible, we rely on the following lemma, which provides a bound on the error in discretizing the weak derivative relation

$$-\int \phi' f \, dt = \int \phi f' \, dt \tag{2.2.7}$$

using the trapezoidal rule for compactly supported ϕ. Following the lemma we introduce two strategies for choosing the parameters of the test functions $(\phi_k)_{k\in[K]} \subset \mathcal{S}$.

Lemma 1 (Numerical Error in Weak Derivatives)**.** *Let f, ϕ have continuous derivatives of order p and define $t_j = a + j\frac{b-a}{N} = a + j\Delta t$. If ϕ has roots $\phi(a) = \phi(b) = 0$ of multiplicity p, then*

$$\frac{\Delta t}{2} \sum_{j=0}^{N-1} \left[g(t_j) + g(t_{j+1}) \right] = \mathcal{O}(\Delta t^{p+1}), \tag{2.2.8}$$

where $g(t) = \phi'(t)f(t) + \phi(t)f'(t)$*. In other words, the composite trapezoidal rule discretizes the weak derivative relation* (2.2.7) *to order* $p+1$*.*

Proof. This is a simple consequence of the Euler-Maclaurin formula. If $g : [a, b] \to \mathbb{C}$ is a smooth function, then the following asymptotic expansion holds:

$$\frac{\Delta t}{2}\sum_{j=0}^{N-1}[g(t_j) + g(t_{j+1})] \sim \int_a^b g(t)\,dt + \sum_{k=1}^{\infty}\frac{\Delta t^{2k}B_{2k}}{(2k)!}\left(g^{(2k-1)}(b) - g^{(2k-1)}(a)\right),$$

where B_{2k} are the Bernoulli numbers. The asymptotic expansion provides corrections to the trapezoidal rule that realize machine precision accuracy up until a certain value of k, after which terms in the expansion grow and the series diverges [36, Ch. 3]. In our case, $g(t) = \phi'(t)f(t) + \phi(t)f'(t)$ where the root conditions on ϕ imply that

$$\int_a^b g(t)\,dt = 0 \quad \text{and} \quad g^{(k)}(b) = g^{(k)}(a) = 0, \quad 0 \le k \le p-1.$$

So for p odd, we have that

$$\begin{aligned}\frac{\Delta t}{2}\sum_{j=0}^{N-1}[g(t_j) + g(t_{j+1})] &\sim \sum_{k=(p+1)/2}^{\infty}\frac{\Delta t^{2k}B_{2k}}{(2k)!}\left(g^{(2k-1)}(b) - g^{(2k-1)}(a)\right)\\ &= \frac{B_{p+1}}{(p+1)!}(\phi^{(p)}(b)f(b) - \phi^{(p)}(a)f(a))\Delta t^{p+1} + \mathcal{O}\left(\Delta t^{p+2}\right).\end{aligned}$$

For even p, the leading term is $\mathcal{O}(\Delta t^{p+2})$ with a slightly different coefficient. □

For $\phi \in \mathcal{S}$ with $p = q$, the exact leading order error in term in (2.2.8) is

$$\frac{2^p\,B_{p+1}}{p+1}\big(f(b) - f(a)\big)\Delta t^{p+1}, \tag{2.2.9}$$

which is negligible for a wide range of reasonable p and Δt values. The Bernoulli numbers eventually start growing like p^p, but for smaller values of p they are moderate. For instance, with $\Delta t = 0.1$ and $f(b) - f(a) = 1$, this error term is $o(1)$ up until $p = 85$, where it takes the value 0.495352, while for $\Delta t = 0.01$, the error is below machine precision for all p between 7 and 819. For these reasons, in what follows we choose test functions $(\phi_k)_{k\in[K]} \subset \mathcal{S}$ and discretize all integrals using the trapezoidal rule. Unless otherwise stated, each function ϕ_k satsifies $p = q$ and so is fully determined by the tuple $\{p_k, a_k, b_k\}$ indicating its polynomial degree and support. In the next two subsections we propose two different strategies for determining ϕ_k using the data $\mathbf{y}$.

Strategy 1: Uniform Grid

The simplest strategy for choosing a basis of test functions $(\phi_k)_{k\in[K]} \subset \mathcal{S}$ is to place ϕ_k uniformly on the interval $[0,T]$ with fixed degree p and fixed support size

$$L := \#\{\mathbf{t} \cap \mathrm{supp}(\phi_k)\}$$

(i.e. L is the number of timepoints in $\mathbf{t}$ that ϕ_k is supported on). The triple (L,p,K) then defines the scheme, where each piece effects the distribution of the residual $\mathcal{R}(\mathbf{w})$.

Step 1: Choosing L: Heuristically, the support size of ϕ_k relates to the Fourier transform of the data. If $\mathrm{supp}(\phi_k)$ is small compared to the dominant wavemodes in the dynamics, then high-frequency noise will dominate the values of the inner products $\langle \phi_k', \mathbf{y}\rangle$. If $\mathrm{supp}(\phi_k)$ is much larger than the dominant wavemodes, then too much averaging may occur, leading to unresolved dynamics. A natural choice is then to set L equal to the period of a known active wavemode[1] k:

$$L = \left\lfloor \frac{1}{\Delta t}\frac{2\pi}{(2\pi T/k)} \right\rfloor = \left\lfloor \frac{M}{k} \right\rfloor.$$

In the noise-free and small-noise experiments below we set $L = \lfloor \frac{M}{25} \rfloor$ and leave optimal selection of L based on Fourier analysis to future work.

Step 2: Determining p: In light of the derivation above of the approximate covariance matrix $\Sigma = \mathbf{V}'(\mathbf{V}')^T$, we define the parameter $\rho := \|\phi_k'\|_2 \,/\, \|\phi_k\|_2$, which serves as an estimate for the ratio $\sqrt{\mathbb{V}[R_3]/\mathbb{V}[R_2]}$ between the standard deviations of the two dominant error terms R_3 and R_2 in the residual $\mathcal{R}(\mathbf{w}^\star)$. Larger ρ indicates better agreement with the approximate covariance matrix Σ, since $\Sigma \propto \mathrm{Cov}(R_3)$. Furthermore, for $\phi_k \in \mathcal{S}$ we have the exact formula

$$\rho^2 = \frac{8p^2}{(b-a)^2}\left(\frac{\Gamma(2p-1)\Gamma(2p+\frac{1}{2})}{\Gamma(2p+1)\Gamma(2p+\frac{3}{2})}\right) = \frac{p}{(b-a)^2}\left(\frac{4p+1}{p-\frac{1}{2}}\right),$$

where $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\,dt$ is the Gamma function. Given $\rho^2 \geq (5+2\sqrt{6})/(b-a)^2$, a polynomial degree p may be selected from ρ using the formula

$$p = \left\lfloor \frac{1}{8}\left(((b-a)^2\rho^2 - 1) + \sqrt{((b-a)^2\rho^2-1)^2 - 8(b-a)^2\rho^2}\right)\right\rfloor.$$

Step 3: Determining K: Next we introduce the shift parameter $s \in [0,1]$ defined by

$$s := \phi_k(t^*) \;\text{ s.t. }\; \phi_k(t^*) = \phi_{k+1}(t^*),$$

[1] Such that $\mathcal{F}_k(\mathbf{y}) := \sum_{j=0}^{M-1} \mathbf{y}_m e^{-2\pi i j k/M}$ is not negligible.

which determines K from p and L. In words, s is the height of intersection between ϕ_k and ϕ_{k+1} and measures the amount of overlap between successive test functions. More overlap increases the correlation between rows in the residual $\mathcal{R}(\mathbf{w})$ and hence leads to larger off-diagonal elements in the covariance matrix Σ. Larger s implies that neighboring functions overlap on more points, with $s = 1$ indicating that $\phi_k = \phi_{k+1}$. Specifically, neighboring test functions overlap on $\lfloor L(1 - \sqrt{1 - s^{1/p}}) \rfloor$ timepoints. In Figures 2.3 and 2.4 we vary the parameters ρ and s and observe that results agree with intuition: larger ρ (better agreement with Σ) and larger s (more test functions) lead to better recovery of $\mathbf{w}^\star$. We summarize the uniform grid algorithm below.

$\widehat{\mathbf{w}} = \mathbf{WSINDy\_UG}\left(\mathbf{y}, \mathbf{t}\ ;\ (f_j)_{j\in[J]}, L, \rho, s, \lambda, \gamma\right)$:

1. Construct matrix of trial gridfunctions $\Theta(\mathbf{y}) = \left[f_1(\mathbf{y}) \,|\, \dots \,|\, f_J(\mathbf{y})\right]$
2. Construct integration matrices $\mathbf{V}$, $\mathbf{V}'$ such that

$$\mathbf{V}_{km} = \Delta t \phi_k(t_m), \quad \mathbf{V}'_{km} = \Delta t \phi'_k(t_m)$$

   with the test functions $(\phi_k)_{k\in[K]}$ determined by L, ρ, s as described above
3. Compute Gram matrix $\mathbf{G} = \mathbf{V}\Theta(\mathbf{y})$ and right-hand side $\mathbf{b} = -\mathbf{V}'\mathbf{y}$ so that $\mathbf{G}_{kj} = \langle \phi_k, f_j(\mathbf{y}) \rangle$ and $\mathbf{b}_{kd} = -\langle \phi'_k, \mathbf{y}_d \rangle$
4. Compute approximate covariance and Cholesky factorization $\Sigma = \mathbf{V}'(\mathbf{V}')^T = \mathbf{C}\mathbf{C}^T$
5. Solve the generalized least squares problem with ℓ_2-regularization

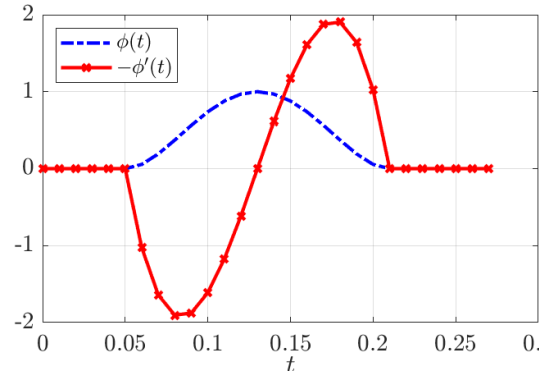
$$\widehat{\mathbf{w}} = \text{argmin}_{\mathbf{w}} \left\{ (\mathbf{G}\mathbf{w} - \mathbf{b})^T \Sigma^{-1} (\mathbf{G}\mathbf{w} - \mathbf{b}) + \gamma^2 \|\mathbf{w}\|_2^2 \right\},$$

   using sequential thresholding with parameter λ to enforce sparsity.

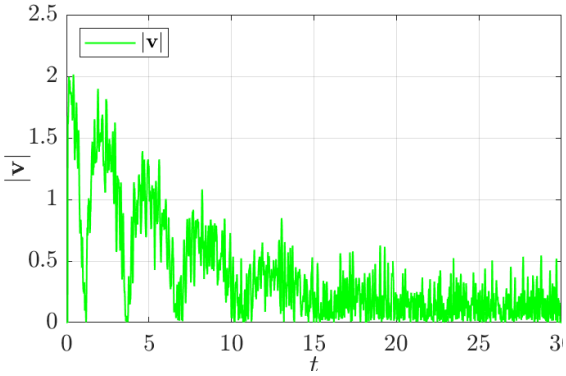
Strategy 2: Adaptive Grid

Motivated by the arguments above, we now introduce an algorithm for constructing a test function basis localized near points of large change in the dynamics. This occurs in three steps: 1) construct a weak approximation to the derivative of the dynamics $\mathbf{v} \approx \dot{\mathbf{x}}$, 2) sample K points $\mathbf{c}$ from a cumulative distribution ψ with density proportional to the total variation $|\mathbf{v}|$, 3) construct test functions centered at $\mathbf{c}$ using a width-at-half-max parameter r_{whm} to determine the parameters (p_k, a_k, b_k) of each function ϕ_k. Each of these steps is numerically stable and carried out independently along each coordinate of the dynamics. A visual diagram is provided in Figure 2.1.

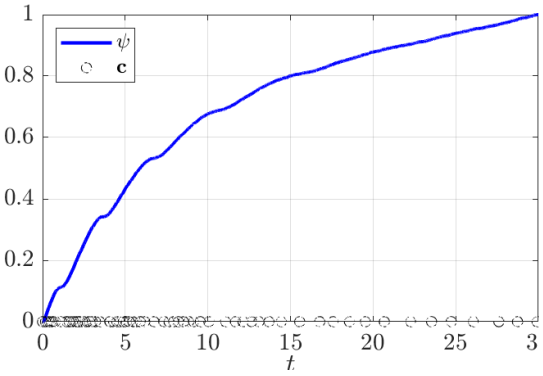
Step 1: Weak Derivative Approximation: Define $\mathbf{v} := -\mathbf{V}'_w \mathbf{y}$, where the matrix $-\mathbf{V}'_w$ enacts a linear convolution with the derivative of a chosen test function $\phi \in \mathcal{S}$ of degree p_w and support size



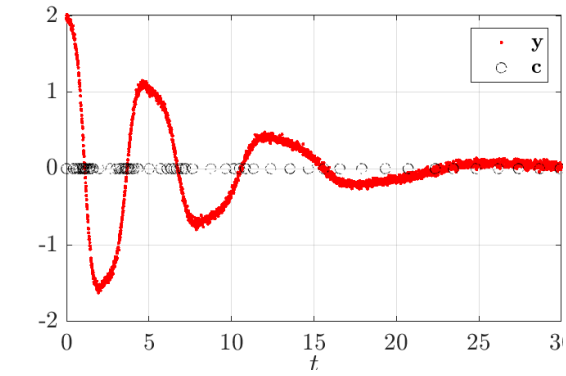
Test function ϕ and derivative $-\phi'$ used to compute $\mathbf{v}$



Approximate total variation $|\mathbf{v}|$



Cumulative distribution $\psi = \int^t |\mathbf{v}|\, dt$



Data $\mathbf{y}$ *from the Duffing equation and resulting grid* $\mathbf{c}$

Figure 2.1: Adaptive grid construction used on data from the Duffing equation with 10% noise ($\sigma_{NR} = 0.1$). As desired, the centers $\mathbf{c}$ are clustered near steep gradients in the dynamics despite large measurement noise. (Note $-\phi(t)'/10$ is plotted in the upper-left instead of $-\phi(t)'$ in order to visualize both ϕ and ϕ'.)

L_w so that

$$\mathbf{v}_m = -\left\langle \phi', \mathbf{y} \right\rangle = \langle \phi, \dot{\mathbf{y}} \rangle \approx \dot{\mathbf{y}}_m.$$

The parameters L_w and p_w are chosen by the user, with $L_w = 5$ and $p_w \geq 2$ corresponding to taking a centered finite difference derivative with 3-point stencil. Smaller p_w results in more smoothing and minimizes the corruption from noise while still accurately locating steep gradients in the dynamics. For the examples below we arbitrarily[2] use $p_w = 2$ and $L_w = 17$.

Step 2: Selecting $\boldsymbol{c}$*:* Having computed $\mathbf{v}$, define ψ to be the cumulative sum of $|\mathbf{v}|$ normalized so that $\max \psi = 1$. In this way ψ is a valid cumulative distribution function with density proportional to the total variation of $\mathbf{y}$. We then find $\mathbf{c}$ by sampling from ψ. Let $U = [0, \frac{1}{K}, \frac{2}{K}, \ldots, \frac{K-1}{K}]$ with K being the number of the test functions, we then define $\mathbf{c} = \psi^{-1}(U)$, or numerically,

$$c_k = \min\{t \in \mathbf{t} \ : \ \psi(t) \geq U_k\}.$$

This stage requires the user to select the number of test functions K.

Step 3: Construction of Test Functions $(\phi_k)_{k\in[K]}$*:* Having chosen the location c_k of the centerpoint for each test function ϕ_k, we are left to choose the degree p_k of the polynomial and the supports $[a_k, b_k]$. The degree is chosen according to the width-at-half-max parameter r_{whm}, which specifies the difference in timepoints between each center c_k and $\arg_t\{\phi_k(t) = 1/2\}$, while the supports are chosen such that $\phi_k(b_k - \Delta t) = 10^{-16}$. This gives us a nonlinear system of two equations in two unknowns which can be easily solved (i.e. using MATLAB's `fzero`). This can be done for one reference test functions and the rest of the weights obtained by translation. The optimal value of r_{whm} depends on the timescales of the dynamics and can be chosen from the data using the Fourier transform as in the uniform grid case, however for simplicity we set $r_{whm} = \lfloor M/100 \rfloor$ in the large noise examples below.

The adaptive grid Weak SINDy algorithm is summarized as follows:

$\widehat{\mathbf{w}} = \mathbf{WSINDy\_AG}\left(\mathbf{y}, \mathbf{t} \ ; \ (f_j)_{j\in[J]}, p_w, L_w, K, r_{whm}, \lambda, \gamma\right)$:

1. Construct matrix of trial gridfunctions $\Theta(\mathbf{y}) = \left[f_1(\mathbf{y}) \,|\, \ldots \,|\, f_J(\mathbf{y})\right]$
2. Construct integration matrices $\mathbf{V}$, $\mathbf{V}'$ such that

$$\mathbf{V}_{km} = \Delta t \phi_k(t_m), \quad \mathbf{V}'_{km} = \Delta t \phi'_k(t_m),$$

with test functions $(\phi_k)_{k\in[K]}$ determined by p_w, L_w, K, r_{whm} as described above

[2]We find that a lower-degree test function with small support effectively locates steep gradients in noisy trajectories.

3. Compute Gram matrix $\mathbf{G} = \mathbf{V}\Theta(\mathbf{y})$ and right-hand side $\mathbf{b} = -\mathbf{V}'\mathbf{y}$ so that $\mathbf{G}_{kj} = \langle \phi_k, f_j(\mathbf{y}) \rangle$ and $\mathbf{b}_{kd} = -\langle \phi_k', \mathbf{y}_d \rangle$

4. Compute approximate covariance and Cholesky factorization $\Sigma = \mathbf{V}'(\mathbf{V}')^T = \mathbf{C}\mathbf{C}^T$

5. Solve the generalized least squares problem with ℓ_2-regularization

$$\widehat{\mathbf{w}} = \text{argmin}_{\mathbf{w}} \left\{ (\mathbf{G}\mathbf{w} - \mathbf{b})^T \Sigma^{-1} (\mathbf{G}\mathbf{w} - \mathbf{b}) + \gamma^2 \|\mathbf{w}\|_2^2 \right\},$$

using sequential thresholding with parameter λ to enforce sparsity.

2.3 Numerical Experiments

We now show that WSINDy is capable of recovering the correct dynamics to high accuracy over a range of noise levels. We examine the systems in Table 2.1 which exhibit several canonical dynamics, namely growth and decay, nonlinear oscillations and chaotic dynamics, in dimensions $D \in \{2, 3, 5\}$. To generate true trajectory data we use MATLAB's `ode45` with absolute and relative tolerance 10^{-10} and collect M samples uniformly[3] in time with sampling rate Δt. The parameters M and Δt are chosen to provide a balance between illustrating ODE behaviors and avoiding an overabundance of observations. Gaussian white noise with mean zero and variance σ^2 is added to the exact trajectories, where σ is computed by specifying a noise ratio σ_{NR} and setting

$$\sigma = \sigma_{NR} \frac{\|\mathbf{x}\|_F}{\sqrt{MD}} \tag{2.3.1}$$

where the Frobenius norm of a matrix $\mathbf{x} \in \mathbb{R}^{M \times D}$ is defined by $\|\mathbf{x}\|_F := \sqrt{\sum_{m=1}^{M} \sum_{d=1}^{D} |\mathbf{x}_{md}|^2}$. The ratio of noise to signal is then approximately equal to the square root of the variance: $\|\epsilon\|_F / \|\mathbf{x}\|_F \approx \sigma$.

We measure the accuracy in the recovered dynamical system using the relative $\|\cdot\|_F$ error in the recovered coefficients,

$$E_2(\widehat{\mathbf{w}}) = \frac{\|\widehat{\mathbf{w}} - \mathbf{w}^\star\|_F}{\|\mathbf{w}^\star\|_F}, \tag{2.3.2}$$

and the relative $\|\cdot\|_F$ error between the noise-free data $\mathbf{x}$ and the data-driven dynamics $\mathbf{x}_{dd}$ along the same time points:

$$\mathcal{E}_2(\mathbf{x}_{dd}) = \frac{\|\mathbf{x}_{dd} - \mathbf{x}\|_F}{\|\mathbf{x}\|_F}. \tag{2.3.3}$$

The collection of ODEs in Table 2.1 are all first-order autonomous systems, however they exhibit a diverse range of dynamics. The Linear 5D system (for $\beta < 0$) and Duffing's equation are both examples of damped oscillators, showing that WSINDy is able to discern whether such motion

[3] We leave a detailed study of non-uniform time sampling to future work.

Name	Governing Equations	M	Δt
Linear 5D	$\begin{cases} \dot{x}_1 = -x_5 - \beta x_1 + x_2, \\ \dot{x}_i = -x_{i-1} - \beta x_i + x_{i+1}, \quad i = 2,3,4 \\ \dot{x}_5 = -x_4 - \beta x_5 + x_1 \end{cases}$	1401	0.025
Duffing	$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -0.2x_2 - 0.2x_1 - \beta x_1^3 \end{cases}$	3001	0.01
Van der Pol	$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = \beta x_2(1 - x_1^2) - x_1 \end{cases}$	3001	0.01
Lotka-Volterra	$\begin{cases} \dot{x}_1 = 3x_1 - \beta x_1 x_2, \\ \dot{x}_2 = \beta x_1 x_2 - 6x_2 \end{cases}$	1001	0.01
Nonlinear Pendulum	$\begin{cases} \dot{x}_1 = x_2, \\ \dot{x}_2 = -\sin(x_1) \end{cases}$	501	0.1
Lorenz	$\begin{cases} \dot{x}_1 = 10(x_2 - x_1), \\ \dot{x}_2 = x_1(28 - x_3) - x_2, \\ \dot{x}_3 = x_1 x_2 - \frac{8}{3} x_3 \end{cases}$	10001	0.001

Table 2.1: ODEs used in numerical experiments. For Linear 5D, Duffing, Van der Pol and Lotka-Volterra we measure the accuracy in the recovered system as the parameter β varies (see Table 2.2).

is governed by linear or nonlinear coupling between variables. For $\beta > 0$, the Linear 5D system exhibits exponential growth. The Van der Pol oscillator, Lotka-Volterra system and nonlinear pendulum demonstrate that a stable limit cycle with abrupt changes may manifest from vastly different nonlinear mechanisms, which turn out to be identifiable using the weak form. Finally, the Lorenz system exhibits deterministic chaos and hence the dynamics cover a wide range of Fourier modes, which easily become corrupted with noise.

2.3.1 Noise-Free Data

The goal of the following noise-free experiments is to demonstrate convergence of the recovered weights $\widehat{\mathbf{w}}$ to the true weights $\mathbf{w}^\star$ to within the accuracy tolerance of the ODE solver (fixed 10^{-10} throughout). In light of Lemma 1, this should occur as the decay rate of the test functions $(\phi_k)_{k\in[K]}$ is increased, which for test functions in class $\mathcal{S}$ (see equation (2.2.6)) is realized by increasing the polynomial degree p. Hence, over the range of parameter values in Table 2.2, for each system we test convergence as p increases. We use the uniform grid approach with shift parameter s chosen such that the number of test functions equals to the number of trial functions ($K = J$), resulting in square Gram matrices $\mathbf{G} = \mathbf{V}\Theta(\mathbf{y})$. The support of the basis functions along the timegrid $\mathbf{t}$ is set to $L = \lfloor \frac{M}{25} \rfloor$ points. The data-driven trial basis $(f_j)_{j\in[J]}$ includes all monomials in the state variables up to degree 5 as well as the trigonometric terms $\cos(n\mathbf{y}_d)$, $\sin(n\mathbf{y}_d)$ for $n = 1, 2$ and

ODE	β	$\mathbf{x}(0)$	L	ΔL	$J(=K)$
Linear 5D	$(-0.3, -0.2, -0.1, 0.1\}$	$(10, 0, 0, 0, 0)^T$	57	5	252
Duffing	$(0.01, 0.1, 1, 10)$	$(0, 2)^T$	121	99	29
Van der Pol	$(0.01, 0.1, 1, 10)$	$(0, 1)^T$	121	99	29
Lotka-Volterra	$(0.005, 0.01, 0.1, 1)$	$(1, 1)^T$	41	33	29
Pendulum	—	$x_2(0) = 0$, $x_1(0) \in \{\frac{15}{16}\pi, \frac{10}{16}\pi, \frac{5}{16}\pi, \frac{1}{16}\pi\}$	21	16	29
Lorenz	—	$\sim U_{[-15,15]^2\times[10,40]}$	401	141	68

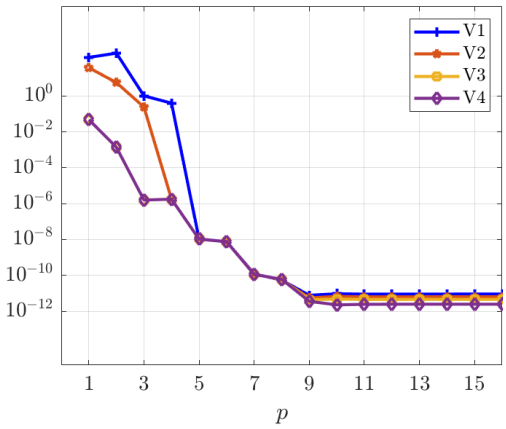
Table 2.2: Specifications for parameters used in illustrating simulations in Figure 2.2.

$d \in [D]$. We set the regularization parameter to zero ($\gamma = 0$), with the exception of the nonlinear pendulum where $\gamma = 10^{-8}$, and the sparsity threshold to $\lambda = 0.001$. We note that a non-zero γ is always necessary to discover the nonlinear pendulum from combined trigonometric and polynomial libraries since $\sin(x_1)$ is well-approximated by polynomial terms, however the same is not true for low-order polynomial systems. In cases considered here, sequential thresholding successfully removes trigonometric library terms for ODE systems with polynomial dynamics despite initially ill-conditioned Gram matrices $\mathbf{G}$ resulting from combining polynomial and trigonometric terms.

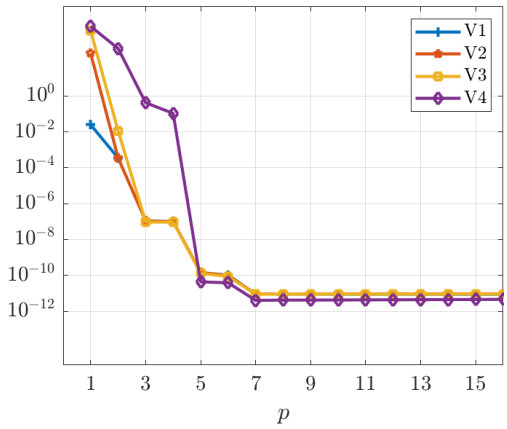
Figure 2.2 shows that in the limit of large p, WSINDy recovers the correct weight matrix $\mathbf{w}^\star$ of each system in Table 2.1 to an accuracy of $\mathcal{O}(10^{-10})$. For the Linear 5D system, we vary the growth/decay parameter, showing that the system is identifiable to high accuracy despite an excessively large trial library (252 terms). For Duffing's equation and the Van der Pol oscillator, the same convergence trend is observed for β values spanning several orders of magnitude. Accuracy is slightly worse for the Lotka-Volterra equation when $\beta = 0.005$, which corresponds to highly infrequent predator-prey interactions and leads to solutions with large amplitudes and gradients. For the nonlinear pendulum, we test that WSINDy is able to identify the $\sin(x_1)$ nonlinearity for both large and small initial amplitudes, noting that $x_1(0) = \frac{15}{16}\pi \approx \pi$ produces strongly nonlinear oscillations, while $x_1(0) = \frac{1}{16}\pi$ produces small-angle oscillations where $\sin(x_1) \approx x_1$. In addition, for the pendulum we use fewer samples ($M = 501$) and a larger time step $\Delta t = 0.1$, and hence observe a decreased the convergence rate. For the Lorenz equations we vary the initial conditions, generating 40 random initial conditions from a region covering the strange attractor, and show convergence over all cases.

2.3.2 Small-Noise Regime

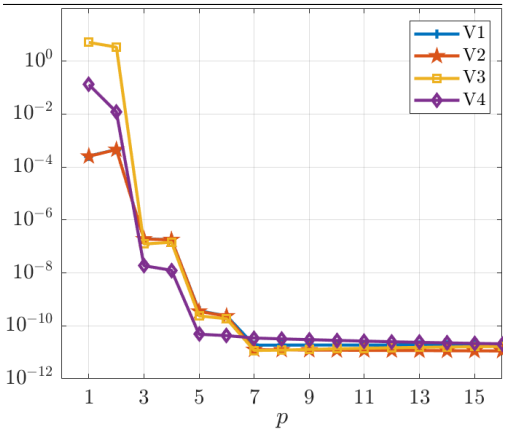
We now turn to the case of low to moderate noise levels, examining a noise ratio σ_{NR} in the range $[10^{-5}, 0.04]$ for the Van der Pol oscillator and Duffing's equation. We examine $\rho \in [1, 7]$ and $s \in [0.3, 0.95]$ where $\rho := \|\phi_k'\|_2 / \|\phi_k\|_2$ and s is the height of intersection of two neighboring test functions ϕ_k and ϕ_{k+1} (with $s = 1$ leading to $\phi_k = \phi_{k+1}$ and $s = 0$ indicating $\text{supp}(\phi_k) \cap \text{supp}(\phi_{k+1}) = \emptyset$). Using the analysis from Section 2.2.3, increasing ρ affects the distribution of the residual $\mathcal{R}(\mathbf{w})$ by magnifying the portion $R_3 = \langle \phi_k', \epsilon \rangle$ that is linear in the noise. For $\phi \in \mathcal{S}$, larger ρ corresponds to a higher polynomial degree p, with $\rho \in [1, 7]$ leading to $p \in [2, 98]$.



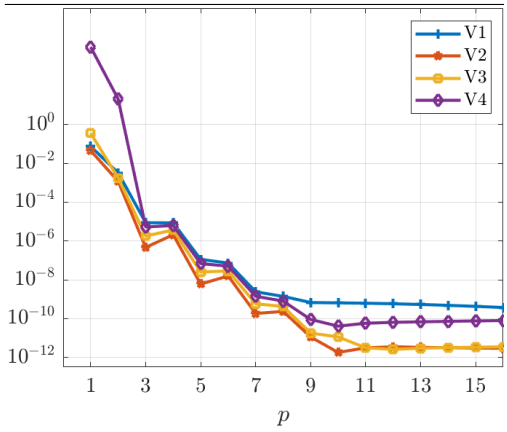
(a) Linear 5D



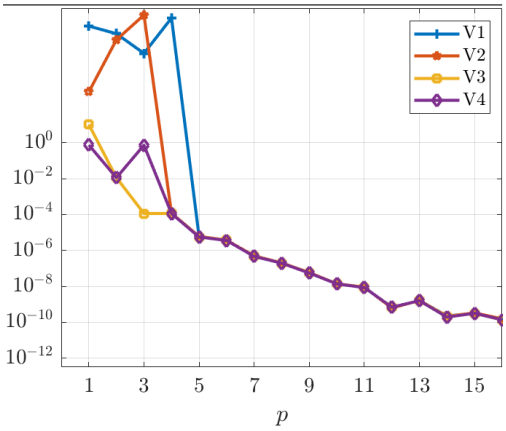
(b) Duffing



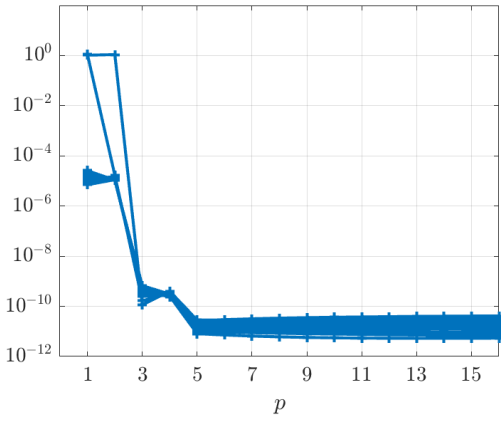
(c) Van der Pol.



(d) Lotka-Volterra.



(e) Pendulum.



(f) Lorenz.

Figure 2.2: Noise-free data ($\sigma_{NR} = 0$): plots of relative coefficient error $E_2(\widehat{\mathbf{w}})$ (defined in (2.3.2)) vs. p. V1-V4 indicate different ODE parameters (see Table 2.2). For the Lorenz system the parameters are fixed and 40 different initial conditions are sampled from a uniform distribution. In each case, the recovered coefficients $\widehat{\mathbf{w}}$ rapidly converge to within the accuracy of the ODE solver (10^{-10}).

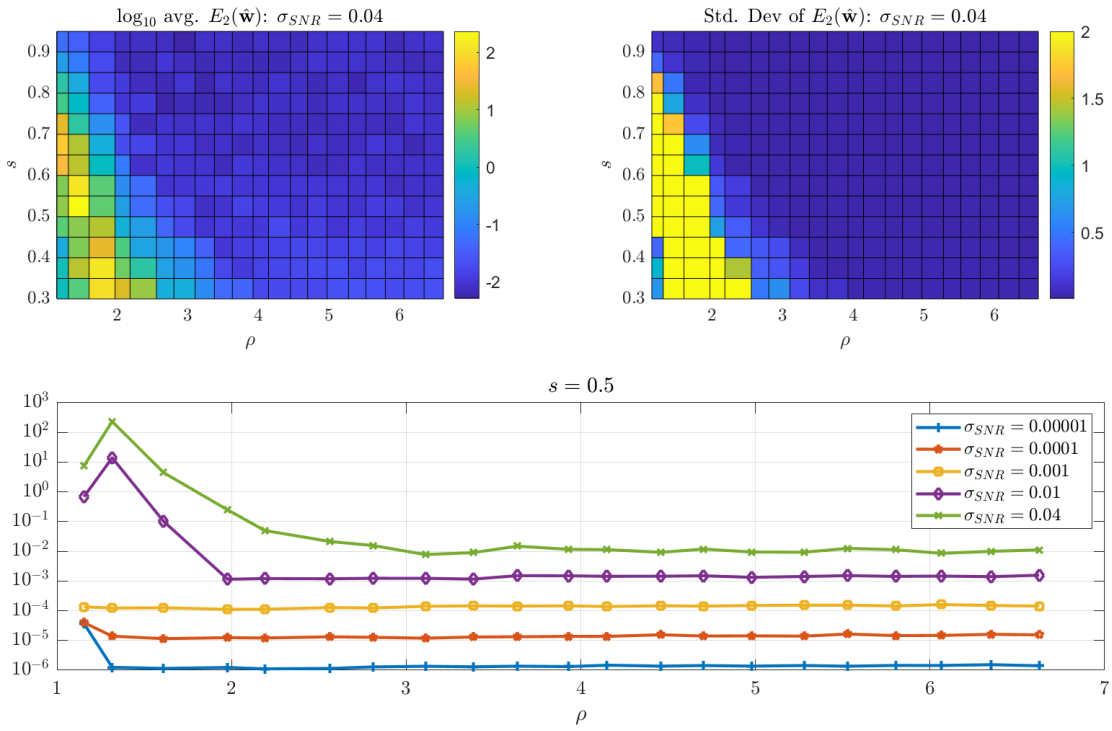


Figure 2.3: Small-noise regime: dynamic recovery of the Duffing equation with $\beta = 1$. Top: heat map of the $\log_{10}$ average error $E_2(\widehat{\mathbf{w}})$ (left) and sample standard deviation of $E_2(\widehat{\mathbf{w}})$ (right) over 200 instantiations of noise with $\sigma_{NR} = 0.04$ (4% noise) vs. ρ and s. Bottom: $E_2(\widehat{\mathbf{w}})$ vs. ρ for fixed $s = 0.5$ and various σ_{NR}. For $\rho > 3$ the average error is roughly an order of magnitude below σ_{NR}.

Larger shift parameter s corresponds to more test functions (higher K), but also to higher correlation between rows in $\mathbf{G}$, as $\langle \phi_k, f_j(\mathbf{y}) \rangle \approx \langle \phi_{k+1}, f_j(\mathbf{y}) \rangle$ when the supports of ϕ_k and ϕ_{k+1} sufficiently overlap. Here $s \in [0.3, 0.95]$ corresponds to $K \in [14, 451]$. We again use the uniform grid approach with $\gamma = 0$ and $\lambda = \frac{1}{4} \min_{\mathbf{w}_j^\star \neq 0} |\mathbf{w}_j^\star|$. For each system we generate 200 instantiations of noise and record the coefficient error over the range of s and ρ values.

From Figures 2.3 and 2.4 we observe two properties. Firstly, the coefficient error $E_2(\widehat{\mathbf{w}})$ monotonically deceases with increasing s and ρ, hence accurate recovery requires sufficient overlap between test functions (large enough shift parameter s) and sufficiently localized test functions that amplify the portion of the residual that is linear in the noise. Secondly, for large enough ρ and s, the error in the coefficients scales linearly with σ_{NR}, leading to an accuracy of $E_2(\widehat{\mathbf{w}}) \approx 0.1\sigma_{NR}$, or $-\log_{10}(0.1\sigma_{NR})$ significant digits in the recovered coefficients. In Appendix 2.A we show that this second property does not hold for standard SINDy, in particular the method of differentiation must change depending on the noise level in order to reach a desired accuracy.

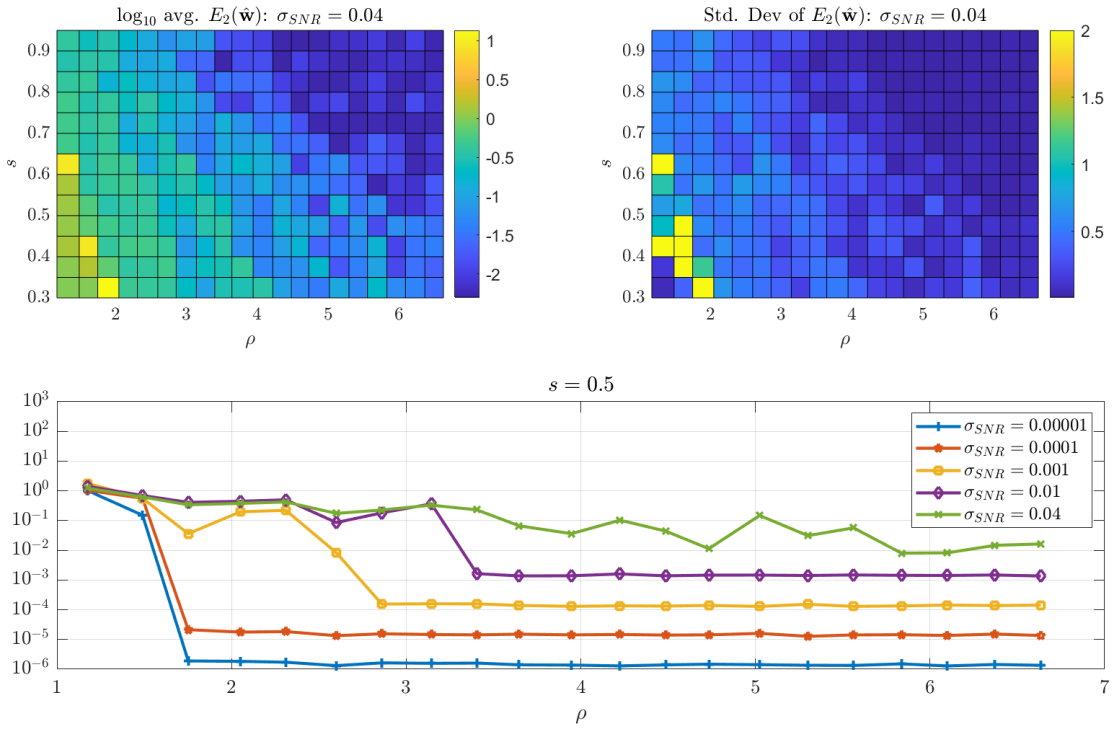


Figure 2.4: Small-noise regime: dynamic recovery of the Van der Pol oscillator with $\beta = 4$. Top: heat map of the $\log_{10}$ average error $E_2(\widehat{\mathbf{w}})$ (left) and sample standard deviation of $E_2(\widehat{\mathbf{w}})$ (right) over 200 instantiations of noise with $\sigma_{NR} = 0.04$ (4% noise) vs. ρ and s. Bottom: $E_2(\widehat{\mathbf{w}})$ vs. ρ for fixed $s = 0.5$ and various σ_{NR}. Similar to the Duffing equation, average error falls to roughly an order of magnitude below σ_{NR}, although for Van der Pol this regime is reached when $\rho \approx 6$.

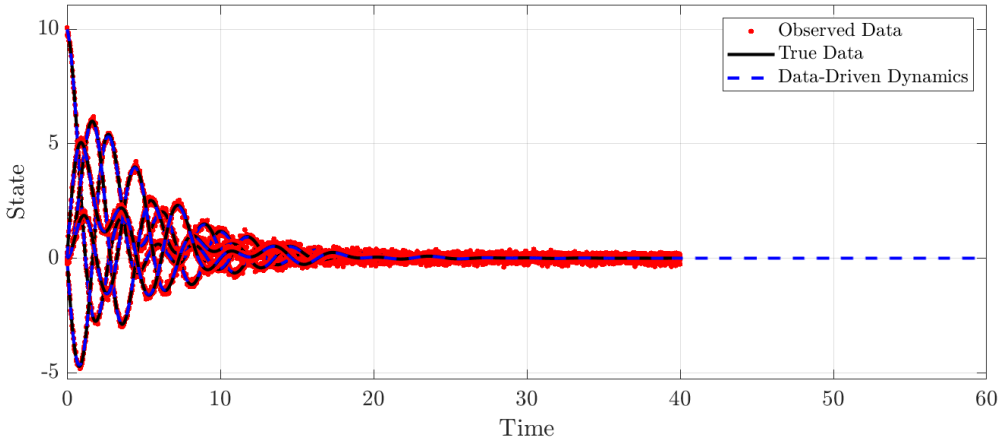


Figure 2.5: Large-noise regime: Linear 5D Equation with damping $\beta = -0.2$. All correct terms were identified with an error in the weights of $E_2(\widehat{\mathbf{w}}) = 0.0064$ and a trajectory error of $\mathcal{E}_2(\widehat{\mathbf{w}}) = 0.013$.

2.3.3 Large-Noise Regime

Figures 2.5 to 2.10 show that adaptive placement of test functions (Strategy 2) can be employed to discover dynamics in the large noise regime with fewer test functions. We test that each system in Table 2.1 can be discovered under $\sigma_{NR} = 0.1$ (10% noise) from only 250 test functions distributed near steep gradients in $\mathbf{y}$, which are located using the scheme in Section 2.2.4 with $p_w = 2$ and $L_w = 17$. We set the width-at-half-max of the test functions to $r_{whm} = \lfloor M/100 \rfloor$ timepoints. To exemplify the separation of scales and the severity of the corruption from noise, the noisy data $\mathbf{y}$, true data $\mathbf{x}$ and trajectories $\mathbf{x}_{dd}$ from the learned dynamical systems are shown in dynamo view and in phase space (for $D \leq 3$). We extend $\mathbf{x}_{dd}$ by 50% to show that the data-driven system captures the true limiting behavior. We set the sparsity to $\lambda = \frac{1}{4}\min_{\mathbf{w}^\star \neq 0} |\mathbf{w}^\star|$ and $\gamma = 0$ except in the linear 5D and nonlinear pendulum examples, where $\gamma = \sqrt{\sigma_{NR}} \approx 0.32$. For the trial basis we use all monomials up to degree 5 in the state variables, and for the pendulum we include the trigonometric terms $\sin(k\mathbf{y}_d), \cos(k\mathbf{y}_d)$ for $k = 1, 2$ and $d = 1, 2$.

In each case the correct terms are identified with coefficient error $E_2(\widehat{\mathbf{w}}) < 10^{-2}$, in agreement with the trend $E_2(\widehat{\mathbf{w}}) \approx 0.1\sigma_{NR}$ observed in the small noise regime. For the Linear 5D, Duffing, and Lotka-Volterra systems (Figures 2.5, 2.6, 2.8) the data-driven trajectory $\mathbf{x}_{dd}$ is indistinguishable from the true data to the eye, with trajectory error $\mathcal{E}_2(\widehat{\mathbf{w}}) < 0.02$. For the Van der Pol oscillator and nonlinear pendulum (Figures 2.7 and 2.9), $\mathbf{x}_{dd}$ follows a limit cycle with an attractor that is indistiguishable from the true data (see phase plane plots) however an error in the period of oscillation of roughly 0.6% leads to a larger trajectory error. The data-driven trajectory for the Lorenz equation diverges from the true trajectory around $t = 2.5$ (Figure 2.10), which is expected from chaotic dynamics, but still remains close to the Lorenz attractor.

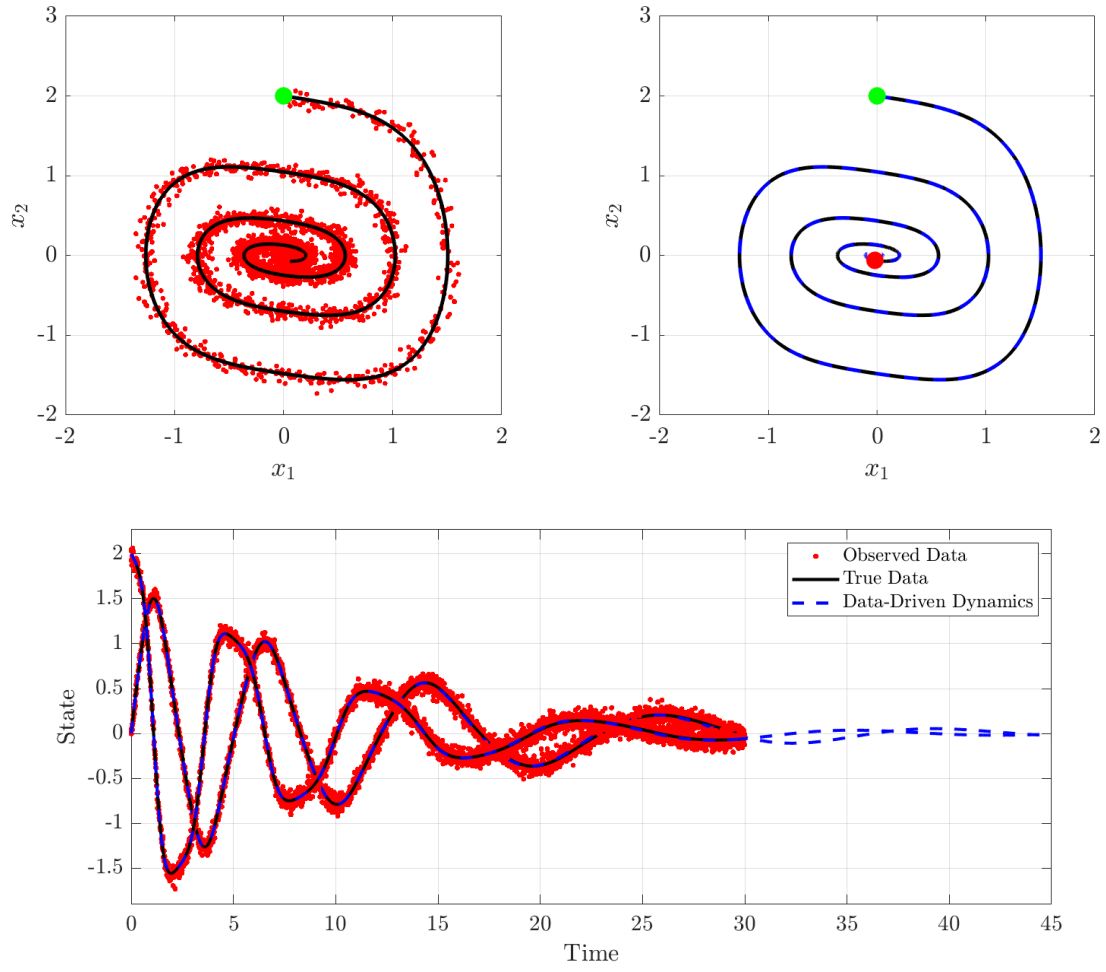


Figure 2.6: Large-noise regime: Duffing Equation, $\beta = 1$. All correct terms were identified with an error in the weights of $E_2(\widehat{\mathbf{w}}) = 0.0075$ and a trajectory error of $\mathcal{E}_2(\widehat{\mathbf{w}}) = 0.014$.

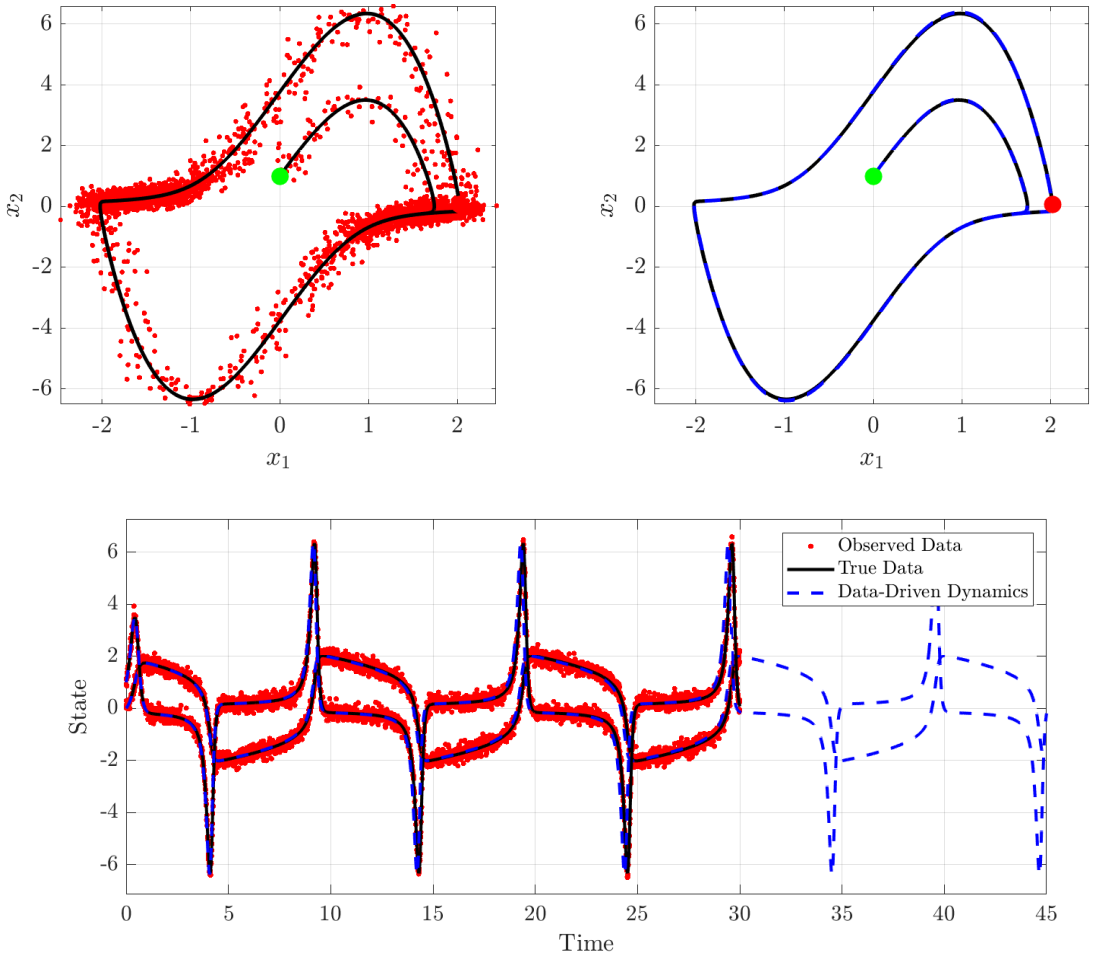


Figure 2.7: Large-noise regime: Van der Pol oscillator, $\beta = 4$. All correct terms were identified with coefficient error $E_2(\widehat{\mathbf{w}}) = 0.0073$ and trajectory error $\mathcal{E}_2(\widehat{\mathbf{w}}) = 0.32$. The data-driven trajectory $\mathbf{x}_{dd}$ has a slightly shorter oscillation period of 10.14 time units compared to the true 10.2, resulting in an eventual offset from the true data $\mathbf{x}$ and hence a larger trajectory error. Measured over the time interval $[0, 8]$ the trajectory error is 0.065.

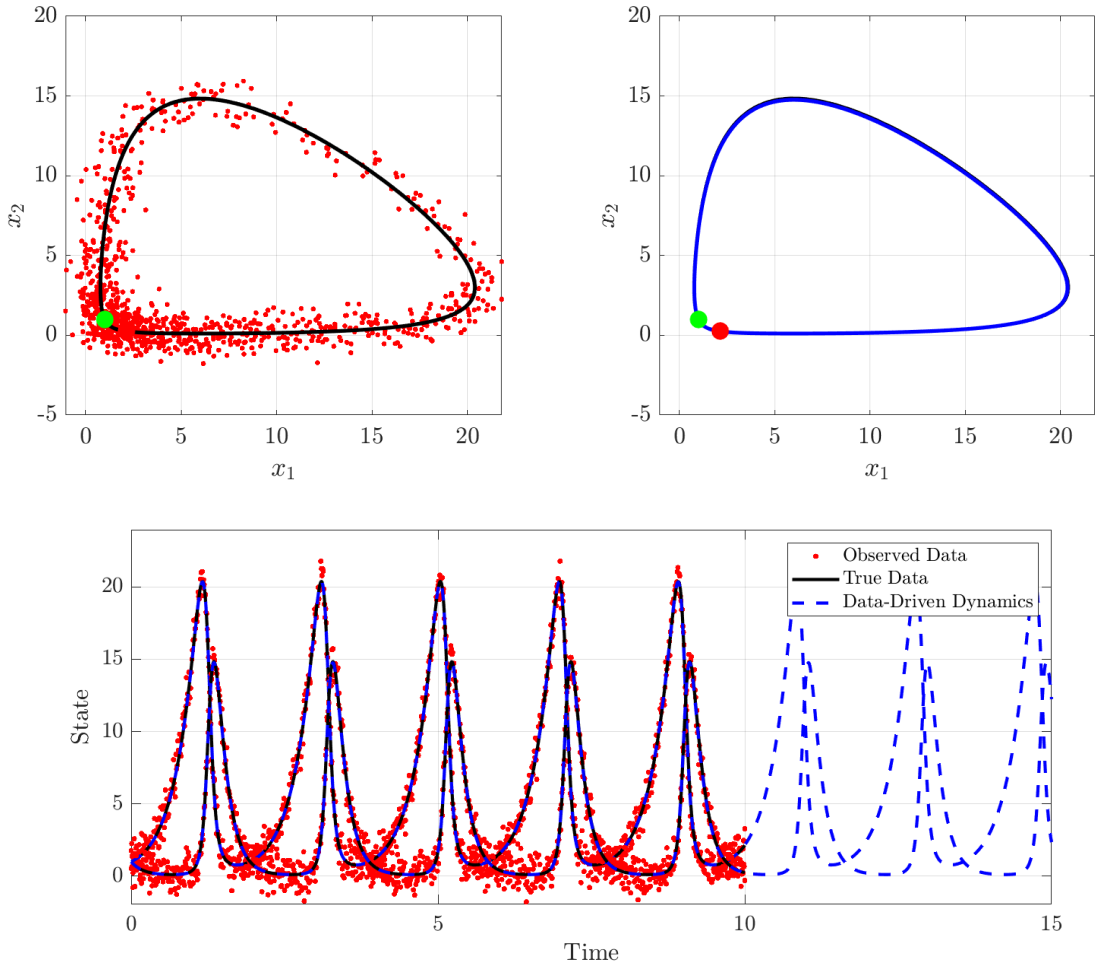


Figure 2.8: Large-noise regime: Lotka-Volterra system with $\beta = 1$. All correct nonzero terms were identified with an error in the weights of $E_2(\widehat{\mathbf{w}}) = 0.0013$ and trajectory error $\mathcal{E}_2(\widehat{\mathbf{w}}) = 0.0082$.

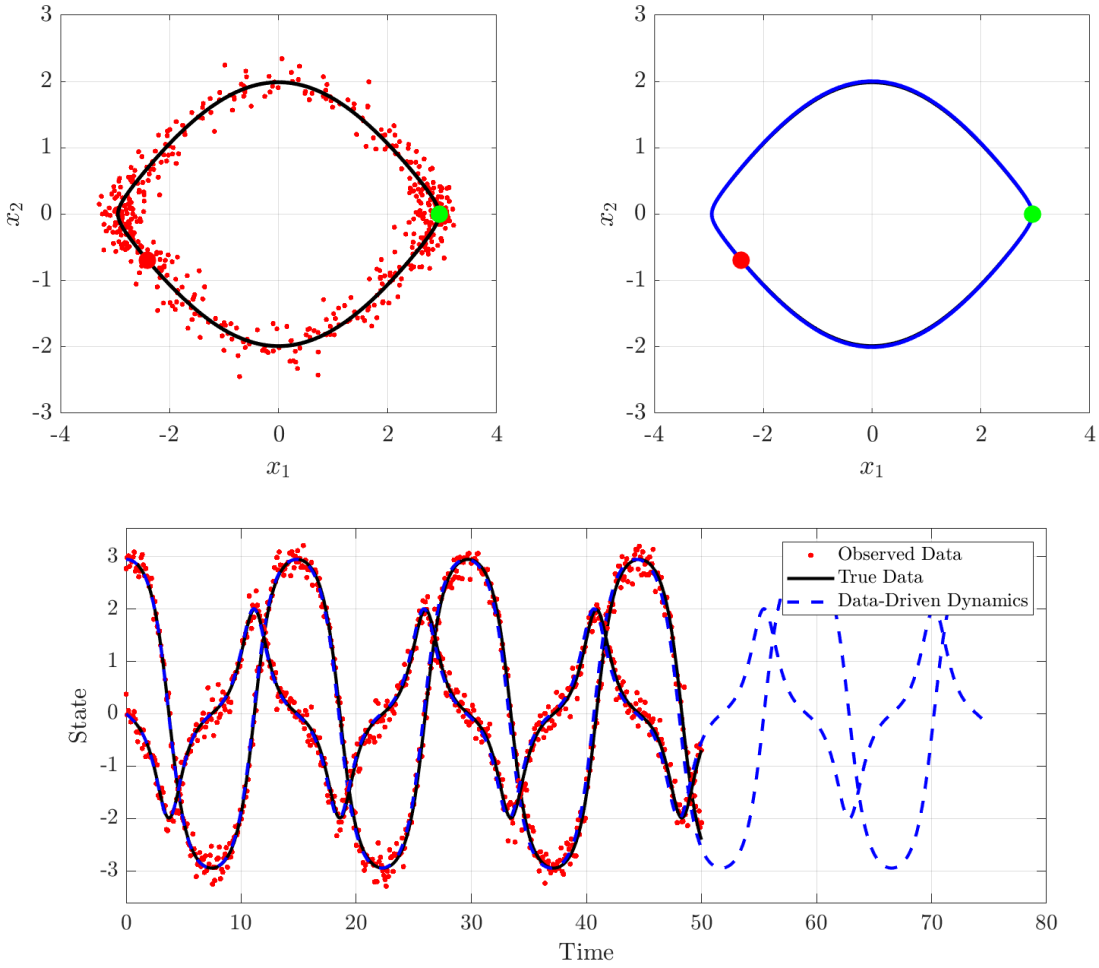


Figure 2.9: Large-noise regime: nonlinear pendulum with initial conditions $\mathbf{x}(0) = (15\pi/16,\ 0)^T$. All correct nonzero terms were identified with an error in the weights of $E_2(\widehat{\mathbf{w}}) = 0.0089$ and an error between $\mathcal{E}_2(\widehat{\mathbf{w}}) = 0.076$.

