



Weak SINDy for partial differential equations

Daniel A. Messenger*, David M. Bortz



Department of Applied Mathematics, University of Colorado Boulder, 11 Engineering Dr., Boulder, CO 80309, USA

ARTICLE INFO

Article history:

Available online 23 June 2021

Keywords:

Data-driven model selection
Partial differential equations
Weak solutions
Sparse recovery
Galerkin method
Convolution

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 [6,39]. 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 [28] 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.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Stemming from Akaike's seminal work in the 1970's [1,2], research into the automatic creation of accurate mathematical models from data has progressed dramatically. In the last 20 years, substantial developments have been made at the interface of applied mathematics and statistics to design data-driven model selection algorithms that are both statistically rigorous and computationally efficient (see [5,22,23,48,55,56] for both theory and applications). An important achievement in this field was the formulation and subsequent discretization of the system discovery problem in terms of a candidate basis of nonlinear functions evaluated at the given dataset, together with a sparsification measure to avoid overfitting [9]. In [50] the authors extended this framework to the context of catastrophe prediction and used compressed sensing techniques to enforce sparsity. More recently, this approach has been generalized as the SINDy algorithm (Sparse Identification of Nonlinear Dynamics) [6] and successfully used to identify a variety of discrete and continuous dynamical systems.

* Corresponding author.

E-mail addresses: daniel.messenger@colorado.edu (D.A. Messenger), dmbortz@colorado.edu (D.M. Bortz).

The wide applicability, computational efficiency, and interpretability of the SINDy algorithm has spurred an explosion of interest in the problem of identifying nonlinear dynamical systems from data [8,34,10,11,15,49,27]. In addition to the sparse regression approach adopted in SINDy, some of the primary techniques include Gaussian process regression [31,36], deep neural networks [26,25,24,40,51,22], Bayesian inference [61,62,54] and classical methods from numerical analysis [16,19,57]. The variety of approaches for model discovery from data qualitatively differ in the interpretability of the resulting data-driven dynamical system, the computational efficiency of the algorithm, and the robustness to noise, scale separation, etc. For instance, a neural-network based data-driven dynamical system does not easily lend itself to physical interpretation.¹ The SINDy algorithm allows for direct interpretations of the dynamics from identified differential equations and uses sequential-thresholding least-squares (STLS) to enforce a sparse solution $x \in \mathbb{R}^n$ to a linear system $Ax = b$. STLS has been proven to converge to a local minimizer of the non-convex functional $F(x) = \|Ax - b\|_2^2 + \lambda^2 \|x\|_0$ in at-most n iterations [60].

The aim of the present article is to extend the Weak SINDy method (WSINDy) for recovering ordinary differential equations (ODEs) from data to the context of partial differential equations (PDEs) [28]. WSINDy is a Galerkin-based data-driven model selection algorithm that utilizes the weak form of the dynamics in a sparse regression framework. By integrating in time against compactly-supported test functions, WSINDy avoids approximation of pointwise derivatives which are known to result in low robustness to noise [39]. In [28] we showed that by integrating against a suitable choice of test functions, correct ODE model terms can be identified together with machine-precision recovery of coefficients (i.e. below the tolerance of the data simulation scheme) from noise-free synthetic data, and for datasets with large noise, WSINDy successfully recovers the correct model terms without explicit data denoising. The use of integral equations for system identification was proposed as early as the 1980's [9] and was carried out in a sparse regression framework in [42] in the context of ODEs, however neither works utilized the full generality of the weak form.

Sparse regression approaches for learning PDEs from data have seen a tremendous spike in activity in the years since 2016. Pioneering works include [41], [39], and [43], where the Douglas-Rachford algorithm, sequential-thresholding least squares (STLS), and basis pursuit with denoising, respectively, are used to regularize the NP-hard problem of finding an optimal sparse solution. Many other predominant approaches for learning dynamical systems (Gaussian processes, deep learning, Bayesian inference, etc.) have since been extended to the discovery of PDEs [7,30,21,22,52,53,59,58,45,55]. A significant disadvantage for the vast majority of PDE discovery methods is the requirement of pointwise derivative approximations. Steps to alleviate this are taken by the authors of [35] and [58], where neural network-based recovery schemes are combined with integral and abstract evolution equations to recover PDEs, and in [53], 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 PDEs 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 PDEs from datasets with much higher noise levels, and from truly weak solutions (see Fig. 3 in Section 5). This works surprisingly well even as the signal-to-noise ratio approaches one. Furthermore, as in the ODE setting, WSINDy achieves high-accuracy recovery in the low-noise regime. These overwhelming improvements resulting from a fully weak² identification method have also been discovered independently by other groups [37,13]. 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 further 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.

The outline of the article is as follows. In Section 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. Key ingredients of the WSINDy algorithm for PDEs (Algorithm 4.2) are covered in Section 4, including a discussion of spectral properties of test functions and robustness to noise (4.1), our modified sequential thresholding scheme (4.2), and regularization using scale invariance of the underlying PDE (4.3). Section 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 6 which summarizes the exposition and includes natural next directions for this line of research. Lastly, additional numerical details are included in the Appendix.

¹ There have been efforts to address the interpretability of neural networks, see e.g. [29,47,38].

² The underlying true solution need only have bounded variation and the only derivatives approximated are weak derivatives.

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 $h : \mathbb{R}^D \times \mathbb{R} \rightarrow \mathbb{C}$ 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 d th 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 (3.6) below) where $(h_j)_{j \in [J]}$ is a collection of J functions $h_j : \mathbb{R}^D \times \mathbb{R} \rightarrow \mathbb{C}$ 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. noise³ ϵ and weak solution u of the PDE

$$D^{\alpha^0} u(x, t) = D^{\alpha^1} g_1(u(x, t)) + D^{\alpha^2} g_2(u(x, t)) + \cdots + D^{\alpha^S} g_S(u(x, t)), \quad x \in \Omega, t \in (0, T). \quad (2.1)$$

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 evolution⁴ 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 $\alpha^s = (\alpha_1^s, \dots, \alpha_D^s, \alpha_{D+1}^s) \in \mathbb{N}^{D+1}$ to denote partial differentiation⁵ with respect to $x = (x_1, \dots, x_D)$ and t , so that

$$D^{\alpha^s} u(x, t) = \frac{\partial^{\alpha_1^s + \cdots + \alpha_D^s + \alpha_{D+1}^s}}{\partial x_1^{\alpha_1^s} \cdots \partial x_D^{\alpha_D^s} \partial t^{\alpha_{D+1}^s}} u(x, t).$$

We emphasize that a wide variety of PDEs can be written in the form (2.1). In particular, in this paper 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 admissible 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. Weak formulation

To arrive at a computationally 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 $(g_s)_{s \in [S]} \subset \text{span}(f_j)_{j \in [J]}$ where the family of functions $(f_j)_{j \in [J]}$ (referred to as the *trial functions*) is known beforehand. This enables us to rewrite (2.1) as

$$D^{\alpha^0} u = \sum_{s=1}^S \sum_{j=1}^J \mathbf{w}_{(s-1)J+j}^* D^{\alpha^s} f_j(u), \quad (3.1)$$

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

To convert the PDE into its weak form, we multiply equation (3.1) by a smooth *test function* $\psi(x, t)$, compactly-supported in $\Omega \times (0, T)$, and integrate over the spacetime domain,

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

³ Here ϵ is used to denote a multi-dimensional array of i.i.d. random variables and has the same dimensions as \mathbf{U} .

⁴ Commonly D^{α^0} is a time derivative ∂_t or ∂_{tt} , although this is not required.

⁵ 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 n th derivative of f .

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

$$\left\langle (-1)^{|\alpha^0|} D^{\alpha^0} \psi, u \right\rangle = \sum_{s=1}^S \sum_{j=1}^J \mathbf{w}_{(s-1)J+j}^* \left\langle (-1)^{|\alpha^s|} D^{\alpha^s} \psi, f_j(u) \right\rangle, \quad (3.2)$$

where $|\alpha^s| := \sum_{d=1}^{D+1} \alpha_d^s$ is the order of the multi-index.⁶ Using an ensemble of test functions $(\psi_k)_{k \in [K]}$, we then discretize the integrals in (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

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

defined by

$$\begin{cases} \mathbf{b}_k = \left\langle (-1)^{|\alpha^0|} D^{\alpha^0} \psi_k, \mathbf{U} \right\rangle, \\ \mathbf{G}_{k, (s-1)J+j} = \left\langle (-1)^{|\alpha^s|} D^{\alpha^s} \psi_k, f_j(\mathbf{U}) \right\rangle, \end{cases} \quad (3.3)$$

where $\mathbf{b} \in \mathbb{R}^K$, $\mathbf{G} \in \mathbb{R}^{K \times SJ}$ and $\mathbf{w}^* \in \mathbb{R}^{SJ}$ 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 (3.2) and a numerical approximation in (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 4.1 and [28]). In this way, solving $\mathbf{b} = \mathbf{G} \mathbf{w}^*$ for the model coefficients \mathbf{w}^* allows for recovery of the PDE (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) 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 $\alpha := (\alpha^s)_{1 \leq s \leq S}$.

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

$$\left(D^{\alpha^0} \psi \right) * u(\mathbf{x}_k, t_k) = \sum_{s=1}^S \sum_{j=1}^J \mathbf{w}_{(s-1)J+j}^* \left(D^{\alpha^s} \psi \right) * f_j(u)(\mathbf{x}_k, t_k). \quad (3.4)$$

The sign factor $(-1)^{|\alpha^s|}$ appearing in (3.2) after integrating by parts is eliminated in (3.4) due to the sign convention in the integrand of the space-time convolution, which is defined by

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

Construction of the linear system $\mathbf{b} = \mathbf{G} \mathbf{w}^*$ as a discretization of the *convolutional* weak form (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 $(\mathbf{Y}, \mathbf{t}) \subset \mathbb{R}^D \times \mathbb{R}$ 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 $\mathbf{Y}_d = (n \Delta x)_{-m_d \leq n \leq m_d}$ and $\mathbf{t} = (n \Delta t)_{-m_{D+1} \leq n \leq m_{D+1}}$. In this way \mathbf{Y} contains $2m_d + 1$ points along each dimension $d \in [D]$, with equal spacing Δx , and \mathbf{t} contains $2m_{D+1} + 1$ points with equal spacing Δt . As with (\mathbf{X}, \mathbf{t}) , points in $(\mathbf{y}_k, t_k) \in (\mathbf{Y}, \mathbf{t})$ take the form

⁶ 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 $(-1)^{|\alpha^s|} = -1$.

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

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

$$(\mathbf{x}_k - \mathbf{x}_j, t_k - t_j) = (\mathbf{y}_{k-j}, \mathbf{t}_{k-j}). \quad (3.5)$$

We stress that (\mathbf{Y}, \mathbf{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 A).

