

## SINDy-BVP: Sparse identification of nonlinear dynamics for boundary value problems

Daniel E. Shea<sup>1,\*</sup> Steven L. Brunton<sup>2</sup>, and J. Nathan Kutz<sup>3</sup>

<sup>1</sup>Department of Materials Science and Engineering, University of Washington, Seattle, Washington 98195, USA

<sup>2</sup>Department of Mechanical Engineering, University of Washington, Seattle, Washington 98195, USA

<sup>3</sup>Department of Applied Mathematics, University of Washington, Seattle, Washington 98195, USA



(Received 31 August 2020; accepted 7 June 2021; published 29 June 2021)

We develop a data-driven model discovery and system identification technique for spatially-dependent boundary value problems (BVPs). Specifically, we leverage the *sparse identification of nonlinear dynamics* (SINDy) algorithm and group sparse regression techniques with a set of forcing functions and corresponding state variable measurements to yield a parsimonious model of heterogeneous material systems. The technique models forced systems governed by linear or nonlinear operators of the form  $L[u(x)] = f(x)$  on a prescribed domain  $x \in [a, b]$ . We demonstrate the approach on a range of example systems, including Sturm-Liouville operators, beam theory (elasticity), and a class of nonlinear BVPs. The generated data-driven model is used to infer the governing operator and spatially-dependent parameters that describe the heterogenous, physical quantities of the system. Our SINDy-BVP framework enables the characterization of a broad range of systems, including for instance, the discovery of anisotropic materials with heterogeneous variability.

DOI: 10.1103/PhysRevResearch.3.023255

### I. INTRODUCTION

Boundary value problems (BVPs) are ubiquitous in the engineering and physical sciences [1,2]. From heat transfer to elasticity, many fundamental technologies developed in the 20th century are formulated as linear BVPs whose solutions are used in engineering design. For example, the semiconductor industry developed many critical technologies and chip architectures by solving BVPs that characterize the underlying quantum, thermal, and electromagnetic physics. Modern BVPs of interest often arise in complex systems characterized by nonlinearity and spatial heterogeneity, thus rendering standard analytic and computational techniques intractable since the governing equations and spatial variability are often unknown. Indeed, the governing BVPs for many emerging applications are often unknown and/or their spatial dependencies undetermined. Modern anisotropic material system design provides a canonical example of the ability to leverage nonlinearity and heterogeneity in order to produce remarkable new materials. Data-driven methods provide a potential theoretical framework for characterizing such materials by discovering both the governing BVPs (linear and nonlinear) and their spatial dependencies through measurements alone. Toward this goal, we develop a sparse regression framework, previously used for the discovery of dynamical systems, in

order to discover interpretable and parsimonious BVPs and their spatial dependencies.

The formulation of many canonical problems in physics resulted in the first BVPs. From as early as 1822, when Fourier formulated and solved the heat equation [3], BVPs played a central role in electromagnetism, wave propagation, quantum mechanics, and elasticity. Many of these BVPs resulted from applying a space-time separation of variables decomposition to a governing partial differential equation (PDE). In different geometries and dimensions, the solutions to many canonical BVPs became known as special functions: Bessel, Laguerre, Hermite, Legendre, Chebyshev, spherical harmonics, radial basis, etc. More broadly, these canonical linear equations of mathematical physics were unified under the aegis of Sturm-Liouville theory. The impact of Sturm-Liouville theory in the 20th century is difficult to overestimate given its enormous breadth of applications ranging from the underlying theory of quantum mechanics to the propagation of electromagnetic energy in waveguides. The BVP theory for these two applications arise from a separation of variables solution of the Schrödinger equation and Maxwell's equations, respectively.

Linear BVPs are amenable to a number of solution strategies, foremost among these being eigenfunction expansions [1]. Such a solution technique is highly advantageous given the interpretability of the eigenfunctions (e.g., quantum mechanical states or propagating waveguide modes) and the many guaranteed mathematical properties of Sturm-Liouville operators, including an orthonormal and complete basis of real eigenfunctions with real eigenvalues for representing solutions. In addition to eigenfunction expansions, there are other methods for generating solutions to BVPs. Most notably is the Green's function [2], which provides an inverse to the Sturm-Liouville operator that can be used to evaluate any forcing of the governing BVP through integration over the

\*sheadan@uw.edu

so-called fundamental (Green's function) solution. These two traditional and ubiquitous mathematical methods rely on a critical property: linear superposition. Thus any solution can be constructed as a sum of the eigenfunctions appropriately weighted, or the integral (sum) over the fundamental solution. Nonlinear BVPs cannot be handled with such mathematical techniques. Moreover, the spatially varying coefficients of either a linear or nonlinear operator typically requires computational methods to produce solutions. Thus, in many emerging BVPs in the physical and engineering sciences, classical methods, which rely on linearity to produce interpretable eigenfunctions or fundamental solutions are ineffective for characterizing the system.

Modern BVPs in science and engineering, which generically take the form  $L[u(x)] = f(x)$  with the state variable  $u(x)$  and forcing  $f(x)$ , are typically characterized by the operator  $L$ , which is nonlinear and highly heterogeneous in nature, rendering many of our traditional linear solution strategies ineffectual. This dilemma prohibits development of interpretable solutions, such as Green's functions, which describe the impulse response of the system or eigenfunctions, which may describe eigenfrequencies in vibration analysis. Historically, many approaches to this problem have focused on modifying linear models to approximate the nonlinear effects of nonlinear systems. For example, perturbation theory has been used to effectively model weakly nonlinear systems. Perturbation theory has been applied to a wide variety of nonlinear problems including, among others, nonlinear anisotropic material modeling. These models generally focus on the observed macroscopic system response to applied external stimuli and the agreement between derived theoretical models and experimental data [4–8]. Entire texts have been written on the subject of anisotropic heterogeneous materials modeling [9], and research in the area remains active.

In this paper, we propose a mathematical framework for identifying the governing operator  $L$ , including its spatially-varying coefficients, to provide an interpretable understanding of nonlinear and heterogeneous steady-state BVP systems by discovering the spatially-varying coefficients and the operator structure. The ability to recover the governing operator not only provides insight into the physics, it also enables modern computational tools for generating numerical solutions of the problem since they require governing equations (i.e., shooting methods or relaxation methods). In many materials systems, spatially-varying parametric coefficients in the operator  $L$  are directly tied to the properties of materials in the system, which can provide scientists and engineers with physical intuition about their systems. In anisotropic and heterogeneous media, the materials' properties vary with composition and structure, and the mapping between spatial position and local material properties (e.g., heat transfer coefficients, conductivity, diffusivity, or porosity) are often not known. Spatially-localized changes in composition and structure can yield significantly different response to external stimuli. Although this paper focuses on material science applications, which provide great canonical examples of heterogeneous systems, this mathematical architecture for BVP discovery is fundamentally domain-agnostic and highly flexible.

We propose the SINDy-BVP framework, which utilizes data-driven modeling to learn BVP operators directly from

data. Our sparse regression framework, which is based upon the *sparse identification of nonlinear dynamics* (SINDy) [10] algorithm, gives rise to parsimonious models characterizing the BVP. Our SINDy-BVP framework can identify linear or nonlinear governing equations and/or spatially-varying parametric coefficients of the system from measurement data alone, providing a robust model discovery framework for BVPs. Examples are provided for the Sturm-Liouville operator, a nonlinear modification of the Sturm-Liouville operator, and two illustrative anisotropic, heterogeneous material systems.

In the case of the materials systems, it is important to note that data-driven modeling represents a paradigm shift in materials modeling. The current standard approach is to model heterogeneous material by their effective macroscopic properties. In contrast, data-driven modeling aims to map the spatially-local material properties.

To our knowledge, there are no other algorithms that focus on identifying the governing operator of a boundary value problem *and* its spatially-varying coefficients simultaneously from data. In [11,12], a known conservation law or operator is applied to a boundary value problem and a data-driven approach is employed to develop a model where spatially-varying parameters match experimental data set. A recent approach developed a novel neural network architecture to approximate the operator governing a system and compute new solutions to a given system, but it fails to identify the specific parametric coefficients in the operator, nor the form of the operator in an interpretable form [13]. One approach seeks to find a constitutive relationship between measured variables and applied external forces, but does not identify differential operators thus preventing future numerical solutions from being generated [14]. Other approaches utilize a combination of two separate modules where one module solves the BVP and the other parameterizes the spatially-varying properties of the system [15–20]. However, one of the two modules is a neural network in all of these two-module approaches, thus preventing discovery of a fully interpretable model. The application of different data-driven modeling approaches for materials modeling has been studied in [21], thus underscoring the importance of this subject.

The paper is outlined as follows: Section II gives a short background of the SINDy algorithm used extensively in this work. Section III then formulates the SINDy architecture with boundary value problems for discovery of governing equations and/or their spatially dependent variations. The method developed is applied to a broad range of problems in Sec. IV, including nonlinear boundary value problems. The paper is concluded in Sec. V with an overview of the method and a discussion of its outlook on modern nonlinear and heterogeneous BVPs.