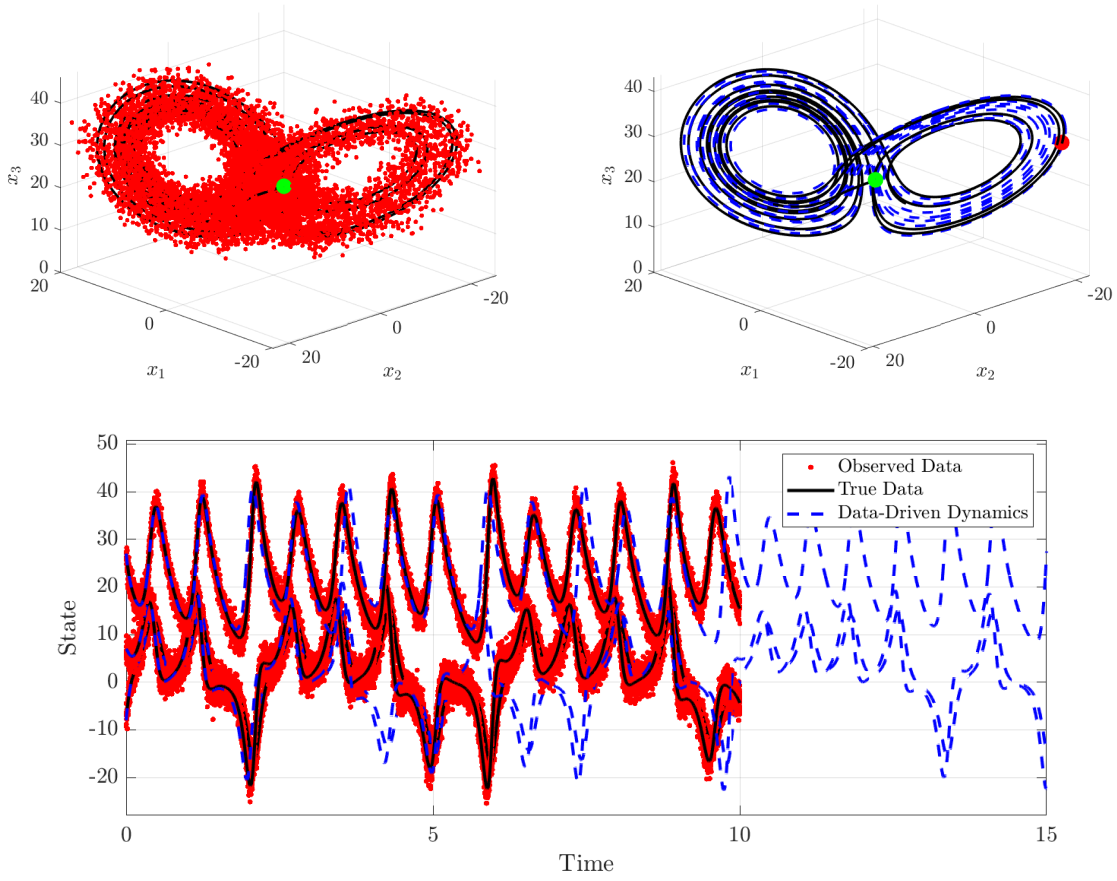


Figure 2.10: Large-noise regime: Lorenz system with $\mathbf{x}_0 = (-8,\ 7,\ 27)^T$. All correct terms were identified with an error in the weights of $E_2(\widehat{\mathbf{w}}) = 0.0084$ and trajectory error $\mathcal{E}(\widehat{\mathbf{w}}) = 0.56$. The large trajectory error is expected due to the chaotic nature of the solution. Using data up until $t = 1.5$ (first 1500 timepoints) the trajectory error is 0.027.

2.4 Concluding Remarks

We have developed and investigated a data-driven model selection algorithm based on the weak formulation of differential equations. The algorithm utilizes the reformulation of the model selection problem as a sparse regression problem for the weights $\mathbf{w}^\star$ of a candidate function basis $(f_j)_{j\in[J]}$ introduced in [140] and generalized in [23] as the SINDy algorithm. Our Weak SINDy algorithm (WSINDy) can be seen as a generalization of the sparse recovery scheme using integral terms found in [121], where dynamics were recovered from noisy data using the integral equation. We have shown that by extending the integral equation to the weak form, and using test functions with certain localization and smoothness properties, one may discovery the dynamics over a wide range of noise levels, with accuracy scaling favorably with noise: $E_2(\widehat{\mathbf{w}}) \approx 0.1\sigma_{NR}$.

A natural line of inquiry is to consider how WSINDy compares with conventional SINDy. There are several notable advantages of WSINDy; in particular, by considering the weak form of the equations, WSINDy completely avoids approximation of pointwise derivatives which significantly reduce the accuracy in conventional SINDy. When using SINDy, one must choose an appropriate numerical differentiation scheme depending on the noise level (e.g. finite differences are not robust to large noise but work well for small noise). For WSINDy, test functions from the space $\mathcal{S}$ (see Section (2.2.4)) together with the trapezoidal rule are effective in both low noise and high noise regimes. We demonstrate these observations in Appendix 2.A by comparing WSINDy to SINDy under several numerical differentiation schemes. On the other hand, it may be the case that less data is required by standard SINDy. For the examples show here, WSINDy works optimally for test functions supported on at least 15 timepoints, while many derivative approximations require fewer consecutive points.

WSINDy also utilizes the linearity of inner products with test functions to estimate the covariance structure of the residual, performing model selection in a generalized least squares framework. This is a much more appropriate setting given that residuals are neither independent nor uniformly distributed, however we note that our implementations in this chapter employ approximate covariance matrices and could benefit from further refinements and investigation. In Appendix 2.B we show that using generalized least squares with approximate covariance improves some results over ordinary least squares, but not significantly. We leave incorporation of more detailed knowledge of the covariance structure to future work. In addition, generalized least squares could potentially improve traditional model selection algorithms that rely on pointwise derivative estimates by similarly exploiting linear operators. Ultimately, a thorough analysis of the advantages of generalized least squares for model selection deserves further study.

Lastly, the most obvious extensions lie in generalizing the WSINDy method to spatiotemporal datasets. WSINDy as presented here in the context of ODEs is an exciting proof of concept, with natural extensions to spatiotemporal and multiresolution settings building upon the extensive results in numerical and functional analysis for weak and variational formulations of physical

problems.

2.5 Acknowledgements

This research was supported in part by the NSF/NIH Joint DMS/NIGMS Mathematical Biology Initiative grant R01GM126559 and in part by the NSF Computing and Communications Foundations Division grant CCF-1815983.

Appendix

2.A Comparison between WSINDy and SINDy

Here we compare WSINDy and SINDy using the Van der Pol oscillator, Lotka-Volterra system and Lorenz equation. For WSINDy we place test functions along the time axis according to the uniform grid strategy. For SINDy, we examine three differentiation methods: total-variation regularized derivatives (SINDy-TV), centered 2nd-order finite difference (SINDy-FD-2), and centered 4th-order finite difference (SINDy-FD-4). For SINDy-TV we use default settings and set the regularization parameter equal to the timestep.

For each system and noise level we generate 200 independent instantiations of noise and record the average coefficient error $E_2(\widehat{\mathbf{w}})$ (equation (2.3.2)) as well as the average True Positivity Ratio (TPR) [80]

$$\mathrm{TPR}(\widehat{\mathbf{w}}) = \frac{\mathrm{TP}(\widehat{\mathbf{w}})}{\mathrm{TP}(\widehat{\mathbf{w}}) + \mathrm{FP}(\widehat{\mathbf{w}}) + \mathrm{FN}(\widehat{\mathbf{w}})}, \tag{2.A.1}$$

where $\mathrm{TP}(\widehat{\mathbf{w}})$ is the number of correctly identified nonzero terms, $\mathrm{FP}(\widehat{\mathbf{w}})$ is the number of falsely identified nonzero terms, and $\mathrm{FN}(\widehat{\mathbf{w}})$ is the number of terms that are falsely identified as having a coefficient of zero. Since the feasible range of sparsity thresholds λ depends on the noise level, we adopt the selection methodology in [99] to choose an appropriate λ value for each instantiation of noise: λ is chosen from the set $10^{\{-5+\frac{i}{10}, i\in\{0,\dots,50\}\}}$ (i.e. the 51 values from 10^{-5} to 1 equally spaced $\log_{10}$) as the minimizer of the loss function

$$\mathcal{L}(\lambda) = \frac{\left\|\mathbf{A}\mathbf{w}^{\lambda} - \mathbf{A}\mathbf{w}^{0}\right\|_2}{\left\|\mathbf{A}\mathbf{w}^{0}\right\|_2} + \frac{\#\{j : \mathbf{w}_j^{\lambda} \neq 0\}}{J},$$

where $\mathbf{A} = \mathbf{V}\Theta(\mathbf{y})$ for WSINDy and $\mathbf{A} = \Theta(\mathbf{y})$ for SINDy; $\mathbf{w}^{\lambda}$ is the sequential-thresholding least-squares solution for sparsity threshold λ and J is the number of terms in the model library (for further details see [99]).

From Figures (2.A.2)-(2.A.3) we observe that for small noise (up to $\sigma_{NR} = 10^{-1}$), the coefficient error for WSINDy follows the linear trend $E_2(\widehat{\mathbf{w}}) \approx 0.1\sigma_{NR}$ (observed in the text) and that SINDy-FD-4 behaves similarly but with slightly worse accuracy. For larger noise, SINDy diverges in accuracy and identification of the correct nonzero terms for each differentiation scheme, while WSINDy maintains a TPR of at least 0.8 up to 40% noise for each system. WSINDy thus pro-

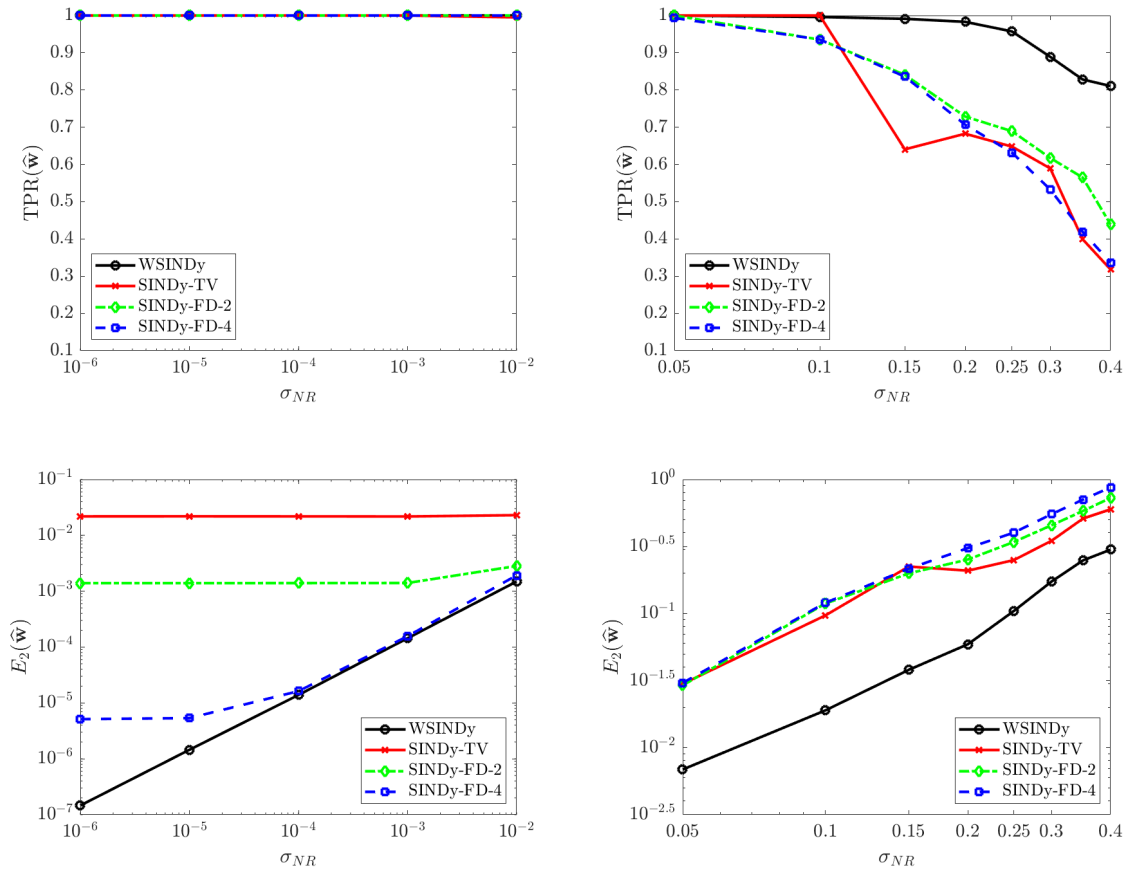


Figure 2.A.1: Comparison between WSINDy and SINDy: Van der Pol. Clockwise from top left: small-noise TPR($\widehat{\mathbf{w}}$) (defined in (2.A.1)), large-noise TPR($\widehat{\mathbf{w}}$), large-noise $E_2(\widehat{\mathbf{w}})$ (defined (2.3.2)), small-noise $E_2(\widehat{\mathbf{w}})$.

---

vides an advantage across the entire noise spectrum examined, all while employing the same weak discretization scheme.

2.B Generalized Least Squares vs. Ordinary Least Squares

Generalized least squares (GLS) aims to account for correlations between the residuals [70]. Given a linear model $y = \mathbf{X}\beta + \epsilon$ where $\mathrm{Cov}(\epsilon) = \Sigma$ and $\mathbb{E}[\epsilon|\mathbf{X}] = 0$, the GLS estimator of the parameters β upon observing $\hat{y}$ is

$$\widehat{\beta} = \left(\mathbf{X}^T\Sigma^{-1}\mathbf{X}\right)^{-1}\mathbf{X}^T\Sigma^{-1}\hat{y}.$$

This provides the best linear unbiased estimator of β in the sense that if $\widetilde{\beta}$ is any other unbiased estimator, then $\widehat{\beta}$ has lower variance: $\mathbb{V}[\widehat{\beta}_i] \leq \mathbb{V}[\widetilde{\beta}_i]$, $i = 1, \ldots, n$.

Above we derived an approximate covariance matrix $\Sigma \approx \mathbf{V}'(\mathbf{V}')^T$ to use in the GLS implemen-

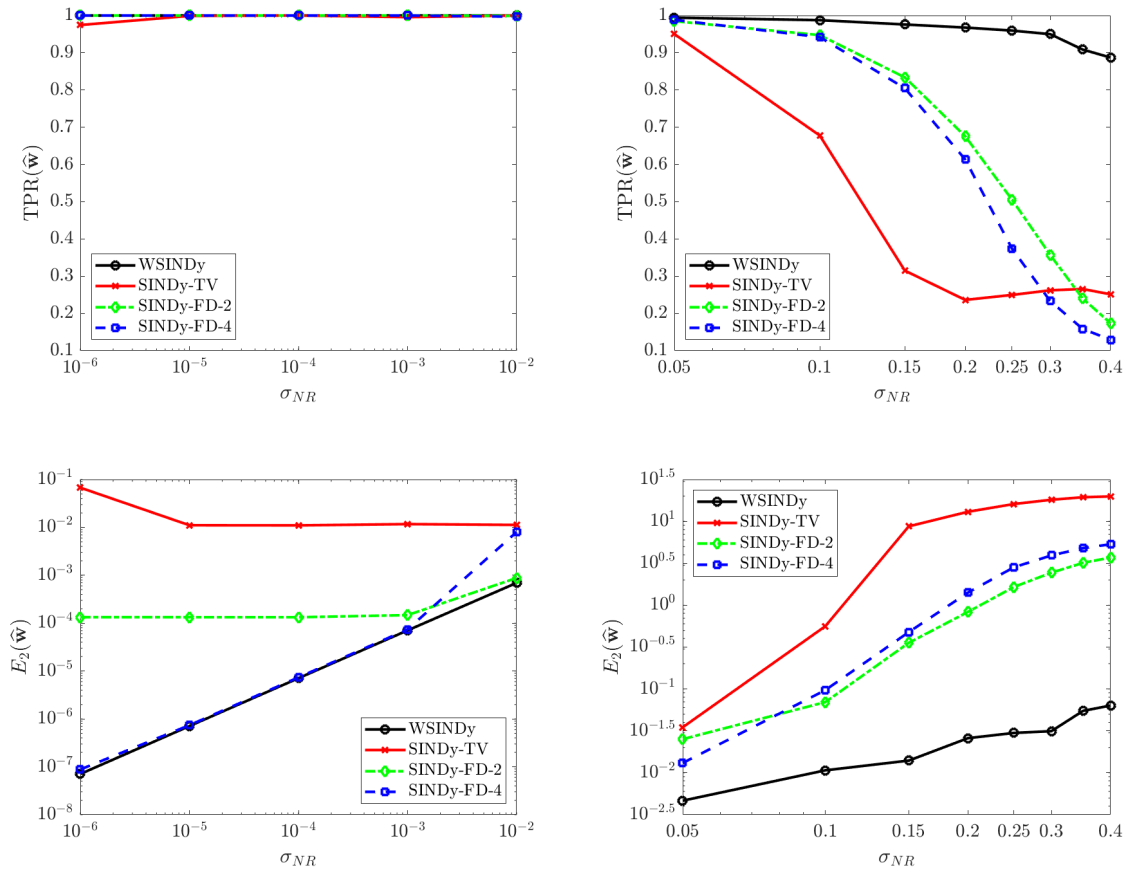


Figure 2.A.2: Comparison between WSINDy and SINDy: Lotka-Volterra. Clockwise from top left: small-noise TPR($\widehat{\mathbf{w}}$) (defined in (2.A.1)), large-noise TPR($\widehat{\mathbf{w}}$), large-noise $E_2(\widehat{\mathbf{w}})$ (defined (2.3.2)), small-noise $E_2(\widehat{\mathbf{w}})$.

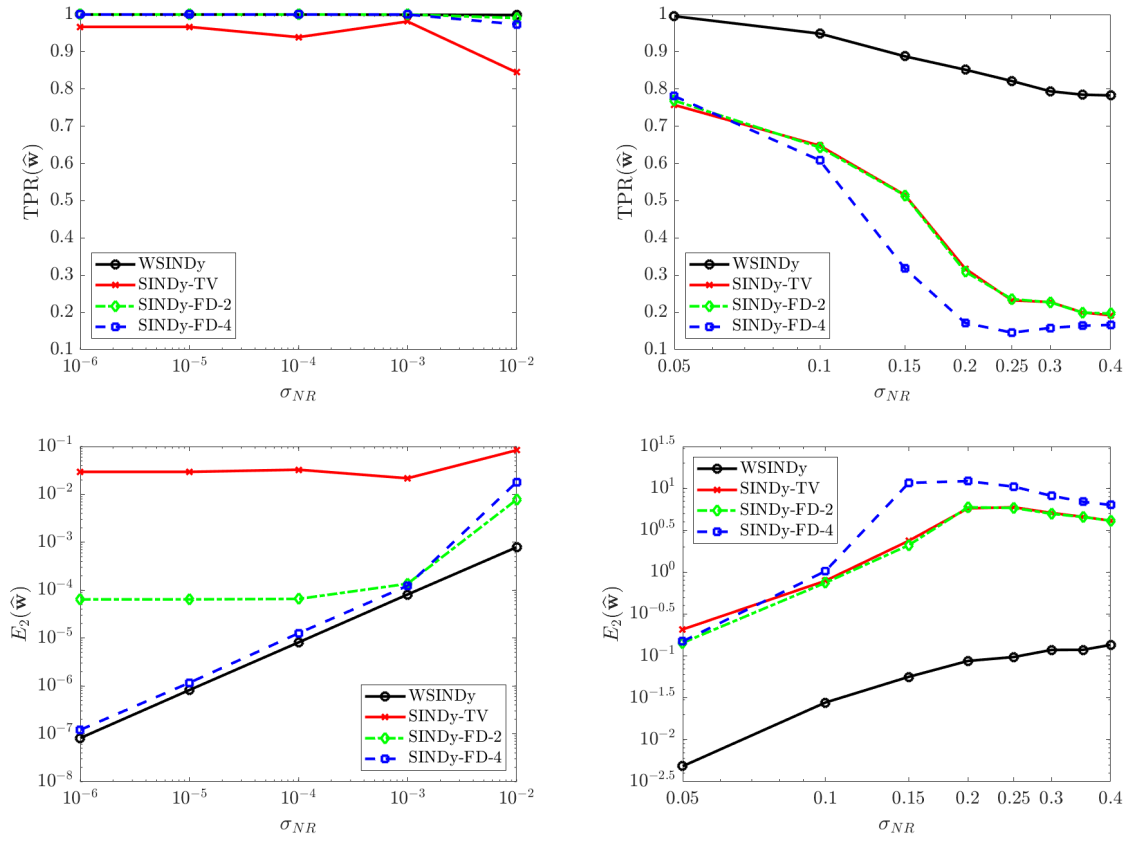


Figure 2.A.3: Comparison between WSINDy and SINDy: Lorenz system. Clockwise from top left: small-noise TPR($\widehat{\mathbf{w}}$) (defined in (2.A.1)), large-noise TPR($\widehat{\mathbf{w}}$), large-noise $E_2(\widehat{\mathbf{w}})$ (defined (2.3.2)), small-noise $E_2(\widehat{\mathbf{w}})$.

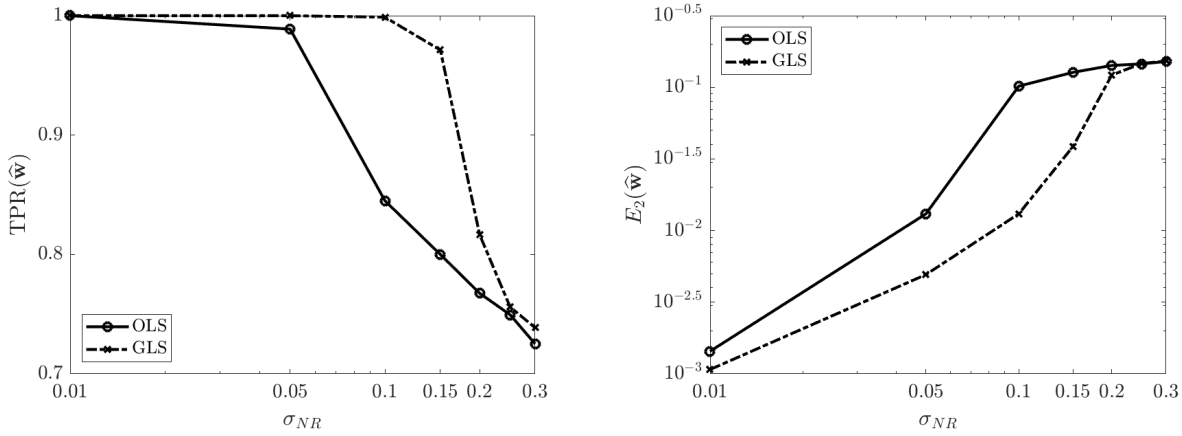


Figure 2.B.1: Comparison between WSINDy with generalized least squares and WSINDy with ordinary least squares using the Duffing equation. Results are averaged over 200 instantiations of noise.

---

tation of WSINDy, although the true covariance depends on the underlying unknown dynamical system and hence is unattainable. In addition, since in our case $\mathbf{X} = \mathbf{G} = \mathbf{V}\Theta(\mathbf{y})$ depends on the noise ϵ, the assumption $\mathbb{E}[\epsilon|\mathbf{X}] = 0$ is violated. Nevertheless, we find that the noise regime $\sigma_{NR} \in [\ 0.01, 0.3]$ does benefit from using GLS over LS. Figure 2.B.1 shows that for the Duffing equation, GLS extends the region $\{\sigma_{NR} \mid \text{TPR}(\widehat{\mathbf{w}}) > 0.95\}$ from $\sigma_{NR} \leq 0.05$ to $\sigma_{NR} \leq 0.15$, as well as increases the accuracy in the recovered coefficients. This suggests that further improvements can be made with a more refined covariance matrix.

Chapter 3

WSINDy for PDEs

Abstract

Sparse Identification of Nonlinear Dynamics (SINDy) is a method of system discovery that has been shown to successfully recover governing dynamical systems from data [23, 117]. Recently, several groups have independently discovered that the weak formulation provides orders of magnitude better robustness to noise. Here we extend our Weak SINDy (WSINDy) framework introduced in [100] to the setting of partial differential equations (PDEs). The elimination of pointwise derivative approximations via the weak form enables effective machine-precision recovery of model coefficients from noise-free data (i.e. below the tolerance of the simulation scheme) as well as robust identification of PDEs in the large noise regime (with signal-to-noise ratio approaching one in many well-known cases). This is accomplished by discretizing a convolutional weak form of the PDE and exploiting separability of test functions for efficient model identification using the Fast Fourier Transform. The resulting WSINDy algorithm for PDEs has a worst-case computational complexity of $\mathcal{O}(N^{D+1}\log(N))$ for datasets with N points in each of $D+1$ dimensions. Furthermore, our Fourier-based implementation reveals a connection between robustness to noise and the spectra of test functions, which we utilize in an *a priori* selection algorithm for test functions. Finally, we introduce a learning algorithm for the threshold in sequential-thresholding least-squares (STLS) that enables model identification from large libraries, and we utilize scale invariance at the continuum level to identify PDEs from poorly-scaled datasets. We demonstrate WSINDy's robustness, speed and accuracy on several challenging PDEs. Code is publicly available on GitHub at https://github.com/MathBioCU/WSINDy_PDE.

3.1 Chapter Outline

The outline of the chapter is as follows. In Section 3.2 we define the system discovery problem that we aim to solve and the notation to be used throughout. We then introduce the convolutional weak formulation along with our FFT-based discretization in Section 3.3. Key ingredients of the WSINDy algorithm for PDEs (Algorithm 3.4.2) are covered in Section 3.4, including a discussion of spectral properties of test functions and robustness to noise (3.4.1), our modified sequential thresholding scheme (3.4.2), and regularization using scale invariance of the underlying PDE (3.4.3). Section 3.5 contains numerical model discovery results for a range of nonlinear PDEs, including several vast improvements on existing results in the literature. We conclude the main text in Section 3.6 which summarizes the exposition and includes natural next directions for this line of research. Lastly, additional numerical details are included in the Appendix.

3.2 Problem Statement and Notation

Let $\mathbf{U}$ be a spatiotemporal dataset given on the spatial grid $\mathbf{X} \subset \overline{\Omega}$ over timepoints $\mathbf{t} \subset [0,T]$ where Ω is an open, bounded subset in $\mathbb{R}^D$, $D \geq 1$. In the cases we consider here, Ω is rectangular and the spatial grid is given by a tensor product of one-dimensional grids $\mathbf{X} = \mathbf{X}_1 \otimes \cdots \otimes \mathbf{X}_D$, where each $\mathbf{X}_d \in \mathbb{R}^{N_d}$ for $1 \leq d \leq D$ has equal spacing Δx, and the time grid $\mathbf{t} \in \mathbb{R}^{N_{D+1}}$ has equal spacing Δt. The dataset $\mathbf{U}$ is then a $(D+1)$-dimensional array with dimensions $N_1 \times \cdots \times N_{D+1}$. We write $h(\mathbf{X},\mathbf{t})$ to denote the $(D+1)$-dimensional array obtained by evaluating the function $h : \mathbb{R}^D \times \mathbb{R} \to \mathbb{C}$ at each of the points in the computational grid $(\mathbf{X},\mathbf{t})$. Individual points in $(\mathbf{X},\mathbf{t})$ will often be denoted by $(\mathbf{x}_k, t_k) \in (\mathbf{X},\mathbf{t})$ where

$$(\mathbf{x}_k, t_k) = (\mathbf{X}_{k_1,\dots,k_D}, t_{k_{D+1}}) = (x_{k_1},\dots,x_{k_D},t_{k_{D+1}}) \in \mathbb{R}^D \times \mathbb{R}.$$

In a mild abuse of notation, for a collection of points $\{(\mathbf{x}_k,t_k)\}_{k\in[K]} \subset (\mathbf{X},\mathbf{t})$, the index k plays a double role as a single index in the range $[K] := \{1,\dots,K\}$ referencing the point $(\mathbf{x}_k,t_k) \in \{(\mathbf{x}_k,t_k)\}_{k\in[K]}$ and as a multi-index on $(\mathbf{x}_k,t_k) = (\mathbf{X}_{k_1,\dots,k_D},t_{k_{D+1}})$, where k_d references the dth coordinate. This is particularly useful for defining a matrix $\mathbf{G} \in \mathbb{C}^{K\times J}$ of the form

$$\mathbf{G}_{k,j} = h_j(\mathbf{x}_k,t_k)$$

(as in equation (3.3.6) below) where $(h_j)_{j\in[J]}$ is a collection of J functions $h_j : \mathbb{R}^D \times \mathbb{R} \to \mathbb{C}$ evaluated at the set of K points $\{(\mathbf{x}_k,t_k)\}_{k\in[K]} \subset (\mathbf{X},\mathbf{t})$.

We assume that the data satisfies $\mathbf{U} = u(\mathbf{X},\mathbf{t}) + \epsilon$ for i.i.d. noise[1] ϵ and weak solution u of the

[1]Here ϵ is used to denote a multi-dimensional array of i.i.d. random variables and has the same dimensions as $\mathbf{U}$.

PDE

$$D^{\alpha^0}u(x,t) = D^{\alpha^1}g_1(u(x,t)) + D^{\alpha^2}g_2(u(x,t)) + \cdots + D^{\alpha^S}g_S(u(x,t)), \qquad x \in \Omega,\ t \in (0,T). \quad (3.2.1)$$

The problem we aim to solve is the identification of functions $(g_s)_{s\in[S]}$ and corresponding differential operators $(D^{\alpha^s})_{s\in[S]}$ that govern the evolution[2] of u according to $D^{\alpha^0}u$ given the dataset $\mathbf{U}$ and computational grid $(\mathbf{X}, \mathbf{t})$. Here and throughout we use the multi-index notation $\alpha^s = (\alpha^s_1, \ldots, \alpha^s_D, \alpha^s_{D+1}) \in \mathbb{N}^{D+1}$ to denote partial differentiation[3] with respect to $x = (x_1, \ldots, x_D)$ and t, so that

$$D^{\alpha^s}u(x,t) = \frac{\partial^{\alpha^s_1+\cdots+\alpha^s_D+\alpha^s_{D+1}}}{\partial x_1^{\alpha^s_1}\ldots\partial x_D^{\alpha^s_D}\partial t^{\alpha^s_{D+1}}}u(x,t).$$

We emphasize that a wide variety of PDEs can be written in the form (3.2.1). In particular, in this chapter we demonstrate our method of system identification on inviscid Burgers, Korteweg-de Vries, Kuramoto-Sivashinsky, nonlinear Schrödinger's, Sine-Gordon, a reaction-diffusion system and Navier-Stokes. The list of admissable PDEs that can be transformed into a weak form without any derivatives on the state variables includes many other well-known PDEs (Allen-Cahn, Cahn-Hilliard, Boussinesq,...).

3.3 Weak Formulation

To arrive at a computatonally tractable model recovery problem, we assume that the set of multi-indices $(\alpha^s)_{s\in[S]}$ together with α^0 enumerates the set of possible true differential operators that govern the evolution of u and that $(g_s)_{s\in[S]} \subset \operatorname{span}(f_j)_{j\in[J]}$ where the family of functions $(f_j)_{j\in[J]}$ (referred to as the *trial functions*) is known beforehand. This enables us to rewrite (3.2.1) as

$$D^{\alpha^0}u = \sum_{s=1}^{S}\sum_{j=1}^{J}\mathbf{w}^\star_{(s-1)J+j}D^{\alpha^s}f_j(u), \quad (3.3.1)$$

so that discovery of the correct PDE is reduced to a finite-dimensional problem of recovering the true vector of coefficients $\mathbf{w}^\star \in \mathbb{R}^{SJ}$, which is assumed to be sparse.

To convert the PDE into its weak form, we multiply equation (3.3.1) by a smooth *test function* $\psi(x,t)$, compactly-supported in $\Omega \times (0,T)$, and integrate over the spacetime domain,

$$\left\langle \psi,\ D^{\alpha^0}u \right\rangle = \sum_{s=1}^{S}\sum_{j=1}^{J}\mathbf{w}^\star_{(s-1)J+j}\left\langle \psi,\ D^{\alpha^s}f_j(u)\right\rangle,$$

where the L^2-inner product is defined $\langle \psi, f\rangle := \int_0^T\int_\Omega \psi^*(x,t)f(x,t)\,dxdt$ and ψ^* denotes the com-

[2]Commonly D^{α^0} is a time derivative ∂_t or ∂_{tt}, although this is not required.

[3]We will avoid using subscript notation such as u_x to denote partial derivatives, instead using $D^\alpha u$ or $\partial_x u$. For functions $f(x)$ of one variable, $f^{(n)}(x)$ denotes the nth derivative of f.

plex conjugate of ψ, although in what follows we integrate against only real-valued test functions and will omit the complex conjugation. Using the compact support of ψ and Fubini's theorem, we then integrate by parts as many times as necessary to arrive at the following weak form of the dynamics:

$$\left\langle (-1)^{|\alpha^0|} D^{\alpha^0}\psi,\ u \right\rangle = \sum_{s=1}^{S}\sum_{j=1}^{J} \mathbf{w}^\star_{(s-1)J+j} \left\langle (-1)^{|\alpha^s|} D^{\alpha^s}\psi,\ f_j(u) \right\rangle, \tag{3.3.2}$$

where $|\alpha^s| := \sum_{d=1}^{D+1} \alpha_d^s$ is the order of the multi-index[4]. Using an ensemble of test functions $(\psi_k)_{k\in[K]}$, we then discretize the integrals in (3.3.2) with $f_j(u)$ replaced by $f_j(\mathbf{U})$ (i.e. evaluated at the observed data $\mathbf{U}$) to arrive at the linear system

$$\mathbf{b} = \mathbf{G}\mathbf{w}^\star$$

defined by

$$\begin{cases} \mathbf{b}_k = \left\langle (-1)^{|\alpha^0|} D^{\alpha^0}\psi_k,\ \mathbf{U} \right\rangle, \\ \mathbf{G}_{k,(s-1)J+j} = \left\langle (-1)^{|\alpha^s|} D^{\alpha^s}\psi_k,\ f_j(\mathbf{U}) \right\rangle, \end{cases} \tag{3.3.3}$$

where $\mathbf{b} \in \mathbb{R}^K$, $\mathbf{G} \in \mathbb{R}^{K\times SJ}$ and $\mathbf{w}^\star \in \mathbb{R}^{SJ}$ are referred to throughout as the *left-hand side*, *Gram matrix* and *model coefficients*, respectively. In a mild abuse of notation, we use the inner product both in the sense of a continuous and exact integral in (3.3.2) and a numerical approximation in (3.3.3) which depends on a chosen quadrature rule. Building off of its success in the ODE setting, we use the trapezoidal rule throughout, as it has been shown to yield nearly negligible quadrature error with the test functions employed below (see Section 3.4.1 and [100]). In this way, solving $\mathbf{b} = \mathbf{G}\mathbf{w}^\star$ for the model coefficients $\mathbf{w}^\star$ allows for recovery of the PDE (3.3.1) without pointwise derivative approximations. The Gram matrix $\mathbf{G} \in \mathbb{R}^{K\times SJ}$ and left-hand side $\mathbf{b} \in \mathbb{R}^K$ defined in (3.3.3) conveniently take the same form regardless of the spatial dimension D, as their dimensions only depend on the number of test functions K and the size SJ of the model library, composed of J trial functions $(f_j)_{j\in[J]}$ and S candidate differential operators enumerated by the multi-index set $\boldsymbol{\alpha} := (\alpha^s)_{1\le s\le S}$.

3.3.1 Convolutional Weak Form and Discretization

We now restrict to the case of each test function ψ_k being a translation of a reference test function ψ, i.e. $\psi_k(x,t) = \psi(\mathbf{x}_k - x, t_k - t)$ for some collection of points $\{(\mathbf{x}_k, t_k)\}_{k\in[K]} \subset (\mathbf{X}, \mathbf{t})$ (referred to as the *query points*). The weak form of the dynamics (3.3.2) over the test function

[4]For example, with $D^{\alpha^s} = \frac{\partial^{2+1}}{\partial x^2 \partial y}$, integration by parts occurs twice with respect to the x-coordinate and once with respect to y, so that $|\alpha^s| = 3$ and $(-1)^{|\alpha^s|} = -1$.

basis $(\psi_k)_{k\in[K]}$ then takes the form of a convolution:

$$\left(D^{\alpha^0}\psi\right) * u(\mathbf{x}_k, t_k) = \sum_{s=1}^{S}\sum_{j=1}^{J} \mathbf{w}^\star_{(s-1)J+j} \left(D^{\alpha^s}\psi\right) * f_j(u)(\mathbf{x}_k, t_k). \tag{3.3.4}$$

The sign factor $(-1)^{|\alpha^s|}$ appearing in (3.3.2) after integrating by parts is eliminated in (3.3.4) due to the sign convention in the integrand of the space-time convolution, which is defined by

$$\psi * u(x,t) := \int_0^T \int_\Omega \psi(x-y, t-s)u(y,s)\,dyds = \langle \psi(x-\cdot, t-\cdot),\ u(\cdot,\cdot)\rangle\,.$$

Construction of the linear system $\mathbf{b} = \mathbf{G}\mathbf{w}^\star$ as a discretization of the *convolutional* weak form (3.3.4) over the query points $\{(\mathbf{x}_k, t_k)\}_{k\in[K]}$ can then be carried out efficiently using the FFT as we describe below.

To relate the continuous and discrete convolutions, we assume that the support of ψ is contained within some rectangular domain

$$\Omega_R := [-b_1, b_1] \times \cdots \times [-b_D, b_D] \times [-b_{D+1}, b_{D+1}] \subset \mathbb{R}^D \times \mathbb{R}$$

where $b_d = m_d\Delta x$ for $d \in [D]$ and $b_{D+1} = m_{D+1}\Delta t$. We then define a reference computational grid $(\mathbf{Y}, \mathfrak{t}) \subset \mathbb{R}^D \times \mathbb{R}$ for ψ centered at the origin and having the same sampling rates $(\Delta x, \Delta t)$ as the data $\mathbf{U}$, where $\mathbf{Y} = \mathbf{Y}_1 \otimes \cdots \otimes \mathbf{Y}_D$ for $\mathbf{Y}_d = (n\Delta x)_{-m_d\le n\le m_d}$ and $\mathfrak{t} = (n\Delta t)_{-m_{D+1}\le n\le m_{D+1}}$. In this way $\mathbf{Y}$ contains $2m_d + 1$ points along each dimension $d \in [D]$, with equal spacing Δx, and $\mathfrak{t}$ contains $2m_{D+1} + 1$ points with equal spacing Δt. As with $(\mathbf{X}, \mathbf{t})$, points in $(\mathbf{y}_k, \mathfrak{t}_k) \in (\mathbf{Y}, \mathfrak{t})$ take the form

$$(\mathbf{y}_k, \mathfrak{t}_k) = (\mathbf{Y}_{k_1,\ldots,k_D}, \mathfrak{t}_{k_{D+1}})$$

where each index k_d for $d \in [D+1]$ takes values in the range $\{-m_d, \ldots, 0, \ldots, m_d\}$, and for valid indices $k - j$, the two grids $(\mathbf{X}, \mathbf{t})$ and $(\mathbf{Y}, \mathfrak{t})$ are related by

$$(\mathbf{x}_k - \mathbf{x}_j, t_k - t_j) = (\mathbf{y}_{k-j}, \mathfrak{t}_{k-j}). \tag{3.3.5}$$

We stress that $(\mathbf{Y}, \mathfrak{t})$ is completely defined by the integers $\mathbf{m} = (m_d)_{d\in[D+1]}$, specified by the user, and that the values of $\mathbf{m}$ have a significant impact on the algorithm. For this reason we develop an automatic selection algorithm for $\mathbf{m}$ using spectral properties of the data $\mathbf{U}$ (see Appendix 3.A).

The linear system (3.3.3) can now be rewritten

$$\begin{cases} \mathbf{b}_k = \Psi^0 * \mathbf{U}(\mathbf{x}_k, t_k), \\ \mathbf{G}_{k,(s-1)J+j} = \Psi^s * f_j(\mathbf{U})(\mathbf{x}_k, t_k), \end{cases} \tag{3.3.6}$$

where $\Psi^s := D^{\alpha^s}\psi(\mathbf{Y}, \mathfrak{t})\Delta x^D\Delta t$ and the factor $\Delta x^D\Delta t$ characterizes the trapezoidal rule. We

define the discrete $(D+1)$-dimensional convolution between Ψ^s and $f_j(\mathbf{U})$ at a point $(\mathbf{x}_k, t_k) = (\mathbf{X}_{k_1,\ldots,k_D}, t_{k_{D+1}}) \in (\mathbf{X}, \mathbf{t})$ by

$$\Psi^s * f_j\left(\mathbf{U}\right)(\mathbf{x}_k, t_k) := \sum_{\ell_1=1}^{N_1} \cdots \sum_{\ell_{D+1}=1}^{N_{D+1}} \Psi^s_{k_1-\ell_1,\ldots,k_{D+1}-\ell_{D+1}} f_j\left(\mathbf{U}_{\ell_1,\ldots,\ell_{D+1}}\right),$$

which, substituting the definition of Ψ^s,

$$:= \sum_{\ell_1=1}^{N_1} \cdots \sum_{\ell_{D+1}=1}^{N_{D+1}} D^{\alpha^s}\psi\left(\mathbf{Y}_{k_1-\ell_1,\ldots,k_D-\ell_D},\ \mathfrak{t}_{k_{D+1}-\ell_{D+1}}\right) f_j\left(\mathbf{U}_{\ell_1,\ldots,\ell_{D+1}}\right) \Delta x^D \Delta t \tag{3.3.7}$$

truncating indices appropriately and using (3.3.5),

$$= \sum_{\ell_1=k_1-m_1}^{k_1+m_1} \cdots \sum_{\ell_{D+1}=k_{D+1}-m_{D+1}}^{k_{D+1}+m_{D+1}} D^{\alpha^s}\psi\left(\mathbf{Y}_{k_1-\ell_1,\ldots,k_D-\ell_D},\ \mathfrak{t}_{k_{D+1}-\ell_{D+1}}\right) f_j\left(\mathbf{U}_{\ell_1,\ldots,\ell_{D+1}}\right) \Delta x^D \Delta t \tag{3.3.8}$$

$$= \sum_{\ell_1=k_1-m_1}^{k_1+m_1} \cdots \sum_{\ell_{D+1}=k_{D+1}-m_{D+1}}^{k_{D+1}+m_{D+1}} D^{\alpha^s}\psi\left(\mathbf{x}_k - \mathbf{x}_\ell, t_k - t_\ell\right) f_j\left(\mathbf{U}_{\ell_1,\ldots,\ell_{D+1}}\right) \Delta x^D \Delta t \tag{3.3.9}$$

$$\approx \int_0^T \int_\Omega D^{\alpha^s}\psi(\mathbf{x}_k - x, t_k - t) f_j\left(u(x,t)\right)\, dx\, dt. \tag{3.3.10}$$

3.3.2 FFT-based Implementation and Computational Complexity for Separable ψ

Convolutions in the linear system (3.3.6) may be computed rapidly if the reference test function ψ is separable over the given coordinates, i.e.

$$\psi(x,t) = \phi_1(x_1)\cdots\phi_2(x_D)\phi_{D+1}(t)$$

for univariate functions $(\phi_d)_{d\in[D+1]}$. In this case,

$$D^{\alpha^s}\psi(\mathbf{Y}, \mathfrak{t}) = \phi_1^{(\alpha_1^s)}(\mathbf{Y}_1) \otimes \cdots \otimes \phi_D^{(\alpha_D^s)}(\mathbf{Y}_D) \otimes \phi_{D+1}^{(\alpha_{D+1}^s)}(\mathfrak{t}),$$

so that only the vectors

$$\phi_d^{(\alpha_d^s)}(\mathbf{Y}_d) \in \mathbb{R}^{2m_d+1}, \quad d \in [D] \quad \text{and} \quad \phi_{D+1}^{(\alpha_{D+1}^s)}(\mathfrak{t}) \in \mathbb{R}^{2m_{D+1}+1},$$

need to be computed for each $0 \leq s \leq S$ and the multi-dimensional arrays $(\Psi^s)_{s=0,\dots,S}$ are never directly constructed. Convolutions can be carried out sequentially in each coordinate[5], so that the overall cost of computing each column $\Psi^s * f_j(\mathbf{U})$ of $\mathbf{G}$ is

$$T_I(N, n, D) := CN \log(N) \sum_{d=1}^{D+1} N^{D+1-d} (N - n + 1)^{d-1}, \tag{3.3.11}$$

if the computational grid $(\mathbf{X}, \mathbf{t})$ and reference grid $(\mathbf{Y}, \mathfrak{t})$ have N and $n \leq N$ points along each of the $D+1$ dimensions, respectively. Here $CN\log(N)$ is the cost of computing the 1D convolution between vectors vectors $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ and $\mathbf{y} = (y_1, \dots, y_N) \in \mathbb{R}^N$ using the FFT,

$$\mathbf{x} * \mathbf{y} = \mathcal{P}\mathcal{F}^{-1}\left(\mathcal{F}(\mathbf{x}^0) \odot \mathcal{F}(\mathbf{y})\right), \tag{3.3.12}$$

where $\mathbf{x}^0 = (0, \dots, 0, x_1, \dots, x_n) \in \mathbb{R}^N$, $\odot$ denotes element-wise multiplication and $\mathcal{P}$ projects onto the first $N - n + 1$ components. The discrete Fourier transform $\mathcal{F}$ is defined

$$\mathcal{F}_k(\mathbf{y}) = \sum_{j=1}^{N} \mathbf{y}_j e^{-2\pi i (j-1)(k-1)}$$

with inverse

$$\mathcal{F}_k^{-1}(\mathbf{z}) = \frac{1}{N} \sum_{j=1}^{N} \mathbf{z}_j e^{2\pi i (j-1)(k-1)}.$$

The projection $\mathcal{P}$ ensures that the convolution only includes points that correspond to integrating over test functions ψ that are compactly supported in $(\mathbf{X}, \mathbf{t})$, which is necessary for integration by parts to hold in the weak form. The spectra of the test functions $\phi_d^{(\alpha_d^s)}(\mathbf{Y}_d)$ can be precomputed and in principle each convolution $\Psi^s * f_j(\mathbf{U})$ can be carried out in parallel[6], making the total cost of the WSINDy Algorithm (3.4.2) in the PDE setting equal to (3.3.11) (ignoring the cost of the least-squares solves which are negligible in comparison to computing $(\mathbf{G}, \mathbf{b})$). In addition, subsampling reduces the term $(N-n+1)$ in (3.3.11) to $(N-n+1)/s$ where $s \geq 1$ is the subsampling rate such that $(N - n + 1)/s$ points are kept along each dimension.

For most practical combinations of n and N, (say $n > N/10$ and $N > 150$) using the FFT and separability provides a considerable reduction in computational cost. See Figure 3.3.1 for a comparison between T_I and the naive cost T_{II} of an $(N+1)$-dimensional convolution:

$$T_{II}(N, n, D) := (2n^{D+1} - 1)(N - n + 1)^{D+1}. \tag{3.3.13}$$

[5]The technique of exploiting separability in high-dimensional integration is not new (see [111] for an early introduction) and is frequently utilized in scientific computing (see [9, 58] for examples in computational chemistry).

[6]For the examples in Section 3.5 the walltimes are reported for serial computation of $(\mathbf{G}, \mathbf{b})$.

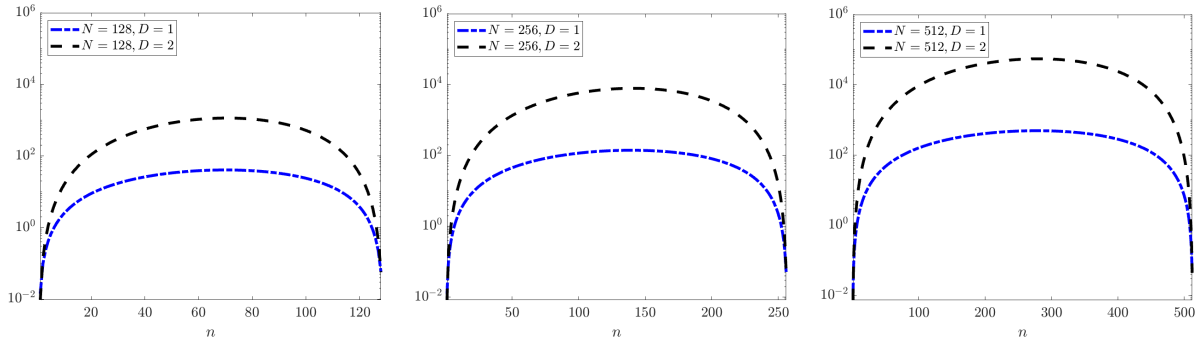


Figure 3.3.1: Reduction in computational cost of multi-dimensional convolution $\Psi^s * f_j(\mathbf{U})$ when Ψ^s and $f_j(\mathbf{U})$ have n and N points in each of $D+1$ dimensions, respectively. Each plot shows the ratio T_{II}/T_I (equations (3.3.13) and (3.3.11)), i.e. the factor by which the separable FFT-based convolution reduces the cost of the naive convolution, for $D+1=2$ and $D+1=3$ space-time dimensions and $n \in [N]$. The right-most plot shows that when $N = 512$ and $D+1=3$, the separable FFT-based convolution is 10^4 times faster for $100 \leq n \leq 450$.

For example, with $n = N/4$ (a typical value) we have $T_{II} = \mathcal{O}(N^{2D+2})$ and $T_I = \mathcal{O}(N^{D+1}\log(N))$, hence exploiting separability reduces the computational complexity by a factor of $N^{D+1}/\log(N)$.

3.4 WSINDy Algorithm for PDEs and Hyperparameter Selection

WSINDy for PDE discovery is given in Algorithm 3.4.2, where the user must specify each of the hyperparameters in Table 3.4.1. The key pieces of the algorithm are (i) the choice of reference test function ψ, (ii) the method of a sparsification, (iii) the method of regularization, (iv) selection of convolution query points $\{(\mathbf{x}_k, t_k)\}_{k\in K}$, and (v) the model library. At first glance, the number of hyperparameters is quite large. We now discuss several simplifications that either reduce the number of hyperparameters or provide methods of choosing them automatically. In Section 3.4.1 we discuss connections between the convolutional weak form and spectral properties of ψ that determine the scheme's robustness to noise and inform the selection of test function hyperparameters. In Section 3.4.2 we introduce a *modified* sequential-thresholding least-squares algorithm (MSTLS) which includes automatic selection of the threshold λ and allows for PDE discovery from large libraries. In Section 3.4.3 we describe how scale invariance of the PDE is used to rescale the data and coordinates in order to regularize the model recovery problem in the case of poorly-scaled data. In Sections 3.4.4 and 3.4.5 we briefly discuss selection of query points and an appropriate model library, however these components of the algorithm will be investigated more thoroughly in future research.

3.4.1 Selecting a Reference Test Function ψ

Convolutional Weak Form and Fourier Analysis

Computation of $\mathbf{G}$ and $\mathbf{b}$ in (3.3.6) with ψ separable requires the selection of appropriate 1D coordinate test functions $(\phi_d)_{d\in[D+1]}$. Computing convolutions using the FFT (3.3.12) suggests a mechanism for choosing appropriate test functions. Define the Fourier coefficients of a function $u \in L^2([0,T])$ by

$$\widehat{u}(k) = \frac{1}{\sqrt{T}} \int_0^T u(t) e^{-\frac{2\pi i k}{T} t}\, dt, \qquad k \in \mathbb{Z}.$$

Consider data $\mathbf{U} = u(\mathbf{t}) + \epsilon \in \mathbb{R}^N$ for a T-periodic function u, $\mathbf{t}_k = k\frac{T}{N} = k\Delta t$, and i.i.d. noise $\epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. The discrete Fourier transform of the noise $\mathcal{F}(\epsilon) := \epsilon_R + i\epsilon_I$ is then distributed $\epsilon_R, \epsilon_I \sim \mathcal{N}(0, (N\sigma^2/2)\mathbf{I})$. In addition, there exist constants $C > 0$ and $\ell > 1/2$ such that $|\widehat{u}_k| \leq C|k|^{-\ell}$ for each $k \in \mathbb{Z}$. There then exists a noise-dominated region of the spectrum $\mathcal{F}(\mathbf{U})$ determined by the noise-to-signal ratio

$$NSR_k := \mathbb{E}\left[\frac{|\mathcal{F}_k(\epsilon)|^2}{|\mathcal{F}_k(u(\mathbf{t}))|^2}\right] = \frac{N\sigma^2}{|\mathcal{F}_k(u(\mathbf{t}))|^2} \approx \frac{T\sigma^2}{N|\widehat{u}(k)|^2} \geq \frac{1}{C^2}\Delta t \sigma^2 k^{2\ell},$$

where '$\approx$' corresponds to omitting the aliasing error. For $NSR_k \geq 1$ the kth Fourier mode is by definition noise-dominated, which corresponds to wavenumbers

$$|k| \geq k^* \approx \left(\frac{C}{\sigma\sqrt{\Delta t}}\right)^{1/\ell}. \tag{3.4.1}$$

If the critical wavenumber k^* between the noise dominated ($NSR_k \geq 1$) and signal-dominated ($NSR_k \leq 1$) modes can be estimated from the dataset $\mathbf{U}$, then it is possible to design test functions ψ such that the noise-dominated region of $\mathcal{F}(\mathbf{U})$ lies in the tail of $\widehat{\psi}$. The convolutional weak form (3.3.6) can then be interpreted as an approximate low-pass filter on the noisy dataset, offering robustness to noise without altering the frequency content of the data[7].

In summary, spectral decay properties of the reference test function ψ serve to damp high-frequency noise in the convolutional weak form, which acts together with the natural variance-reducing effect of integration, as described in [57], to allow for quantification and control of the scheme's robustness to noise. Specifically, coordinate test functions ϕ_d with wide support in real space (larger m_d) will reduce more variance, but will have a faster-decaying spectrum $\widehat{\phi}_d$, so that *signal*-dominated modes may not be resolved, leading to model misidentification. On the other hand, if ϕ_d decays too swiftly in real space (smaller m_d), then the spectrum $\widehat{\phi}_d$ will decay more slowly and may put too much weight on *noise*-dominated frequencies. In addition, smaller m_d may not sufficiently reduce variance. A balance must be struck between (a) effectively reducing

[7]This is in contrast to explicit data-denoising, where a filter is applied to the dataset prior to system identification and may fundamentally alter the underlying clean data. The implicit filtering of the convolutional weak form is made explicit by the FFT-based implementation (3.3.12).

variance, which is ultimately determined by the decay of ψ in physical space, and (b) resolving the underlying dynamics, determined by the decay of $\widehat{\psi}$ in Fourier space.

Piecewise-Polynomial Test Functions

Many test functions achieve the necessary balance between decay in real space and decay in Fourier space in order to offer both variance reduction and resolution of signal-dominated modes (defined by (3.4.1)). For simplicity, in this chapter we use the same test function space used in the ODE setting [100] and leave an investigation of the performance of different test functions to future work. Define $\mathcal{S}$ to be the space of functions

$$\phi(v) = \begin{cases} C(v-a)^p(b-v)^q & a < v < b, \\ 0 & \text{otherwise}, \end{cases} \tag{3.4.2}$$

where $p, q \geq 1$ and v is a variable in time or space. The normalization

$$C = \frac{1}{p^p q^q}\left(\frac{p+q}{b-a}\right)^{p+q}$$

ensures that $\|\phi\|_\infty = 1$. Functions $\phi \in \mathcal{S}$ are non-negative, unimodal, compactly-supported in $[a, b]$, and have $\lfloor \min\{p, q\} \rfloor$ weak derivatives[8]. Larger p and q imply faster decay towards the endpoints (a, b) and for $p = q$ we refer to p as the *degree* of ϕ. See Figure 3.4.1 for a visualization of ψ and partial derivatives $D^{\alpha^s}\psi$ constructed from tensor products of functions from $\mathcal{S}$. In addition to having nice integration properties combined with the trapezoidal rule (see Lemma 1 of [100]), (a, b, p, q) can be chosen to localize $\widehat{\phi}$ around signal-dominated frequencies in $\mathcal{F}(\mathbf{U})$ using the fact that for any reference domain length $L \geq |b-a|$,

$$|\widehat{\phi}(k)| = o\left(\left(\frac{|b-a|}{L}|k|\right)^{-\lfloor \min\{p,q\} \rfloor - 1/2}\right).$$

To assemble the reference test function ψ from one-dimensional test functions $(\phi_d)_{d\in[D+1]} \subset \mathcal{S}$ along each coordinate, we must determine the parameters (a_d, b_d, p_d, q_d) in the formula (3.4.2) for each ϕ_d. For convenience we center $(\mathbf{Y}, \mathfrak{t})$ at the origin so that each ϕ_d is supported on a centered interval $[a_d, b_d] = [-b_d, b_d]$, where $b_d = m_d\Delta x$ for $d \in [D]$ and $b_{D+1} = m_{D+1}\Delta t$, and set $p_d = q_d$ so that ψ is symmetric[9]. In this way only $\mathbf{m} := (m_d)_{d\in[D+1]}$ and degrees $\mathbf{p} := (p_d)_{d\in[D+1]}$ need to be specified, hence the vectors $(\phi_d^{(\alpha_d^s)}(\mathbf{Y}_d))_{0\leq s\leq S}$ can be computed from an analogous function $\overline{\phi}_{p_d}$

[8] $\mathcal{S}$ can also be seen as a scaled subset of the Bernstein polynomials, which, among other considerations, are used in the construction of B-Splines [41].

[9] Test function asymmetry may provide an advantage in some cases, for instance along the time axis, however we do not investigate this here.

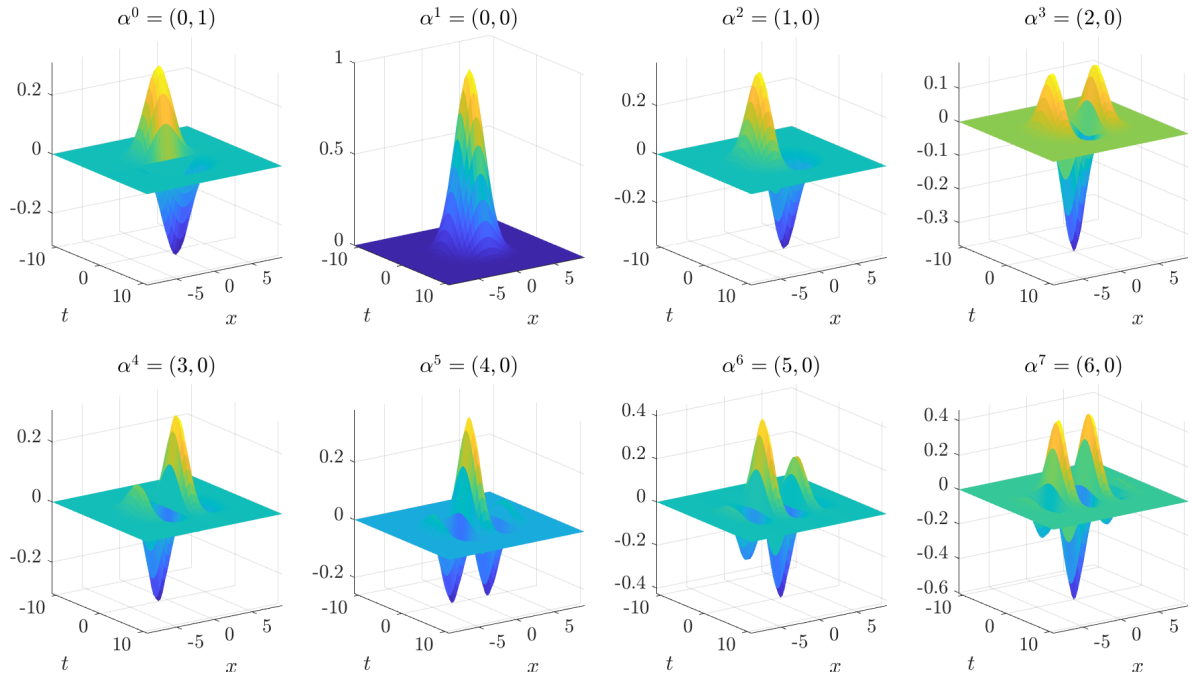


Figure 3.4.1: Plots of reference test function ψ and partial derivatives $D^{\alpha^s}\psi$ used for identification of the Kuramoto-Sivashinsky equation. The upper left plot shows $\partial_t\psi$, the bottom right shows $\partial_x^6\psi$. See Tables 3.5.1-3.5.3 for more details.