The linear system (3.3) can now be rewritten

$$\begin{cases} \mathbf{b}_k = \Psi^0 * \mathbf{U}(\mathbf{x}_k, t_k), \\ \mathbf{G}_{k, (s-1)J+j} = \Psi^s * f_j(\mathbf{U})(\mathbf{x}_k, t_k), \end{cases} \quad (3.6)$$

where $\Psi^s := D^{\alpha^s} \psi(\mathbf{Y}, \mathbf{t}) \Delta x^D \Delta t$ 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 $(\mathbf{x}_k, t_k) = (\mathbf{X}_{k_1, \dots, k_D}, t_{k_{D+1}}) \in (\mathbf{X}, \mathbf{t})$ by

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

which, substituting the definition of Ψ^s ,

$$:= \sum_{\ell_1=1}^{N_1} \dots \sum_{\ell_{D+1}=1}^{N_{D+1}} D^{\alpha^s} \psi(\mathbf{Y}_{k_1-\ell_1, \dots, k_{D+1}-\ell_{D+1}}, \mathbf{t}_{k_{D+1}-\ell_{D+1}}) f_j(\mathbf{U}_{\ell_1, \dots, \ell_{D+1}}) \Delta x^D \Delta t \quad (3.7)$$

truncating indices appropriately and using (3.5),

$$= \sum_{\ell_1=k_1-m_1}^{k_1+m_1} \dots \sum_{\ell_{D+1}=k_{D+1}-m_{D+1}}^{k_{D+1}+m_{D+1}} D^{\alpha^s} \psi(\mathbf{Y}_{k_1-\ell_1, \dots, k_{D+1}-\ell_{D+1}}, \mathbf{t}_{k_{D+1}-\ell_{D+1}}) f_j(\mathbf{U}_{\ell_1, \dots, \ell_{D+1}}) \Delta x^D \Delta t \quad (3.8)$$

$$= \sum_{\ell_1=k_1-m_1}^{k_1+m_1} \dots \sum_{\ell_{D+1}=k_{D+1}-m_{D+1}}^{k_{D+1}+m_{D+1}} D^{\alpha^s} \psi(\mathbf{x}_k - \mathbf{x}_\ell, t_k - t_\ell) f_j(\mathbf{U}_{\ell_1, \dots, \ell_{D+1}}) \Delta x^D \Delta t \quad (3.9)$$

$$\approx \int_0^T \int_{\Omega} D^{\alpha^s} \psi(\mathbf{x}_k - \mathbf{x}, t_k - t) f_j(u(\mathbf{x}, t)) dx dt. \quad (3.10)$$

3.2. FFT-based implementation and computational complexity for separable ψ

Convolutions in the linear system (3.6) may be computed rapidly if the reference test function ψ is separable over the given coordinates, i.e.

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

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

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

so that only the vectors

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

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,⁷ so that the overall cost of computing each column $\Psi^s * f_j(\mathbf{U})$ of \mathbf{G} is

⁷ The technique of exploiting separability in high-dimensional integration is not new (see [33] for an early introduction) and is frequently utilized in scientific computing (see [4,14] for examples in computational chemistry).

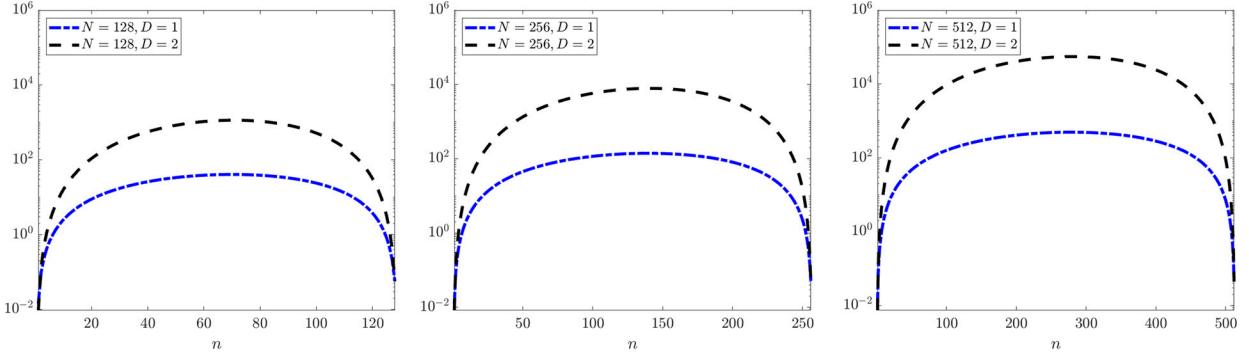


Fig. 1. Reduction in computational cost of multi-dimensional convolution $\Psi^s * f_j(\mathbf{U})$ when Ψ^s and $f_j(\mathbf{U})$ have n and N points in each of $D+1$ dimensions, respectively. Each plot shows the ratio T_{II}/T_I (equations (3.13) and (3.11)), i.e. the factor by which the separable 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 $100 \leq n \leq 450$.

$$T_I(N, n, D) := CN \log(N) \sum_{d=1}^{D+1} N^{D+1-d} (N-n+1)^{d-1}, \quad (3.11)$$

if the computational grid (\mathbf{X}, \mathbf{t}) and reference grid (\mathbf{Y}, \mathbf{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 $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ and $\mathbf{y} = (y_1, \dots, y_N) \in \mathbb{R}^N$ using the FFT,

$$\mathbf{x} * \mathbf{y} = \mathcal{P} \mathcal{F}^{-1} \left(\mathcal{F}(\mathbf{x}^0) \odot \mathcal{F}(\mathbf{y}) \right), \quad (3.12)$$

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,⁸ making the total cost of the WSINDy Algorithm (4.2) in the PDE setting equal to (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 (3.11) 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 Fig. 1 for a comparison between T_I and the naive cost T_{II} of an $(N+1)$ -dimensional convolution:

$$T_{II}(N, n, D) := (2n^{D+1} - 1)(N-n+1)^{D+1}. \quad (3.13)$$

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 $N^{D+1}/\log(N)$.

4. WSINDy algorithm for PDEs and hyperparameter selection

WSINDy for PDE discovery is given in Algorithm 4.2, where the user must specify each of the hyperparameters in Table 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,

⁸ For the examples in Section 5 the walltimes are reported for serial computation of (\mathbf{G}, \mathbf{b}) .

Table 1

Hyperparameters for the WSINDy Algorithm 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 $\hat{\tau} > 0$ if test functions are automatically selected from the data using the method in Appendix A, and K is determined from \mathbf{m} and \mathbf{s} using (4.10).

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 $\{(\mathbf{x}_k, t_k)\}_{k \in [K]}$
λ	$[0, \infty)$	search space for sparsity threshold $\hat{\lambda}$
τ	$(0, 1]$	ψ real-space decay tolerance

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 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 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 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 4.4 and 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.

4.1. Selecting a reference test function ψ

4.1.1. Convolutional weak form and Fourier analysis

Computation of \mathbf{G} and \mathbf{b} in (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 (3.12) suggests a mechanism for choosing appropriate test functions. Define the Fourier coefficients of a function $u \in L^2([0, T])$ by

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

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 k th Fourier mode is by definition noise-dominated, which corresponds to wavenumbers

$$|k| \geq k^* \approx \left(\frac{C}{\sigma \sqrt{\Delta t}} \right)^{1/\ell}. \quad (4.1)$$

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 (3.6) 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.⁹

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 [13], 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 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.

⁹ 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 (3.12).

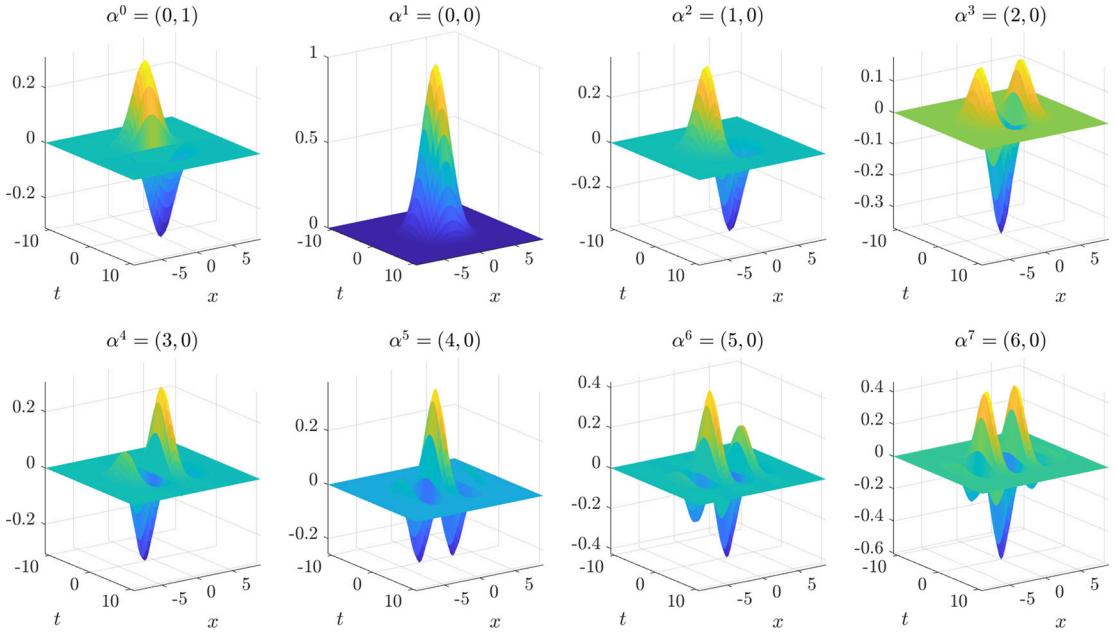


Fig. 2. Plots of reference test function ψ and partial derivatives $D^{\alpha^s}\psi$ used for identification of the Kuramoto-Sivashinsky equation. The upper left plot shows $\partial_t\psi$, the bottom right shows $\partial_x^6\psi$. See Tables 2-4 for more details.

4.1.2. 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 (4.1)). For simplicity, in this article we use the same test function space used in the ODE setting [28] and leave an investigation of the performance of different test functions to future work. Define \mathcal{S} to be the space of functions

$$\phi(v) = \begin{cases} C(v-a)^p(b-v)^q & a < v < b, \\ 0 & \text{otherwise,} \end{cases} \quad (4.2)$$

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.¹⁰ 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 Fig. 2 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 [28]), (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 (4.2) for each ϕ_d . For convenience we center (\mathbf{Y}, 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.¹¹ 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 $(\phi_d^{(\alpha^s_d)}(\mathbf{Y}_d))_{0 \leq s \leq S}$ can be computed from an analogous function $\overline{\phi}_{p_d}$ with support $[-1, 1]$,

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

¹⁰ \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 [12].

¹¹ Test function asymmetry may provide an advantage in some cases, for instance along the time axis, however we do not investigate this here.

using

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

where $\mathbf{n}_d := \{-1 + \frac{n}{m_d} : n \in \{0, \dots, 2m_d\}\}$ 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

$$p_d = \min \left\{ p \geq \bar{\alpha}_d + 1 : \bar{\phi}_p \left(1 - \frac{1}{m_d} \right) \leq \tau \right\}, \quad (4.3)$$

where $\bar{\alpha}_d := \max_{0 \leq s \leq S} (\alpha_d^s)$ is the maximum derivative along the d th 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, (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 $p \geq \bar{\alpha}_d + 1$ ensures that $\phi_d \in C^{\bar{\alpha}_d}(\mathbb{R})$, which is necessary to integrate by parts as many times as required by the multi-index set α . The steps for arriving at the test function values on the reference grid $(\phi_d^{(\alpha_d^s)}(\mathbf{Y}_d))_{0 \leq s \leq S}$ are contained in Algorithm 4.1.