## II. BACKGROUND

This work extends the SINDy family [10,22,23] of algorithms to learn the differential operator  $L$  in BVPs of the form  $L[u(x)] = f(x)$ , along with parametric and spatial heterogeneous dependencies. SINDy is a model discovery algorithm originally designed to discover governing equations for nonlinear dynamical systems. This method uses a sparse

regression framework with a large library of candidate physics models to determine governing equations for physical systems that are often characterized with relatively few terms. This makes the governing equation sparse in the space of possible candidate functions included in the library. SINDy considers dynamical systems of the form

$$\dot{\mathbf{u}} = \frac{d}{dt}\mathbf{u}(t) = \mathbf{N}(\mathbf{u}, \mathbf{u}^2, \dots, \sin(\mathbf{u}), \cos(\mathbf{u}), \dots), \quad (1)$$

where  $\mathbf{u}(t) \in \mathbb{R}^n$  represents the measured variables of the system at time  $t$ . The regression is formulated in a discrete matrix formulation where  $\mathbf{u}$  is measured at discrete snapshots in time  $t$ . The snapshots are used to form the matrices  $\mathbf{U}$  and  $\dot{\mathbf{U}}$ , where  $\dot{\mathbf{U}}$  is either directly measured or numerically computed from the snapshots  $\mathbf{u}(t)$ . If the interval  $[0, T]$  is discretized into  $m$  points, the two data matrices are the snapshot data matrix  $\mathbf{U}$ :

$$\mathbf{U} = \begin{bmatrix} u_1(t_1) & u_2(t_1) & \dots & u_n(t_1) \\ u_1(t_2) & u_2(t_2) & \dots & u_n(t_2) \\ \vdots & \vdots & & \vdots \\ u_1(t_m) & u_2(t_m) & \dots & u_n(t_m) \end{bmatrix},$$

and the matrix of corresponding time derivatives,  $\dot{\mathbf{U}}$ :

$$\dot{\mathbf{U}} = \begin{bmatrix} \dot{u}_1(t_1) & \dot{u}_2(t_1) & \dots & \dot{u}_n(t_1) \\ \dot{u}_1(t_2) & \dot{u}_2(t_2) & \dots & \dot{u}_n(t_2) \\ \vdots & \vdots & & \vdots \\ \dot{u}_1(t_m) & \dot{u}_2(t_m) & \dots & \dot{u}_n(t_m) \end{bmatrix}.$$

Since  $\mathbf{u}(t)$  is an  $n$ -dimensional vector, then the matrices  $\mathbf{U}$  and  $\dot{\mathbf{U}} \in \mathbb{R}^{m \times n}$ . The system identification problem is formulated in matrix form as an overdetermined linear regression problem ( $\mathbf{Ax} = \mathbf{b}$ ) for learning the governing equations

$$\dot{\mathbf{U}} = \Theta(\mathbf{U})\mathbf{E}, \quad (2)$$

where the matrix  $\Theta(\mathbf{U}) \in \mathbb{R}^{m \times p}$  contains  $p$  column vectors, each representing a possible candidate term in the governing equation to be learned. These columns contain candidate symbolic functions for characterizing the governing equations  $\mathbf{N}$  in (1) by numerically evaluating the state-space at  $m$  discrete time points. The unknown coefficient matrix of loadings  $\mathbf{E} \in \mathbb{R}^{p \times n}$  is learned via sparse regression. Candidate model terms in  $\Theta$  can be excluded from the learned governing equation by setting the corresponding coefficient in  $\mathbf{E}$  to 0, which is naturally implemented by a sparse regression. The sparse regression minimizes the  $\ell_2$  reconstruction error (i.e.,  $\|\dot{\mathbf{U}} - \Theta\mathbf{E}\|_2$ ) while enforcing sparsity. Traditional sparse regression uses  $\ell_1$  (i.e., the absolute value  $\|\dot{\mathbf{U}} - \Theta\mathbf{E}\|$ ) regularization terms, which approximate the computationally challenging nonconvex idealized  $\ell_0$  (i.e., number of nonzero entries in  $\mathbf{E}$ ) regularization. In the SINDy algorithm, sparsity is achieved through an iterative thresholding procedure [10] whose convergence properties have been studied under various assumptions [24,25]. However, this problem can be solved using any sparse regression algorithm, such as lasso [26], sparse relaxed regularized regression (SR3) [24,27], stepwise sparse regression (SSR) [28], or Bayesian methods [29–31]. The iterative thresholding algorithm for SINDy is outlined in Algorithm 1.

### Algorithm 1. SINDy.

**Input:** Candidate functions  $\Theta$ ; time derivatives  $\mathbf{U}_t$ ; regularizer  $\lambda$ ; threshold  $\epsilon$ ; score function  $r(\mathbf{x}) = \|\mathbf{x}\|_2$ ; *iters*

**Output:** Candidate function loadings  $\mathbf{E}$

```

1: procedure
2:    $\mathbf{E} \leftarrow \text{argmin}_{\mathbf{E}'} \|\mathbf{U}_t - \Theta\mathbf{E}'\|_2$  ▷ Initial  $\mathbf{E}$  guess
3:   for  $i = 1, \dots, \text{iters}$  do
4:      $terms \leftarrow \{i : r(\mathbf{E}(i)) > \epsilon\}$  ▷ Threshold terms
5:      $\Theta \leftarrow \Theta[terms]$ 
6:      $\mathbf{E} \leftarrow \text{argmin}_{\mathbf{E}'} \|\mathbf{U}_t - \Theta\mathbf{E}'\|_2 + \lambda\|\mathbf{E}\|_2$  ▷ Repeat fit
7:   end for
8:   return  $\mathbf{E}$ 
9: end procedure

```

Classical SINDy works well for model discovery and system identification on problems where the terms in the governing equation can be well-represented in the candidate library ( $\Theta$ ) and where the learned terms have constant coefficients with respect to the independent variable(s) of the system (i.e., time-invariant constant coefficients). Parametric PDE-FIND was developed as an extension of the SINDy algorithm to accommodate dynamical systems governed by partial differential equations with time-variant or space-variant coefficients [23]. The PDE-FIND algorithm is modified for the data-driven modeling of BVPs with SINDy-BVP, with a special emphasis on operator identification and parametric coefficient estimation.

### III. METHODS

Our proposed method makes two specific innovations. First, the method learns the differential operator  $L$  for BVPs, including spatially varying coefficients. Second, the method is the first application of SINDy to time-invariant systems; all prior works focused on dynamical systems.

These innovations require the method to be adapted for use on BVPs. Previous SINDy methods, including PDE-FIND, were designed for dynamical systems. In dynamical systems, a system can be sampled indefinitely into the future from its initial state. This enables the creation of arbitrarily large data sets, sometimes consisting of thousands or tens of thousands of points. Furthermore, data sets can be easily enriched by sampling systems with different initial states. Neither of these approaches apply for BVP systems.

BVP systems are a more constrained environment for deploying the SINDy algorithm compared to dynamical systems. In BVPs, samples are constrained between spatial boundaries that specify the domain of the problem, therefore eliminating the possibility of sampling a system's dynamics indefinitely into the future. As previously studied, evaluating a system under a variety of different conditions helps identify the governing model in SINDy and similar algorithms for dynamical systems and is generally superior to increasing the number of samples on a single trajectory generated from one initial condition [32]. This motivates the need to study BVP systems in SINDy-BVP under a variety of different condi-

tions; SINDy-BVP achieves this by applying different forcing functions to the system.

In dynamical systems SINDy, the variable  $u = u(t)$  represents the dynamical state variable. In this paper, the variable  $u = u(x)$  is the state variable for steady-state BVP systems. SINDy-BVP learns the differential operator of a time-invariant system by subjecting the system to a collection of known spatially-varying forcing functions and measuring the system's response. Each response  $u_j(x)$  to a forcing function  $f_j(x)$  is recorded as a trial and each trial is governed by the relationship

$$\begin{aligned} L[u_j] &= f_j \\ j &= 1, 2, \dots, m \\ x &\in [a, b], \end{aligned} \quad (3)$$

where  $L$  is the linear or nonlinear differential operator to be discovered,  $x$  is the independent spatial variable,  $u_j(x)$  is the measured system state variable quantifying the system's response when subjected to the force  $f_j(x)$ , and there are  $m$  total trials. The  $f_j(x)$  are known applied forcing functions, which can be considered as *probes* for the system. The variable  $x$  is used to denote a single spatial scalar variable rather than a position vector. The interval  $x \in [a, b]$  defines the spatial region of interest, where  $x = a$  and  $x = b$  are the boundaries of the BVP. Although a variety of boundary conditions can be realized in physical systems, this work uses Dirichlet boundary conditions, which specify  $u(x = a)$  and  $u(x = b)$ . The general principle of SINDy-BVP is presented in Fig. 1 for an operator with two spatially-varying parameters,  $p(x)$  and  $q(x)$ .

### A. Problem statement

To begin, we assume  $L$  is a second-order differential operator. This operator order assumption will be relaxed in later sections (Sec. IV D). If  $L$  is second order, it is known that  $L[u]$  contains the term  $u_{xx}$ . If  $u_{xx}$  is in the governing equation  $L[u(x)] = f(x)$ , we assume it can be represented as some generalized function  $N$ , which contains  $f(x)$  and other terms in  $L$ :

$$u_{xx} = N(u, u^2, u^3, \dots, u_x, \dots, f(x)). \quad (4)$$

This is the BVP equivalent to (1). The BVP problem (4), which is formulated as a continuous variable over the domain  $x \in [a, b]$ , is discretized into  $n$  spatial locations. We assume these to be equally spaced measurements or discretization locations. The discretized function  $u(x)$  is mapped to the vector  $\mathbf{u} = [u(x_1) \ u(x_2) \ u(x_3) \ \dots \ u(x_n)]^T$  where  $x_1 = a$  and  $x_n = b$ . With the vectorization of the data, we can adopt the SINDy nomenclature and restate the sparse regression for BVPs as

$$\mathbf{U}_{xx} = \Theta(\mathbf{U}, \mathbf{F})\mathbf{\Xi}, \quad (5)$$

where  $\mathbf{U}_{xx}$  is the second spatial derivative of the discretized vector of the state space  $u(x)$ ,  $\Theta$  is a library of candidate basis functions believed to comprise  $N(\cdot)$ , and  $\mathbf{\Xi}$  is a vector of coefficients, which prescribe the loadings of the columns of  $\Theta$ . The coefficient vector can vary spatially with  $x$ , or more precisely, the discretization of  $x$ .

