J. Fluid Mech. (2024), *vol.* 996, A25, doi:10.1017/jfm.2024.813

JFM PAPERS

Learning fluid physics from highly turbulent data using sparse physics-informed discovery of empirical relations (SPIDER)

Daniel R. Gurevich[1],†, **Matthew R. Golden**[2], **Patrick A.K. Reinbold**[2] **and Roman O. Grigoriev**[2]

[1]Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544, USA

[2]School of Physics, Georgia Institute of Technology, Atlanta, GA 30332, USA

(Received 13 November 2023; revised 1 April 2024; accepted 18 June 2024)

We show how a complete mathematical model of a physical process can be learned directly from data via a hybrid approach combining three simple and general ingredients: physical assumptions of smoothness, locality and symmetry, a weak formulation of differential equations and sparse regression. To illustrate this, we extract a complete system of governing equations of fluid dynamics – the Navier–Stokes equation, the continuity equation and the boundary conditions – as well as the pressure-Poisson and energy equations, from numerical data describing a highly turbulent channel flow in three dimensions. Whether they represent known or unknown physics, relations discovered using this approach take the familiar form of partial differential equations, which are easily interpretable and readily provide information about the relative importance of different physical effects. The proposed approach offers insight into the quality of the data, serving as a useful diagnostic tool. It is also remarkably robust, yielding accurate results for very high noise levels, and should thus be well suited for analysis of experimental data.

Key words: machine learning

1. Introduction

Physical theories are traditionally constructed in an iterative manner. At each step, discrepancies between predictions and existing experimental observations are used to improve the theory, making it more general and accurate. These improvements are usually instructed and constrained by first principles, including both general and domain

† Email address for correspondence: dgurevich@princeton.edu

knowledge. After this, new predictions are made and new experiments are designed to test these predictions, closing the loop. Humans play a key role in all aspects of this traditional procedure and can become a weak link when the amount of data becomes overwhelming or the patterns in the data are too complex. Recent advances in machine learning have started to change the scientific paradigm guiding the construction of physical theories by gradually taking humans out of the loop. For low-dimensional systems, physical relations in the form of algebraic and even differential equations can be constructed using symbolic regression directly from experimental data without using any physical intuition (Crutchfield & McNamara 1987; Bongard & Lipson 2007; Schmidt & Lipson 2009). For high-dimensional systems such as fluid flows, purely data-driven approaches often become intractable, and some physical intuition becomes necessary to guide the process (Karpatne *et al.* 2017).

The question is therefore what physical considerations can and should be used to constrain the problem sufficiently for the data-driven analysis to become tractable while leaving enough freedom to enable identification of physically meaningful relationships. Among the most general and least restrictive physical constraints are smoothness, locality and the relevant symmetries. In fact, some or all of these constraints have been implicitly assumed in most efforts to identify evolution equations via some form of regression from synthetic data (Bär, Hegger & Kantz 1999; Xu & Khanmohamadi 2008; Rudy *et al.* 2017; Schaeffer 2017; Reinbold, Gurevich & Grigoriev 2020) or experimental data (Reinbold *et al.* 2021). However, evolution equations are just one type of a relation that may be required to fully describe a physical system. Other examples include constraints, such as the divergence-free condition for the velocity field representing mass conservation for an incompressible fluid or the curl-free condition for the electric field in electrostatics, as well as boundary conditions. Previous studies have largely ignored the problem of identifying these equally important classes of relations for high-dimensional systems.

Sparse linear regression has so far proven to be the most versatile and robust approach for equation inference. Its original implementations, such as the sparse identification of nonlinear dynamics (SINDy) algorithm (Brunton, Proctor & Kutz 2016), were aimed at discovering evolution equations. Generalizations of this algorithm such as SINDy-PI (Kaheman, Kutz & Brunton 2020), which find sparse solutions to a collection of inhomogeneous linear systems, can be used to discover other types of relations as well. A number of alternatives for nonlinear regression aimed at inference of partial differential equations (PDEs) have been proposed as well. These include Gaussian processes (Raissi & Karniadakis 2018), gene expression programming (Ferreira 2001; Ma & Zhang 2022; Xing *et al.* 2022) and several neural network-based approaches such as equation learner (Martius & Lampert 2016; Sahoo, Lampert & Martius 2018), neural symbolic regression that scales (Biggio *et al.* 2021), PDE-LEARN (Stephany & Earls 2022) and PDE-Net (Long *et al.* 2018). While most of these approaches have been validated by reconstructing canonical PDEs or known governing equations, their potential for discovering previously unknown physics remains unclear, especially for spatially extended systems in more than one spatial dimension.

All of the above approaches suffer from inherent sensitivity to noise in the data which is amplified by spatial and/or temporal derivatives that appear in any physical relation described by a PDE. When the strong form of PDEs is used, it becomes difficult or even impossible to correctly identify governing equations involving higher-order derivatives for noise levels as low as a few per cent (Rudy *et al.* 2017; Raissi & Karniadakis 2018; Raissi, Perdikaris & Karniadakis 2019). This sensitivity can be addressed by using the weak form of the governing equations (Gurevich, Reinbold & Grigoriev 2019), as illustrated by its

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

successful application in equation inference approaches employing both linear regression (Reinbold *et al.* 2020; Messenger & Bortz 2021; Alves & Fiuza 2022) and nonlinear regression (Stephany & Earls 2023). Weak formulation was also found to be useful in problems involving latent variables (Reinbold *et al.* 2021) and unreliable or missing data (Golden *et al.* 2023).