In the examples below, we set $\tau = 10^{-10}$ throughout¹² and we use the method introduced in Appendix A to choose \mathbf{m} , which involves estimating the critical wavenumber k^* (defined in (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, $\phi_x = \phi_1 = \phi_2 = \dots = \phi_D$ 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 4.1 WSINDy Test Function Creation $(\phi_d^{(\alpha_d^s)}(\mathbf{Y}_d))_{0 \leq s \leq S} = \text{get_test_fcns}(m_d, \tau; \mathbf{X}_d, \alpha)$.

```

1:  $N_d = \text{length}(\mathbf{X}_d)$ 
2:  $\Delta x = \text{gridwidth}(\mathbf{X}_d)$ 
3: if  $m_d > \frac{N_d - 1}{2}$  or  $m_d \leq 1$  then
4:   return ("ERROR: invalid support size  $m_d$ ")
5:   BREAK
6: end if
7: Set  $\bar{\alpha}_d = \max_{0 \leq s \leq S} (\alpha_d^s)$ 
8: Solve  $p_d = \min \left\{ p \geq \bar{\alpha}_d + 1 : \bar{\phi}_p \left( 1 - \frac{1}{m_d} \right) \leq \tau \right\}$ 
9: Initialize  $\mathbf{A} = \mathbf{0} \in \mathbb{R}^{(S+1) \times (2m_d + 1)}$ 
10: Set  $\mathbf{n}_d := \{-1 + \frac{n}{m_d} : n \in \{0, \dots, 2m_d\}\}$ 
11: for  $s = 0 : S$  do
12:   Compute analytical order- $(\alpha_d^s)$  derivatives  $\mathbf{A}_s = \bar{\phi}_{p_d}^{(\alpha_d^s)}(\mathbf{n}_d)$ 
13:   Set  $\phi_d^{(\alpha_d^s)}(\mathbf{Y}_d) = \frac{1}{(m_d \Delta x)^{\alpha_d^s}} \mathbf{A}_s$ 
14: end for

```

4.2. Sparsification

To enforce a sparse solution we present a *modified sequential-thresholding least-squares* algorithm MSTLS($\mathbf{G}, \mathbf{b}; \lambda$), defined in (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

$$\mathcal{L}(\lambda) = \frac{\|\mathbf{G}(\mathbf{w}^\lambda - \mathbf{w}^{LS})\|_2}{\|\mathbf{G}\mathbf{w}^{LS}\|_2} + \frac{\#\{\mathcal{I}^\lambda\}}{SJ} \quad (4.4)$$

to select an optimal threshold $\hat{\lambda}$, where \mathbf{w}^λ is the output of MSTLS($\mathbf{G}, \mathbf{b}; \lambda$) defined in equation (4.6), $\#\{\cdot\}$ denotes cardinality, $\mathcal{I}^\lambda := \{1 \leq i \leq SJ : \mathbf{w}_i^\lambda \neq 0\}$ is the index set of non-zero coefficients of \mathbf{w}^λ , $\mathbf{w}^{LS} := (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{b}$ 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 \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}^λ), while the second term penalizes the number of nonzero terms in the resulting model. The normalization simply enforces $\mathcal{L}(0) = \mathcal{L}(\infty) = 1$.

¹² WSINDy appears not to be particularly sensitive to τ , similar results were obtained for $\tau = 10^{-6}, 10^{-10}, 10^{-16}$.

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

$$\begin{cases} L_i^\lambda = \lambda \max \left\{ 1, \frac{\|\mathbf{b}\|}{\|\mathbf{G}_i\|} \right\} \\ U_i^\lambda = \frac{1}{\lambda} \min \left\{ 1, \frac{\|\mathbf{b}\|}{\|\mathbf{G}_i\|} \right\} \end{cases}, \quad 1 \leq i \leq SJ. \quad (4.5)$$

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

$$\text{MSTLS}(\mathbf{G}, \mathbf{b}; \lambda) \quad \begin{cases} \mathcal{I}^\ell = \{1 \leq i \leq SJ : L_i^\lambda \leq |\mathbf{w}_i^\ell| \leq U_i^\lambda\} \\ \mathbf{w}^{\ell+1} = \underset{\text{supp}(\mathbf{w}) \subset \mathcal{I}^\ell}{\text{argmin}} \|\mathbf{G}\mathbf{w} - \mathbf{b}\|_2^2. \end{cases} \quad (4.6)$$

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}^λ 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 [60], 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($\mathbf{G}, \mathbf{b}; \mathcal{L}, \lambda$) is

$$\text{MSTLS}(\mathbf{G}, \mathbf{b}; \mathcal{L}, \lambda) \quad \begin{cases} \widehat{\lambda} = \min \{\lambda \in \lambda : \mathcal{L}(\lambda) = \min_{\lambda \in \lambda} \mathcal{L}(\lambda)\} \\ \widehat{\mathbf{w}} = \text{MSTLS}(\mathbf{G}, \mathbf{b}; \widehat{\lambda}), \end{cases} \quad (4.7)$$

where λ is a finite set of candidate thresholds.¹³ The learned threshold $\widehat{\lambda}$ is the smallest minimizer of \mathcal{L} over the range λ and hence marks the boundary between identification and misidentification of the minimum-cost model, such that $\{\lambda \in \lambda : \lambda < \widehat{\lambda}\}$ results in overfitting. A similar learning method for $\widehat{\lambda}$ combining STLS and Tikhonov regularization (or *ridge regression*) was developed in [39]. We have found that our approach of combining MSTLS($\mathbf{G}, \mathbf{b}; \mathcal{L}, \lambda$) with rescaling, as introduced in the next section, regularizes the sparse regression problem in the case of large model libraries without adding hyperparameters¹⁴ and definitely deserves further study.

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}^* , especially when the underlying data is poorly scaled.¹⁵ 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 5.4).

First, we note that the true model is scale invariant in the following way. If u solves (3.1), then for any scales $\gamma_x, \gamma_t, \gamma_u > 0$, the rescaled function

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

solves

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

where \tilde{D}^{α^s} denotes differentiation with respect to $(\tilde{x}, \tilde{t}) = (\gamma_x x, \gamma_t t)$. For homogeneous functions f_j with power β_j , we have $\tilde{f}_j(\tilde{u}) = f_j(\tilde{u}) = \gamma_u^{\beta_j} f_j(u)$, otherwise $\tilde{f}_j(\tilde{u}) = f_j\left(\frac{\tilde{u}}{\gamma_u}\right) = f_j(u)$ (in which case we set $\beta_j = 0$). The linear system in the rescaled coordinates $\tilde{\mathbf{b}} = \tilde{\mathbf{G}}\tilde{\mathbf{w}}$ is constructed by discretizing the convolutional weak form as before but with a reference

¹³ Other methods of minimizing \mathcal{L} can be used, however minimizers are not unique (there exists a set of minimizers - see Fig. 5). Our approach is efficient and returns the minimizer $\widehat{\lambda}$ which has the useful characterization of defining the thresholds λ that result in overfitting.

¹⁴ Tikhonov regularization involves solving $\widehat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}} \|\mathbf{G}\mathbf{w} - \mathbf{b}\|_2^2 + \gamma^2 \|\mathbf{w}\|_2^2$.

¹⁵ 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.

test function $\tilde{\psi}$ on the rescaled grid $\tilde{\Omega}_R$. We recover the coefficients¹⁶ $\hat{\mathbf{w}}$ at the original scales by setting $\hat{\mathbf{w}} = \mathbf{M}\tilde{\mathbf{w}}$, where $\mathbf{M} = \text{diag}(\boldsymbol{\mu})$ is the diagonal matrix with entries

$$\mu_{(s-1)J+j} := \gamma_u^{-(\beta_j-1)} \gamma_x^{\sum_{d=1}^D (\alpha_d^s - \alpha_d^0)} \gamma_t^{(\alpha_{D+1}^s - \alpha_{D+1}^0)}. \quad (4.8)$$

There is flexibility in choosing the scales $\gamma_u, \gamma_x, \gamma_t$, and a natural choice is to enforce that the columns of $\tilde{\mathbf{G}}$ are similar in norm. Motivated by this, we find that for polynomial and trigonometric libraries, the scales¹⁷

$$\gamma_u = \left(\frac{\|\mathbf{U}\|_{2'}}{\left\| \mathbf{U}^{\bar{\beta}} \right\|_{2'}} \right)^{1/\bar{\beta}}, \quad \gamma_x = \frac{1}{m_x \Delta x} \left(\left(\frac{p_x}{\bar{\alpha}_x} \right) \bar{\alpha}_x! \right)^{1/\bar{\alpha}_x}, \quad \gamma_t = \frac{1}{m_t \Delta t} \left(\left(\frac{p_t}{\bar{\alpha}_t} \right) \bar{\alpha}_t! \right)^{1/\bar{\alpha}_t} \quad (4.9)$$

are sufficient to regularize ill-conditioning due to poor scaling. Here $\bar{\alpha}_x$ and $\bar{\alpha}_t$ are the maximum spatial and temporal derivatives appearing in the library and $\bar{\beta} = \max_j \beta_j$ is the highest monomial power of the functions $(f_j)_{j \in [J]}$. From (4.9) we get that

$$\left\| \tilde{\mathbf{U}}^{\bar{\beta}} \right\|_{2'} = \|\mathbf{U}\|_{2'}$$

and

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

hence, using Young's inequality for convolutions,

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

This shows that with scales $\gamma_u, \gamma_x, \gamma_t$ set according to (4.9), the columns of $\tilde{\mathbf{G}}$ 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 (4.9), which results in a low condition number $\kappa(\tilde{\mathbf{G}})$ (see Table 4). Throughout what follows, quantities defined over scaled coordinates will be denoted by tildes.

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.¹⁸ In WSINDy for ODEs ([28]), 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 article we uniformly subsample $\{(\mathbf{x}_k, t_k)\}_{k \in [K]}$ from (\mathbf{X}, \mathbf{t}) using subsampling frequencies $\mathbf{s} = (s_1, \dots, s_{D+1})$ 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 $\lfloor \frac{N_1 - 2m_1}{s_1} \rfloor \times \dots \times \lfloor \frac{N_{D+1} - 2m_{D+1}}{s_{D+1}} \rfloor$, which determines the number of query points

$$K = \prod_{d=1}^{D+1} \left\lfloor \frac{N_d - 2m_d}{s_d} \right\rfloor. \quad (4.10)$$

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.

¹⁶ Note that thresholding in equation (4.6) occurs on $\hat{\mathbf{w}}$ and the terms $\frac{\|\mathbf{b}\|}{\|\mathbf{G}_i\|}$ in the bounds (4.5) become $\frac{\|\tilde{\mathbf{b}}\|}{\mu_i \|\tilde{\mathbf{G}}_i\|}$.

¹⁷ Here $\|\mathbf{U}\|_{2'}$ is the 2-norm of \mathbf{U} stretched into a column vector (and similarly for $\|\cdot\|_{1'}$).

¹⁸ Note that the projection operation in (3.12) restricts the admissible set of query points to those for which $\psi(\mathbf{x}_k - \mathbf{x}, t_k - t)$ is compactly supported within $\Omega \times [0, T]$, which is necessary for integration by parts to be valid.