This regression uses input data consisting of matrices  $\mathbf{U} \in \mathbb{R}^{m \times n}$  and  $\mathbf{F} \in \mathbb{R}^{m \times n}$  with  $n$  discrete sampled spatial positions

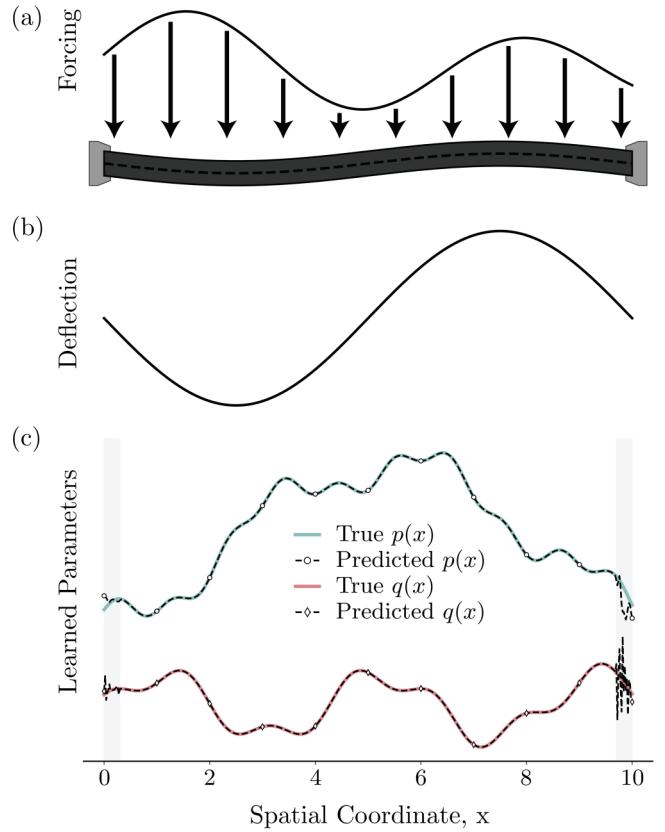


FIG. 1. SINDy-BVP studies steady-state systems subjected to a forcing function. One simple example system is a beam clamped at both ends subjected to a static load (a). The beam deflects (b) in response to the load, and the forcing function and deflection are used for data-driven modeling via SINDy-BVP to learn the parametric coefficients (c) in the governing operator. The coefficients  $p(x)$  and  $q(x)$  vary spatially. The coefficients are directly related to the beam's spatially-varying mechanical properties. The grey boxes in (d) indicate that error can occur in the learned coefficients near the boundaries.

and  $m$  unique trials or forcings. Each trial is a system response  $\mathbf{u}_j$  to a corresponding forcing function  $\mathbf{f}_j$  governed by the same operator  $L$ . The input data set  $\mathbf{U}$  and  $\mathbf{F}$  have the structure

$$\mathbf{U} = \begin{bmatrix} u_1(x_1) & u_1(x_2) & \dots & u_1(x_n) \\ u_2(x_1) & u_2(x_2) & \dots & u_2(x_n) \\ \vdots & \vdots & & \vdots \\ u_m(x_1) & u_m(x_2) & \dots & u_m(x_n) \end{bmatrix}, \quad (6)$$

$$\mathbf{F} = \begin{bmatrix} f_1(x_1) & f_1(x_2) & \dots & f_1(x_n) \\ f_2(x_1) & f_2(x_2) & \dots & f_2(x_n) \\ \vdots & \vdots & & \vdots \\ f_m(x_1) & f_m(x_2) & \dots & f_m(x_n) \end{bmatrix}. \quad (7)$$

Note this is different from dynamical systems SINDy, where  $m$  temporal snapshots of a dynamical system are sampled and the state vector  $\mathbf{u}(t)$  has  $n$  components. In SINDy-BVP, the  $\mathbf{U}$  samples the spatial positions  $x_1, x_2, \dots, x_n$  for  $m$  different trials, where each trial is forced by a different forcing function. Additionally, the outcome variable (or left-hand side) in this formulation is a spatial derivative of  $\mathbf{U}$ , *not* the typical  $\mathbf{U}_t$

seen in dynamical systems SINDy formulations. The spatial derivatives of  $\mathbf{U}$  are generated by numerical differentiation to produce  $\mathbf{U}_{xx}$ ,  $\mathbf{U}_{xxx}$ , and higher-order derivatives as needed. The numerically differentiated data is stacked as a vector and used as the outcome variable for the SINDy regression (5). The stacked vector  $\mathbf{U}_{xx}$  has the structure

$$\mathbf{U}_{xx} = \begin{bmatrix} u_{1,xx}(x_1) \\ u_{2,xx}(x_1) \\ \vdots \\ u_{m,xx}(x_1) \\ u_{1,xx}(x_2) \\ u_{2,xx}(x_2) \\ \vdots \\ u_{m,xx}(x_2) \\ u_{1,xx}(x_n) \\ u_{2,xx}(x_n) \\ \vdots \\ u_{m,xx}(x_n) \end{bmatrix}. \quad (8)$$

The candidate function matrix  $\Theta(\mathbf{U}, \mathbf{F}) \in \mathbb{R}^{(m \times n) \times (p \times n)}$  is constructed as a sparse block matrix to enable discovery of spatially-varying parametric coefficients. The library is constructed to allow a separate coefficient to be discovered for each spatial position,  $x_k$ .  $\Theta(\mathbf{U}, \mathbf{F})$  contains columns for each of the  $p$  symbolic candidate basis functions, each evaluated at the  $n$  spatial coordinates for all of the  $m$  trials. The candidate model functions included in the library  $\Theta(\mathbf{U}, \mathbf{F})$  are further described in Sec. III E. The matrix  $\Theta(\mathbf{U}, \mathbf{F})$  is a diagonal sparse matrix with the structure

$$\Theta = \begin{bmatrix} \Theta^{(1)} & & & \\ & \ddots & & \\ & & \Theta^{(k)} & \\ & & & \ddots \\ & & & & \Theta^{(n)} \end{bmatrix}, \quad (9)$$

where  $\Theta^{(k)} \in \mathbb{R}^{m \times p}$  is a symbolic function library with  $p$  candidate function columns evaluated for each of the  $m$  trials at spatial coordinates,  $x_k$ :

$$\Theta^{(k)} = \begin{bmatrix} u_{1,k} & \dots & (u_{1,k})_x & \dots & (u_{1,k})^2 & \dots & f_{1,k} \\ & & \vdots & & & & \\ u_{j,k} & \dots & (u_{j,k})_x & \dots & (u_{j,k})^2 & \dots & f_{j,k} \\ & & \vdots & & & & \\ u_{m,k} & \dots & (u_{m,k})_x & \dots & (u_{m,k})^2 & \dots & f_{m,k} \end{bmatrix},$$

where the subscripts  $j$  and  $k$  refer to the trial number and spatial coordinate, respectively. The library  $\Theta^{(k)}$  allows discovery of the parametric coefficients at spatial position  $x_k$ . This construction requires different forcings for each  $\Theta^{(k)}$  so that the regression (5) is not underdetermined and lacking insufficient constraints. The forcing functions  $f_j(x)$  are included in the library because they are known to influence the observed

behavior of the system. Further, as described in Sec. IID, they must be in the learned function  $N(\cdot)$ , and therefore must be included in the candidate term library  $\Theta(\mathbf{U}, \mathbf{F})$  to learn an accurate operator.

The problem is formulated as a group regression problem. Candidate model functions are tied together with a set of group indices  $G$ .  $G$  is a set of tuples of indices, where there are  $p$  tuples in the set  $G$  and each tuple contains  $n$  indices. Each tuple identifies related rows in  $\Theta$  and columns in  $\mathbf{U}$  that correspond to the same candidate function. For example, consider the candidate function  $u_{1,k}$  in the library  $\Theta^{(k)}$ . It is the candidate function in the first of  $p$  columns in  $\Theta^{(k)}$ . There is a tuple of indices that can be constructed, which refers to the first column of every matrix in equation (9) (i.e.,  $\Theta^{(1)}, \Theta^{(k)}, \dots, \Theta^{(n)}$ ). This example tuple would contain values  $(1, (p+1), (2p+1), \dots, (n-1)p+1)$ . The tuples in set  $G$  can be generated by the relationship  $G = \{g_l = l + (p \times k) : k = 1, \dots, n; l = 1, \dots, p\}$  where  $l$  counts through the  $p$  candidate functions and  $k$  counts through the  $n$  discrete spatial positions. Each tuple  $g_l \in G$  contains column indices of  $\Theta(\mathbf{U}, \mathbf{F})$  and row indices of  $\mathbf{U}$  corresponding to a single candidate function at all of the  $n$  discrete spatial positions.

Group sparsity is imposed to produce a solution, which is sparse in the space of possible candidate functions and where the coefficient can vary with spatial position  $x$ . Group sparse regression is performed using the *Sequential Grouped Threshold Ridge Regression* (SGTR) algorithm developed by Rudy [23]. An intuitive way of thinking about this approach is presented in Fig. 2. In the figure, a separate sparse regression is constructed for each of the  $n$  spatial coordinates. The regression aggregates data from  $m$  trials and enforces the solution's sparsity pattern across all of the spatial positions. As implied by the figure, this allows for inference of the operator  $L$  and its parametric coefficients.

## B. Sequential grouped threshold ridge regression

SGTR [23] is a group regression technique, which accomplishes group-level sparsity through an iterative thresholding process. This example assumes SINDy-BVP will be performed with the outcome variable  $\mathbf{U}_{xx}$ . Using the construction provided above, each group in  $G$  contains a set of indices, which represent columns of a single candidate term in  $\Theta(\mathbf{U}, \mathbf{F})$  at all spatial positions and its corresponding coefficient in  $\Theta$  at all spatial positions. The algorithm, shown in Algorithm 2, achieves sparsity at the group level through a combination of ridge regression and iterative thresholding across all groups.

The iterative thresholding loop in the SGTR algorithm progressively eliminates groups from  $\Theta$  by setting the columns in  $\Theta$  to zero. This thresholding imposes sparsity on the candidate functions in  $\Theta(\mathbf{U}, \mathbf{F})$  based on the candidate function's coefficient vector  $\mathbf{E}$ . The evaluation function  $r$  in this paper is the  $\ell_2$  norm, which means SGTR performs ridge regression and thresholds out candidate functions based on the  $\ell_2$  norm of the coefficient vector (i.e.,  $r(\mathbf{E}^{(g)}) = \|\mathbf{E}^{(g)}\|_2$ ). The result is a parsimonious function for  $\mathbf{U}_{xx}$  where the nonzero coefficients are allowed to vary at each spatial position  $x_k$ .

