Data-driven discovery of partial differential equation models with latent variables

Patrick A. K. Reinbold and Roman O. Grigoriev

Phys. Rev. E **100**, 022219 — Published 21 August 2019

DOI: 10.1103/PhysRevE.100.022219

Data-driven discovery of partial differential equation models with latent variables

Patrick A.K. Reinbold and Roman O. Grigoriev
School of Physics, Georgia Institute of Technology, Atlanta, Georgia 30332-0430, USA

(Dated: August 5, 2019)

In spatially extended systems, it is common to find latent variables that are hard, or even impossible, to measure with acceptable precision, but are crucially important for the proper description of the dynamics. This substantially complicates construction of an accurate model for such systems using data-driven approaches. The present paper illustrates how physical constraints can be employed to overcome this limitation using the example of a weakly turbulent quasi-two-dimensional Kolmogorov flow driven by a steady Lorenz force with an unknown spatial profile. Specifically, the terms involving latent variables in the partial differential equations governing the dynamics can be eliminated at the expense of raising the order of that equation. We show that local polynomial interpolation combined with sparse regression can handle data on spatiotemporal grids that are representative of typical experimental measurement techniques such as particle image velocimetry. However, we also find that the reconstructed model is sensitive to measurement noise and trace this sensitivity to the presence of high order spatial and/or temporal derivatives.

I. INTRODUCTION

Due to advances in data acquisition, storage, and computational power, data-driven discovery of mathematical models of physical systems, which relies primarily on the empirical observations, has emerged as a viable alternative to more traditional approaches based on, e.g., first-principles derivation. While methods for constructing linear models of dynamical systems are very well established [1], the progress in model discovery for nonlinear processes is more recent, with the earliest efforts focusing on nonlinear ordinary differential or difference equation models of low-dimensional dynamics [2–6]. Some progress has also been made in model discovery for spatially-distributed systems described by nonlinear partial differential equations (PDEs).

The earliest study by Vallette *et al.* [7] showed that it is possible to estimate the coefficients in a PDE of a known form (modified complex Ginzburg-Landau equation) from data available on a spatiotemporal grid. Their algorithm relied on a least squares solution of a system of linear equations constructed by evaluating individual terms in the equation at multiple locations on the grid. Bär *et al.* [8] extended this approach by constructing candidate PDE models from a broad set of polynomial combinations of the observable and its derivatives. While the allowed terms were constrained by symmetry considerations, the study made no attempt to eliminate any terms to obtain a parsimonious description based on the data itself. It was found that using finite differences for evaluating partial derivatives based on noisy data resulted in significant errors and even applying a Savitzky-Golay filter [9] failed to noticeably improve the results.

The earliest successful attempt to construct a parsimonious PDE model is due to Xu and Khanmohamadi [10, 11], who introduced an iterative procedure to eliminate the terms that have a small effect on the residual. These were also the first studies to allow spatially varying coefficients by including in the candidate model terms polynomial in the observable (dependent variable) and its partial derivatives as well as the independent variable(s). The studies have also shown that, by evaluating terms using spectral derivatives (Fourier or Chebyshev, depending on boundary conditions) instead of finite differences, the coefficients of a PDE model can be reconstructed with high accuracy from noiseless data even in the presence of high-order derivatives. Similar sparse symbolic regression approaches have been proposed a decade later by Rudy *et al.* [12] for models with constant coefficients and by Li *et al.* [13] for models with nonconstant coefficients.

We should also mention the work of Raissi *et al.* [14] which has demonstrated that unknown coefficients in a PDE model with known structure can be estimated from spatially sparse data using statistical techniques. In comparison, approaches based on sparse regression [10–13] require data on a fairly dense spatiotemporal grid. The statistical approach is based on treating a temporally discretized, linearized version of the model as a Gaussian processes and becomes both cumbersome and numerically expensive for models that involve more than a few terms or variables.

The above approaches assume that all state (dependent) variables that appear in the model can be either directly measured or computed directly from the measured data. In many cases of practical interest, one might be interested in determining the model equations using only some of those variables. For instance, the primitive variable description of fluid flows relies on two physical fields: velocity and pressure, only the first of which can typically be measured in experiment with meaningful accuracy. The presence of latent variables such as pressure makes data-driven model discovery substantially more complicated or incomplete, since existing approaches based on sparse regression [6, 10], crucially rely on direct measurements of every state variable which appears in the model. In particular, using velocity measurements alone only allows reconstruction of the vorticity equation [12] which describes the evolution of the curl of the velocity, but not its individual components. The statistical approach [15] allows the coefficients in the 2D Navier-Stokes equation to be determined without any knowledge of the pressure [14], but assumes the form of the model is known.

Another common problem in data-driven model discovery is sensitivity to noise in the data [8, 10]. It is especially acute for spatially distributed systems due to the difficulty in accurately estimating spatial derivatives using noisy data. As an example, adding just 1% noise to the data causes errors in the model parameters of order 10% for the nonlinear Schrödinger, KdV, and the vorticity equations, 50% for the Kuramoto-Sivashinsky equation, and introduces spurious terms in the $\lambda - \omega$ model [12]. It is not currently understood on a quantitative level what the impact of noise is on the accuracy of the model reconstruction, however. Neither is it clear how the accuracy of the model reconstruction based on gridded noisy data can be quantified in the absence of some sort of a reference.