Algorithm 4.2 WSINDy for PDEs $(\hat{\mathbf{w}}, \hat{\lambda}) = \text{WSINDy}((f_j)_{j \in [J]}, \alpha, \mathbf{m}, \mathbf{s}, \lambda, \tau; \mathbf{U}, (\mathbf{X}, \mathbf{t}))$.

```

1: for  $d = 1 : D + 1$  do
2:   Compute  $(\phi_d^{(s)}(\mathbf{Y}_d))_{0 \leq s \leq S} = \text{get\_test\_fcns}(m_d, \tau; \mathbf{X}_d, \alpha)$  using Algorithm 4.1
3: end for
4: Compute scales  $\{\gamma_u, (\gamma_d)_{d=1}^{D+1}\}$  and scale matrix  $\mathbf{M} = \text{diag}(\mu)$  using (4.9)
5: Subsample query points  $\{(\mathbf{x}_k, t_k)\}_{k \in [K]} \subset (\mathbf{X}, \mathbf{t})$  using subsampling frequencies  $\mathbf{s} = (s_1, s_2, \dots, s_{D+1})$ ;
6: Compute left-hand side  $\mathbf{b} = \tilde{\Psi}^0 * \tilde{\mathbf{U}}$  over  $\{(\mathbf{x}_k, t_k)\}_{k \in [K]}$  using FFT and separability of  $\psi$ ;
7: for  $j = 1 : J$  do
8:   Compute  $\tilde{f}_j(\tilde{\mathbf{U}})$ ;
9:   for  $s = 1 : S$  do
10:    Compute column  $(s-1)J + j$  of Gram matrix  $\tilde{\mathbf{G}}_{\cdot, (s-1)J+j} = \tilde{\Psi}^s * \tilde{f}_j(\tilde{\mathbf{U}})$  over  $\{(\mathbf{x}_k, t_k)\}_{k \in [K]}$  using FFT and separability of  $\psi$ 
11:   end for
12: end for
13:  $(\hat{\mathbf{w}}, \hat{\lambda}) = \text{MSTLS}(\tilde{\mathbf{G}}, \tilde{\mathbf{b}}; \mathcal{L}, \lambda)$ 

```

Table 2

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

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)	$\partial_t \omega = -\partial_x(\omega u) - \partial_y(\omega v) + \frac{1}{100} \partial_{xx}\omega + \frac{1}{100} \partial_{yy}\omega$

5. Examples

We now demonstrate the effectiveness of WSINDy by recovering the PDEs listed in Table 2 over a range of noise levels. These examples show that WSINDy provides orders of magnitude improvements over derivative-based methods [39], with reliable and accurate recovery of four out of the eight PDEs under noise levels as high as 100% (defined in (5.1) and (5.2)) and for all examples under 20% noise. In contrast to the weak recovery methods in [37,13], WSINDy uses (i) the convolutional weak form (3.6) and FFT-based implementation (3.12), (ii) improved thresholding and automatic selection of the sparsity threshold $\hat{\lambda}$ via (4.6) and (4.7), and (iii) rescaling using (4.9). The effects of these improvements are discussed in Sections 5.4 and 5.5.

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

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

is obtained from a simulation \mathbf{U}^* of the true PDE¹⁹ by adding i.i.d. Gaussian noise with variance σ^2 to each data point, where

$$\sigma := \sigma_{NR} \|\mathbf{U}^*\|_{RMS} := \sigma_{NR} \left(\frac{1}{(N_1 \cdots N_D N_{D+1})} \sum_{k_1=1}^{N_1} \cdots \sum_{k_{D+1}=1}^{N_{D+1}} \left(\mathbf{U}_{k_1, \dots, k_{D+1}}^* \right)^2 \right)^{1/2}. \quad (5.1)$$

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

$$\sigma_{NR}^* := \frac{\|\epsilon\|_{RMS}}{\|\mathbf{U}^*\|_{RMS}} \quad (5.2)$$

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 2), a separate noise variance σ^2 is computed for each vector component so that the noise ratio σ_{NR} of each component satisfies (5.2).

¹⁹ Details on the numerical methods and boundary conditions used to simulate each PDE can be found in Appendix B.

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 [22]) defined by

$$\text{TPR}(\hat{\mathbf{w}}) = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}} \quad (5.3)$$

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 (4.4)) and the resulting learned sparsity threshold $\hat{\lambda}$ (defined in (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 $\hat{\lambda}$ 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

$$E_\infty(\hat{\mathbf{w}}) := \max_{\{j : \mathbf{w}_j^* \neq 0\}} \frac{|\hat{\mathbf{w}}_j - \mathbf{w}_j^*|}{|\mathbf{w}_j^*|}, \quad (5.4)$$

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

$$E_2(\hat{\mathbf{w}}) := \frac{\|\hat{\mathbf{w}} - \mathbf{w}^*\|_{RMS}}{\|\mathbf{w}^*\|_{RMS}}. \quad (5.5)$$

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 identification of the true model.

Finally, when $\text{TPR}(\hat{\mathbf{w}}) = 1$, we report the prediction accuracy between the true data \mathbf{U}^* and a numerical solution \mathbf{U}^{dd} to the data-driven PDE using the same initial conditions. We compute the relative L_2 error $P_t(\hat{\mathbf{w}})$ at time $t = 0.5T$ (i.e. at the half-way point in time) defined by

$$P_t(\hat{\mathbf{w}}) := \frac{\|\mathbf{U}_t^{dd} - \mathbf{U}_t^*\|_{RMS}}{\|\mathbf{U}_t^*\|_{RMS}} \quad (5.6)$$

where \mathbf{U}_t^{dd} , \mathbf{U}_t^* 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

$$T_{\text{tol}}(\hat{\mathbf{w}}) := \frac{1}{T} \inf\{t \in [0, T] : P_t(\hat{\mathbf{w}}) > \text{tol}\}, \quad (5.7)$$

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 (5.7) is computed over $t \in \mathbf{t}$ and we set $\text{tol} = 0.1$. We provide results for $P_{0.5T}(\hat{\mathbf{w}})$ and $T_{0.1}(\hat{\mathbf{w}})$ averaged over the weights $\hat{\mathbf{w}}$ satisfying $\text{TPR}(\hat{\mathbf{w}}) = 1$.

For each system in Table 2 and each noise level $\sigma_{NR} \in \{0.025q : q \in \{0, \dots, 40\}\}$ we run WSINDy on 200 instantiations of noise²⁰ and average the results of error statistics (5.3)-(5.7). Computations were carried out on a University of Colorado Boulder Blanca Condo cluster.²¹

5.2. Implementation details

The hyperparameters used in WSINDy applied to each of the PDEs in Table 2 are given in Table 3. To select test function discrete support lengths we used a combination of the changepoint method²² described in Appendix 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 $\hat{\mathbf{w}} = \text{MSTLS}(\mathbf{G}, \mathbf{b}; \mathcal{L}, \lambda)$ (see equation (4.7)), the search space λ for the learned threshold $\hat{\lambda}$ is fixed for all examples at:

²⁰ We find that 200 runs sufficiently reduces variance in the results.

²¹ 2X Intel Xeon 5218 at 2.3 GHz with 22 MB cache, 16 cores per cpu, and 384 GB ram.

²² For Burgers, KdV, and KS we set $\hat{\tau} = 3$ (defined in Appendix A.2) while for NLS, PM, SG, RD, and NS we used $\hat{\tau} = 1$. For KS and NLS we chose (m_x, m_t) values nearby that had better performance.

Table 3

WSINDy hyperparameters used to identify each example PDE.

PDE	U	f_j	α	(m_x, m_t)	(s_x, s_t)
IB	256×256	$(u^{j-1})_{j \in [7]}$	$((\ell, 0))_{0 \leq \ell \leq 6}$	(60, 60)	(5, 5)
KdV	400×601	$(u^{j-1})_{j \in [7]}$	$((\ell, 0))_{0 \leq \ell \leq 6}$	(45, 80)	(8, 12)
KS	256×301	$(u^{j-1})_{j \in [7]}$	$((\ell, 0))_{0 \leq \ell \leq 6}$	(23, 22)	(5, 6)
NLS	$2 \times 256 \times 251$	$(u^i v^j)_{0 \leq i+j \leq 6}$	$((\ell, 0))_{0 \leq \ell \leq 6}$	(19, 25)	(5, 5)
PM	$200 \times 200 \times 128$	$(u^{i-1})_{i \in [5]}$	$((\ell_1, \ell_2, 0))_{0 \leq \ell_1, \ell_2 \leq 4}$	(37, 20)	(8, 5)
SG	$129 \times 403 \times 205$	$(u^{i-1})_{i \in [5]}, (\sin(ju), \cos(ju))_{j=1,2}$	$((\ell, 0, 0), (0, \ell, 0))_{0 \leq \ell \leq 4}$	(40, 25)	(5, 8)
RD	$2 \times 256 \times 256 \times 201$	$(u^i v^j)_{0 \leq i+j \leq 4}$	$((\ell, 0, 0), (0, \ell, 0))_{0 \leq \ell \leq 5}$	(13, 14)	(13, 12)
NS	$3 \times 324 \times 149 \times 201$	$\begin{cases} (\omega^i u^j v^k)_{0 \leq i+j+k \leq 2}, & \alpha^s = 0 \\ (\omega^i u^j v^k)_{0 \leq i+j+k \leq 3, i>0}, & \alpha^s > 0 \end{cases}$	$((\ell, 0, 0), (0, \ell, 0))_{0 \leq \ell \leq 2}$	(31, 14)	(12, 8)

Table 4

Additional specifications resulting from the choices in Table 3. The last column shows the start-to-finish walltime of Algorithm 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.

PDE	$\tilde{\mathbf{G}}$	$\kappa(\tilde{\mathbf{G}})$	(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

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

in other words λ contains 50 points with $\log_{10}(\lambda)$ equally spaced from -4 to 0 . This implies a stopping criteria of 50SJ thresholding iterations.²³We fix the subsampling frequencies (s_x, s_t) to $(\frac{N_1}{50}, \frac{N_3}{50})$ 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 4, such as the dimensions and condition number of the rescaled Gram matrix $\tilde{\mathbf{G}}$ (computed from a typical 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 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.

5.3. Comments on chosen examples

The primary reason for choosing the examples in Table 2 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 [13]). We use analytical weak solutions, with inviscid Burgers data forming a shock which propagates at constant speed (see Fig. 3 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 demonstrate that WSINDy can identify the cross-diffusion term $\partial_{xy}(u^2)$ to high accuracy from a large candidate model library.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 $t \in [0, 10^{-3}]$), and hence are not identifiable from large polynomial libraries using naive approaches. The sparsification and rescaling measures in Sections 4.2 and 4.3 are essential to removing this barrier.The Sine-Gordon equation²⁴ 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 equa-

²³ 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 $\lambda \in \lambda$ as in equation (4.6)), hence in practice the full MSTLS($\mathbf{G}, \mathbf{b}; \mathcal{L}, \lambda$) algorithm requires far fewer iterations than the theoretical maximum of $\#\lambda SJ$.

²⁴ 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.