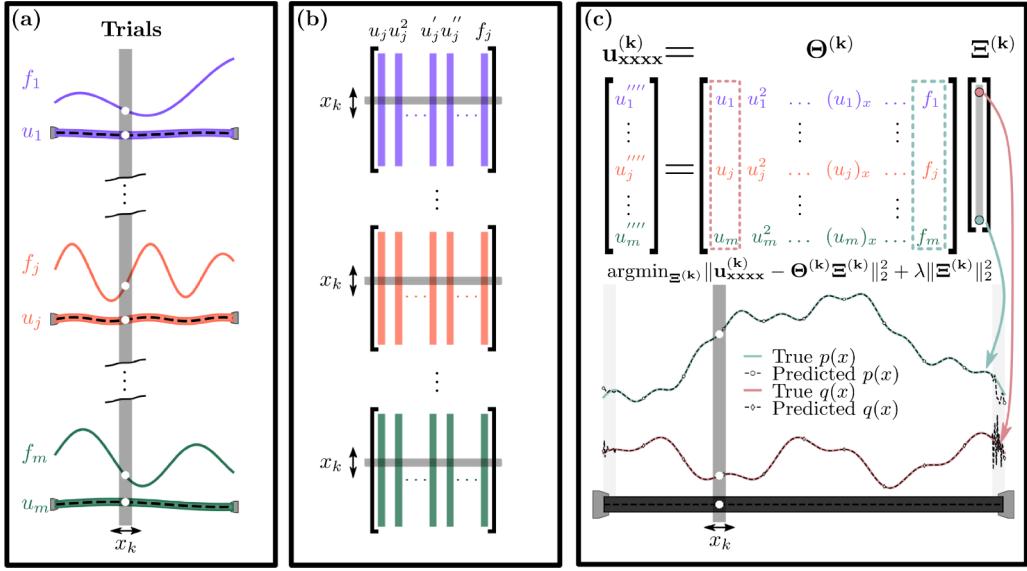


FIG. 2. Overview of constructing the data sets and regression for each discrete spatial point in the data set. Part (a) shows a collection of trials, where different forcings ( $f_j(x)$ ) are applied to a system yielding different responses ( $u_j(x)$ ). A matrix containing a library of candidate terms  $\Theta(U, F)$  is produced for each trial, where the rows are each a spatial point  $x_k$  and each column contains a candidate function, as seen in part (b). A sliding procedure is used to select rows of a single  $x_k$  from the libraries in part (b) to produce an aggregated library  $\Theta^{(k)}$  for each  $x_k$  in the data set. In (c), the regression is performed for each  $x_k$  to produce a vector of the parametric coefficients  $p(x)$  and  $q(x)$  at each  $x_k$ . The regression in (c) is a Ridge regression algorithm. However, a sparsity constraint is applied by an iterative thresholding and grouping mechanism of the algorithm.

### C. Model selection

The optimal model is selected from a set of candidate models generated by varying the thresholding value  $\epsilon$  in the SGTR algorithm. A range of tolerance values  $\epsilon$  are computed by

$$\epsilon_{\max} = \max_{g \in G} \|\Xi_{\text{ridge}}^{(g)}\|_2$$

$$\epsilon_{\min} = \min_{g \in G} \|\Xi_{\text{ridge}}^{(g)}\|_2$$

$$\Xi_{\text{ridge}} = (\Theta(U, F)^T \Theta(U, F) + \lambda I)^{-1} \Theta(U, F)^T \mathbf{U}_{xx},$$

where  $\epsilon_{\max}$  and  $\epsilon_{\min}$  are the highest and lowest tolerances that affect the sparsity of the predicted model, and  $\lambda$  is a regularization constant. The ridge regression regularization constant is held constant at  $\lambda = 10^{-5}$  for all problems in this paper.

#### Algorithm 2. Sequential grouped threshold ridge regression.

**Input:** Candidate functions; derivatives  $\mathbf{U}_{xx}$ ; groups  $G$ ; regularizer  $\lambda$ ; threshold  $\epsilon$ ; score function  $r(\mathbf{x}) = \|\mathbf{x}\|_2$ ;  $iters$

**Output:** Candidate function coefficients  $\Xi$

```

1: procedure
2:    $\Xi \leftarrow \text{argmin}_{\Xi'} \|\mathbf{U}_{xx} - \Theta \Xi'\|_2$             $\triangleright$  Initial  $\Xi$  guess
3:   for  $i = 1, \dots, iters$  do
4:      $P \leftarrow \{g_i \in G : r(\Xi^{(g_i)}) < \epsilon\}$             $\triangleright$  Select groups
5:      $\Xi^{(P)} \leftarrow 0$                                       $\triangleright$  Set to zero
6:      $\Xi \leftarrow \text{argmin}_{\Xi'} \|\mathbf{U}_{xx} - \Theta \Xi'\|_2 + \lambda \|\Xi\|_2$   $\triangleright$  Repeat regression
7:   end for
8:    $\Xi^{(G)} \leftarrow \text{argmin}_{\Xi'} \|\mathbf{U}_{xx} - \Theta^{(G)} \Xi'^{(G)}\|_2$   $\triangleright$  Final fit
9:   return  $\Xi$ 
10: end procedure

```

At a thresholding value of  $\epsilon_{\max}$ , all coefficients are set to 0 after the first thresholding step with SGTR. Conversely, using  $\epsilon_{\min}$  as the thresholding tolerance would not eliminate any candidate functions with SGTR. A number of values, typically 50, spaced logarithmically between  $\epsilon_{\min}$  and  $\epsilon_{\max}$  are used to compute the candidate models.

The optimal model is then selected from the candidate models by choosing the model which minimizes the PDE-FIND loss function [23]:

$$\mathcal{L} = N \ln \left( \frac{\|\Theta(U, F)\Xi - \mathbf{U}_{xx}\|_2^2}{N_{\text{rows}}} + \beta \right) + 2k, \quad (10)$$

where  $k$  is the number of nonzero coefficients in the identified model ( $k := \|\Xi\|_0/m$ ),  $\beta$  is a small constant, and  $N_{\text{rows}} := m \times n$  is the number of rows in  $\Theta(U, F)$ . The loss function used to select the model assumes there is error in numerical differentiation used to compute  $\mathbf{U}_{xx}$ , and thus a model that minimizes only mean squared error ( $\|\Theta(U, F)\Xi - \mathbf{U}_{xx}\|_2^2$ ) is likely overfit. Overfit models are balanced by the parameter  $\beta$ , which allows for some misfit and simultaneously prevents the occurrence of  $\ln(0)$  in the loss function. The constant  $\beta$  is fixed in this work as  $\beta = 10^{-6}$ .

Similar to previous SINDy works, the algorithm exhibits improved performance when each candidate function is normalized to unit length [10]. When constructing the block diagonal matrix  $\Theta(U, F)$ , the entries  $\Theta^{(k)}$  are stacked and each column is normalized to unit length. More precisely, the matrix  $\hat{\Theta} \in \mathbb{R}^{(m \times n) \times p}$ , which is assembled as  $\hat{\Theta}^T = [\Theta^{(1)}, \dots, \Theta^{(k)}, \dots, \Theta^{(n)}]$ .  $\hat{\Theta}$  is normalized column-wise over each of the  $p$  columns, each containing a candidate function. Similarly, the outcome variable vector (e.g.,  $\mathbf{U}_{xx}$ ) is normalized such that  $\|\mathbf{U}_{xx}\| = 1$ .

#### D. Learning the operator $L$

In the previous sections, a method was described for learning a function that describes  $u_{xx}(x)$ , but without connecting that function to the operator. In this section, we will show how the operator  $L$  can be inferred from the learned function for  $u_{xx}(x)$ . Using a simple ansatz that the differential operator is at least second order, it is presumed the operator contains a  $u_{xx}(x)$  term. For this example, SINDy-BVP learns a function  $N(\cdot)$ , which approximates  $u_{xx}(x)$ :

$$u_{xx}(x) = N(f, u, u^2, u^3, u_x, \dots).$$

If  $u_{xx}(x)$  is in the operator  $L$  and the learned model satisfies the relationship  $Lu = f$ , then the model learned for  $N(\cdot)$  must contain  $f$ . Specifically, the model may take the form

$$u_{xx}(x) = N(f, u, u^2, u^3, u_x, \dots) = \sum_{g \in G} \xi^{(g)} \Theta^{(g)}, \quad (11)$$

where the symbolic candidate functions with nonzero coefficient vectors in  $G$  must include  $F$ . The learned model can then be rearranged to satisfy the relationship  $L[u] = f$ , where the left-hand side of the equation contains the terms in  $L$  and the right-hand side is described by the forcing functions. An example demonstrating discovery of the operator is provided within the Supplemental Material [33].

There is a simpler formulation with  $f(x)$  as the outcome variable (or left-hand side term) in the regression, which theoretically provides the opportunity to directly learn the operator  $L$  through a regression of the form  $\mathbf{F} = \Theta(\mathbf{U}, \mathbf{F})\Xi$ . However, including a numerically accurate  $\mathbf{F}$  in the library  $\Theta$  improves the ability of SINDy-BVP to handle noise while identifying the operator and its parameters. If  $\mathbf{F}$  also contained significant noise, there may not be an advantage to this construction.

#### E. Candidate function library

The candidate function library,  $\Theta(\mathbf{U}, \mathbf{F})$ , contains columns for derivatives of  $u(x)$ , nonlinearities of  $u(x)$ , and forcing functions  $f(x)$ . In all cases,  $\Theta$  contains  $u(x)$ , and polynomials of  $u(x)$  up to fifth order.