The success of any approach to equation inference ultimately depends on the availability of a sufficiently rich function library (or, more typically, multiple libraries) which define the search space for one or more parsimonious relations describing the data. With rare exceptions, these libraries have previously been constructed in a largely *ad hoc* manner, either with little regard for the specifics of the physical problem or, alternatively, relying too much on the presumed-to-be-known physics. In this article, we describe a flexible and general data-driven approach for identifying a complete mathematical description of a physical system, including relevant boundary conditions, which we call sparse physics-informed discovery of empirical relations (SPIDER). Unlike SINDy and its variants, SPIDER is more than a linear regression algorithm: it is based on a systematic procedure for library generation informed by the symmetries of the system. We illustrate SPIDER by discovering the evolution equations, constraints and boundary conditions governing the flow of an incompressible Newtonian fluid from noisy numerical data using only very mild constraints which require no detailed knowledge of the physics. The implementation of SPIDER described here is publicly available at https://github.com/sibirica/SPIDER_channelflow.

The paper is organized as follows. Our hybrid equation inference approach is introduced and illustrated using an example of data representing numerical simulation of a highly turbulent flow in § 2. The results are discussed in § 3, and our conclusions are presented in § 4.

2. Sparse physics-informed discovery of empirical relations

It is well known that, in order for a data-driven approach to identify a sufficiently general mathematical model, the data must exhibit enough variation to sample the state space of the physical problem (Schaeffer, Tran & Ward 2018). Here, this is accomplished by using the numerical solution of a high-Reynolds-number flow through a rectangular channel from the Johns Hopkins University turbulence database (http://turbulence.pha.jhu.edu/Channel_Flow.aspx). The data set includes the flow velocity $\boldsymbol{u}$ and pressure p fully resolved in space and time. The channel dimensions are $L_x \times L_y \times L_z \times L_t = 8\pi \times 2 \times 3\pi \times 26$ (in non-dimensional units) and the data are stored on a spatio-temporal grid of size $2048 \times 512 \times 1536 \times 4000$. The (non-dimensional) viscosity is $\nu = 5 \times 10^{-5}$ and the corresponding friction Reynolds number is $Re_\tau \sim 10^3$. A representative snapshot of the data is shown in figure 1.

The immense size of the entire data set comprising 2.6×10^{13} 'measurements' illustrates the challenges faced by a purely data-driven approach. The locality property radically reduces the number of possible functional relations between measurements by constraining these to a small spatio-temporal neighbourhood of a given point. In particular, for smooth continuous fields, such functional relations have to be expressed in terms of their local values and local partial derivatives. For systems that are invariant with respect to spatial and temporal translation, a functional relation can be expressed in the form of a Volterra series

$$\sum_{n=1}^{N} c_n f_n \equiv \boldsymbol{c} \cdot \boldsymbol{f} = 0, \tag{2.1}$$

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

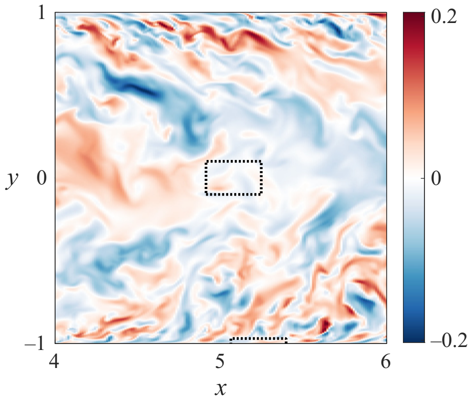

Figure 1. Snapshot of the velocity component u_z in a $z =$ const. plane over a portion of the entire computational domain. Sample integration domains (shown as dotted boxes) near the edge of the channel are much narrower than those in the middle due to the non-uniform grid spacing in the y direction.

where c_n are coefficients and f_n are products of the fields and their partial derivatives. For systems with translational symmetry in space and time, the most general relations of this type are nonlinear PDEs with constant coefficients. Most prior work focused on evolution equations, which are special cases of (2.1) where $c_1 = 1$ and f_1 is the first-order temporal derivative of one of the fields. Other special cases include differential equations that do not involve temporal derivatives and algebraic relations between the fields that involve no derivatives at all, which have largely been ignored by the machine learning literature in the context of spatially extended systems.

Our aim here is to identify a parsimonious mathematical model of the flow in the form of a system of PDEs, along with appropriate boundary conditions, directly from data representing the velocity and pressure fields, $\boldsymbol{u}$ and p. The key observation here is that the form of the functional relations (2.1) can be restricted sufficiently using the rotational symmetry constraint. All terms f_n have to transform in the same way under rotations and reflections, with the transformation rule corresponding to a particular representation of the orthogonal symmetry group $O(3)$. For non-relativistic systems, the symmetry group involves rotations about any axis in three-dimensional space and reflections across any plane, with the representations corresponding to tensors of various ranks. Here, we will restrict our attention to the two lowest rank tensors, i.e. scalars and vectors, although the same approach trivially extends to tensors of any rank (Golden *et al.* 2023).

2.1. *Learning evolution equations and constraints*

The functional form of the mathematical model will always depend on the choice of the variables. The best choice may not be obvious, and this is where relevant domain knowledge is extremely helpful. In the present problem, we will assume that the variables are the pressure field p and the velocity field $\boldsymbol{u}$ and that both variables are fully observed. The pressure is a scalar and the velocity is a vector. The differential operators ∂_t and ∇ transform as a scalar and a vector, respectively. Using these four objects, we can construct tensors of any rank using tensor products and contractions (Golden *et al.* 2023). For instance, the terms $\boldsymbol{u}$, $\partial_t \boldsymbol{u}$ and ∇p all transform as vectors. To illustrate the procedure, we will include all possible terms f_n up to cubic in p, $\boldsymbol{u}$, ∂_t and/or ∇ that can be constructed