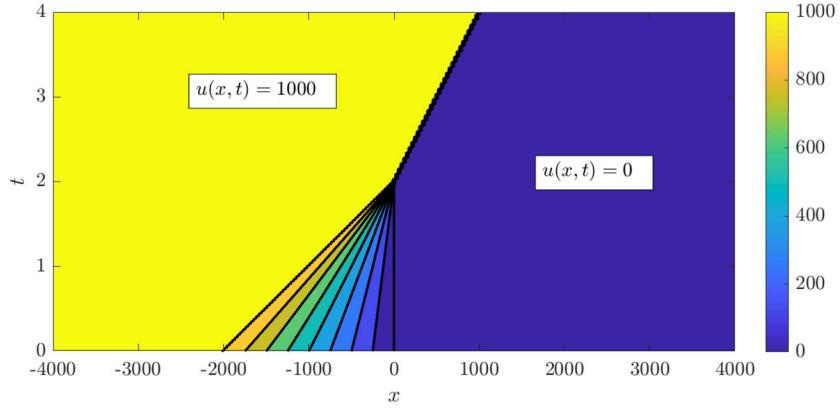


Fig. 3. Characteristics of the shock-forming solution (B.2) used to identify the inviscid Burgers equation. A shock forms at time $t = 2$ and travels along the line $x = 500(t - 2)$.

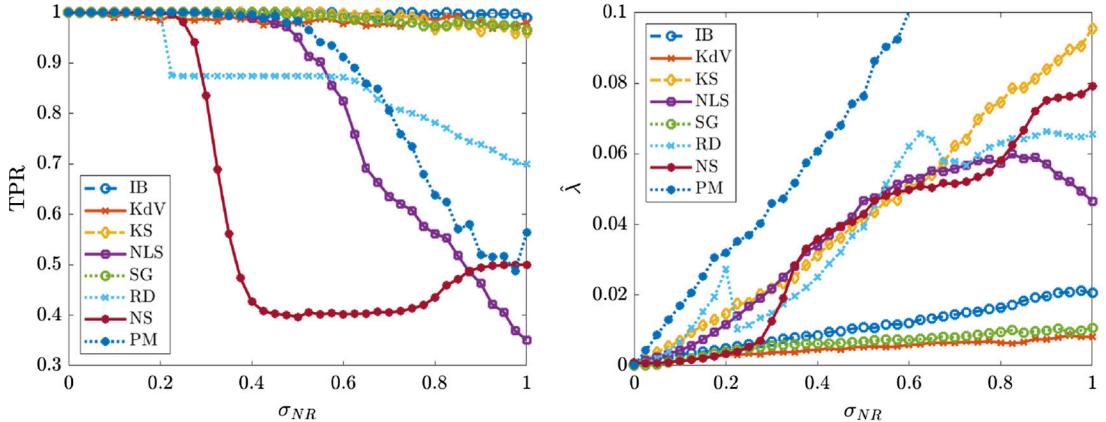


Fig. 4. Left: average TPR (true positivity ratio, defined in (5.3)) for each of the PDEs in Table 2 computed from 200 instantiations of noise for each noise level σ_{NR} . Right: average learned threshold $\hat{\lambda}$ (defined in (4.7)). For the porous medium equation (PM), $\hat{\lambda}$ increases to 0.2 as σ_{NR} approaches 1 (we omit this from the plot in order to make visible the $\hat{\lambda}$ trends for the other systems).

tion 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 \tilde{G} in Table 4), 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.²⁵ Finally, the vortex-shedding limit cycle for the 2D Navier-Stokes equations is used primarily to compare to previous results in [39], and we find that at large-noise WSINDy conveniently selects the Euler equations.

5.4. Results: model identification

Performance regarding the identification of correct nonzero terms in each model is reported in Figs. 4 and 5, which include plots of the average TPR, the learned threshold $\hat{\lambda}$, and the loss function $\mathcal{L}(\lambda)$ (defined in (5.3), (4.7), and (4.4), respectively). As we will discuss, significant decreases in average TPR are often accompanied by transitions in the identified $\hat{\lambda}$.

Fig. 4 (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 [39], especially considering the large library of 190 terms used.

²⁵ We note that discovery of the same reaction-diffusion system from a much smaller library of terms is shown in [39,37], but with different initial conditions that result in a spiral wave limit cycle. Our choice of initial conditions is motivated below in Appendix B.

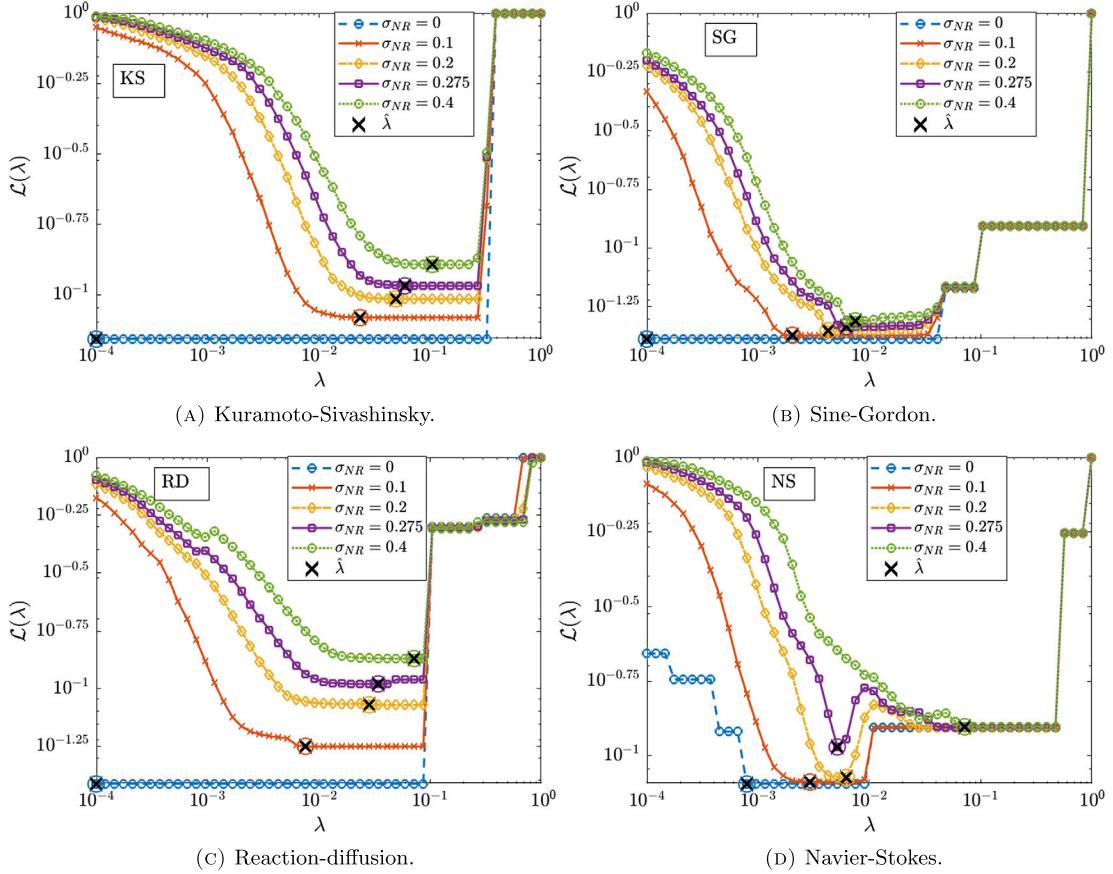


Fig. 5. Plots of the average loss function $\mathcal{L}(\lambda)$ and resulting optimal threshold $\hat{\lambda}$ for the Kuramoto Sivashinsky, Sine-Gordon, Reaction diffusion and Navier-Stokes equations.

We observe in Fig. 4 (right) that the learned threshold $\hat{\lambda}$ increases with σ_{NR} , suggesting that automatic selection of $\hat{\lambda}$ in the learning algorithm (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 $\hat{\lambda}$ approaches 0.1 as σ_{NR} approaches 1, which implies that at higher noise levels the range of $\hat{\lambda}$ values that is necessary²⁶ for correct model identification is approximately ($\sim 0.1, \sim 0.5$). Since it is highly unlikely that this range of admissible values would be known *a priori*, the chances of manually selecting a feasible $\hat{\lambda}$ for Kuramoto-Sivashinsky are prohibitively low in the large noise regime (see Fig. 5a for visualizations of the loss \mathcal{L} applied to KS data). This effect is even greater for the porous medium equation. Automatic selection of $\hat{\lambda}$ thus removes this sensitivity. In contrast, $\hat{\lambda}$ is largely unaffected by increases in σ_{NR} for Burgers, Korteweg-de Vries and Sine-Gordon. In particular, Fig. 5b 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

$$\partial_t \mathbf{u} = \beta \mathcal{A}(\mathbf{u}) + \alpha \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{u} \quad (5.8)$$

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 (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 conditions²⁷). 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

²⁶ By definition (4.7), $\hat{\lambda}$ is the minimum value in λ that minimizes the loss \mathcal{L} (4.7), hence values in λ below $\hat{\lambda}$ are precisely the thresholds that result in misidentification of the correct model by overfitting, while thresholds above $\min_{\{j: \mathbf{w}_j^* \neq 0\}} |\mathbf{w}_j^*|$ necessarily underfit the model.

²⁷ This is discussed further in Appendix B.7.

Table 5Accuracy of WSINDy applied to noise-free data ($\sigma_{NR} = 0$).

	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 6Accuracy comparison between WSINDy and PDE-FIND with $\sigma_{NR} = 0.01$ (results for PDE-FIND reproduced from [39]).

	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}

shift between detection of the correct model and the oscillatory version (5.8) is also detectable in the learned threshold $\hat{\lambda}$, which decreases at $\sigma_{NR} = 0.22$ (see RD data in Fig. 4 (right)), and in the loss function \mathcal{L} (Fig. 5c). At $\sigma_{NR} = 0.275$ we see that \mathcal{L} in Fig. 5c is minimized for λ in the approximate range ($\sim 0.02, \sim 0.05$) 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 (5.8) and the true model, respectively, but since the true model has a slightly higher loss at $\sigma_{NR} = 0.275$, model (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 WINSIDy returns (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} , Fig. 5d 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 setting, Table 5 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 Fig. 6). Nevertheless, with reliable recovery up to 27% noise, WSINDy makes notable improvements on previous results ([39]). Moreover, recovery of the Euler equations at high noise is desirable as this can be seen as the correct leader-order model.

5.5. Results: coefficient accuracy

Accuracy in the recovered coefficients is measured by E_∞ and E_2 (defined in (5.4) and (5.5), respectively) and shown in Table 5 for $\sigma_{NR} = 0$ and in Fig. 6 for $\sigma_{NR} > 0$. As in the ODE case, the coefficient error E_∞ for smooth, noise-free data is determined by the order of accuracy of the numerical simulation method,²⁸ since the error resulting from the trapezoidal rule is of lower order for the values (p_x, p_t) used in Table 4 (see [28], Lemma 1). Table 5 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 6 shows that WSINDy improves over PDE-FIND by about two digits.²⁹

For $\sigma_{NR} > 0$, in Fig. 6 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 [13] 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 $\Psi^s * \mathbf{U}^*$ and approximate variance $\Delta x^D \Delta t \|D^{\alpha^s} \psi\|_2^2 \sigma^2$, hence are *unbiased*³⁰ and lead to perturbations that scale linearly with σ_{NR} . On the other hand, general monomial nonlinearities³¹ $\Psi^s * \mathbf{U}^j$ with $j > 1$ are *biased* and have approximate variance $\Delta x^D \Delta t \|D^{\alpha^s} \psi\|_2^2 p_{2j}(\sigma)$ 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 $\|D^{\alpha^s} \psi\|_2$. Thus, for larger noise and higher-degree monomial nonlinearities, we expect superlinear growth of the error, as observed in particular with

²⁸ 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 B for more details).