---

with support $[-1,1]$,

$$\overline{\phi}_{p_d}(v) := \begin{cases} (1-v^2)^{p_d}, & -1 < v < 1 \\ 0, & \text{otherwise,} \end{cases}$$

using

$$\phi_d^{(\alpha_d^s)}(\mathbf{Y}_d) = \frac{1}{b_d^{\alpha_d^s}} \overline{\phi}_{p_d}^{(\alpha_d^s)} \left(\frac{\mathbf{Y}_d}{b_d} \right) = \frac{1}{(m_d \Delta)^{\alpha_d^s}} \overline{\phi}_{p_d}^{(\alpha_d^s)} \left(\mathbf{n}_d \right),$$

where $\mathbf{n}_d := \{-1 + \frac{n}{m_d} \, : \, n \in \{0, \ldots, 2m_d\}\}$ is an associated scaled grid and $\Delta \in \{\Delta x, \Delta t\}$.

The discrete support lengths $\mathbf{m}$ and degrees $\mathbf{p}$ determine the smoothness of ψ, as well as its decay in real and in Fourier space, hence are critical to the method's performance. The degrees $\mathbf{p}$ can be chosen from $\mathbf{m}$ to ensure necessary smoothness and decay in real space using

$$p_d = \min \left\{ p \geq \overline{\alpha}_d + 1 \, : \, \overline{\phi}_p \left(1 - \frac{1}{m_d} \right) \leq \tau \right\}, \tag{3.4.3}$$

where $\overline{\alpha}_d := \max_{0 \leq s \leq S}(\alpha_d^s)$ is the maximum derivative along the dth coordinate and τ is a chosen (real-space) decay tolerance. By enforcing that ϕ_d decays to τ at the first interior gridpoint of its support, (3.4.3) controls the integration error (specifically, $\tau \leq \left(\frac{2m_d - 1}{m_d^2} \right)^q$ ensures $\mathcal{O}(\Delta x^{q+1})$ integration error for noise-free data), while $p \geq \overline{\alpha}_d + 1$ ensures that $\phi_d \in C^{\overline{\alpha}_d}(\mathbb{R})$, which is necessary to integrate by parts as many times as required by the multi-index set $\boldsymbol{\alpha}$. The steps for arriving at

the test function values on the reference grid $(\phi_d^{(\alpha_d^s)}(\mathbf{Y}_d))_{0\le s\le S}$ are contained in Algorithm 3.4.1.

In the examples below, we set $\tau = 10^{-10}$ throughout[10] and we use the method introduced in Appendix 3.A to choose $\mathbf{m}$, which involves estimating the critical wavenumber k^* (defined in (3.4.1)) between noise-dominated and signal-dominated modes of $\mathcal{F}(\mathbf{U})$. We also simplify things by choosing a single test function for all spatial coordinates, $\phi_x = \phi_1 = \phi_2 = \cdots = \phi_D$ where ϕ_x has degree p_x and support m_x, and a (possibly different) test function $\phi_t = \phi_{D+1}$ for the time axis with degree p_t and support m_t (recall that subscripts x and t are indices, not partial derivatives). This convention is used in the following sections.

Algorithm 3.4.1 WSINDy Test Function Creation
$(\phi_d^{(\alpha_d^s)}(\mathbf{Y}_d))_{0\le s\le S} = \textbf{get\_test\_fcns}\,(m_d, \tau;\ \mathbf{X}_d, \boldsymbol{\alpha})$

$N_d = \text{length}(\mathbf{X}_d)$
$\Delta x = \text{gridwidth}(\mathbf{X}_d)$
if $m_d > \frac{N_d-1}{2}$ or $m_d \le 1$ **then**
  return ("ERROR: invalid support size m_d")
  BREAK
end if
Set $\overline{\alpha}_d = \max_{0\le s\le S}(\alpha_d^s)$
Solve $p_d = \min\left\{p \ge \overline{\alpha}_d + 1\ :\ \overline{\phi}_p\left(1 - \frac{1}{m_d}\right) \le \tau\right\}$
Initialize $\mathbf{A} = \mathbf{0} \in \mathbb{R}^{(S+1)\times(2m_d+1)}$
Set $\mathbf{n}_d := \{-1 + \frac{n}{m_d}\ :\ n \in \{0,\dots,2m_d\}\}$
for $s = 0 : S$ **do**
  Compute analytical order-(α_d^s) derivatives $\mathbf{A}_s = \overline{\phi}_{p_d}^{(\alpha_d^s)}(\mathbf{n}_d)$
  Set $\phi_d^{(\alpha_d^s)}(\mathbf{Y}_d) = \frac{1}{(m_d\Delta x)^{\alpha_d^s}}\mathbf{A}_s$
end for

3.4.2 Sparsification

To enforce a sparse solution we present a *modified sequential-thresholding least-squares* algorithm MSTLS($\mathbf{G}, \mathbf{b}$; λ), defined in (3.4.6), which accounts for terms that are outside of the dominant balance physics of the data, as determined by the left-hand side $\mathbf{b}$, as well as terms with small coefficients. We then utilize the loss function

$$\mathcal{L}(\lambda) = \frac{\left\|\mathbf{G}(\mathbf{w}^\lambda - \mathbf{w}^{LS})\right\|_2}{\left\|\mathbf{G}\mathbf{w}^{LS}\right\|_2} + \frac{\#\{\mathcal{I}^\lambda\}}{SJ} \tag{3.4.4}$$

to select an optimal threshold $\widehat{\lambda}$, where $\mathbf{w}^\lambda$ is the output of MSTLS($\mathbf{G}, \mathbf{b}$; λ) defined in equation (3.4.6), $\#\{\cdot\}$ denotes cardinality, $\mathcal{I}^\lambda := \{1 \le i \le SJ\ :\ \mathbf{w}_i^\lambda \ne 0\}$ is the index set of non-zero coefficients of $\mathbf{w}^\lambda$, $\mathbf{w}^{LS} := \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{b}$ is the least squares solution, and SJ is the total number of terms in the library (S differential operators and J functions of the data). The first term in

[10] WSINDy appears not to be particularly sensitive to τ, similar results were obtained for $\tau = 10^{-6}, 10^{-10}, 10^{-16}$.

$\mathcal{L}$ penalize the distance between $\mathbf{G}\mathbf{w}^{LS}$ (the projection of $\mathbf{b}$ onto the range of $\mathbf{G}$) and $\mathbf{G}\mathbf{w}^{\lambda}$ (the projection of $\mathbf{b}$ onto the columns of $\mathbf{G}$ restricted to $\mathcal{I}^{\lambda}$), while the second term penalizes the number of nonzero terms in the resulting model. The normalization simply enforces $\mathcal{L}(0) = \mathcal{L}(\infty) = 1$.

The MSTLS($\mathbf{G}, \mathbf{b}$; λ) iteration is as follows. For a given $\lambda \geq 0$, define the set of lower bounds L^{λ} and upper bound U^{λ} by

$$\begin{cases} L_i^{\lambda} = \lambda \max\left\{1, \ \dfrac{\|\mathbf{b}\|}{\|\mathbf{G}_i\|}\right\} \\ U_i^{\lambda} = \dfrac{1}{\lambda} \min\left\{1, \ \dfrac{\|\mathbf{b}\|}{\|\mathbf{G}_i\|}\right\} \end{cases}, \qquad 1 \leq i \leq SJ. \tag{3.4.5}$$

Then with $\mathbf{w}^0 = \mathbf{w}^{LS}$, define the iterates

$$\text{MSTLS}(\mathbf{G}, \mathbf{b}; \lambda) \qquad \begin{cases} \mathcal{I}^{\ell} = \{1 \leq i \leq SJ \ : \ L_i^{\lambda} \leq |\mathbf{w}_i^{\ell}| \leq U_i^{\lambda}\} \\ \mathbf{w}^{\ell+1} = \underset{\text{supp}(\mathbf{w}) \subset \mathcal{I}^{\ell}}{\arg\min} \|\mathbf{G}\mathbf{w} - \mathbf{b}\|_2^2 . \end{cases} \tag{3.4.6}$$

The constraint $L_i^{\lambda} \leq |\mathbf{w}_i^{\ell}| \leq U_i^{\lambda}$ is clearly more restrictive than standard sequential thresholding, but it enforces two desired qualities of the model: (i) that the coefficients $\mathbf{w}^{\lambda}$ do not differ too much from 1, since 1 is the coefficient of the "evolution" term $D^{\alpha^0}u$ (assumed known), and (ii) that the ratio $\|\mathbf{w}_i\mathbf{G}_i\|_2 \,/\, \|\mathbf{b}\|_2$ lies in $[\lambda, \lambda^{-1}]$, enforcing an empirical dominant balance rule (e.g. $\lambda = 0.01$ allows terms in the model to be at most two orders of magnitude from $D^{\alpha^0}u$). Using previous results on the convergence of STLS [158], for MSTLS($\mathbf{G}, \mathbf{b}; \lambda$) we employ the stopping criteria $\mathcal{I}^{\ell} \setminus \mathcal{I}^{\ell+1} = \emptyset$, which must occur for some $\ell \leq SJ$. The overall sparsification algorithm MSTLS($\mathbf{G}, \mathbf{b}$; $\mathcal{L}, \boldsymbol{\lambda}$) is

$$\text{MSTLS}(\mathbf{G}, \mathbf{b}; \ \mathcal{L}, \boldsymbol{\lambda}) \qquad \begin{cases} \widehat{\lambda} = \min\left\{\lambda \in \boldsymbol{\lambda} \ : \ \mathcal{L}(\lambda) = \min\limits_{\lambda \in \boldsymbol{\lambda}} \mathcal{L}(\lambda)\right\} \\ \widehat{\mathbf{w}} = \text{MSTLS}(\mathbf{G}, \mathbf{b}; \ \widehat{\lambda}), \end{cases} \tag{3.4.7}$$

where $\boldsymbol{\lambda}$ is a finite set of candidate thresholds[11]. The learned threshold $\widehat{\lambda}$ is the smallest minimizer of $\mathcal{L}$ over the range $\boldsymbol{\lambda}$ and hence marks the boundary between identification and misidentification of the minimum-cost model, such that $\{\lambda \in \boldsymbol{\lambda} \ : \ \lambda < \widehat{\lambda}\}$ results in overfitting. A similar learning method for $\widehat{\lambda}$ combining STLS and Tikhonov regularization (or *ridge regression*) was developed in [117]. We have found that our approach of combining MSTLS($\mathbf{G}, \mathbf{b}$; $\mathcal{L}, \boldsymbol{\lambda}$) with rescaling, as introduced in the next section, regularizes the sparse regression problem in the case of large model libraries without adding hyperparameters[12] and definitely deserves further study.

[11] Other methods of minimizing $\mathcal{L}$ can be used, however minimizers are not unique (there exists a set of minimizers - see Figure 3.5.3). Our approach is efficient and returns the minimizer $\widehat{\lambda}$ which has the useful characterization of defining the thresholds λ that result in overfitting.

[12] Tikhonov regularization involves solving $\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{G}\mathbf{w} - \mathbf{b}\|_2^2 + \gamma^2 \|\mathbf{w}\|_2^2$

3.4.3 Regularization through Rescaling

Construction of the linear system $\mathbf{b} = \mathbf{G}\mathbf{w}$ involves taking nonlinear transformations of the data $f_j(\mathbf{U})$ and then integrating against $D^{\alpha^s}\psi$, which oscillates for large $|\alpha^s|$. This can lead to a large condition number $\kappa(\mathbf{G})$ and prevent accurate inference of the true model coefficients $\mathbf{w}^\star$, especially when the underlying data is poorly scaled[13]. In particular, identification of polynomial terms such as $\partial_x(u^2)$ from a large library of polynomial terms is ill-conditioned for large (or small) amplitude data. Naively rescaling the data can easily lead to unreliable inference of model coefficients, since characteristic scales often effect the dynamics in nontrivial ways. For example, solution amplitude determines the wavespeed in the inviscid Burgers and Korteweg-de Vries equations, hence the solution and space-time coordinates must be rescaled in a principled manner in order to preserve the dynamics. To overcome this problem we propose to rescale the data using scale invariance of the PDE and choose scales that achieve a lower condition number, as described below. This approach allows for reliable identification of the Burgers and KdV equations from highly-corrupted large-amplitude data ($\mathbf{U} \sim \mathcal{O}(10^3)$, see Section 3.5.4).

First, we note that the true model is scale invariant in the following way. If u solves (3.3.1), then for any scales $\gamma_x, \gamma_t, \gamma_u > 0$, the rescaled function

$$\widetilde{u}(\widetilde{x}, \widetilde{t}) := \gamma_u \, u\left(\frac{\widetilde{x}}{\gamma_x}, \frac{\widetilde{t}}{\gamma_t}\right) := \gamma_u \, u(x,t)$$

solves

$$\widetilde{D}^{\alpha^0}\widetilde{u} = \sum_{s=1}^{S}\sum_{j=1}^{J} \widetilde{\mathbf{w}}_{(s-1)J+j} \widetilde{D}^{\alpha^s}\widetilde{f}_j(\widetilde{u})$$

where $\widetilde{D}^{\alpha^s}$ denotes differentiation with respect to $(\widetilde{x}, \widetilde{t}) = (\gamma_x x, \gamma_t t)$. For homogeneous functions f_j with power β_j, we have $\widetilde{f}_j(\widetilde{u}) = f_j(\widetilde{u}) = \gamma_u^{\beta_j} f_j(u)$, otherwise $\widetilde{f}_j(\widetilde{u}) = f_j\left(\frac{\widetilde{u}}{\gamma_u}\right) = f_j(u)$ (in which case we set $\beta_j = 0$). The linear system in the rescaled coordinates $\widetilde{\mathbf{b}} = \widetilde{\mathbf{G}}\widetilde{\mathbf{w}}$ is constructed by discretizing the convolutional weak form as before but with a reference test function $\widetilde{\psi}$ on the rescaled grid $\widetilde{\Omega}_R$. We recover the coefficients[14] $\widehat{\mathbf{w}}$ at the original scales by setting $\widehat{\mathbf{w}} = \mathbf{M}\widetilde{\mathbf{w}}$, where $\mathbf{M} = \mathrm{diag}\,(\boldsymbol{\mu})$ is the diagonal matrix with entries

$$\mu_{(s-1)J+j} := \gamma_u^{-(\beta_j - 1)} \, \gamma_x^{\sum_{d=1}^{D}(\alpha_d^s - \alpha_d^0)} \, \gamma_t^{(\alpha_{D+1}^s - \alpha_{D+1}^0)}. \tag{3.4.8}$$

There is flexibility in choosing the scales $\gamma_u, \gamma_x, \gamma_t$, and a natural choice is to enforce that the columns of $\widetilde{\mathbf{G}}$ are similar in norm. Motivated by this, we find that for polynomial and trigonometric

[13] A common remedy for this is to scale $\mathbf{G}$ to have columns of unit 2-norm, however this has no connection with the underlying physics.

[14] Note that thresholding in equation (3.4.6) occurs on $\widehat{\mathbf{w}}$ and the terms $\frac{\|\mathbf{b}\|}{\|\mathbf{G}_i\|}$ in the bounds (3.4.5) become $\frac{\|\widetilde{\mathbf{b}}\|}{\mu_i\|\widetilde{\mathbf{G}}_i\|}$.

libraries, the scales[15]

$$\gamma_u = \left(\frac{\|\mathbf{U}\|_{2'}}{\left\|\mathbf{U}^{\overline{\beta}}\right\|_{2'}}\right)^{1/\overline{\beta}}, \quad \gamma_x = \frac{1}{m_x \Delta x}\left(\left(\frac{p_x}{\frac{\overline{\alpha}_x}{2}}\right)\overline{\alpha}_x!\right)^{1/\overline{\alpha}_x}, \quad \gamma_t = \frac{1}{m_t \Delta t}\left(\left(\frac{p_t}{\frac{\overline{\alpha}_t}{2}}\right)\overline{\alpha}_t!\right)^{1/\overline{\alpha}_t} \tag{3.4.9}$$

are sufficient to regularize ill-conditioning due to poor scaling. Here $\overline{\alpha}_x$ and $\overline{\alpha}_t$ are the maximum spatial and temporal derivatives appearing in the library and $\overline{\beta} = \max_j \beta_j$ is the highest monomial power of the functions $(f_j)_{j\in[J]}$. From (3.4.9) we get that

$$\left\|\widetilde{\mathbf{U}}^{\overline{\beta}}\right\|_{2'} = \|\mathbf{U}\|_{2'}$$

and

$$\max_s \|\Psi^s\|_{1'} \leq \max_s \left\|D^{\alpha^s}\widetilde{\psi}\right\|_\infty |\widetilde{\Omega}_R| \leq |\widetilde{\Omega}_R|,$$

hence, using Young's inequality for convolutions,

$$\left\|\Psi^s * \widetilde{\mathbf{U}}^{\overline{\beta}}\right\|_{2'} \leq \|\Psi^s\|_{1'} \left\|\widetilde{\mathbf{U}}^{\overline{\beta}}\right\|_{2'} \leq \left|\widetilde{\Omega}_R\right| \|\mathbf{U}\|_{2'}.$$

This shows that with scales $\gamma_u, \gamma_x, \gamma_t$ set according to (3.4.9), the columns of $\widetilde{\mathbf{G}}$ are close in norm to the original dataset $\mathbf{U}$. Similar scales $\gamma_x, \gamma_t, \gamma_u$ can be chosen for different model libraries and reference test functions, and a more refined analysis will lead to scales that achieve closer agreement in norm. In the examples below we rescale the data and coordinates according to (3.4.9), which results in a low condition number $\kappa(\widetilde{\mathbf{G}})$ (see Table 3.5.3). Throughout what follows, quantities defined over scaled coordinates will be denoted by tildes.

3.4.4 Query Points and Subsampling

Placement of $\{(\mathbf{x}_k, t_k)\}_{k\in[K]}$ determines which regions of the observed data will most influence the recovered model[16]. In WSINDy for ODEs ([100]), an adaptive algorithm was designed for placement of test functions near steep gradients along the trajectory. Improvements in this direction in the PDE setting are a topic of active research, however, for simplicity in this chapter we uniformly subsample $\{(\mathbf{x}_k, t_k)\}_{k\in[K]}$ from $(\mathbf{X}, \mathbf{t})$ using subsampling frequencies $\mathbf{s} = (s_1, \ldots, s_{D+1})$ along each coordinate, specified by the user. That is, along each one-dimensional grid $\mathbf{X}_d$, $\lfloor \frac{N_d - 2m_d}{s_d} \rfloor$ points are selected with uniform spacing $s_d \Delta x$ for $d \in [D]$ and $s_{D+1}\Delta t$ for $d = D+1$. This results in a $(D+1)$-dimensional coarse grid with dimensions $\lfloor \frac{N_1-2m_1}{s_1} \rfloor \times \cdots \times \lfloor \frac{N_{D+1}-2m_{D+1}}{s_{D+1}} \rfloor$, which determines the number of query points

$$K = \prod_{d=1}^{D+1} \left\lfloor \frac{N_d - 2m_d}{s_d} \right\rfloor. \tag{3.4.10}$$

[15] Here $\|\mathbf{U}\|_{2'}$ is the 2-norm of $\mathbf{U}$ stretched into a column vector (and similarly for $\|\cdot\|_{1'}$).

[16] Note that the projection operation in (3.3.12) restricts the admissable set of query points to those for which $\psi(\mathbf{x}_k - x, t_k - t)$ is compactly supported within $\Omega \times [0, T]$, which is necessary for integration by parts to be valid.

Hyperparameter	Domain	Description
$(f_j)_{j\in[J]}$	$C(\mathbb{R})$	trial function library
$\boldsymbol{\alpha} = (\alpha_s)_{s=0,\dots,S}$	$\mathbb{N}^{(S+1)\times(D+1)}$	partial derivative multi-indices
$\mathbf{m} = (m_d)_{d\in[D+1]}$	$\mathbb{N}^{D+1}$	discrete support lengths of 1D test functions $(\phi_d)_{d\in[D+1]}$
$\mathbf{s} = (s_d)_{d\in[D+1]}$	$\mathbb{N}^{D+1}$	subsampling frequencies for query points $\{(\mathbf{x}_k, t_k)\}_{k\in[K]}$
$\boldsymbol{\lambda}$	$[0,\infty)$	search space for sparsity threshold $\hat{\lambda}$
τ	$(0,1]$	ψ real-space decay tolerance

Table 3.4.1: Hyperparameters for the WSINDy Algorithm 3.4.2. Note that f_j piecewise continuous is sufficient (we just need convergence of the trapezoidal rule), $\mathbf{m}$ may be replaced by a spectral-decay tolerance $\widehat{\tau} > 0$ if test functions are automatically selected from the data using the method in Appendix 3.A, and K is determined from $\mathbf{m}$ and $\mathbf{s}$ using (3.4.10).

3.4.5 Model Library

The model library is determined by the nonlinear functions $(f_j)_{j\in[J]}$ and the partial derivative indices $\boldsymbol{\alpha}$ and is crucial to the well-posedness of the recovery problem. In the examples below we choose $(f_j)_{j\in[J]}$ to be polynomials and trigonometric functions as these sets are dense in many relevant function spaces. When the true PDE does not contain cross derivatives (e.g. $\frac{\partial^2}{\partial x_1 \partial x_2}$), we remove them from the derivative library $\boldsymbol{\alpha}$ and note that including these terms does not have a significant impact on the results.

Algorithm 3.4.2 WSINDy for PDEs
$(\widehat{\mathbf{w}}, \hat{\lambda}) = \mathbf{WSINDy}((f_j)_{j\in[J]}, \boldsymbol{\alpha}, \mathbf{m}, \mathbf{s}, \boldsymbol{\lambda}, \tau;\ \mathbf{U}, (\mathbf{X}, \mathbf{t}))$

for $d = 1 : D+1$ **do**
  Compute $(\phi_d^{(\alpha_d^s)}(\mathbf{Y}_d))_{0\le s\le S} = \mathbf{get\_test\_fcns}\,(m_d, \tau;\ \mathbf{X}_d, \boldsymbol{\alpha})$ using Algorithm 3.4.1
end for
Compute scales $\{\gamma_u, (\gamma_d)_{d=1}^{D+1}\}$ and scale matrix $\mathbf{M} = \mathrm{diag}(\boldsymbol{\mu})$ using (3.4.9)
Subsample query points $\{(\mathbf{x}_k, t_k)\}_{k\in[K]} \subset (\mathbf{X}, \mathbf{t})$ using subsampling frequencies $\mathbf{s} = (s_1, s_2, \dots, s_{D+1})$;
Compute left-hand side $\widetilde{\mathbf{b}} = \widetilde{\Psi}^0 * \widetilde{\mathbf{U}}$ over $\{(\mathbf{x}_k, t_k)\}_{k\in[K]}$ using FFT and separability of ψ;
for $j = 1 : J$ **do**
  Compute $\widetilde{f_j}(\widetilde{\mathbf{U}})$;
  for $s = 1 : S$ **do**
    Compute column $(s-1)J+j$ of Gram matrix $\widetilde{\mathbf{G}}_{:,(s-1)J+j} = \widetilde{\Psi}^s * \widetilde{f_j}(\widetilde{\mathbf{U}})$ over $\{(\mathbf{x}_k, t_k)\}_{k\in[K]}$ using FFT and separability of ψ
  end for
end for
$(\widehat{\mathbf{w}}, \widehat{\lambda}) = \mathrm{MSTLS}(\widetilde{\mathbf{G}}, \widetilde{\mathbf{b}};\ \mathcal{L}, \boldsymbol{\lambda})$

3.5 Examples

Inviscid Burgers (IB)	$\partial_t u = -\frac{1}{2}\partial_x(u^2)$
Korteweg-de Vries (KdV)	$\partial_t u = -\frac{1}{2}\partial_x(u^2) - \partial_{xxx}u$
Kuramoto-Sivashinsky (KS)	$\partial_t u = -\frac{1}{2}\partial_x(u^2) - \partial_{xx}u - \partial_{xxxx}u$
Nonlinear Schrödinger (NLS)	$\begin{cases}\partial_t u = \frac{1}{2}\partial_{xx}v + u^2v + v^3 \\ \partial_t v = -\frac{1}{2}\partial_{xx}u - uv^2 - u^3\end{cases}$
Anisotropic Porous Medium (PM)	$\partial_t u = (0.3)\partial_{xx}(u^2) - (0.8)\partial_{xy}(u^2) + \partial_{yy}(u^2)$
Sine-Gordon (SG)	$\partial_{tt} u = \partial_{xx}u + \partial_{yy}u - \sin(u)$
Reaction-Diffusion (RD)	$\begin{cases}\partial_t u = \frac{1}{10}\partial_{xx}u + \frac{1}{10}\partial_{yy}u - uv^2 - u^3 + v^3 + u^2v + u \\ \partial_t v = \frac{1}{10}\partial_{xx}v + \frac{1}{10}\partial_{yy}v + v - uv^2 - u^3 - v^3 - u^2v\end{cases}$
2D Navier-Stokes (NS)	$\partial_t \omega = -\partial_x(\omega u) - \partial_y(\omega v) + \frac{1}{100}\partial_{xx}\omega + \frac{1}{100}\partial_{yy}\omega$

Table 3.5.1: PDEs used in numerical experiments, written in the form identified by WSINDy. Domain specification and boundary conditions are given in Appendix 3.B.

We now demonstrate the effectiveness of WSINDy by recovering the PDEs listed in Table 3.5.1 over a range of noise levels. These examples show that WSINDy provides orders of magnitude improvements over derivative-based methods [117], with reliable and accurate recovery of four out of the eight PDEs under noise levels as high as 100% (defined in (3.5.1) and (3.5.2)) and for all examples under 20% noise. In contrast to the weak recovery methods in [114, 57], WSINDy uses (i) the convolutional weak form (3.3.6) and FFT-based implementation (3.3.12), (ii) improved thresholding and automatic selection of the sparsity threshold $\hat{\lambda}$ via (3.4.6) and (3.4.7), and (iii) rescaling using (3.4.9). The effects of these improvements are discussed in Sections 3.5.4 and 3.5.5.

To test robustness to noise, a noise ratio σ_{NR} is specified and a synthetic "observed" dataset

$$\mathbf{U} = \mathbf{U}^\star + \epsilon$$

is obtained from a simulation $\mathbf{U}^\star$ of the true PDE[17] by adding i.i.d. Gaussian noise with variance σ^2 to each data point, where

$$\sigma := \sigma_{NR}\,\|\mathbf{U}^\star\|_{RMS} := \sigma_{NR}\left(\frac{1}{(N_1\cdots N_D N_{D+1})}\sum_{k_1=1}^{N_1}\cdots\sum_{k_{D+1}=1}^{N_{D+1}}\left(\mathbf{U}^\star_{k_1,\ldots,k_{D+1}}\right)^2\right)^{1/2}. \tag{3.5.1}$$

We examine noise ratios σ_{NR} in the range $[0,1]$ and often refer to the noise level as σ_{NR} or equivalently that the data contains $100\sigma_{NR}$% noise. We note that the resulting true noise ratio

$$\sigma^\star_{NR} := \frac{\|\epsilon\|_{RMS}}{\|\mathbf{U}^\star\|_{RMS}} \tag{3.5.2}$$

[17] Details on the numerical methods and boundary conditions used to simulate each PDE can be found in Appendix 3.B.

matches the specified σ_{NR} to at least four significant digits in all cases and so we only list σ_{NR}. When the state variable is vector-valued, as with the nonlinear Schrödinger, reaction-diffusion, and Navier-Stokes equations (see Table 3.5.1), a separate noise variance σ^2 is computed for each vector component so that the noise ratio σ_{NR} of each component satisfies (3.5.2).

3.5.1 Performance Measures

To measure the ability of the algorithm to correctly identify the terms having nonzero coefficients, we use the *true positivity ratio* (introduced in [80]) defined by

$$\mathrm{TPR}(\widehat{\mathbf{w}}) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{3.5.3}$$

where TP is the number of correctly identified nonzero coefficients, FN is the number of coefficients falsely identified as zero, and FP is the number of coefficients falsely identified as nonzero. Identification of the true model results in a TPR of 1, while identification of half of the correct nonzero terms and no falsely identified nonzero terms results in TPR of 0.5 (e.g. the 2D Euler equations $\partial_t \omega = -\partial_x(\omega u) - \partial_y(\omega v)$ result in a TPR of 0.5 if the underlying true model is the 2D Navier-Stokes vorticity equation). We will see that in several cases that the average TPR remains above 0.95 even as the noise level approaches 1. The loss function $\mathcal{L}(\lambda)$ (defined in (3.4.4)) and the resulting learned sparsity threshold $\widehat{\lambda}$ (defined in (3.4.7)) provide additional information on the algorithm's ability to identify the correct model terms with respect to the noise level. In particular, sensitivity to the sparsity threshold suggests that automatic selection of $\widehat{\lambda}$ is essential to successful recovery in the relatively large noise regime.

To assess the accuracy of the recovered coefficients we use two metrics. We measure the maximum error in the true non-zero coefficients using

$$E_\infty(\widehat{\mathbf{w}}) := \max_{\{j \,:\, \mathbf{w}_j^\star \neq 0\}} \frac{|\widehat{\mathbf{w}}_j - \mathbf{w}_j^\star|}{|\mathbf{w}_j^\star|}, \tag{3.5.4}$$

where $|\cdot|$ denotes absolute value, and the ℓ^2 distance in parameter space using

$$E_2(\widehat{\mathbf{w}}) := \frac{\|\widehat{\mathbf{w}} - \mathbf{w}^\star\|_{RMS}}{\|\mathbf{w}^\star\|_{RMS}}. \tag{3.5.5}$$

E_∞ determines the number of significant digits in the recovered true coefficients while E_2 provides information about the magnitudes of coefficients that are falsely identified as nonzero. Often when a term is falsely identified and the resulting nonzero coefficient is small, a larger sparsity factor will result in idenfitication of the true model.

Finally, when $\mathrm{TPR}(\widehat{\mathbf{w}}) = 1$, we report the prediction accuracy between the true data $\mathbf{U}^\star$ and a numerical solution $\mathbf{U}^{dd}$ to the data-driven PDE using the same initial conditions. We compute the

relative L_2 error $P_t(\widehat{\mathbf{w}})$ at time $t = 0.5T$ (i.e. at the half-way point in time) defined by

$$P_t(\widehat{\mathbf{w}}) := \frac{\left\|\mathbf{U}_t^{dd} - \mathbf{U}_t^{\star}\right\|_{RMS}}{\|\mathbf{U}_t^{\star}\|_{RMS}} \tag{3.5.6}$$

where $\mathbf{U}_t^{dd}$, $\mathbf{U}_t^{\star}$ denote the numerical solutions over the spatial domain at time t. Since solutions to the data-driven dynamics and the true dynamics will eventually drift apart, we also measure

$$T_{\text{tol}}(\widehat{\mathbf{w}}) := \frac{1}{T}\inf\{t \in [0,T] \ : \ P_t(\widehat{\mathbf{w}}) > \text{tol}\}, \tag{3.5.7}$$

or, the first time t (relative to the final time T) that the numerical solution $\mathbf{U}_t^{dd}$ reaches a relative L_2 distance of tol from the truth. The minimum in (3.5.7) is computed over $t \in \mathbf{t}$ and we set tol $= 0.1$. We provide results for $P_{0.5T}(\widehat{\mathbf{w}})$ and $T_{0.1}(\widehat{\mathbf{w}})$ averaged over the weights $\widehat{\mathbf{w}}$ satisfying $\text{TPR}(\widehat{\mathbf{w}}) = 1$.

For each system in Table 3.5.1 and each noise level $\sigma_{NR} \in \{0.025q \ : \ q \in \{0,\dots,40\}\}$ we run WSINDy on 200 instantiations of noise[18] and average the results of error statistics (3.5.3)-(3.5.7). Computations were carried out on a University of Colorado Boulder Blanca Condo cluster[19].

3.5.2 Implementation Details

The hyperparameters used in WSINDy applied to each of the PDEs in Table 3.5.1 are given in Table 3.5.2. To select test function discrete support lengths we used a combination of the changepoint method[20] described in Appendix 3.A and manual tuning. Across all examples the real-space decay tolerance for test functions is fixed at $\tau = 10^{-10}$.

In computing a sparse solution $\widehat{\mathbf{w}} = \text{MSTLS}(\mathbf{G},\mathbf{b};\mathcal{L},\boldsymbol{\lambda})$ (see equation (3.4.7)), the search space $\boldsymbol{\lambda}$ for the learned threshold $\widehat{\lambda}$ is fixed for all examples at:

$$\boldsymbol{\lambda} = \left\{10^{-4+j\frac{4}{49}} \ : \ j \in \{0,\dots,49\}\right\},$$

in other words $\boldsymbol{\lambda}$ contains 50 points with $\log_{10}(\boldsymbol{\lambda})$ equally spaced from -4 to 0. This implies a stopping criteria of $50SJ$ thresholding iterations[21].

We fix the subsampling frequencies (s_x, s_t) to $(\frac{N_1}{50}, \frac{N_2}{50})$ for PDEs in one spatial dimension and to $(\frac{N_1}{25}, \frac{N_3}{25})$ for two spatial dimensions, where the dimensions (N_1, N_2, N_3) depend on the dataset. Additional information about the convolutional weak discretization is included in Table 3.5.3, such as the dimensions and condition number of the rescaled Gram matrix $\widetilde{\mathbf{G}}$ (computed from a typical

[18] We find that 200 runs sufficiently reduces variance in the results.

[19] 2X Intel Xeon 5218 at 2.3 GHz with 22 MB cache, 16 cores per cpu, and 384 GB ram.

[20] For Burgers, KdV, and KS we set $\widehat{\tau} = 3$ (defined in Appendix 3.A.2) while for NLS, PM, SG, RD, and NS we used $\widehat{\tau} = 1$. For KS and NLS we chose (m_x, m_t) values nearby that had better performance.

[21] In the examples shown here we observed an average of 5 thresholding iterations and a maximum of 14 in any given inner MSTLS$(\mathbf{G},\mathbf{b};\lambda)$ loop (i.e. for each $\lambda \in \boldsymbol{\lambda}$ as in equation (3.4.6)), hence in practice the full MSTLS$(\mathbf{G},\mathbf{b};\mathcal{L},\boldsymbol{\lambda})$ algorithm requires far fewer iterations than the theoretical maximum of $\#\{\boldsymbol{\lambda}\}SJ$.

PDE	$\mathbf{U}$	f_j	$\boldsymbol{\alpha}$	(m_x, m_t)	(s_x, s_t)
IB	256×256	$(u^{j-1})_{j\in[7]}$	$((\ell,0))_{0\le\ell\le6}$	$(60,60)$	$(5,5)$
KdV	400×601	$(u^{j-1})_{j\in[7]}$	$((\ell,0))_{0\le\ell\le6}$	$(45,80)$	$(8,12)$
KS	256×301	$(u^{j-1})_{j\in[7]}$	$((\ell,0))_{0\le\ell\le6}$	$(23,22)$	$(5,6)$
NLS	$2 \times 256 \times 251$	$(u^i v^j)_{0\le i+j\le6}$	$((\ell,0))_{0\le\ell\le6}$	$(19,25)$	$(5,5)$
PM	$200 \times 200 \times 128$	$(u^{i-1})_{i\in[5]}$	$((\ell_1,\ell_2,0))_{0\le\ell_1,\ell_2\le4}$	$(37,20)$	$(8,5)$
SG	$129 \times 403 \times 205$	$(u^{i-1})_{i\in[5]}$, $(\sin(ju),\cos(ju))_{j=1,2}$	$((\ell,0,0),(0,\ell,0))_{0\le\ell\le4}$	$(40,25)$	$(5,8)$
RD	$2 \times 256 \times 256 \times 201$	$(u^i v^j)_{0\le i+j\le4}$	$((\ell,0,0),(0,\ell,0))_{0\le\ell\le5}$	$(13,14)$	$(13,12)$
NS	$3 \times 324 \times 149 \times 201$	$\begin{cases}(\omega^i u^j v^k)_{0\le i+j+k\le2}, & \lvert\alpha^s\rvert = 0\\ (\omega^i u^j v^k)_{0\le i+j+k\le3, i>0}, & \lvert\alpha^s\rvert > 0\end{cases}$	$((\ell,0,0),(0,\ell,0))_{0\le\ell\le2}$	$(31,14)$	$(12,8)$

Table 3.5.2: WSINDy hyperparameters used to identify each example PDE.

PDE	$\widetilde{\mathbf{G}}$	$\kappa(\widetilde{\mathbf{G}})$	(p_x, p_t)	$(\gamma_u, \gamma_x, \gamma_t)$	Walltime (sec)
IB	784×43	1.4×10^6	$(7,7)$	$(4.5 \times 10^{-4}, 0.0029, 1.1)$	0.12
KdV	1443×43	3.2×10^6	$(8,7)$	$(5.7 \times 10^{-4}, 8.3, 1250)$	0.39
KS	1806×43	3.7×10^3	$(10,10)$	$(0.26, 0.74, 0.091)$	0.24
NLS	1804×190	1.2×10^5	$(11,10)$	$(0.33, 3.1, 9.4)$	2.5
PM	4608×65	2.4×10^4	$(8,10)$	$(1.6, 2.7, 3.2)$	16
SG	13000×73	1.3×10^4	$(8,10)$	$(0.23, 8.1, 8.1)$	29
RD	11638×181	4.5×10^3	$(13,12)$	$(0.86, 6.5, 1.4)$	75
NS	3872×50	8.2×10^2	$(9,12)$	$(0.53, 0.72, 2.4)$	12

Table 3.5.3: Additional specifications resulting from the choices in Table 3.5.2. The last column shows the start-to-finish walltime of Algorithm 3.4.2 with all computations in serial measured on a laptop with an 8-core Intel i7-2670QM CPU with 2.2 GHz and 8 GB of RAM.

dataset with 20% noise), test function polynomial degrees (p_x, p_t), scale factors $(\gamma_u, \gamma_x, \gamma_t)$, and start-to-finish walltime of Algorithm 3.4.2 with all computations performed serially on a laptop with an 8-core Intel i7-2670QM CPU with 2.2 GHz and 8 GB of RAM.

3.5.3 Comments on Chosen Examples

The primary reason for choosing the examples in Table 3.5.1 is to demonstrate that WSINDy can successfully recover models over a wide range of physical phenomena such as spatiotemporal chaos, nonlinear waves, nonlinear diffusion, shock-forming solutions, complex limit cycles, and pattern formation in reaction diffusion equations.

Recovery of the inviscid Burgers and anisotropic porous medium equations demonstrates (i) that WSINDy can discover PDEs from solutions that can only be understood in a weak sense and (ii) that discovery in this case is just as accurate and robust to noise and scaling as with smooth data (i.e. no special modifications of the algorithm are required to discover models from non-smooth data, as conjectured in [57]). We use analytical weak solutions, with inviscid Burgers data forming a shock which propagates at constant speed (see Figure 3.5.1 for plots of the characteristic curves) and porous medium data having a jump in the gradient ∇u. In addition, we discover the porous medium equation using an anisotropic diffusivity tensor to demonstate that WSINDy can identify the cross-diffusion term $\partial_{xy}(u^2)$ to high accuracy from a large candidate model library.

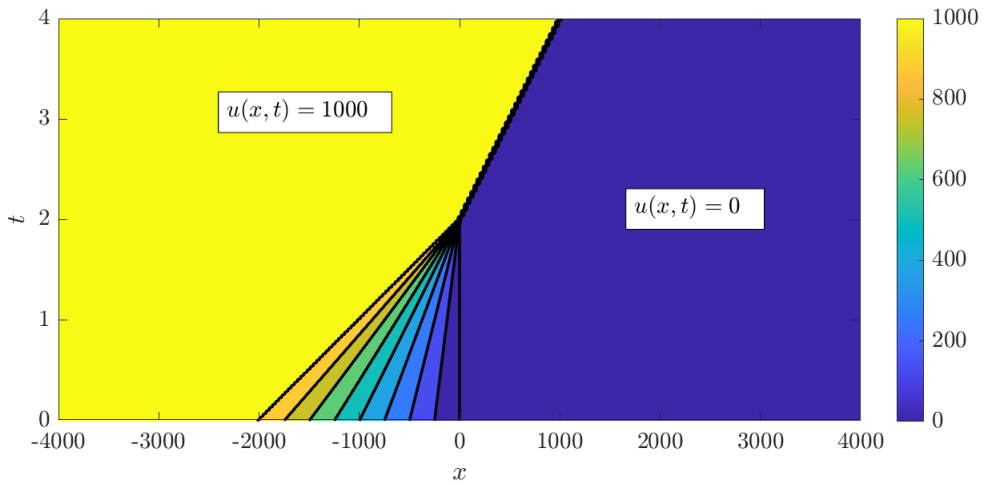


Figure 3.5.1: Characteristics of the shock-forming solution (3.B.2) used to identify the inviscid Burgers equation. A shock forms at time $t = 2$ and travels along the line $x = 500(t - 2)$.

The inviscid Burgers and Korteweg-de Vries equations demonstrate that WSINDy successfully recovers the correct models for nonlinear transport data with large amplitude. Both datasets have mean amplitudes on the order of 10^3 (in addition KdV is given over a short time window of $\mathbf{t} \in [0, 10^{-3}]$), and hence are not identifiable from large polynomial libraries using naive approaches. The sparsification and rescaling measures in Sections 3.4.2 and 3.4.3 are essential to removing this barrier.

The Sine-Gordon equation[22] is used to show both that trigonometric library terms can easily be identified alongside polynomials and that hyperbolic problems do not seem to present further challenges. Discovery of the Sine-Gordon equation also appears to be particularly robust to noise, which suggests that the added complexity of having multiple spatial dimensions is not in general a barrier to identification.

For the nonlinear Schrödinger and reaction-diffusion systems, we test the ability of WSINDy to select the correct monomial nonlinearities from an excessively large model library. Using a library of 190 terms for nonlinear Schrödinger's and 181 terms for reaction-diffusion (see the dimensions of $\widetilde{\mathbf{G}}$ in Table 3.5.3), we demonstrate successful identification of the correct nonzero terms. Moreover, for the reaction-diffusion system, misidentified terms directly reflect the existence of a limit cycle[23]. Finally, the vortex-shedding limit cycle for the 2D Navier-Stokes equations is used primarily to compare to previous results in [117], and we find that at large-noise WSINDy conveniently selects the Euler equations.

[22] We have not included experiments involving multiple-soliton solutions to Sine-Gordon, however the success of WSINDy applied to KdV, nonlinear Schrödinger and Sine-Gordon suggests that the class of integrable systems could be a fruitful avenue for future research.

[23] We note that discovery of the same reaction-diffusion system from a much smaller library of terms is shown in [117, 114], but with different initial conditions that result in a spiral wave limit cycle. Our choice of initial conditions is motivated below in Appendix 3.B.

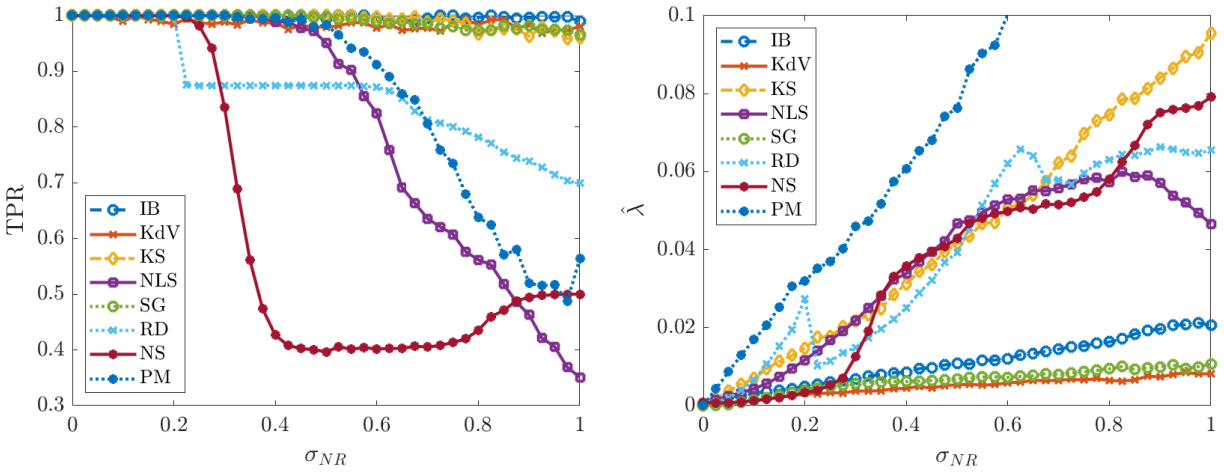


Figure 3.5.2: Left: average TPR (true positivity ratio, defined in (3.5.3)) for each of the PDEs in Table 3.5.1 computed from 200 instantiations of noise for each noise level σ_{NR}. Right: average learned threshold $\widehat{\lambda}$ (defined in (3.4.7)). For the porous medium equation (PM), $\widehat{\lambda}$ increases to 0.2 as σ_{NR} approaches 1 (we omit this from the plot in order to make visible the $\widehat{\lambda}$ trends for the other systems).

3.5.4 Results: Model Identification

Performance regarding the identification of correct nonzero terms in each model is reported in Figures 3.5.2 and 3.5.3, which include plots of the average TPR, the learned threshold $\widehat{\lambda}$, and the loss function $\mathcal{L}(\lambda)$ (defined in (3.5.3), (3.4.7), and (3.4.4), respectively). As we will discuss, significant decreases in average TPR are often accompanied by transitions in the identified $\widehat{\lambda}$.

Figure 3.5.2 (left) shows that for inviscid Burgers, Korteweg-de Vries, Kuramoto-Sivashinsky and Sine-Gordon, the average TPR stays above 0.95 even for noise levels as high as 100% (i.e. WSINDy reliably identifies these models in the presence of noise that has the same L^2-norm as the underlying clean data). The average TPR for the nonlinear Schrödinger and porous medium equations stays above 0.95 until 50% noise, after which identification of the correct monomial nonlinearity is not as reliable. For NLS, this is a drastic improvement over previous studies [117], especially considering the large library of 190 terms used.

We observe in Figure 3.5.2 (right) that the learned threshold $\widehat{\lambda}$ increases with σ_{NR}, suggesting that automatic selection of $\widehat{\lambda}$ in the learning algorithm (3.4.7) is crucial to the algorithm's robustness to noise. For example, the Kuramoto-Sivashinsky equation has a minimum nonzero coefficient of 0.5 (multiplying $\partial_x(u^2)$), and we find that $\widehat{\lambda}$ approaches 0.1 as σ_{NR} approaches 1, which implies that at higher noise levels the range of $\widehat{\lambda}$ values that is necessary[24] for correct model identification is approximately (~ 0.1, ~ 0.5). Since it is highly unlikely that this range of admissible values would

[24]By definition (3.4.7), $\widehat{\lambda}$ is the minimum value in $\boldsymbol{\lambda}$ that minimizes the loss $\mathcal{L}$ (3.4.7), hence values in $\boldsymbol{\lambda}$ below $\widehat{\lambda}$ are precisely the thresholds that result in misidentification of the correct model by overfitting, while thresholds above $\min_{\{j\,:\,\mathbf{w}^\star_j\neq 0\}}|\mathbf{w}^\star_j|$ necessarily underfit the model.

be known *a priori*, the chances of manually selecting a feasible $\widehat{\lambda}$ for Kuramoto-Sivashinsky are prohibitively low in the large noise regime (see Figure 3.5.3a for visualizations of the loss $\mathcal{L}$ applied to KS data). This effect is even greater for the porous medium equation. Automatic selection of $\widehat{\lambda}$ thus removes this sensitivity. In contrast, $\widehat{\lambda}$ is largely unaffected by increases in σ_{NR} for Burgers, Korteweg-de Vries and Sine-Gordon. In particular, Figure 3.5.3b shows little qualitative changes in the loss landscape for Sine-Gordon in the range $0.1 \leq \sigma_{NR} \leq 0.4$.

Intriguingly, for reaction-diffusion, the average TPR falls below 0.95 at 22% noise, after which WSINDy falsely identifies linear terms in u and v. If the true model is given by the compact form $\partial_t \mathbf{u} = \mathcal{A}(\mathbf{u})$ for $\mathbf{u} = (u, v)^T$, then the misidentified model in all trials for noise levels in the range $0.25 \leq \sigma_{NR} \leq 0.55$ is given by

$$\partial_t \mathbf{u} = \beta \mathcal{A}(\mathbf{u}) + \alpha \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{u} \tag{3.5.8}$$

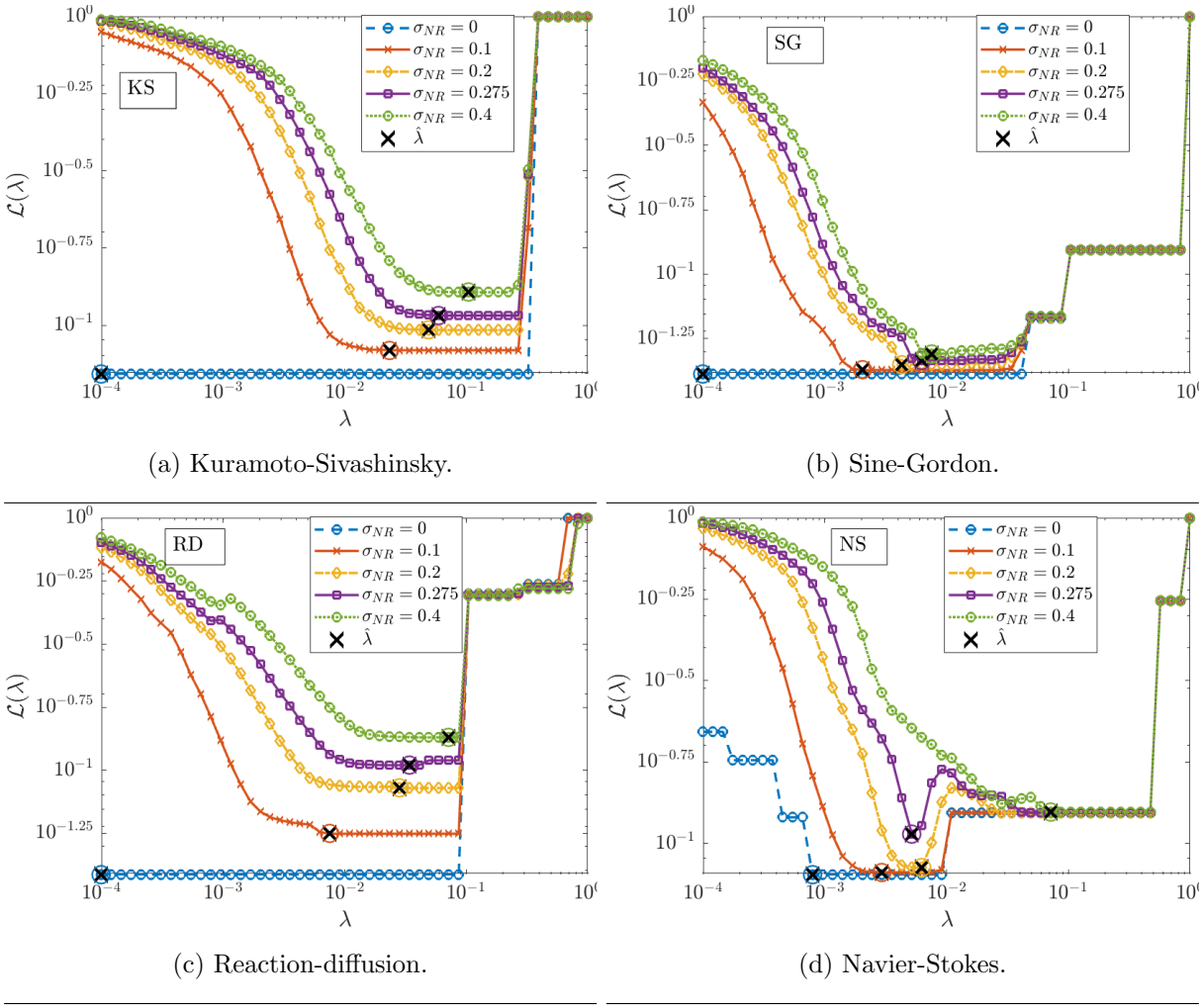
for some $\alpha > 0$ and $\beta \approx 1$ dependent on σ_{NR}. This is explainable by the fact that the underlying solution settles into a limit cycle, which means that at every point in space the solution oscillates. Indeed, the falsely identified nonzero terms in (3.5.8) exactly convey that at each point in space the solution is oscillating at a uniform frequency (albeit with variable amplitude and phase determined by the initial conditions[25]). Hence, in the presence of certain lower-dimensional structures (in this case a limit cycle), higher noise levels result in a mixture of the true model with a spatially-averaged reduced model. This shift between detection of the correct model and the oscillatory version (3.5.8) is also detectable in the learned threshold $\widehat{\lambda}$, which decreases at $\sigma_{NR} = 0.22$ (see RD data in Figure 3.5.2 (right)), and in the loss function $\mathcal{L}$ (Figure 3.5.3c). At $\sigma_{NR} = 0.275$ we see that $\mathcal{L}$ in Figure 3.5.3c is minimized for λ in the approximate range (~ 0.02, ~ 0.05) but also has a near-minimum for $\lambda \in (\sim 0.05, \sim 0.1)$. These two regions correspond to discovery of the oscillatory model (3.5.8) and the true model, respectively, but since the true model has a slightly higher loss at $\sigma_{NR} = 0.275$, model (3.5.8) is selected. For $\sigma_{NR} \geq 0.4$ there is no longer (on average) a region of λ that results in discovery of the true model, and WINSDy returns (3.5.8) to compensate for noise.

For Navier-Stokes we see an averaging effect at higher noise, similar to the reaction-diffusion system. TPR drops below 0.95 for noise levels above 27% with the resulting misidentified model being simply Euler's equations in vorticity form:

$$\partial_t \omega = -\partial_x(\omega u) - \partial_y(\omega v).$$

This is due primarily to the small viscosity $\nu = 0.01$ which prevents identification of the viscous forces at higher noise levels. Examining the loss function $\mathcal{L}$, Figure 3.5.3d shows that above $\sigma_{NR} \approx 0.275$, minimizers of $\mathcal{L}$ are above 0.01, hence the viscous terms will be thresholded out. Another possible explanation is the low-accuracy simulation used for the clean dataset: in the noise-free

[25] This is discussed further in Appendix 3.B.7.



(a) Kuramoto-Sivashinsky.

(b) Sine-Gordon.

(c) Reaction-diffusion.

(d) Navier-Stokes.

Figure 3.5.3: Plots of the average loss function $\mathcal{L}(\lambda)$ and resulting optimal threshold $\widehat{\lambda}$ for the Kuramoto Sivashinsky, Sine-Gordon, Reaction diffusion and Navier-Stokes equations.

setting, Table 3.5.4 shows that WSINDy recovers the model coefficients of Navier-Stokes to less than 3 significant digits in the absence of noise, which is the same level of accuracy exhibited on each of the other systems under 5% noise (see Figure 3.5.4). Nevertheless, with reliable recovery up to 27% noise, WSINDy makes notable improvements on previous results ([117]). Moreover, recovery of the Euler equations at high noise is desirable as this can be seen as the correct leader-order model.

3.5.5 Results: Coefficient Accuracy

Accuracy in the recovered coefficients is measured by E_∞ and E_2 (defined in (3.5.4) and (3.5.5), respectively) and shown in Table 3.5.4 for $\sigma_{NR} = 0$ and in Figure 3.5.4 for $\sigma_{NR} > 0$. As in the ODE case, the coefficient error E_∞ for smooth, noise-free data is determined by the order of

	IB	KdV	KS	NLS	PM	SG	RD	NS
E_∞	4.3×10^{-5}	3.1×10^{-7}	8.1×10^{-7}	9.4×10^{-8}	2.2×10^{-6}	4.3×10^{-5}	3.9×10^{-10}	1.1×10^{-3}

Table 3.5.4: Accuracy of WSINDy applied to noise-free data ($\sigma_{NR} = 0$).

	KdV	KS	NLS	RD	NS
WSINDy	6.7×10^{-4}	1.8×10^{-4}	2.9×10^{-4}	6.0×10^{-4}	1.2×10^{-3}
PDE-FIND	7.0×10^{-2}	0.52	3.0×10^{-2}	3.8×10^{-2}	7.0×10^{-2}

Table 3.5.5: Accuracy comparison between WSINDy and PDE-FIND with $\sigma_{NR} = 0.01$ (results for PDE-FIND reproduced from [117]).

accuracy of the numerical simulation method[26], since the error resulting from the trapezoidal rule is of lower order for the values (p_x, p_t) used in Table 3.5.3 (see [100], Lemma 1). Table 3.5.4 also shows that the algorithm returns reasonable accuracy for non-smooth data, with $E_\infty = 4.3 \times 10^{-5}$ and $E_\infty = 2.2 \times 10^{-6}$ for the inviscid Burgers and porous medium equations, respectively. For reference, Table 3.5.5 shows that WSINDy improves over PDE-FIND by about two digits[27].

For $\sigma_{NR} > 0$, in Figure 3.5.4 it is apparent that E_∞ scales approximately as a power law $E_\infty \sim \sigma_{NR}^r$ for some r approximately in the range $(\sim 1, \sim 2)$ in all systems except Navier-Stokes. It was observed in [57] that E_∞ will approximately scale linearly with σ_{NR} for Kuramoto-Sivashinsky, however our results show that in general, for larger σ_{NR}, the rate will be superlinear and dependent on the reference test function and the nonlinearities present. A simple explanation for this in the case of normally-distributed noise is the following: linear terms $\Psi^s * \mathbf{U}$ will be normally-distributed with mean $\Psi^s * \mathbf{U}^\star$ and approximate variance $\Delta x^D \Delta t \left\| D^{\alpha^s} \psi \right\|_2^2 \sigma^2$, hence are *unbiased*[28] and lead to perturbations that scale linearly with σ_{NR}. On the other hand, general monomial nonlinearities[29] $\Psi^s * \mathbf{U}^j$ with $j > 1$ are *biased* and have approximate variance $\Delta x^D \Delta t \left\| D^{\alpha^s} \psi \right\|_2^2 p_{2j}(\sigma)$ for p_{2j} a polynomial of degree $2j$. Hence, nonlinear terms $\Psi^s * f_j(\mathbf{U})$ lead to biased columns of the Gram matrix $\mathbf{G}$ with variance scaling with σ^{2r} for some $r > 1$ and proportional to $\left\| D^{\alpha^s} \psi \right\|_2$. Thus, for larger noise and higher-degree monomial nonlinearities, we expect superlinear growth of the error, as observed in particular with nonlinear Schrödinger's, Sine-Gordon, and reaction-diffusion. Nevertheless, Figure 3.5.4 suggests that a conservative estimate on the coefficient error is $E_\infty \leq \frac{\sigma_{NR}}{10}$, indicating $1 - \log_{10}(\sigma_{NR})$ significant digits (e.g. for $\sigma_{NR} = 0.1$ we have $E_\infty \leq 10^{-2}$ for each system except KdV, indicating two significant digits), which is consistent with the ODE case [100].

For Burgers and Korteweg-De Vries, the average error E_2 at higher noise levels is affected by outliers containing a falsely-identified advection term $\partial_x u$. This is due to the large amplitude

[26] For example, Sine-Gordon and Navier-Stokes are both integrated in time using second-order methods, hence have lower accuracy than the other examples (see Appendix 3.B for more details).

[27] Results shown for $\sigma_{NR} = 0.01$ reproduced from [117] (note that PDE-FIND is unreliable at higher noise levels).

[28] In other words, equal to the noise-free case in expectation (recall that $\mathbf{U}^\star$ is the underlying noise-free data).

[29] With the exception of $j = 2$ and odd $|\alpha^s|$, due to the fact that $\mathbb{E}[\Psi^s * \epsilon^2] \approx \mathbb{E}[\epsilon^2] \int_{\Omega_R} D^{\alpha^s} \psi \, dx dt = 0$.