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

from the data and its derivatives, yielding a scalar library

$$\mathcal{L}_0^{(3)} = \{1, p, \nabla \cdot \boldsymbol{u}, \partial_t p, p^2, u^2, p^3, \boldsymbol{u} \cdot \nabla p, \nabla^2 p, p\partial_t p, \partial_t^2 p, p\nabla \cdot \boldsymbol{u}, u^2 p, \boldsymbol{u} \cdot \partial_t \boldsymbol{u}\}, \quad (2.2)$$

where $u^2 = \boldsymbol{u} \cdot \boldsymbol{u}$ and a vector library

$$\begin{aligned}\mathcal{L}_1^{(3)} = \{&\boldsymbol{u}, \partial_t \boldsymbol{u}, \nabla p, p\boldsymbol{u}, (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}, \nabla^2 \boldsymbol{u}, \partial_t^2 \boldsymbol{u}, u^2 \boldsymbol{u}, p^2 \boldsymbol{u},\\ &\partial_t \nabla p, p\nabla p, \boldsymbol{u}(\nabla \cdot \boldsymbol{u}), (\nabla \boldsymbol{u}) \cdot \boldsymbol{u}, \nabla(\nabla \cdot \boldsymbol{u}), p\partial_t \boldsymbol{u}, \boldsymbol{u}\partial_t p\}, \end{aligned} \quad (2.3)$$

where the superscript denotes the maximal complexity (order, for short) of the terms included in the library and the subscript, the irreducible representation. Note that the vector library constructed in this way generalizes the model of flocking in active matter due to Toner & Tu (1998) by including the most general dependence on the pressure p allowed by the symmetry. These two libraries, together with the relation (2.1), will form the search space containing all of the candidate relations describing the fluid physics in the bulk. Note that scalars and vectors (i.e. rank-0 and rank-1 tensors) are irreducible representations of the symmetry group $O(3)$. This is not the case for rank-2 tensors, for instance, which can be broken into three different irreducible representations corresponding to the symmetric traceless component, antisymmetric component and the trace. Similarly, scalars and pseudoscalars, or vectors and pseudovectors, belong to different irreducible representations of $O(3)$. Reflection covariance can be used to exclude pseudoscalars and pseudovectors such as $\boldsymbol{u} \cdot (\nabla \times \boldsymbol{u})$ and $\nabla \times \boldsymbol{u}$ from the scalar and vector libraries.

It should be emphasized that no domain knowledge specific to the system, aside from the symmetry (rotational and translational) and the choice of variables, has been used in constructing these libraries. For instance, it is not necessary to know that $\boldsymbol{u}$ and p represent the velocity and pressure of a fluid. This is in direct contrast to most prior studies (Raissi & Karniadakis 2018; Reinbold *et al.* 2020, 2021; Messenger & Bortz 2021; Ma & Zhang 2022) that used model libraries directly inspired by first-principles analysis of the fluid flows considered there.

It is also useful to put the very modest size of libraries $\mathcal{L}_0^{(3)}$ and $\mathcal{L}_1^{(3)}$ in perspective. In order to identify the evolution equation for the vorticity $\omega = \nabla \times \boldsymbol{u}$, Rudy *et al.* (2017) used a library analogous to $\mathcal{L}_1^{(3)}$ that was constructed using a brute-force approach ignoring the symmetries of the problem. The terms that were chosen by the authors included $\partial_t \omega$ as well as 'polynomial terms of vorticity and all velocity components up to second degree, multiplied by derivatives of the vorticity up to second order', yielding a set of $N = 1 + (1 + 2d + d(d-1)/2)^2$ terms in d spatial dimensions. For the three-dimensional geometry considered here, the corresponding library would contain 101 distinct terms, almost an order of magnitude more than what is included in our more physically comprehensive library $\mathcal{L}_1^{(3)}$, which was constructed using symmetry constraints. In fact, incorporating knowledge of the Galilean invariance in our system would have allowed for even more compact libraries to be used without loss of expressivity

$$\mathcal{L}_{0G}^{(3)} = \{1, p, \nabla \cdot \boldsymbol{u}, p^2, p^3, \partial_t p + \boldsymbol{u} \cdot \nabla p, \nabla^2 p, p\nabla \cdot \boldsymbol{u}\}, \quad (2.4)$$

$$\mathcal{L}_{1G}^{(3)} = \{\partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}, \nabla p, \nabla^2 \boldsymbol{u}, p\nabla p, \nabla(\nabla \cdot \boldsymbol{u})\}. \quad (2.5)$$

We will, however, use the libraries $\mathcal{L}_0^{(3)}$ and $\mathcal{L}_1^{(3)}$ rather than $\mathcal{L}_{0G}^{(3)}$ and $\mathcal{L}_{1G}^{(3)}$ in the subsequent analysis to illustrate that, while the knowledge of all the symmetries is useful, it is not essential.

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

2.1.1. *The effect of noise*

Two common scenarios where equation inference would be of particular value are when the data are generated experimentally (Reinbold *et al.* 2021; Joshi *et al.* 2022; Golden *et al.* 2023) or when the data represent coarse graining of the results of direct numerical simulation. An example of the latter is fully kinetic simulations of plasma used to obtain a hydrodynamic description (Alves & Fiuza 2022). In both instances, the data will inevitably be noisy, e.g. due to measurement inaccuracies in experiment or fluctuations of the computed coarse-grained fields. To investigate the effects of noise, in addition to the original simulation data downloaded from the turbulence database, we also used synthetic data with varying levels of additive uniform noise. Specifically, we define the noisy data $f_\sigma = f + \sigma \xi_f s_f$, where $f \in \{p, u_x, u_y, u_z\}$ are the hydrodynamic fields, σ is the noise level and ξ_f is noise independently sampled from the uniform distribution over $[-1, 1]$ at each space–time point.