²⁹ Results shown for $\sigma_{NR} = 0.01$ reproduced from [39] (note that PDE-FIND is unreliable at higher noise levels).

³⁰ In other words, equal to the noise-free case in expectation (recall that \mathbf{U}^* is the underlying noise-free data).

³¹ With the exception of $j = 2$ and odd $|\alpha^s|$, due to the fact that $\mathbb{E}[\Psi^s * \epsilon^2] \approx \mathbb{E}[\epsilon^2] \int_{\Omega_R} D^{\alpha^s} \psi dx dt = 0$.

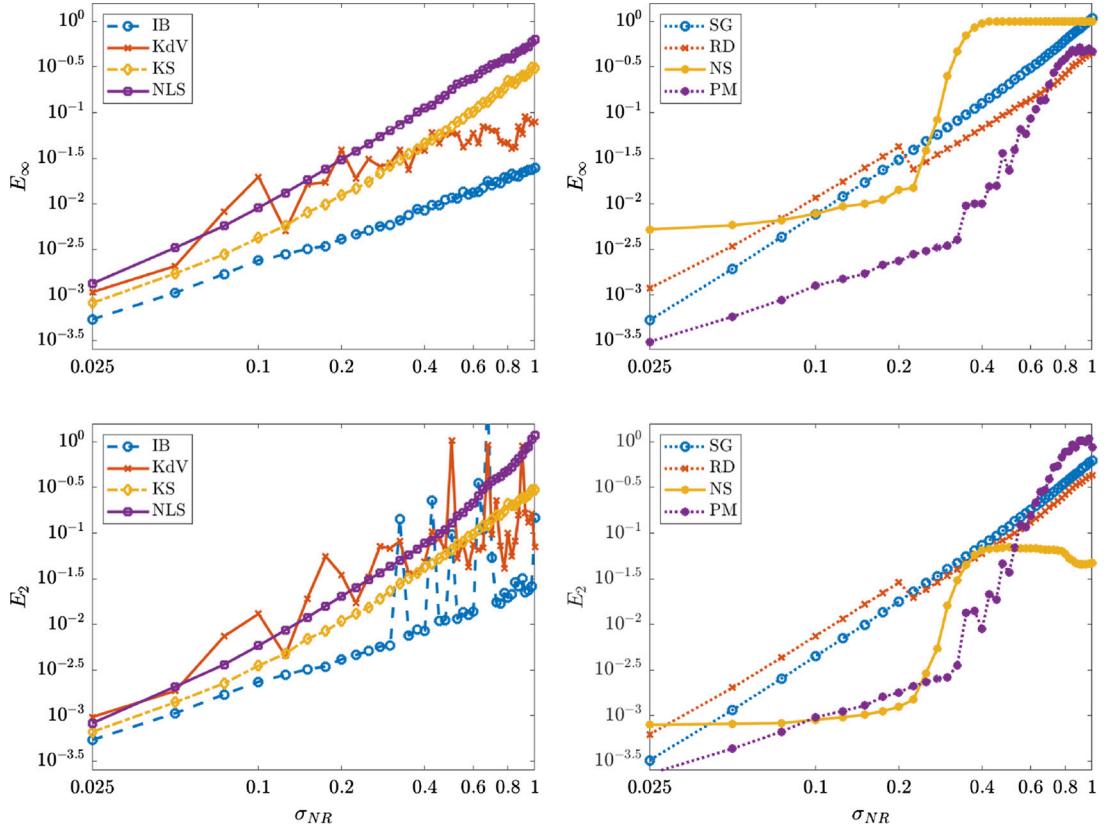


Fig. 6. Coefficient errors E_∞ and E_2 (equations (5.4) and (5.5)) for each of the seven models Table 2. Models in one and two spatial dimensions are shown on the left and right, respectively.

nonlinear Schrödinger's, Sine-Gordon, and reaction-diffusion. Nevertheless, Fig. 6 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 [28].

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 datasets used, which lead to the closest pure-advection model for each system being given by³²

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

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.

5.6. Results: prediction accuracy

Lastly, Fig. 7 shows the prediction accuracy on a subset of the systems in Table 2 as measured by $P_{0.5T}(\hat{\mathbf{w}})$ and $T_{0.1}(\hat{\mathbf{w}})$ (defined in (5.6) and (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).

³² This is found by projecting the left-hand side \mathbf{b} onto the column $\partial_x \psi * \mathbf{U}^*$ (i.e. in the noise-free case).

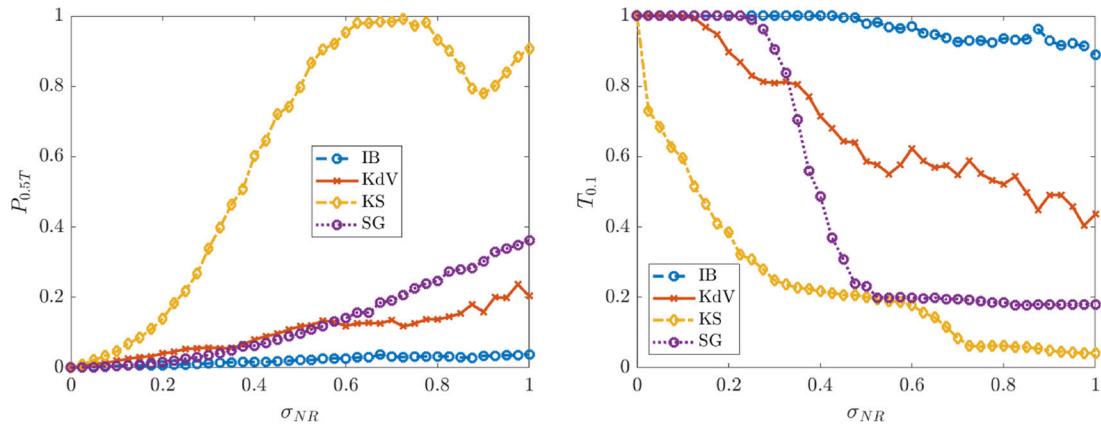


Fig. 7. Prediction accuracy measured by $P_{0.5T}(\hat{w})$ and $T_{0.1}(\hat{w})$ (defined in (5.6) and (5.7), respectively).

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 $\hat{\lambda}$, demonstrated the necessity of avoiding manual hyperparameter tuning. The underlying convolutional weak form (3.4) allows for efficient implementation using the FFT. This naturally leads to a selection criterion for admissible 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 5.4).

We close with a summary of possible future directions. In Section 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 (4.2) is optimal, but it does appear to work very well, as demonstrated above (as well as in the ODE setting [28]) and also observed in [37,13]. 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 A.

In the ODE setting, adaptive placement of test functions provided increased robustness to noise. Convolution query points can similarly 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 [28]. Previous results on generalized sensitivity analysis for PDEs may provide improvements in this direction [18,46].

Accuracy in the recovered coefficients is still not entirely understood and is needed to derive recovery guarantees. It is claimed in [13] 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 $\hat{\lambda}$ (see Fig. 4 (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.

CRediT authorship contribution statement

The authors contributed equally.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

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 manuscript 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 (MIT, Department of Mechanical Engineering), and Zofia Stanley (University of Colorado at Boulder, Department of Applied Mathematics), for helpful discussions.

Appendix 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 $(k_1^*, \dots, k_{D+1}^*)$ 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 $\mathbf{t} \in \mathbb{R}^{N_2}$. Figs. 8–9 illustrate this approach using Kuramoto-Sivashinsky data with 50% noise. Below, \mathcal{F}^x and \mathcal{F}^t denote the discrete Fourier transform (DFT) along the x and t coordinates, respectively, while \mathcal{F} denotes the full two-dimensional DFT.

A.1. Detection of critical wavenumbers

Assume the data has additive white noise $\mathbf{U} = \mathbf{U}^* + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and that $\mathcal{F}(\mathbf{U}^*)$ decays. The power spectrum of the noise $|\mathcal{F}^x(\epsilon)|$ is then i.i.d, hence as discussed in Section 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 :

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

where $\overline{|\mathcal{F}_j^x(\mathbf{U})|}$ is the time-average of the j th 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 approximation³³ to \mathbf{H}^x using two pieces (see Fig. 8). An algorithm for this is given in [20] and implemented in MATLAB using the function `findchangepts`.

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 $\hat{\tau}$. As in Section 4.1, τ specifies the rate of decay of ϕ_x and ϕ_t in real space through equation (4.3). The hyperparameter $\hat{\tau}$ is introduced to specify the rate of decay of ϕ_x and ϕ_t in Fourier space. Specifically, for a chosen $\hat{\tau}$ we enforce that the changepoints k_x^* and k_t^* fall approximately $\hat{\tau}$ standard deviations into the tail of the spectra $\hat{\phi}_x$ and $\hat{\phi}_t$. 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 (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

³³ In the weighted least-squares sense with weights $\omega_k = |\mathbf{H}_k^x|^{-1}$.

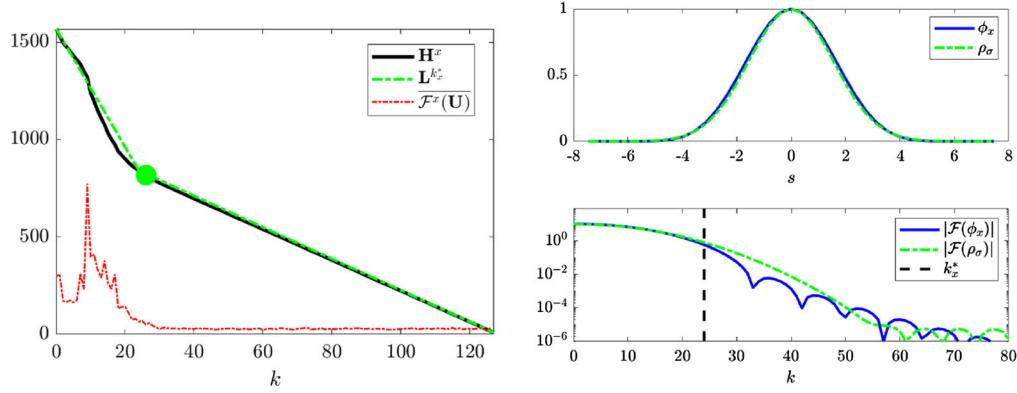


Fig. 8. Visualization of the changepoint algorithm for KS data with 50% noise. Left: \mathbf{H}^x (defined in (A.1)) and best two-piece approximation $\mathbf{L}^{k_x^*}$ along with resulting changepoint $k_x^* = 24$. The noise-dominated region of \mathbf{H}^x ($k > 24$) is approximately linear as expected from the i.i.d. noise. (The time-averaged power spectrum $\overline{F^x(\mathbf{U})}$ 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.)

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

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 moments³⁴:

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

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

$$\begin{aligned} \frac{2\pi}{N_1 \Delta x} k_x^* &= \frac{\widehat{\tau}}{\sigma} = \widehat{\tau} \frac{\sqrt{2p_x + 3}}{m_x \Delta x} \\ \implies 2\pi k_x^* m_x &= \widehat{\tau} N_1 \sqrt{2p_x + 3}, \end{aligned} \quad (\text{A.2})$$

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 (4.3) and (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 $m_x \geq 2$ in the nonempty interval

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

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 (4.3) or (A.2).