This work uses a representative example of a fluid flow to address several of the open questions, mainly (1) how can we get around the lack of direct measurements of latent variables and (2) how can we quantify the accuracy of the resulting model when the measurements of observable variables are gridded and noisy? The structure of the paper is as follows. Section II describes the physical problem the model of which we are trying to construct and the relevant physical constraints. Section III describes our sparse regression approach. Section IV discusses the use of polynomial approximations for estimating spatial and temporal derivatives. The results are presented in Sect. V and our conclusions in Sect. VI.

II. PROBLEM STATEMENT

We will focus on the Lorenz-force-driven flow in a thin electrolyte layer supported by a stationary bottom plate [16, 17] as it provides an excellent illustration of the challenges in data-driven discovery of a model for a spatially distributed system with latent variables. The basic physics and symmetry of the problem imposes a number of constraints on the form of the model and the choice of the fundamental variables. Being a fluid flow, it is described by two fields, velocity $\mathbf{u}$ and pressure p, so we would expect the dynamics of the fluid flow to be described by evolution equations of the general form

$$\begin{aligned}\partial_t \mathbf{u} &= \mathbf{N}_u(\mathbf{u}, p),\\ \partial_t p &= N_p(\mathbf{u}, p),\end{aligned} \tag{1}$$

where $\mathbf{N}_u$ and N_p are some (generally nonlinear) differential operators.

The form of these operators can be constrained by both the physics and the symmetry of the problem; we will start with the latter. In particular, in order to preserve the rotational symmetry, $\mathbf{N}_u$ has to be a vector, and so it can be constructed as a linear superposition of terms each of which is a vector. Since the fluid layer is thin, the vertical component of the velocity is small compared to the horizontal component and we can consider $\mathbf{u}$ to be two-dimensional (we can think of $\mathbf{u}$ as describing the flow at the free surface of the electrolyte). Furthermore, again due to the small thickness of the fluid layer, both $\mathbf{u}$ and p can be considered functions of horizontal coordinates x and y and time t, but not the vertical coordinate z.

There are several ways to construct a vector out of $\mathbf{u}$, p, the gradient operator ∇, and the external forcing field $\mathbf{f}$ (assumed to be time-independent). The gravitational acceleration $\mathbf{g}$, the only other vector quantity in the problem, cannot be included in the two-dimensional model, since the latter does not explicitly include the vertical direction. Using one vector object, we can construct three vector fields that are linear in $\mathbf{u}$, p, and $\mathbf{f}$: ∇p, $\mathbf{u}$, and $\mathbf{f}$. More complicated vector fields can be constructed using powers of ∇ and/or nonlinear functions of p, $\mathbf{u}$, and $\mathbf{f}$. We will only consider terms that are linear in p and $\mathbf{f}$, since ∇p and $\mathbf{f}$ both describe the (volumetric) force density and they are linearly related to the time rate of change of the momentum density $\rho\partial_t\mathbf{u}$ according to Newton's 2nd law.

More vector fields can be constructed using several copies of $\mathbf{u}$ and ∇. Keeping terms up to third order in $\mathbf{u}$ and second order in ∇, we obtain the following evolution equation for the velocity field

$$\begin{aligned}\partial_t \mathbf{u} = {}& c_1(\mathbf{u}\cdot\nabla)\mathbf{u} + c_2\nabla^2\mathbf{u} + c_3\mathbf{u} + c_4(\nabla\cdot\mathbf{u})\mathbf{u}\\ &+ c_5(\nabla\cdot\mathbf{u})^2\mathbf{u} + c_6(\nabla\times\mathbf{u})^2\mathbf{u} + c_7\mathbf{u}^2\mathbf{u}\\ &+ c_8\nabla p + c_9\mathbf{f}.\end{aligned} \tag{2}$$

The evolution equation for the pressure can be constructed in a similar manner, with N_p that should be a scalar. The pressure should be a function of the velocity only, so keeping the leading order (in ∇ and $\mathbf{u}$) term, we will find

$$\partial_t p = -\kappa\nabla\cdot\mathbf{u}, \tag{3}$$

where κ is another unknown parameter. Using the scaling freedom in defining the latent field p explicit in the equations (2) and (3), without the loss of generality we can set $|c_8| = 1$. Similarly, we can set $c_9 = 1$, which amounts to choosing a particular scale for the (unknown) forcing. The remaining constant (to preserve the translational symmetry in space and time) parameters c_1 through c_7 (and possibly κ) need to be determined from data using sparse regression.

The combination of symmetry and basic physics we used constrains the form of the evolution equations rather significantly, yielding a set of evolution equations with rather few superfluous terms. Indeed, under certain assumptions (e.g., lubrication approximation), a two-dimensional model of the form

$$\partial_t \mathbf{u} = -\beta(\mathbf{u}\cdot\nabla)\mathbf{u} + \nu\nabla^2\mathbf{u} - \alpha\mathbf{u} + \rho^{-1}(\mathbf{f} - \nabla p), \tag{4}$$

$$\partial_t p = -\kappa\nabla\cdot\mathbf{u}, \tag{5}$$

which is a special case of the general model (2)-(3), can be derived analytically for this flow by depth-averaging the three-dimensional Navier-Stokes equation [16, 18]. Here α, ν, β, ρ, and κ are constants representing, respectively, the vertical momentum transport, the horizontal momentum transport, the attenuation of inertia due to vertical velocity stratification, the density of the fluid, and the scale of the hydrostatic pressure.