Parsimonious scalar and vector relations describing velocity and pressure data can be identified by performing sparse regression using the libraries $\mathcal{L}_0$ and $\mathcal{L}_1$, respectively. In the strong form, the terms involving higher-order derivatives, such as $\nabla^2 p$ and $\nabla^2 \boldsymbol{u}$, will be extremely sensitive to noise (Rudy *et al.* 2017; Reinbold & Grigoriev 2019). To make the regression more robust, we use the weak form of both PDEs following the approach introduced in our earlier work (Gurevich *et al.* 2019). Specifically, we multiply each equation by a smooth weight function $w_j(\boldsymbol{x}, t)$ and then integrate it over a rectangular spatio-temporal domain $\varOmega_i$ of size $H_x \times H_y \times H_z \times H_t$. All side lengths of $\varOmega_i$ are fixed to $H_i = 32$ grid points of the numerical grid; this size roughly corresponds to the characteristic length and time scales of the flow field (cf. figure 1).

The derivatives are shifted from the data ($\boldsymbol{u}$ and p) onto the weight functions w whenever possible via integration by parts, after which the integrals are evaluated numerically using trapezoidal quadratures. (In the few cases where it is not possible to fully integrate a term by parts, remaining derivatives are evaluated using finite differences.) For the scalar library $\mathcal{L}_0$, we use scalar weight functions of the form

$$\left.\begin{aligned} w(\boldsymbol{x}, t) &= \tilde{w}(\bar{x})\tilde{w}(\bar{y})\tilde{w}(\bar{z})\tilde{w}(\bar{t}), \\ \tilde{w}(s) &= (1 - s^2)^\beta, \end{aligned}\right\} \tag{2.6}$$

where the bar denotes non-dimensionalization using the order-preserving affine map $[x_{min}, x_{max}] \to [-1, 1]$. In the case of the vector library $\mathcal{L}_1$, we instead use vector weight functions $w(\boldsymbol{x}, t)\boldsymbol{e}_k$ aligned along each of the coordinate axes. Note that, for the term libraries considered in this problem which involve a Laplacian of p or $\boldsymbol{u}$, we should have $\beta \geq 2$, as this allows us to discard the boundary terms generated during integration by parts. We set $\beta = 8$ in our analysis since this choice (i) ensures that all the boundary terms vanish and (ii) maximizes the accuracy of numerical quadrature along the uniformly gridded dimensions (Gurevich *et al.* 2019). The non-uniform grid in the y-direction will control the error of the quadrature and increasing β further has no benefit. This is illustrated in figure 2, which shows how the quadrature error scales with β for a test function $\cos(2\pi y)$. The error decreases quickly with increasing β for the uniform grid, plateauing at $\beta \geq 8$. In contrast, the error is found to be almost independent of β for the non-uniform grid.

By repeating this procedure for a number of integration domains $\varOmega_i$ contained within the full dataset, we construct a feature matrix $Q = [\boldsymbol{q}_1 \cdots \boldsymbol{q}_N]$ whose the columns $\boldsymbol{q}_n$ correspond to the various terms in either $\mathcal{L}_0$ or $\mathcal{L}_1$. For instance, for the scalar

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

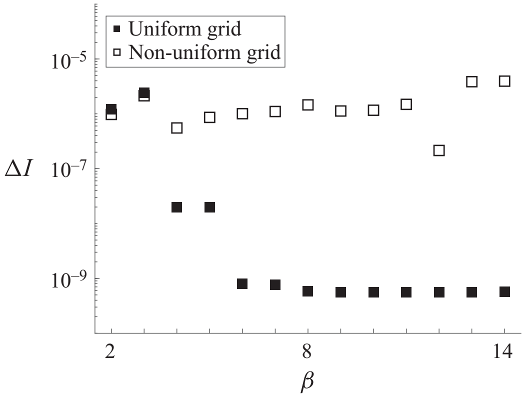

Figure 2. Numerical error ΔI of the integral $I_\beta = \int_{-1}^{1} \cos(2\pi y)(1-y^2)^\beta \, \mathrm{d}y$ evaluated using the trapezoid rule on a 32-point grid. The results are shown for the uniform grid (black squares) and the non-uniform grid (white squares) representing the y-coordinates of points taken from the middle of the channel.

library $\mathcal{L}_0$,

$$Q_{ij} = \frac{1}{V_i S_j} \int_{\Omega_i} w(\boldsymbol{x}, t) f_j(\boldsymbol{x}, t) \, \mathrm{d}^3\boldsymbol{x} \, \mathrm{d}t, \quad V_i = \int_{\Omega_i} |w(\boldsymbol{x}, t)| \, \mathrm{d}^3\boldsymbol{x} \, \mathrm{d}t, \tag{2.7a,b}$$

where $S_j = S[f_j]$ is the scale of the library term f_j in dimensional units. We estimate the scales of all library terms using the characteristic time scales $T_{\boldsymbol{u}}$, T_p, length scales $L_{\boldsymbol{u}}$, L_p, mean values $\mu_{\boldsymbol{u}}$, μ_p and standard deviations $\sigma_{\boldsymbol{u}}$, σ_p of the velocity and pressure fields across the dataset. For instance, for a vector field $\boldsymbol{u}$, the length and time scales can be estimated from the magnitudes of its finite-difference derivatives

$$T_{\boldsymbol{u}} \equiv \frac{\sigma_{\boldsymbol{u}}}{\sqrt{\langle \partial_t \boldsymbol{u} \cdot \partial_t \boldsymbol{u} \rangle}}, \tag{2.8}$$