Fig. 9 illustrates the implicit filtering of this process using the Burgers-type nonlinearity $\partial_x(\mathbf{U}^2)$ and the same KS dataset as in Fig. 9 with 50% noise. The top panel compares a one-dimensional slice in x taken at fixed time $t = 99$ of the clean data $(\mathbf{U}^*)^2$ and noisy data $(\mathbf{U})^2$. The middle panel shows the Fourier transforms of $(\mathbf{U}^*)^2$ 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 $(\mathbf{U}^*)^2$, making this agreement in spectrum nontrivial).

³⁴ This also shows that with $\sigma = a / \sqrt{2p + 3}$, if we take $a = \sqrt{2p}$ then we get pointwise convergence $\phi_{a,p} \rightarrow \rho_1$ as $p \rightarrow \infty$.

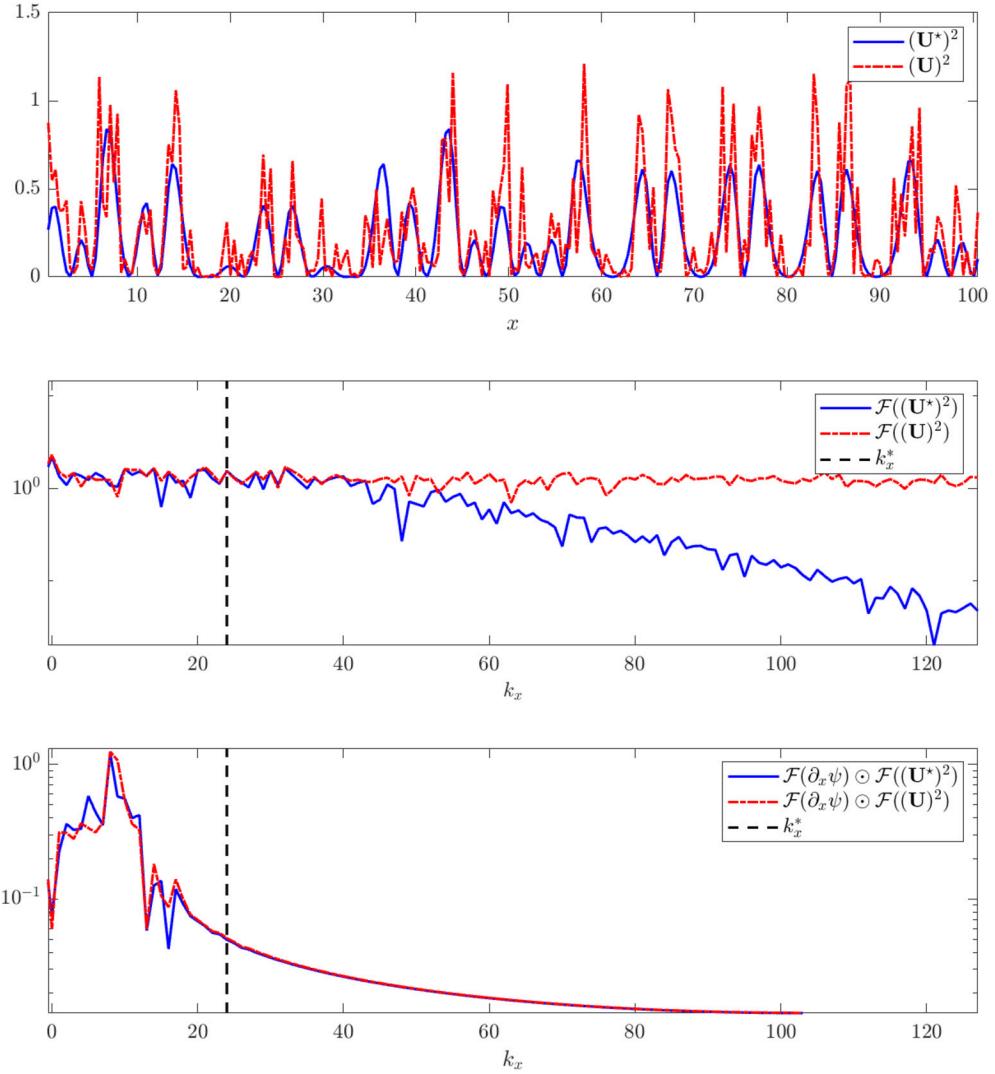


Fig. 9. Illustration of the test function learning algorithm using computation of $\partial_x \psi * (\mathbf{U}^2)$ along a slice in x at fixed time $t = 99$ for the same dataset used in Fig. 8. From top to bottom: (i) clean \mathbf{U}^* and noisy \mathbf{U} variables, (ii) power spectra of the clean vs. noisy data along with the learned corner point k_x^* , (iii) power spectra of the element-wise products $\mathcal{F}(\partial_x \psi) \odot \mathcal{F}((\mathbf{U}^*)^2)$ and $\mathcal{F}(\partial_x \psi) \odot \mathcal{F}((\mathbf{U})^2)$ (recall that these computations are embedded in the FFT-based convolution (3.12)).

Appendix B. Simulation methods

We now review the numerical methods used to simulate noise-free datasets for each of the PDEs in Table 2 (note that dimensions of the datasets are given in Table 3). 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++ [44], all computations were performed 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 [30].

B.1. Inviscid Burgers

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

We take for exact data the shock-forming solution

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

which becomes discontinuous at $t = \alpha^{-1}$ with a shock traveling along $x = \frac{A}{2} \left(t - \frac{1}{\alpha} \right)$ (see Fig. 3). 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 (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 $1 \leq i, j \leq 256$.

B.2. Korteweg-de Vries

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

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 [17] 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

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

B.3. Kuramoto-Sivashinsky

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

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 [17] 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

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

B.4. Nonlinear Schrödinger

$$w_t = -\frac{i}{2} \partial_{xx} w + |w|^2 w \quad (\text{B.5})$$

For the nonlinear Schrödinger equation (NLS) we reuse the same dataset from [39], 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

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

B.5. Anisotropic porous medium

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

The equation can be rewritten

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

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 (Fig. 10)

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

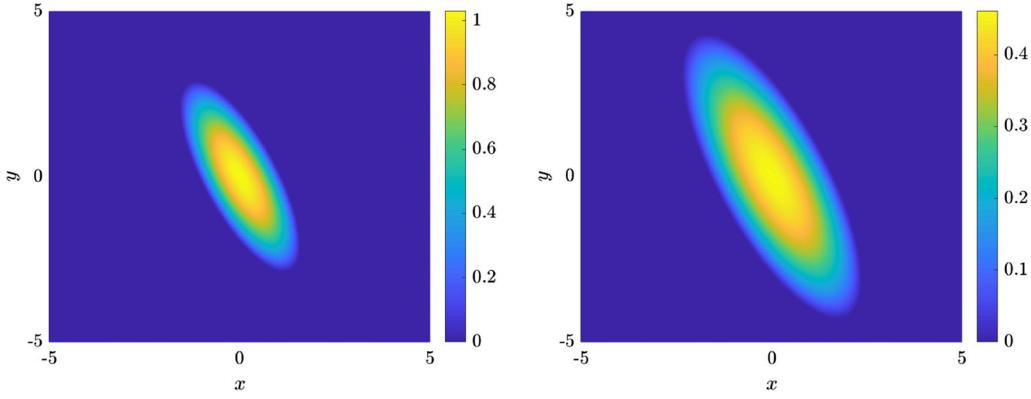


Fig. 10. Noise-free data used for the anisotropic porous medium equation (B.7) at the initial time $t = 0.5$ (left) and final time $t = 2.5$ (right).

where $\mathbf{x} = (x, y)^T$ and $C = (8\pi\sqrt{\det(\mathbf{D})})^{-1/2}$ 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 [3,32]. 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 $\Delta t = 0.0157$.

B.6. Sine-Gordon

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

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 $\Delta t = 6e-5$. 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 MSLTS (see Section 4.2) for sparsity enforcement, WSINDy returns a combination of $\sin(u)$ and terms from Taylor expansion of $\sin(u)$,

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

MSLTS removes this problem. Furthermore, the test function selection method in Appendix A is essential for allowing robust recovery of the Sine-Gordon equation as $\sigma_{NR} \rightarrow 1$ (see Fig. 4).

B.7. Reaction-diffusion

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

The system (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

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

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 [39] (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

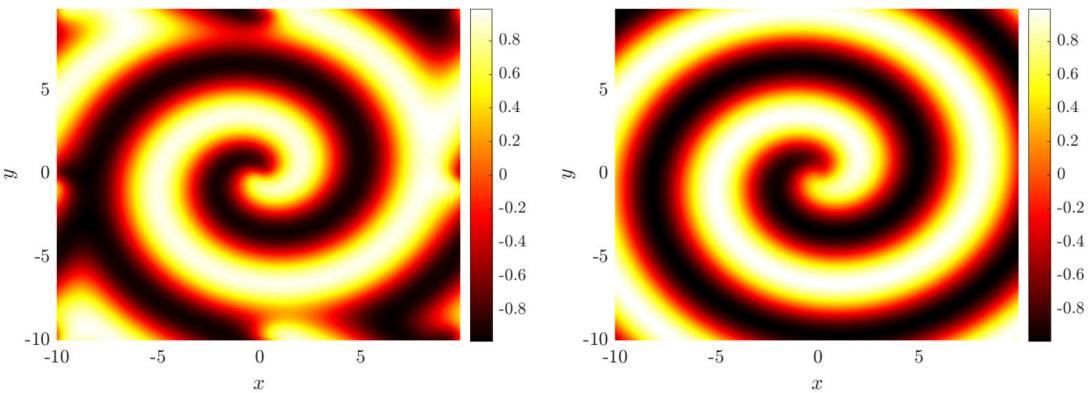


Fig. 11. Comparison between the full reaction-diffusion model (B.10) (left) and the pure-oscillatory reduced model (B.11) (right) at the final time $T = 10$ with both models simulated from the same initial conditions leading to a spiral wave (only the v component is shown, results for u are similar).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

resulting model is purely oscillatory. In other words, the stable spiral limit cycle happens to be well-approximated by the pure-oscillatory model

$$\partial_t \mathbf{u} = \alpha \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{u} \quad (\text{B.11})$$

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 Fig. 11. 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.

B.8. Navier-Stokes

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

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 [44] 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 $(\mathbf{U}, \mathbf{V}, \mathbf{W})$ 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 $\Delta t = 0.2$.