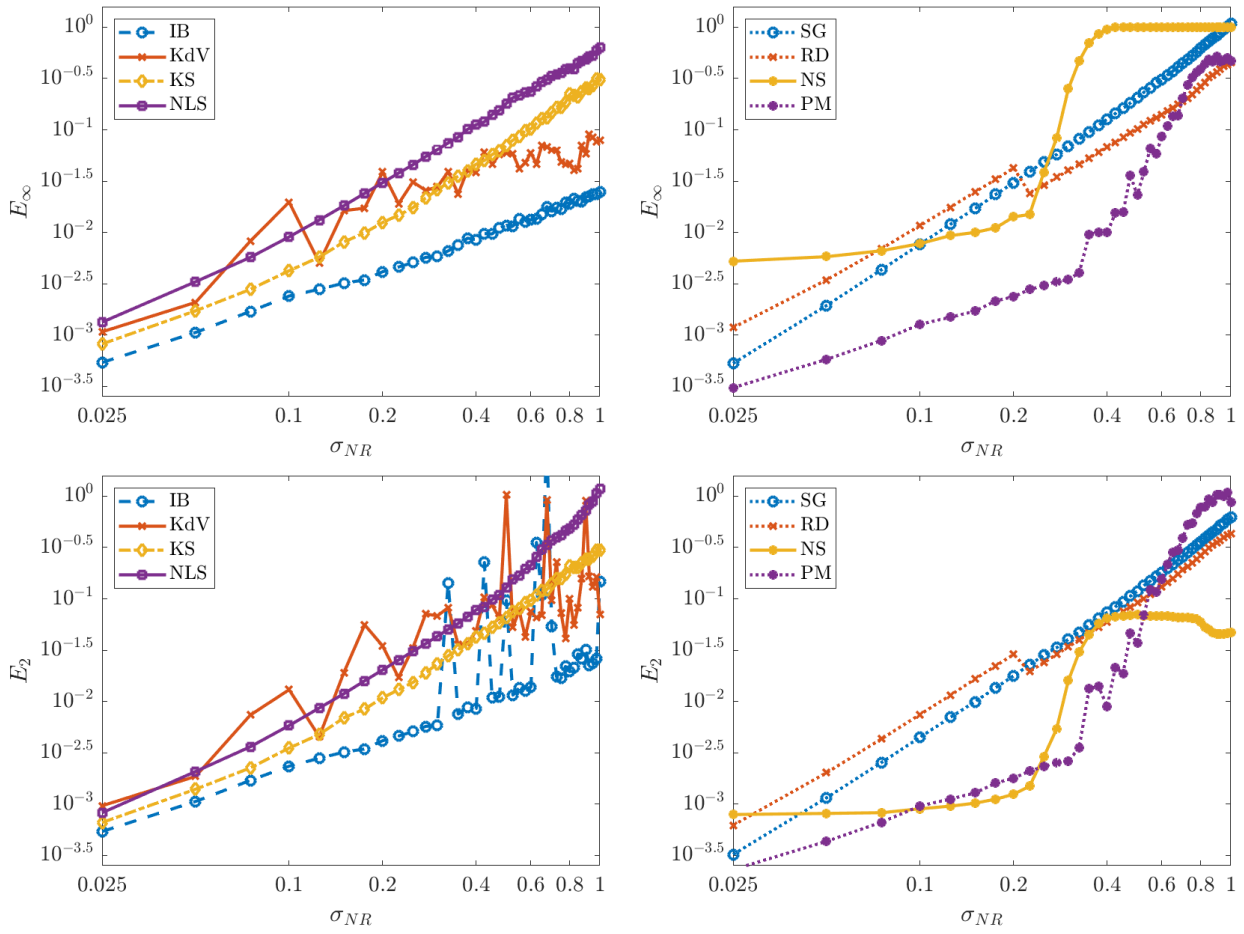


Figure 3.5.4: Coefficient errors E_∞ and E_2 (equations (3.5.4) and (3.5.5)) for each of the seven models Table 3.5.1. Models in one and two spatial dimensions are shown on the left and right, respectively.

---

datasets used, which lead to the closest pure-advection model for each system being given by[30]

$$\text{(Burgers)}\ \ \partial_t u = -(498)\partial_x u, \qquad \text{(KdV)}\ \ \partial_t u = -(512)\partial_x u.$$

Hence, a falsely identified $\partial_x u$ term will have a large coefficient compared to the true model coefficients which have magnitudes 0.5 or 1. In all other cases, the values of E_2 and E_∞ are comparable, which implies that misidentified terms do not have large coefficients and might be removed with a larger threshold. Lastly, the sigmoidal shape of E_∞ and E_2 for Navier-Stokes is due again to the unidentified diffusive terms at larger noise. It is interesting to note that for $\sigma_{NR} \leq 0.27$ the coefficient error for Navier-Stokes is relatively constant, in contrast to the other systems, and does not exhibit a power-law. However, at present, we do not have a concrete explanation for this behavior.

[30]This is found by projecting the left-hand side $\mathbf{b}$ onto the column $\partial_x \psi * \mathbf{U}^\star$ (i.e. in the noise-free case).

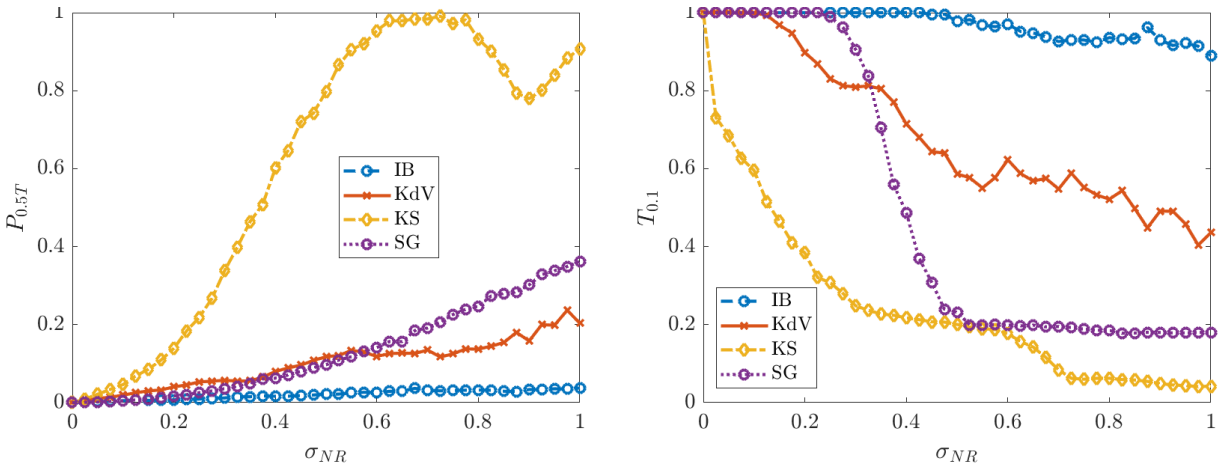


Figure 3.5.5: Prediction accuracy measured by $P_{0.5T}(\widehat{\mathbf{w}})$ and $T_{0.1}(\widehat{\mathbf{w}})$ (defined in (3.5.6) and (3.5.7), respectively).

3.5.6 Results: Prediction Accuracy

Lastly, Figure 3.5.5 shows the prediction accuracy on a subset of the systems in Table 3.5.1 as measured by $P_{0.5T}(\widehat{\mathbf{w}})$ and $T_{0.1}(\widehat{\mathbf{w}})$ (defined in (3.5.6) and (3.5.7), respectively). We report that data-driven solutions attain greater than 90% accuracy in the L_2 sense up to time $0.8T$ (80% of the trajectory) for noise levels as high as 40%. (This excludes the KS equation, which exhibits spatiotemporal chaos and cannot be expected to remain close to the noise-free data.) Data-driven solutions to the KS equation, while eventually divergent, also attain 90% accuracy up to time $0.5T$ for noise levels below 15%. Lastly, we note that for lower noise levels (up to 10%), the accuracy of data-driven solutions to the inviscid Burgers, Korteweg-de Vries and Sine-Gordon equations is on average above 96% along the entire trajectory (not shown in the figures).

3.6 Conclusion

We have extended the WSINDy algorithm to the setting of PDEs for the purpose of discovering models for spatiotemporal dynamics without relying on pointwise derivative approximations, black-box closure models (e.g. deep neural networks), dimensionality reduction, or other noise filtering. We have provided methods for learning many of the algorithm's hyperparameters directly from the given dataset, and in the case of the threshold $\widehat{\lambda}$, demonstrated the necessity of avoiding manual hyperparameter tuning. The underlying convolutional weak form (3.3.4) allows for efficient implementation using the FFT. This naturally leads to a selection criterion for admissable test functions based on spectral decay, which is implemented in the examples above. In addition, we have shown that by utilizing scale invariance of the PDE together with a modified sparsification measure, models may be recovered from large candidate model libraries and from data that is poorly

scaled. When unsuccessful, WSINDy appears to discover a nearby sparse model that captures the dominant spatiotemporal behavior (see the discussions surrounding misidentification of the reaction-diffusion and Navier-Stokes equations in Section 3.5.4).

We close with a summary of possible future directions. In Section 3.4.1 we discussed the significance of decay properties of test functions in real and in Fourier space, as well as general test function regularity. We do not make any claim that the class $\mathcal{S}$ defined by (3.4.2) is optimal, but it does appear to work very well, as demonstrated above (as well as in the ODE setting [100]) and also observed in [114, 57]. A valuable tool for future development of weak identification schemes would be the *identification of optimal test functions*. A preliminary step in this direction is our use of the changepoint method described in Appendix 3.A.

In the ODE setting, adaptive placement of test functions provided increased robustness to noise. Convolution query points can similary be strategically placed near regions of the dynamics with high information content, which may be crucial for model selection in higher dimensions. Defining regions of high information content and *adaptively placing query points* accordingly would allow for identification from smaller datasets.

Ordinary least squares makes the assumption of i.i.d. residuals and should be replaced with generalized least squares to accurately reflect the true error structure. The current framework could be vastly improved by incorporating more precise statistical information about the linear system $(\mathbf{G}, \mathbf{b})$. The first step in this direction is the derivation of an *approximate covariance matrix* as in WSINDy for ODEs [100]. Previous results on generalized sensitivity analysis for PDEs may provide improvements in this direction [73, 134].

Accuracy in the recovered coefficients is still not entirely understood and is needed to derive recovery guarantees. It is claimed in [57] that at higher noise levels the scaling will approximately be linear in σ_{NR}, while we have demonstrated that this is not the case in general: the scaling depends on the nonlinearities present in the true model, the decay properties of the test functions, and accuracy of the underlying clean data. Analysis of *coefficient error dependence* (on noise, amplitudes, number of datapoints, etc.) could occur in tandem with development of a generalized least-squares framework.

The examples above show that WSINDy is very robust to noise for problems involving nonlinear waves (Burgers, Korteweg de-Vries, nonlinear Schrödinger, Sine-Gordon), spatiotemporal chaos (Kuramoto-Sivashinsky), and even nonlinear diffusion (porous medium), but is less robust for data with limit cycles (reaction-diffusion, Navier-Stokes). Further, identification of Burgers, Korteweg de-Vries, and Sine-Gordon appears robust to changes in the sparsity threshold $\widehat{\lambda}$ (see Figure 3.5.2 (right)). A *structural identifiability criteria* for measuring uncertainty in the recovery process based on identified structures (transport processes, mixing, spreading, limit cycles, etc.) would also be invaluable for general model selection.

3.7 Acknowledgements

This research was supported in part by the NSF/NIH Joint DMS/NIGMS Mathematical Biology Initiative grant R01GM126559 and in part by the NSF Computing and Communications Foundations Division grant CCF-1815983. This work also utilized resources from the University of Colorado Boulder Research Computing Group, which is supported by the National Science Foundation (awards ACI-1532235 and ACI-1532236), the University of Colorado Boulder, and Colorado State University. Code used in this chapter is publicly available on GitHub at https://github.com/MathBioCU/WSINDy_PDE. The authors would like to thank Prof. Vanja Dukic (University of Colorado at Boulder, Department of Applied Mathematics), Kadierdan Kaheman (University of Washington, Department of Applied Mathematics), Samuel Rudy (Massachusetts, Department of Mechanical Engineering), and Zofia Stanley (University of Colorado at Boulder, Department of Applied Mathematics), for helpful discussions.

Appendix

3.A Learning Test Functions From Data

We present the following algorithm for automatic selection of test functions which utilizes the implicit smoothing of high-frequency noise afforded by the convolution. This approach is useful in practice but we leave rigorous justification of it to future work. We proceed in two steps: (1) estimation of critical wavenumbers $(k_1^*, \ldots, k_{D+1}^*)$ separating noise- and signal-dominated modes in each coordinate and (2) enforcing decay in real and in Fourier space.

We will describe the process for detecting $k_x^* = k_1^*$ from data $\mathbf{U} \in \mathbb{R}^{N_1 \times N_2}$ given over the one-dimensional spatial grid $\mathbf{x} \in \mathbb{R}^{N_1}$ at timepoints $\mathbf{t} \in \mathbb{R}^{N_2}$. Figures 3.A.1-3.A.2 illustrate this approach using Kuramoto-Sivashinsky data with 50% noise. Below, $\mathcal{F}^x$ and $\mathcal{F}^t$ denote the discete Fourier transform (DFT) along the x and t coordinates, respectively, while $\mathcal{F}$ denotes the full two-dimensional DFT.

3.A.1 Detection of Critical Wavenumbers

Assume the data has additive white noise $\mathbf{U} = \mathbf{U}^\star + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and that $\mathcal{F}(\mathbf{U}^\star)$ decays. The power spectrum of the noise $|\mathcal{F}^x(\epsilon)|$ is then i.i.d, hence as discussed in Section 3.4.1, there will be a critical wavenumber k_x^* in the power spectrum of the data $\mathcal{F}^x(\mathbf{U})$ after which the modes become noise-dominated. To detect k_x^*, we collapse $|\mathcal{F}^x(\mathbf{U})|$ into a one-dimensional array by averaging in time and then take the cumulative sum in x:

$$\mathbf{H}_k^x := \sum_{j=-N_1/2}^{k} \overline{|\mathcal{F}_j^x(\mathbf{U})|} \tag{3.A.1}$$

where $\overline{|\mathcal{F}_j^x(\mathbf{U})|}$ is the time-average of the jth mode of the discrete Fourier transform along the x-coordinate. Since $|\mathcal{F}^x(\epsilon)|$ is i.i.d., $\mathbf{H}^x$ will be approximately linear over the noise-dominated modes, which is an optimal setting for locating k_x^* as a *changepoint*, or in other words the corner point of the best piecewise-linear approximation[31] to $\mathbf{H}^x$ using two pieces (see Figure 3.A.1). An algorithm for this is given in [76] and implemented in MATLAB using the function `findchangepts`.

[31] In the weighted least-squares sense with weights $\omega_k = |\mathbf{H}_k^x|^{-1}$.

3.A.2 Enforcing Decay

Having detected the changepoints k_x^* and k_t^*, we compute hyperparameters for the coordinate test functions ϕ_x and ϕ_t using user-specified hyperparameters τ and $\widehat{\tau}$. As in Section 3.4.1, τ specifies the rate of decay of ϕ_x and ϕ_t in real space through equation (3.4.3). The hyperparameter $\widehat{\tau}$ is introduced to specify the rate of decay of ϕ_x and ϕ_t in Fourier space. Specifically, for a chosen $\widehat{\tau}$ we enforce that the changepoints k_x^* and k_t^* fall approximately $\widehat{\tau}$ standard deviations into the tail of the spectra $\widehat{\phi}_x$ and $\widehat{\phi}_t$. This is done by utilizing that ϕ_x and ϕ_t are functions of the form

$$\phi_{a,p}(s) := C\left(1 - \left(\frac{s}{a}\right)^2\right)_+^p,$$

(i.e. centered, symmetric functions in the class $\mathcal{S}$ defined in (3.4.2)) which are well-approximated by Gaussians for large enough p and appropriate scaling C. Indeed, letting C be such that $\|\phi_{a,p}\|_1 = 1$ and setting $\sigma := a/\sqrt{2p+3}$, we have that $\phi_{a,p}$ matches the first three moments of the Gaussian

$$\rho_\sigma(s) := \frac{1}{\sqrt{2\pi\sigma^2}} e^{-s^2/2\sigma^2},$$

which provides a bound on the error in the Fourier transforms $\widehat{\phi}_{a,p}$ and $\widehat{\rho}_\sigma$ for small frequencies ξ in terms of their 4th moments[32]:

$$|\widehat{\phi}_{a,p}(\xi) - \widehat{\rho}_\sigma(\xi)| \le |\xi|^4 \left(\frac{a^4}{2}\left[\frac{p+3/2}{(4p^2+12p+9)(4p^2+16p+15)}\right] + o(1)\right) = \mathcal{O}(|\xi|^4 a^4 p^{-3}).$$

This implies that for small ξ and a and large p, it suffices to use $\widehat{\rho}_\sigma(\xi) = \rho_{1/\sigma}(\xi)$ to estimate $\widehat{\phi}_{a,p}$. Hence, we enforce decay of $\widehat{\phi}_x$ (and similarly for $\widehat{\phi}_t$) by choosing m_x and p_x such that

$$\frac{2\pi}{N_1 \Delta x} k_x^* = \frac{\widehat{\tau}}{\sigma} = \widehat{\tau}\frac{\sqrt{2p_x+3}}{m_x \Delta x}$$

$$\implies \quad 2\pi k_x^* m_x = \widehat{\tau} N_1 \sqrt{2p_x+3}. \tag{3.A.2}$$

so that k_x^* is $\widehat{\tau}$ standard deviations into the tail of $\widehat{\rho}_\sigma(\xi)$, where $\sigma = m_x \Delta x/\sqrt{2p_x+3}$. To solve (3.4.3) and (3.A.2) simultaneously, we compute m_x as a root of

$$F(m) := F(m;\ k_x, N_1, \widehat{\tau}, \tau) := \log\left(\frac{2m-1}{m^2}\right)\left(4\pi^2 {k_x^*}^2 m^2 - 3N_1^2\widehat{\tau}^2\right) - 2N_1^2\widehat{\tau}^2\log(\tau).$$

$F(m)$ has a unique root $m_x \ge 2$ in the nonempty interval

$$\left[\frac{\sqrt{3}}{\pi}\left(\frac{N_1/2}{k_x^*}\right)\widehat{\tau},\quad \frac{\sqrt{3}}{\pi}\left(\frac{N_1/2}{k_x^*}\right)\widehat{\tau}\sqrt{1-(8/\sqrt{3})\log(\tau)}\right]$$

[32] This also shows that with $\sigma = a/\sqrt{2p+3}$, if we take $a = \sqrt{2p}$ then we get pointwise convergence $\phi_{a,p} \to \rho_1$ as $p \to \infty$.

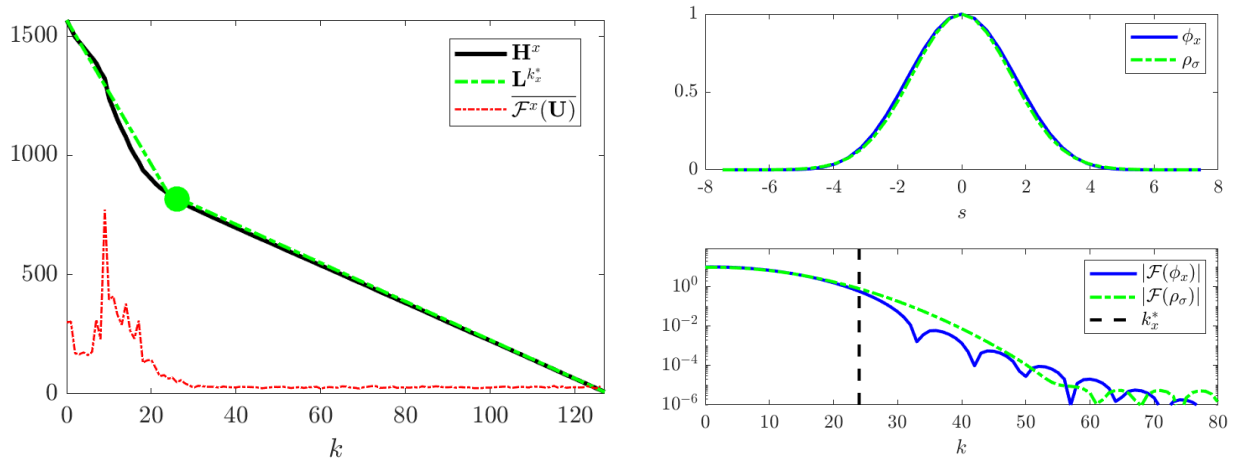


Figure 3.A.1: Visualization of the changepoint algorithm for KS data with 50% noise. Left: $\mathbf{H}^x$ (defined in (3.A.1)) and best two-piece approximation $\mathbf{L}^{k_x^*}$ along with resulting changepoint $k_x^* = 24$. The noise-dominated region of $\mathbf{H}^x$ ($k > 24$) is approximately linear as expected from the i.i.d. noise. (The time-averaged power spectrum $\overline{|\mathcal{F}^x(\mathbf{U})|}$ is overlaid and magnified for scale). Right: resulting test function ϕ_x and power spectrum $|\mathcal{F}(\phi_x)|$ along with reference Gaussian ρ_σ with $\sigma = m_x \Delta x / \sqrt{2p_x + 3}$. The power spectra $|\mathcal{F}(\phi_x)|$ and $|\mathcal{F}(\rho_\sigma)|$ are in agreement over the signal-dominated modes ($k \leq 24$). (Note that the power spectrum is symmetric about zero.)

---

on which F monotonically decreases and changes sign, provided $N_1 > 4$, $\tau \in (0, 1)$ and $\frac{\sqrt{3}}{\pi}\frac{\widehat{\tau}}{k_x^*} \in [4/N_1, 1]$, constraints which are easily satisfied. After finding m_x we can solve for p_x using either (3.4.3) or (3.A.2).

Figure 3.A.2 illustrates the implicit filtering of this process using the Burgers-type nonlinearity $\partial_x(\mathbf{U}^2)$ and the same KS dataset as in Figure 3.A.2 with 50% noise. The top panel compares a one-dimensional slice in x taken at fixed time $t = 99$ of the clean data $(\mathbf{U}^\star)^2$ and noisy data $(\mathbf{U})^2$. The middle panel shows the Fourier transforms of $(\mathbf{U}^\star)^2$ and $(\mathbf{U})^2$ along the given slice, showing that modes after k_x^* become noise-dominated. Finally, the bottom panel shows that after convolution with $\widehat{\partial_x \psi}$, where m_x and k_x are chosen with $\tau = 10^{-10}$ and $\widehat{\tau} = 2$, the clean and noisy spectra agree well, indicating successful filtering of noise-dominated modes (note that $(\mathbf{U})^2$ is highly-corrupted, nonlinearly-transformed, and biased from the noise-free term $(\mathbf{U}^\star)^2$, making this agreement in spectrum nontrivial).

3.B Simulation Methods

We now review the numerical methods used to simulate noise-free datasets for each of the PDEs in Table 3.5.1 (note that dimensions of the datasets are given in Table 3.5.2). Resolutions in space and time were chosen to limit computational overhead while exemplifying the dominant features of the solution. With the exception of the Navier-Stokes equations, which was simulated using the immersed boundary projection method in C++ [133], all computations were performed

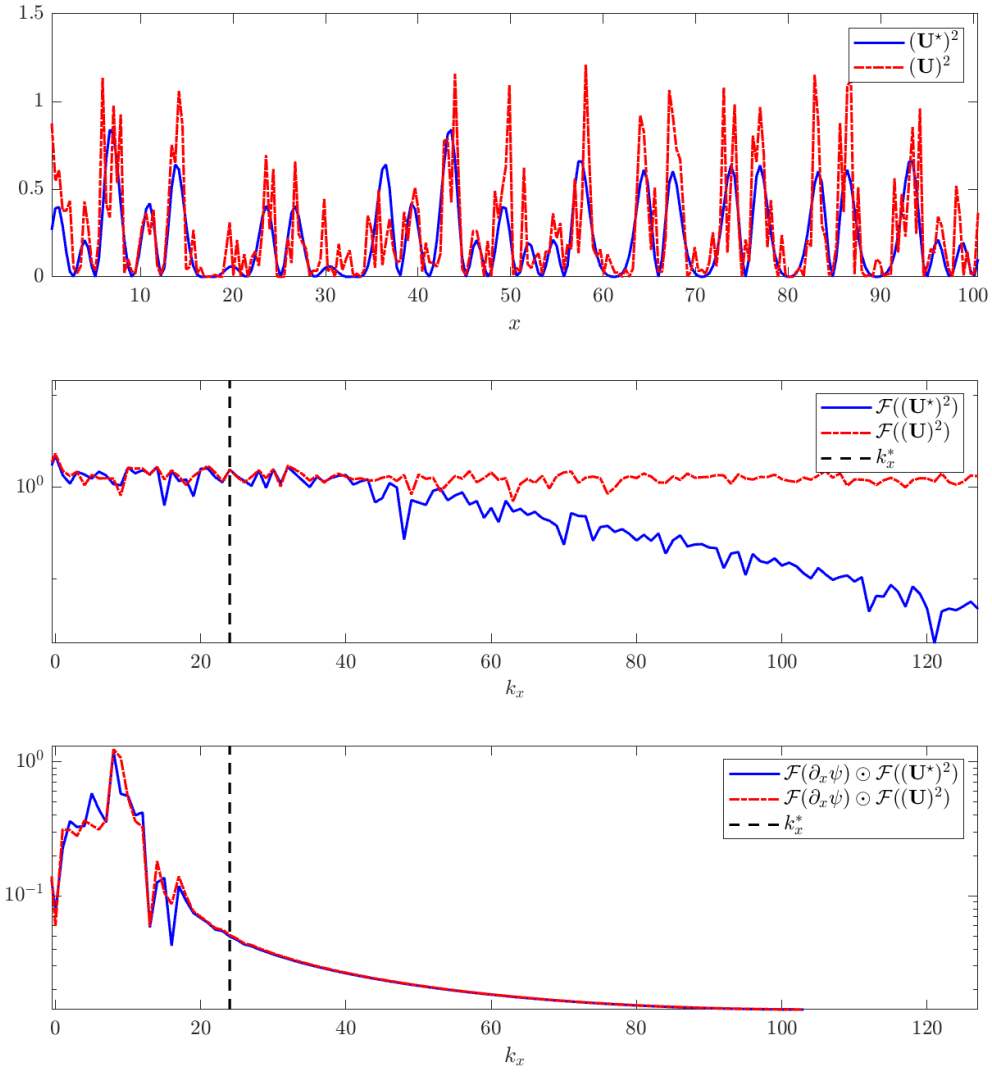


Figure 3.A.2: Illustration of the test function learning algorithm using computation of $\partial_x\psi * (\mathbf{U}^2)$ along a slice in x at fixed time $t = 99$ for the same dataset used in Figure 3.A.1. From top to bottom: (i) clean $\mathbf{U}^\star$ and noisy $\mathbf{U}$ variables, (ii) power spectra of the clean vs. noisy data along with the learned corner point k_x^*, (iii) power spectra of the element-wise products $\mathcal{F}(\partial_x\psi) \odot \mathcal{F}((\mathbf{U}^\star)^2)$ and $\mathcal{F}(\partial_x\psi) \odot \mathcal{F}((\mathbf{U})^2)$ (recall that these computations are embedded in the FFT-based convolution (3.3.12)).

in MATLAB 2019b. An interesting extension for future work would be to examine the dependence of WSINDy on the resolution, similar to the work in [106].

3.B.1 Inviscid Burgers

$$\partial_t u = -\frac{1}{2}\partial_x(u^2) \tag{3.B.1}$$

We take for exact data the shock-forming solution

$$u(x,t) = \begin{cases} A, & t \geq \max\left\{\frac{1}{A}x + \frac{1}{\alpha},\ \frac{2}{A}x + \frac{1}{\alpha}\right\} \\ -\frac{\alpha x}{1-\alpha t}, & A\left(t - \frac{1}{\alpha}\right) < x \leq 0 \\ 0, & \text{otherwise} \end{cases}. \tag{3.B.2}$$

which becomes discontinuous at $t = \alpha^{-1}$ with a shock travelling along $x = \frac{A}{2}\left(t - \frac{1}{\alpha}\right)$ (see Figure 3.5.1). We choose $\alpha = 0.5$ and an extreme value of $A = 1000$ to demonstrate that WSINDy still has excellent performance for large amplitude data. The noise-free data consists of (3.B.2) evaluated at the points $(x_i, t_j) = (-4000 + i\Delta x, j\Delta t)$ with $\Delta x = 31.25$ and $\Delta t = 0.0157$ for $1 \leq i, j \leq 256$.

3.B.2 Korteweg-de Vries

$$\partial_t u = -\frac{1}{2}\partial_x(u^2) - \partial_{xxx}u \tag{3.B.3}$$

A solution is obtained for $(x,t) \in [-\pi, \pi] \times [0, 0.006]$ with periodic boundary conditions using ETDRK4 timestepping and Fourier-spectral differentiation [72] with $N_1 = 400$ points in space and $N_2 = 2400$ points in time. We subsample 25% of the timepoints for system identification and keep all of the spatial points for a final resolution of $\Delta x = 0.0157$, $\Delta t = 10^{-5}$. For initial conditions we use the two-soliton solution

$$u(x,0) = 3A^2\text{sech}(0.5(A(x+2)))^2 + 3B^2\text{sech}(0.5(B(x+1)))^2, \qquad A = 25, B = 16.$$

3.B.3 Kuramoto-Sivashinsky

$$\partial_t u = -\frac{1}{2}\partial_x(u^2) - \partial_{xx}u - \partial_{xxxx}u. \tag{3.B.4}$$

A solution is obtained for $(x,t) \in [0, 32\pi] \times [0, 150]$ with periodic boundary conditions using ETDRK4 timestepping and Fourier-spectral differentiation [72] with $N_1 = 256$ points in space and $N_2 = 1500$ points in time. For system identification we subsample 20% of the time points for a

final resolution of $\Delta x = 0.393$ and $\Delta t = 0.5$. For initial conditions we use

$$u(x, 0) = \cos(x/16)(1 + \sin(x/16)).$$

3.B.4 Nonlinear Schrödinger

$$w_t = -\frac{i}{2}\partial_{xx} w + |w|^2 w \tag{3.B.5}$$

For the nonlinear Schrödinger equation (NLS) we reuse the same dataset from [117], containing $N_1 = 512$ points in space and $N_2 = 502$ timepoints, although we subsample 50% of the spatial points and 50% of the time points for a final resolution of $\Delta x = 0.039$, $\Delta t = 0.0125$. For system identification, we break the data into real and imaginary parts ($w = u + iv$) and recover the system

$$\begin{cases} \partial_t u = \frac{1}{2}\partial_{xx} v + u^2 v + v^3 \\ \partial_t v = -\frac{1}{2}\partial_{xx} u - u v^2 - u^3. \end{cases} \tag{3.B.6}$$

3.B.5 Anisotropic Porous Medium

$$\partial_t u = (0.3)\partial_{xx}(u^2) - (0.8)\partial_{xy}(u^2) + \partial_{yy}(u^2). \tag{3.B.7}$$

The equation can be rewritten

$$\partial_t u = \nabla \cdot \left(\mathbf{D}\nabla(u^2)\right)$$

for diffusivity tensor

$$\mathbf{D} = \begin{pmatrix} 0.3 & -0.4 \\ -0.4 & 1 \end{pmatrix}.$$

For noise-free data we use the analytical weak solution

$$u(\mathbf{x}, t) = \frac{1}{\sqrt{t}} \max\left(C - \frac{\mathbf{x}^T \mathbf{D}^{-1} \mathbf{x}}{16\sqrt{t}},\ 0\right)$$

where $\mathbf{x} = (x, y)^T$ and $C = \left(8\pi\sqrt{\det(\mathbf{D})}\right)^{-1/2}$ is chosen to enforce that $\int_{\mathbb{R}^2} u(\mathbf{x}, t)\, d\mathbf{x} = 1$ for all time. The solution has a finite jump in the gradient ∇u. For reference, this is the anisotropic version of the classical Barenblatt-Pattle solution to the (isotropic) porous medium equation [7, 110]. For the computation grid we use 200 points equally spaced from -5 to 5 in both x and y and 128 timepoints equally spaced from 0.5 to 2.5. The resolution is then $\Delta x = 0.05$ and $\Delta t = 0.0157$.

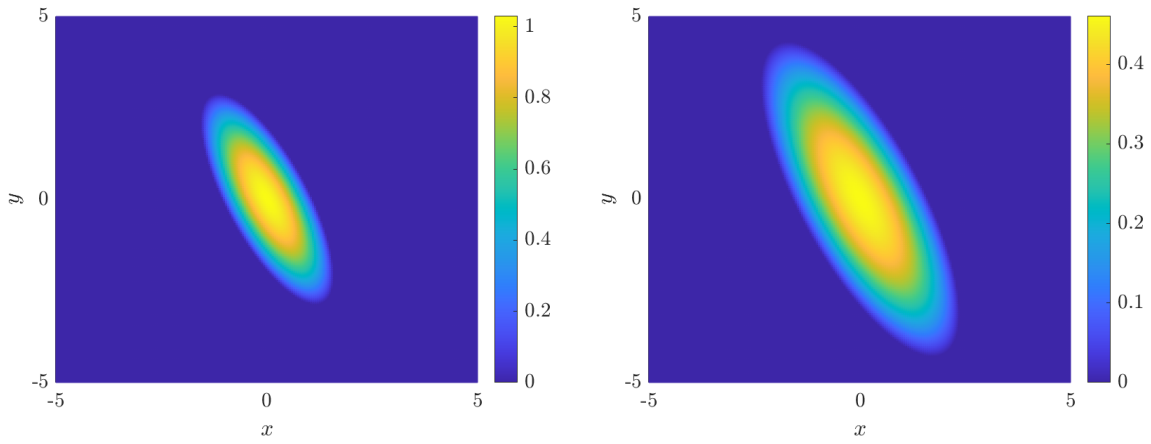


Figure 3.B.1: Noise-free data used for the anisotropic porous medium equation (3.B.7) at the initial time $t = 0.5$ (left) and final time $t = 2.5$ (right).

3.B.6 Sine-Gordon

$$\partial_{tt} u = \partial_{xx} u + \partial_{yy} u - \sin(u) \tag{3.B.8}$$

A numerical solution is obtained using a pseudospectral method on the spatial domain $[-\pi, \pi] \times [-1, 1]$ with 64 equally-spaced points in x and 64 Legendre nodes in y. Periodic boundary conditions are enforced in x and homogeneous Dirichlet boundaries in y. Geometrically, waves can be thought of as propagating on a right cylindrical sheet with clamped ends. Leapfrog time-stepping is used to generate the solution until $T = 5$ with $\Delta t = 6\text{e}{-5}$. We then subsample 0.25% of the timepoints and interpolate onto a uniform grid in space with $N_1 = 403$ points in x and $N_2 = 129$ points in y. The final resolution is $\Delta x = 0.0156$, $\Delta t = 0.025$. We arbitrarily use Gaussian data for the initial wave disturbance:

$$u(x, y, 0) = 2\pi \exp(-8(x - 0.5)^2 - 8y^2).$$

It is worth noting that when STLS is used instead of MSTLS (see Section 3.4.2) for sparsity enforcement, WSINDy returns a combination of $\sin(u)$ and terms from Taylor expansion of $\sin(u)$,

$$\alpha \left(u - \frac{1}{6} u^3 + \cdots \right) + (1 - \alpha) \sin(u). \tag{3.B.9}$$

MSTLS removes this problem. Furthermore, the test function selection method in Appendix 3.A is essential for allowing robust recovery of the Sine-Gordon equation as $\sigma_{NR} \to 1$ (see Figure 3.5.2).

3.B.7 Reaction-Diffusion

$$\begin{cases} \partial_t u = \frac{1}{10}\partial_{xx} u + \frac{1}{10}\partial_{yy} u - uv^2 - u^3 + v^3 + u^2 v + u \\ \partial_t v = \frac{1}{10}\partial_{xx} v + \frac{1}{10}\partial_{yy} v + v - uv^2 - u^3 - v^3 - u^2 v \end{cases} \tag{3.B.10}$$

The system (3.B.10) is simulated over a doubly-periodic domain $(x, y) \in [-10, 10] \times [-10, 10]$ with $t \in [0, 10]$ using Fourier-spectral differentiation in space and method-of-lines time integration via MATLAB's `ode45` with default tolerance. The computational domain has dimensions $N_1 = N_2 = 256$ and $N_3 = 201$, for a final resolution of $\Delta x = 0.078$, $\Delta t = 0.0498$. For initial conditions we use the spiral data

$$\begin{cases} u(x, y, 0) = \tanh(\sqrt{x^2 + y^2}) \cos\left(\theta(x + iy) - \pi\sqrt{x^2 + y^2}\right) \\ v(x, y, 0) = \tanh(\sqrt{x^2 + y^2}) \sin\left(\theta(x + iy) - \pi\sqrt{x^2 + y^2}\right), \end{cases}$$

where $\theta(z)$ is the principle angle of $z \in \mathbb{C}$. Note that this is an unstable spiral which breaks apart over time but still settles into a limit cycle.

Using the traditional (stable) spiral wave data [117] (differing only from the dataset used here in that the term $\pi\sqrt{x^2 + y^2}$ in the initial conditions above is replaced by $\sqrt{x^2 + y^2}$) we noticed an interesting behavior in that for high noise the resulting model is purely oscillatory. In other words, the stable spiral limit cycle happens to be well-approximated by the pure-oscillatory model

$$\partial_t \mathbf{u} = \alpha \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{u} \tag{3.B.11}$$

with $\alpha \approx 0.91496$. A comparison between this purely oscillatory reduced model and the full model simulated from the same initial conditions is shown in Figure 3.B.2. For $\sigma_{NR} \leq 0.1$ WSINDy applied to the stable spiral dataset returns the full model, while for $\sigma_{NR} > 0.1$ the oscillatory reduced model is more frequently detected. This suggests that although the stable spiral wave is a hallmark of the λ-ω reaction-diffusion system, from the perspective of data-driven model selection it is not an ideal candidate for identification of the full model.

3.B.8 Navier-Stokes

$$\partial_t \omega = -\partial_x(\omega u) - \partial_y(\omega u) + \frac{1}{100}\partial_{xx}\omega + \frac{1}{100}\partial_{yy}\omega \tag{3.B.12}$$

A solution is obtained on a spatial grid $(x, y) \subset [-1, 8] \times [-2, 2]$ with a "cylinder" of diameter 1 located at $(0, 0)$. The immersed boundary projection method [133] with 3rd-order Runge-Kutta timestepping is used to simulate the flow at spatial and temporal resolutions $\Delta x = \Delta t = 0.02$ for 2000 timesteps following the onset of the vortex shedding limit cycle. The dataset $(\mathbf{U}, \mathbf{V}, \mathbf{W})$

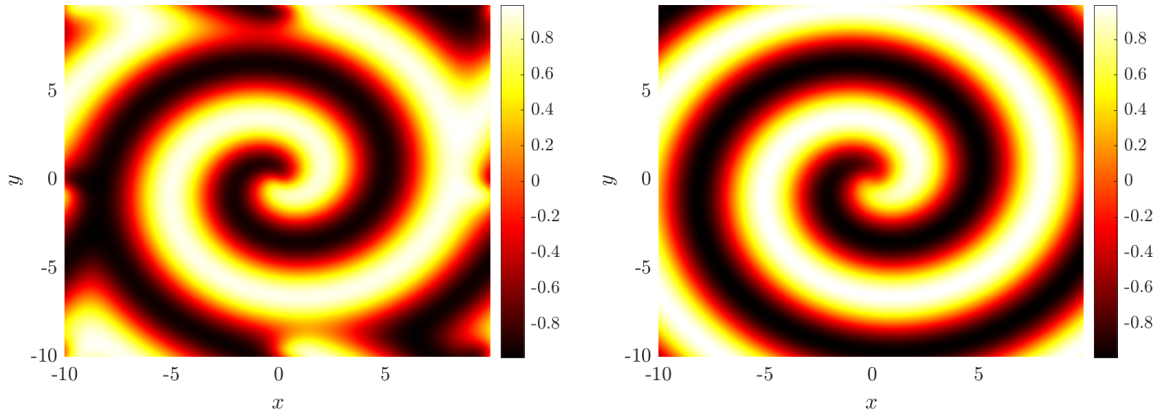


Figure 3.B.2: Comparison between the full reaction-diffusion model (3.B.10) (left) and the pure-oscillatory reduced model (3.B.11) (right) at the final time $T = 10$ with both models simulated from the same initial conditions leading to a spiral wave (only the v component is shown, results for u are similar). The reduced model provides a good approximation away from the boundaries.

---

contains the velocity components as well as the vorticity for points away from the cylinder and boundaries in the rectangle $(x, y) \in [1, 7.5] \times [-1.5, 1.5]$. We subsample 10% of the data in time for a final resolution of $\Delta x = 0.02$ and $\Delta t = 0.2$.

Chapter 4

WSINDy for IPS

Abstract

We develop a weak-form sparse identification method for interacting particle systems (IPS) with the primary goals of reducing computational complexity for large particle number N and offering robustness to either intrinsic or extrinsic noise. In particular, we use concepts from mean-field theory of IPS in combination with the weak-form sparse identification of nonlinear dynamics algorithm (WSINDy) to provide a fast and reliable system identification scheme for recovering the governing stochastic differential equations for an IPS when the number of particles per experiment N is on the order of several thousands and the number of experiments M is less than 100. This is in contrast to existing work showing that system identification for N less than 100 and M on the order of several thousand is feasible using strong-form methods. We prove that under some standard regularity assumptions the scheme converges with rate $\mathcal{O}(N^{-1/2})$ in the ordinary least squares setting and we demonstrate the convergence rate numerically on several systems in one and two spatial dimensions. Our examples include a canonical problem from homogenization theory (as a first step towards learning coarse-grained models), the dynamics of an attractive-repulsive swarm, and the IPS description of the parabolic-elliptic Keller-Segel model for chemotaxis. Code is available at https://github.com/MathBioCU/WSINDy_IPS.

4.1 Problem Statement

Recently there has been considerable interest in the methodology of data-driven discovery for governing equations. Building on the Sparse Identification of Nonlinear Dynamics (SINDy) [23], we developed a weak form version (WSINDy) for ODEs [100] and for PDEs [99]. In this work, we develop a formulation for discovering governing stochastic differential equations (SDEs) for interacting particle systems (IPS). To promote clarity and for reference later in the chapter, we first state the problem of interest. Subsequently, we will provide a discussion of background concepts and current results in the literature.

Consider a particle system $\mathbf{X}_t = (X_t^{(1)}, \dots, X_t^{(N)}) \in \mathbb{R}^{Nd}$ where on some fixed time window $t \in [0, T]$, each particle $X_t^{(i)} \in \mathbb{R}^d$ evolves according to the overdamped dynamics

$$dX_t^{(i)} = \left(-\nabla K * \mu_t^N \left(X_t^{(i)}\right) - \nabla V \left(X_t^{(i)}\right)\right) dt + \sigma(X_t^{(i)})\, dB_t^{(i)} \tag{4.1.1}$$

with initial data $X_0^{(i)}$ each drawn independently from some probability measure $\mu_0 \in \mathcal{P}_p(\mathbb{R}^d)$, where $\mathcal{P}_p(\mathbb{R}^d)$ is the space probability measures on $\mathbb{R}^d$ with finite pth moment[1]. Here, K is the *interaction potential* defining the pairwise forces between particles, V is the *local potential* containing all exogenous forces, σ is the *diffusivity*, and $\left(B_t^{(i)}\right)_{i=1,\dots,N}$ are independent Brownian motions each adapted to the same filtered probability space $(\Omega, \mathcal{B}, \mathbb{P}, (\mathcal{F}_t)_{t\geq 0})$. The *empirical measure* is defined

$$\mu_t^N := \frac{1}{N} \sum_{i=1}^N \delta_{X_t^{(i)}},$$

and the convolution $\nabla K * \mu_t^N$ is defined

$$\nabla K * \mu_t^N(x) = \nabla \int_{\mathbb{R}^d} K(x-y)\, d\mu_t^N(y) = \frac{1}{N} \sum_{i=1}^N \nabla K\left(x - X_t^{(i)}\right)$$

where we set $\nabla K(0) = 0$ whenever $\nabla K(0)$ is undefined. The recovery problem we wish to solve is the following.

(P) Let $\mathbb{X} = (\mathbf{X}_{\mathbf{t}}^{(1)}, \dots, \mathbf{X}_{\mathbf{t}}^{(M)})$ be discrete-time data at L timepoints $\mathbf{t} := (t_1, \dots, t_L)$ for M i.i.d. trials of the process (4.1.1) with $K = K^\star$, $V = V^\star$, and $\sigma = \sigma^\star$ and let $\mathbb{Y} = \mathbb{X} + \varepsilon$ be a corrupted dataset. For some fixed compact domain $\mathcal{D} \subset \mathbb{R}^d$ containing $\operatorname{supp}(\mathbb{Y})$, and finite-dimensional hypothesis spaces[2] $\mathcal{H}_K \subset L^2(\mathcal{D}-\mathcal{D})$, $\mathcal{H}_V \subset L^2(\mathcal{D})$, and $\mathcal{H}_\sigma \subset L^2(\mathcal{D})$, solve

$$\left(\widehat{K}, \widehat{V}, \widehat{\sigma}\right) = \underset{K\in\mathcal{H}_K, V\in\mathcal{H}_V, \sigma\in\mathcal{H}_\sigma}{\operatorname{argmin}} \|\nabla K - \nabla K^\star\|_{L^2(\mathcal{D}-\mathcal{D})} + \|\nabla V - \nabla V^\star\|_{L^2(\mathcal{D})} + \|\sigma - \sigma^\star\|_{L^2(\mathcal{D})}.$$

The problem **(P)** is clearly intractable because we do not have access to $K^\star$, $V^\star$, or $\sigma^\star$, and

[1] We define the pth moment of a probability measure μ for $p \geq 0$ by $M_p(\mu) := \int_{\mathbb{R}^d} |x|^p d\mu(x)$.

[2] The set $\mathcal{D} - \mathcal{D}$ is defined $\mathcal{D} - \mathcal{D} = \{x - y \ : \ (x, y) \in \mathcal{D} \times \mathcal{D}\}$.

moreover the interactions between these terms render simultaneous identification of them ill-posed. We consider two cases: (i) $\varepsilon \neq 0$ and $\sigma^\star = 0$, corresponding to purely *extrinsic noise*, and (ii) $\varepsilon = 0$ and $\sigma^\star \neq 0$, corresponding to purely *intrinsic noise*. The extrinsic noise case is important for many applications, such as cell tracking, where uncertainty is present in the position measurements. In this case we examine ε representing i.i.d. Gaussian noise with mean zero and variance[3] $\epsilon^2 \mathbf{I}_d$ added to each particle position in $\mathbf{X}$. In the case of purely intrinsic noise, identification of the diffusivity $\sigma^\star$ is required as well as the deterministic forces on each particle as defined by $K^\star$ and $V^\star$. A natural next step is to consider the case with both extrinsic and intrinsic noise. However, the combined noise case is sufficiently nuanced as to render it beyond the scope of this chapter, and we leave it for future work.

4.2 Background

Interacting particle systems (IPS) such as (4.1.1) are used to describe physical and artificial phenomena in a range of fields including astrophysics [146, 56], molecular dynamics [82], cellular biology [125, 138, 10], and opinion dynamics [14]. In many cases the number of particles N is large, with cell migration experiments often tracking 10^3-10^6 cells and simulations in physics (molecular dynamics, particle-in-cell, etc.) requiring N in the range 10^6-10^{12}. Inference of such systems from particle data thus requires efficient means of computing pairwise forces from $\mathcal{O}(N^2)$ interactions at each timestep for multiple candidate interaction potentials K. Frequently, so-called *mean-field* equations at the continuum level are sufficient to describe the evolution of the system, however in many cases (e.g. chemotaxis in biology [74]) only phenomenological mean-field equations are available. Moreover, it is often unclear how many particles N are needed for a mean-field description to suffice. Many disciplines are now developing machine learning techniques to extract coarse-grained dynamics from high-fidelity simulations (see [54] for a recent review in molecular dynamics). In this work we provide a means for inferring governing mean-field equations from particle data assumed to follow the dynamics (4.1.1) that is highly efficient for large N, and is effective in learning mean-field equations when N is in range 10^3-10^5.

Inference of the drift and diffusion terms for stochastic differential equations (SDEs) is by now a mature field, with the primary method being maximum-likelihood estimation, which uses Girsanov's theorem together with the Radon-Nykodym derivative to arrive at a log-likelihood function for regression. See [11, 87] for some early works and [12] for a textbook on this approach. More recently, sparse regression approaches using the Kramers-Moyal expansion have been developed [20, 25, 83] and the authors of [105] use sparse regression to learn population level ODEs from agent-based modeling simulations. The authors of [22] also derived a bias-correcting regression framework for inferring the drift and diffusion in underdamped Langevin dynamics, and in [33] a neural network-based algorithm for inferring SDEs was developed.

[3] By $\mathbf{I}_d$ we mean the identity in $\mathbb{R}^d$.

Only in the last few years have significant strides been made towards parameter inference of *interacting* particle systems such as (4.1.1) from data. Apart from some exceptions, such as a Gaussian process regression algorithm recently developed in [46], applications of maximum likelihood theory are by far the most frequently studied. An early but often overlooked work by Kasonga [71] extends the maximum-likelihood approach to inference of the interaction potential K, assuming full availability of the continuous particle trajectories and the diffusivity σ. Two decades later, Bishwal [13] further extended this approach to discrete particle observations in the specific context of linear particle interactions. In both cases, a sequence of finite-dimensional subspaces is used to approximate the interaction function, and convergence is shown as the dimension of the subspace J and number of particles N both approach infinity. More recently, the maximum likelihood approach has been carried out in [19, 90] in the case of radial interactions and in [31] in the case of linear particle interactions and single-trajectory data (i.e. one instance of the particle system). The authors of [127] recently developed an online maximum likelihood method for inference of IPS, and in [55] maximum likelihood is applied to parameter estimation in an IPS for pedestrian flow. It should also be noted that parameter estimation for IPS is common in biological sciences, with the most frequently used technique being nonlinear least squares with a cost function comprised of summary statistics [92, 125].

Problem **(P)** is made challenging by the coupled effects of K, V, and σ. In each of the previously mentioned algorithms, the assumption is made that σ is known and/or that K takes a specific form (radial or linear). In addition, the maximum likelihood-based approach approximates the differential $dX_t^{(i)}$ of particle i using a 1st-order finite difference: $dX_t^{(i)} \approx X_{t+\Delta t}^{(i)} - X_t^{(i)}$, which is especially ill-suited to problems involving *extrinsic* noise in the particle positions. Our primary goal is to show that the weak-form sparse regression framework allows for identification of the full model (K, V, σ), with significantly reduced computational complexity, when N is on the order of several thousands or more. We use a two-step process: the density of particles is approximated using a density kernel G and then the WSINDy algorithm (weak-form sparse identification of nonlinear dynamics) is applied in the PDE setting [100, 99]. WSINDy is a modified version of the original SINDy algorithm [23, 117] where the weak formulation of the dynamics is enforced using a family of test functions that offers reduced computational complexity, high-accuracy recovery in low-noise regimes, and increased robustness to high-noise scenarios. The feasibility of this approach for IPS is grounded in the convergence of IPS to associated mean-field equations. The reduction in computational complexity follows from the reduction in evaluation of candidate potentials (as discussed in Section 4.4.2), as well as the convolutional nature of the weak-form algorithm.

To the best of our knowledge, we present here the first *weak-form sparse regression* approach for inference of interacting particle systems, however we now review several related approaches that have recently been developed. In [131], the authors learn local hydrodynamic equations from active matter particle systems using the SINDy algorithm in the strong-form PDE setting. In contrast to [131], our approach learns nonlocal equations using the weak-form, however similarly to [131] we

perform model selection and inference of parameters using sparse regression at the continuum level. The weak form provides an advantage because no smoothness is required on the particle density (for requisite smoothness the authors of [131] use a Gaussian kernel, which is more expensive to compute than simple particle binning as done here). The authors of [3] developed an integral formulation for inference of plasma physics models from PIC data using SINDy, however their method involves first computing strong-form derivatives and then averaging, rather than integration by parts against test functions as done here, and as in [131], the learned models are local. In [81], the authors apply the maximum likelihood approach in the continuum setting on the underlying nonlocal Fokker-Planck equation and learn directly the nonlocal PDE using strong-form discretizations of the dynamics. While we similarly use the continuum setting for inference (albiet in weak form), our approach differs from [81] in that it is designed for the more realistic setting of discrete-time *particle* data, rather than pointwise data on the particle *density* (assumed to be smooth in [81]).

4.2.1 Contributions

The purpose of the present chapter is to show that the weak form provides an advantage in speed and accuracy compared with existing inference methods for particle systems when the number of particles is sufficiently large (on the order of several thousand or more). The key points of this chapter include:

(I) Formulation of a weak-form sparse recovery algorithm for simultaneous identification of the particle interaction force K, local potential V, and diffusivity σ from discrete-time particle data.

(II) Convergence with rate $\mathcal{O}(N^{-1/2})$ of the resulting full-rank least-squares solution as the number of particles $N \to \infty$ and timestep $\Delta t \to 0$.

(III) Numerical illustration of (II) along with robustness to either intrinsic randomness (e.g. Brownian motion) or extrinsic randomness (e.g. additive measurement noise).

4.2.2 Chapter Organization

In Section 4.3 we review results from mean-field theory used to show convergence of the weak-form method. In Section 4.4 we introduce the WSINDy algorithm applied to interacting particles, including hyperparameter selection, computational complexity, and convergence of the method under suitable assumptions in the limit of large N. Section 4.5 contains numerical examples exhibiting the convergence rates of the previous section and examining the robustness of the algorithm to various sources of corruption, and Section 4.6 contains a discussion of extensions and future directions. In the Appendix we provide information on the hyperparameters used (4.A), derivation of the homogenized equation (4.5.3) (4.A.1), results and discussion for the case of small N and large M (in

comparison with [90]) (4.B), and proofs to technical lemmas (4.C). Table 4.4.1 includes a list of notations used throughout.

4.3 Review of mean-field theory

Our weak-form approach utilizes that under fairly general assumptions the empirical measure μ_t^N of the process $\mathbf{X}_t$ defined in (4.1.1) converges weakly to μ_t, the distribution of the associated mean-field process X_t defined in (4.3.2). Specifically, under suitable assumptions on V, K, σ, and μ_0, there exists $T > 0$ such that for all $t \in [0, T]$, the mean-field limit[4]

$$\lim_{N\to\infty} \mu_t^N = \mu_t$$

holds in the weak topology of measures[5], where μ_t is a weak-measure solution to the mean-field dynamics

$$\partial_t \mu_t = \nabla \cdot (\mu_t \nabla K * \mu_t) + \nabla \cdot (\mu_t \nabla V) + \frac{1}{2} \sum_{i,j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j} \left(\sigma\sigma^T \mu_t\right), \quad \mu_0 \in \mathcal{P}_2(\mathbb{R}^d). \tag{4.3.1}$$

Equation 4.3.1 describes the evolution of the distribution of the McKean-Vlasov process

$$dX_t = -\nabla K * \mu_t (X_t)\, dt - \nabla V (X_t)\, dt + \sigma(X_t)\, dB_t. \tag{4.3.2}$$

This implies that as $N \to \infty$, an initially correlated particle system driven by pairwise interaction becomes uncorrelated and only interacts with its mean-field distribution μ_t. In particular, the following theorem summarizes several mean-field results taken from the review article [65] with proofs in [132, 96]. [6]

Theorem. *[65, 132, 96] Assume that ∇K is globally Lipschitz, $V = 0$, and $\sigma(x) = \sigma = const.$ In addition assume that $\mu_0 \in \mathcal{P}_2(\mathbb{R}^d)$. Then for any $T > 0$, for all $t \le T$ it holds that*

(i) There exists a unique solution (X_t, μ_t) where X_t is a strong solution to (4.3.2) and μ_t is a weak-measure solution to (4.3.1).

[4] We use the notation $t \to \mu_t$ to denote the evolution of probability measures. Subscripts will not be used to denote differentiation.

[5] Meaning that for all continuous bounded functions $\phi : \mathbb{R}^d \to \mathbb{R}$, $\int_{\mathbb{R}^d} \phi(x) d\mu_t^N(x) \to \int_{\mathbb{R}^d} \phi(x) d\mu_t(x)$.

[6] For a function $f : \mathbb{R}^d \to Y$, where Y is a metric space with metric ρ, we define $\mathrm{Lip}(f)$ by

$$\mathrm{Lip}(f) := \sup_{x,y\in\mathbb{R}^d} \frac{\rho(f(x), f(y))}{|x - y|}$$

where $|\cdot|$ denotes the Euclidean norm. We say f is Lipschitz when $\mathrm{Lip}(f) < \infty$. Also, $\|f\|_{C^1} := \|f\|_\infty + \sum_{i=1}^d \left\| \frac{\partial f}{\partial x_i} \right\|_\infty$.

(ii) For any $\phi \in C_b^1(\mathbb{R}^d)$,

$$\mathbb{E}\left|\frac{1}{N}\sum_{i=1}^{N}\phi(X_t^{(i)}) - \int_{\mathbb{R}^d}\phi(x)d\mu_t(x)\right| \leq C\frac{\|\phi\|_{C^1}}{\sqrt{N}} \tag{4.3.3}$$

with C *depending on* $Lip(\nabla K)$ *and* T.

(iii) For any $k \in \mathbb{N}$, *a.e.-*$t < T$, *the* k*-particle marginal*

$$\rho_t^{(k),N}(x_1,\dots,x_k) := \int_{\mathbb{R}^{d(N-k)}} F_t^N(x_1,\dots,x_k,x_{k+1},\dots,x_N)\,dx_{k+1}\cdots dx_N$$

converges weakly to $\mu_t^{\otimes k}$ *as* $N \to \infty$, *where* $F_t^N \in \mathcal{P}(\mathbb{R}^{Nd})$ *is the distribution of* $\mathbf{X}_t$.

The previous result immediately extends to the case of ∇V and σ both globally Lipschitz and has been extended to ∇K only locally-Lipschitz in [18], ∇K with Coulomb-type singularity at the origin in [17], and domains with boundaries in [48, 47]. Analysis of the model (4.3.1) continues to evolve in various contexts, including analysis of equilibria [97, 50, 28] and connections to deep learning [4]. For our convergence result below we simply assume that $K^\star, V^\star, \sigma^\star$ and μ_0 are such that (i) and (ii) from the above theorem hold.

4.3.1 Weak form

Despite the $\mathcal{O}(N^{-1/2})$ convergence of the empirical measure in previous theorem, it is unclear at what particle number N the mean-field equations become a suitable framework for inference using particle data, due to the complex variance structure at any finite N. A key piece of the present work is to show that the weak form of the mean-field equations does indeed provide a suitable setting when N is at least several thousands. Moreover, since in many cases (4.3.1) can only be understood in a weak sense, the weak form is the natural framework for identification. We say that μ_t is a weak solution to (4.3.1) if for any $\psi \in C^2(\mathbb{R}^d \times (0,T))$ compactly supported it holds that

$$\begin{aligned}\int_0^T\int_{\mathbb{R}^d}\partial_t\psi(x,t)\,d\mu_t(x)dt = \int_0^T\int_{\mathbb{R}^d}\Big(&\nabla\psi(x,t)\cdot\nabla K * \mu_t(x) + \nabla\psi(x,t)\cdot\nabla V(x)\\ &- \frac{1}{2}\mathrm{Tr}\left(\nabla^2\psi(x,t)\sigma(x)\sigma^T(x)\right)\Big)d\mu_t(x)dt,\end{aligned} \tag{4.3.4}$$

where $\nabla^2\psi$ denotes the Hessian of ψ and $\mathrm{Tr}(\mathbf{A})$ is the trace of the matrix $\mathbf{A}$. Our method requires discretizing (4.3.4) for all $\psi \in \Psi$ where $\Psi = (\psi_1,\dots,\psi_n)$ is a suitable test function basis, and approximating the mean-field distribution μ_t with a density U_t constructed from discrete particle data at time t. We then find K, V, and σ within specified finite-dimensional function spaces.