$$L_{\boldsymbol{u}} \equiv \frac{\sigma_{\boldsymbol{u}}}{\sqrt{\langle (\boldsymbol{\nabla u}) \cdot (\boldsymbol{\nabla u}) \rangle}}, \tag{2.9}$$

where the dot products in the denominators are taken over all the indices. We then use the heuristic that the scale of a library term is the product of the scales of its factors: $S[\boldsymbol{uv}] = S[\boldsymbol{u}]S[\boldsymbol{v}]$, and for $k \geq 1$

$$S[p] = \mu_p, \quad S[\boldsymbol{u}] = \mu_{\boldsymbol{u}}, \tag{2.10a,b}$$

$$S[\nabla^k p] = L_p^{-k} \sigma_p, \quad S[\nabla^k \boldsymbol{u}] = L_{\boldsymbol{u}}^{-k} \sigma_{\boldsymbol{u}}, \tag{2.11a,b}$$

$$S[\partial_t^k p] = T_p^{-k}, \sigma_p \quad S[\partial_t^k \boldsymbol{u}] = T_{\boldsymbol{u}}^{-k} \sigma_{\boldsymbol{u}}. \tag{2.12a,b}$$

Note that the mass scale does not appear explicitly, as the data are given in units in which the density $\rho = 1$. The non-dimensionalization procedure ensures the magnitudes of all columns are comparable, which can dramatically improve the accuracy and robustness of regression. The problem of determining the unknown coefficients $\boldsymbol{c} = [c_1, \ldots, c_N]^\top$ is cast as the solution of an overdetermined linear system of the form

$$\boldsymbol{Qc} = 0. \tag{2.13}$$

In this article, we sample Ω_i from a 64^4-point region of the data lying either in the middle (i.e. the symmetry plane of the flow) or at the edge of the channel (see table 5). Examples

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

of both types of domains are shown in figure 1. We use 256 randomly sampled integration domains Ω_i with 32^4 gridpoints to construct the system (2.13). This yields $M = 256$ linear equations on the coefficients of $\mathcal{L}_0$ and $3M = 768$ linear equations on the coefficients of $\mathcal{L}_1$. A discussion of how the results are affected by correlated noise, the number of integration domains used and the resolution of the data can be found in Appendix B.

2.1.2. *Selection of parsimonious relations*

Note that the linear system (2.13) is homogeneous and treats all terms in the library on equal terms. This is in contrast to SINDy (Brunton *et al.* 2016) and its variants (Messenger & Bortz 2021) that solve an inhomogeneous linear system, or many such systems in the case of SINDy-PI (Kaheman *et al.* 2020). Its solutions have a degree of freedom corresponding to the normalization of $\boldsymbol{c}$, which can be eliminated by arbitrarily setting one of the coefficients, say c_1, to unity, as done in SINDy, or by fixing the norm of $\boldsymbol{c}$ as in this study. The solutions of a constrained least squares problem

$$\boldsymbol{c} = \underset{\|\boldsymbol{c}\|=1}{\arg\min} \|\boldsymbol{Q}\boldsymbol{c}\|, \tag{2.14}$$

are given by the right singular vector of $\boldsymbol{Q}$ corresponding to the smallest singular value. It is worth noting that, when multiple singular values of $\boldsymbol{Q}$ are small, there may be several 'good' independent solutions for $\boldsymbol{c}$ representing different dominant balances. This is a less restrictive approach compared with SINDy and allows a broader class of functional relations to be identified. It is also more computationally efficient than SINDy-PI, which aims to address the same limitation.

In order to obtain a parsimonious physical relation, we must find a sparse coefficient vector $\boldsymbol{c}^*$ such that the residual $\|\boldsymbol{Q}\boldsymbol{c}^*\|$ is comparable to the residual $\|\boldsymbol{Q}\boldsymbol{c}\|$ with dense $\boldsymbol{c}$ given by (2.14). The identified relations either contain a single term or several terms. If the matrix $\boldsymbol{Q}$ has been properly non-dimensionalized, single-term relations will correspond to columns with small norms. We will use the heuristic that $f_j = 0$ is a valid single-term relation if $\|\boldsymbol{q}_j\| \ll \sqrt{M}$, where M is the number of integration domains. In particular, for the data without added noise, the scalar library $\mathcal{L}_0$ is found to contain terms with $\|\boldsymbol{q}_j\| \approx 10^{-6}\sqrt{M}$, which correspond to the incompressibility condition

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2.15}$$

and its trivial corollary $p\nabla \cdot \boldsymbol{u} = 0$. Both single-term relations are found using data from the middle of the channel as well as data near the boundary. The single-term relation heuristic crucially relies on proper non-dimensionalization such that velocity gradients are $O(1)$. A more general approach is to compare with the characteristic size of the uncontracted tensor, which in this case is the rate of strain $\nabla_i u_j$. In contrast, prior sparsification algorithms such as SINDy (Brunton *et al.* 2016), implicit SINDy (Mangan *et al.* 2016) and SINDy-PI (Kaheman *et al.* 2020) are unable to identify single-term relations, as these are not examined separately. Note that direct identification of single-term relations is both more robust and more computationally efficient than identification through regression.

Once the library has been pruned, multiple-term relations can be identified by an iterative greedy algorithm. At each iteration, we use the singular value decomposition of $Q^{(N)} = [\boldsymbol{q}_1 \cdots \boldsymbol{q}_N]$ to find $\boldsymbol{c}^{(N)}$ as described previously. We also compute the residual $r^{(N)} = \|Q^{(N)}\boldsymbol{c}^{(N)}\|$. Next, we consider all of the candidate relations formed by dropping one of the terms and eliminating the corresponding column from $Q^{(N)}$. We select

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

the candidate relation with $N-1$ terms that achieves the smallest residual and then repeat until only one term remains. This yields a sequence of increasingly sparse relations described by N-dimensional coefficient vectors $\boldsymbol{c}^{(N)}$, forming an approximately Pareto-optimal set (Miettinen 2012). Note that the use of the absolute residual r guarantees that the residual is a monotonic function of the number of terms N, which is not the case for the relative residual $\eta = r/\max_n \|c_n \boldsymbol{q}_n\|$ used by Reinbold *et al.* (2021) and Golden *et al.* (2023).