The derivatives in the library depend on the outcome variable. Assume the outcome variable for SINDy-BVP is  $\mathbf{U}_{xxxx}$ , the discrete form of  $d^4 u(x)/dx^4 j$ . In this case,  $\Theta(\mathbf{U}, \mathbf{F})$  contains derivatives  $d^a u(x)/dx^a$  of order  $a$ , for integers  $0 < a < A$ . For example, if  $\mathbf{U}_{xxxx}$  is the outcome variable, the library contains columns for the derivatives  $u_x(x)$ ,  $u_{xx}(x)$ , and  $u_{xxx}(x)$ . Furthermore, the products of  $u(x)$  and nonlinearities in  $u(x)$  with the spatial derivatives of  $u(x)$  are included in  $\Theta$  (e.g.,  $uu_x$  and  $u^2 u_{xx}$ ). Finally, a column for the forcing functions is included in  $\Theta$  containing all data in  $\mathbf{F}$ .

In general, the constructed candidate basis function library must include the basis functions in the governing model. If the terms contained in the operator are *not* present in the library, the learned governing operator will be inaccurate and/or incomplete. These failure modes have been previously discussed in [10].

## IV. COMPUTATIONAL RESULTS

#### A. Boundary value problem models

The models used in this work are solved on the interval  $x \in [0, 10]$  using the shooting method [34] with 1000 grid points. A tolerance of 0.001 is used for the right-side boundary condition, such that solutions, which aim to achieve  $u(x = 10) = 0$  can have an actual value  $u(x = 10) \in [-0.001, 0.001]$ . The following subsections describe the models used for this work.

##### 1. Linear Sturm-Liouville

Sturm-Liouville form operators are an extremely common class of linear, self-adjoint, Hermitian operators. Sturm-Liouville theory is especially important in engineering applications, and its study focuses on operators of the form in Eq. (12):

$$L[u] = [-pu_x]_x + qu \quad x \in [0, 10], \quad (12)$$

where the state variable  $u(x)$  is a function of the spatial variable  $x$ , and  $p(x)$  and  $q(x)$  are in general functions of the spatial variable. In our example model, the parametric coefficients are described by the functions

$$\begin{aligned} p(x) &= 0.5 \sin(x) + 0.1 \sin(12x) + 0.25 \cos(4x) + 2, \\ q(x) &= 0.4 \sin(3x) + 0.15 \cos(8x) + 1. \end{aligned}$$

The boundary conditions  $u(0) = 0$  and  $u(10) = 0$  are enforced for solutions of this model. The parametric coefficients in this model,  $p(x)$  and  $q(x)$ , were selected to provide an example to demonstrate SINDy-BVP on a linear model with rapidly-changing spatially-varying coefficients.

##### 2. Nonlinear Sturm-Liouville

A quadratic nonlinearity can be introduced to the Sturm-Liouville model in the following form

$$L[u] = [-pu_x]_x + qu + \alpha qu^2 \quad x \in [0, 10], \quad (13)$$

where  $\alpha$  controls the extent of nonlinearity in the term  $\alpha qu^2$ . The value  $\alpha = 0.4$  is used. The parametric coefficients  $p(x)$  and  $q(x)$  are described by

$$\begin{aligned} p(x) &= 0.5 \sin(x) + 0.1 \sin(11x) + 0.25 \cos(4x) + 3, \\ q(x) &= 0.6 \sin(x + 1) + 0.3 \sin(2.5x) + 0.2 \cos(5x) + 1.5. \end{aligned}$$

Boundary conditions  $u(0) = 0$  and  $u(10) = 0$  are used for this model. Again, the parametric coefficients in this model,  $p(x)$  and  $q(x)$ , were selected to demonstrate SINDy-BVP on a nonlinear model with rapidly changing spatially-varying coefficients.

##### 3. Linear second-order Poisson

Many simple physical systems are described by Poisson's equation. These elliptic differential equations are described by a Laplacian operator subjected to a force:  $\Delta u = f$ . In our system, a parametric coefficient describing a material property  $p(x)$  is introduced to the model.

$$L[u] = [-pu_x]_x \quad x \in [0, 10]. \quad (14)$$

Steady-state heat conduction is one example of a system that follows from this model. The coefficient  $p(x)$  could thus be

considered as thermal diffusivity (often  $\kappa$ ) of the material and is allowed to vary spatially. The material in this example system is a two-component composite that is anisotropic along the  $x$  coordinate and contains an exponentially-varying quantity of the two materials along the  $x$  direction. The model for  $p(x)$  in this problem is the simple arithmetic average:

$$p(x) = v_a(x)p_a + v_b(x)p_b,$$

where  $v_a(x)$  and  $v_b(x)$  are the volume fractions of component  $a$  and  $b$ , respectively, and vary spatially. The values  $p_a$  and  $p_b$  are the material properties for pure  $a$  and  $b$ . The components' material properties hold the value  $p_a = 12$  and  $p_b = 3$ , which do not change.

Although this model is simple and the arithmetic average often overestimates the true observed material properties of composites [9], it is instructive to consider the ability of SINDy-BVP to learn an operator for a system with a spatially varying anisotropic material property. The volume fraction of component  $b$  is described by an exponential decay function while component  $a$  makes up the remainder of the volume:

$$\begin{aligned} v_a(x) &= 1 - v_b(x), \\ v_b(x) &= 0.1 - 0.7 \exp(0.4x). \end{aligned}$$

A steady-state heat conduction problem, where one end has a higher temperature than the other, is modeled in this problem. Boundary conditions of  $u(0) = 0.8$  and  $u(10) = 0$  are applied.

#### 4. Euler-bernoulli beam theory

The Euler-Bernoulli beam theory uses the biharmonic fourth-order linear operator from elasticity theory to describe beam deflections given mechanical properties of the beam. The operator takes the form

$$L[u] = [-EIu_{xx}]_{xx} \quad x \in [0, 10], \quad (15)$$

where  $EI$  is the flexural rigidity of the material. In our model, the flexural rigidity varies spatially following a stepwise function as expected for a lamellar, laminate composite with the lamella oriented perpendicular to the  $x$  coordinate:

$$EI(x) = \begin{cases} 10 & 0 \leq x < 2, \\ 2.5 & 2 \leq x < 4, \\ 10 & 4 \leq x < 6, \\ 5 & 6 \leq x < 8, \\ 2.5 & 8 \leq x \leq 10 \end{cases}$$

The stepwise function  $EI(x)$  is a challenge for SINDy-BVP because of the discontinuities at the jumps in flexural rigidity, which occur at  $x = 2$ ,  $x = 4$ ,  $x = 6$ , and  $x = 8$ . The beam in this problem is considered clamped at both ends such that  $u(0) = 0$  and  $u(10) = 0$ .

#### 5. Forcing functions

The forcing functions for all examples are sinusoidal functions of the form  $a \sin(bx) + c$ . The amplitude  $a$ , frequency  $b$ , and positive offset  $c$  are selected from a set of values, which varies for each model. This family of functions was selected to enable rapid generation of a large data set with solutions of similar order of magnitude. The parameters  $a$ ,

$b$ , and  $c$  are selected for each system to produce solutions  $u(x)$  where  $\|u\|_\infty \approx 1$ . Tables and examples describing the range of parameters  $a$ ,  $b$ , and  $c$  are presented within the Supplemental Material [33] along with figures showing the solutions generated by the forcings and the spatially-varying parameters.

The approach for this work is to generate a large library of solutions to the problem  $L[u_j(x)] = f_j(x)$ , and then randomly subsample the library of solutions to test the SINDy-BVP algorithm. Prior works suggest that a variety of different system conditions must be tested to optimize discovery of the underlying governing equations of a system [32]. A weak formulation of SINDy also suggests that the optimal test functions to use for a system would have high values of their derivatives at points where the highest error in the model exists [35]. However, knowledge of the region with the highest model error would not be known prior to collecting data for a physical test system, as it would require fitting and testing a model. For this reason, we choose to randomly sample a database of randomly generated functions producing solutions of similar magnitude.

This approach is physically and experimentally relevant. In real systems, a number of different conditions could be tested and compiled into a database of forcings  $\mathbf{F}$  and corresponding responses  $\mathbf{U}$  on the discretized spatial vector  $\mathbf{x}$ .

#### B. Operator identification and parametric coefficient estimation

SINDy-BVP aims to achieve two primary goals: identification of the structure of a differential operator  $L$  and discovery of the parametric coefficients present in  $L$  for a forced system governed by the model  $L[u(x)] = f(x)$ . The method is applied to the four models described in Sec. IV A. Operator identification is only required in cases where the governing operator is unknown, and so two cases can be considered: known operator and unknown operator. The data used in this section is noise-free (up to numerical precision). Derivatives are computed using the finite differences method. Although this is physically unrealistic since measurements would introduce noise or rounding errors, this exercise provides insight into the capability of the method.

Figure 3 shows the four models used in this paper, three example trials used for training the SINDy-BVP models, and a plot of the parametric coefficients learned by SINDy-BVP compared to the true parameters in the operator  $L$  over the interval  $x \in [0, 10]$ . The parametric coefficient plots are taken from the case of an “unknown operator”, where both the operator and the parametric coefficients are learned by SINDy-BVP.

SINDy-BVP is effective at learning the coefficients  $p(x)$  and (if applicable)  $q(x)$  with relatively few trials for numerical precision data. Table I shows the number of trials required for SINDy-BVP to estimate the parametric coefficients to within 1% error for the middle 98% of the interval (i.e.,  $[0.1, 9.9]$ ). This metric is used to quantify the accuracy of learned coefficients because the error in the learned coefficients happens almost exclusively at the boundaries (inspect Fig. 3).