References

- [1] H. Akaike, A new look at the statistical model identification, *IEEE Trans. Autom. Control* 19 (6) (December 1974) 716–723.
- [2] Hirotugu Akaike, On entropy maximization principle, in: P.R. Krishnaiah (Ed.), *Applications of Statistics*, North Holland, Amsterdam, Netherlands, 1977, pp. 27–41.
- [3] G.I. Barenblatt, On some unsteady fluid and gas motions in a porous medium, *Prikl. Mat. Meh.* 16 (1) (1952) 67–78.
- [4] Gregory Beylkin, Martin J. Mohlenkamp, Algorithms for numerical analysis in high dimensions, *SIAM J. Sci. Comput.* 26 (6) (2005) 2133–2159.
- [5] D.M. Bortz, P.W. Nelson, Model selection and mixed-effects modeling of HIV infection dynamics, *Bull. Math. Biol.* 68 (8) (November 2006) 2005–2025.
- [6] Steven L. Brunton, Joshua L. Proctor, J. Nathan Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci.* 113 (15) (2016) 3932–3937.
- [7] Xiaoli Chen, JinQiao Duan, George Karniadakis, Learning and meta-learning of stochastic advection–diffusion–reaction systems from sparse measurements, *Eur. J. Appl. Math.* 32 (3) (2020) 1–24.
- [8] Alexandre Cortella, Kwang-Chun Park, Alireza Doostan, Sparse identification of nonlinear dynamical systems via reweighted ℓ_1 -regularized least squares, *Comput. Methods Appl. Mech. Eng.* 376 (2021) 113620.
- [9] James P. Crutchfield, Bruce S. McNamara, Equations of motion from a data series, *Complex Syst.* 1 (417–452) (1987) 121.
- [10] Min Dai, Ting Gao, Yubin Lu, Yayun Zheng, JinQiao Duan, Detecting the maximum likelihood transition path from data of stochastic dynamical systems, *Chaos, Interdiscip. J. Nonlinear Sci.* 30 (11) (2020) 113124.
- [11] M. Brian de Silva, Kathleen Champion, Markus Quade, Jean-Christophe Loiseau, J. Nathan Kutz, Steven L. Brunton, PySINDy: a python package for the sparse identification of nonlinear dynamics from data, arXiv–2004, 2020.
- [12] S.N. Ershov, B-splines and bernstein basis polynomials, *Phys. Part. Nucl. Lett.* 16 (6) (2019) 593–601.
- [13] R. Daniel Gurevich, Patrick AK Reinbold, Roman O. Grigoriev, Robust and optimal sparse regression for nonlinear pde models, *Chaos, Interdiscip. J. Nonlinear Sci.* 29 (10) (2019) 103113.
- [14] 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 J. Sci. Comput.* 38 (5) (2016) S123–S142.

- [15] Moritz Hoffmann, Christoph Fröhner, Frank Noé, Reactive sindy: discovering governing reactions from concentration data, *J. Chem. Phys.* 150 (2) (2019) 025101.
- [16] Sung Ha Kang, Wenjing Liao, Yingjie Liu Ident, Identifying differential equations with numerical time evolution, preprint, arXiv:1904.03538, 2019.
- [17] Aly-Khan Kassam, Lloyd N. Trefethen, Fourth-order time-stepping for stiff pdes, *SIAM J. Sci. Comput.* 26 (4) (2005) 1214–1233.
- [18] Dustin D. Keck, David M. Bortz, Generalized sensitivity functions for size-structured population models, *J. Inverse Ill-Posed Probl.* 24 (3) (2016) 309–321.
- [19] Rachael T. Keller, Qiang Du, Discovery of dynamics using linear multistep methods, *SIAM J. Numer. Anal.* 59 (1) (2021) 429–455.
- [20] Rebecca Killick, Paul Fearnhead, Idris A. Eckley, Optimal detection of changepoints with a linear computational cost, *J. Am. Stat. Assoc.* 107 (500) (2012) 1590–1598.
- [21] John H. Lagergren, John T. Nardini, Ruth E. Baker, Matthew J. Simpson, Kevin B. Flores, Biologically-informed neural networks guide mechanistic modeling from sparse experimental data, *PLoS Comput. Biol.* 16 (12) (2020) e1008462.
- [22] John H. Lagergren, John T. Nardini, G. Michael Lavigne, Erica M. Rutter, Kevin B. Flores, Learning partial differential equations for biological transport models from noisy spatio-temporal data, *Proc. R. Soc. A* 476 (2234) (February 2020) 20190800.
- [23] Gabriele Lillacci, Mustafa Khammash, Parameter estimation and model selection in computational biology, *PLoS Comput. Biol.* 6 (3) (March 2010) e1000696.
- [24] Zichao Long, Yiping Lu, Bin Dong, Pde-net 2.0: learning pdes from data with a numeric-symbolic hybrid deep network, *J. Comput. Phys.* 399 (2019) 108925.
- [25] Zichao Long, Yiping Lu, Xianzhong Ma, Bin Dong Pde-net, Learning pdes from data, in: International Conference on Machine Learning, PMLR, 2018, pp. 3208–3216.
- [26] Yiping Lu, Aoxiao Zhong, Quanzheng Li, Bin Dong, Beyond finite layer neural networks: bridging deep architectures and numerical differential equations, in: International Conference on Machine Learning, PMLR, 2018, pp. 3276–3285.
- [27] Niall M. Mangan, J. Nathan Kutz, Steven L. Brunton, Joshua L. Proctor, Model selection for dynamical systems via sparse regression and information criteria, *Proc. R. Soc. A, Math. Phys. Eng. Sci.* 473 (2024) (2017) 20170009.
- [28] Daniel A. Messenger, David M. Bortz, Weak SINDY: Galerkin-based data-driven model selection, preprint, arXiv:2005.04339, 2020.
- [29] Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Methods for interpreting and understanding deep neural networks, *Digit. Signal Process.* 73 (2018) 1–15.
- [30] John T. Nardini, John H. Lagergren, Andrea Hawkins-Daarud, Lee Curtin, Bethan Morris, Erica M. Rutter, Kristin R. Swanson, Kevin B. Flores, Learning equations from biological data with limited time samples, *Bull. Math. Biol.* 82 (9) (2020) 1–33.
- [31] Owhadi Houman, Bayesian numerical homogenization, *Multiscale Model. Simul.* 13 (3) (2015) 812–828.
- [32] R.E. Pattle, Diffusion from an instantaneous point source with a concentration-dependent coefficient, *Q. J. Mech. Appl. Math.* 12 (4) (1959) 407–409.
- [33] Victor Pereyra, G. Scherer, Efficient computer manipulation of tensor products with applications to multidimensional approximation, *Math. Comput.* 27 (123) (1973) 595–605.
- [34] Tong Qin, Zhen Chen, John Jakeman, Dongbin Xiu, Data-driven learning of non-autonomous systems, preprint, arXiv:2006.02392, 2020.
- [35] Tong Qin, Kailiang Wu, Dongbin Xiu, Data driven governing equations approximation using deep neural networks, *J. Comput. Phys.* 395 (2019) 620–635.
- [36] Maziar Raissi, Paris Perdikaris, George Em Karniadakis, Machine learning of linear differential equations using Gaussian processes, *J. Comput. Phys.* 348 (2017) 683–693.
- [37] Patrick AK Reinbold, Daniel R. Gurevich, Roman O. Grigoriev, Using noisy or incomplete data to discover models of spatiotemporal dynamics, *Phys. Rev. E* 101 (1) (2020) 010203.
- [38] Cynthia Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, *Nat. Mach. Intell.* 1 (5) (2019) 206–215.
- [39] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, J. Nathan Kutz, Data-driven discovery of partial differential equations, *Sci. Adv.* 3 (4) (2017) e1602614.
- [40] Samuel H. Rudy, J. Nathan Kutz, Steven L. Brunton, Deep learning of dynamics and signal-noise decomposition with time-stepping constraints, *J. Comput. Phys.* 396 (2019) 483–506.
- [41] Hayden Schaeffer, Learning partial differential equations via data discovery and sparse optimization, *Proc. R. Soc. A, Math. Phys. Eng. Sci.* 473 (2197) (2017) 20160446.
- [42] Hayden Schaeffer, Scott G. McCalla, Sparse model selection via integral terms, *Phys. Rev. E* 96 (2) (2017) 023302.
- [43] Hayden Schaeffer, Giang Tran, Rachel Ward, Extracting sparse high-dimensional dynamics from limited data, *SIAM J. Appl. Math.* 78 (6) (2018) 3279–3295.
- [44] Kunihiko Taira, T. Colonius, The immersed boundary method: a projection approach, *J. Comput. Phys.* 225 (2) (August 2007) 2118–2137.
- [45] Stephan Thaler, Ludger Paepler, Nikolaus A. Adams, Sparse identification of truncation errors, *J. Comput. Phys.* 397 (2019) 108851.
- [46] Karl Thomaseth, Claudio Cobelli, Generalized sensitivity functions in physiological system identification, *Ann. Biomed. Eng.* 27 (5) (1999) 607–616.
- [47] Benjamin A. Toms, Elizabeth A. Barnes, Imme Ebert-Uphoff, Physically interpretable neural networks for the geosciences: applications to earth system variability, *J. Adv. Model. Earth Syst.* 12 (9) (2020) e2019MS002002.
- [48] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, Michael P.H. Stumpf, Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems, *J. R. Soc. Interface* 6 (31) (February 2009) 187–202.
- [49] Giang Tran, Rachel Ward, Exact recovery of chaotic systems from highly corrupted data, *Multiscale Model. Simul.* 15 (3) (2017) 1108–1129.
- [50] Wen-Xu Wang, Rui Yang, Ying-Cheng Lai, Vassiliios Kovanci, Celso Grebogi, Predicting catastrophes in nonlinear dynamical systems by compressive sensing, *Phys. Rev. Lett.* 106 (15) (2011) 154101.
- [51] Yating Wang, Siu Wun Cheung, Eric T. Chung, Yalchin Efendiev, Min Wang, Deep multiscale model learning, *J. Comput. Phys.* 406 (2020) 109071.
- [52] Z. Wang, X. Huan, K. Garikipati, Variational system identification of the partial differential equations governing microstructure evolution in materials: inference over sparse and spatially unrelated data, *Comput. Methods Appl. Mech. Eng.* 377 (2021) 113706.
- [53] Zhenlin Wang, Xun Huan, Krishna Garikipati, Variational system identification of the partial differential equations governing the physics of pattern-formation: inference under varying fidelity and noise, *Comput. Methods Appl. Mech. Eng.* 356 (2019) 44–74.
- [54] Zhenlin Wang, Bowei Wu, Krishna Garikipati, Xun Huan, A perspective on regression and bayesian approaches for system identification of pattern formation dynamics, *Theor. Appl. Mech. Lett.* 10 (3) (2020) 188–194.
- [55] David J. Warne, Ruth E. Baker, 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) (June 2019) 1760–1804.
- [56] Hulin Wu, Lang Wu, Identification of significant host factors for HIV dynamics modelled by non-linear mixed-effects models, *Stat. Med.* 21 (5) (March 2002) 753–771.
- [57] Kailiang Wu, Dongbin Xiu, Numerical aspects for approximating governing equations using data, *J. Comput. Phys.* 384 (2019) 200–221.
- [58] Kailiang Wu, Dongbin Xiu, Data-driven deep learning of partial differential equations in modal space, *J. Comput. Phys.* 408 (2020) 109307.
- [59] Hao Xu, Haibin Chang, Dongxiao Zhang Dlga-pde, Discovery of pdes with incomplete candidate library via combination of deep learning and genetic algorithm, *J. Comput. Phys.* (2020) 109584.
- [60] Linan Zhang, Hayden Schaeffer, On the convergence of the SINDy algorithm, *Multiscale Model. Simul.* 17 (3) (2019) 948–972.

- [61] Sheng Zhang, Guang Lin, Robust data-driven discovery of governing physical laws with error bars, Proc. R. Soc. A, Math. Phys. Eng. Sci. 474 (2217) (2018) 20180305.
- [62] Sheng Zhang, 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, preprint, arXiv:1907.07788, 2019.