There are many reasonable ways to select a final relation from this sequence based on the trade-off between their parsimony (i.e. number of terms N) and accuracy, quantified by the residuals $r^{(N)}$. For instance, one might select the simplest relation which achieves a relative residual of less than, say, 1 % or the relation for which discarding a single term results in the largest relative increase in the residual. In this article, we follow Gurevich *et al.* (2019): specifically, we choose the relation described by the coefficient vector $\boldsymbol{c}^{(N)}$ where $N = \max\{n : r^{(n)}/r^{(n-1)} > \gamma\}$, where the parameter $\gamma = 1.25$ was selected empirically. Once the functional form of a parsimonious relation is determined, the mean values of the coefficients and their standard deviations are computed by subsampling $\boldsymbol{Q}$ 128 times, with each member of the ensemble constructed using one half of the rows of $\boldsymbol{Q}$ selected at random. Finally, to enhance the interpretability of the result, the equation is rescaled by setting the largest coefficient to unity, which defines the mean values $\bar{c}_n$ and their uncertainties s_n.

After a sparse single-term or multi-term relation has been identified, one may search for additional sparse relations contained within the same library. Note that the form of our libraries implies a simple connection between a relation and its direct algebraic implications: if $\boldsymbol{c} \cdot \boldsymbol{f} = 0$, then $\boldsymbol{c} \cdot (g\boldsymbol{f}) = 0$ and $\boldsymbol{c} \cdot (\partial_s \boldsymbol{f}) = 0$ for any term g in any of the libraries and set of partial derivatives s. Moreover, iteratively applying these rules produces all implied relations (with a higher complexity) for a given base relation (of a lower complexity). Each implied relation is an equation which allows one of the terms, say the highest-order one, to be eliminated from the respective library without loss of expressivity. (In principle, this procedure could reduce sparsity of future identified equations if applied to many-term relations, but in such cases we find that the most complex term is unlikely to reappear in another independent equation.)

This can be leveraged to devise a simple and efficient algorithm for finding all relations contained within each library $\mathcal{L}_m$. Consider a nested sequence of sub-libraries: $\mathcal{L}_m^{(1)}$ containing terms up to first order, $\mathcal{L}_m^{(2)}$ containing terms up to second order and so on. On each sub-library, we repeatedly run single-term and multi-term regressions. Whenever a new relation is identified, we construct all implied relations in each library and then eliminate from its appropriate library the highest-order term from the base relation as well as from each implied relation before re-running the regression. For instance, the discovery of the incompressibility condition $\nabla \cdot \boldsymbol{u} = 0$ from $\mathcal{L}_0^{(2)}$ would lead to the identification of the implied relations $p\nabla \cdot \boldsymbol{u} = 0$, $\boldsymbol{u}(\nabla \cdot \boldsymbol{u}) = 0$, and $\nabla(\nabla \cdot \boldsymbol{u}) = 0$ and the elimination of these terms from $\mathcal{L}_0^{(3)}$ and $\mathcal{L}_1^{(3)}$. If no more relations remain after eliminating terms, we advance to the next sub-library (either the same-order sub-library from the next library, or after all sub-libraries of a given order have been exhausted, the next-order sub-library from the first library), continuing until each library has been completely examined. This procedure can be used to identify, for instance, the pressure-Poisson equation and the energy equation from a scalar library of sufficiently high order, as discussed below.

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

(a)

	σ	$\partial_t \boldsymbol{u}$	$(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}$	∇p	$\nabla^2 \boldsymbol{u}$
$\bar{c}_n$	0 %	0.999996	0.999998	1	-4.99996×10^{-5}
	50 %	0.9928	0.995	1	-5.1×10^{-5}
	100 %	0.9927	0.99	1	-5.2×10^{-5}
s_n	0 %	5×10^{-8}	3×10^{-6}	4×10^{-6}	3×10^{-10}
	50 %	1×10^{-4}	5×10^{-3}	8×10^{-3}	8×10^{-7}
	100 %	2×10^{-4}	1×10^{-2}	2×10^{-2}	1×10^{-6}
χ_n	0 %	0.79	1	0.50	0.45
	50 %	0.79	1	0.51	0.46
	100 %	0.82	1	0.51	0.47

(b)

	σ	$\partial_t \boldsymbol{u}$	$(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}$	∇p	$\nabla^2 \boldsymbol{u}$
$\bar{c}_n$	0 %	0.99986	1	0.99986	-5.003×10^{-5}
	50 %	0.991	0.990	1	0
	100 %	0.988	0.986	1	0
	300 %	1	0.983	1	0
s_n	0 %	2×10^{-7}	1×10^{-6}	2×10^{-5}	2×10^{-8}
	50 %	1×10^{-4}	1×10^{-3}	1×10^{-2}	0
	100 %	2×10^{-4}	1×10^{-3}	2×10^{-2}	0
	300 %	6×10^{-4}	5×10^{-3}	0	0
χ_n	0 %	0.99	1	0.07	0.006
	50 %	0.99	1	0.07	0
	100 %	0.99	1	0.07	0
	300 %	1	0.99	0	0

Table 1. Coefficients of the momentum equation (2.16) in the presence of varying levels of noise and data from (*a*) the edge of the channel and (*b*) the middle of the channel. The rows show the mean values of the coefficients $\bar{c}_n$ (normalized by the magnitude of the largest one), their uncertainties s_n and the magnitudes of the terms χ_n (normalized by the magnitude of the largest term).