#### C. Effects of noise

The effect of noise is studied separately for the problems of operator identification and parameter estimation. Noise is

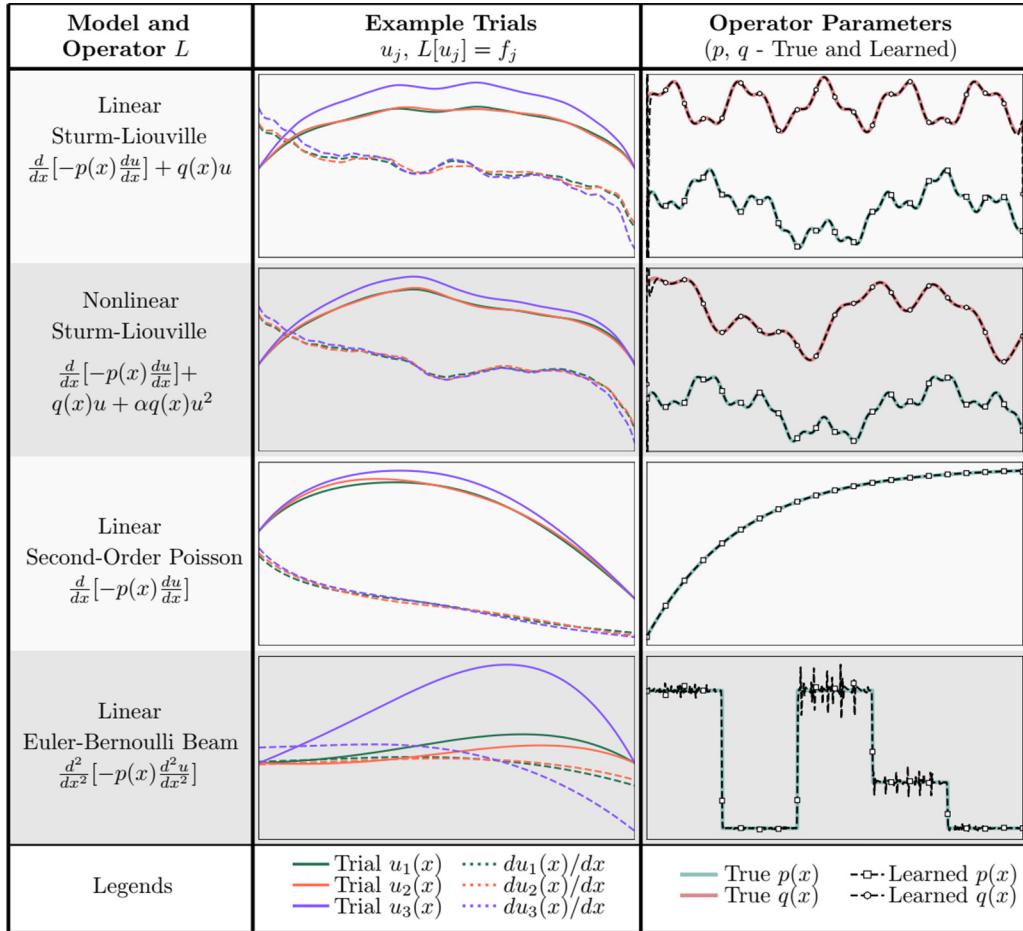


FIG. 3. Summary of the models and operators studied with SINDy-BVP. The center column shows three example trials from the training data set ( $u_1(x)$ ,  $u_2(x)$ , and  $u_3(x)$ ). The solutions are all of  $\mathcal{O}(1)$ . The final column shows the parametric coefficients in the operator, as well as the inferred parameters for clean data. Coefficients  $p(x)$  and  $q(x)$  are plotted with an offset, with markers every 30 points.

introduced to each system by applying Gaussian white noise to the measurement data in  $\mathbf{U}$  over each row,  $\mathbf{U}_j$ . The noise is defined by a signal-to-noise ratio (SNR) using the relationship  $\text{SNR} = 10 \log_{10}(\|u(t)\|_2^2 / \|\tilde{u}(t) - u(t)\|_2^2)$ . In order to enable differentiation of noisy input data, numerical differentiation is performed using a windowed Chebychev polynomial interpolation method. In the windowed interpolation method, a subset of 20 continuous data points is selected and fit to a fifth-order Chebychev polynomial. In the last example with a

TABLE I. Trials required for estimating spatial parametric coefficients within 1% error. Error is evaluated in the middle 98% of the problem domain (the interval  $x \in [0.1, 9.9]$ ) with the expression  $\|\mathbf{p}_{\text{learned}} - \mathbf{p}_{\text{true}}\|_2 / \|\mathbf{p}_{\text{true}}\|_2$ .

Trials required	Known $L$	Unknown $L$
Linear Sturm-Liouville	6	25
Nonlinear Sturm-Liouville	6	10
Linear second-order Poisson	2	8
Euler-Bernoulli beam theory <sup>a</sup>	4	4

<sup>a</sup>Parameter error in the Euler-Bernoulli beam model never decreases below 5% error due to numerical differentiation.

fourth-order derivative, 30 points are used in the window with a sixth-order polynomial. Derivatives of the fit polynomial function are used as derivative data for the regression.

### 1. Noisy operator identification

Operator identification is a challenging task for SINDy-BVP with noise. Prior works with SINDy have also described challenges in dealing with noise, so this is not a surprising finding. Figure 4(a) shows the effect of varying the SNR ratio, and Fig. 4(b) shows the effect of increasing the number of trials used in regression at a constant SNR of 100. The phrase “spurious terms” in Fig. 4 refers to the number of incorrectly identified monomial basis functions in the operator  $L$ , including both erroneous terms (terms that do not exist in the operator) and missing terms (terms that should be in the operator, but are not identified by the algorithm). The “erroneous terms” category is the most common error, where the SGTR algorithm fails to find a parsimonious model describing the system and includes additional terms to approximate the noisy training data accurately.

The Sturm-Liouville model shown in the top row of Fig. 4 show that the operator identification task succeeds with an  $\text{SNR} = 100$ , but begins to fail at lower SNR. Furthermore, the

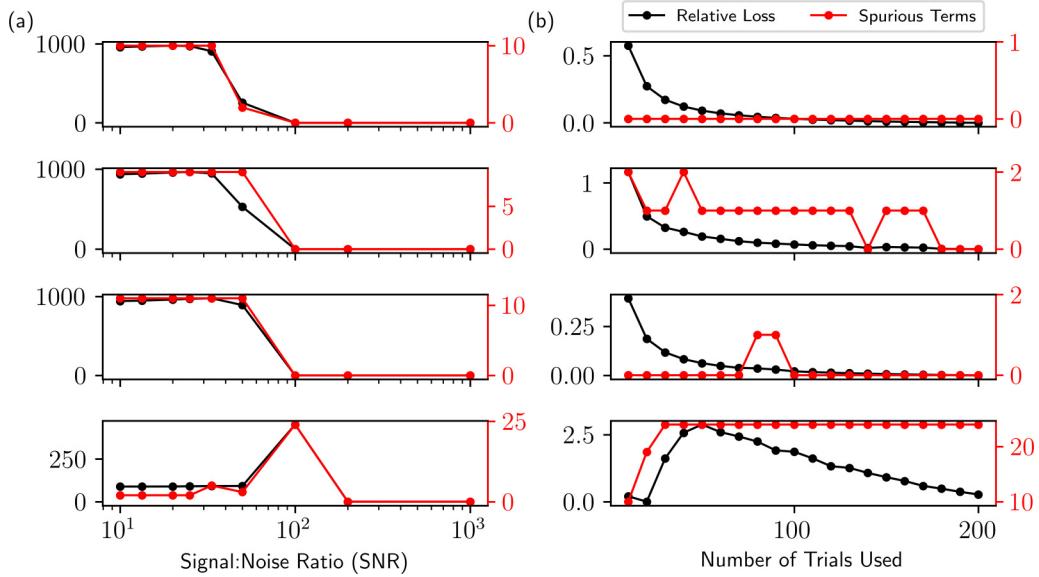


FIG. 4. SINDy-BVP succeeds at operator identification in signals with high SNR. A collection of trials with varying SNR is used for data-driven operator identification. The plots in (a) and (b), from top to bottom, are the linear Sturm-Liouville, nonlinear Sturm-Liouville, Poisson, and Euler-Bernoulli beam models. Relative loss [Eq. (10)] and a count of spurious terms in the identified model are quantified. In (a), 200 trials are used for regression and the SNR is varied. In (b), the number of trials is varied while holding a constant SNR of 100.

operator identification task succeeds with as little as 10 trials at  $\text{SNR} = 100$ , showing that SINDy-BVP is robust at high SNR. In the case of the nonlinear Sturm-Liouville model, over 180 trials are required to routinely identify the correct model at  $\text{SNR} = 100$ . With a constant 200 trials used, spurious terms show up in the learned function starting below the threshold of  $\text{SNR} = 100$ . Operator identification results are shown for the Poisson model in the third row. Similar to the linear and nonlinear Sturm-Liouville models, the Poisson model requires at minimum a SNR of 100. With an SNR of 100, the operator identification task succeeds for nearly any number of trials. However, a curious peak happens at 80 and 90 trials where an extra term is identified in the operator (specifically, “u” is added to the model). Despite the added term, the change in the loss value is relatively low, indicating that the added term makes a small difference in the accuracy of the model. Although SINDy-BVP identifies the extra term, the identified model is still relatively sparse and would provide an excellent starting point for parameter estimation. The Euler-Bernoulli beam, shown in the last row, is prohibitively challenging to identify with noise in the signal. Although signals with SNR over 200 can successfully identify the beam model, operator identification fails with any number of trials at SNR of 100.

One common error in model convergence is the inclusion of candidate model terms with a few “large” values in its coefficient vector  $\mathbf{E}^{(g)}$ . The large values pass the thresholding step, which is based on the  $\ell_2$  norm of the entire coefficient vector. This error effectively includes candidate terms in the final model that have relatively small influence on the model predictions, but that satisfy errors from noise in the training data. This error could potentially be mitigated by enforcing  $\ell_\infty$  constraints on the coefficient vectors, or by enforcing local smoothness of the learned coefficient.

In contrast, a term is occasionally excluded from a learned model. For example, let the differential operator be the linear

Sturm-Liouville form  $L[u] = -p(x)u_{xx} - p_x(x)u_x + q(x)u = f$ . If the training data exhibits relatively little contribution from the term  $-p_x(x)u_x$ , SINDy-BVP may exclude the  $u_x$  candidate term from the learned model. This results in an inaccurate model of  $L[u] = -p(x)u_{xx} + q(x)u$ . Although this model is missing the  $u_x$  term, it has relatively low loss function values and may be selected as the correct model.