4.4 Algorithm

We propose the general Algorithm 4.4.1 for discovery of mean-field equations from particle data. The inputs are a discrete-time sample $\mathbb{Y}$ containing M experiments each with N particle positions over L timepoints $\mathbf{t} = (t_1, \dots, t_L)$. The following hyperparameters are defined by the user: (i) a kernel G used to map the empirical measure μ_t^N to an approximate density U_t, (ii) a spatial grid $\mathbf{C}$ over which to evaluate the approximate density $\mathbf{U}_t = U_t(\mathbf{C})$, (iii) a library of trial functions $\mathbb{L} = \{\mathbb{L}_K, \mathbb{L}_V, \mathbb{L}_\sigma\} = \{(K_j)_{j=1}^{J_K}, (V_j)_{j=1}^{J_V}, (\sigma_j)_{j=1}^{J_\sigma}\}$, (iv) a basis of test functions $\Psi = (\psi_k)_{k=1}^n$, (v) a quadrature rule over the spatiotemporal grid $(\mathbf{C}, \mathbf{t})$ denoted by an inner product $\langle \cdot, \cdot \rangle$, and (vi) sparsity factors $\boldsymbol{\lambda}$ for the modified sequential thresholding least-squares Algorithm 4.4.2 (MSTLS) reviewed below. We discuss choices of these hyperparameters in Section 4.4.1, computational complexity of the algorithm in Section 4.4.2, convergence of the algorithm in Section 4.4.3. In Section 4.4.4 we briefly discuss gaps between theory and practice. Table 4.4.1 includes a list of notations used throughout.

4.4.1 Hyperparameter Selection

Quadrature

We assume that the set of gridpoints $\mathbf{C}$ in Algorithm 4.4.1 is chosen from some compact domain $\mathcal{D} \subset \mathbb{R}^d$ containing $\text{supp}(\mathbb{Y})$. The choice of $\mathbf{C}$ (and $\mathcal{D}$) must be chosen in conjunction with the quadrature scheme, which includes integration in time using the given timepoints $\mathbf{t}$ as well as space. For completeness, the inner products in lines 10, 16, 22, and 27 of Algorithm 4.4.1 are defined in the continuous setting by

$$\langle f, g \rangle = \int_0^T \int_{\mathcal{D}} f(x,t) g(x,t) dx dt,$$

and the convolution in line 10 is defined by

$$\nabla K_j * U_t(x) = \int_{\mathcal{D}} \nabla K_j(x-y) U_t(y) dy.$$

In the present work we adopt the scheme used in the application of WSINDy for local PDEs [99], which includes the trapezoidal rule in space and time with test functions ψ compactly supported in $\mathcal{D} \times (0,T)$. We take $\mathcal{D}$ to be a rectangular domain enclosing $\text{supp}(\mathbb{Y})$ and $\mathbf{C} \subset \mathcal{D}$ to be equally-spaced in order to efficiently evaluate convolution terms. In what follows we denote by $\langle \cdot, \cdot \rangle$ the continuous inner product, $\langle \cdot, \cdot \rangle_h$ the inner product over $\mathcal{D} \times [0,T]$ evaluated using the composite trapezoidal rule in space with meshwidth h and Lebesgue integration in time, and by $\langle \cdot, \cdot \rangle_{h,\Delta t}$ the trapezoidal rule in both space and time, with meshwidth h in space and Δt in time. With some abuse of notation, $f * g$ will denote the convolution of f and g, understood to be discrete or continuous by the context. Note also that we denote by μ^N, μ, and U the measures over $\mathbb{R}^d \times [0,T]$ defined by $\mu_t^N \Lambda_{[0,T]}$, $\mu_t \Lambda_{[0,T]}$ and $U_t \Lambda_{[0,T]}$, respectively, where $\Lambda_{[0,T]}$ is the Lebesgue measure on

Algorithm 4.4.1 WSINDy for identifying mean-field equation (4.3.1) from particle data $\mathbb{Y}$
$(\widehat{\mathbf{w}}, \hat{\lambda}) = \mathbf{WSINDy}(\mathbb{Y}, \mathbf{t}\,;\ G,\ \mathbf{C},\ \mathbb{L},\ \Psi,\ \langle\cdot,\cdot\rangle\,,\ \boldsymbol{\lambda})$

for $\ell = 1 : L$ **do**
  for $m = 1 : M$ **do**
    $\mathbf{U}_\ell^{(m)} = \int_{\mathbb{R}^d} G(\mathbf{C}, y) d\mu_{t_\ell}^N(y)$ where $\mu_{t_\ell}^N$ is the empirical measure for $\mathbf{Y}_{t_\ell}^{(m)}$
  end for
  $\mathbf{U}_\ell = \frac{1}{M} \sum_{m=1}^M \mathbf{U}_\ell^{(m)}$
end for

for $j = 1 : J_K$ **do**
  for $k = 1 : n$ **do**
    $\mathbf{G}_{kj}^K = \langle \nabla\psi_k, \mathbf{U} \nabla K_j * \mathbf{U} \rangle$
  end for
end for

for $j = 1 : J_V$ **do**
  for $k = 1 : n$ **do**
    $\mathbf{G}_{kj}^V = \langle \nabla\psi_k, \mathbf{U} \nabla V_j \rangle$
  end for
end for

for $j = 1 : J_\sigma$ **do**
  for $k = 1 : n$ **do**
    $\mathbf{G}_{kj}^\sigma = \frac{1}{2} \sum_{p,q=1}^d \left\langle \partial_{x_p x_q} \psi_k, (\sigma_j \sigma_j^T)_{pq} \mathbf{U} \right\rangle$
  end for
end for
$\mathbf{G} = [\mathbf{G}^K\ \mathbf{G}^V\ \mathbf{G}^\sigma]$

for $k = 1 : n$ **do**
  $\mathbf{b}_k = \langle \partial_t \psi_k, \mathbf{U} \rangle$
end for

$(\widehat{\mathbf{w}}, \widehat{\lambda}) = \mathrm{MSTLS}(\mathbf{G}, \mathbf{b}; \boldsymbol{\lambda})$ (see Algorithm 4.4.2)

$[0, T]$.

Density Kernel

Having chosen the domain $\mathcal{D} \subset \mathbb{R}^d$ containing the particle data $\mathbb{Y}$, let $P^h = \{B_k\}_k$ be a partition of $\mathcal{D}$ ($\cup_k B_k = \mathcal{D}$) with h indicating the size of the atoms B_k. For the remainder of the chapter we take B_k to be hypercubes of equal side length h in order to minimize computation time for integration, although this is by no means necessary. For particle positions $\mathbf{X}_t$, we define the histogram[7]

$$U_t = \sum_k \frac{1}{|B_k|} \mathbb{1}_{B_k}(x) \left(\frac{1}{N} \sum_i \mathbb{1}_{B_k}(X_t^{(i)}) \right) = \int_{\mathcal{D}} G(x, y) d\mu_t^N(y). \tag{4.4.1}$$

Here the *density kernel* is defined

$$G(x, y) = \sum_k \frac{1}{|B_k|} \mathbb{1}_{B_k}(x) \mathbb{1}_{B_k}(y),$$

and in this setting the corresponding spatial grid $\mathbf{C} = (\mathbf{c}_k)_k$ is the set of center-points of the bins B_k, from which we define the discrete histogram data $\mathbf{U}_t = U_t(\mathbf{C})$. The discrete histogram $\mathbf{U}_t$ then serves as an approximation to the mean-field distribution μ_t.

Pointwise estimation of densities from samples of particles usually requires large numbers of particles to achieve reasonably low variance, and in general the variance grows inversely proportional to the bin width h. One benefit of the weak form is that integrating against a histogram U does not suffer from the same increase in variance with small h. In particular,

Lemma 2. *Let $(Y^{(1)}, Y^{(2)}, \dots)$ be a sequence of $\mathbb{R}^d$-valued random variables such that the empirical measure μ^N of $\mathbf{Y} := (Y^{(1)}, \dots, Y^{(N)})$ converges weakly to $\mu \in \mathcal{P}(\mathbb{R}^d)$ according to*

$$\mathbb{E}\left[(\langle \psi, \mu^N \rangle - \langle \psi, \mu \rangle)^2 \right] \leq C \|\psi\|_{C^1}^2 N^{-1} \tag{4.4.2}$$

for all $\psi \in C^1(\mathbb{R}^d)$ and C a universal constant. Let U be the histogram computed with kernel G using (4.4.1) with n bins and equal sidelength h. Then for any ψ in $C^1(\mathbb{R}^d)$ compactly supported in $\mathcal{D}$, we have the mean-squared error (for $\widetilde{C}$ depending on C and d)

$$\mathbb{E}\left[(\langle \psi, U \rangle_h - \langle \psi, \mu \rangle)^2 \right] \leq \widetilde{C} \|\psi\|_{C^1}^2 \left(h^2 + N^{-1} \right).$$

Remark 2. We note that (4.4.2) follows immediately for $Y^{(i)} \sim \mu$ i.i.d.[8], and also for $\mathbf{Y} = \mathbf{X}_t$ a solution to (4.1.1) at time t with mean-field distribution $\mu = \mu_t$ according to (4.3.3) (for suitable K, V, and σ), which is the setting of the current chapter.

[7] The indicator function is defined $\mathbb{1}_A(x) := \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$.

[8] In this case (4.4.2) is the variance of a Monte-Carlo estimator for $\int \psi(x) d\mu(x)$.

of Lemma 2. First we note that by compact support of ψ, the trapezoidal rule can be written

$$\langle \psi, U \rangle_h = \left\langle \psi, \int_{\mathbb{R}^d} G(\cdot, y) d\mu^N(y) \right\rangle_h = \langle \psi^{\mathbf{C}}, \mu^N \rangle = \frac{1}{N} \sum_{i=1}^{N} \psi^{\mathbf{C}}(Y^{(i)})$$

where the midpoint approximation $\psi^{\mathbf{C}}$ of ψ is given by

$$\psi^{\mathbf{C}}(x) = \sum_{k=1}^{K} \psi(c_k) \mathbb{1}_{B_k}(x). \tag{4.4.3}$$

Hence we simply split the error and use (4.4.2):

$$\begin{aligned} \mathbb{E}\left[\left(\langle \psi, U \rangle_h - \langle \psi, \mu \rangle \right)^2 \right] &\leq 2\mathbb{E}\left[\langle \psi^{\mathbf{C}} - \psi, \mu^N \rangle^2 \right] + 2\mathbb{E}\left[\left(\langle \psi, \mu^N \rangle - \langle \psi, \mu \rangle \right)^2 \right] \\ &\leq \|\psi\|_{C^1}^2 \left(\frac{d}{2} h^2 + 2CN^{-1} \right). \end{aligned}$$

□

The previous lemma in particular shows that small bin width h does not negatively impact $\langle \psi, U \rangle_h$ as an estimator of $\langle \psi, \mu \rangle$, which is in contrast to $U(x)$ as a pointwise estimator of $\mu(x)$. For example, if we assume that $\mathbf{Y}$ is sampled from a C^1 density μ, it is well known that the mean-square optimal bin width is $h = \mathcal{O}(N^{-1/3})$ [53]. Summarizing this result, elementary computation reveals the pointwise bias for $x \in B_k$,

$$\text{bias}(U(x)) = \mathbb{E}\left[U(x) \right] - \mu(x) = \frac{\mu(B_k)}{|B_k|} - \mu(x) := \mu(\xi) - \mu(x)$$

for some $\xi \in B_k$. Letting $L_k = \max_{x \in B_k} |\nabla \mu(x)|$, we have

$$\text{bias}(U(x))^2 \leq L_k^2 2^{d-1} h^2.$$

For the variance we get

$$\text{Var}\left(U(x) \right) = \frac{1}{N} \frac{\mu(B_k)(1 - \mu(B_k))}{|B_k|^2} = \frac{\mu(\xi)}{N} \left(1 - \mu(B_k) \right) \frac{1}{\sqrt{2}^{d-1} h},$$

and hence a bound for the mean-squared error

$$\mathbb{E}\left[\left(U(x) - \mu(x) \right)^2 \right] \leq L_k^2 2^{d-1} h^2 + \frac{\mu(\xi)}{N \sqrt{2}^{d-1}} h^{-1}.$$

Minimizing the bound over h we find an approximately optimal bin width

$$h^* = \left(\frac{\rho(\xi)}{2^{\frac{3d-1}{2}} L_k^2} \right)^{1/3} N^{-1/3} = \mathcal{O}(N^{-1/3}),$$

which provides an overall pointwise root-mean-squared error of $\mathcal{O}(N^{-1/3})$. Hence, not only does the weak form remove the inverse h dependence in the variance, but fewer particles are needed to accurately approximate integrals of the density μ.

Test Function Basis

For the test functions $(\psi_k)_{1\leq k\leq n}$ we use the same approach as the PDE setting [99], namely we fix a *reference test function* ψ and set

$$\psi_k(x,t) = \psi(\mathbf{x}_k - x, t_k - t)$$

where $\mathcal{Q} := \{(\mathbf{x}_k, t_k)\}_{1\leq k\leq n}$ is a fixed set of *query points*. This, together with a separable representation

$$\psi(x,t) = \phi_1(x_1)\cdots\phi_d(x_d)\phi_{d+1}(t),$$

enables construction of the linear system $(\mathbf{G}, \mathbf{b})$ using the FFT. We choose ϕ_j, $1 \leq j \leq d+1$, of the form

$$\phi_{m,p}(v;\,\Delta) := \max\left(1 - \left(\frac{v}{m\Delta}\right)^2, 0\right)^p \tag{4.4.4}$$

where m is the integer *support parameter* such that $\phi_{m,p}$ is supported on $2m+1$ points of spacing $\Delta \in \{h, \Delta t\}$ and $p \geq 1$ is the *degree* of $\phi_{m,p}$. For simplicity we set $\phi_j = \phi_{m_x,p_x}$ for $1 \leq j \leq d$ and $\phi_{d+1} = \phi_{m_t,p_t}$, so that only the numbers m_x, p_x, m_t, p_t need to be specified.

Since $\phi_{m,p}$ has exactly p weak derivatives, p_x and p_t must be at least as large as the maximum spatial and temporal derivatives appearing in the library $\mathbb{L}$, or $p_x \geq 2$, $p_t \geq 1$. Larger p results in higher-accuracy enforcement of the weak form (4.3.4) in low-noise situations (see Lemma 2 of [100] for details), however the convergence analysis below indicates that smaller $\mathrm{Lip}(\partial^\alpha\psi)$, $|\alpha| \leq 2$, may reduce variance. The support parameter m determines the length and time scales of interest and must be chosen small enough to extract relevant scales yet large enough to sufficiently reduce variance.

In [99, Appendix A] the authors developed a changepoint algorithm to choose m_x, m_t, p_x, p_t automatically from the Fourier spectrum of the data $\mathbf{U}$. Here, for each of the three examples in Section 4.5, we fix ψ across all particle numbers N, extrinsic noises ε, and intrinsic noises σ, in order to instead focus on convergence in N. To strike a balance between accuracy and small $\mathrm{Lip}(\psi)$ we choose $p_t = 3$ and $p_x = 5$ throughout. We used a combination of the changepoint algorithm and manual tuning to arrive at m_x and m_t which work well across all noise levels and numbers of particles examined. Query points $\mathcal{Q}$ are taken to be an equally-spaced subgrid of $\mathbf{C}$ with spacing

s_x and s_t for spatial and temporal coordinates. The resulting values p_x, p_t, m_x, m_t, s_x, and s_t determinine the weak discretization scheme and can be found in Appendix 4.A for each example below.

The results in Section 4.5 appear robust to $3 \leq p_x, p_t \leq 9$. In addition, choosing m_x and m_t specific to each dataset $\mathbb{Y}$ using the changepoint method often improves results. Although automated in the changepoint algorithm, we recommend visualizing the overlap between the Fourier spectra of ψ and $\mathbf{U}$ when choosing p_x, p_t, m_x, m_t in order to directly observe which the modes in the data will experience filtering under convolution with ψ. In general, there is much flexibility in the choice of ψ. Optimizing ψ continues to be an active area of research.

Trial Function Library

The general Algorithm 4.4.1 does not impose a radial structure for the interaction potential K, nor does it assume any prior knowledge that the particle system is in fact interacting. In the examples below[9], the libraries $\mathbb{L}_K, \mathbb{L}_V, \mathbb{L}_\sigma$ are composed of monomial and/or trigonometric terms to demonstrate that sparse regression is effective in selecting the correct combination of nonlocal drift, local drift, and diffusion terms. Rank deficiency can result, however, from naive choices of nonlocal and local bases. Consider the kernel $K(x) = \frac{1}{2}|x|^2$, which satisfies

$$\nabla K * \mu_t = x - M_1(\mu_t) = \nabla V(x)$$

where $V(x) = \frac{1}{2}|x - M_1(\mu_t)|^2$ and $M_1(\mu_t)$ is the first moment of μ_t. Since $M_1(\mu_t)$ is conserved in the model (4.3.2) posed in free-space[10], including the same power-law terms in both libraries $\mathbb{L}_K$ and $\mathbb{L}_V$ will lead to rank deficiency. This is easily avoided by incorporating known symmetries of the model (4.3.2), however in general we recommend that the user build the library $\mathbb{L}$ incrementally and monitor the condition number of $\mathbf{G}$ while selecting terms.

Sparse Regression

As in [99], we enforce sparsity using a *modified* sequential thresholding least-squares algorithm (MSTLS), included as Algorithm 4.4.2 below, where the "modifications" are two-fold. First, we incorporate into the thresholding step the magnitude of the overall term $\|\mathbf{w}_j \mathbf{G}_j\|_2$ as well as the coefficient magnitude $|\mathbf{w}_j|$, by defining non-uniform lower and upper thresholds

$$\begin{cases} L_j^\lambda = \lambda \max\left\{1, \ \dfrac{\|\mathbf{b}\|}{\|\mathbf{G}_j\|}\right\} \\ U_j^\lambda = \dfrac{1}{\lambda} \min\left\{1, \ \dfrac{\|\mathbf{b}\|}{\|\mathbf{G}_j\|}\right\} \end{cases}, \qquad 1 \leq j \leq \mathfrak{J}, \tag{4.4.5}$$

[9] Details of the libraries used in examples can be found in Tables 4.A.1-4.A.3 in Appendix 4.A.

[10] This is not true in domains with boundaries, where nonlocalities can be seen to impart mean translation [97].

where $\mathfrak{J} = J_K + J_V + J_\sigma$ is the number of columns in $\mathbf{G}$. Second, we perform a grid search[11] over candidate sparsity parameters $\boldsymbol{\lambda}$ and choose the parameter $\widehat{\lambda}$ that is the smallest minimizer over $\boldsymbol{\lambda}$ of the cost function

$$\mathcal{L}(\lambda) = \frac{\left\|\mathbf{G}(\mathbf{w}^\lambda - \mathbf{w}^0)\right\|_2}{\|\mathbf{G}\mathbf{w}^0\|_2} + \frac{\left\|\mathbf{w}^\lambda\right\|_0}{\mathfrak{J}} \tag{4.4.6}$$

where $\mathbf{w}^\lambda$ is the output of the sequential thresholding algorithm with non-uniform thresholds (4.4.5) and $\mathbf{w}^0 = \mathbf{G}^\dagger\mathbf{b}$ is the least-squares solution[12]. The final coefficient vector is then set to $\widehat{\mathbf{w}} = \mathbf{w}^{\widehat{\lambda}}$.

Algorithm 4.4.2 Modified sequential thresholding with automatic threshold selection
$(\widehat{\mathbf{w}}, \hat{\lambda}) = \mathbf{MSTLS}(\mathbf{G} \in \mathbb{R}^{n\times\mathfrak{J}}, \mathbf{b} \in \mathbb{R}^n, \boldsymbol{\lambda} \in \mathbb{R}^m, \text{maxits})$

$\mathbf{W} = \mathbf{0} \in \mathbb{R}^{\mathfrak{J}\times m}$
$\mathbf{w}^0 = \mathbf{G}^\dagger\mathbf{b}$
for $i = 1 : m$ **do**
  $\lambda = \boldsymbol{\lambda}_i$
  $\ell = 0$
  while $\ell <$ maxits **do**
    $\mathcal{I}^\ell = \{1 \le j \le \mathfrak{J} \ : \ L_j^\lambda \le |\mathbf{w}_j^\ell| \le U_j^\lambda\}$ (Thresholding step: see equation (4.4.5))
    $\mathbf{w}^{\ell+1} = \arg\min_{\text{supp}(\mathbf{w})\subset\mathcal{I}^\ell} \|\mathbf{G}\mathbf{w} - \mathbf{b}\|_2^2$
    $\ell = \ell + 1$
  end while
  $\mathbf{w}^\lambda = \mathbf{w}^\ell$
  $\mathbf{W}_{:,i} = \mathbf{w}^\lambda$
end for
$\widehat{\lambda} = \min\left(\arg\min_{\lambda\in\boldsymbol{\lambda}} \mathcal{L}(\lambda)\right)$ (Identificaiton of best λ: see equation (4.4.6))
$\widehat{\mathbf{w}} = \mathbf{w}^{\widehat{\lambda}}$

We now review some aspects of Algorithm 4.4.2. Results from [158] on the convergence of STLS carry over for the inner loop of Algorithm 4.4.2, namely if $\mathbf{G}$ is full-rank, the inner loop terminates in at most $\mathfrak{J}$ iterations, and if maxits$\ge \mathfrak{J}$, the resulting coefficient vector $\mathbf{w}^\lambda$ is a local minimizer of the cost function $F(\mathbf{w}) = \|\mathbf{G}\mathbf{w} - \mathbf{b}\|_2^2 + \lambda^2 \|\mathbf{w}\|_0$. This implies that the full algorithm terminates in at-most $m\mathfrak{J}$ least-squares solves (each on a subset of columns of $\mathbf{G}$).

When considering recovery of the true weight vector $\mathbf{w}^\star$, Theorem 1 below implies convergence in particle number N of $\widehat{\mathbf{w}}$ to $\mathbf{w}^\star$ when $\mathbf{G}$ is full-rank. The rate of convergence depends implicitly on the condition number of $\mathbf{G}$, hence it is recommended that one build the library $\mathbb{L}$ incrementally, stopping before the conditional number $\kappa(\mathbf{G})$ grows too large. If $\mathbf{G}$ is rank deficient, classical recovery guarantees from compressive sensing do not necessarily apply, due to high correlations between the columns of $\mathbf{G}$ (recall each column is constructed from the same dataset $\mathbf{U}$)[13]. One may employ additional regularization (e.g. Tikhonov regularization as in [117]); however, in general, im-

[11] Note that this is feasible because the STLS algorithm terminates in finitely many iterations.

[12] The Moore-Penrose inverse $\mathbf{A}^\dagger$ is defined for a rank-r matrix $\mathbf{A}$ using the reduced SVD $\mathbf{A} = U_r\Sigma_r V_r^*$ as $\mathbf{A}^\dagger := V_r\Sigma_r^{-1}U_r^*$. The subscript r denotes restriction to the first r columns.

[13] In particular, correlations result in large mutual incoherence, which renders algorithms such as Basis Pursuit, Orthogonal Matching Pursuit, and Hard Thresholding Pursuit useless (see [52, Chapter 5] for details).

provements to existing sparse regression algorithms for rank-deficient, noisy, and highly-correlated matrices is an active area of research.

The bounds (4.4.5) enforce a quasi-dominant balance rule, such that $\|\mathbf{w}_j\mathbf{G}_j\|_2$ is within $-\log_{10}(\lambda)$ orders of magnitude from $\|\mathbf{b}\|_2$ and $|\mathbf{w}_j|$ is within $-\log_{10}(\lambda)$ orders of magnitude from 1 (the coefficient of time derivative $\partial_t\mu_t$). This is specifically designed to handle poorly-scaled data (see the Burgers and Korteweg-de Vries examples in [99]), however we leave a more thorough examination of the thresholding requirements necessary for models with multiple scales to future work.

As the sum of two relative errors, minimizers of the cost function $\mathcal{L}$ equally weight the accuracy and sparsity of $\mathbf{w}^{\widehat{\lambda}}$. By choosing $\widehat{\lambda}$ to be the smallest minimizer of $\mathcal{L}$ over $\boldsymbol{\lambda}$, we identify the thresholds $\lambda \in \boldsymbol{\lambda}$ such that $\lambda < \widehat{\lambda}$ as those resulting in an overfit model. We commonly choose $\boldsymbol{\lambda}$ to be log-equally spaced (e.g. 50 points from 10^{-4} to 1), and starting from a coarse grid, refine $\boldsymbol{\lambda}$ until the minimum of $\mathcal{L}$ is stationary.

4.4.2 Computational Complexity

To compute convolutions against ∇K for each $K \in \mathbb{L}_K$, we first evaluate $(\partial_{x_i}K)_{1\le i\le d}$ at the grid $\mathbf{C}-\mathbf{C}$ defined by

$$\mathbf{C}-\mathbf{C} := \{x \in \mathbb{R}^d \ : \ x = (i_1h, \ldots, i_dh), \quad -n_\ell \le i_\ell \le n_\ell\},$$

where h is the spacing of $\mathbf{C}$ and n_ℓ, $1 \le \ell \le d$, is the number of points in $\mathbf{C}$ along the ℓth coordinate. Computing[14] $\partial_{x_i}\mathbf{K} := \partial_{x_i}K(\mathbf{C}-\mathbf{C})$ requires $2^d|\mathbf{C}|$ evaluations of K, where $|\mathbf{C}| = \prod_{\ell=1}^d n_\ell$ is the number of points in $\mathbf{C}$. We then use the d-dimensional FFT to compute the convolutions

$$\partial_{x_i}\mathbf{K} * \mathbf{U}_t \approx \partial_{x_i}K * U_t(\mathbf{C}), \quad t \in \mathbf{t}$$

where only entries corresponding to particle interactions within $\mathbf{C}$ are retained. For $d = 1$ this amounts to $\mathcal{O}(|\mathbf{C}|\log|\mathbf{C}|)$ flops per timestep. For $d = 2$ and higher dimensions, the d-dimensional FFT is considerably slower unless one of the arrays is separable. To enforce separability, trial interaction potentials in $\mathbb{L}_K$ can be chosen to be a sum of separable functions,

$$K(x) = \sum_{q=1}^{Q} k_{1,q}(x_1)\cdots k_{d,q}(x_d), \tag{4.4.7}$$

in which case only a series of one-dimensional FFTs are needed to compute $\partial_{x_i}\mathbf{K} * \mathbf{U}_t$, and again the cost is $\mathcal{O}(|\mathbf{C}|\log|\mathbf{C}|)$ per timestep. When K is not separable, a low-rank approximation can

[14]Note that $\mathbf{C}-\mathbf{C}$ is simply $\mathbf{C}$ shifted to lie in the positive orthant $\{x \in \mathbb{R}^d \ : \ x_\ell \ge 0, \quad 1 \le \ell \le d\}$ and reflected through each coordinate plane $x_\ell = 0$. In this way $\mathbf{C}-\mathbf{C}$ discretizes the set $\mathcal{D}-\mathcal{D} := \{x-y \in \mathbb{R}^d \ : \ (x,y) \in \mathcal{D}\times\mathcal{D}\}$ containing all observed interparticle distances.

be computed from $\partial_{x_i}\mathbf{K}$,

$$\partial_{x_i}\mathbf{K} \approx \sum_{q=1}^{Q} \sigma_q \mathbf{k}_{1,q} \otimes \cdots \otimes \mathbf{k}_{d,q} \tag{4.4.8}$$

which again reduces convolutions to a series of one-dimensional FFTs. For $d = 2$, this is accomplished using the truncated SVD, while for higher dimensions there does not exist a unique *best* rank-Q tensor approximation, although several efficient algorithms are available to compute a sufficiently accurate decomposition [94, 130, 66] (and the field of fast tensor decompositions is advancing rapidly).

We propose to compute convolutions by first computing a low-rank decomposition of $\partial_{x_i}\mathbf{K}$ using the randomized truncated SVD [155] or a suitable randomized tensor decomposition and then applying the d-dimensional FFT as a series of one-dimensional FFTs. In the examples below we consider only $d = 1$ and $d = 2$, and leave extension to higher dimensions to future work.

Using low-rank approximations, the mean-field approach provides a significant reduction in computational complexity compared to direct evaluations of particle trajectories when N is sufficiently large. A particle-level computation of the nonlocal force in weak-form requires evaluating terms of the form

$$\sum_{\ell=1}^{L}\left(\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\partial_x\psi(X_{t_\ell}^{(i)},t_\ell)\partial_x K(X_{t_\ell}^{(i)}-X_{t_\ell}^{(j)})\right)\Delta t = \left\langle \partial_x\psi, \mu^N(\partial_x K * \mu^N)\right\rangle_{h,\Delta t}.$$

For a single candidate interaction potential K, a collection of J test functions ψ, and M experiments, this amounts to $MLN^2 + MLNJ$ function evaluations in $\mathbb{R}^d$ and $\mathcal{O}(MLN^2J)$ flops. If we use the proposed method, employing the convolutional weak form with a separable reference test function ψ (as in WSINDy for PDEs [99]) and exploiting a rank Q approximation of $\partial_x\mathbf{K}$ when computing convolutions against interaction potential, we instead evaluate

$$\partial_x\psi * (U(\partial_x K * U))$$

using $\mathcal{O}(LQ|\mathbf{C}|\log(|\mathbf{C}|))$ flops and only $2^d|\mathbf{C}|$ evaluations of $\partial_x K$, reused at each of the L timepoints[15]. Figure 4.4.1 provides a visualization of the reduction in function evaluations for $L = 100$ timepoints and $M = 10$ experiments over a range of N and $|\mathbf{C}|^{1/d}$ (points along each spatial dimension when $|\mathbf{C}|$ is a hypercube) in $d = 2$ and $d = 3$ spatial dimensions. Table 4.A.4 in Appendix 4.A lists walltimes for the examples below, showing that with $N = 64,000$ particles the full algorithm implemented in MATLAB runs in under 10 seconds with all computations in serial on a laptop with an AMD Ryzen 7 pro 4750u processor, and requiring less than 8Gb of RAM. The dependence on N is only through the $\mathcal{O}(N)$ computation of the histrogram, hence this approach may find applications in physical coarse-graining (e.g. of molecular dynamics or plasma simulations).

[15] We neglect the cost of computing the histogram $\mathbf{U}$ and evaluating $\psi(\mathbf{C})$, together amounting to an additional $\mathcal{O}(NML + |\mathbf{C}|)$ flops, as these terms are lower order and reused in each column of $\mathbf{G}$ and $\mathbf{b}$.

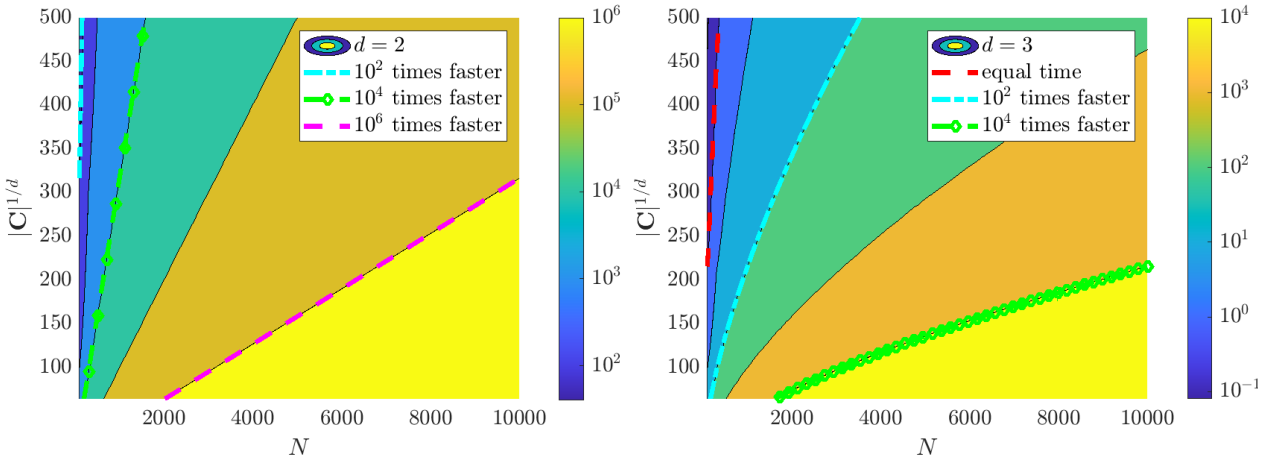


Figure 4.4.1: Factor by which the mean-field evaluation of interaction forces using histograms reduces total function evaluations as a function of particle number N and average gridpoints per dimension $|\mathbf{C}|^{1/d}$ for data with $M = 10$ experiments each with $L = 100$ timepoints. For example, with $d = 2$ spatial dimensions (left) and $N > 2000$ particles, the number of function evaluations is reduced by at least a factor of 10^4.

4.4.3 Convergence

We now show that the estimators $\widehat{K}$, $\widehat{V}$, and $\widehat{\sigma}$ of the weak-form method converge with a rate $\mathcal{O}(h + N^{-1/2} + \Delta t^{\eta})$ when ordinary least squares is used (i.e. $\boldsymbol{\lambda} = 0$) and only $M = 1$ experiment is available. Here $\eta > 0$ is the Hölder exponent of the sample paths of the process $\mathbf{X}_t$. We assume that $\mathcal{D}$, $\mathbf{C}$, G, P^h and the resulting histogram $\mathbf{U} = (\mathbf{U}_t)_{t \leq T}$ are as in Section 4.4.1. We make the following assumptions on the true model and resulting linear system throughout this section.

Assumptions H. Let $p \geq 1$ be fixed.

(H.1) For each $N \geq 2$, $\mathbf{X}_t = (X_t^{(1)}, \ldots, X_t^{(N)})$ is a strong solution to (4.1.1) for $t \in [0, T]$, and for some $\eta > 0$ the sample paths $t \to X_t^{(i)}(\omega)$ are almost-surely η-Hölder continuous, i.e. for some $C_\eta > 0$,

$$|X_t^{(i)}(\omega) - X_s^{(i)}(\omega)| \leq C_\eta |t - s|^\eta, \quad \forall\, 0 \leq s \leq t \leq T, \quad \forall\, 1 \leq i \leq N, \quad \text{for a.e. } \omega \in \Omega.$$

(H.2) The initial particle distribution μ_0 satisfies the moment bound

$$\int_{\mathbb{R}^d} |x|^p d\mu_0(x) := M_p < \infty.$$

(H.3) $\nabla K^\star$ and $\nabla V^\star$ satisfy for some $C_p > 0$ the growth bound:

$$|\nabla V^\star(x) - \nabla V^\star(y)| + |\nabla K^\star(x) - \nabla K^\star(y)| \leq C_p |x - y|(1 + \max\{|x|, |y|\}^{p-1}), \quad x, y \in \mathbb{R}^d.$$

(H.4) For the same constant $C_p > 0$, it holds that[16]

$$\|\sigma^\star(x) - \sigma^\star(y)\|_F \leq C_p|x-y|^{1/2}(1 + \max\{|x|,|y|\}^{p/2-1/2}), \quad x, y \in \mathbb{R}^d$$

(H.5) The test functions $(\psi_k)_{1\leq k\leq n} \subset C^2(\mathbb{R}^d \times (0,T))$ are compactly supported and together with the library $\mathbb{L}$ are such that $\mathbf{G}$ has full column rank with[17] $\left\|\mathbf{G}^\dagger\right\|_1 \leq C_{\mathbf{G}}$ almost surely for some constant $C_{\mathbf{G}} > 0$.

(H.6) The true functions $K^\star$, $V^\star$, and $\sigma^\star$ are in the span of $\mathbb{L}$.

We will now define some notation and state some technical lemmas with proofs found in Appendix 4.C. Define the weak-form operator

$$\mathscr{L}(\rho, \psi, \langle\cdot,\cdot\rangle) := \left\langle \partial_t\psi - \nabla\psi\cdot\nabla K^\star * \rho - \nabla\psi\cdot\nabla V^\star + \frac{1}{2}\mathrm{Tr}\left(\nabla^2\psi\sigma^\star(\sigma^\star)^T\right), \rho \right\rangle, \tag{4.4.9}$$

where $\rho = (\rho_t)_{t\leq T}$ is a curve in $\mathcal{P}_p(\mathbb{R}^d)$, ψ is a C^2 function compactly supported over $\mathbb{R}^d \times (0,T)$, and $\langle\cdot,\cdot\rangle$ is an inner product over $\mathbb{R}^d \times (0,T)$. If $\rho = (\mu_t)_{t\leq T}$ is a weak solution to (4.3.1) and $\langle\cdot,\cdot\rangle$ is the $L^2(\mathbb{R}^d)$ inner product then $\mathscr{L}(\rho,\psi,\langle\cdot,\cdot\rangle) = 0$. If instead $\rho = (\mu_t^N)_{t\leq T}$, then by Itô's formula $\mathscr{L}(\rho,\psi,\langle\cdot,\cdot\rangle)$ takes the form of an Itô integral, and we have the following:

Lemma 3. *Under Assumptions (H.1)-(H.5), there exists a constant $C > 0$ independent of N such that*

$$\mathbb{E}\left[\left|\mathscr{L}(\mu^N,\psi,\langle\cdot,\cdot\rangle)\right|\right] \leq \frac{C}{\sqrt{N}}.$$

Proof. See Appendix 4.C. □

With the following lemma, we can relate the histogram U to the empirical measure μ^N through $\mathscr{L}$ using the inner product $\langle\cdot,\cdot\rangle_h$ defined by trapezoidal-rule integration in space and continuous integration in time.

Lemma 4. *Under Assumptions (H.1)-(H.5), for C independent of N and h, it holds that*

$$\mathbb{E}\left[\left|\mathscr{L}(U,\psi,\langle\cdot,\cdot\rangle_h) - \mathscr{L}(\mu^N,\psi,\langle\cdot,\cdot\rangle)\right|\right] \leq Ch.$$

Proof. See Appendix 4.C. □

To incorporate discrete-time effects, we consider the difference between $\mathscr{L}(U,\psi,\langle\cdot,\cdot\rangle_h)$ and $\mathscr{L}(U,\psi,\langle\cdot,\cdot\rangle_{h,\Delta t})$, where recall that $\langle\cdot,\cdot\rangle_{h,\Delta t}$ denotes trapezoidal rule integration in space with meshwidth h and in time with sampling rate Δt.

[16] For $\mathbf{A} \in \mathbb{R}^{d\times d}$ the Frobenius norm is defined $\|\mathbf{A}\|_F = \sqrt{\mathrm{Tr}(\mathbf{A}^T\mathbf{A})}$

[17] $\left\|\mathbf{G}^\dagger\right\|_q$ is the induced matrix q-norm of $\mathbf{G}^\dagger$.

Lemma 5. *Under Assumptions (H.1)-(H.5), for C independent of N, h, and Δt, it holds that*

$$\mathbb{E}\left[\left|\mathscr{L}(U,\psi,\langle\cdot,\cdot\rangle_h) - \mathscr{L}(U,\psi,\langle\cdot,\cdot\rangle_{h,\Delta t})\right|\right] \leq C(h+\Delta t^\eta).$$

Proof. See Appendix 4.C. □

The previous estimates directly lead to the following bound on the model coefficients $\widehat{\mathbf{w}}$:

Theorem 1. *Assume that Assumptions **H** hold. Let $\widehat{\mathbf{w}}$ be the learned model coefficients and $\mathbf{w}^\star$ the true model coefficients. For C independent of N, h, and Δt it holds that*

$$\mathbb{E}\left[\|\widehat{\mathbf{w}} - \mathbf{w}^\star\|_1\right] \leq C\left(h + N^{-1/2} + \Delta t^\eta\right).$$

Proof. Using that $K^\star$, $V^\star$, and $\sigma^\star$ are in the span of $\mathbb{L}$ (H.6), we have that

$$\mathbf{b}_k = \langle \partial_t \psi_k, \mathbf{U}\rangle_{h,\Delta t} = \mathscr{L}(U,\psi_k,\langle\cdot,\cdot\rangle_{h,\Delta t}) + \mathbf{G}_k^T\mathbf{w}^\star := \mathbf{L}_k + \mathbf{G}_k^T\mathbf{w}^\star,$$

where $\mathbf{G}_k^T$ is the kth row of $\mathbf{G}$. From Lemmas 3-5 we have

$$\mathbb{E}\left[|\mathbf{L}_k|\right] \leq \mathbb{E}\left[\left|\mathscr{L}(U,\psi_k,\langle\cdot,\cdot\rangle_{h,\Delta t}) - \mathscr{L}(U,\psi_k,\langle\cdot,\cdot\rangle_h)\right|\right]$$

$$+\mathbb{E}\left[\left|\mathscr{L}(U,\psi_k,\langle\cdot,\cdot\rangle_h) - \mathscr{L}(\mu^N,\psi_k,\langle\cdot,\cdot\rangle)\right|\right] + \mathbb{E}\left[\left|\mathscr{L}(\mu^N,\psi_k,\langle\cdot,\cdot\rangle)\right|\right]$$

$$\leq C'\left(h + N^{-1/2} + \Delta t^\eta\right).$$

Using that $\mathbf{G}$ is full rank, it holds that $\widehat{\mathbf{w}} = \mathbf{G}^\dagger\mathbf{b} = \mathbf{G}^\dagger\mathbf{L} + \mathbf{w}^\star$, hence the result follows from the uniform bound on $\left\|\mathbf{G}^\dagger\right\|_1$ (H.5):

$$\mathbb{E}\left[\|\widehat{\mathbf{w}} - \mathbf{w}^\star\|_1\right] \leq \mathbb{E}\left[\|\mathbf{G}^\dagger\|_1 \|\mathbf{L}\|_1\right] \leq C'C_{\mathbf{G}}\left(h + N^{-1/2} + \Delta t^\eta\right).$$

□ □

Under the assumption (H.6), an immediate corollary is

$$\begin{aligned}\mathbb{E}\Big[\left\|K^\star - \widehat{K}\right\|_{L^2(\mathcal{D}-\mathcal{D})} + \left\|V^\star - \widehat{V}\right\|_{L^2(\mathcal{D})} + \left\|\left\|\sigma^\star(\sigma^\star)^T - \widehat{\sigma}(\widehat{\sigma})^T\right\|_F\right\|_{L^2(\mathcal{D})}\Big] \\ \leq C\left(h + N^{-1/2} + \Delta t^\eta\right),\end{aligned} \tag{4.4.10}$$

This follows from

$$\left\|K^\star - \widehat{K}\right\|_{L^2(\mathcal{D}-\mathcal{D})} \leq \sum_{j=1}^{\mathfrak{J}} |\mathbf{w}_j^\star - \widehat{\mathbf{w}}_j| \, \|K_j\|_{L^2(\mathcal{D}-\mathcal{D})} \leq \left(\sup_j \|K_j\|_{L^2(\mathcal{D}-\mathcal{D})}\right) \|\mathbf{w}^\star - \widehat{\mathbf{w}}\|_1,$$

and similarly for $\widehat{V}$ and $\widehat{\sigma}$. Finally, setting $h = N^{-\alpha}$ for $\alpha > 0$ will ensure convergence as $N \to \infty$

and $\Delta t \to 0$.

4.4.4 Theory vs. Practice

We now make several remarks about the practical performance of Algorithm 4.4.1 with respect to the theoretical convergence of Theorem 1.

Remark 3. An important case of Theorem 1 is $\sigma^\star = 0$, in which case μ_t^N itself is a weak-measure solution to the mean-field equation (4.3.1) and the algorithm returns, for $\eta \geq 2$, $\|\widehat{\mathbf{w}} - \mathbf{w}^\star\|_1 \leq C(h + \Delta t^\eta)$. This partially explains the accuracy observed for purely-extrinsic noise examples in Figures 4.5.4 and 4.5.8. We note further that in the absence of noise ($\varepsilon = 0$ and $\sigma^\star = 0$, not included in this work) Algorithm 4.4.1 recovers systems to high accuracy similarly to WSINDy applied to local dynamical systems [100, 99].

Remark 4. Algorithm 4.4.1 in general implements sparse regression, yet Theorem 1 deals with ordinary least squares. Since least squares is a common subroutine of many sparse regression algorithms (inluding the MSTLS algorithm used here), the result is still relevant to sparse regression. Lastly, the full-rank assumption on $\mathbf{G}$ implies that as $N \to \infty$ sequential thresholding reduces to least squares.

Remark 5. Theorem 1 assumes data from a single experiment ($M = 1$), while the examples below show that $M > 1$ experiments improves results. For any fixed $M > 1$, the $N \to \infty$ limit results in convergence, however, the N-fixed and $M \to \infty$ limit does not result in convergence, as this does not lead to the mean-field equations[18]. The examples below show that using $M > 1$ has a practical advantage, and in Appendix 4.B we demonstrate that even for small particle systems ($N = 10$) the large M regime yields satisfactory results.

Remark 6. Many interesting examples have non-Lipschitz ∇K, in particular a lack of smoothness at $x = 0$. If μ_t^N does not converge to a singular measure as $N \to \infty$, then the bound (4.C.4) holds for ∇K with a jump discontinuity at $x = 0$, where an additional $\mathcal{O}(h)$ term arises from pairwise interactions within an $\mathcal{O}(h)$ distance. The examples below are chosen in part to show that $\mathcal{O}(N^{-1/2})$ convergence holds for ∇K with jumps at the origin.

4.5 Examples

We now demonstrate the successful identification of several particle systems in one and two spatial dimensions as well as the $\mathcal{O}(N^{-1/2})$ convergence predicted in Theorem 1. In each case we use Algorithm 4.4.1 to discover a mean-field equation of the form (4.3.1) from discrete-time particle data. For each dataset we simulate the associated interacting particle system $\mathbf{X}_t$ given by (4.1.1)

[18]Note that the opposite convergence holds for the algorithm introduced in [90]: N-fixed, $M \to \infty$ results in recovery of K.

Variable	Definition	Domain
K	pairwise interaction potential	$L^1_{loc}(\mathbb{R}^d, \mathbb{R})$
V	local potential	$C(\mathbb{R}^d, \mathbb{R})$
σ	diffusivity	$C(\mathbb{R}^d, \mathbb{R}^{d\times d})$
N	number of particles per experiment	$\{2, 3, \dots\}$
d	dimension of latent space	$\mathbb{N}$
T	final time	$(0, \infty)$
$(\Omega, \mathcal{B}, \mathbb{P}, (\mathcal{F}_t)_{t\geq 0})$	filtered probability space	
$(B_t^{(i)})_{i=1}^N$	independent $\mathbb{R}^d$ Brownian motions on $(\Omega, \mathcal{B}, \mathbb{P}, (\mathcal{F}_t)_{t\geq 0})$	
$X_t^{(i)}$	ith particle in the particle system (4.1.1) at time t	$\mathbb{R}^d$
$\mathbf{X}_t$	N-particle system (4.1.1) at time t	$\mathbb{R}^{Nd}$
μ_t^N	empirical measure of $\mathbf{X}_t$	$\mathcal{P}(\mathbb{R}^d)$
F_t^N	distribution of $\mathbf{X}_t$	$\mathcal{P}(\mathbb{R}^{Nd})$
X_t	mean-field process (4.3.2) at time t	$\mathbb{R}^d$
μ_t	distribution of X_t	$\mathcal{P}(\mathbb{R}^d)$
$\mathbf{t}$	L discrete timepoints	$[0, T]$
$\mathbb{X}_\mathbf{t}$	Collection of M independent samples of $\mathbf{X}_t$ at $\mathbf{t}$	$\mathbb{R}^{MLNd}$
$\mathbb{Y}_\mathbf{t}$	Sample of $\mathbf{X_t}$ corrupted with i.i.d. additive noise	$\mathbb{R}^{MLNd}$
U_t	approximate density from particle positions	$\mathcal{P}(\mathbb{R}^d)$
G	density kernel mapping μ_t^N to U_t	$L^1(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R})$
$\mathcal{D}$	spatial support of U_t, $t \in [0, T]$	compact subset of $\mathbb{R}^d$
$\mathbf{C}$	discretization of $\mathcal{D}$	
$\mathbf{U}_t$	discrete approximate density $U_t(\mathbf{C})$	
$\langle \cdot, \cdot \rangle_h$	semi-discrete inner product, trapezoidal rule over $\mathbf{C}$	
$\langle \cdot, \cdot \rangle_{h,\Delta t}$	fully-discrete inner product, trapezoidal rule over $\mathbf{C} \times \mathbf{t}$	
$\mathbb{L}_K$	library of candidate interaction forces	
$\mathbb{L}_V$	library of candidate local forces	
$\mathbb{L}_\sigma$	library of candidate diffusivities	
$\mathbb{L}$	$(\mathbb{L}_K, \mathbb{L}_V, \mathbb{L}_\sigma)$	
Ψ	set of n test functions $(\psi_k)_{k=1}^n$	$C^2(\mathbb{R}^d \times (0, T))$
$\phi_{m,p}(v; \Delta)$	test functions used in this work (equation (4.4.4))	
$\boldsymbol{\lambda}$	set of sparsity thresholds	
$\mathcal{L}$	loss function for sparsity thresholds (equation (4.4.6))	

Table 4.4.1: Notations used throughout.

using the Euler-Maruyama scheme (initial conditions and timestep are given in each example). We assess the ability of WSINDy to select the correct model using the *true positivity ratio*[19]

$$\mathrm{TPR}(\widehat{\mathbf{w}}) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{4.5.1}$$

where TP is the number of correctly identified nonzero coefficients, FN is the number of coefficients falsely identified as zero, and FP is the number of coefficients falsely identified as nonzero [80]. To demonstrate the $\mathcal{O}(N^{-1/2})$ convergence given by (4.4.10), for correctly identified models (i.e. $\mathrm{TPR}(\widehat{\mathbf{w}}) = 1$) we compute the relative ℓ_2-error of the recovered interaction force $\nabla\widehat{K}$, local force $\nabla\widehat{V}$, and diffusivity $\widehat{\sigma}$ over $\mathbf{C} - \mathbf{C}$ and $\mathbf{C}$, respectively, denoting this by $\|\cdot\|$ in the plots below. Results are averaged over 100 trials.

For the computational grid $\mathbf{C}$ we first compute the sample standard deviation s of $\mathbb{Y}$ and we choose $\mathcal{D}$ to be the rectangular grid extending $3s$ from the mean of $\mathbb{Y}$ in each spatial dimension. We then set $\mathbf{C}$ to have 128 points in x and y for $d = 2$ dimensions, and 256 points in x for $d = 1$, noting that these numbers are fairly arbitrary, and used to show that the grid need not be too large. We set the sparsity factors so that $\log_{10}(\boldsymbol{\lambda})$ contains 100 equally spaced points from -4 to 0. More information on the specifications of each example can be found in Appendix 4.A. (MATLAB code used to generate examples is available at https://github.com/MathBioCU/WSINDy_IPS.)

4.5.1 Two-Dimensional Local Model and Homogenization

The first system we examine is a local model ($K^\star(x, y) = 0$) defined by the local potential $V^\star(x, y) = -x - y$ and diffusivity $\sigma^\star(x, y) = \sqrt{2\,(1 + 0.95\cos(\omega x)\cos(\omega y))}\mathbf{I}_2$, where $\mathbf{I}_2$ is the identity in $\mathbf{R}^2$. This results in a constant advection, variable diffusivity mean-field model[20]

$$\partial_t \mu_t = -\partial_x \mu_t - \partial_y \mu_t + \Delta\left[(1 + 0.95\cos(\omega x)\cos(\omega y))\,\mu_t\right]. \tag{4.5.2}$$

The purpose of this example is three-fold. First, we are interested in the ability of Algorithm 4.4.1 to correctly identify a local model from a library containing both local and nonlocal terms. Next, we evaluate whether the $\mathcal{O}(N^{-1/2})$ convergence is realized. Lastly, we investigate whether for large ω the weak-form identifies the associated homogenized equation (derived in Appendix 4.A.1)

$$\partial_t \mu_t = -\partial_x \mu_t - \partial_y \mu_t + \overline{\omega}\Delta\mu_t, \tag{4.5.3}$$

where $\overline{\omega}$ is given by the harmonic mean of diffusivity:

$$\overline{\omega} = \left(\int_{\mathcal{D}} \frac{dxdy}{1 + 0.95\cos(x)\cos(y)}\right)^{-1}.$$

[19] For example, identification of the true model ($\mathrm{supp}\,(\widehat{\mathbf{w}}) = \mathrm{supp}\,(\mathbf{w}^\star)$) results in a $\mathrm{TPR}(\widehat{\mathbf{w}}) = 1$, while identification of only half of the correct nonzero terms and no additional falsely identified terms results in $\mathrm{TPR}(\widehat{\mathbf{w}}) = 0.5$.

[20] Since the model is local, (4.5.2) is the Fokker-Planck equation for the distribution of each particle, rather than only in the limit of infinite particles.

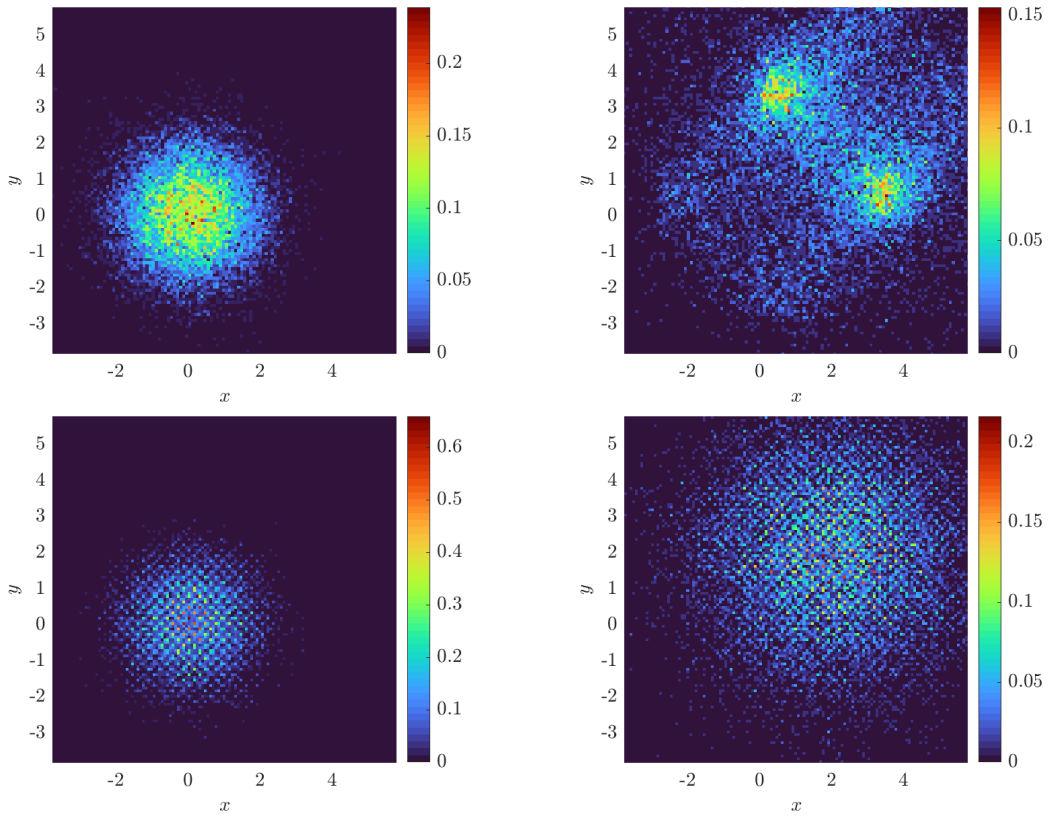


Figure 4.5.1: Snapshots at time $t = 2\Delta t = 0.06$ (left) and $t = 100\Delta t = 2$ (right) of histograms computed with 128 bins in x and y from 16,384 particles evolving under (4.5.2) with $\omega = 1$ (top) and $\omega = 20$ (bottom).

For $\omega \in \{1, 20\}$ we evolve the particles from an initial Gaussian distribution with mean zero and covariance $\mathbf{I}_2$ and record particle positions for 100 timesteps with $\Delta t = 0.02$ (subsampled from a simulation with timestep 10^{-4}). We use a rectangular domain $\mathcal{D}$ of approximate sidelength 10 and compute histograms with 128 bins in x and y for a spatial resolution of $\Delta x \approx 0.078$ (see Figure 4.5.1 for solution snapshots), over which $\overline{\omega} \approx 0.62$. For $\omega = 1$ we compare recovered equations with the full model (4.5.2), while for $\omega = 20$ we compare with (4.5.3), for comparison computing $\overline{\omega}$ over each domain $\mathcal{D}$ using MATLAB's `integral2`. Figure 4.5.2 shows that as the particle number increases, we do in fact recover the desired equations, with TPR($\widehat{\mathbf{w}}$) approaching one as N increases. For $\omega = 1$ we observe $\mathcal{O}(N^{-1/2})$ convergence of the local potential $\widehat{V}$ and the diffusivity $\widehat{\sigma}$. For $\omega = 20$, we observe approximate $\mathcal{O}(N^{-1/2})$ convergence of $\widehat{V}$, and $\widehat{\sigma}$ converging to within 2% of $\sqrt{2\overline{\omega}}$, the homogenized diffusivity (higher accuracy can hardly be expected for $\omega = 20$ since (4.5.3) is itself an approximation in the limit of infinite ω).

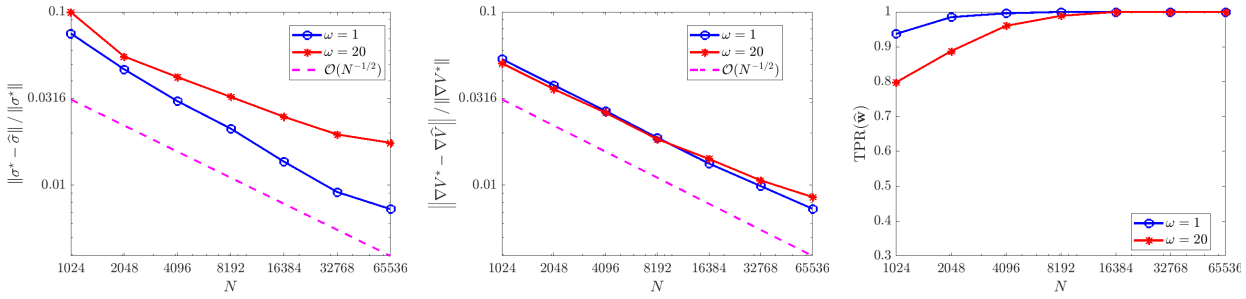


Figure 4.5.2: Convergence of $\widehat{\sigma}$ (left) and $\nabla\widehat{V}$ (middle), recall $\|\cdot\|$ denotes the ℓ_2 norm, for (4.5.2) with $\omega \in \{1, 20\}$, as well as $\mathrm{TPR}(\widehat{\mathbf{w}})$ (right). For $\omega = 1$, results are compared to the exact model (4.5.2), while for $\omega = 20$ results are compared to the homogenized equation (4.5.3).

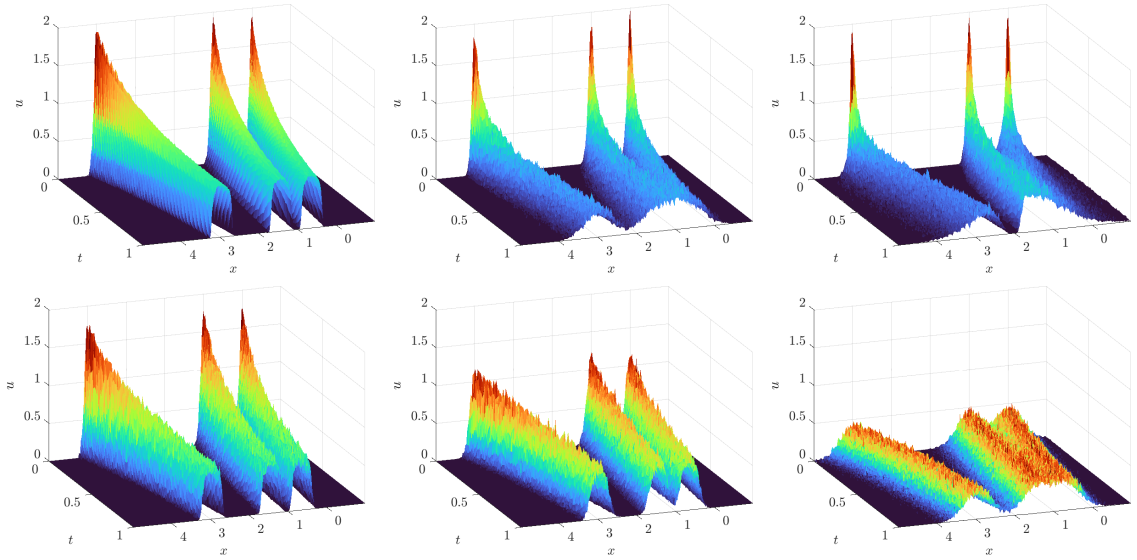


Figure 4.5.3: Histograms computed with 256 bins width $h = 0.0234$ from 8000 particles in 1D evolving under $K^\star = K_{\mathrm{QANR}}(x)$ (4.5.4). Top left to top right: $\sigma^\star(x) = 0$, $\sigma^\star(x) = \sqrt{2(0.1)}$, $\sigma^\star(x) = \sqrt{2(0.1)}|x-2|$. Bottom: deterministic particles with i.i.d. Gaussian noise added to particle positions with resulting noise ratios (left to right) $\epsilon = 0.0316, 0.1, 0.316$.