Comparison of numerical simulations and experimental observations over a range of Reynolds numbers Re suggests that the model (4)-(5) with constant parameters α, ν, and β is qualitatively accurate [17], but there is a systematic discrepancy that can be mostly accounted for by making these parameters weakly dependent on the Reynolds number Re. The assumptions made in deriving the model mainly affect the vertical momentum transport represented by the Rayleigh friction term $-\alpha\mathbf{u}$. This momentum transport increases with Re due to the advection, which can be accounted for by making the scalar coefficient α velocity-dependent. This is what the last few terms in the general model (2) represent. Specifically,

$$\begin{aligned}\alpha = &-c_3 - c_4\nabla\cdot\mathbf{u} - c_5(\nabla\cdot\mathbf{u})^2 \\ &- c_6(\nabla\times\mathbf{u})^2 - c_7\mathbf{u}^2\end{aligned} \tag{6}$$

can be thought of as a second-order (in $\mathbf{u}$ and ∇) model of the Rayleigh friction coefficient.

III. SPARSE REGRESSION

The data characterizing both components of the velocity field $\mathbf{u}$ can be obtained, for instance, using particle image velocimetry [16, 17] and is assumed to be on a uniform grid (i, j, k), where i, j, and k correspond to the x, y, and t directions, respectively. However, unlike standard sparse regression problems where all the variables are directly observable, in our problem neither the pressure p nor the forcing $\mathbf{f}$ are, so both fields have to be either determined independently or eliminated. In principle, for a fully resolved incompressible ($\kappa \to \infty$) flow field, if $\mathbf{f}$ were known, p could be obtained in a standard way by applying a divergence to (2), which yields a pressure Poisson equation. Typical experimental data however have a resolution that is too poor (and noise level that is too high) to make it possible to compute pressure in this manner.

In the following, we will focus just on the evolution equation for the velocity field; the evolution equation for the pressure is very simple and the coefficient κ can be eliminated altogether by rescaling c_8 and/or p. The terms involving both latent fields can be eliminated from (2) by applying an operator $\hat{P} = \hat{S}\hat{C}\hat{T}$ composed of three operations: $\hat{C} = \hat{z}\cdot\nabla\times$ removes the dependence on ∇p which is curl-free, $\hat{T} = \partial_t$ removes the dependence on $\mathbf{f}$ which is constant, and the sparsification operator $\hat{S}$ subsamples the original data in a random fashion. The corresponding discretization of the resulting PDE (which is second order in time, third order in space, and fourth order overall) has the form

$$\mathbf{q}_0 = Q\mathbf{c}, \tag{7}$$

where $Q = [\mathbf{q}_1 \ \ldots \ \mathbf{q}_7]$, $\mathbf{c} = [c_1 \ \cdots \ c_7]^T$ is a vector composed of scalar coefficients to be determined, and the columns

$$\begin{aligned}
\mathbf{q}_0 &= \hat{P}(\partial_t\mathbf{u} - \mathbf{f} + \nabla p), & \mathbf{q}_1 &= \hat{P}(\mathbf{u}\cdot\nabla)\mathbf{u}, \\
\mathbf{q}_2 &= \hat{P}\nabla^2\mathbf{u}, & \mathbf{q}_3 &= \hat{P}\mathbf{u}, \\
\mathbf{q}_4 &= \hat{P}(\nabla\cdot\mathbf{u})\mathbf{u}, & \mathbf{q}_5 &= \hat{P}(\nabla\cdot\mathbf{u})^2\mathbf{u}, \\
\mathbf{q}_6 &= \hat{P}(\nabla\times\mathbf{u})^2\mathbf{u}, & \mathbf{q}_7 &= \hat{P}\mathbf{u}^2\mathbf{u}
\end{aligned} \tag{8}$$

correspond to different terms in (2). Note that $\mathbf{q}_0 = \hat{P}\partial_t\mathbf{u}$, so none of the terms $\mathbf{q}_i$ in fact depend on either p or $\mathbf{f}$. For the number K of points in the sample exceeding the number of unknown coefficients, this yields an overdetermined system (7) of linear equations for $\mathbf{c}$, where the "library" Q and the "target" $\mathbf{q}_0$ can be evaluated using any algorithm sufficiently robust with respect to noise and sparsity of the data. The particular procedure used in the present paper is described in the next section.

We performed sparse regression using a variation of the iterative algorithm for sparse identification of nonlinear dynamical systems (SINDy) [6], which involves computing the solution $\mathbf{c}$ that minimizes the residual

$$\eta = \|\mathbf{q}_0 - Q\mathbf{c}\|_1 \tag{9}$$

of the linear system (7), followed by a thresholding procedure to remove dynamically irrelevant terms. Note that the library terms $\mathbf{q}_i$ themselves can differ by many orders of magnitude (an example of this is presented below). Since it is the product, $c_i\mathbf{q}_i$, that determines a given term's role in the model, we employ a slightly modified thresholding procedure. We compare the norms of the products $c_i\mathbf{q}_i$ to the residual η: the columns of Q for which $\|c_i\mathbf{q}_i\|_1 < \gamma\eta$ are removed, and the process is repeated until all remaining terms are above the threshold. Here γ is a constant that can be above or below, but is close to, unity. This approach requires no *a priori* knowledge of the system, in contrast to previous studies [6, 13] that have defined various arbitrary thresholds as stopping conditions for the iterative procedure.