These two errors account for the most common forms of error at low noise levels. At higher noise levels ( $>5\%$  noise, or  $\text{SNR} = 20$ ) the regression begins to add terms to the learned model to accommodate noise in the measurements, which is a common form of error in SINDy and other regression algorithms. The added terms tend to get added in groups, where the improved accuracy of the model from adding multiple additional terms outweighs the penalty for additional terms in the loss function [Eq. (10)]. This causes sharp changes in the plotted lines when different model types are identified.

## 2. Noisy parameter estimation

If the operator is known, the focus shifts towards estimating the spatially-dependent parametric coefficients. Figure 5 shows the effect of noise on parameter estimation by computing a “Coefficient Error”. The coefficient error is defined as  $E_p = \|\hat{\mathbf{p}} - \mathbf{p}\|_2 / \|\mathbf{p}\|_2$ , where  $\hat{\mathbf{p}}$  is the predicted coefficient vector and  $\mathbf{p}$  is the true coefficient vector. The task is quantified in a similar way to the operator identification task, where the SNR of input data and number of trials used for regression are varied. In Fig. 5(a), 200 trials are used as input data and the SNR is modulated between 10 and 1000. In Fig. 5(b), the SNR is fixed to 100 and the number of trials is varied between 10 and 200. For visual clarity in the plots, the coefficient error is capped at 1, so that  $\min(1, E_p)$  is plotted. Like the operator identification task, added terms tend to get added in groups, where the improved accuracy of the model from

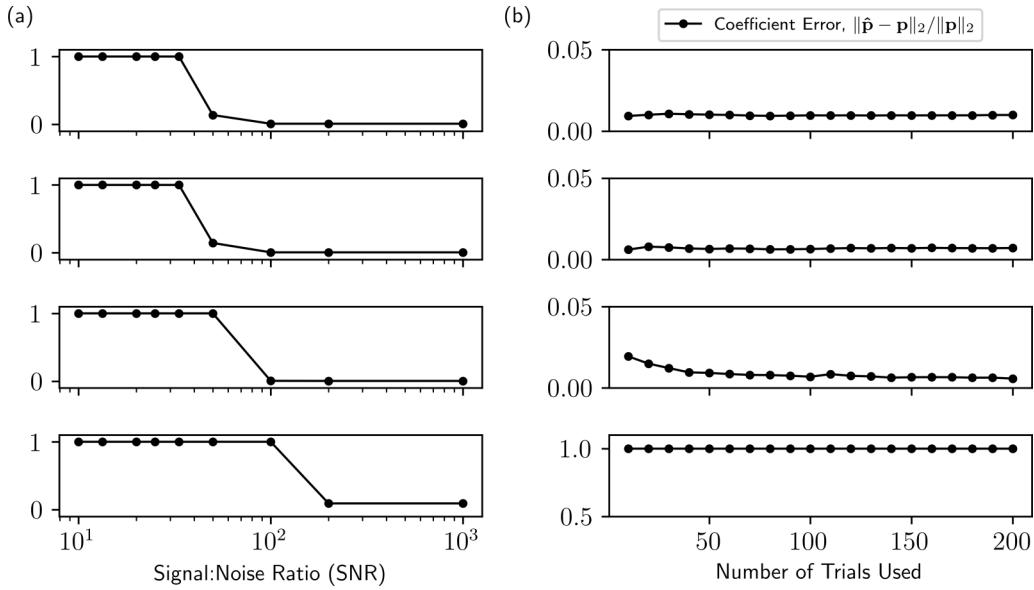


FIG. 5. Parametric coefficient estimation in noisy data. With a known operator  $L$ , SINDy-BVP can estimate the parametric coefficients. Coefficient error is used to compare the effect of varying SNR (a) and the number of trials used as input data (b). In (a), the number of trials used as input data are held constant at 200 and in (b) the SNR is fixed at 100. The plots in (a) and (b), from top to bottom, are the linear Sturm-Liouville, nonlinear Sturm-Liouville, Poisson, and Euler-Bernoulli beam models.

adding multiple additional terms outweighs the penalty for additional terms in the loss function [Eq. (10)].

The parameter estimation task is successful for all four models at SNR greater than 200. Parameters for the linear Sturm-Liouville, nonlinear Sturm-Liouville, and Poisson models can also be estimated at SNR of 100, and the Sturm-Liouville models are somewhat successful at SNR = 50. With a fixed SNR = 100, the parameters for the linear Sturm-Liouville, nonlinear Sturm-Liouville, and Poisson models can be accurately identified within about 5% for any number of trials over 10 trials. However, the parameters for the Euler-Bernoulli Beam equation cannot be accurately identified with even 200 trials at SNR of 100. Similar to the operator identification task, SINDy-BVP appears to succeed in most cases with SNR greater than 100. These results indicate that collection of clean data is the most important aspect of using SINDy-BVP for operator identification in experimental systems. Hyperparameter tuning and data filtering may improve these results, but aiming for an SNR over 100 is a critical step for practical use of SINDy-BVP.

#### D. Model differential order selection

This section addresses the need to identify the derivative order of the model's left-hand side. In the Euler-Bernoulli beam theory example, for instance, the outcome variable should be  $\mathbf{U}_{xxxx}$ . This example will be used to show that a set of test trials can be used to determine the best model for a collection of different outcome variables.

Using the methods described in Sec. III D, the operator  $L$  can be identified from a generalized equation  $N$ , which describes a given left-hand side term. A series of SINDy-BVP regressions is used to identify a model operator  $L$  for each outcome variable in a set with increasing differential order ( $\mathbf{U}_x$ ,  $\mathbf{U}_{xx}$ ,  $\mathbf{U}_{xxx}$ ,  $\mathbf{U}_{xxxx}$ , etc.). Each of these operators

is then evaluated with the test trials for the error  $\mathcal{L}_{\text{test}} = 1/T \sum_{j=1, \dots, T} \|L[\mathbf{U}_j] - \mathbf{F}_j\|$  for the  $T$ -test trials.

Figure 6 shows how  $\mathcal{L}_{\text{test}}$  compares between data-driven models generated with different outcome variables. The model that minimizes the test error is the model for  $\mathbf{U}_{xxxx}$ , indicating this is the correct model to use. This approach emphasizes the governing relationship,  $Lu = f$ . In this example, the test data set contains 45 trials.

## V. DISCUSSION

SINDy-BVP successfully extends the data-driven modeling approach of SINDy from dynamical systems to time-

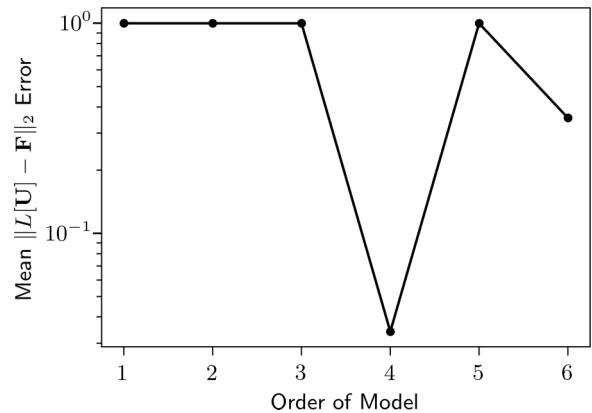


FIG. 6. Determination of correct order equation to use for building the data-driven model. This example is the Euler-Bernoulli beam theory, which should use the left hand side term  $d^4u(x)/dx^4$  to build the correct model. The model that best captures the relationship  $Lu(x) = f(x)$  is determined from the validation error  $\|L[\mathbf{U}_j] - \mathbf{F}_j\|_2$  from a test data set. The fourth-order model ( $d^4u(x)/dx^4$ ) exhibits the lowest error.

invariant, steady-state, spatially-varying BVP systems. The method is used to identify a differential operator,  $L$ , governing forced systems of the form  $L[u_j(x)] = f_j(x)$ , where  $f_j(x)$  is a known forcing function and  $u_j(x)$  is the measured variable, which quantifies the system's response to the forcing. The operator  $L$  can be nonlinear or linear.

Operator identification and parametric coefficient estimation are the two most important tasks. With numerical precision data, SINDy-BVP is effective at identifying the operator and the parametric coefficients within 1% error with relatively little data (see Table I). However, noisy data makes both tasks more difficult. Figure 4 indicates operator identification is challenging with as little as 1% noise (SNR = 100). Parameter estimation can succeed within 15% error with as little as 10–15 trials ( $f_j(x)$ - $u_j(x)$  pairs) in 1% noise [Fig. 5(b)]. However, parameter estimation error does not improve significantly unless much larger sets of training data (over 100 trials) are used. With and without noise in the data, SINDy-BVP often incurs error in both operator identification and parameter estimation near the boundaries of the system. The boundary error is likely a result of the ill-conditioned inverse problem near the boundaries. The problem is ill-conditioned because the boundary conditions are identical in all trials. Specifically, the Dirichlet boundary conditions used in this work stipulate the boundary values  $u_j(a)$  and  $u_j(b)$  are the same in all trials.

There are three primary challenges facing SINDy-BVP. First, model identification relies on having the correct terms in the regression basis function library. If the library is missing one of the basis functions in the operator, the regression will attempt to approximate the missing term using remaining basis functions in the library [10]. The difference between an incomplete learned operator and true operator is beyond the scope of the present work. The second challenge for SINDy-BVP is its susceptibility to noise. Noise can result in both extra added terms and missing excluded terms, both contributing error to the learned model. Noise is often amplified by numerical differentiation methods, so one promising approach to reducing noise is integral-based formulations of SINDy, which were shown to improve noise-handling [36]. Alternatively, improved differentiation methods could be developed or black-box interpolation methods (e.g., neural networks) could be used to build “clean” signals  $\mathbf{U}_j$  from noisy data. Finally, judicious selection of training data is critically important. This is true for any data-driven modeling approach. In the case of SINDy-BVP, the data must exhibit relatively equal contribution to the system behavior from each of the terms in the governing operator for the algorithm to learn the complete and correct operator.