4.5.2 One-Dimensional Nonlocal Model

We simulate the evolution of particle systems under the quadratic attraction / Newtonian repulsion potential

$$K_{\text{QANR}}(x) = \frac{1}{2}x^2 - |x| \tag{4.5.4}$$

with no external potential ($V = 0$). The $-|x|$ portion of K_{QANR} , leading to a discontinuity in ∇K, is the one-dimensional free-space Green's function for $-\Delta$. For $d \geq 1$, when replaced by the corresponding Green's function in d dimensions, the distribution of particles evolves under K_{QANR} into the characteristic of the unit ball in $\mathbb{R}^d$, which has implications for design and control of autonomous systems [49]. We compare three diffusivity profiles, $\sigma(x) = 0$ corresponding to zero intrinsic noise, $\sigma(x) = \sqrt{2(0.1)}$ leading to constant-diffusivity intrinsic noise, and $\sigma(x) = \sqrt{2(0.1)}|x-2|$ leading to variable-diffusivity intrinsic noise. With zero intrinsic noise ($\sigma(x) = 0$), we examine the effect of extrinsic noise on recovery, and assume uncertainty in the particle positions due to measurement noise at each timestep, $\mathbb{Y} = \mathbb{X} + \varepsilon$, for $\varepsilon \sim \mathcal{N}(0, \epsilon^2 \|\mathbf{X_t}\|^2_{\text{RMS}})$ i.i.d. and $\epsilon \in \{0.01, 0.0316, 0.1, 0.316\}$. In this way ϵ is the *noise ratio*, such that $\|\varepsilon\|_F \,/\, \|\mathbb{X}\|_F \approx \epsilon$ (computed with ε and $\mathbb{X}$ stretched into column vectors).

Measurement data consists of 100 timesteps at resolution $\Delta t = 0.01$, coarsened from simulations with timestep 0.001. Initial particle positions are drawn from a mixture of three Gaussians each with standard deviation 0.005. Histograms are constructed with 256 bins of width $h = 0.0234$. Typical histograms for each noise level are shown in Figure 4.5.3 computed one experiment with $N = 8000$ particles.

For the case of extrinsic noise (Figure 4.5.4), we use only one experiment ($M = 1$) and examine the number of particles N and the noise ratio ϵ. We find that recovery is accurate and reliable for $\epsilon \leq 0.1$, yielding correct identification of K_{QANR} with less than 1% relative error in at least 98/100 trials. Increasing N from 500 to 8000 leads to minor improvements in accuracy for $\epsilon \leq 0.1$, but otherwise has little effect, implying that for low to moderate noise levels the mean-field equations are readily identifiable even from smaller particle systems. For $\epsilon = 10^{-1/2} \approx 0.3162$ (see Figure 4.5.3 (bottom right) for an example histogram), we observe a decrease in $\text{TPR}(\widehat{\mathbf{w}})$ (Figure 4.5.4 middle panel) resulting from the generic identification of a linear diffusion term $\nu\partial_{xx}u$ with $\nu \approx 0.05$. Using that $\sqrt{2\nu} \approx \sqrt{2(0.05)} = \epsilon$, we can identify this as the best-fit *intrinsic* noise model. Furthermore, increases in N lead to reliable identification of the drift term, as measured by $\text{TPR}(\widehat{\mathbf{w}}_{drift})$ (rightmost panel Figure 4.5.4) which is the restriction of TPR to drift terms $\mathbb{L}_K$ and $\mathbb{L}_V$.

For constant diffusivity $\sigma(x) = \sqrt{2(0.1)}$ (Figure 4.5.5), the full model is recovered with less than 3% errors in $\hat{K}$ and $\hat{\sigma}$ in at least 98/100 trials when the total particle count NM is at least 8000, and yields errors less than 1% for $NM \geq 16{,}000$. The error trends for $\hat{K}$ and $\hat{\sigma}$ in this case both strongly agree with the predicted $\mathcal{O}(N^{-1/2})$ rate. For non-constant diffusivity $\sigma(x) = \sqrt{2(0.1)}|x-2|$ (Figure 4.5.6), we also observe robust recovery ($\text{TPR}(\widehat{\mathbf{w}}) \geq 0.95$) for $NM \geq 8000$ with error trends

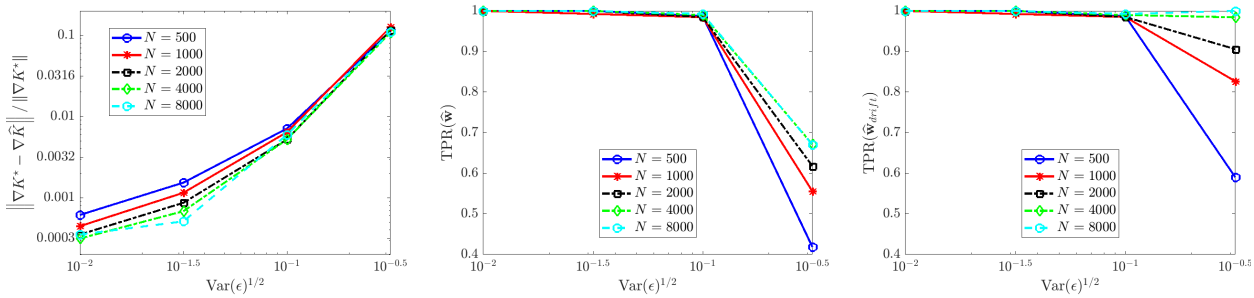


Figure 4.5.4: Recovery of (4.3.1) in one spatial dimension for $K^\star = K_{\mathrm{QANR}}$ and $\sigma^\star = 0$ under different levels of observational noise ϵ. Left: relative error in learned interaction kernel $\widehat{K}$. Middle: true positivity ratio for full model (4.3.1). Right: true positivity ratio for drift term.

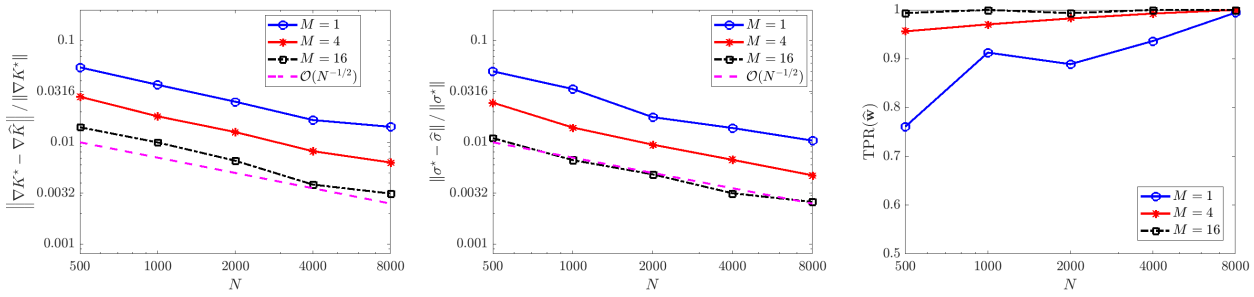


Figure 4.5.5: Recovery of (4.3.1) in one spatial dimension for $K^\star = K_{\mathrm{QANR}}$ and $\sigma^\star = \sqrt{2(0.1)}$

close to $\mathcal{O}(N^{-1/2})$, although the accuracy in $\hat{K}$ and $\hat{\sigma}$ is diminished due to the strong order $\Delta t^{1/2}$ convergence of Euler-Maruyama applied to diffusivities σ that are unbounded in x [103].

4.5.3 Two-Dimensional Nonlocal Model

We now discuss an example of singular interaction in two spatial dimensions using the logarithmic potential

$$K(x) = \frac{1}{2\pi} \log |x| \tag{4.5.5}$$

with constant diffusivity $\sigma(x) = \sigma \in \{0, \frac{1}{\sqrt{4\pi}}\}$. This example corresponds to the parabolic-elliptic Keller-Segel model of chemotaxis, where $\sigma_c := \frac{1}{\sqrt{4\pi}}$ is the critical diffusivity such that $\sigma > \sigma_c$ leads diffusion-dominated spreading of particles throughout the domain (vanishing particle density at every point in $\mathbb{R}^2$) and $\sigma < \sigma_c$ leads to aggregation-dominated concentration of the particle density to the dirac-delta located at the center of mass of the initial particle density [39, 28]. For $\sigma = 0$ we examine the affect of additive i.i.d. measurement noise $\varepsilon \sim \mathcal{N}(0, \epsilon^2 \|\mathbf{X_t}\|^2_{\mathrm{RMS}})$ for $\epsilon \in \{0.01, 0.0316, 0.1, 0.316, 1\}$.

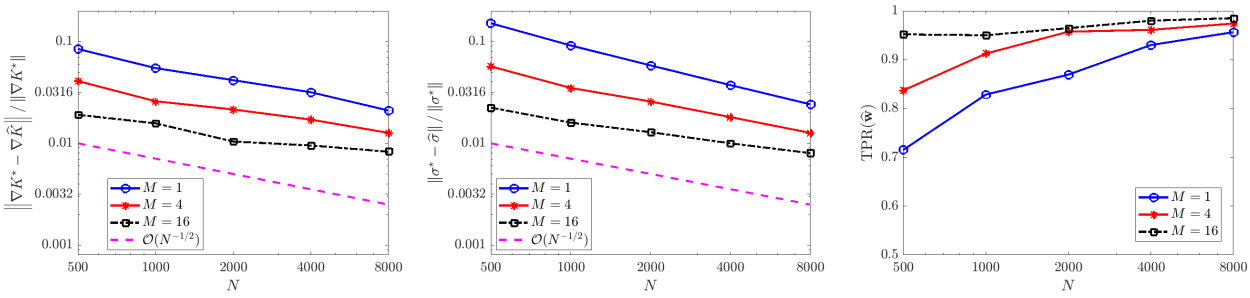


Figure 4.5.6: Recovery of (4.3.1) in one spatial dimension for $K^\star = K_{\text{QANR}}$ and $\sigma^\star = \sqrt{2(0.1)}|x-2|$

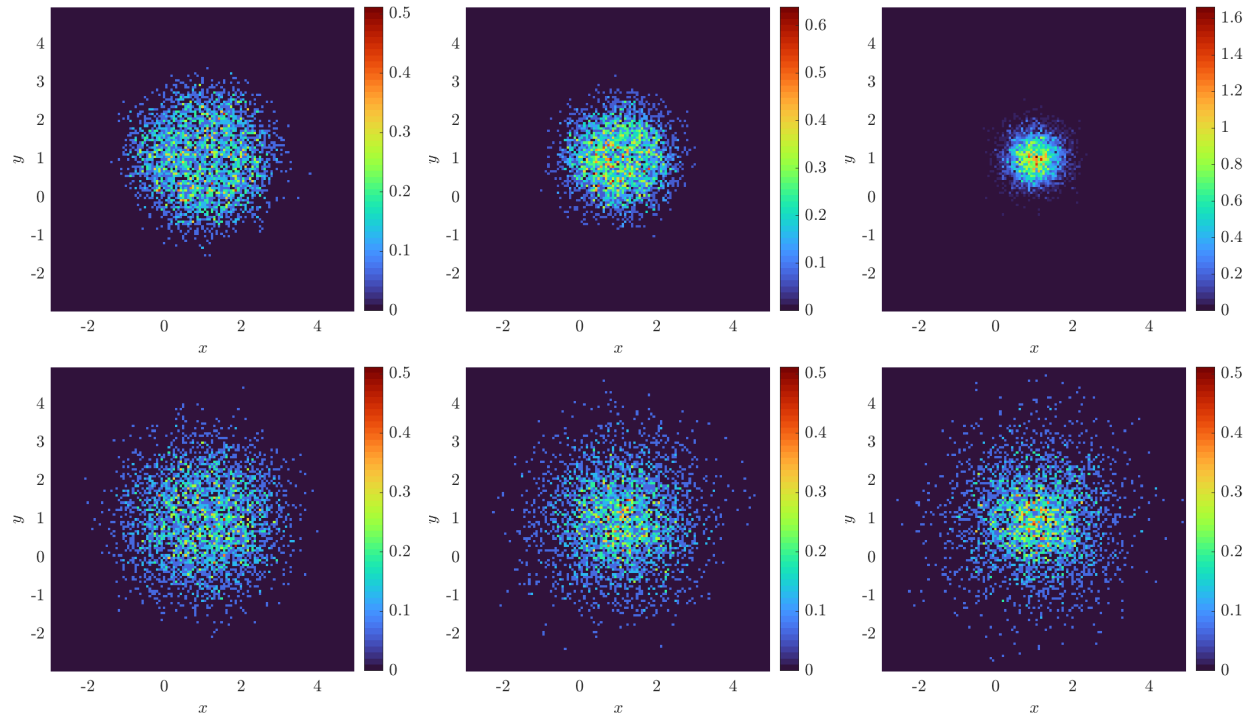


Figure 4.5.7: Histograms created from 4000 particles evolving under logarithmic attraction (equation (4.5.5)) with varying noise levels at times (left to right) $t = 4$, $t = 8$, and $t = 12$. Top: $\epsilon = 0.316$, $\sigma = 0$ (extrinsic only). Bottom: $\epsilon = 0$, $\sigma = (4\pi)^{-1/2} \approx 0.28$ (intrinsic only).

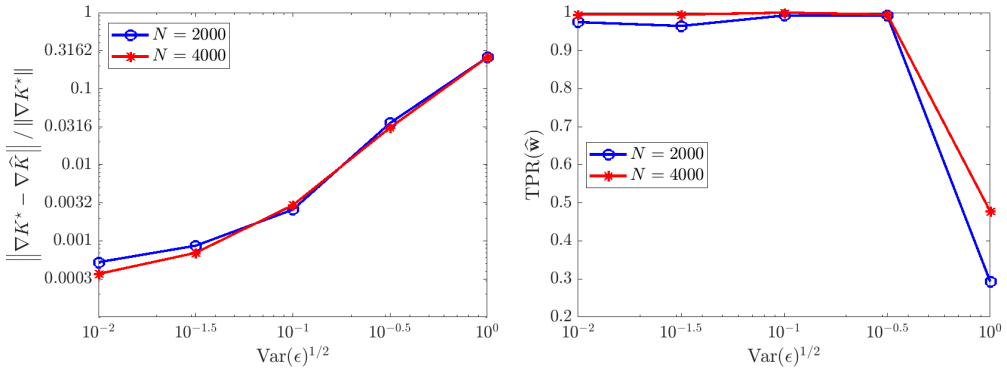


Figure 4.5.8: Recovery of (4.3.1) in two spatial dimensions with $K^\star$ given by (4.5.5) from deterministic particles ($\sigma^\star = 0$) with extrinsic noise ϵ.

We simulate the particle system with a cutoff potential

$$K_\delta(x) = \begin{cases} \dfrac{1}{2\pi}\left(\log(\delta) - 1 + \dfrac{|x|}{\delta}\right), & |x| < \delta \\ \dfrac{1}{2\pi}\log|x|, & |x| \geq \delta \end{cases} \tag{4.5.6}$$

with $\delta = 0.01$, so that K_δ is Lipschitz and ∇K_δ has a jump discontinuity at the origin. Initial particle positions are uniformly distributed on a disk of radius 2 and the particle position data consists of 81 timepoints recorded at a resolution $\Delta t = 0.1$, coarsened from 0.0025. Histograms are created with 128×128 bins in x and y of sidelength $h = 0.0469$ (see Figure 4.5.7 for histogram snapshots over time). We examine $M = 2^0, \ldots, 2^6$ experiments with $N = 2000$ or $N = 4000$ particles.

In Figure 4.5.8 we observe a similar trend in the $\sigma = 0$ case as in the 1D nonlocal example, namely that recovery for $\epsilon \leq 0.1$ is robust with low errors in $\widehat{K}$ (on the order of 0.0032), only in this case the full model is robustly recovered up to $\epsilon = 0.316$. At $\epsilon = 1$, with $N = 4000$ the method frequently identifies a diffusion term $\nu\Delta u$ with $\nu \approx 0.5 = \epsilon^2/2$, and for $N = 2000$ the method occasionally identifies the backwards diffusion equation $\partial_t \mu_t = -\alpha\Delta\mu_t$, $\alpha > 0$. This is easily prevented by enforcing positivity of σ, however we leave this and other constraints as an extension for future work.

With diffusivity $\sigma = \frac{1}{\sqrt{4\pi}}$, we obtain $\mathrm{TPR}(\widehat{\mathbf{w}})$ approximately greater than 0.95 for $NM \geq 16,000$ (Figure 4.5.9, right), with an error trend in $\widehat{K}$ following an $\mathcal{O}(N^{-1/2})$ rate, and a trend in $\widehat{\sigma}$ of roughly $\mathcal{O}(N^{-2/3})$. Since convergence in M for any fixed N is not covered by the theorem above, this shows that combining multiple experiments may yield similar accuracy trends for moderately-sized particle systems.

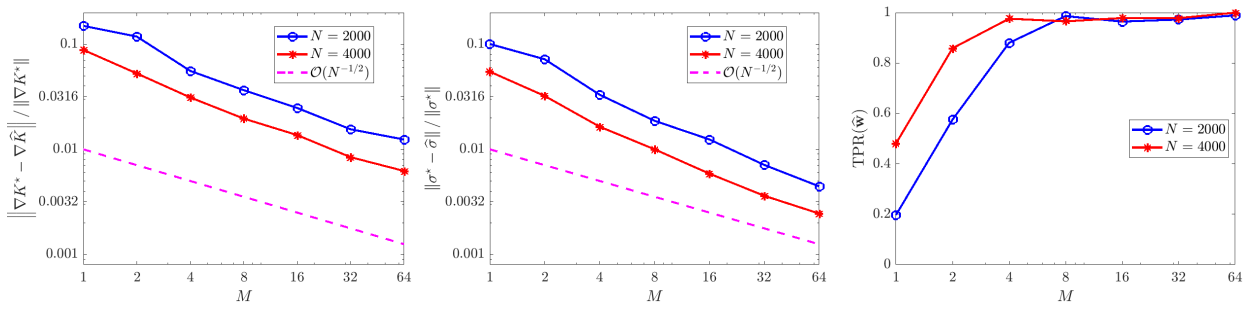


Figure 4.5.9: Recovery of (4.3.1) in two spatial dimensions with $K^\star$ given by (4.5.5) and $\sigma^\star = \frac{1}{\sqrt{4\pi}}$.

4.6 Discussion

We have developed a weak-form method for sparse identification of governing equations for interacting particle systems using the formalism of mean-field equations. In particular, we have investigating two lines of inquiry, (1) is the mean-field setting applicable for inference from medium-size batches of particles? And (2) can a low-cost, low-regularity density approximation such as a histogram be used to enforce weak-form agreement with the mean-field PDE? We have demonstrated on several examples that the answer is yes to both questions, despite the fact that the mean-field equations are only valid in the limit of infinitely many particles ($N \to \infty$). This framework is suitable for systems of several thousand particles in one and two spatial dimensions, and we have proved convergence in N for the associated least-squares problem using simple histograms as approximate particle densities. In addition, the sparse regression approach allows one to identify the full system, including interaction potential K, local potential V, and diffusivity σ.

It was initially unclear whether the mean-field setting could be utilized in weak form for finite particle batches, hence this can be seen as a proof of concept for particle systems with N in the range $10^3 - 10^5$. With convergence in N and low computational complexity, our weak-form approach is well-suited *as is* for much larger particle systems. In the opposite regime, for small fixed N, the authors of [90] show that their maximum likelihood-based method converges as $M \to \infty$ (i.e. in the limit of infinite experiments). While the same convergence does not hold for our weak-form method, the results in Section 4.5 suggest that in practice, combining M independent experiments each with N particles improves results. Furthermore, we include evidence in Appendix 4.B that even for small N, our method correctly identifies the mean-field model when M is large enough, with performance similar to that in [90]. We leave a full investigation of the interplay between M and N to future work.

In the operable regime of $N > 10^3$, there is potential for improvements and extensions in many directions. On the subject of density estimation, histograms are highly efficient, yet they lead to piecewise-constant approximations of μ_t and hence $\mathcal{O}(h)$ errors. Choosing a density kernel G to achieve high-accuracy quadrature without sacrificing the $\mathcal{O}(N)$ runtime of histogram computation

seems prudent, although one must be cautious about making assumptions on the smoothness of mean-field distribution μ_t. For instance, in the 1D nonlocal example 4.5.2, discontinuities develop in μ_t for the case $\sigma = 0$, hence a histogram approximation is more appropriate than using e.g. a Gaussian kernel.

The computational grid $\mathbf{C}$, quadrature method $\langle \cdot, \cdot \rangle_{h,\Delta t}$, and reference test function ψ may also be optimized further or adapted to specific problems. The approach chosen here of $\mathbf{C}$ equally-spaced and separable piecewise-polynomial ψ, along with intergration using the trapezoidal quadrature, has several advantages, including high accuracy and fast computation using convolutions. However, this may need adjustment for higher dimensions. It might be advantageous to adapt $\mathbf{C}$ to the data $\mathbb{Y}$, however this may prevent one from evaluating $(\mathbf{G}, \mathbf{b})$ using the FFT if a non-uniform grid results, hence increases the overall computational complexity. One could also use multiple reference test functions ψ. The possibilities of varying the test functions (within the smoothness requirements of the library $\mathbb{L}$) has been largely unexplored in weak-form identification methods.

Several theoretical questions remain unanswered, namely model recovery statistics for finite N. As a consequence of Theorem 1, as well as convergence results on sequential thresholding [158], we have that $\mathbf{G}$ being full-rank and $\mathbb{L}$ containing the true model is sufficient to guarantee convergence $\widehat{\mathbf{w}} \to \mathbf{w}^\star$ as $N \to \infty$ at the rate $\mathcal{O}(N^{-1/2})$. Noise, whether extrinsic or intrinsic, for finite N may result in identification of an incorrect model when $\mathbf{G}$ is poorly-conditioned. The effect is more severe if the true model has a small coefficient, which requires a small threshold λ, which correspondingly may lead to a non-sparse solution. These are sensitivities of any sparse regression algorithm (see e.g. [24]) and accounting for the effect of noise and poor conditioning is an active area of research in equation discovery.

We also note that several researchers have focused on the uniqueness in kernel identifiability [81, 84]. This issue does not directly apply to our scenario[21] of identifying the triple (K, V, σ). Moreover, in the cases we considered, we do not see any identifiability issues (e.g. rank deficiency) even in the high noise case with low particle number. Quantifying the transition to identifiability as $N \to \infty$ as a function of the condition number $\kappa(\mathbf{G})$ is an important subject for future work.

For extensions, the example system (4.5.2) and resulting homogenization motivates further study of effective equations for systems with complex microstructure. In other fields this is described as *coarse-graining*. A related line of study is inference of 2nd-order particle systems, as explored in [131], which often lead to an infinite hierachy of mean-field equations. Our weak-form approach may provide a principled method for truncated and closing such hierarchies using particle data. Another extension is to enforce convex constraints in the regression problem, such as lower bounds on diffusivity, or K with long-range attraction depending on the distribution $\rho_{rr} \in \mathcal{P}([0, \infty))$ of pairwise distances (see [90] for further use of ρ_{rr}). Finally, the framework we've introduced can easily be used to find nonlocal models from continuous solution data (e.g. given $\mathbf{U}$ instead of $\mathbb{Y}$), whereby questions of nonlocal representations of models can be investigated.

[21] E.g. due to multiple representations of the drift combining both nonlocal and local terms - see Section 4.4.1

Lastly, we note that MATLAB code is available at https://github.com/MathBioCU/WSINDy_IPS.

4.7 Acknowledgements

This research was supported in part by the NSF Mathematical Biology MODULUS grant 2054085, in part by the NSF/NIH Joint DMS/NIGMS Mathematical Biology Initiative grant R01GM126559, and in part by the NSF Computing and Communications Foundations grant 1815983. This work also utilized resources from the University of Colorado Boulder Research Computing Group, which is supported by the National Science Foundation (awards ACI-1532235 and ACI-1532236), the University of Colorado Boulder, and Colorado State University. The authors would also like to thank Prof. Vanja Dukić (University of Colorado at Boulder, Department of Applied Mathematics) for insightful discussions and helpful suggestions of references.

Appendix

4.A Specifications for Examples

In Tables 4.A.1 - 4.A.4 we include hyperparameter specifications and resulting attributes of Algorithm 4.4.1 applied to the three examples in Section 4.5. In particular, we report the typical walltime in Table 4.A.4, showing that on each example Algorithm 4.4.1 learns the mean-field equation from a dataset with $\sim 64,000$ particles in under 10 seconds.

4.A.1 Derivation of homogenized equation (4.5.3)

We briefly provide a derivation of the homogenized equation (4.5.3) in the static case. Let $\Omega \in \mathbb{R}^d$ be an open bounded domain with smooth boundary and $\mathbb{T}^d$ be the d-dimensional torus. Let $a(x, y) : \Omega \times \mathbb{T}^d \to \mathbb{R}$ be continuous and uniformly bounded below,

$$a(x, y) \geq \alpha > 0, \quad (x, y) \in \Omega \times \mathbb{T}^d.$$

Then for any $f \in L^2(\Omega)$, the equation

$$-\Delta \left(a(x, x/\epsilon) u^\epsilon(x) \right) = f(x), \qquad u^\epsilon\big|_{\partial\Omega} = 0$$

has a unique weak solution $u^\epsilon \in L^2(\Omega)$ given by

$$u^\epsilon(x) = \frac{(Gf)(x)}{a(x, x/\epsilon)},$$

Mean-field Term	Trial Function Library
$\nabla \cdot (U \nabla K * U)$	$\nabla \cdot (U \nabla \lvert x\rvert^m * U)$, $m \in \{1, 2, 3, 4, 5, 6, 7\}$
$\nabla \cdot (U \nabla V)$	$\partial_{x_i} (U \cos(mx_1) \cos(nx_2))$, $(m, n) \in \{0, 1, 2, 3, 4, 5\}$, $i \in \{1, 2\}$
$\frac{1}{2} \sum_{i,j=1}^d \frac{\partial^2 (U\sigma\sigma^T)_{ij}}{\partial x_i \partial x_j}$	$\Delta(U \cos(mx_1) \cos(nx_2))$, $(m, n) \in \{0, 1, 2, 3, 4, 5\}$

Table 4.A.1: Trial function library for local 2D example (Section 4.5.1).

Mean-field Term	Trial Function Library
$\nabla \cdot (U \nabla K * U)$	$\partial_x \cdot (U \partial_x \lvert x \rvert^m * U)$, $m \in \{1,2,3,4,5,6,7\}$
$\nabla \cdot (U \nabla V)$	$\partial_x (U x^m)$, $m \in \{0,2,3,4,5,6,7,8\}$
$\frac{1}{2}\sum_{i,j=1}^{d} \frac{\partial^2 (U\sigma\sigma^T)_{ij}}{\partial x_i \partial x_j}$	$\partial_{xx}(U x^m)$, $m \in \{0,1,2,3,4,5,6,7,8\}$

Table 4.A.2: Trial function library for nonlocal 1D example (Section 4.5.2).

Mean-field Term	Trial Function Library
$\nabla \cdot (U \nabla K * U)$	$\begin{cases} \nabla \cdot (U \nabla \lvert x \rvert^m * U) & m \in \{2,3,4,5,6\} \\ \nabla \cdot (U \nabla \left[\lvert x \rvert^{1/2} \right]_\delta * U) & \\ \nabla \cdot (U \nabla \left[\lvert x \rvert (\log \lvert x \rvert - 1) \right]_\delta * U) & \\ \nabla \cdot (U \nabla \left[\log \lvert x \rvert \right]_\delta * U) & \end{cases}$
$\nabla \cdot (U \nabla V)$	$\partial_{x_i} (U x_1^m x_2^n) \qquad 0 \le m+n \le 5,\ i \in \{1,2\}$
$\frac{1}{2}\sum_{i,j=1}^{d} \frac{\partial^2 (U\sigma\sigma^T)_{ij}}{\partial x_i \partial x_j}$	$\Delta(U \cos(m x_1) \cos(n x_2))$, $(m,n) \in \{0,1,2\}$

Table 4.A.3: Trial function library for nonlocal 2D example (Section 4.5.3). Interaction potentials $[\, K \,]_\delta$ indicate cutoff potentials of the form (4.5.6) with $\delta = 0.01$ such that the resulting potential is Lipschitz.

Example	(m_x, m_t)	(p_x, p_t)	(s_x, s_t)	size($\mathbf{U}$)	$(h, \Delta t)$
Local 2D	(31,16)	(5,3)	(10,5)	$128 \times 128 \times 101$	$(0.078, 0.02)$
Nonlocal 1D	(29,8)	(5,3)	(5,1)	256×101	$(0.023, 0.01)$
Nonlocal 2D	(25,8)	(5,3)	(8,1)	$128 \times 128 \times 81$	$(0.047, 0.1)$

Example	$\lVert \mathbf{w}^\star \rVert_0$	size($\mathbf{G}$)	$\lVert \mathbf{G}^\dagger \rVert_1$	$\kappa_2(\mathbf{G})$	Walltime
Local 2D	$\{4,3\}$	686×85	2.0×10^3	3.0×10^7	9.2s
Nonlocal 1D	$\{2,3,5\}$	3400×24	1.3×10^5	8.7×10^8	0.7s
Nonlocal 2D	$\{1,2\}$	6500×59	1.1×10^4	6.4×10^6	8.5s

Table 4.A.4: Discretization parameters and general information for examples. The number of nonzeros in the true weight vector $\lVert \mathbf{w}^\star \rVert_0$ is given for each parameter set examined. Namely, for the local 2D example, $\omega = 1$ results in a 4-term model, while the homogenized case $\omega = 20$ results in a three-term model. For the nonlocal 1D example, $\sigma \in \{0, \sqrt{2(0.1)}, \sqrt{2(0.1)}|x-2|\}$ result in 2-term, 3-term, and 5-term models, respectively, and for the nonlocal 2D example $\sigma \in \{0, (4\pi)^{-1}\}$ results in 1-term and 2-term models. The norm $\left\lVert \mathbf{G}^\dagger \right\rVert_1$, condition number $\kappa_2(\mathbf{G})$ and walltime are listed for representative samples with 64,000 total particles.

where G is the Green's function for $(-\Delta)^{-1}$ with homogenenous Dirichlet boundary conditions on $\partial\Omega$. By the coercivity of a we have that $\|u^\epsilon\|_{L^2(\Omega)}$ is uniformly bounded in ϵ. By the lemma in [147, Section 2.4], up to a subsequence $\{\epsilon_j\}_{j\in\mathbb{N}}$, there exists a function $u(x,y)$ periodic in its second variable such that for any continuous function $\phi(x,x/\epsilon)$, we have

$$\lim_{\epsilon\to 0}\int u^\epsilon(x)\phi(x,x/\epsilon)dx = \iint u(x,y)\phi(x,y)dydx.$$

Setting $\phi(x,y)=\phi(x)$, we see that on the same subsequence, $u^\epsilon \rightharpoonup \int u(x,y)dy$. Applying the same lemma to the constant series $u^\epsilon = 1$ and letting $\phi(x,x/\epsilon)=\phi(x)a^{-1}(x,x/\epsilon)$, we see that (up to possibly a second subsequence),

$$a^{-1}(x,x/\epsilon) \rightharpoonup \int \frac{dy}{a(x,y)}.$$

Letting $a^*(x) := \left(\int \frac{dy}{a(x,y)}\right)^{-1}$ and putting together the previous limits, we see that

$$u^\epsilon(x) \rightharpoonup u^*(x) := \int u(x,y)dy = (Gf)(x)\int \frac{dy}{a(x,y)} =: \frac{(Gf)(x)}{a^*(x)},$$

and hence u^* solves the homogenized equation

$$\Delta\left(a^*u^*\right) = f.$$

4.B Recovery for small N and large M

The related maximum-likelihood approach [90] is shown to be suitable for small N and large M, hence a natural line of inquiry is the performance of Algorithm 4.4.1 in this regime. Theorem 1 does not apply to this regime, and in fact convergence of the algorithm is not expected: letting $U_t^{M,N} = \frac{1}{M}\sum_{m=1}^M U_t^{(m),N}$ where $U_t^{(m),N}$ is the approximate density constructed from experiment m with N particles, we have the weak-measure convergence $U_t^{M,N} \to \rho_t^{(1),N}$ as $M\to\infty$, where $\rho_t^{(1),N}$ is the 1-particle marginal of the distribution of $\mathbf{X}_t$ in $\mathbb{R}^{Nd}$. Unlike the mean-field distribution μ_t, $\rho_t^{(1),N}$ is not a weak solution to the mean-field Fokker-Planck equation (4.3.1), instead we have

$$\partial_t\rho_t^{(1),N} = \frac{N-1}{N}\nabla\cdot\int_{\mathbb{R}^d}\nabla K(x-y)\rho_t^{(2),N}(x,y)dy + \nabla\cdot\left(\nabla V\rho_t^{(1),N}\right) + \frac{1}{2}\sum_{i,j=1}^d \partial_{x_ix_j}(\sigma\sigma^T\rho_t^{(1),N}),$$

holding weakly, which depends on the 2-particle marginal $\rho_t^{(2),N}$ [65]. Nevertheless, using the 1D nonlocal example in Section 4.5.2 with $\sigma=\sqrt{2(0.1)}\approx 0.45$, we observe in Figure 4.B.1 (right panel) that our weak-form algorithm correctly identifies the model in $> 96\%$ of trials with just $N=10$ particles per experiment when $M\in[2^{10},2^{12}]$, and that error in K (left panel) follows a $\mathcal{O}(M^{-1/2})$

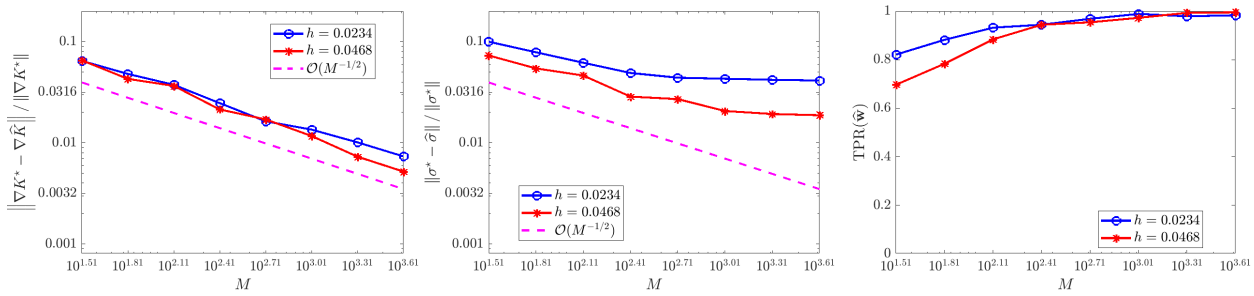


Figure 4.B.1: Recovery of (4.3.1) in one spatial dimension for $K^\star = K_{\text{QANR}}$ and $\sigma^\star = \sqrt{2(0.1)}$ with only $N = 10$ particles per experiment.

---

trend. At $M = 4096 \approx 10^{3.61}$ experiments the error[22] in K is less than 1% and the runtime is approximately 0.9s. The lack of convergence in M is reflected in the diffusivity (middle panel of Figure 4.B.1), where the error appears to plateau at around 1.7% for $h \approx 0.0468$ and at 3.5% for $h \approx 0.0234$. The lower resolution (larger binwidth h) appears to yield slightly better results, possibly indicating that larger h produces a coarse-graining effect such that $\rho^{(2),N} \approx \rho^{(1),N} \otimes \rho^{(1),N}$ over larger distances, although this effect deserves more thorough study in future work.

4.C Technical Lemmas

We now prove Lemmas 3-5 under Assumption **H**. First, some consequences of Assumption **H**. (I) The η-Hölder continuity of sample paths (H.1) implies that for each $t \in [0, T]$,

$$\int_{\mathbb{R}^d} |x|^p d\mu_t^N = \frac{1}{N} \sum_{i=1}^N |X_t^{(i)}|^p \leq \frac{2^p}{N} \sum_{i=1}^N |X_0^{(i)}|^p + C_\eta 2^p t^{p\eta}.$$

Together with the pth moment bound on μ_0 (H.2), this implies

$$\mathbb{E}\left[\sup_{t \leq T} \int_{\mathbb{R}^d} |x|^p d\mu_t^N\right] \leq 2^p (M_p + C_\eta T^{p\eta}), \tag{4.C.1}$$

independent of N.

(II) The growth bounds on $\nabla K^\star$, $\nabla V^\star$, and $\sigma^\star$ (H.3)-(H.4) imply that for some $C > 0$,

$$|\nabla K^\star(x)| + |\nabla V^\star(x)| + \left\|\sigma^\star(x)(\sigma^\star(x))^T\right\|_F \leq C(1 + |x|^p), \tag{4.C.2}$$

where $\|\cdot\|_F$ is the Frobenius norm.

[22] For comparison, in [90] Fig. 4 the error in recovering K using the maximum-likelihood approach on an opinion dynamics example for $M = 10^{3.6}$, $N = 10$, and $\sigma = 0.5$ is approximately $100 \times 10^{-1.2}\% = 6.3\%$.

Proof of Lemma 3. Applying Itô's formula to the process $\frac{1}{N}\sum_{i=1}^N \psi(X^{(i)},t)$, we get that

$$\mathscr{L}(\mu^N,\psi,\langle\cdot,\cdot\rangle) = \frac{1}{N}\sum_{i=1}^N \int_0^T \nabla\psi(X_t^{(i)},t)^T\sigma^\star(X_t^{(i)})dB_t^{(i)}.$$

Note that each integral on the right-hand side is a local martingale, since (4.C.2) and (H.5) ensures boundedness of $\nabla\psi(x,t)^T\sigma^\star(x)$ over any compact set in $\mathbb{R}^d$, hence has mean zero. By independence of the Brownian motions $B_t^{(i)}$, exchangeability of $X_t^{(i)}$, the moment bound (4.C.1), and the growth bounds on σ (H.4), the Itô isometry gives us

$$\begin{aligned}
\mathbb{E}\left[\mathscr{L}(\mu^N,\psi,\langle\cdot,\cdot\rangle)^2\right] &= \frac{1}{N}\int_0^T \mathbb{E}_{X\sim\rho_t^{(1)}}\left[\left|\nabla\psi(X,t)^T\sigma^\star(X)\right|^2\right]dt \\
&= \frac{1}{N}\int_0^T \mathbb{E}\left[\int_{\mathbb{R}^d}\left|\nabla\psi(x,t)^T\sigma^\star(x)\right|^2 d\mu_t^N(x)\right]dt \\
&\le \frac{C'}{N}\|\nabla\psi\|_{2,\infty}^2\int_0^T \mathbb{E}\left[1+\int_{\mathbb{R}^d}|x|^p d\mu_t^N(x)\right]dt \\
&\le CN^{-1}
\end{aligned}$$

where C depends on M_p, C_p, T, and ψ. The result follows from Jensen's inequality[23]. □

Proof of Lemma 4. Using the notation $f^{\mathbf{C}}$ from Lemma 2 to denote piecewise constant approximation of a function f over the domain $\mathcal{D}$ using the grid $\mathbf{C}$, we have

$$\begin{aligned}
\mathscr{L}(U,\psi,\langle\cdot,\cdot\rangle_h) - \mathscr{L}(\mu^N,\psi,\langle\cdot,\cdot\rangle) &= -\underbrace{\left(\left\langle(\nabla\psi\cdot((\nabla K^\star)^{\mathbf{C}}*\mu^N))^{\mathbf{C}},\mu^N\right\rangle - \left\langle\nabla\psi\cdot\nabla K^\star*\mu^N,\mu^N\right\rangle\right)}_{E_{\text{interact}}} \\
&\quad + \left\langle\partial_t\psi^{\mathbf{C}} - \partial_t\psi,\mu^N\right\rangle - \left\langle((\nabla\psi\cdot\nabla V^\star)^{\mathbf{C}} - \nabla\psi\cdot\nabla V^\star,\mu^N\right\rangle \\
&\quad + \frac{1}{2}\left\langle\mathrm{Tr}\left(\nabla^2\psi\sigma^\star(\sigma^\star)^T\right)^{\mathbf{C}} - \mathrm{Tr}\left(\nabla^2\psi\sigma^\star(\sigma^\star)^T\right),\mu^N\right\rangle \\
&= E_{\text{interact}} + E_{\text{linear}}.
\end{aligned}$$

The right-hand side includes an interaction error E_{interact} followed by a sum E_{linear} of terms that are linear in the difference between a locally Lipschitz function and its piecewise constant approximation. Hence, we can bound E_{linear} using smoothness of ψ (H.5), the moment assumptions on μ_t^N (H.2), and the growth assumptions on V and σ (H.3)-(H.4). Specifically, for $x\in B_k$ with center $\mathbf{c}_k$, the growth assumptions imply

$$|\nabla\psi(x)\cdot\nabla V^\star(x) - \nabla\psi(\mathbf{c}_k)\cdot\nabla V^\star(\mathbf{c}_k)| \le Ch\Big((\|\nabla\psi\|_{2,\infty} + \mathrm{Lip}(\nabla\psi))(1+|x|^p)\Big)$$

$$|\mathrm{Tr}\left(\nabla^2\psi(x)\sigma^\star(x)(\sigma^\star(x))^T\right) - \mathrm{Tr}\left(\nabla^2\psi(\mathbf{c}_k)\sigma^\star(\mathbf{c}_k)(\sigma^\star(\mathbf{c}_k))^T\right)| \le C'h\Big((\left\|\nabla^2\psi\right\|_{F,\infty} + \mathrm{Lip}(\nabla^2\psi))(1+|x|^p)\Big)$$

[23] $\|f\|_{p,q}$ for vector-valued functions $f:\mathbb{R}^d\to\mathbb{R}^d$ denotes the L^q norm over x of the ℓ^p norm of $f(x)$. Also recall that $\rho_t^{(1)}$ is the $X_t^{(1)}$-marginal of the process $\mathbf{X}_t\in\mathbb{R}^{dN}$.

for C and C' depending on p, d, and C_p, hence

$$|E_{\text{linear}}| \leq C'' \sup_{|\alpha|\leq 2} \text{Lip}(\partial^\alpha \psi) \left(T + \int_0^T \int_{\mathbb{R}^d} |x|^p d\mu_t^N dt \right) h. \tag{4.C.3}$$

Similarly, for the interaction error we use that for $x \in B_k$ and $y \in B_j$ with centers $\mathbf{c}_k$ and $\mathbf{c}_j$, we have

$$\begin{aligned} |\nabla\psi(\mathbf{c}_k) \cdot \nabla K^\star(\mathbf{c}_k - \mathbf{c}_j) - \nabla\psi(x) \cdot \nabla K^\star(x-y)| &\leq |\nabla\psi(\mathbf{c}_k)|\,|\nabla K^\star(\mathbf{c}_k - \mathbf{c}_j) - \nabla K^\star(x-y)| \\ &\quad + |\nabla\psi(\mathbf{c}_k) - \nabla\psi(x)|\,|\nabla K^\star(x-y)| \\ &\leq C''' h \left(\|\nabla\psi\|_{2,\infty} + \text{Lip}(\nabla\psi) \right) (1 + |x-y|^p) \end{aligned}$$

with C''' also depending on p, d, and C_p. From this we have

$$|E_{\text{interact}}| \leq C'''' \left(T + \int_0^T \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} |x-y|^p d\mu_t^N(y) d\mu_t^N(x) dt \right) h. \tag{4.C.4}$$

The result follows from taking expectation and using the moment bound (4.C.1), where the final constant C depends on p, d, C_p, M_p, T, η, and ψ.

□

Proof of Lemma 5. Again rewriting the spatial trapezoidal-rule integration in the form $\int_{\mathbb{R}^d} \varphi^{\mathbf{C}}(x) d\mu_t^N$, we see that

$$\mathscr{L}(U, \psi, \langle \cdot, \cdot \rangle_h) - \mathscr{L}(U, \psi, \langle \cdot, \cdot \rangle_{h,\Delta t}) \tag{4.C.5}$$

reduces to four terms of the form

$$A(\varphi) := \frac{1}{N} \sum_{i=1}^N \left(\int_0^T \varphi^{\mathbf{C}}(X_t^{(i)}) dt - \frac{\Delta t}{2} \sum_{\ell=1}^L \left(\varphi^{\mathbf{C}}(X_{t_{\ell+1}}^{(i)}) + \varphi^{\mathbf{C}}(X_{t_\ell}^{(i)}) \right) \right),$$

for $\varphi \in \left\{ \partial_t\psi, \nabla\psi \cdot \nabla V^\star, \text{Tr}(\nabla^2\psi\sigma^\star(\sigma^\star)^T), \nabla\psi \cdot \nabla K^\star * \mu_t^N \right\}$. Similarly to the bounds derived for $|\varphi(x) - \varphi^{\mathbf{C}}(x)|$ in Lemma 4, the growth bounds on $V^\star, K^\star$ and $\sigma^\star$ imply in general that

$$|\varphi(x) - \varphi(y)| \leq C|x-y| \left(1 + \max\{|x|, |y|\}^p\right).$$

Rewriting the summands in $A(\varphi)$,

$$\int_0^T \varphi^{\mathbf{C}}(X_t^{(i)}) dt - \frac{\Delta t}{2} \sum_{\ell=1}^L \left(\varphi^{\mathbf{C}}(X_{t_{\ell+1}}^{(i)}) + \varphi^{\mathbf{C}}(X_{t_\ell}^{(i)}) \right)$$

$$= \sum_{\ell=1}^L \underbrace{\int_{t_\ell}^{t_{\ell+1}} \left(\frac{t - t_\ell}{\Delta t} \right) (\varphi^{\mathbf{C}}(X_t^{(i)}) - \varphi^{\mathbf{C}}(X_{t_{\ell+1}}^{(i)})) dt}_{I_1} + \underbrace{\int_{t_\ell}^{t_{\ell+1}} \left(\frac{t_{\ell+1} - t}{\Delta t} \right) (\varphi^{\mathbf{C}}(X_t^{(i)}) - \varphi^{\mathbf{C}}(X_{t_\ell}^{(i)})) dt}_{I_2},$$

and using

$$|\varphi^{\mathbf{C}}(x)-\varphi^{\mathbf{C}}(y)| \le |\varphi(x)-\varphi(\mathbf{c}_k)|+|\varphi(x)-\varphi(y)|+|\varphi(y)-\varphi(\mathbf{c}_\ell)| \le C(2h+|x-y|)(1+\max\{|x|,|y|\}^p)$$

where $x \in B_k$ and $y \in B_\ell$, we see that for I_1,

$$\left|\int_{t_\ell}^{t_{\ell+1}} \left(\frac{t-t_\ell}{\Delta t}\right)(\varphi^{\mathbf{C}}(X_t^{(i)}) - \varphi^{\mathbf{C}}(X_{t_{\ell+1}}^{(i)}))dt\right|$$

$$\le \int_{t_\ell}^{t_{\ell+1}} \left(\frac{t-t_\ell}{\Delta t}\right) C(2h + |X_t^{(i)} - X_{t_{\ell+1}}^{(i)}|)(1 + \max\{|X_t^{(i)}|, |X_{t_{\ell+1}}^{(i)}|\}^p)dt$$

$$\le \int_{t_\ell}^{t_{\ell+1}} \left(\frac{t-t_\ell}{\Delta t}\right) C'(2h + |t_{\ell+1} - t|^\eta|)(1 + \max\{|X_t^{(i)}|, |X_{t_{\ell+1}}^{(i)}|\}^p)dt.$$

Taking expectation on both sides and using the moment bound (4.C.1), we get

$$\mathbb{E}\left[\left|\int_{t_\ell}^{t_{\ell+1}} \left(\frac{t-t_\ell}{\Delta t}\right)(\varphi^{\mathbf{C}}(X_t^{(i)}) - \varphi^{\mathbf{C}}(X_{t_{\ell+1}}^{(i)}))dt\right|\right] \le C\left(\Delta t h + \Delta t^{1+\eta}\right).$$

We get the same bound for I_2. Summing over ℓ, and taking the average in i, we then get

$$\mathbb{E}\left[|A(\varphi)|\right] \le C(h + \Delta t^\eta),$$

which implies the desired bound on the difference (4.C.5). □

Chapter 5

Online WSINDy

Abstract

This chapter presents an online algorithm for identification of partial differential equations (PDEs) based on the weak-form sparse identification of nonlinear dynamics algorithm (WSINDy). The algorithm is online in a sense that if performs the identification task by processing solution snapshots that arrive sequentially. The core of the method combines a weak-form discretization of candidate PDEs with an online proximal gradient descent approach to the sparse regression problem. In particular, we do not regularize the ℓ_0-pseudo-norm, instead finding that directly applying its proximal operator (which corresponds to a hard thresholding) leads to efficient online system identification from noisy data. We demonstrate the success of the method on the Kuramoto-Sivashinsky equation, the nonlinear wave equation with time-varying wavespeed, and the linear wave equation, in one, two, and three spatial dimensions, respectively. In particular, our examples show that the method is capable of identifying and tracking systems with coefficients that vary abruptly in time, and offers a streaming alternative to problems in higher dimensions.

5.1 Context and Motivations

System identification (SID) and parameter estimation of dynamical systems are ubiquitous tasks in scientific research and engineering, and are required steps in many control frameworks. A typical strategy is to solve a regression problem based on sample trajectories from the underlying system, with few samples available in practice. Identification of dynamical systems is a classical field of research [86]; recently, several works provided new theoretical insights on the efficacy of classical first-order optimization methods in solving SID problems based on single trajectories (see, e.g., [44, 51, 119, 128] and references therein). Existing results in this context are heavily focused on discrete-time, finite-dimensional systems of known functional form, yet the focus on single-trajectory data paves the way for identification of more complex dynamical systems in the *online* setting, which is the subject of the current chapter.

By suitably discretizing candidate dynamical systems using data and employing sparse regression, SID and parameter estimation can be accomplished simultaneously. A notable development in this pursuit is the sparse identification of nonlinear dynamics (SINDy) algorithm ([23]), a general framework for discovering dynamical systems using sparse regression. Since the inception of SINDy in the context of autonomous ordinary differential equations (ODEs), sparse recovery algorithms have been developed for autonomous partial differential equations (PDEs) ([117, 120]), stochastic differential equations (SDEs) ([20]), non-autonomous systems ([116]), and coarse-grained equations ([6]), to name a few. Outside of sparse regression approaches, deep learning has also been successful in identifying PDEs from data [89, 88, 151, 112].

A significant challenge in using SINDy to solve real-world problems is the computation of derivatives from noisy data. This was initially addressed in the context of ODEs in [121], by simply integrating candidate ODEs. Within the last few years, the consensus has emerged that weak-form SINDy (WSINDy, see [99, 100, 98]), where integration against test functions replaces numerical differentiation, is a powerful method that is significantly more robust to noisy data, particularly in the context of PDEs. Furthermore, WSINDy's efficient convolutional formulation makes it a viable method for identifying PDEs under the constraints of limited memory capacity and computing power that exist in the online setting[1].

The development of online algorithms is a relatively recent pursuit ([162, 60]), yet much progress has been made in applications to finance ([61]), data processing ([38]), and predictive control ([77]) (see [37, 64] for a recent surveys). In the context of sparse regression, several works have addressed online ℓ_1-minimization and other methods of regularizing the ℓ_0 pseudo-norm, although not in the context of learning dynamical systems ([153, 157, 67, 156, 78, 154, 129]). To the best of our knowledge, neither SINDy nor WSINDy have been merged with an online learning algorithm for PDEs[2].

[1] The method developed here could also be adapted to the standard SINDy algorithm, however we choose to focus on the weak form for its demonstrated abilities to handle noisy data with low computational overhead.

[2] There has, however, been work related to leveraging the equation learning ability of SINDy with Model Predictive

A successful approach for identifying PDEs and tracking parameters "on the fly" using multi-dimensional snapshots of data arriving sequentially over time would greatly benefit many areas of science and engineering. Possible paradigms in this online setting include identifying time-varying coefficients, SID in higher dimensions (where memory constraints require data to be streamed even for offline problems), and detecting changes in the dominant balance physics of the system, as terms become active or inactive dynamically. In this way, online sparse equation discovery has the potential to open doors to new application areas, and even improve performance of existing batch methods.

We confront some of these challenges in this work by considering spatiotemporal dynamical systems and incoming data snapshots at every timestep. In the spirit of classical online algorithms, we develop an online WSINDy framework to this setting of streaming data with memory constraints by replacing full-data availability and batch optimization capabilities with data bursts and light-weight proximal gradient descent iterations to approximately solve the sparse regression problem. At each iteration we process only the incoming snapshot in time, and we do not assume the ability to compute least-squares projections apart from the initial guess. We focus on three prototypical systems, (1) the Kuramoto-Sivashinsky (KS) equation, which exhibits spatiotemporal chaos and thus has time-fluctuating Fourier content, (2) the nonlinear wave equation in a time-variable medium in two spatial dimensions, and (3) the linear wave equation in three spatial dimensions, a preliminary example of a system in higher dimensions.

5.1.1 Notation

Vector-valued objects will be bold and lower-case, $\mathbf{x} \in \mathbb{R}^d$ for $d > 1$, while multi-dimensional arrays will be bold and upper-case, $\mathbf{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ for $n_i \in \mathbb{N}$, $1 \leq i \leq d$. To disambiguate between iteration and exponentiation, we refer to the qth element in a list of multi-dimensional arrays using superscripts in parentheses (e.g. $\mathbf{x}^{(q)}$ or $\mathbf{X}^{(q)}$), whereas raising to the power q (where applicable) is simply denoted $\mathbf{X}^q$. Reference to an element within a multi-dimensional array is given as a subscript (e.g. $\mathbf{x}_i$ or $\mathbf{X}_{i_1,\ldots,i_d}$). For a matrix $\mathbf{G} \in \mathbb{C}^{m \times n}$, we denote by $\mathbf{G}_S$ the restriction of $\mathbf{G}$ to the columns in $S \subset \{1, \ldots, n\}$. By some abuse of notation, $\mathbf{G}_S^T = (\mathbf{G}_S)^T$. Similary, for a vector $\mathbf{w} \in \mathbb{C}^n$, we let $\mathbf{w}_S \in \mathbb{R}^{|S|}$ be the restriction of $\mathbf{w}$ to the entries in S, where $|S|$ denotes the number of elements of S. The complement of S within $\{1, \ldots, n\}$ is denoted S^c. All scalar-valued objects will be in lower-case, with iteration, set membership, etc. denoted by subscripts (i.e. u_q is the qth element in the list $\{u_1, \ldots, u_{q-1}, u_q, u_{q+1}, \ldots\}$).

Control ([68]).

5.2 Problem Formulation

We consider PDEs of the form

$$D^{\boldsymbol{\alpha}^{(0)}}u(\mathbf{x},t) = \sum_{i,j=1}^{I,J} \mathbf{w}^\star_{(i-1)J+j}(t) D^{\boldsymbol{\alpha}^{(i)}} f_j(u(\mathbf{x},t),\mathbf{x}), \quad (\mathbf{x},t) \in \Omega \times [0,\infty), \tag{5.2.1}$$

where $\Omega \subset \mathbb{R}^d$ is a bounded open set. The operators $D^{\boldsymbol{\alpha}^{(i)}}$ for $1 \leq i \leq I$ represent any linear differential operator in the variables $(\mathbf{x},t) \in \mathbb{R}^{d+1}$, where $\boldsymbol{\alpha}^{(i)} = (\boldsymbol{\alpha}_1^{(i)}, \dots, \boldsymbol{\alpha}_{d+1}^{(i)})$ is a multi-index such that

$$D^{\boldsymbol{\alpha}^{(i)}} v = \frac{\partial^{\boldsymbol{\alpha}_1^{(i)} + \cdots + \boldsymbol{\alpha}_d^{(i)} + \boldsymbol{\alpha}_{d+1}^{(i)}}}{\partial \mathbf{x}_1^{\boldsymbol{\alpha}_1^{(i)}} \cdots \partial \mathbf{x}_d^{\boldsymbol{\alpha}_d^{(i)}} \partial t^{\boldsymbol{\alpha}_{d+1}^{(i)}}} v.$$

In this work we consider left-hand side operators $D^{\boldsymbol{\alpha}^{(0)}}$ to be either ∂_t or ∂_{tt}, which are given in two spatial dimensions ($d = 2$) by the multi-indices $\boldsymbol{\alpha}^{(0)} = (0,0,1)$ and $\boldsymbol{\alpha}^{(0)} = (0,0,2)$, respectively. The functions $f_j : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$, $1 \leq j \leq J$, include all possible nonlinearities present in the model, and together with the linear operators $D^{\boldsymbol{\alpha}^{(i)}}$ comprise the feature library $\Theta := \{D^{\boldsymbol{\alpha}^{(i)}} f_j\}_{i,j=1}^{I,J}$. The weight vector $\mathbf{w}^\star(t) \in \mathbb{R}^{IJ}$ is assumed to be *sparse* in Θ at each time t, and is allowed to vary in t.

We assume that at each time $t = k\Delta t$ for $k \in \mathbb{N}$ and fixed timestep Δt we are given a solution snapshot $\mathbf{U}^{(t)} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ of the form

$$\mathbf{U}^{(t)} = u(\mathbf{X},t) + \epsilon \tag{5.2.2}$$

where u solves (5.2.1) for some weight vector $\mathbf{w}^\star$ and $\mathbf{X} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a fixed known spatial grid of points in Ω having n_i points in the ith dimension and equal spacing Δx in each dimension. Here ϵ represents i.i.d. mean-zero noise with fixed finite variance σ^2 associated with sampling the underlying solution $u(\mathbf{x},t)$ at any point $\mathbf{x} \in \Omega$. We write $\mathbf{U} = (\mathbf{U}^{(0)}, \mathbf{U}^{(\Delta t)}, \dots, \mathbf{U}^{(k\Delta t)}, \dots)$ to denote the entire dataset in time. The problem is stated as follows.

Problem: Assume that a total of K_{mem} snapshots $\{\mathbf{U}^{(t-(K_{\text{mem}}-1)\Delta t)}, \dots, \mathbf{U}^{(t)}\}$ can be stored in memory at each time t and that at time $t + \Delta t$ a new snapshot $\mathbf{U}^{(t+\Delta t)}$ arrives, replacing the oldest snapshot in memory. Given the sampling model (5.2.2) for unknown σ^2, unknown ground truth PDE (5.2.1), and fixed library $\Theta := \{D^{\boldsymbol{\alpha}^{(i)}} f_j\}_{i,j=1}^{I,J}$, solve for coefficients $\widehat{\mathbf{w}}^{(t)}$ such that $\sup_{t>0} \left\| \widehat{\mathbf{w}}^{(t)} - \mathbf{w}^\star(t) \right\|$ is bounded.