IV. POLYNOMIAL INTERPOLATION

All of the library terms involve spatial and/or temporal derivatives of the velocity field. Using total variation regularization of the data [19] to reduce the influence of noise is both prohibitively expensive in higher dimensions and unnecessary given the sparse nature of the system (7). Additionally, although spectral derivative estimates perform better in the noiseless case, their high frequency components are corrupted by noise, diminishing their accuracy severely [10]. Therefore, to accomplish the task of smoothing noisy data and taking numerical derivatives concurrently, a higher-dimensional generalization of the polynomial interpolation in [9] was used instead. (We also investigated computation of derivatives using discrete Fourier transform, and found the results to be comparable). At each point chosen by the sparsification operator $\hat{S}$, the velocity fields were approximated by a polynomial in x, y, and t fitted to discrete data on a rectangular domain Ω of size $2H_x \times 2H_y \times 2H_t$ centered at a grid point (x_i, y_j, t_k).

In particular, the x-component of the velocity $u(x, y, t)$ near (x_i, y_j, t_k) was approximated as

$$\tilde{u}(x, y, t) = \sum_{l=0}^{L} \sum_{m=0}^{M} \sum_{n=0}^{N} U_{ijk}^{lmn} \bar{x}^l \bar{y}^m \bar{t}^n, \tag{10}$$

where the overbar denotes the shifted and rescaled coordinates in which the domain Ω becomes a cube $\Omega' = [-1, 1] \times [-1, 1] \times [-1, 1]$, e.g.,

$$\bar{x} = \frac{x - x_i}{H_x}, \tag{11}$$

etc. The order of the polynomial in each direction should be at least as large as the order of the highest derivative appearing in the model equation (2) after the operator $\hat{C}\hat{T}$ is applied, but ultimately is a tunable parameter, with the specific choice to be discussed in more detail in the next section.

The coefficients U_{ijk}^{lmn} were found by minimizing the cost function

$$F = \sum_{(i,j,k)\in\Omega} w_{ijk}(u_{ijk} - \tilde{u}(x_i, y_j, t_k))^2, \tag{12}$$

where w_{ijk} is a weighting function. This is a standard least squares problem whose solution is given by setting $\partial F/\partial U_{ijk}^{qrs} = 0$. This yields a system of $(L + 1)(M + 1)(N + 1)$ linear equations

$$\langle w\, u\, \bar{x}^q \bar{y}^r \bar{t}^s \rangle_\Omega = \sum_{l,m,n}^{L,M,N} U_{ijk}^{lmn} \langle w\, \bar{x}^{l+q} \bar{y}^{m+r} \bar{t}^{n+s} \rangle_\Omega, \tag{13}$$

where $\langle \cdot \rangle_\Omega$ denotes the average over the spatiotemporal sub-domains for which the local fits are defined. The weighting function w_{ijk} was used to bias the accuracy of the approximation toward the central point of the domain Ω (where all of the derivatives are evaluated) and is defined as a Gaussian

$$w_{ijk} = \exp\left(-\frac{\bar{x}^2 + \bar{y}^2 + \bar{t}^2}{\lambda^2}\right), \tag{14}$$

with the width λ being another tunable parameter of the model (we set $\lambda = 0.5$). The same procedure was used to determine the coefficients V_{ijk}^{lmn} for the y-component of the velocity $v(x, y, t)$.

After the polynomial coefficients have been determined, the row of the library Q and the target $\mathbf{q}_0$ corresponding to the point (x_i, y_j, t_k) can be constructed by evaluating the respective derivatives of u and v at $(\bar{x}, \bar{y}, \bar{t}) = (0, 0, 0)$ using (10). For instance,

$$q_2^{ijk} = 6V_{ijk}^{301} + 2V_{ijk}^{121} - 2U_{ijk}^{211} - 6U_{ijk}^{031}. \tag{15}$$

The process was repeated for each point defined by $\hat{S}$ in order to completely evaluate the library and the target. $\hat{S}$ was defined by randomly selecting the points on the entire 3D grid representing the spatially and temporally discretized trajectory. Throughout the paper, $K = 250$ points were used to construct the library; neither the mean nor the standard deviation of the coefficients c_i were found to exhibit meaningful variation for a larger number of points.

V. RESULTS

Surrogate data used for testing the sparse regression procedure was generated using the model (2)-(3) with the parameters $c_1 = -0.826$, $c_2 = 0.0487$, $c_3 = -0.157$, $c_4 = 0.164$, $c_5 = c_6 = c_7 = 0$, $c_8 = -1$, and $\kappa = 2015$. This set of parameters describes a nearly incompressible flow found in the experiment described in Ref. [17], which features a forcing field with a sinusoidal profile in the y direction with period $2\chi = 2$ and amplitude equal to 1.0649 in nondimensional units. The solution describing a weakly turbulent flow was obtained using a numerical integration scheme based on operator splitting as described in Ref. [18]. The linear terms were evolved in time implicitly, while the nonlinear terms were handled via a 2^{nd} order Adams-Bashforth scheme. The solution was integrated on a computational grid with $\Delta x_c = \Delta y_c = 0.025$, and $\Delta t_c \approx 0.02$. Gaussian random noise with variance σ was added to both components of the flow velocity $\mathbf{u}$. For reference, the maximal flow velocity is $O(1)$ in nondimensional units.