2.1.3. *Identified equations and robustness to noise*

Regression performed using the vector library $\mathcal{L}_1^{(3)}$ identifies a single relation representing momentum balance

$$c_1 \partial_t \boldsymbol{u} + c_2 (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + c_3 \nabla p + c_4 \nabla^2 \boldsymbol{u} = 0. \tag{2.16}$$

Table 1 lists the surviving terms f_n, the corresponding coefficients c_n, their uncertainties s_n and the respective magnitudes $\chi_n = \|c_n \boldsymbol{q}_n\|$ of the terms which can be used to identify dominant balances in different regions. For data from the edge of the channel, the Navier–Stokes equation with accurate coefficients, including the small viscosity, is identified for all noise levels. In particular, for noiseless data, the magnitude of the viscosity is identified correctly to several significant digits, even though the viscous term involves a second-order derivative, which is the highest in the equation. The corresponding sequence of residuals $r^{(N)}$ can be found in figure 3(*c*).

For data from the middle of the channel, different special cases of (2.16) are identified for different noise levels. For noise levels up to approximately 15 %, sparse regression still identifies the Navier–Stokes equation. For higher noise levels (up to 100 %), the Euler equation is identified instead, also with fairly accurate coefficients. For extreme levels of noise (up to 300 %), the inviscid Burgers equation is identified. Note that the sequence

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

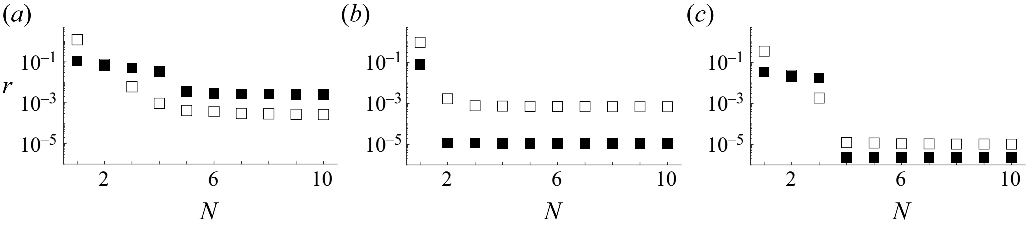

Figure 3. Dependence of the residual r on the number of terms N retained in a given relation in the noiseless case. Black (white) squares represent data collected near the edge (in the middle) of the channel. The identified relations are (a) the energy equation, (b) the pressure equation and (c) the momentum equation.

in which the terms in the momentum equation stop being recovered, as the noise level is increased, corresponds to their magnitudes χ_n in the middle of the channel.

No multi-term relations are found, for any data set, from the scalar library $\mathcal{L}_0^{(3)}$, suggesting it is incomplete. In order to identify any multi-term relations, this library was expanded by including terms quartic in $\boldsymbol{u}$ and/or ∇

$$\mathcal{L}_0^{(4)} = \mathcal{L}_0^{(3)} \cup \{\nabla^2(u^2), \nabla \cdot [(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}], (\nabla \cdot \boldsymbol{u})^2, \nabla \boldsymbol{u} : \nabla \boldsymbol{u}^\top, \nabla \boldsymbol{u} : \nabla \boldsymbol{u}, \nabla \cdot (u^2 \boldsymbol{u}), \boldsymbol{u}^4\}. \tag{2.17}$$

Note that this expanded library is not exhaustive, i.e. it does not include any fourth-order terms which involve either p or ∂_t. Whenever possible, the additional terms were written in conservative form to decrease numerical error associated with evaluation of higher-order derivatives in weak form. Two relations are identified from this expanded scalar library via sparsification

$$c_1 \partial_t E + c_2 \nabla \cdot (\boldsymbol{u}E) + c_3 \boldsymbol{u} \cdot \nabla p + c_4 \nabla^2 E + c_5 \nabla \boldsymbol{u} : \nabla \boldsymbol{u} = 0, \tag{2.18}$$

$$c_6 \nabla^2 p + c_7 \nabla \cdot [(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}] + c_8 = 0, \tag{2.19}$$

where $E = u^2/2$ is the energy density. The two relations are discovered robustly for both data sampled from the edge and the middle of the channel; however, the order in which they are found depends on the sampled region. The corresponding sequences of residuals $r^{(N)}$ can be found in figure 3(a,b), and the values of the coefficients c_n for different noise levels are summarized in tables 2 and 3.

For sufficiently low levels of noise (below approximately 10 %), relation (2.18) corresponds to the well-known energy equation, again with fairly accurate coefficients, no matter which region of the flow the data comes from. The accuracy of the coefficients decreases somewhat as the level of noise is increased, as expected. For data from the middle of the channel, small terms such as $\nabla \boldsymbol{u} : \nabla \boldsymbol{u}$ start to disappear as the noise level is increased, similar to what we found for the momentum equation. Relation (2.19) takes the form of the pressure-Poisson equation for moderately noisy data from both the middle and the edge of the channel, also with fairly accurate coefficients. The surprising observation is that, for noiseless data from the middle of the channel, sparse regression reliably identifies a small correction to the pressure-Poisson equation, a term proportional to unity. We will therefore refer to relation (2.19) simply as the pressure equation.