5.3 Batch WSINDy

In the batch setting, assuming $\mathbf{w}^\star$ is constant in time, the weak-form sparse identification of nonlinear dynamics algorithm (WSINDy) proposed in [99, 100] solves this problem efficiently by first convolving equation (5.2.1) with a smooth function $\psi(\mathbf{x},t)$, compactly supported in $\Omega \times [0,T]$. After integrating by parts to put all partial derivatives onto ψ, this leads to the *convolutional weak*

form:

$$D^{\boldsymbol{\alpha}^{(0)}}\psi * u(\mathbf{x},t) = \sum_{i,j=1}^{I,J} \mathbf{w}^{\star}_{(i-1)J+j} D^{\boldsymbol{\alpha}^{(i)}}\psi * f_j(u,\cdot)(\mathbf{x},t), \tag{5.3.1}$$

where convolutions are performed over space and time. For efficiency, the test function ψ is chosen to be separable,

$$\psi(\mathbf{x},t) = \phi_1(\mathbf{x}_1)\cdots\phi_d(\mathbf{x}_d)\phi_{d+1}(t). \tag{5.3.2}$$

For example, it can be chosen using the Fourier spectrum of the noisy data to mitigate high-frequency noise (see [99]). Once ψ is chosen, we discretize the problem by selecting a finite set of *query points* $\mathcal{Q} := \{(\mathbf{x}^{(q)}, t_q)\}_{q=1}^{Q} \subset \Omega \times (0,T)$ and evaluating (5.3.1) at $\mathcal{Q}$, replacing u with the full dataset $\mathbf{U}$. Convolutions can be efficiently computed using the fast Fourier transform (FFT), which, due to the compact support of ψ, is equivalent to the trapezoidal rule and is highly accurate in the noise-free case ($\sigma^2 = 0$). This gives us the linear system

$$\mathbf{b} \approx \mathbf{G}\mathbf{w}^{\star},$$

where the qth entry of $\mathbf{b}$ is $\mathbf{b}_q = D^{\boldsymbol{\alpha}^{(0)}}\psi * \mathbf{U}(\mathbf{x}^{(q)}, t_q)$ and qth entry of the $((i-1)J+j)$th column of $\mathbf{G}$ is $\mathbf{G}_{q,(i-1)J+j} = D^{\boldsymbol{\alpha}^{(i)}}\psi * f_j(\mathbf{U},\cdot)(\mathbf{x}^{(q)}, t_q)$. Using the assumption that $\mathbf{w}^{\star}$ is sparse, we solve this linear system for $\widehat{\mathbf{w}} \approx \mathbf{w}^{\star}$ by solving the sparse recovery problem

$$\min_{\mathbf{w}\in\mathbb{R}^{IJ}} F(\mathbf{w};\lambda) = \min_{\mathbf{w}\in\mathbb{R}^{IJ}} \frac{1}{2}\left\|\mathbf{G}\mathbf{w}-\mathbf{b}\right\|_2^2 + \frac{1}{2}\lambda^2\left\|\mathbf{w}\right\|_0. \tag{5.3.3}$$

The sparsity threshold $\lambda > 0$ must be set by the user and is designed to strike a balance between fitting the data, associated with low residual $\|\mathbf{G}\mathbf{w}-\mathbf{b}\|_2$, and finding a parsimonious model, indicated by low $\|\mathbf{w}\|_0$ (and its value is typically calibrated via cross-validation) [59, 52].

With a large enough library Θ, a sparse vector $\widehat{\mathbf{w}}$ is required in order to interpret and efficiently simulate the resulting PDE. Replacing the ℓ_0-pseudonorm with e.g. an ℓ_2 penalty (i.e. ridge regression) may shrink coefficients, but will not result in a sparse $\widehat{\mathbf{w}}$. In addition, the columns of $\mathbf{G}$ are typically highly correlated since they are each constructed from the same dataset $\mathbf{U}$, which leads to many popular algorithms for solving (5.3.3) performing poorly, such as convex relaxation using the ℓ_1-norm [95, 42]. In the batch setting, the following approach has proved to be successful under various noise levels and systems of interest. For $\lambda > 0$ define the inner sequential thresholding step

$$\text{MSTLS}(\mathbf{G},\mathbf{b};\lambda) \qquad \begin{cases} \mathbf{w}^{(0)} = \mathbf{G}^{\dagger}\mathbf{b} \\ \mathcal{I}^{(\ell)} = \{1 \le k \le IJ \ : \ L_k(\lambda) \le |\mathbf{w}_k^{(\ell)}| \le U_k(\lambda)\} \\ \mathbf{w}^{(\ell+1)} = \underset{\text{supp}(\mathbf{w})\subset\mathcal{I}^{(\ell)}}{\arg\min} \left\|\mathbf{G}\mathbf{w}-\mathbf{b}\right\|_2^2. \end{cases} \tag{5.3.4}$$

Letting $\mathbf{G}_k$ be the kth column of $\mathbf{G}$, the lower and upper bounds are defined

$$\begin{cases} L_k(\lambda) = \lambda \max\left\{1,\ \dfrac{\|\mathbf{b}\|}{\|\mathbf{G}_k\|}\right\} \\ U_k(\lambda) = \dfrac{1}{\lambda} \min\left\{1,\ \dfrac{\|\mathbf{b}\|}{\|\mathbf{G}_k\|}\right\} \end{cases}, \qquad 1 \le k \le IJ. \tag{5.3.5}$$

The sparsity threshold $\widehat{\lambda}$ is then selected as the smallest minimizer of the cost function

$$\mathcal{L}(\lambda) = \frac{\|\mathbf{G}(\mathbf{w}(\lambda) - \mathbf{w}(0))\|_2}{\|\mathbf{G}\mathbf{w}(0)\|_2} + \frac{\|\mathbf{w}(\lambda)\|_0}{IJ} \tag{5.3.6}$$

where $\mathbf{w}(\lambda) := \text{MSTLS}(\mathbf{G}, \mathbf{b}; \lambda)$. We find $\widehat{\lambda}$ via grid search and set $\widehat{\mathbf{w}} = \text{MSTLS}(\mathbf{G}, \mathbf{b}; \widehat{\lambda})$ as the output of the algorithm. In words, this is a modified sequential thresholding algorithm with non-uniform thresholds (5.3.5) chosen based on the norms of the underlying library terms $\mathbf{G}_{(i-1)J+j} \approx D^{\boldsymbol{\alpha}^{(i)}}\psi * f_j(u)$ relative to the response vector $\mathbf{b} \approx D^{\boldsymbol{\alpha}^{(0)}}\psi * u$. The purpose of this is to (a) incorporate relative sizes of library terms $\mathbf{G}_k\mathbf{w}_k^\star$ along with absolute sizes of coefficients $\mathbf{w}^\star$ in the thresholding step, and (b) choose λ automatically.

5.4 Online WSINDy

The online setting is defined by data snapshots $\mathbf{U}^{(t)}$ arriving sequentially over time. An estimate $\widehat{\mathbf{w}}^{(t)}$ of the true parameters $\mathbf{w}^\star(t)$ must be computed before the arrival of the next snapshot $\mathbf{U}^{(t+\Delta t)}$ using only a fixed number K_{mem} of previous snapshots. Without access to the full time series $\mathbf{U}$, combined effects of the sample rate Δt, the number of snapshots K_{mem}, and the intrinsic timescales of the data determine the identifiability of the system: Δt must be small enough to accurately compute time integrals, but large enough that the data $\mathbf{U}$ is sufficiently dynamic over the time window $K_{\text{mem}}\Delta t$. Corruptions from noise have a greater impact because variance is not reduced by considering many samples in time, as was the case in the batch setting. Moreover, in realistic settings, solving for $\widehat{\mathbf{w}}^{(t)}$ before arrival of the next snapshot $\mathbf{U}^{(t+\Delta t)}$ fundamentally limits the size of $(\mathbf{G}, \mathbf{b})$ and the number of iterations one may perform using any sparse solver.

The online setting is inherently restrictive, yet it appears well-suited for an important set of problems that are challenging offline and for settings where $\widehat{\mathbf{w}}^{(t)}$ must be obtained without revisiting past data. In the batch setting, when the coefficient vector $\mathbf{w}^\star$ varies over time, the library Θ must include time-dependent terms and may grow too large to successfully solve for an accurate sparse solution. Another issue arises with high-dimensional datasets (as in cosmology, turbulence, molecular dynamics, etc.), which cannot easily be processed in a single batch. In these cases an online approach is natural and advantageous even if solutions $\widehat{\mathbf{w}}^{(t)}$ are not themselves required "online".

For the online approach, at each time t we seek to minimize the online cost function

$$\min_{\mathbf{w}\in\mathbb{R}^{IJ}} F_t(\mathbf{w};\lambda_t) = \min_{\mathbf{w}\in\mathbb{R}^{IJ}} \frac{1}{2}\left\|\mathbf{G}^{(t)}\mathbf{w}-\mathbf{b}^{(t)}\right\|_2^2 + \frac{1}{2}\lambda_t^2\left\|\mathbf{w}\right\|_0, \tag{5.4.1}$$

where $(\mathbf{G}^{(t)},\mathbf{b}^{(t)})$ is the linear system created from the K_{mem} slices $\{\mathbf{U}^{(t-(K_{\text{mem}}-1)\Delta t)},\dots,\mathbf{U}^{(t)}\}$ at time t. Notice also that we allow λ_t to change, as the initial guess λ_0 may not be optimal. In this online setting, we assume that we do not have the luxury of computing least-squares solutions (other than the initial guess), so we cannot use the approach outlined in (5.3.4)-(5.3.6), where (5.3.4) requires multiple least-squares solves, and performing a grid search over λ values requires multiple solves of (5.3.4). Hence, we consider the following online algorithm, which is simply the online proximal gradient descent combined with a decision tree update for λ_t at each step:

$$\begin{cases} \mathbf{z}^{(t)} = \widehat{\mathbf{w}}^{(t)} - \alpha_t(\mathbf{G}^{(t)})^T\left(\mathbf{G}^{(t)}\widehat{\mathbf{w}}^{(t)}-\mathbf{b}^{(t)}\right) \\ \widehat{\mathbf{w}}^{(t+\Delta t)} = H_{\lambda_t}\left(\mathbf{z}^{(t)}\right) \\ \lambda_{t+\Delta t} = \mathcal{T}(\lambda_t,\widehat{\mathbf{w}}^{(t+\Delta t)},\Delta\lambda,\lambda_{\max}). \end{cases} \tag{5.4.2}$$

The hard thresholding operator $H_\lambda(\mathbf{w})$ is the proximal operator of $\frac{1}{2}\lambda^2\left\|\mathbf{w}\right\|_0$ and is defined as

$$(H_\lambda(\mathbf{w}))_k = \begin{cases} \mathbf{w}_k, & |\mathbf{w}_k|\geq\lambda \\ 0, & \text{otherwise.} \end{cases} \tag{5.4.3}$$

The map $\mathcal{T}$ updates λ_t according to

$$\mathcal{T}(\lambda_t,\widehat{\mathbf{w}}^{(t+\Delta t)};\ \Delta\lambda,\lambda_{\max}) = \\ \begin{cases} (1-\Delta\lambda)\lambda_t, & F_t(\widehat{\mathbf{w}}^{(t+\Delta t)},\lambda_t) > F_{t-\Delta t}(\widehat{\mathbf{w}}^{(t)},\lambda_t)\ \&\ S_{t+\Delta t}\subsetneq S_t \\ (1-\Delta\lambda)\lambda_t+\lambda_{\max}\Delta\lambda, & \begin{cases} F_t(\widehat{\mathbf{w}}^{(t+\Delta t)},\lambda_t) > F_{t-\Delta t}(\widehat{\mathbf{w}}^{(t)},\lambda_t)\ \&\ S_t\subsetneq S_{t+\Delta t} \\ F_t(\widehat{\mathbf{w}}^{(t+\Delta t)},\lambda_t) \leq F_{t-\Delta t}(\widehat{\mathbf{w}}^{(t)},\lambda_t)\ \&\ S_t = S_{t+\Delta t}. \end{cases} \\ \lambda_t, & \text{otherwise.} \end{cases} \tag{5.4.4}$$

In words, there are two possible updates to λ_t: a convex combination between λ_t and 0 and a convex combination between λ_t and $\lambda_{\max}$. The former decreases λ_t and occurs when library terms are thresholded to zero *and* the objective function F_t increases. The latter increase λ_t and occurs when either (a) library terms are added *and* F_t increases or (b) the support set $S_t := \text{supp}\left(\widehat{\mathbf{w}}^{(t)}\right)$ doesn't change *and* F_t does not increase[3]. At each step we set $\alpha_t = 1/\|(\mathbf{G}^{(t)})^T\mathbf{G}^{(t)}_{S_t}\|_2$, the optimal stepsize for pure gradient descent given the support S_t. As an initial guess we set $\widehat{\mathbf{w}}^{(0)} = \left(\mathbf{G}^{(0)}\right)^\dagger\mathbf{b}^{(0)}$, which is the only least squares solve performed.

[3]The value for λ_t (and similarly for α_t) can easily be replaced by a constant when additional knowledge is available (e.g. when $\mathbf{w}^\star$ is known to satisfy certain bounds).

Remark 7. It is well-known in the batch case that picking λ is problem specific and prone to errors particularly in the presence of noise (see [99] for a discussion). Commonly some form of cross-validation is used to select λ offline. This is carried out in [93] for offline PDE identification using several sparse regression algorithms including proximal gradient descend applied to (5.3.3). It is less common to update λ over the course of the algorithm, although several strategies for this are presented in [40]. We stress that for variable-coefficient PDEs, as considered here, a time-varying λ is *necessary*, and offline cross validation can at best provide a good initial guess. The update policy given by $\mathcal{T}$ encodes simple objectives of any algorithm for (5.4.1) and works in all examples presented, however we leave optimizing the update rule as a topic for future work.

Remark 8. Similar to the batch case, we find that non-uniform thresholding greatly improves results. For brevity, we include in Appendix 5.A a description of how non-uniform thresholds such as (5.3.5) are incorporated into the online framework. We also note that the theoretical results in the next section carry over analogously in the non-uniform thresholding case.

5.4.1 Regret and Fixed Point Analysis

The behavior of the online algorithm is in large part dictated by the behavior of the batch proximal gradient descent method. The proximal gradient descent algorithm applied to the ℓ_0 norm is referred to as iterative hard thresholding (IHT) and was first studied rigorously in [15]. The lemmas below review some useful properties that can be found in that work relating solutions of (5.3.3) and stationary points of the proximal gradient descent algorithm (5.4.2) in the offline case and for fixed λ. We then use these results to bound the dynamic online regret, which we define as

$$Reg_D(T) := \sum_{\substack{k=0 \\ t=k\Delta t}}^{T} F_t(\widehat{\mathbf{w}}^{(t)}; \lambda_t) - F_t(\mathbf{w}^\star(t); \lambda_t), \tag{5.4.5}$$

where $\mathbf{w}^\star(t)$ is a global minimizer of $F_t(\mathbf{w}, \lambda_t)$. In particular, we first have the following:

Lemma 6. *Consider* $\mathbf{w}$ *such that one of the following holds:*

(i) $\mathbf{w}$ *is a local minimizer of* (5.3.3)

(ii) $\mathbf{w} = H_\lambda\left(\mathbf{w} - \mathbf{G}^T\left(\mathbf{G}\mathbf{w} - \mathbf{b}\right)\right)$

(iii) With $S = supp\left(\mathbf{w}\right)$*, we have that* $\mathbf{w}_S \in \arg\min_{\mathbf{z}} \|\mathbf{G}_S\mathbf{z} - \mathbf{b}\|_2^2$ *and*

$$\max_{i \in S^c} \left|\mathbf{G}_i^T(\mathbf{G}\mathbf{w} - \mathbf{b})\right| < \lambda \leq \min_{i \in S} \left|\mathbf{w}_i\right|.$$

Then it holds that $(ii) \iff (iii) \implies (i)$*. Moreover, if* $\mathbf{w}$ *a global minimizer, then* $(i) \implies (iii)$*.*

For completeness, a proof of Lemma 6 can be found in Appendix 5.C. For convergence of the algorithm, we also have the following from [15].

Lemma 7. *Assume that* $\|\mathbf{G}\|_2 < 1$. *Then the iterates* $\mathbf{w}^{n+1} = H_\lambda(\mathbf{w}^n - \mathbf{G}^T(\mathbf{G}\mathbf{w}^n - \mathbf{b}))$ *converge to a fixed point of* (5.3.3).

Lemma 6 implies that fixed points of the batch proximal gradient descent algorithm are local minimizers of $F(\mathbf{w};\lambda)$, and moreover that fixed points satisfy a necessary condition for global optimality given by (iii). Lemma 7 then guarantees[4] that iterates $\mathbf{w}^n$ do indeed converge to a local minimizer $\widehat{\mathbf{w}}$, and further that $\operatorname{supp}(\mathbf{w}^n) = \operatorname{supp}(\widehat{\mathbf{w}})$ for all $n \geq N$, for some finite N. However, we are not aware of results that guarantee recovery of the true support $\operatorname{supp}(\mathbf{w}^\star)$, where it is assumed that $\mathbf{b} = \mathbf{G}\mathbf{w}^\star + \mathbf{e}$ for noise $\mathbf{e}$. In [16], support recovery is proved for a related algorithm where $H_\lambda(\mathbf{w})$ is replaced by $H_s(\mathbf{w})$, which selects the largest s elements of $\mathbf{w}$, but this relies on several assumptions including a restricted isometry property, small noise $\mathbf{e}$, and knowledge of the sparsity level s. In the current setting of PDE identification from noisy data, none of these assumptions are realistic, although a similar support recovery result for algorithm (5.4.2) in the batch case would fill a gap in the literature.

If a fixed point $\widehat{\mathbf{w}}$ with $\operatorname{supp}(\widehat{\mathbf{w}}) = S$ satisfies that $\mathbf{G}_S^T\mathbf{G}_S$ is full rank, then $\widehat{\mathbf{w}}_S = \mathbf{G}_S^\dagger\mathbf{b}$ is the unique least squares solution over the columns in S. In [107] it is shown that this is sufficient for $\widehat{\mathbf{w}}$ to be a *strict* local minimizer, and moreover the only local minimizer with support S. Also in [107] is an extensive treatment of global minimizers of $F(\mathbf{w};\lambda)$, where it is shown that apart from a measure-zero set of linear systems $(\mathbf{G},\mathbf{b})$, the global minimizer is unique. We use this to bound the dynamic regret below.

Theorem 2. *Let* $\sigma_{1,t}$ *and* $\sigma_{n,t}$ *denote the first and last singular values of the matrix* $\mathbf{G}^{(t)} \in \mathbb{R}^{m\times n}$. *Assume the following:* $\max_t \lambda_t \leq \overline{\lambda} < \infty$, $\min_t \sigma_{n,t} \geq \overline{\sigma}_{\min} > 0$, $\max_t \sigma_{1,t} \leq \overline{\sigma}_{\max}$, *and* $\sup_t \alpha_t < \sigma_{\max}^{-2}$, $\inf_t \alpha_t > 0$. *In addition, assume that the global minimizer* $\mathbf{w}^\star(t)$ *of* $F_t(\mathbf{w};\lambda_t)$ *is unique for every* t *and satisfies* $|S_t^\star| \geq \overline{s} > 0$ *where* $S^\star = \operatorname{supp}(\mathbf{w}^\star(t))$. *Finally, assume that the tracking gap is globally bounded:* $\|\mathbf{w}^\star(t) - \mathbf{w}^\star(t+\Delta t)\|_2 := d_t \leq \overline{d}$. *Then the dynamic regret* (5.4.5) *grows at-worst linearly:*

$$Reg_D(T) \leq C_1 + C_2 T$$

for some $C_1 > 0$ *and* $C_2 > 0$. *In particular,* $\frac{1}{T}Reg_D(T)$ *remains bounded.*

The constants C_1 and C_2 are specified in the proof, which is presented in Appendix 5.D.

Remark 9. The above result establishes that $Reg_D(T)$ increases at-worst linearly in T, but this is only qualitative (the constants C_1 and C_2 are not meant to be sharp). Asymptotically, this is the same rate as online gradient descent applied to the time-varying ordinary least squares problem ([162]), and is a well-known fundamental limit for cases where the tracking gap d_t does not go to zero (see e.g. [8]).

Remark 10. Lines (5.D.1)-(5.D.2) of the proof establish error bounds on the coefficients, which

[4]The condition $\|\mathbf{G}\|_2 < 1$ in Lemma 7 can be replaced by stepsize $\alpha > 1$ satisfying $\alpha < 1/\|\mathbf{G}\|_2^2$.

lead to the asymptotic bound

$$\limsup_{t\to\infty} \left\| \widehat{\mathbf{w}}^{(t)} - \mathbf{w}^\star(t) \right\|_2 \leq \frac{1}{1-\overline{\rho}} \limsup_{t\to\infty} \left(d_t + \alpha_t \lambda_t \sqrt{|S_{t+\Delta t} \triangle S_t^\star|} \right),$$

where $\overline{\rho} := \sup_{t\geq 0} \left\| \mathbf{I} - \alpha_t (\mathbf{G}^{(t)})^T \mathbf{G}^{(t)} \right\|_2$ and $S_{t+\Delta t} \triangle S_t^\star$ is the set difference between $S_{t+\Delta t} = \operatorname{supp}\left(\widehat{\mathbf{w}}^{(t+\Delta t)}\right)$ and $S_t^\star := \operatorname{supp}\left(\mathbf{w}^\star(t)\right)$. This implies that if the tracking error and support difference go to zero ($d_t \to 0$, $S_{t+\Delta t} \triangle S_t^\star \to \emptyset$) then we recover the true coefficients in the limit.

Remark 11. The assumptions of Theorem 2 are standard for overdetermined $\mathbf{G}^{(t)}$ and data that is not pathological. In particular, upper bounds on $\sigma_{1,t}$ and d_t merely imply that the data does not blow up, while lower bounds on $\sigma_{n,t}$ and $|S_t^\star|$ imply that the data does not reach an equilibrium state. While both of these cases, blow up and equilibration, are interesting, the former rarely occurs in practice, and the latter is sufficiently challenging as to require new developments in a future work. Upper bounds on λ_t and α_t are cosmetic and required for the algorithm to produce nonzero solutions that are bounded. A lower bound on α_t is crucial to ensure $\overline{\rho} < 1$, which is necessary for convergence once the correct support has been recovered. We leave the case of underdetermined $\mathbf{G}^{(t)}$ to future work, but note that in practice the algorithm generally reaches an overdetermined subset after finitely many iterations.

Below we only examine cases where $S_t^\star = S^\star$ is fixed, and find that over a wide range of parameters the correct support is found in finitely many iterations. This leads to scenarios where the dynamic regret depends only on d_t asymptotically (see Figure 5.5.3 for a visualization of this case for the time-varying wave equation). We leave online discovery of PDEs with time-varying support to future work.

5.5 Numerical Experiments

Our primary focuses are the performance of the algorithm as a function of the number of snapshots K_{mem} allowed in memory and the sensitivity of the algorithm to noise. We examine the following three examples which display a range of dynamics over one to three spatial dimensions: the Kuramoto-Sivashinsky equation in 1D, a time-varying nonlinear wave equation in 2D, and the linear wave equation in 3D. We abbreviate each by KS, W2D and W3D. For each experiment we simulate a noise-free solution $\mathbf{U}_{exact}$ to the given PDE over a long time horizon. We then add i.i.d. Gaussian noise with mean zero and standard deviation $\sigma = \sigma_{NR} \left\| \mathbf{U}^\star \right\|_{rms}$ to each data point for a range of *noise ratios*[5] σ_{NR}. After an offline phase where a least squares solution is found from the first K_{mem} snapshots, we feed in one new snapshot at each time t and apply the online algorithm (5.4.2).

[5]Note that σ_{NR} is approximately equal to the ratio $\|\epsilon(:)\|_2 / \|\mathbf{U}_{exact}(:)\|_2$ of the noise to the true data, where "$\mathbf{U}_{exact}(:)$" denotes $\mathbf{U}_{exact}$ stretched into a column vector.

Algorithm Hyperparameters

We fix as many hyperparameters across examples as possible, and differences are summarized in Table 5.5.1. In all examples we fix the sparsity threshold update to $\Delta\lambda = 0.1$, the initial sparsity threshold to $\lambda_0 = 0.0001$, and the maximum sparsity threshold to $\lambda_{\max} = 0.1$. For the library we use

$$\Theta = \{\partial^k_{\mathbf{x}_i}(u^j)\}, \qquad 1 \le i \le d,\ 0 \le k \le 4,\ 0 \le j \le 4$$

in other words all spatial derivatives up to degree 4 of monomials up to degree 4 of the data (excluding mixed derivatives). For direct comparison of the effects of K_{mem} and σ_{NR} across examples, we fix the test function ψ in the representation 5.3.2 so that[6]

$$\phi_i(\mathbf{x}_i) = \left(1 - \left(\frac{\mathbf{x}_i}{21\Delta x}\right)^2\right)_+^{11}, \qquad 1 \le i \le d \tag{5.5.1}$$

$$\tag{5.5.2}$$

and

$$\phi_{d+1}(t) = \left(1 - \left(\frac{t}{(K_{\text{mem}} - 1)\Delta t/2}\right)^2\right)_+^9, \tag{5.5.3}$$

where $(z)_+ := \max\{z, 0\}$. In this way ψ is supported on $2 \times 21 + 1 = 43$ points in each spatial dimension and K_{mem} points in time, although note that $(\Delta x, \Delta t)$ change across examples. Since $\phi_{d+1}(t)$ is supported on K_{mem} points, there is only one integration in time at each iteration, so that the query points are given by $\mathcal{Q} = \{(\mathbf{x}^{(q)}, t_q)\}_{q=1}^{Q} = \mathcal{Q}_{\mathbf{x}} \times \{t - (K_{\text{mem}} - 1)\Delta t/2\}$ where for each example $\mathcal{Q}_{\mathbf{x}}$ is fixed across all values of K_{mem} and σ_{NR}. We take $\mathcal{Q}_{\mathbf{x}} \subset \mathbf{X}$ to be equally-spaced and such that the linear system $(\mathbf{G}^{(t)}, \mathbf{b}^{(t)})$ contains less than 10,000 rows (see Table 5.5.1 for exact dimensions). Online iteration times are reported below for computations performed on a laptop with 1.7GHz base clockspeed AMD Ryzen 7 pro 4750u processor and 38.4 GB of RAM.

Remark 12. By defining the temporal test function $\phi_{d+1}(t)$ to depend on K_{mem} according to (5.5.3), the implied strategy is that increasing K_{mem} (keeping more snapshots in memory) leads to more accurate integration in the time domain. One could instead fix the test function

$$\phi_{d+1}(t) = \left(1 - \left(\frac{t}{m\Delta t}\right)^2\right)_+^9$$

for some $m \le (K_{\text{mem}} - 1)/2$ for all K_{mem} considered, leading to a fixed integration window of length $2m + 1$ in time. Increasing K_{mem} would then allow for more integrations in time (i.e. a

[6]Test functions (5.5.2) and (5.5.3) can be made general by replacing powers $p_x = 11$, $p_t = 9$ and spacings $m_x = 21$, $m_t = (K_{\text{mem}} - 1)/2$ with general values p_x, p_t and m_x, m_t as in [99, 98]. The resulting general form for ψ has been shown to be successful across a wide range of systems. However, optimal test function selection is an active area of research.

	$\dim(\mathbf{X})$	T	$\dim(\mathbf{G}^{(t)})$	$(\Delta x, \Delta t)$
KS	256×1	3946	214×21	$(0.939, 0.586)$
W2D	129×403	1639	7964×37	$(0.0156, 0.0122)$
W3D	$128 \times 128 \times 128$	960	8192×53	$(0.0491, 0.0122)$

Table 5.5.1: Resolution and dimensions of datasets used in examples.

larger set of query points $\mathcal{Q}$), adding rows to the linear system $(\mathbf{G}^{(t)}, \mathbf{b}^{(t)})$. Our chosen strategy fixes the dimensions of $(\mathbf{G}^{(t)}, \mathbf{b}^{(t)})$, leading to a more direct comparison across examples. We leave this trade-off between the number of time integrations and the accuracy of time integrations to future work.

Performance Analysis

We are concerned with the ability of the algorithm to recover the support of the true model coefficients $S^\star := \operatorname{supp}(\mathbf{w}^\star)$ as well as the accuracy of $\widehat{\mathbf{w}}^{(t)}$ over time, depending primarily on the number K_{mem} of solution snapshots allowed in memory and the noise level σ_{NR} corrupting the data. To assess support recovery, we measure the *true positivity ratio* (TPR)

$$\text{TPR}(\widehat{\mathbf{w}}^{(t)}) := \frac{\text{TP}(\widehat{\mathbf{w}}^{(t)})}{\text{TP}(\widehat{\mathbf{w}}^{(t)}) + \text{FP}(\widehat{\mathbf{w}}^{(t)}) + \text{FN}(\widehat{\mathbf{w}}^{(t)})}$$

where $\text{TP}(\widehat{\mathbf{w}}^{(t)}) := |S_t \cap S^\star|$ is the number of correctly identified nonzero coefficients, $\text{FP}(\widehat{\mathbf{w}}^{(t)}) := |S_t \cap (S^\star)^c|$ is the number of falsely identified nonzero coefficients, and $\text{FN}(\widehat{\mathbf{w}}^{(t)}) := |S_t^c \cap S^\star|$ is the number of falsely identified zero coefficients. A TPR of 1 indicates successful support recovery, while $\text{TPR} = 0.75$ indicates 3/4 terms were correctly identified, and so on. We measure the accuracy of $\widehat{\mathbf{w}}^{(t)}$ in the relative ℓ_2-norm:

$$E_2(\widehat{\mathbf{w}}^{(t)}) := \left\| \widehat{\mathbf{w}}^{(t)} - \mathbf{w}^\star(t) \right\|_2 / \left\| \mathbf{w}^\star(t) \right\|_2 .$$

We report the results of $\text{TPR}(\widehat{\mathbf{w}}^{(t)})$ and $E_2(\widehat{\mathbf{w}}^{(t)})$ averaged over 100 instantiations of noise.

5.5.1 Kuramoto-Sivashinsky (KS)

$$\partial_t u = -\partial_x \left(u^2\right) - \partial_{xx} u - \partial_{xxxx} u. \tag{5.5.4}$$

The Kuramoto-Sivashinsky (KS) equation is challenging because the solution exhibits spatiotemporal chaos and so has a Fourier spectrum that varies in time. This leads to potentially different dynamics at each timestep in the online learning perspective. The PDE also has a 4th-order derivative in space which is difficult to compute accurately and to identify via sparse regression, especially

when noise is present. We simulate the solution using a high-order method (accurate to 6-7 digits) and use a dataset of 256×3496 points in space and time at resolution $(\Delta x, \Delta t) = (0.393, 0.586)$. Online iterations take less than 0.01 seconds, which includes building the linear system $(\mathbf{G}^{(t)}, \mathbf{b}^{(t)})$, which is the most costly step.

In Figure 5.5.1 the average evolution of $E_2(\widehat{\mathbf{w}}^{(t)})$ and $\mathrm{TPR}(\widehat{\mathbf{w}}^{(t)})$ is depicted for various noise levels σ_{NR} and memory capacities K_{mem}. The system is correctly identified for all trials when $K_{\mathrm{mem}} \in \{13, 17, 21, 25\}$ and $\sigma_{NR} \in \{0, 0.001, 0.01\}$, with relative errors E_2 less than 10^{-2} once the system is identified. For larger noise $\sigma_{NR} = 0.1$, results stagnate at sub-optimal values, indicating that more data is needed to identify the system (note that $\mathbf{G}^{(t)}$ only has 214 rows). With $K_{\mathrm{mem}} = 5$ we recover the correct system only in the noiseless case ($\sigma_{NR} = 0$), indicating that 5 points in time does not result in accurate resolution of the dynamics.

5.5.2 Variable-medium nonlinear wave equation in 2D (W2D)

$$\partial_{tt} u = c(t)\left(\partial_{xx} u + \partial_{yy} u\right) - u^3 \tag{5.5.5}$$

We examine a variable-medium nonlinear wave equation in 2D, given by equation (5.5.5), where the variable medium is modeled by the time-varying wavespeed

$$c(t) = 1 + (0.2)\frac{2}{\pi}\arctan(40\cos(2\pi(0.1)t)),$$

The wavespeed is a smoothed square wave and represents a system with abrupt speed modulation (see Figure 5.5.3 for depictions). We simulate the solution using a Fourier $\otimes$ Legendre spectral method in space with leap-frog timestepping. The exact data $\mathbf{U}_{exact}$ has dimensions $129 \times 403 \times 1639$ in (x, y, t) with resolution $(\Delta x, \Delta t) = (0.0156, 0.0122)$. Each snapshot $\mathbf{U}^{(t)}$ is 0.42 megabytes (Mb) and online iterations take approximately 0.08 seconds.

Figure 5.5.2 shows robust recovery for $K_{\mathrm{mem}} \in \{13, 17, 21, 25\}$ up to $\sigma_{NR} = 0.1$, with rapid identification for small noise. This is despite abrupt changes in the wavespeed c. For $K_{\mathrm{mem}} = 9$ we see recovery up to $\sigma_{NR} = 0.001$, indicating that for larger noise 9 points in time is insufficient to discretize the integrals $\partial_{tt}\psi * u$ accurately, analogous to the case $K_{\mathrm{mem}} = 5$ for KS.

The left panel of Figure 5.5.2 shows that once the system is identified, abrupt changes in the wavespeed temporarily increase the coefficient error E_2, but the correct support $S^\star$ remains identified and the errors swiftly decay. In Figure 5.5.3 we plot the average learned wavespeed $\widehat{c}(t)$ as well as the maximum and minimum values of $\widehat{c}(t)$ attained over all 100 trials, revealing that increasing K_{mem} from 17 to 25 leads to a significant decrease in the variance of $\widehat{c}$ after the system has been identified. This is purely an affect of using the weak form to discretize the time derivatives, and demonstrates that even under large noise and abruptly changing coefficients, the algorithm is able to maintain support recovery and accuracy.

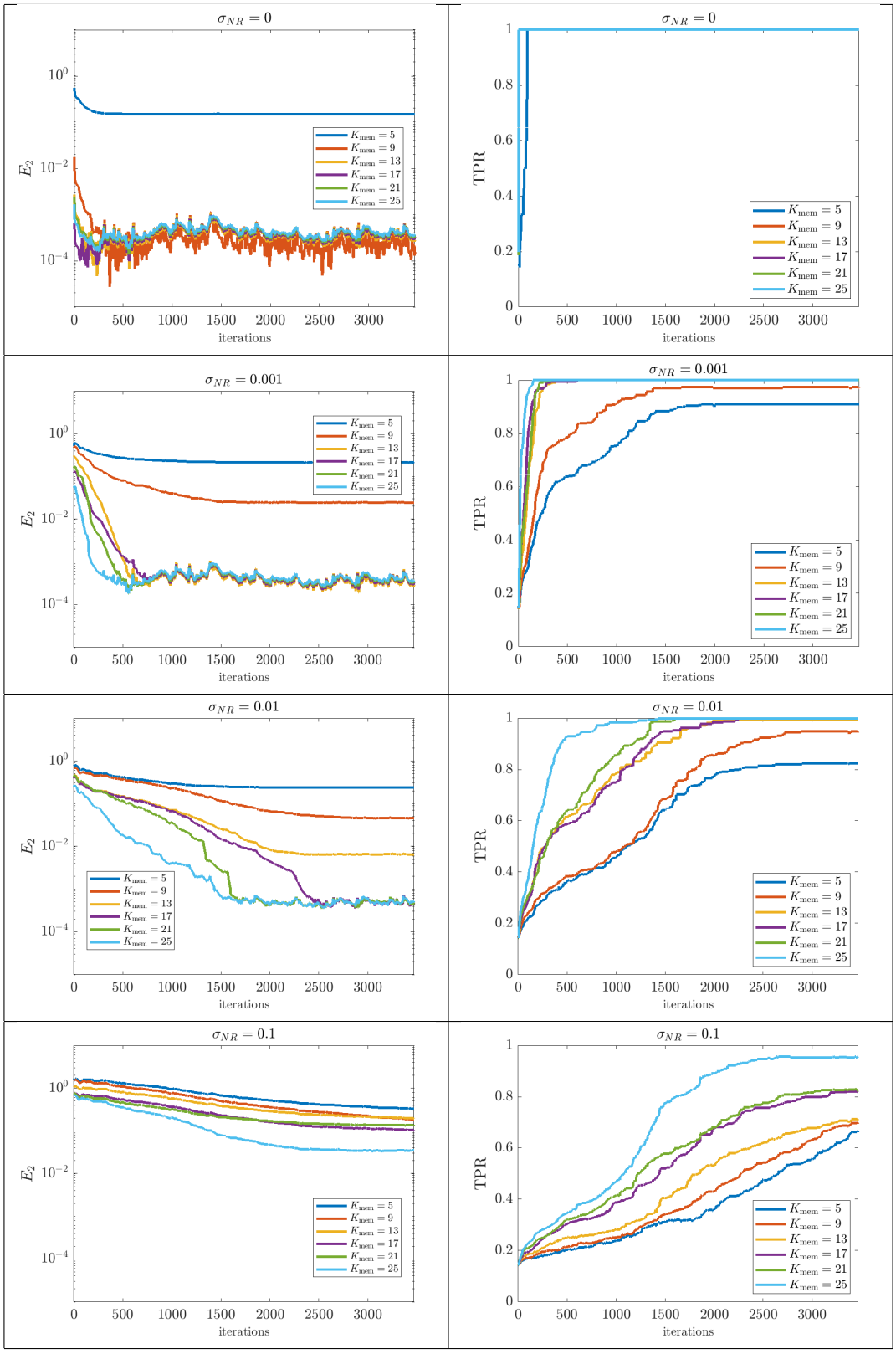


Figure 5.5.1: Online identification of the Kuramoto-Sivashinsky equation (5.5.4) for $K_{\text{mem}} \in \{5, 9, 13, 17, 21, 25\}$ and (top to bottom) $\sigma_{NR} \in \{0, 0.001, 0.01, 0.1\}$. Left: average coefficient error $E_2(\widehat{\mathbf{w}}^{(t)})$. Right: average total positivity ratio $\text{TPR}(\widehat{\mathbf{w}}^{(t)})$.

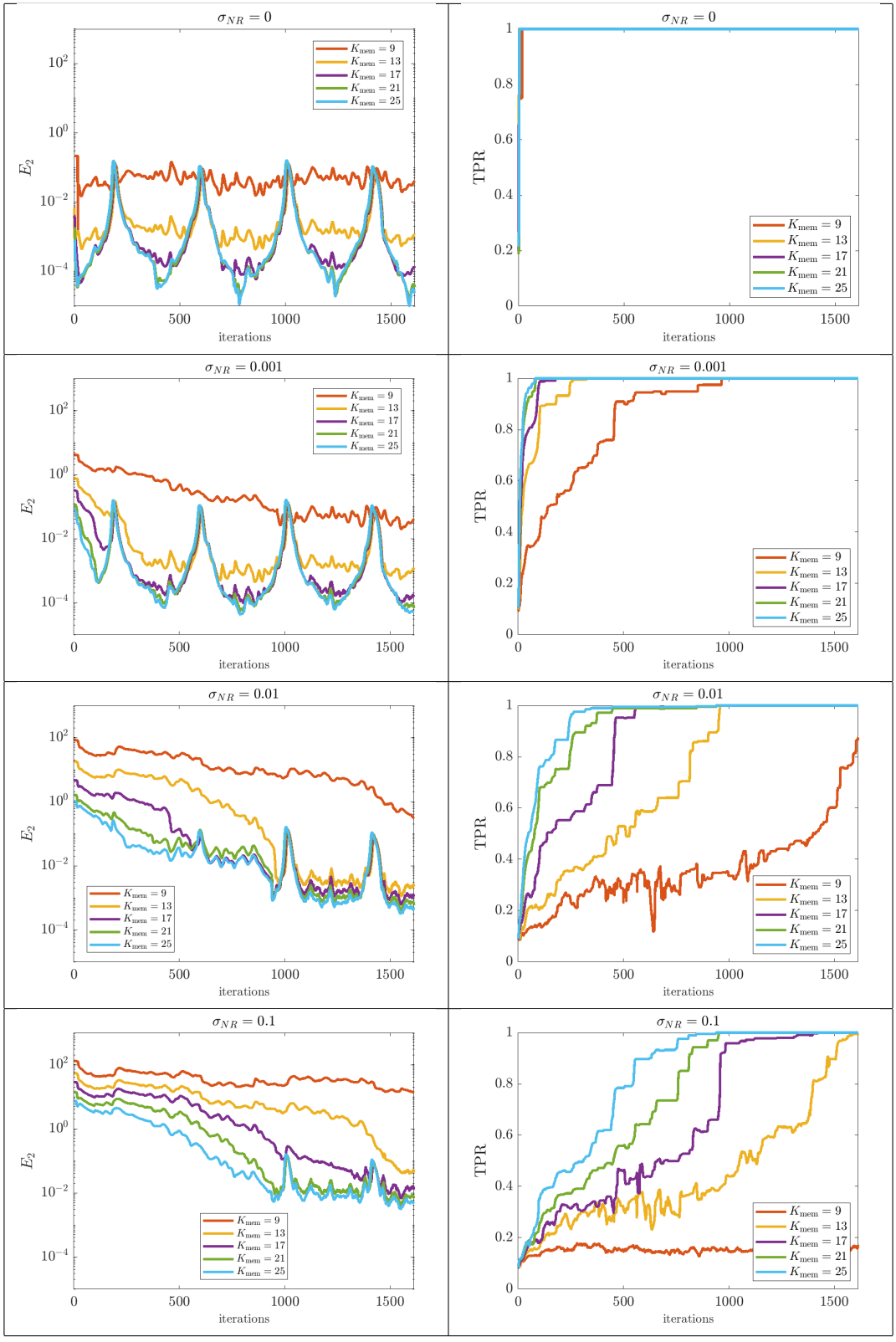


Figure 5.5.2: Online identification of the variable medium nonlinear wave equation (5.5.5) for $K_{\text{mem}} \in \{9, 13, 17, 21, 25\}$ and (top to bottom) $\sigma_{NR} \in \{0, 0.001, 0.01, 0.1\}$. Left: average coefficient error $E_2(\widehat{\mathbf{w}}^{(t)})$. Right: average total positivity ratio $\text{TPR}(\widehat{\mathbf{w}}^{(t)})$.

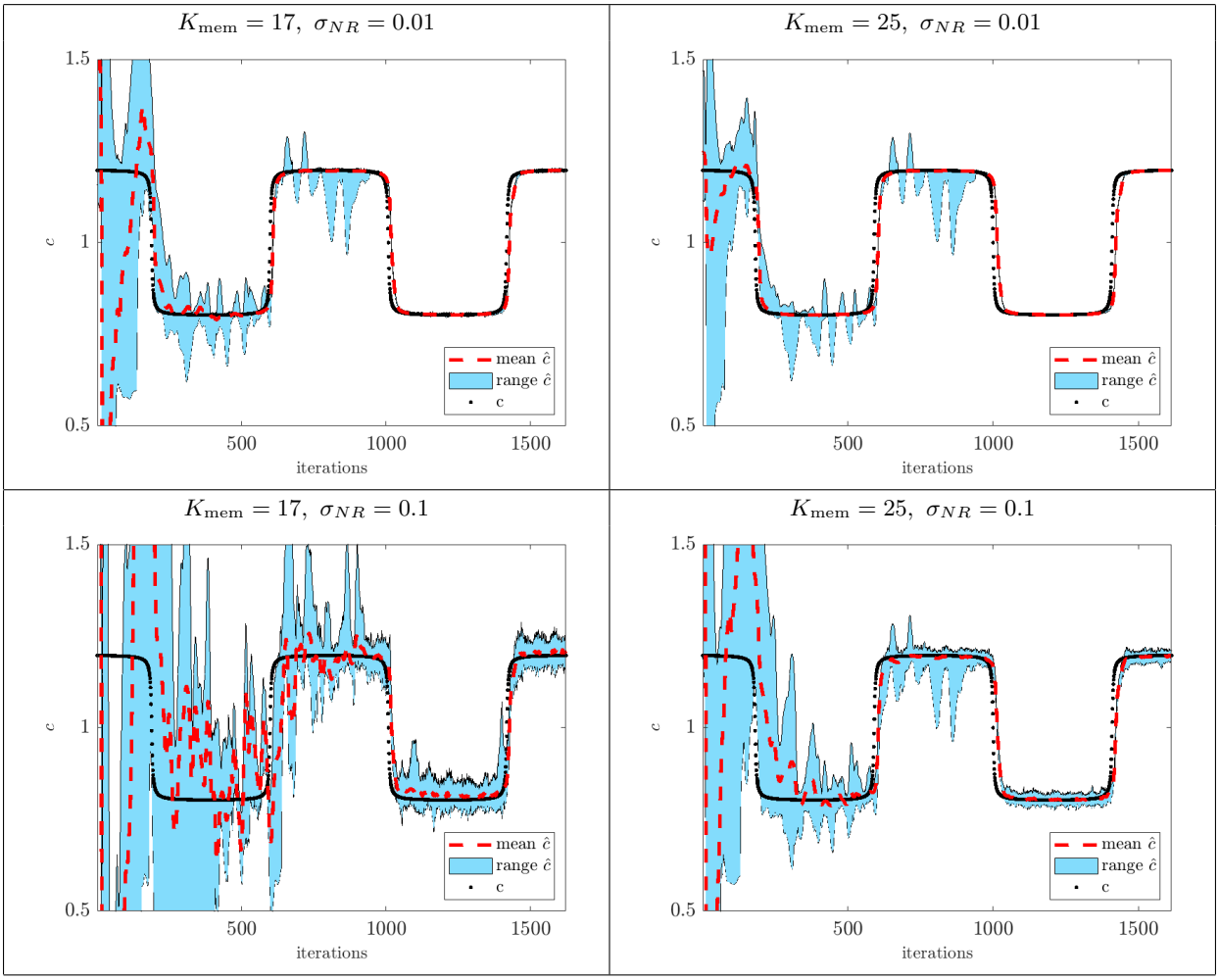


Figure 5.5.3: Online estimation of the wavespeed $c(t)$ (shown in black) for PDE (5.5.5). The average learned wavespeed $\widehat{c}(t)$ is shown in red while the blue shaded region shows the maximum and minimum values attained over all 100 trials. Notice the accuracy for later iterations when $\sigma_{NR} = 0.01$, and the reduction in variance moving from $K_{\text{mem}} = 17$ to $K_{\text{mem}} = 25$ when $\sigma_{NR} = 0.1$.

5.5.3 Wave equation in 3D (W3D)

$$\partial_{tt}u = \partial_{xx}u + \partial_{yy}u + \partial_{zz}u \tag{5.5.6}$$

For our last example we treat the linear wave equation in 3D. Exact data $\mathbf{U}_{exact}$ has dimensions $128 \times 128 \times 128 \times 960$ in (x, y, t) with resolution $(\Delta x, \Delta t) = (0.0491, 0.0122)$. Each snapshot $\mathbf{U}^{(t)}$ is 16.8 Mb and online iterations take approximately 1.3 seconds.

Results are depicted in Figure 5.5.4. We again find robust recovery for $K_{\text{mem}} \in \{13, 17, 21, 25\}$ up to $\sigma_{NR} = 0.1$, although in 5% of trials at $\sigma_{NR} = 0.1$ the $K_{\text{mem}} = 13$ case finds a spurious term $\approx -0.8u$. Even at $\sigma_{NR} = 0.1$ the coefficients are accurate to more than 2 digits once recovered for $K_{\text{mem}} \geq 17$. For $K_{\text{mem}} = 9$ we see poor performance for the same reason as above with W2D, but now manifesting as recovery of the spurious term $\approx -0.8u$, indicating that the inaccurate computation of $\partial_{tt}\psi * u$ produces spurious damping. This is not an altogether unreasonable affect if computing $\partial_{tt}\psi * u$ numerically is viewed as an attenuated second derivative calculation, although it does imply that higher-order time derivatives require more snapshots to be saved in memory.

5.6 Conclusions

We have demonstrated on several protoypical examples, and over a wide range of noise and memory scenarios, the viability of an online algorithm for PDE identification based on the weak-form sparse identification of nonlinear dynamics algorithm (WSINDy). The core of the method combines a weak-form discretization of candidate PDEs with the online proximal gradient descent algorithm applied directly to the least squares cost function with ℓ_0-pseudo-norm regularization (5.4.2). Compared with the more common approach of regularizing the ℓ_0-pseudo-norm (e.g. with $\|\cdot\|_1$ or weighted variants [26]), we find that directly applying $\text{prox}_{\lambda\|\cdot\|_0}$, leading to hard thresholding, and adaptively selecting λ_t, exhibits good performance in efficiently identifying systems, handling noise, and tracking time-varying coefficients.

Numerical experiments with an abruptly changing wavespeed indicate that our method is a lightweight counterpart to existing methods for variable coefficients (e.g. [116]), which may be of independent interest in the control of wave equations in variable-media ([43, 45, 108, 126, 30, 139]). Examination of the wave equation in 3D also offers a different perspective on PDE identification in higher dimensions: problems with large datasets can be implemented in an online data-streaming fashion (not necessarily along the time axis as implemented here). It may therefore be advantageous from the standpoint of memory usage to solve certain batch problems in the online manner we have presented.

The algorithm's successes warrant further investigation in a number of areas. While we have characterized stationary points of the batch algorithm and proved boundedness of the average dynamic regret, we leave a more complete analysis to future work. In particular, one could analyze

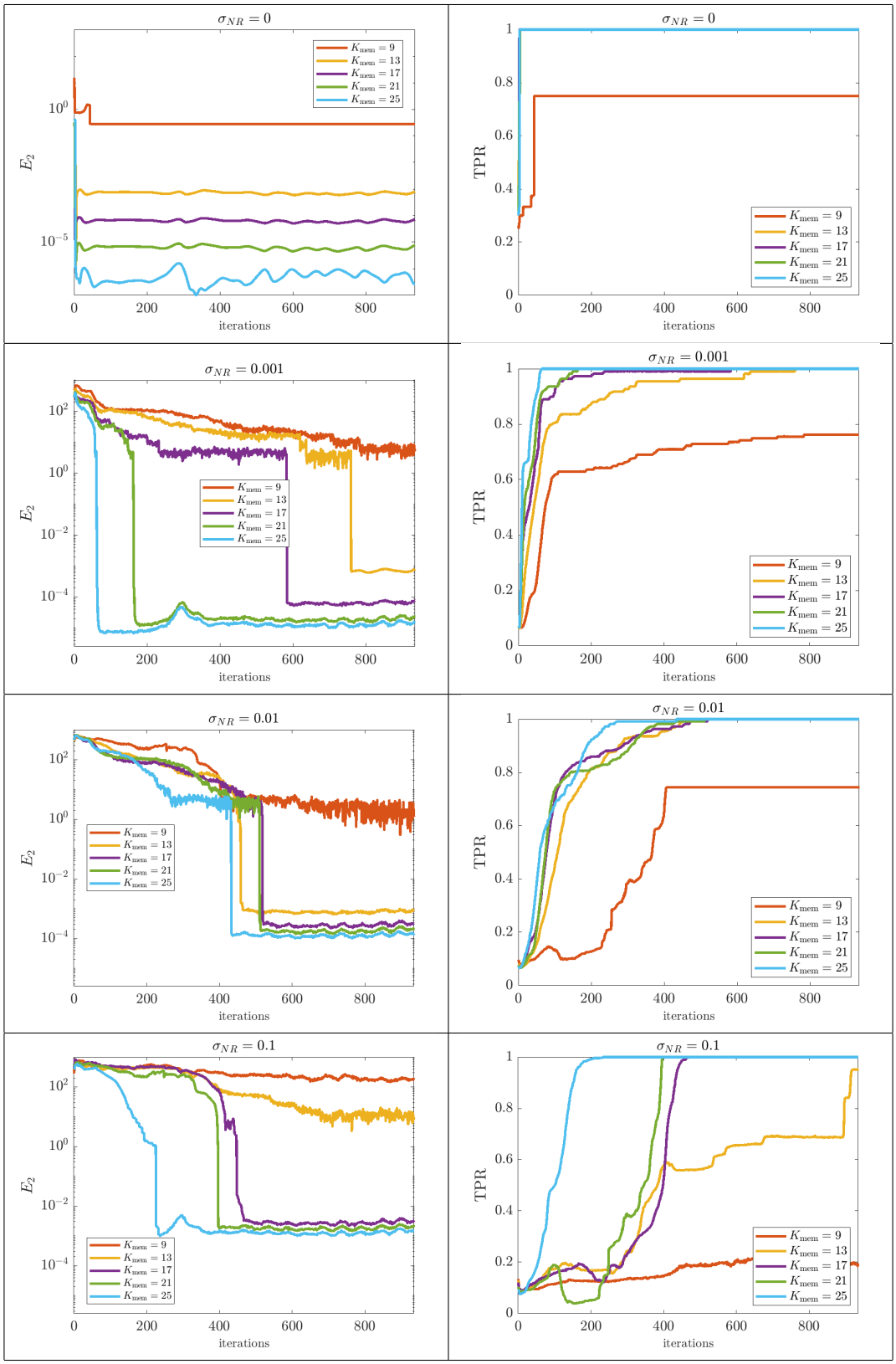


Figure 5.5.4: Online identification of the wave equation in three spatial dimensions (5.5.6) for $K_{\text{mem}} \in \{9, 13, 17, 21, 25\}$ and (top to bottom) $\sigma_{NR} \in \{0, 0.001, 0.01, 0.1\}$. Left: average coefficient error $E_2(\widehat{\mathbf{w}}^{(t)})$. Right: average total positivity ratio $\text{TPR}(\widehat{\mathbf{w}}^{(t)})$.

the error $\left\|\mathbf{w}^{(t)} - \mathbf{w}^{\star}(t)\right\|$ as a function of library Θ, test function ψ, data sampling rates $(\Delta x, \Delta t)$, memory size K_{mem}, noise ratio σ_{NR}, etc. It may also advantageous to design adaptive schemes which update Θ and ψ throughout the course of the algorithm, depending on the dynamics of the data and previously learned equations. We also note that an obvious next direction is to identify switching systems where the true support $S_t^{\star}$ changes with time. Nevertheless, the current framework is well-suited for a large variety of problems and opens the door to online PDE identification as well as the possibility of solving batch problems in an online manner.

Appendix

5.A Column scaling and non-uniform thresholds

For stability, we normalize the columns of $\mathbf{G}^{(t)}$ at each step, defining $\widetilde{\mathbf{G}}^{(t)} = \mathbf{G}^{(t)}\mathbf{M}^{(t)}$ with

$$\mathbf{M}^{(t)} = \operatorname{diag}\left(\left\|\mathbf{G}_1^{(t)}\right\|_2^{-1}, \ldots, \left\|\mathbf{G}_{IJ}^{(t)}\right\|_2^{-1}\right).$$

In particular, this allows for a larger stepsize $\widetilde{\alpha}_t = 1/\left\|(\widetilde{\mathbf{G}}^{(t)})^T\widetilde{\mathbf{G}}_{S_t}^{(t)}\right\|_2$ and leads to a reasonable estimate $\widetilde{\alpha}_t = 1/\sqrt{|S_t|IJ}$ for a stepsize that does not require computation of the matrix 2-norm.

For more flexibility, we allow for non-uniform thresholding. For a set of thresholds $\boldsymbol{\lambda} \in \mathbb{R}^{IJ}$, we define the non-uniform thresholding operator $H_{\boldsymbol{\lambda}}$ by

$$(H_{\boldsymbol{\lambda}}(\mathbf{x}))_i = \begin{cases} \mathbf{x}_i, & |\mathbf{x}_i| \geq \boldsymbol{\lambda}_i \\ 0, & \text{otherwise.} \end{cases}$$

This happens to be the proximal operator of the non-uniform ℓ_0-norm

$$\|\mathbf{x}\|_{0,\boldsymbol{\lambda}} := \sum_{i=1}^{IJ} \boldsymbol{\lambda}_i^2 \mathbb{1}_{\mathbb{R}\setminus\{0\}}(\mathbf{x}_i), \tag{5.A.1}$$

where $\|\mathbf{x}\|_{0,\boldsymbol{\lambda}} = \lambda^2 \|\mathbf{x}\|_0$ when $\boldsymbol{\lambda} = (\lambda, \ldots, \lambda)$. The resulting online cost function being minimized after incorporation of both non-uniform thresholding and column rescaling is

$$\widetilde{F}_t(\mathbf{w}; \boldsymbol{\lambda}^{(t)}) = \frac{1}{2}\left\|\widetilde{\mathbf{G}}^{(t)}\mathbf{w} - \mathbf{b}^{(t)}\right\|_2^2 + \frac{1}{2}\|\mathbf{w}\|_{0,(\mathbf{M}^{(t)})^{-1}\boldsymbol{\lambda}^{(t)}}, \tag{5.A.2}$$

whose fixed points $\widetilde{\mathbf{w}}^{\star,(t)}$ coincide with those of the desired cost function

$$F_t(\mathbf{w}; \boldsymbol{\lambda}^{(t)}) = \frac{1}{2}\left\|\mathbf{G}^{(t)}\mathbf{w} - \mathbf{b}^{(t)}\right\|_2^2 + \frac{1}{2}\|\mathbf{w}\|_{0,\boldsymbol{\lambda}^{(t)}} \tag{5.A.3}$$

after a diagonal transformation $\mathbf{w}^{\star,(t)} = \mathbf{M}^{(t)}\widetilde{\mathbf{w}}^{\star,(t)}$. With these two pieces, the online algorithm

for (5.A.2) becomes

$$\begin{cases} \widetilde{\mathbf{w}}^{(t)} = (\mathbf{M}^{(t)})^{-1}\widehat{\mathbf{w}}^{(t)} \\ \widehat{\mathbf{w}}^{(t+\Delta t)} = H_{\widetilde{\alpha}_t \boldsymbol{\lambda}^{(t)}}\left(\mathbf{M}^{(t)}\left(\widetilde{\mathbf{w}}^{(t)} - \widetilde{\alpha}_t(\widetilde{\mathbf{G}}^{(t)})^T\left(\widetilde{\mathbf{G}}^{(t)}\widetilde{\mathbf{w}}^{(t)} - \mathbf{b}^{(t)}\right)\right)\right), \end{cases}$$

however this can equivalently be written in terms of the desired coefficients $\widehat{\mathbf{w}}^{(t)}$ as

$$\widehat{\mathbf{w}}^{(t+\Delta t)} = H_{\widetilde{\alpha}_t \boldsymbol{\lambda}^{(t)}}\left(\widehat{\mathbf{w}}^{(t)} - \widetilde{\alpha}_t(\mathbf{M}^{(t)})^2(\mathbf{G}^{(t)})^T\left(\mathbf{G}^{(t)}\widehat{\mathbf{w}}^{(t)} - \mathbf{b}^{(t)}\right)\right). \tag{5.A.4}$$

In direct analogy to the batch WSINDy thresholding scheme (5.3.4)-(5.3.5), we use thresholds $\boldsymbol{\lambda}^{(t)} = \max(1, \|\mathbf{b}^{(t)}\| \operatorname{diag}(\mathbf{M}^{(t)}))\lambda_t$, which eliminate small coefficient values $\min_{i\in S_t} |\widehat{\mathbf{w}}_i^{(t)}| \geq \lambda_t$ as well as small terms in the sense of dominant balance with respect to $\mathbf{b}^{(t)}$:

$$\min_{i\in S_t} \frac{\left\|\mathbf{G}_i^{(t)}\widehat{\mathbf{w}}_i^{(t)}\right\|_2}{\left\|\mathbf{b}^{(t)}\right\|_2} \geq \lambda_t.$$

The update rule (5.4.4) for λ_t is unchanged after replacing $F_t(\mathbf{w};\lambda_t)$ with $F_t(\mathbf{w};\boldsymbol{\lambda}^{(t)})$ defined in (5.A.3).

5.B Implementation and Computational Complexity

The offline phase has four components:

1. Initialize hyperparameters $\psi(\mathbf{x},t) = \phi(\mathbf{x})\theta(t)$, $\Theta = \{D^{\boldsymbol{\alpha}^{(i)}} f_j\}_{i=0,j=1}^{I,J}$, $\Delta\lambda$, $\lambda_{\max}$, λ_0, where the test function ψ is either prescribed manually or selected using the changepoint algorithm from [99] using the initial K_{mem} slices $\{\mathbf{U}^{(0)},\dots,\mathbf{U}^{((K_{\text{mem}}-1)\Delta t)}\}$.