In order for the algorithm to produce meaningful results, its various tunable parameters must be properly set. The noiseless case exhibits the least amount of sensitivity to variation of fitting parameters; the only restriction is that the polynomial orders L, M, and N be high enough to capture the variation in the data over Ω. While higher order interpolation allows better approximation of the data, it is also more sensitive to noise. To mitigate the influence of noise, a larger number of measurements can be used. There are two ways to achieve this: by increasing the size of the sampling domain Ω or by using a finer grid on which data are measured. The largest size of Ω is effectively limited by the characteristic length and time scales for the problem. In the present problem, the natural length scale is defined by χ. Consequently, we will set $H_x = H_y = \chi/2$, such that the width of Ω in both spatial dimensions is equal to $\chi = 1$. There is no natural time scale, so we will choose one based on the autocorrelation time $\tau \approx 9.9$. In the following we set $H_t \approx 0.85\tau$ which is an optimal choice for $\sigma = 10^{-3}$ and $M = L$. Furthermore we use the finest

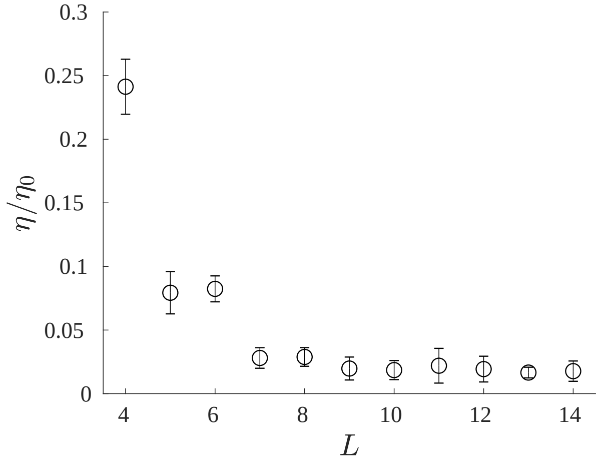

FIG. 1. Residual as a function of polynomial order for $N = 10$ and $\sigma = 0$. Here, error bars denote standard deviation and symbols denote mean values.

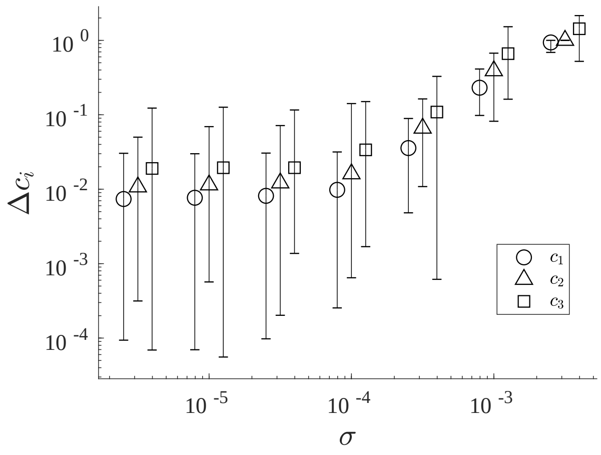

FIG. 2. Parameter error as a function of noise amplitude for $L = N = 10$. The error for c_4 is not shown because SINDy discards the corresponding term. Here and below markers are shifted left or right to avoid overlap. Error bars indicate the full range of data, and markers indicate mean values.

grid available to us in space, i.e., $\Delta x = \Delta x_c$, while in time we use $\Delta t = 25\Delta t_c$. With this choice, Ω corresponds to a $40 \times 40 \times 34$ block of data with dimensions that are roughly comparable in the spatial and temporal directions. At higher grid sizes evaluating the averages in (13) becomes computationally expensive.

To investigate how the choice of polynomial order affects the accuracy of the fit and hence the accuracy with which various partial derivatives of $\mathbf{u}$ are evaluated, we computed the residual (9). The dependence of η (normalized by the magnitude of the target $\eta_0 = \|\mathbf{q}_0\|_1$) is shown in Fig. 1. Here and below, the averages and standard deviation are computed using an ensemble of 40 different realizations of the sampling operator $\hat{S}$. Note that η generally does not vanish even for the noiseless perfect model of the problem due to discretization errors of the numerical solution. Also note that the magnitude of η describes the accuracy with which equation (7) is satisfied, not the accuracy of the numerical solution to the model (2)-(3). As expected, η decreases for low L, but beyond some threshold (in this case $L = 7$), increasing the polynomial order has little effect on the residual. In particular, $L = N = 10$ results in both a low value of the residual and a small error in parameter estimation in the noiseless case, as we will see below.

To determine how the results depend on the amplitude σ of measurement noise, we performed sparse regression and compared the coefficients $\tilde{c}_i$ produced by SINDy with the reference values c_i used to generate the surrogate data for $\kappa = 2015$. Of the four nonzero parameters used in generating the data, three (c_1, c_2, and c_3) were correctly identified as being nonzero and estimated with a small relative error

$$\Delta c_i = \left|\frac{c_i - \tilde{c}_i}{c_i}\right| \tag{16}$$

(of order one percent) for sufficiently small σ, as illustrated by Fig. 2. However, the coefficient c_4 was incorrectly set to zero by the algorithm for all σ. Furthermore, the accuracy in estimating all of the remaining parameters decreased sharply for $\sigma \gtrsim 10^{-4}$.