The block matrix regression in Eq. (9) empowers SINDy-BVP to learn parametric coefficients described by nontrivial functions including multimodal sinusoidal and piecewise functions. However, it also imparts minor drawbacks. The discovered parametric coefficients are learned as a vector  $\Xi^{(g)} \in \mathbb{R}^n$ , where each value of the coefficient is mapped to a measured spatial position  $x_k$ . This explicitly ties the resolution of the learned parameters to the measurement grid. If it is reasonable to expect the parametric coefficients to be described by a set of basis functions, it is plausible to use

the learned coefficient vectors as part of a sparse symbolic regression problem akin to SINDy where the coefficients are described by a sparse combination of basis functions. Although this approach could reduce the number of measurement points required to learn the parametric coefficients, it implicitly depends on projecting the functions describing the coefficients into a (known) sparse function basis. If SINDy-BVP was applied to higher-dimensional systems (2D or 3D), the regression (5) grows exponentially with each dimension. For example, a 2D model on a grid with  $n$  samples in each dimension requires a regression (9) with  $n^2$  matrices  $\Theta^{(k)}$  comprising the block diagonal matrix  $\Theta$ . However, a recent work shows promising results on handling higher-dimensional systems with tensor-based SINDy methods [37]. Applying SINDy-BVP to higher-dimensional systems is a subject for future work.

A variety of other adaptations to the SINDy-BVP architecture could also be made that may improve model convergence for operator identification and the accuracy of parametric coefficient estimation. Noise handling may be improved by implementing  $\ell_\infty$  norm constraints on the coefficient vectors  $\Xi^{(g)}$ . Additionally, physics-informed constraints could be imposed on the optimization. For example, in the case of known Sturm-Liouville form operators, constraints could be added to the optimization that directly relate  $p(x)$  to its derivative  $p_x(x)$ . Conservation laws can also be included in the optimization to provide additional constraints, for example, on the energy within the system.

One important consideration is the practical applications of SINDy-BVP to real physical systems. Logistically, SINDy-BVP requires input data  $\{\mathbf{U}, \mathbf{F}\}$ , which are paired matrices of measurements of the system ( $\mathbf{U}$ ) and different forcing functions applied to the system ( $\mathbf{F}$ ). The matrix  $\mathbf{U}$  can be constructed using a grid of sensors measuring the desired state variable ( $u(x)$ ). In order to apply a variety of forcings, a system-specific testing jig would likely need to be constructed in which a forcing could be applied and measured simultaneously. For example, suppose the goal is to measure the thermal properties of a composite bar. A measurement jig could be constructed using an array of evenly spaced thermocouples along the bar, while the forcings could be applied to the bar using thermoelectric heaters.

The SINDy-BVP method proposed in this work successfully enables simultaneous discovery of the governing linear or nonlinear operator  $L$  of a BVP and the parametric coefficients in the operator. It extends the SINDy methodology from dynamical systems to time-invariant BVPs, and is demonstrated to work on systems with piecewise and multimodal sinusoidal parametric coefficients commonly found in heterogeneous materials systems.

The code for this project is available on GitHub at [38].

## ACKNOWLEDGMENTS

This research was primarily supported by the U.S. National Science Foundation (NSF) through the UW Molecular Engineering Materials Center (MEM-C), a Materials Research

Science and Engineering Center (Grant No. DMR-1719797). S.L.B. and J.N.K. acknowledge further funding support from the UW Engineering Data Science Institute, NSF HDR Award

No. 1934292. S.L.B. also acknowledges support from the Army Research Office (Grant No. W911NF-19-1-0045; Program Manager Matthew Munson).

- 
- [1] I. Stakgold, *Boundary Value Problems of Mathematical Physics*, Vol. 29 (Society for Industrial and Applied Mathematics, Philadelphia, 2000).
- [2] I. Stakgold and M. J. Holst, *Green's Functions and Boundary Value Problems*, Vol. 99 (John Wiley & Sons, New York, 2011).
- [3] J.-B. J. Fourier, *Theorie Analytique de la Chaleur* (Chez Firmin Didot, père et fils, Paris, France, 1822).
- [4] C. Aristégui and S. Baste, Optimal recovery of the elasticity tensor of general anisotropic materials from ultrasonic velocity data, *J. Acoust. Soc. Am.* **101**, 813 (1997).
- [5] B. Ganapathysubramanian and N. Zabaras, Modeling diffusion in random heterogeneous media: Data-driven models, stochastic collocation and the variational multiscale method, *J. Comput. Phys.* **226**, 326 (2007).
- [6] M. Kachanov, I. Sevostianov, and B. Shafiro, Explicit cross-property correlations for porous materials with anisotropic microstructures, *J. Mech. Phys. Solids* **49**, 1 (2001).
- [7] S. K. Kim, B. Sung Jung, H. June Kim, and W. Il Lee, Inverse estimation of thermophysical properties for anisotropic composite, *Exp. Therm. Fluid Sci.* **27**, 697 (2003).
- [8] R. Bachrach, M. Sengupta, A. Salama, and P. Miller, Reconstruction of the layer anisotropic elastic parameters and high-resolution fracture characterization from P-wave data: A case study using seismic inversion and Bayesian rock physics parameter estimation, *Geophys. Prospect.* **57**, 253 (2009).
- [9] S. Torquato, *Random Heterogeneous Materials*, Vol. 16 (Springer-Verlag New York, 2002).
- [10] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci. USA* **113**, 3932 (2016).
- [11] T. Kirchdoerfer and M. Ortiz, Data-driven computational mechanics, *Comput. Methods Appl. Mech. Eng.* **304**, 81 (2016).
- [12] T. Kirchdoerfer and M. Ortiz, Data driven computing with noisy material data sets, *Comput. Methods Appl. Mech. Eng.* **326**, 622 (2017).
- [13] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* **3**, 218 (2021).
- [14] D. Z. Huang, K. Xu, C. Farhat, and E. Darve, Learning constitutive relations from indirect observations using deep neural networks, *J. Comput. Phys.* **416**, 109491 (2020).
- [15] L. T. K. Nguyen and M.-A. Keip, A data-driven approach to nonlinear elasticity, *Comput. Struct.* **194**, 97 (2018).
- [16] A. Leygue, M. Coret, J. Réthoré, L. Stainier, and E. Verron, Data-based derivation of material response, *Comput. Methods Appl. Mech. Eng.* **331**, 184 (2018).
- [17] M. Bessa, R. Bostanabad, Z. Liu, A. Hu, D. W. Apley, C. Brinson, W. Chen, and W. K. Liu, A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality, *Comput. Methods Appl. Mech. Eng.* **320**, 633 (2017).
- [18] K. Xu and E. Darve, ADCME: Learning spatially-varying physical fields using deep neural networks, [arXiv:2011.11955](https://arxiv.org/abs/2011.11955).
- [19] P. Nath, Z. Hu, and S. Mahadevan, Sensor placement for calibration of spatially varying model parameters, *J. Comput. Phys.* **343**, 150 (2017).
- [20] P. Koutsourelakis, A multi-resolution, non-parametric, Bayesian framework for identification of spatially-varying model parameters, *J. Comput. Phys.* **228**, 6184 (2009).
- [21] S. L. Brunton and J. N. Kutz, Methods for data-driven multi-scale model discovery for materials, *J. Phys.: Mater.* **2**, 044002 (2019).
- [22] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Data-driven discovery of partial differential equations, *Sci. Adv.* **3**, e1602614 (2017).
- [23] S. Rudy, A. Alla, S. L. Brunton, and J. N. Kutz, Data-driven identification of parametric partial differential equations, *SIAM J. Appl. Dyn. Syst.* **18**, 643 (2019).
- [24] P. Zheng, T. Askham, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin, A unified framework for sparse relaxed regularized regression: Sr3, *IEEE Access* **7**, 1404 (2018).
- [25] L. Zhang and H. Schaeffer, On the convergence of the sindy algorithm, *Multiscale Model. Simul.* **17**, 948 (2019).
- [26] R. Tibshirani, Regression shrinkage and selection via the lasso, *J. R. Stat. Soc. Series B (Methodological)* **58**, 267 (1996).
- [27] K. Champion, P. Zheng, A. Y. Aravkin, S. L. Brunton, and J. N. Kutz, A unified sparse optimization framework to learn parsimonious physics-informed models from data, [arXiv:1906.10612](https://arxiv.org/abs/1906.10612).
- [28] L. Boninsegna, F. Nüske, and C. Clementi, Sparse learning of stochastic dynamical equations, *J. Chem. Phys.* **148**, 241723 (2018).
- [29] S. Zhang and G. Lin, Robust data-driven discovery of governing physical laws with error bars, *Proc. R. Soc.* **474**, 20180305 (2018).
- [30] W. Pan, Y. Yuan, J. Gonçalves, and G. Stan, A sparse Bayesian approach to the identification of nonlinear state-space systems, *IEEE Trans. Autom. Control* **61**, 182 (2016).
- [31] R. K. Niven, A. Mohammad-Djafari, L. Cordier, M. Abel, and M. Quade, Bayesian identification of dynamical systems, *Proceedings* **33**, 33 (2020).
- [32] K. Wu and D. Xiu, Numerical aspects for approximating governing equations using data, *J. Comput. Phys.* **384**, 200 (2019).
- [33] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevResearch.3.023255> for parametric coefficients, forcing functions, and solutions seen in manuscript.
- [34] J. N. Kutz, *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data* (Oxford University Press, Oxford, 2013).

- [35] D. A. Messenger and D. M. Bortz, Weak SINDy: Galerkin-based data-driven model selection, [arXiv:2005.04339](https://arxiv.org/abs/2005.04339).
- [36] P. A. K. Reinbold, D. R. Gurevich, and R. O. Grigoriev, Using noisy or incomplete data to discover models of spatiotemporal dynamics, *Phys. Rev. E* **101**, 010203(R) (2020).
- [37] P. Gelß, S. Klus, J. Eisert, and C. Schütte, Multi-dimensional approximation of nonlinear dynamical systems, *J. Comput. Nonlinear Dyn.* **14**, 061006 (2019).
- [38] <https://github.com/sheadan/SINDy-BVP>.