2. Compute and store the Fourier transforms $\{\widehat{D^{\boldsymbol{\alpha}^{(i)}}\psi}\}_{i=0}^{I}$ to reuse at each step when computing convolutions (recall ψ is separable so this storage cost is negligible).

3. Compute initial library of spatially integrated terms

$$\Psi \;:=\; \{\Psi^{(t)}\}_{t=0}^{(K_{\text{mem}}-1)\Delta t} \;:=\; \left\{\left\{D^{\boldsymbol{\beta}^{(i)}}\phi * f_j(\mathbf{U}^{(t)})(\mathcal{Q}_{\mathbf{x}},t)\right\}_{i=0,j=1}^{I,J}\right\}_{t=0}^{(K_{\text{mem}}-1)\Delta t}$$

   where $\boldsymbol{\beta}^{(i)} = (\boldsymbol{\alpha}_1^{(i)},\dots,\boldsymbol{\alpha}_d^{(i)})$ is the spatial part of the multi-index $\boldsymbol{\alpha}^{(i)}$ operating on the spatial part ϕ of the test function ψ (recall that $\mathcal{Q}_{\mathbf{x}}$ is the set of spatial points over which convolutions are evaluated, also equal to the number of rows in $\mathbf{G}^{(t)}$).

4. Compute initial weights $\widehat{\mathbf{w}}^{(0)} = (\mathbf{G}^{(0)})^{\dagger}\mathbf{b}^{(0)}$ where $\mathbf{b}^{(0)}$ and $\mathbf{G}^{(0)}$ are obtained by integrating the elements of Ψ in time against the corresponding temporal test functions $D^{\boldsymbol{\alpha}_{d+1}^{(i)}}\theta$.

For each t in the online phase we compute $\Psi^{(t)}$ using only the incoming slice $\mathbf{U}^{(t)}$, which replaces $\Psi^{(t-K_{\text{mem}}\Delta t)}$ in memory. $(\mathbf{G}^{(t)}, \mathbf{b}^{(t)})$ are then computed by integrating the elements of Ψ in time against the corresponding temporal test functions $D^{\boldsymbol{\alpha}^{(i)}_{d+1}}\theta$, which amounts to a series of dot products between length-K_{mem} vectors. Computation of $\mathbf{G}^{(t)}$ at each time t thus requires $J|\mathbf{X}|$ function evaluations $f_j(\mathbf{U}^{(t)})$ (each counted as 1 floating point operation (flop)) followed by IJ convolutions against $D^{\boldsymbol{\beta}^{(i)}}\phi$, and finally integration in time. The total flop count at each step is at most

$$J|\mathbf{X}|\left(1 \;+\; CI\log N \;+\; 2IK_{\text{mem}}\frac{|\mathcal{Q}_{\mathbf{X}}|}{|\mathbf{X}|}\right)$$

where C is such that $\mathbf{x}*\mathbf{y}$ costs $CN\log N$ using FFTs for length-N vectors $\mathbf{x}$ and $\mathbf{y}$, minus the cost of one FFT (since we have precomputed these for $D^{\boldsymbol{\beta}^{(i)}}\phi$) and $N \approx |\mathbf{X}|^{1/d}$ is the one-dimensional length scale of the data. In other words, only

$$F = J\left(1 + CI\log N + IK_{\text{mem}}\frac{|\mathcal{Q}_{\mathbf{X}}|}{|\mathbf{X}|}\right)$$

flops are performed per incoming data point in $\mathbf{U}^{(t)}$ (and a more careful analysis leads to a lower cost in the factor $CI\log N$ by incorporating the subsampling $\mathbf{X} \to \mathcal{Q}_{\mathbf{X}}$). Note that F does not depend on the spatial dimension d of the data set (except through library term I, which might increase with d as more differential operators become added). The total working memory W to store Ψ and $(\mathbf{G}^{(t)}, \mathbf{b}^{(t)})$ as outlined above is given by $W = IJ|\mathcal{Q}_{\mathbf{X}}|K_{\text{mem}} + (I+1)J|\mathcal{Q}_{\mathbf{X}}|$ double-precision floating point numbers (DPs).

Remark 13. There are several natural choices to consider to either decrease storage restrictions or increase computational speed. However, it is not clear that the anticipated savings will manifest. For instance, we could instead store the spatial Fourier transforms of the nonlinearities $\{\widehat{f_j(\mathbf{U}^{(t)})}\}_{j=1,t=\ell\Delta t}^{J,(\ell+K_{\text{mem}}-1)\Delta t}$, resulting in a working memory of $J \cdot K_{\text{mem}} \cdot |\mathbf{X}|$ instead of $IJ|\mathcal{Q}_{\mathbf{X}}|K_{\text{mem}}$ to store Ψ. This would require that we compute spatial convolutions over all K_{mem} time slices at each time point, instead of spatial convolutions over just the incoming time slice $\mathbf{U}^{(t)}$, hence resulting in a K_{mem}-fold increase in computation time, as this is the leading-order cost. In addition, the storage "savings" may actually be worse, specifically if $I|\mathcal{Q}_{\mathbf{X}}| \leq |\mathbf{X}|$. We believe that the method outlined above provides a near-optimal balance of computational complexity and storage requirements, with a heavier emphasis on reducing computational complexity.

5.C Proof of Lemma 6

Consider $\mathbf{w}$ such that one of the following holds:

(i) $\mathbf{w}$ is a local minimizer of (5.3.3)

(ii) $\mathbf{w} = H_\lambda\left(\mathbf{w} - \mathbf{G}^T(\mathbf{G}\mathbf{w} - \mathbf{b})\right)$

(iii) With $S = \operatorname{supp}(\mathbf{w})$, we have that $\mathbf{w}_S \in \arg\min_{\mathbf{z}} \|\mathbf{G}_S\mathbf{z} - \mathbf{b}\|_2^2$ and

$$\max_{i \in S^c} \left|\mathbf{G}_i^T(\mathbf{G}\mathbf{w} - \mathbf{b})\right| < \lambda \leq \min_{i \in S} |\mathbf{w}_i|.$$

Then it holds that $(ii) \iff (iii) \implies (i)$. Moreover, if $\mathbf{w}$ a global minimizer, then $(i) \implies (iii)$.

Proof. $(iii) \implies (ii)$ is immediate. To show $(ii) \implies (iii)$, let $S = \operatorname{supp}(\mathbf{w})$. Then we have

$$\mathbf{w}_S = \mathbf{w}_S - \mathbf{G}_S^T(\mathbf{G}\mathbf{w} - \mathbf{b}),$$

which implies that $\min_{i \in S} |\mathbf{w}_i| \geq \lambda$ so that $\mathbf{G}_S^T\mathbf{G}_S\mathbf{w}_S = \mathbf{G}_S^T\mathbf{b}$, so that $\mathbf{w}_S \in \arg\min_{\mathbf{z}} \|\mathbf{G}_S\mathbf{z} - \mathbf{b}\|_2^2$. On S^c we have

$$H_\lambda\left(\mathbf{G}_{S^c}^T(\mathbf{G}\mathbf{w} - \mathbf{b})\right) = 0 \implies \max_{i \in S^c} \left|\mathbf{G}_i^T(\mathbf{G}\mathbf{w} - \mathbf{b})\right| < \lambda.$$

To show that (ii) and (iii) imply (i), we note that under usual assumptions of two closed, convex and proper functions f and g, we have

$$\mathbf{w} \in \operatorname{prox}_g(\mathbf{w} - \partial f(\mathbf{w})) \iff 0 \in \partial f(\mathbf{w}) + \partial g(\mathbf{w}) \implies \mathbf{w} \in \operatorname{argmin}(f + g),$$

however $\|\cdot\|_0$ is clearly not convex[7]. Instead we can directly show that for a perturbed vector $\widetilde{\mathbf{w}} = \mathbf{w} + \boldsymbol{\eta}$, for suitably small $\|\boldsymbol{\eta}\|$ the objective is non-decreasing. Using that $\mathbf{w}_S \in \arg\min_{\mathbf{z}} \|\mathbf{G}_S\mathbf{z} - \mathbf{b}\|_2^2$, let $\mathbf{P}_S^\perp$ be the projection onto $\{\operatorname{span}(\mathbf{G}_S)\}^\perp$. The difference in objective F is then given by

$$F(\widetilde{\mathbf{w}}; \lambda) - F(\mathbf{w}; \lambda) = \frac{1}{2}\left(\left\|\mathbf{P}_S^\perp\mathbf{b} + \mathbf{G}\boldsymbol{\eta}\right\|_2^2 - \left\|\mathbf{P}_S^\perp\mathbf{b}\right\|_2^2\right) + \frac{\lambda^2}{2}\left(\|\widetilde{\mathbf{w}}\|_0 - \|\mathbf{w}\|_0\right)$$

$$= \frac{1}{2}\|\mathbf{G}\boldsymbol{\eta}\|_2^2 + \left\langle \mathbf{P}_S^\perp\mathbf{b}, \mathbf{G}\boldsymbol{\eta}\right\rangle + \frac{\lambda^2}{2}\left(\|\widetilde{\mathbf{w}}\|_0 - \|\mathbf{w}\|_0\right).$$

If $\operatorname{supp}(\boldsymbol{\eta}) \subset \operatorname{supp}(\mathbf{w})$ and $\|\boldsymbol{\eta}\|_\infty < \lambda$, then $\|\widetilde{\mathbf{w}}\|_0 = \|\mathbf{w}\|_0$ and $\left\langle \mathbf{P}_S^\perp\mathbf{b}, \mathbf{G}\boldsymbol{\eta}\right\rangle = 0$, hence $F(\widetilde{\mathbf{w}}; \lambda) - F(\mathbf{w}; \lambda) \geq 0$, with equality only if $\mathbf{G}\boldsymbol{\eta} = \mathbf{0}$, which in particular is not possible when $\mathbf{G}_S$ is full rank unless $\boldsymbol{\eta} = 0$. If $\operatorname{supp}(\boldsymbol{\eta}) \notin S$ and $\|\boldsymbol{\eta}\|_\infty < \lambda$, then $\mathbf{P}_S^\perp\mathbf{b} = \mathbf{0}$ implies a strict increase in F, while if $\mathbf{P}_S^\perp\mathbf{b} \neq \mathbf{0}$ then

$$\|\boldsymbol{\eta}_{S^c}\|_2 < \epsilon := \frac{\lambda^2}{2}\frac{1}{\left\|\mathbf{P}_S^\perp\mathbf{b}\right\|_2 \|\mathbf{G}_{S^c}\|_2},$$

implies a strict increase in F. To see this, note that $\left\langle \mathbf{P}_S^\perp, \mathbf{G}\boldsymbol{\eta}\right\rangle = \left\langle \mathbf{P}_S^\perp, \mathbf{G}_{S^c}\boldsymbol{\eta}_{S^c}\right\rangle$ implies the bound

$$F(\widetilde{\mathbf{w}}; \lambda) - F(\mathbf{w}; \lambda) \geq -\left\|\mathbf{P}_S^\perp\mathbf{b}\right\|_2 \|\mathbf{G}_{S^c}\|_2 \|\boldsymbol{\eta}_{S^c}\|_2 + \frac{\lambda^2}{2} > 0.$$

Note that ϵ is not tight. Combining these conditions gives a ball around $\mathbf{w}$ over which F is non-

[7] In fact the subdifferential $\partial\|\cdot\|_0(\mathbf{w}) = \emptyset$ unless $\mathbf{w} = \mathbf{0}$, upon which $\partial\|\cdot\|_0(\mathbf{w}) = \{\mathbf{0}\}$.

decreasing, hence $\mathbf{w}$ is a local min. Finally, that $\mathbf{w}$ a global minimizer implies (iii) can be found in [158]. □

5.D Proof of Theorem 2

For convenience, we restate the theorem here. Without loss of generality in the following we set $\Delta t = 1$.

Theorem 3. *Let $\sigma_{1,t}$ and $\sigma_{n,t}$ denote the first and last singular values of the matrix $\mathbf{G}^{(t)} \in \mathbb{R}^{m\times n}$. Assume the following: $\max_t \lambda_t \leq \overline{\lambda} < \infty$, $\min_t \sigma_{n,t} \geq \overline{\sigma}_{\min} > 0$, $\max_t \sigma_{1,t} \leq \overline{\sigma}_{\max}$, and $\sup_t \alpha_t < \sigma_{\max}^{-2}$, $\inf_t \alpha_t > 0$. In addition, assume that the global minimizer $\mathbf{w}^\star(t)$ of $F_t(\mathbf{w};\lambda_t)$ is unique for every t and satisfies $|S_t^\star| \geq \overline{s} > 0$ where $S^\star = supp\,(\mathbf{w}^\star(t))$. Finally, assume that the tracking gap is globally bounded: $\|\mathbf{w}^\star(t) - \mathbf{w}^\star(t+1)\|_2 := d_t \leq \overline{d}$. Then the dynamic regret (5.4.5) grows at-worst linearly:*

$$Reg_D(T) \leq C_1 + C_2 T$$

for some $C_1 > 0$ and $C_2 > 0$. In particular, $\frac{1}{T} Reg_D(T)$ remains bounded.

Proof. First we decompose $F_t(\mathbf{w};\lambda_t) = g_t(\mathbf{w}) + h_t(\mathbf{w}) = \left\|\mathbf{G}^{(t)}\mathbf{w} - \mathbf{b}^{(t)}\right\|_2^2 + \lambda^2 \|\mathbf{w}\|_0$. We can bound the difference in g_t as follows:

$$\begin{aligned} g_t(\mathbf{w}^{(t)}) - g_t(\mathbf{w}^\star(t)) &= \left\|\mathbf{G}^{(t)}\mathbf{w}^{(t)}\right\|_2^2 - \left\|\mathbf{G}^{(t)}\mathbf{w}^\star(t)\right\|_2^2 - 2\left\langle \mathbf{b}^{(t)}, \mathbf{G}^{(t)}\left(\mathbf{w}^{(t)} - \mathbf{w}^\star(t)\right)\right\rangle \\ &= \left\|\mathbf{G}^{(t)}(\mathbf{w}^{(t)} - \mathbf{w}^\star(t))\right\|_2^2 - 2\left\langle \mathbf{G}^{(t)}(\mathbf{w}^{(t)} - \mathbf{w}^\star(t)), \mathbf{G}^{(t)}\mathbf{w}^\star(t) - \mathbf{b}^{(t)}\right\rangle \end{aligned}$$

taking $|\cdot|$ of both sides and noting from the Lemma that $\left\|(\mathbf{G}^{(t)})^T(\mathbf{G}^{(t)}\mathbf{w}^\star(t) - \mathbf{b}^{(t)})\right\|_\infty < \lambda_t$ implies that

$$\begin{aligned} g_t(\mathbf{w}^{(t)}) - g_t(\mathbf{w}^\star(t)) &\leq \overline{\sigma}_{\max}^2 \left\|\mathbf{w}^{(t)} - \mathbf{w}^\star(t)\right\|_2^2 + 2\lambda_t\sqrt{|(S^\star)^c|}\left\|\mathbf{w}^{(t)} - \mathbf{w}^\star(t)\right\|_2 \\ &\leq \overline{\sigma}_{\max}^2 \left\|\mathbf{w}^{(t)} - \mathbf{w}^\star(t)\right\|_2^2 + 2\overline{\lambda}\sqrt{n - \overline{s}}\left\|\mathbf{w}^{(t)} - \mathbf{w}^\star(t)\right\|_2. \end{aligned}$$

For h_t we have simply

$$\left|h_t(\mathbf{w}^{(t)}) - h_t(\mathbf{w}^\star(t))\right| = \lambda_t^2 \left||S_t| - |S_t^\star|\right| \leq \overline{\lambda}^2(n - \overline{s}).$$

For any vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, it holds that

$$\|H_\lambda(\mathbf{x}) - H_\lambda(\mathbf{y})\|_2 \leq \|\mathbf{x} - \mathbf{y}\|_2 + \lambda\sqrt{|S_x \triangle S_y|}$$

where $S_x \triangle S_y = (S_x \cap S_y^c) \cup (S_y \cap S_x^c)$ is the symmetric difference of the sets $S_x = \text{supp}\,(H_\lambda(\mathbf{x}))$

and $S_y = \text{supp}\,(H_\lambda(\mathbf{y}))$. This implies, together with stationarity of $\mathbf{w}^\star(t)$,

$$\begin{aligned}
&\left\|\mathbf{w}^{(t+1)} - \mathbf{w}^\star(t)\right\|_2 \\
&= \left\|H_{\alpha_t\lambda_t}\left(\mathbf{w}^{(t)} - \alpha_t(\mathbf{G}^{(t)})^T\left(\mathbf{G}^{(t)}\mathbf{w}^{(t)} - \mathbf{b}^{(t)}\right)\right) - H_{\alpha_t\lambda_t}\left(\mathbf{w}^\star(t) - \alpha_t(\mathbf{G}^{(t)})^T\left(\mathbf{G}^{(t)}\mathbf{w}^\star(t) - \mathbf{b}^{(t)}\right)\right)\right\|_2 \\
&\leq \left\|\left(\mathbf{I} - \alpha_t(\mathbf{G}^{(t)})^T\mathbf{G}^{(t)}\right)(\mathbf{w}^{(t)} - \mathbf{w}^\star(t))\right\|_2 + \alpha_t\lambda_t\sqrt{|S_{t+1}\triangle S_t^\star|} \\
&\leq \max\left(|1-\alpha_t\sigma_{1,t}^2|, |1-\alpha_t\sigma_{n,t}^2|\right)\left\|\mathbf{w}^{(t)} - \mathbf{w}^\star(t)\right\|_2 + \alpha_t\lambda_t\sqrt{|S_{t+1}\triangle S_t^\star|} \\
&:= \rho_t\left\|\mathbf{w}^{(t)} - \mathbf{w}^\star(t)\right\|_2 + \alpha_t\lambda_t\sqrt{|S_{t+1}\triangle S_t^\star|}.
\end{aligned}$$

Using that $\left\|\mathbf{w}^{(t+1)} - \mathbf{w}^\star(t+1)\right\|_2 \leq \left\|\mathbf{w}^{(t+1)} - \mathbf{w}^\star(t)\right\|_2 + d_t$, we have the recurrence relation

$$\left\|\mathbf{w}^{(t+1)} - \mathbf{w}^\star(t+1)\right\|_2 \leq \rho_t\left\|\mathbf{w}^{(t)} - \mathbf{w}^\star(t)\right\|_2 + d_t + \alpha_t\lambda_t\sqrt{|S_{t+1}\triangle S_t^\star|}, \tag{5.D.1}$$

where, by assumptions on $\sigma_{1,t}, \sigma_{n,t}$ and α_t, it holds that $\max_t \rho_t \leq \overline{\rho}$ for some $\overline{\rho} < 1$, hence we get the bound

$$\left\|\mathbf{w}^{(t)} - \mathbf{w}^\star(t)\right\|_2 \leq \overline{\rho}^t\left\|\mathbf{w}^{(0)} - \mathbf{w}^\star(0)\right\|_2 + (\overline{d} + \overline{\alpha}\overline{\lambda}\sqrt{n})\sum_{s=0}^{t}\overline{\rho}^s \leq \overline{\rho}^t\left\|\mathbf{w}^{(0)} - \mathbf{w}^\star(0)\right\|_2 + \frac{\overline{d} + \overline{\alpha}\overline{\lambda}\sqrt{n}}{1-\overline{\rho}}. \tag{5.D.2}$$

We note in passing that this implies a uniform error bound on $\left\|\mathbf{w}^{(t)} - \mathbf{w}^\star(t)\right\|_2$ which asymptotically depends only on the tracking gap d_t and the support difference $|S_t\triangle S_t^\star|$. Finally, using this bound and previous calculations for g and h, we get

$$Reg_D(T) \leq \widetilde{C}_1\sum_{t=0}^{T}\overline{\rho}^t + C_2T \leq C_1 + C_2T$$

where

$$C_1 = \frac{\widetilde{C}_1}{1-\overline{\rho}} = \frac{\overline{\sigma}_{\max}^2\overline{\rho}}{1-\overline{\rho}}\left\|\mathbf{w}^{(0)} - \mathbf{w}^\star(0)\right\|_2^2 + \frac{2\left\|\mathbf{w}^{(0)} - \mathbf{w}^\star(0)\right\|_2}{(1-\overline{\rho})^2}\left(\overline{\lambda}\sqrt{n-\overline{s}} + \frac{(\overline{d} + \overline{\alpha}\overline{\lambda}\sqrt{n})\overline{\sigma}_{\max}^2}{1-\overline{\rho}}\right) \tag{5.D.3}$$

$$C_2 = (\overline{\sigma}_{\max}^2 - 1)\left(\frac{\overline{d} + \overline{\alpha}\overline{\lambda}\sqrt{n}}{1-\overline{\rho}}\right)^2 + \left(\frac{\overline{d} + \overline{\alpha}\overline{\lambda}\sqrt{n}}{1-\overline{\rho}} + \overline{\lambda}\sqrt{n-\overline{s}}\right)^2. \tag{5.D.4}$$

This completes the proof. □

Chapter 6

Conclusion

We have investigated a weak formulation of the equation discovery problem, finding that it is both generally applicable and advantageous for a wide range of scenarios. As is often the case in mathematics, the starting point is to relax assumptions, in this case extending differential equations to their integral counterparts. Beyond that, we were fortunate to find a subclass of test functions within the uncountably infinite space of possibilities that affords computational efficiency, high accuracy, and most importantly, robustness to noise. This has been demonstrated for ordinary differential equations, partial differential equations, and stochastic interacting particle systems, and the computational efficiency shows promise for online discovery of equations. Still, the utility of this method has not been brought to bear on experimental data, and fundamental theoretical limits have not been established. These are among the many future directions that are possible now having extensively verified our approach in the test cases presented here. We will now review several important next directions, and where possible, indicate progress that has been made.

6.1 Future Directions

6.1.1 Coarse-Graining and Homogenization

As demonstrated in [98], the weak form is capable of finding homogenized versions of equations that have highly-oscillatory components. In fact, in this example we discover equations directly from particle trajectories, hence we demonstrate two limits: the passage from particle to continuum, and the passage from a highly-oscillatory continuum description to a homogenized description. This and other mean-field equations can be seen as a form of coarse-graining, where the microscale information (unavailable to the observer) is averaged over, producing an equation only for the macroscale variables. Typically, the weak form is the key analytical tool used to pass from microscale to macroscale. Put another way, the weak form is already designed to "see into the macroscale" using data from intermediate scales. In future works we aim to demonstrate more broadly that this holds computationally and in practice.

6.1.2 Closure Modeling

A related line of inquiry to coarse-graining is closure modeling. In this case the true equations are unavailable (due to e.g. experimental limitations), however an effective equation in terms of observable variables is known apart from a few terms that need modeling. These terms typically can be well-approximated using gradients of state variables (as in turbulence modeling), but may involve more complex dependencies and higher-order derivatives. WSINDy may come to the fore in identifying key closure terms from the data, however it appears necessary to extend the current architecture to include terms that cannot have strong derivatives transfered to test functions using integrated by parts, such as $(\partial_x u)^2$. We discuss this more below.

6.1.3 Biological Applications

The original influence for considering a weak formulation of the discovery problem was biological data on collective cell migration. This scenario creates the perfect storm for consideration of weak formulations, as it is highly multiscale, contains many sources of noise, and results in non-smooth dynamics.

In these experiments, cells are grown in wells and driven to migrate into vacant areas of the wells. The path from experimental imaging to trajectories of cell nuclei over time is frought with noise from imperfect nuclei staining, image tracking errors, and intrinsic Brownian motion, obscuring the interactions between nearby cells. Noise at the cell level suggests a more macroscale approach: viewing the population as a continuum and modeling its evolution using a PDE. This is inherently related to the aims of the previous section on coarse-graining: a macroscale description of cells constitutes the effective dynamics (coarse-grained, mean-field, homogenized, etc.) for a medium driven by a volatile, yet possibly homogeneous, microstructure. Finally, due to sharp fronts between pixels with cells and those without cells, the dynamics at the PDE level are not smooth. Altogether, noise, coarse-graining, and lack of smoothness are generically handled by a weak formulation.

This is ongoing work with the members of Xuedong Liu's lab (CU Biochemistry), with many different directions in the works. Our first submitted work is [102] where we adapt WSINDy to identify the rules of interacting particle systems containing multiple species of particles. In this work we simultaneously perform model selection and classification of the population into different species. However, this is a methodology paper, and has not yet been applied to experimental data. Several experimental challenges in this direction are (i) growing cell cultures to high enough cellular density to generate cell-cell interactions, and (ii) capturing a time window of high collective motility, which is difficult to predict, and (iii) imaging the cells frequently enough to accurately infer dynamics. Luckily, understanding of these challenges is growing rapidly. We believe that it is only a matter of time before viable datasets are produced to conduct real inference of collective cell migration models using WSINDy.

6.2 Open Questions

6.2.1 Fundamental Limits of Recovery

The most important question we can hope to answer about weak-form equation discovery is the following: under what conditions is recovery of the correct model is guaranteed? Conditions may take the form of (i) sufficient data resolution in space and time, (ii) limitations on the level and type of noise corrupting the data, and (iii) bounds on the relative magnitudes of various coefficients and terms in the true model.

Related to (i) and (ii), we would like to know how coarse the data can be gathered and still recover the system. This is of fundamental importance to practitioners because experiments are rarely cheap to perform. The coupled dependence on noise and data resolution is at the heart of theoretical recovery concerns, and we hope to attack this problem in the future.

Also related to (i) and (ii), on the opposite side of the spectrum, is the question of convergence. In the infinite-data limit, does WSINDy provably recovery the system? In the case of interacting particle systems, we partially answered this, but importantly did not take sparse regression into account (only ordinary least squares). There does exist a consistency result for ℓ_1-minimization in the infite-data limit [62], however it is assumed that $\left\|\mathbf{G}_S^\dagger \mathbf{G}_{S^c}\right\|_1 < 1$ on the true support set S, which isn't possible to guarantee when S is unknown, and rarely holds in practice. Still, the overwhelming evidence that WSINDy is robust to high levels of corruption, and improves with data availability, suggests that the MSTLS algorithm employed throughout this dissertation has a provable range of recovery that includes a wide range of models.

Related to (iii) are two ideas. First is the subspace of solutions that the data is sampled from. In particular, if we have a term $\mathbf{w}^\star_{sj} D^{\alpha^s} f_j(u)$ that is outside of the dominant balance of the equation (i.e. $\mathbf{w}^\star_{sj} D^{\alpha^s} f_j(u) \ll \partial_t u$), how much corruption can the data handle and still identify that $\mathbf{w}^\star_{sj} \neq 0$? Or the opposite case of a system close to equilibrium: how small can $\partial_t u$ be and still identify the model? To the best of our knowledge, these questions have not yet been answered quantitatively, but greatly impact the usefulness of any system recovery algorithm in practice.

The second idea related to (iii) concerns the absolute magnitudes of the coefficients $\mathbf{w}^\star$. Many works on sparse regression for PDEs threshold on the magnitude, which may have no bearing on the *term* magnitude $\mathbf{w}^\star_{sj} D^{\alpha^s} f_j(u)$. By rescaling the data as in Chapter 3 we arrive at an intriguing paradigm, that one may threshold the coefficients *at any scale one choose*, so then the question becomes, which scales are appropriate? In any case, there is much work to be done at the interplay of sparse regression and scales in model recovery.

6.2.2 Systems that cannot be integrated by parts

WSINDy is currently only capable of eliminating numerical differentiation for terms that can be directly integrated by parts. In particular, terms that are nonlinear in *derivatives* of state variables, such as $(\partial_x u)^2$, have no representation $D^\alpha f(u)$ for linear (differential) operator D^α and nonlinear

function f (as far as I know). This is a limitation for example in turbulence modeling, as closure terms are often nonlinear in gradients of the velocity field. An obvious solution to this problem is to simply approximate the derivative from data ($\partial_x \mathbf{U} \approx \partial_x u$) as is done in strong formulations of the discovery problem, but then still apply the weak form without integrating by parts (i.e. compute $\psi * (\partial_x \mathbf{U})^2$). However, it remains to be seen what numerical techniques are most advantageous for computing $(\partial_x \mathbf{U})^2$, knowing that convolution with ψ will follow.

Another large class of systems that are not immediately amenable to the weak form are *discrete* dynamical systems. In particular, it is not clear how to integrate the system $\mathbf{u}^{n+1} = F(\mathbf{u}^n)$ against a test function, and whether this offers an advantage. However, there does exist a *summation* by parts formula:

$$\sum_{k=1}^{n} \psi^k \left(\mathbf{u}^{k+1} - \mathbf{u}^k\right) = \left(\psi^n \mathbf{u}^{n+1} - \psi^1 \mathbf{u}^1\right) - \sum_{k=1}^{n} \left(\psi^k - \psi^{k-1}\right) \mathbf{u}^k.$$

If the dynamical system has a recursive structure, $\mathbf{u}^{n+1} = \mathbf{u}^n + F(\mathbf{u}^n)$, it is possible to form the discrete-weak regression problem from the equation

$$-\sum_{k=1}^{n} \left(\psi^k - \psi^{k-1}\right) \mathbf{u}^k = \sum_{k=1}^{n} \psi^k F(\mathbf{u}^k)$$

for test functions ψ with $\psi^n = \psi^1 = 0$. However, in this case the left-hand side is equivalent to the left-hand side of the previous equation, so it is not clear that summation by parts offers any advantage.

6.2.3 Optimal Test Functions and Quadrature

The question of which test functions are best suited for certain problems and certain noise levels remains open. The test functions used throughout this dissertation were first found experimentally to yield high accuracy. We then proved why this was the case using to the Euler-Maclaurin formula. This did not immediately explain why the weak form works well to ameliorate noise. Using Fourier analysis, we realized a connection between the spectrum of the data and the spectrum of the test function, and developed a way of choosing test functions to damp high-frequency noise. We have aimed to provide constraints that are agnostic to the form of test function, namely constraints on the decay rate of the test function in both Fourier space and real space. This does not require the test function to be a piece-wise polynomial, so can easily be extended to another class of test functions.

It is clear that the test function must be chosen in tandem with a quadrature rule. The trapezoidal rule is clearly advantageous for uniformly-spaced data, both in speed and accuracy. While uniformly-spaced data is most common (e.g. pixels are uniformly spaced, and time sampling is often uniform), adapting WSINDy to the case of non-uniformly-spaced data would greatly widen

its scope. For instance, it is well-known that wall-bounded turbulence and other flow-separation problems cannot be simulated on a uniform grid. There are at least two ways to adapt the current framework to these problems. The first is to find test functions and a quadrature rule that are specific to the grid available. The other is to interpolate the data back onto a uniform grid, and then use framework we've developed. An interpolation problem can be made linear in the data, so we can directly compute the effect on the noise distribution resulting from interpolation. Consider $\mathbf{U} = u(\mathbf{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$ i.i.d. where $\mathbf{x}$ is a possibly non-uniform grid, and we would like to interpolate onto a uniform grid $\overline{\mathbf{x}}$. With interpolation operator Φ, the resulting data $\overline{\mathbf{U}}$ satisfies

$$\overline{\mathbf{U}} = \Phi\mathbf{U} = \Phi u(\mathbf{x}) + \eta,$$

where

$$\eta = \Phi\epsilon \sim \mathcal{N}(0, \sigma^2\Phi\Phi^T).$$

Now we see that we are in the framework of *correlated noise*, with variance depending on the conditioning of Φ. This relates back to the previous section of fundamental limits of discovery: how correlated can the noise be and still discover the system? This is one of the many possible future directions to take this work.

Bibliography

[1] H. Akaike. A new look at the statistical model identification. IEEE Transactions on Automatic Control, 19(6):716–723, December 1974.

[2] Hirotugu Akaike. On entropy maximization principle. In P. R. Krishnaiah, editor, Applications of Statistics, pages 27–41. North Holland, Amsterdam, Netherlands, 1977.

[3] E Paulo Alves and Frederico Fiuza. Data-driven discovery of reduced plasma physics models from fully-kinetic simulations. arXiv preprint arXiv:2011.01927, 2020.

[4] Dyego Araújo, Roberto I Oliveira, and Daniel Yukimura. A mean-field limit for certain deep neural networks. arXiv preprint arXiv:1906.00193, 2019.

[5] Aleksandr Y Aravkin, Robert Baraldi, and Dominique Orban. A proximal quasi-newton trust-region method for nonsmooth regularized optimization. arXiv preprint arXiv:2103.15993, 2021.

[6] Joseph Bakarji and Daniel M Tartakovsky. Data-driven discovery of coarse-grained equations. Journal of Computational Physics, 434:110219, 2021.

[7] GI Barenblatt. On some unsteady fluid and gas motions in a porous medium. Prikl. Mat. Mekh, 16(1):67–78, 1952.

[8] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. Operations research, 63(5):1227–1244, 2015.

[9] Gregory Beylkin and Martin J Mohlenkamp. Algorithms for numerical analysis in high dimensions. SIAM Journal on Scientific Computing, 26(6):2133–2159, 2005.

[10] Dapeng Bi, Xingbo Yang, M Cristina Marchetti, and M Lisa Manning. Motility-driven glass and jamming transitions in biological tissues. Physical Review X, 6(2):021011, 2016.

[11] Bo Martin Bibby and Michael Sørensen. Martingale estimation functions for discretely observed diffusion processes. Bernoulli, pages 17–39, 1995.

[12] Jaya PN Bishwal. Parameter estimation in stochastic differential equations. Springer, 2007.

[13] Jaya Prakash Narayan Bishwal et al. Estimation in interacting diffusions: Continuous and discrete sampling. Applied Mathematics, 2(9):1154–1158, 2011.

[14] Vincent D Blondel, Julien M Hendrickx, and John N Tsitsiklis. Continuous-time average-preserving opinion dynamics with opinion-dependent communications. SIAM Journal on Control and Optimization, 48(8):5214–5240, 2010.

[15] Thomas Blumensath and Mike E Davies. Iterative thresholding for sparse approximations. Journal of Fourier analysis and Applications, 14(5):629–654, 2008.

[16] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. Applied and computational harmonic analysis, 27(3):265–274, 2009.

[17] Niklas Boers and Peter Pickl. On mean field limits for dynamical systems. Journal of Statistical Physics, 164(1):1–16, 2016.

[18] François Bolley, José A Canizo, and José A Carrillo. Stochastic mean-field limit: non-lipschitz forces and swarming. Mathematical Models and Methods in Applied Sciences, 21(11):2179–2210, 2011.

[19] Mattia Bongini, Massimo Fornasier, Markus Hansen, and Mauro Maggioni. Inferring interaction rules from observations of evolutive systems i: The variational approach. Mathematical Models and Methods in Applied Sciences, 27(05):909–951, 2017.

[20] Lorenzo Boninsegna, Feliks Nüske, and Cecilia Clementi. Sparse learning of stochastic dynamical equations. The Journal of chemical physics, 148(24):241723, 2018.

[21] D. M. Bortz and P. W. Nelson. Model Selection and Mixed-Effects Modeling of HIV Infection Dynamics. Bulletin of Mathematical Biology, 68(8):2005–2025, November 2006.

[22] David B Brückner, Pierre Ronceray, and Chase P Broedersz. Inferring the dynamics of underdamped stochastic systems. Physical review letters, 125(5):058103, 2020.

[23] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the national academy of sciences, 113(15):3932–3937, 2016.

[24] T Tony Cai and Lie Wang. Orthogonal matching pursuit for sparse signal recovery with noise. IEEE Transactions on Information theory, 57(7):4680–4688, 2011.

[25] Jared L Callaham, J-C Loiseau, Georgios Rigas, and Steven L Brunton. Nonlinear stochastic modelling with langevin regression. Proceedings of the Royal Society A, 477(2250):20210092, 2021.

[26] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. Journal of Fourier analysis and applications, 14(5):877–905, 2008.

[27] Alejandro Carderera, Sebastian Pokutta, Christof Schütte, and Martin Weiser. Cindy: Conditional gradient-based identification of non-linear dynamics–noise-robust recovery. arXiv preprint arXiv:2101.02630, 2021.

[28] JA Carrillo, MG Delgadino, and FS Patacchini. Existence of ground states for aggregation-diffusion equations. Analysis and applications, 17(03):393–423, 2019.

[29] Chris Chatfield. The analysis of time series: an introduction. Chapman and hall/CRC, 2003.

[30] Goong Chen. Control and stabilization for the wave equation in a bounded domain. SIAM Journal on Control and Optimization, 17(1):66–81, 1979.

[31] Xiaohui Chen. Maximum likelihood estimation of potential energy in interacting particle systems from single-trajectory data. Electronic Communications in Probability, 26:1–13, 2021.

[32] Xiaoli Chen, Jinqiao Duan, and George Em Karniadakis. Learning and meta-learning of stochastic advection–diffusion–reaction systems from sparse measurements. European Journal of Applied Mathematics, 32(3):397–420, 2021.

[33] Xiaoli Chen, Liu Yang, Jinqiao Duan, and George Em Karniadakis. Solving inverse stochastic problems from discrete particle observations using the fokker–planck equation and physics-informed neural networks. SIAM Journal on Scientific Computing, 43(3):B811–B830, 2021.

[34] Alexandre Cortiella, Kwang-Chun Park, and Alireza Doostan. Sparse identification of nonlinear dynamical systems via reweighted ℓ_1-regularized least squares. Computer Methods in Applied Mechanics and Engineering, 376:113620, 2021.

[35] James P Crutchfield and Bruce S McNamara. Equations of motion from a data series. Complex systems, 1(417-452):121, 1987.

[36] Germund Dahlquist and Ake Bjorck. Numerical Methods in Scientific Computing: Volume 1, volume 103. Siam, 2008.

[37] Emiliano Dall'Anese, Andrea Simonetto, Stephen Becker, and Liam Madden. Optimization and learning with information streams: Time-varying algorithms and applications. IEEE Signal Processing Magazine, 37(3):71–83, 2020.

[38] Rishabh Dixit, Amrit Singh Bedi, Ruchi Tripathi, and Ketan Rajawat. Online learning with inexact proximal online gradient descent algorithms. IEEE Transactions on Signal Processing, 67(5):1338–1352, 2019.

[39] Jean Dolbeault and Benoît Perthame. Optimal critical mass in the two dimensional keller–segel model in r2. Comptes Rendus Mathematique, 339(9):611–616, 2004.

[40] David L Donoho, Yaakov Tsaig, Iddo Drori, and Jean-Luc Starck. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. IEEE transactions on Information Theory, 58(2):1094–1121, 2012.

[41] SN Ershov. B-splines and bernstein basis polynomials. Physics of Particles and Nuclei Letters, 16(6):593–601, 2019.

[42] Jianqing Fan and Yuan Liao. Endogeneity in high dimensions. Annals of statistics, 42(3):872, 2014.

[43] R Fante. Transmission of electromagnetic waves into time-varying media. IEEE Transactions on Antennas and Propagation, 19(3):417–424, 1971.

[44] Salar Fattahi, Nikolai Matni, and Somayeh Sojoudi. Learning sparse dynamical systems from a single sample trajectory. In 2019 IEEE 58th Conference on Decision and Control (CDC), pages 2682–2689. IEEE, 2019.

[45] L Felsen and G Whitman. Wave propagation in time-varying media. IEEE Transactions on Antennas and Propagation, 18(2):242–253, 1970.

[46] Jinchao Feng, Yunxiang Ren, and Sui Tang. Data-driven discovery of interacting particle systems using gaussian processes. arXiv preprint arXiv:2106.02735, 2021.

[47] Razvan C Fetecau, Hui Huang, Daniel Messenger, and Weiran Sun. Zero-diffusion limit for aggregation equations over bounded domains. arXiv preprint arXiv:1809.01763, 2018.

[48] Razvan C Fetecau, Hui Huang, and Weiran Sun. Propagation of chaos for the keller–segel equation over bounded domains. Journal of Differential Equations, 266(4):2142–2174, 2019.

[49] Razvan C Fetecau, Yanghong Huang, and Theodore Kolokolnikov. Swarm dynamics and equilibria for a nonlocal aggregation model. Nonlinearity, 24(10):2681, 2011.

[50] Razvan C Fetecau and Mitchell Kovacic. Swarm equilibria in domains with boundaries. SIAM Journal on Applied Dynamical Systems, 16(3):1260–1308, 2017.

[51] Dylan Foster, Tuhin Sarkar, and Alexander Rakhlin. Learning nonlinear dynamical systems from a single trajectory. In Learning for Dynamics and Control, pages 851–861. PMLR, 2020.

[52] Simon Foucart and Holger Rauhut. A Mathematical Introduction to Compressive Sensing. Birkhäuser Basel, 2013.

[53] David Freedman and Persi Diaconis. On the histogram as a density estimator: L 2 theory. Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete, 57(4):453–476, 1981.

[54] Paraskevi Gkeka, Gabriel Stoltz, Amir Barati Farimani, Zineb Belkacemi, Michele Ceriotti, John D Chodera, Aaron R Dinner, Andrew L Ferguson, Jean-Bernard Maillet, Hervé Minoux, et al. Machine learning force fields and coarse-grained variables in molecular dynamics: application to materials and biological systems. Journal of Chemical Theory and Computation, 16(8):4757–4775, 2020.

[55] Susana N Gomes, Andrew M Stuart, and Marie-Therese Wolfram. Parameter estimation for macroscopic pedestrian dynamics models from microscopic data. SIAM Journal on Applied Mathematics, 79(4):1475–1500, 2019.

[56] Jiawei Guo. The progress of three astrophysics simulation methods: Monte-carlo, pic and mhd. In Journal of Physics: Conference Series, volume 2012, page 012136. IOP Publishing, 2021.

[57] Daniel R Gurevich, Patrick AK Reinbold, and Roman O Grigoriev. Robust and optimal sparse regression for nonlinear pde models. Chaos: An Interdisciplinary Journal of Nonlinear Science, 29(10):103113, 2019.

[58] Robert J Harrison, Gregory Beylkin, Florian A Bischoff, Justus A Calvin, George I Fann, Jacob Fosso-Tande, Diego Galindo, Jeff R Hammond, Rebecca Hartman-Baker, Judith C Hill, et al. Madness: A multiresolution, adaptive numerical environment for scientific simulation. SIAM Journal on Scientific Computing, 38(5):S123–S142, 2016.

[59] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer, 2009.

[60] Elad Hazan. Efficient algorithms for online convex optimization and their applications. Princeton University, 2006.

[61] Elad Hazan and Satyen Kale. On stochastic and worst-case models for investing. Advances in Neural Information Processing Systems, 22, 2009.

[62] Yuchen He, Namjoon Suh, Xiaoming Huo, Sungha Kang, and Yajun Mei. Asymptotic theory of ℓ_1-regularized pde identification from a single noisy trajectory. arXiv preprint arXiv:2103.07045, 2021.

[63] Moritz Hoffmann, Christoph Fröhner, and Frank Noé. Reactive sindy: Discovering governing reactions from concentration data. The Journal of Chemical Physics, 150(2):025101, 2019.

[64] Steven C.H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. Neurocomputing, 459:249–289, October 2021.

[65] Pierre-Emmanuel Jabin and Zhenfu Wang. Mean field limit for stochastic particle systems. In Active Particles, Volume 1, pages 379–402. Springer, 2017.

[66] Jun-Gi Jang and U Kang. D-tucker: Fast and memory-efficient tucker decomposition for dense tensors. In 2020 IEEE 36th International Conference on Data Engineering (ICDE), pages 1850–1853. IEEE, 2020.

[67] Jialei Wang, Peilin Zhao, Steven C. H. Hoi, and Rong Jin. Online Feature Selection and Its Applications. IEEE Transactions on Knowledge and Data Engineering, 26(3):698–710, March 2014.

[68] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. Proceedings of the Royal Society A, 474(2219):20180335, 2018.

[69] Sung Ha Kang, Wenjing Liao, and Yingjie Liu. Ident: Identifying differential equations with numerical time evolution. Journal of Scientific Computing, 87(1):1–27, 2021.

[70] Takeaki Kariya and Hiroshi Kurata. Generalized least squares. John Wiley & Sons, 2004.

[71] Raphael A Kasonga. Maximum likelihood theory for large interacting systems. SIAM Journal on Applied Mathematics, 50(3):865–875, 1990.

[72] Aly-Khan Kassam and Lloyd N Trefethen. Fourth-order time-stepping for stiff pdes. SIAM Journal on Scientific Computing, 26(4):1214–1233, 2005.

[73] Dustin D Keck and David M Bortz. Generalized sensitivity functions for size-structured population models. Journal of Inverse and Ill-posed Problems, 24(3):309–321, 2016.

[74] Evelyn F Keller and Lee A Segel. Model for chemotaxis. Journal of theoretical biology, 30(2):225–234, 1971.

[75] Rachael T Keller and Qiang Du. Discovery of dynamics using linear multistep methods. SIAM Journal on Numerical Analysis, 59(1):429–455, 2021.

[76] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association, 107(500):1590–1598, 2012.

[77] Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. In 2018 IEEE conference on decision and control (CDC), pages 6059–6066. IEEE, 2018.

[78] Yannis Kopsinis, Konstantinos Slavakis, and Sergios Theodoridis. Online Sparse System Identification and Signal Reconstruction Using Projections Onto Weighted ℓ_1 Balls. IEEE Transactions on Signal Processing, 59(3):936–952, March 2011.

[79] John H Lagergren, John T Nardini, Ruth E Baker, Matthew J Simpson, and Kevin B Flores. Biologically-informed neural networks guide mechanistic modeling from sparse experimental data. PLoS computational biology, 16(12):e1008462, 2020.

[80] John H. Lagergren, John T. Nardini, G. Michael Lavigne, Erica M. Rutter, and Kevin B. Flores. Learning partial differential equations for biological transport models from noisy spatio-temporal data. Proc. R. Soc. A., 476(2234):20190800, February 2020.

[81] Quanjun Lang and Fei Lu. Learning interaction kernels in mean-field equations of first-order systems of interacting particles. SIAM Journal on Scientific Computing, 44(1):A260–A285, 2022.

[82] Tony Lelievre and Gabriel Stoltz. Partial differential equations and stochastic methods in molecular dynamics. Acta Numerica, 25:681–880, 2016.

[83] Yang Li and Jinqiao Duan. Extracting governing laws from sample path data of non-gaussian stochastic dynamical systems. Journal of Statistical Physics, 186(2):1–21, 2022.

[84] Zhongyang Li, Fei Lu, Mauro Maggioni, Sui Tang, and Cheng Zhang. On the identifiability of interaction functions in systems of interacting particles. Stochastic Processes and their Applications, 132:135–163, 2021.

[85] Gabriele Lillacci and Mustafa Khammash. Parameter Estimation and Model Selection in Computational Biology. PLoS Comput Biol, 6(3):e1000696, March 2010.

[86] L. Ljung. System identification: theory for the user. 2nd edition Prentice-Hall, Upper Saddle River, NJ, 1999.

[87] Andrew W Lo. Maximum likelihood estimation of generalized itô processes with discretely sampled data. Econometric Theory, 4(2):231–247, 1988.

[88] Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. Journal of Computational Physics, 399:108925, 2019.

[89] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In International Conference on Machine Learning, pages 3208–3216. PMLR, 2018.

[90] Fei Lu, Mauro Maggioni, and Sui Tang. Learning interaction kernels in heterogeneous systems of agents from multiple trajectories. J. Mach. Learn. Res., 22:32–1, 2021.

[91] Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In International Conference on Machine Learning, pages 3276–3285. PMLR, 2018.

[92] Ryan Lukeman, Yue-Xian Li, and Leah Edelstein-Keshet. Inferring individual rules from collective behavior. Proceedings of the National Academy of Sciences, 107(28):12576–12580, 2010.

[93] Suryanarayana Maddu, Bevan L Cheeseman, Ivo F Sbalzarini, and Christian L Müller. Stability selection enables robust learning of partial differential equations from limited noisy data. arXiv preprint arXiv:1907.07810, 2019.

[94] Osman Asif Malik and Stephen Becker. Low-rank tucker decomposition of large tensors using tensorsketch. Advances in neural information processing systems, 31:10096–10106, 2018.

[95] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. The annals of statistics, 34(3):1436–1462, 2006.

[96] Sylvie Méléard. Asymptotic behaviour of some interacting particle systems; mckean-vlasov and boltzmann models. In Probabilistic models for nonlinear partial differential equations, pages 42–95. Springer, 1996.

[97] Daniel Messenger and Razvan C Fetecau. Equilibria of an aggregation model with linear diffusion in domains with boundaries. Mathematical Models and Methods in Applied Sciences, 30(04):805–845, 2020.

[98] Daniel A Messenger and David M Bortz. Learning mean-field equations from particle data using WSINDy. arXiv preprint arXiv:2110.07756 (in press at Physica D: Nonlinear Phenomena), 2021.

[99] Daniel A. Messenger and David M. Bortz. Weak SINDy For Partial Differential Equations. J. Comput. Phys., 443:110525, October 2021.

[100] Daniel A. Messenger and David M. Bortz. Weak SINDy: Galerkin-Based Data-Driven Model Selection. SIAM Multiscale Model. Simul., 19(3):1474–1497, 2021.

[101] Daniel A Messenger, Emiliano Dall'Anese, and David M Bortz. Online weak-form sparse identification of partial differential equations. arXiv preprint arXiv:2203.03979, 2022.

[102] Daniel A Messenger, Graycen E Wheeler, Xuedong Liu, and David M Bortz. Learning anisotropic interaction rules from individual trajectories in a heterogeneous cellular population. arXiv preprint arXiv:2204.14141, 2022.

[103] Grigorii Noikhovich Milstein. Numerical integration of stochastic differential equations, volume 313. Springer Science & Business Media, 1994.

[104] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. Digital Signal Processing, 73:1–15, 2018.

[105] John T Nardini, Ruth E Baker, Matthew J Simpson, and Kevin B Flores. Learning differential equation models from stochastic agent-based model simulations. Journal of the Royal Society Interface, 18(176):20200987, 2021.

[106] John T Nardini, John H Lagergren, Andrea Hawkins-Daarud, Lee Curtin, Bethan Morris, Erica M Rutter, Kristin R Swanson, and Kevin B Flores. Learning equations from biological data with limited time samples. Bulletin of mathematical biology, 82(9):1–33, 2020.

[107] Mila Nikolova. Description of the minimizers of least squares regularized with ℓ_0-norm. uniqueness of the global minimizer. SIAM Journal on Imaging Sciences, 6(2):904–937, 2013.

[108] Zhen-Hu Ning and Qing-Xu Yan. Stabilization of the wave equation with variable coefficients and a delay in dissipative boundary feedback. Journal of Mathematical Analysis and Applications, 367(1):167–173, 2010.

[109] Houman Owhadi. Bayesian numerical homogenization. Multiscale Modeling & Simulation, 13(3):812–828, 2015.

[110] R.E. Pattle. Diffusion from an instantaneous point source with a concentration-dependent coefficient. The Quarterly Journal of Mechanics and Applied Mathematics, 12(4):407–409, 1959.

[111] Victor Pereyra and G Scherer. Efficient computer manipulation of tensor products with applications to multidimensional approximation. Mathematics of Computation, 27(123):595–605, 1973.

[112] Tong Qin, Kailiang Wu, and Dongbin Xiu. Data driven governing equations approximation using deep neural networks. Journal of Computational Physics, 395:620–635, 2019.

[113] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using Gaussian processes. Journal of Computational Physics, 348:683–693, 2017.

[114] Patrick AK Reinbold, Daniel R Gurevich, and Roman O Grigoriev. Using noisy or incomplete data to discover models of spatiotemporal dynamics. Physical Review E, 101(1):010203, 2020.

[115] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5):206–215, 2019.

[116] Samuel Rudy, Alessandro Alla, Steven L Brunton, and J Nathan Kutz. Data-driven identification of parametric partial differential equations. SIAM Journal on Applied Dynamical Systems, 18(2):643–660, 2019.

[117] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. Science Advances, 3(4):e1602614, 2017.

[118] Samuel H Rudy, J Nathan Kutz, and Steven L Brunton. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. Journal of Computational Physics, 396:483–506, 2019.

[119] Yahya Sattar and Samet Oymak. Non-asymptotic and accurate learning of nonlinear dynamical systems. arXiv preprint arXiv:2002.08538, 2020.

[120] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 473(2197):20160446, 2017.

[121] Hayden Schaeffer and Scott G McCalla. Sparse model selection via integral terms. Physical Review E, 96(2):023302, 2017.

[122] Hayden Schaeffer, Giang Tran, and Rachel Ward. Extracting sparse high-dimensional dynamics from limited data. SIAM Journal on Applied Mathematics, 78(6):3279–3295, 2018.

[123] Hayden Schaeffer, Giang Tran, Rachel Ward, and Linan Zhang. Extracting structured dynamical systems using sparse optimization with very few samples. Multiscale Modeling & Simulation, 18(4):1435–1461, 2020.

[124] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. science, 324(5923):81–85, 2009.

[125] Néstor Sepúlveda, Laurence Petitjean, Olivier Cochet, Erwan Grasland-Mongrain, Pascal Silberzan, and Vincent Hakim. Collective cell motion in an epithelial sheet can be quantitatively described by a stochastic interacting particle model. PLoS computational biology, 9(3):e1002944, 2013.

[126] Brian Seymour and Eric Varley. Exact representations for acoustical waves when the sound speed varies in space and time. Studies in Applied Mathematics, 76(1):1–35, 1987.

[127] Louis Sharrock, Nikolas Kantas, Panos Parpas, and Grigorios A Pavliotis. Parameter estimation for the mckean-vlasov stochastic differential equation. arXiv preprint arXiv:2106.13751, 2021.

[128] Max Simchowitz, Horia Mania, Stephen Tu, Michael I Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In Conference On Learning Theory, pages 439–473. PMLR, 2018.

[129] Lizhe Sun, Mingyuan Wang, Yangzi Guo, and Adrian Barbu. A novel framework for online supervised learning with feature selection. arXiv preprint arXiv:1803.11521, 2018.

[130] Yiming Sun, Yang Guo, Charlene Luo, Joel Tropp, and Madeleine Udell. Low-rank tucker approximation of a tensor from streaming data. SIAM Journal on Mathematics of Data Science, 2(4):1123–1150, 2020.

[131] Rohit Supekar, Boya Song, Alasdair Hastewell, Alexander Mietke, and Jörn Dunkel. Learning hydrodynamic equations for active matter from particle simulations and experiments. arXiv preprint arXiv:2101.06568, 2021.

[132] Alain-Sol Sznitman. Topics in propagation of chaos. In Ecole d'été de probabilités de Saint-Flour XIX—1989, pages 165–251. Springer, 1991.

[133] Kunihiko Taira and T. Colonius. The immersed boundary method: A projection approach. J. Comput. Phys., 225(2):2118–2137, August 2007.

[134] Karl Thomaseth and Claudio Cobelli. Generalized sensitivity functions in physiological system identification. Annals of biomedical engineering, 27(5):607–616, 1999.

[135] Benjamin A Toms, Elizabeth A Barnes, and Imme Ebert-Uphoff. Physically interpretable neural networks for the geosciences: Applications to earth system variability. Journal of Advances in Modeling Earth Systems, 12(9):e2019MS002002, 2020.

[136] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael P.H Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. J. R. Soc. Interface., 6(31):187–202, February 2009.

[137] Giang Tran and Rachel Ward. Exact recovery of chaotic systems from highly corrupted data. Multiscale Modeling & Simulation, 15(3):1108–1129, 2017.

[138] Paul Van Liedekerke, MM Palm, N Jagiella, and Dirk Drasdo. Simulating tissue mechanics with agent-based models: concepts, perspectives and some novel results. Computational particle mechanics, 2(4):401–444, 2015.

[139] Javier Vila, Raj Kumar Pal, Massimo Ruzzene, and Giuseppe Trainiti. A bloch-based procedure for dispersion analysis of lattices with periodic time-varying properties. Journal of Sound and Vibration, 406:363–377, 2017.

[140] Wen-Xu Wang, Rui Yang, Ying-Cheng Lai, Vassilios Kovanis, and Celso Grebogi. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. Physical review letters, 106(15):154101, 2011.

[141] Yating Wang, Siu Wun Cheung, Eric T Chung, Yalchin Efendiev, and Min Wang. Deep multiscale model learning. Journal of Computational Physics, 406:109071, 2020.

[142] Z Wang, X Huan, and K Garikipati. Variational system identification of the partial differential equations governing microstructure evolution in materials: Inference over sparse and spatially unrelated data. Computer Methods in Applied Mechanics and Engineering, 377:113706, 2021.

[143] Zhenlin Wang, Xun Huan, and Krishna Garikipati. Variational system identification of the partial differential equations governing the physics of pattern-formation: inference under varying fidelity and noise. Computer Methods in Applied Mechanics and Engineering, 356:44–74, 2019.

[144] Zhenlin Wang, Bowei Wu, Krishna Garikipati, and Xun Huan. A perspective on regression and bayesian approaches for system identification of pattern formation dynamics. Theoretical and Applied Mechanics Letters, 10(3):188–194, 2020.

[145] David J. Warne, Ruth E. Baker, and Matthew J. Simpson. Using Experimental Data and Information Criteria to Guide Model Selection for Reaction–Diffusion Problems in Mathematical Biology. Bull Math Biol, 81(6):1760–1804, June 2019.

[146] Michael S Warren and John K Salmon. Astrophysical n-body simulations using hierarchical tree data structures. Proceedings of Supercomputing, 1992.

[147] E Weinan. Principles of multiscale modeling. Cambridge University Press, 2011.

[148] Hulin Wu and Lang Wu. Identification of significant host factors for HIV dynamics modelled by non-linear mixed-effects models. Statist. Med., 21(5):753–771, March 2002.

[149] Kailiang Wu, Tong Qin, and Dongbin Xiu. Structure-preserving method for reconstructing unknown hamiltonian systems from trajectory data. SIAM Journal on Scientific Computing, 42(6):A3704–A3729, 2020.

[150] Kailiang Wu and Dongbin Xiu. Numerical aspects for approximating governing equations using data. Journal of Computational Physics, 384:200–221, 2019.

[151] Kailiang Wu and Dongbin Xiu. Data-driven deep learning of partial differential equations in modal space. Journal of Computational Physics, 408:109307, 2020.

[152] Hao Xu, Haibin Chang, and Dongxiao Zhang. Dlga-pde: Discovery of pdes with incomplete candidate library via combination of deep learning and genetic algorithm. Journal of Computational Physics, page 109584, 2020.

[153] Zhenhuan Yang, Baojian Zhou, Yunwen Lei, and Yiming Ying. Stochastic Hard Thresholding Algorithms for AUC Maximization. In 2020 IEEE International Conference on Data Mining (ICDM), pages 741–750, Sorrento, Italy, November 2020. IEEE.

[154] Yilun Chen, Yuantao Gu, and Alfred O. Hero. Sparse LMS for system identification. In 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 3125–3128, Taipei, Taiwan, April 2009. IEEE.

[155] Wenjian Yu, Yu Gu, and Yaohang Li. Efficient randomized algorithms for the fixed-precision low-rank matrix approximation. SIAM Journal on Matrix Analysis and Applications, 39(3):1339–1359, 2018.

[156] Yuantao Gu, Jian Jin, and Shunliang Mei. ℓ_0 norm constraint LMS algorithm for sparse system identification. IEEE Signal Processing Letters, 16(9):774–777, September 2009.

[157] Tingting Zhai, Frederic Koriche, Hao Wang, and Yang Gao. Tracking Sparse Linear Classifiers. IEEE Transactions on Neural Networks and Learning Systems, 30(7):2079–2092, July 2019.

[158] Linan Zhang and Hayden Schaeffer. On the convergence of the SINDy algorithm. Multiscale Modeling & Simulation, 17(3):948–972, 2019.

[159] Sheng Zhang and Guang Lin. Robust data-driven discovery of governing physical laws with error bars. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 474(2217):20180305, 2018.

[160] Sheng Zhang and Guang Lin. Robust subsampling-based sparse Bayesian inference to tackle four challenges (large noise, outliers, data integration, and extrapolation) in the discovery of physical laws from. arXiv preprint arXiv:1907.07788, 2019.

[161] Peng Zheng, Travis Askham, Steven L Brunton, J Nathan Kutz, and Aleksandr Y Aravkin. A unified framework for sparse relaxed regularized regression: Sr3. IEEE Access, 7:1404–1423, 2018.

[162] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In Proceedings of the 20th international conference on machine learning (icml-03), pages 928–936, 2003.