The failure of sparse regression to correctly identify the value of c_4 can be understood qualitatively by recalling that it is the product $c_4\mathbf{q}_4$ whose magnitude is used to determine whether the corresponding term should be retained or discarded. For our choice of parameters, $\|c_4\mathbf{q}_4\|_1 \lesssim \eta$, suggesting that the magnitude of this term is as small or smaller than the accuracy to which the governing equation can be satisfied. As mentioned previously, the parameter set used here corresponds to a nearly incompressible flow where $\nabla \cdot \mathbf{u}$ is nonzero but very small. Indeed, from (2) and (3) we find $\mathbf{q}_4 \sim \nabla \cdot \mathbf{u} \sim \kappa^{-1} \approx 5 \times 10^{-4}$. By eliminating the term $c_4(\nabla \cdot \mathbf{u})\mathbf{u}$ representing the effect of compressibility [18] from the model (2), sparse regression effectively recognized this fact. The accuracy of the incompressible model has been established previously by comparing experimental and numerical results [20, 21].

To verify that this term was indeed eliminated due to the large value of κ (and not some shortcoming of the method), we repeated the analysis, setting $\kappa = 1$ to amplify the compressibility effects. In this case the term $c_4(\nabla \cdot \mathbf{u})\mathbf{u}$ is retained in the model and the value of the parameter c_4 is determined correctly for sufficiently small σ. This is a good example illustrating when sparse regression fails to identify terms that are generally required by the physics of the problem but may be neglected under some conditions.

To make the argument more quantitative, let us intro-

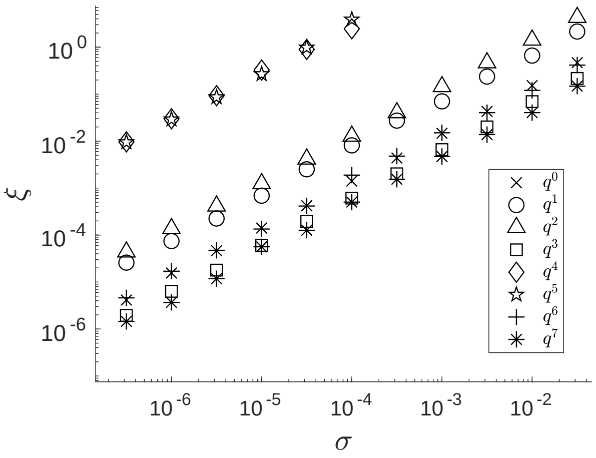

FIG. 3. Relative accuracy of different library terms as a function of noise amplitude for $L = N = 10$.

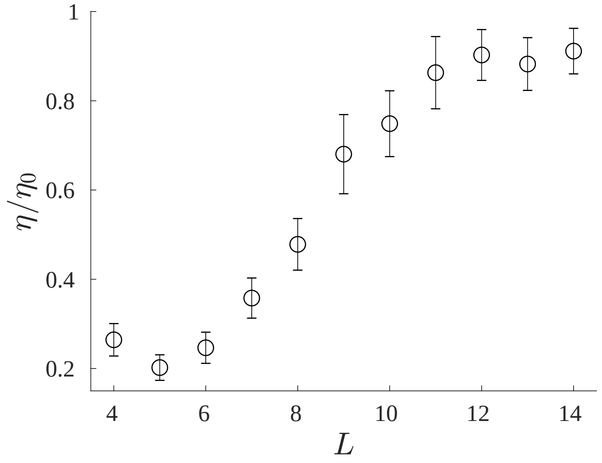

FIG. 4. Residual as a function of polynomial order for $N = 10$ and $\sigma = 10^{-3}$. Here, error bars denote standard deviation and symbols denote mean values.

duce the measure

$$R_i = \frac{\|c_i \mathbf{q}_i\|_1}{\eta}, \tag{17}$$

of the magnitude of a particular term in the linear equation (7) relative to the corresponding residual (9). sparse regression can correctly identify a particular term in the model only if the corresponding $R_i > 1$; furthermore, we can expect the accuracy of parameter reconstruction to decrease as R_i approaches unity. For our choice of fitting parameters, R_4 is below unity for $\kappa = 2015$ and above unity ($R_4 \approx 20$) for $\kappa = 1$. Correspondingly, the terms with lower R_i exhibit the worst fitting accuracy; this explains the larger relative error in c_3 compared with c_1 and c_2 in the $\kappa = 2015$ case (cf. Fig. 2), since although $R_3 \approx 30$, it is much smaller than R_1 and R_2, which both have $R_i > 100$.

Figure 2 also shows that, for the choice of fitting parameters optimized for noiseless data, the accuracy of sparse regression sharply decreases for $\sigma > O(10^{-4})$. To understand why this happens, let us define the relative accuracy with which a particular library term is evaluated over the entire sample

$$\xi_i(\sigma) = \frac{\|\mathbf{q}_i(0) - \mathbf{q}_i(\sigma)\|_\infty}{\|\mathbf{q}_i(0)\|_\infty}. \tag{18}$$

The effect of noise on the accuracy of all the library terms is shown in Fig. 3. Note that the lowest accuracy (highest ξ_i) corresponds to the terms $\mathbf{q}_4$ and $\mathbf{q}_5$ which are linear and quadratic, respectively, in $\nabla \cdot \mathbf{u}$, which is very small. These terms are the most susceptible to corruption by noise but, for large values of κ, they are eliminated by sparse regression anyway. As might be expected, in the absence of these two terms, the term $\mathbf{q}_2$, which involves the highest order derivative (third order in space and first in time), is the least accurate in the presence of noise. This helps explain the difficulties sparse regression has with identifying high order derivatives in all PDE models in the presence of noise. For instance, in a previous study [12], the coefficient of the fourth order derivative term in the Kuramoto-Sivashinsky was determined with a 52% error in the presence of 1% noise. In our case, the terms $\mathbf{q}_3$ and $\mathbf{q}_7$ which involve the lowest order derivative (first in space and time), have the smallest error, suggesting that the order of the derivative is one of the main factors which determine the accuracy of regression in the presence of noise.

The effect of noise can be offset, to some extent by a different choice of parameters. In particular, the order of the polynomial interpolation can be reduced to decrease noise sensitivity. The dependence of the residual η on L is shown in Fig. 4 for noise amplitude $\sigma = 10^{-3}$ at which our previous choice of parameters lead to unacceptably large errors. At this value of σ, we find a minimum around $L = 5$ (with a significant increase in η compared to the noiseless case), which represents a balance between the accuracy of the interpolation in capturing the spatial variation of the data at higher L and the noise sensitivity at lower L. In fact, we found that setting $L = M = N = 6$ is the best choice for minimizing both the residual and the error in parameter estimation.

Using the fitting parameters optimized for higher noise levels, sparse regression identifies the correct model (aside from the negligible term $\mathbf{q}_4$) with all the model parameters estimated to within ∼10% for 0.3% noise and to within ∼30% for 1% noise, as illustrated by Fig. 5. This is comparable to the accuracy achieved for the Kuramoto-Sivashinsky equation [12], which also includes a fourth order derivative. The trade-off of this choice of fitting parameters is the decrease in the accuracy of model parameter estimation at lower noise levels. Furthermore, R_3 becomes close to unity, so sparse regression yields false negatives for a noticeable fraction of the tri-

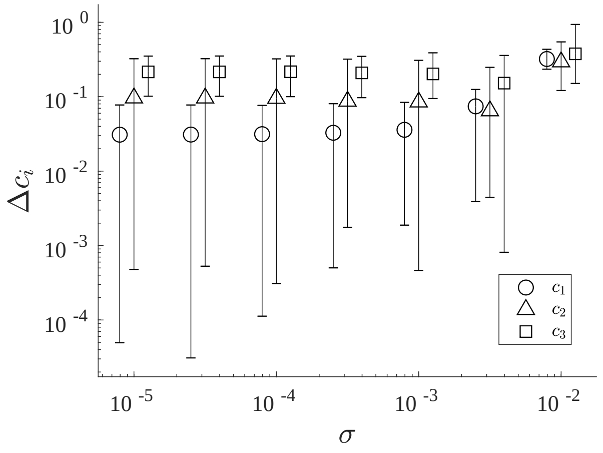

FIG. 5. Parameter error as a function of the noise amplitude for $L = N = 6$. The error for c_4 is not shown because SINDy discards the corresponding term. Error bars indicate the full range of data, and markers indicate mean values.

als (the data shown in Fig. 5 was calculated after discarding the results for which SINDy eliminated the term $\mathbf{q}_3$). In comparison, false negatives did not appear for $L = N = 10$ until fairly high levels of noise. These false negatives occur for lower L and/or N because the magnitude of the residual is determined by the error in the term(s) most affected by the insufficiently accurate approximation (here, the term $\mathbf{q}_2$ which involves the highest order derivative). For the lower L, the variation in the data is not fully resolved, meaning that R_3 is pushed closer to unity (and hence c_3 can be estimated with less accuracy).

VI. CONCLUSIONS

This paper presents an approach that allows sparse regression to be extended for data-driven discovery of PDE-based models which involve unobservable and/or unknown (latent) variables. Our approach relies on two key ideas: (1) using spatial symmetry as well as other physical constraints to select the terms that can appear in the model and (2) applying a differential operator designed to remove terms which involve the latent variables. We illustrated these ideas by using sparse regression to construct a two-dimensional model for a weakly compressible Kolmogorov-like flow in a thin electrolyte layer driven by a steady Lorentz force with an unknown spatial profile. In this particular case, two latent variables – the forcing field which is a vector and the pressure which is a scalar – have been eliminated by applying, respectively, a temporal derivative and a curl to a nonlinear model with nine different terms allowed by symmetry.

It should be noted that this approach does have limitations. The terms in the model that involve latent variables have been *eliminated* rather than *reconstructed.* This is why supplementing the data-driven approach with additional physical constraints, which play a "constructive" role, is essential. It may not be possible to find a differential operator than eliminates the latent variables (although one could also use an integral operator). Also, our approach might not work for models that involve "cross-terms" which include both observable and latent variables.

While previous studies have demonstrated the power of sparse regression for data-driven model discovery, they left a number of questions unanswered. In particular, how should one choose the threshold that determines which terms in the model are relevant? Since those studies mainly focused on reconstructing well-known models, the threshold could be chosen in an ad hoc fashion such that the *a priori* known model was recovered. In case the form of the model is not known *a priori*, the proper choice is less clear, since sparse regression will recover different models for different choices of the threshold. We have shown that a self-consistent choice should be based on the residual η of the linear system (7): in most cases sparse regression can be considered to have successfully reconstructed the model once the corresponding norm of every remaining term $c_i\mathbf{q}_i$ is larger than the residual. However, while the residual is the proper metric for determining the relevance of different terms, fine-tuning the threshold does have an effect on the reconstructed model, mainly affecting the terms with $R_i = O(1)$. For γ somewhat smaller than unity, sparse regression can produce false positives, e.g., the term $c_4(\nabla \cdot \mathbf{u})\mathbf{u}$ is retained even in the essentially incompressible case when $\nabla \cdot \mathbf{u}$ is very small and R_4 is just below unity. However, the value (and sign) of the coefficient c_4 is found to vary drastically for different realizations of the sampling operator $\hat{S}$, suggesting that this term is not dynamically relevant. For γ somewhat larger than unity, this term is correctly removed by the thresholding procedure for sufficiently low noise. However, this choice also causes the term $c_3\mathbf{u}$ to be removed for some realizations of $\hat{S}$ at higher noise amplitudes, when R_3 becomes comparable to unity as well. In case a relevant term (such as $c_3\mathbf{u}$) is removed by the thresholding procedure, the residual increases noticeably, which allows detection of false negatives. Manually including the term $c_3\mathbf{u}$ in such cases decreases the residual by about 10%, indicating that it is dynamically relevant despite being smaller than the residual. Fine-tuning γ based on these metrics can allow more robust results in border-line cases.

Our study has also highlighted the major weakness of all spatially local approaches to sparse regression. Regardless of whether one uses a polynomial interpolation of the data, total variation regularization, or some other similar approach to construct the linear system whose solution determines the model, the procedure is inherently sensitive to noise, especially when higher order derivatives are involved. This difficulty is well-illustrated by both the failure of previous studies to reconstruct with acceptable accuracy the fourth-order Kuramoto-

Sivashinsky equation in the presence of as little as 1% noise and a similar loss in the accuracy for the model considered here, whose latent-variable-free form also involves fourth order derivatives. In both cases, as the noise amplitude increases, the term involving the highest order derivative becomes the largest contributor to the residual at which point its coefficient cannot be reliably determined anymore. Since experiments commonly involve substantially higher amounts of noise, a more robust alternative to such local methods is needed.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. CMMI-1725587.

[1] Karl Johan Åström and Peter Eykhoff. System identificationa survey. *Automatica*, 7(2):123–162, 1971.

[2] James P Crutchfield and Bruce S McNamara. Equation of motion from a data series. *Complex systems*, 1(417-452):121, 1987.

[3] Gregory L Baker, Jerry P Gollub, and James A Blackburn. Inverting chaos: Extracting system parameters from experimental data. *Chaos*, 6(4):528–533, 1996.

[4] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

[5] I-Chun Chou and Eberhard O Voit. Recent developments in parameter estimation and structure identification of biochemical and genomic systems. *Mathematical biosciences*, 219(2):57–83, 2009.

[6] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113:3932–3937, 2016.

[7] D. P. Vallette, G. Jacobs, and J. P. Gollub. Oscillations and spatiotemporal chaos of one-dimensional fluid fronts. *Physical Review E*, 55(4):4274, 1997.

[8] Markus Bär, Rainer Hegger, and Holger Kantz. Fitting partial differential equations to space-time dynamics. *Physical Review E*, 59(1):337, 1999.

[9] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[10] Daolin Xu and Omid Khanmohamadi. Spatiotemporal system reconstruction using fourier spectral operators and structure selection techniques. *Chaos*, 18(4):043122, 2008.

[11] Omid Khanmohamadi and Daolin Xu. Spatiotemporal system identification on nonperiodic domains using chebyshev spectral operators and system reduction algorithms. *Chaos*, 19(3):033117, 2009.

[12] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[13] Xiuting Li, Liang Li, Zuogong Yue, Xiaoquan Tang, Henning U Voss, Jürgen Kurths, and Ye Yuan. Sparse learning of partial differential equations with structured dictionary matrix. *Chaos*, 29(4):043130, 2019.

[14] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Numerical gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 40(1):A172–A198, 2018.

[15] Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.

[16] B. Suri, J. Tithof, R. Mitchell, R. O. Grigoriev, and M. F. Schatz. Velocity profile in a two-layer Kolmogorov-like flow. *Phys. Fluids*, 26(5):053601, 2014.

[17] J. Tithof, B. Suri, R. K. Pallantla, R. O. Grigoriev, and M. F. Schatz. Bifurcations in a quasi-two-dimensional Kolmogorov-like flow. *Journal of Fluid Mechanics.*, 828:837–866, 2017.

[18] Ravi Pallantla. *Exact Coherent Structures and Dynamical Connections in a Quasi 2D Kolmogorov Like Flow*. PhD thesis, Georgia Institute of Technology, 2018.

[19] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[20] B. Suri, J. Tithof, R. O. Grigoriev, and M. F. Schatz. Forecasting fluid flows using the geometry of turbulence. *Phys. Rev. Lett.*, 118:114501, Mar 2017.

[21] Balachandra Suri, Jeffrey Tithof, Roman O. Grigoriev, and Michael F. Schatz. Unstable equilibria and invariant manifolds in quasi-two-dimensional kolmogorov-like flow. *Phys. Rev. E*, 98:023105, Aug 2018.