Note that neither the energy equation nor the pressure-Poisson equation is an independent relation; both can be derived from the Navier–Stokes equation and the incompressibility condition. Further, note that SPIDER is superior to alternative approaches to equation inference in both versatility and accuracy: for instance, neither

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

(a)

	σ	$\partial_t E$	$u_i \nabla_i p$	$\nabla_i (u_i E)$	$(\nabla_i u_j)(\nabla_i u_j)$	$\nabla^2 E$
$\bar{c}_n$	10 %	1	1.00	0.99418	0.0000404	−0.0000473
s_n	10 %	5×10^{-3}	2×10^{-2}	3×10^{-5}	5×10^{-7}	7×10^{-7}
χ_n	10 %	0.69	0.27	0.75	1	0.76

(b)

	σ	$\partial_t E$	$\boldsymbol{u} \cdot \nabla p$	$\nabla \cdot (\boldsymbol{u} E)$	$\nabla \boldsymbol{u} : \nabla \boldsymbol{u}$	$\nabla^2 E$
$\bar{c}_n$	0 %	0.99334	1	0.993162	0.000048	−0.0000494
	1 %	0.99361	1	0.993447	0.000050	−0.0000493
	10 %	0.9811	1	0.98074	0	−0.000047
s_n	0 %	2×10^{-5}	5×10^{-4}	7×10^{-6}	2×10^{-6}	2×10^{-7}
	1 %	3×10^{-5}	9×10^{-4}	1×10^{-5}	3×10^{-6}	3×10^{-7}
	10 %	2×10^{-4}	4×10^{-3}	6×10^{-5}	0	2×10^{-6}
χ_n	0 %	0.99	0.068	1	0.0011	0.0066
	1 %	0.99	0.068	1	0.0012	0.0066
	10 %	0.99	0.069	1	0	0.0064

Table 2. Coefficients of the energy equation (2.18) in the presence of varying levels of noise and data from (*a*) the edge and (*b*) the middle of the channel. The quantities $\bar{c}_n$, s_n, and χ_n are defined in the caption of table 1.

(a)

	σ	$\nabla^2 p$	$\nabla_i \nabla_j (u_i u_j)$
$\bar{c}_n$	0 %	0.99995	1
	100 %	1.000	1
	500 %	0.99	1
s_n	0 %	1×10^{-5}	4×10^{-9}
	100 %	5×10^{-3}	2×10^{-6}
	500 %	5×10^{-2}	2×10^{-5}
χ_n	0 %	1.00	1
	100 %	1.00	1
	500 %	0.79	1

(b)

	σ	$\nabla^2 p$	$\nabla_i \nabla_j (u_i u_j)$	1
$\bar{c}_n$	0 %	1	0.999789	0.00038
	10 %	1	0.9965	0
	20 %	1	0.9926	0
s_n	0 %	6×10^{-5}	4×10^{-6}	1×10^{-5}
	10 %	3×10^{-3}	2×10^{-4}	0
	20 %	6×10^{-3}	4×10^{-4}	0
χ_n	0 %	1	1.00	0.0016
	10 %	1.00	1	0
	20 %	0.99	1	0

Table 3. Coefficients of the pressure equation (2.19) in the presence of varying levels of noise and data from (*a*) the edge and (*b*) the middle of the channel. The quantities $\bar{c}_n$, s_n and χ_n are defined in the caption of table 1.

of the two scalar relations would be identified using SINDy, which assumes either $\partial_t \boldsymbol{u}$ or $\partial_t p$ to be present. The energy equation has the form of an evolution equation but involves a temporal derivative of u^2 rather than $\boldsymbol{u}$, while the pressure-Poisson equation is a constraint which involves no temporal derivatives at all. Finally, note that one can also heuristically identify the incompressibility condition, the pressure-Poisson equation and the energy equation from the right singular vectors of $\boldsymbol{Q}$ corresponding to the three smallest singular values without pruning $\mathcal{L}_0^4$. These singular vectors are dense, but only the coefficients associated with the corresponding dominant balances are $O(1)$.

For both sampled regions, all libraries and all noise levels, the residual r asymptotes to a constant value for large N. In the noiseless case shown in figure 3, the asymptotic value of the residual is determined by the discretization of the data, both in the numerical simulations and in evaluating the integrals using quadratures, as shown by Gurevich *et al.*

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

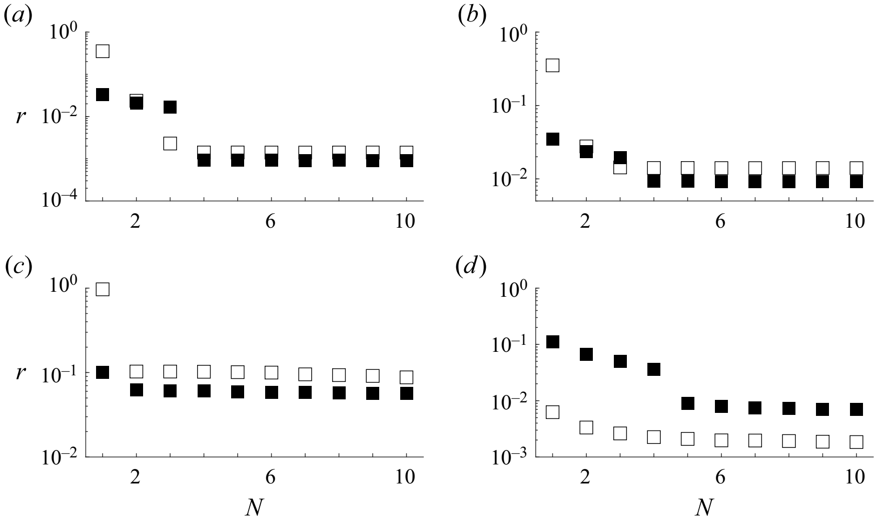

Figure 4. Dependence of the residual r on the number of terms N retained in the momentum equation for synthetic data with added noise for noise levels of (*a*) 10 % and (*b*) 100 %. The Navier–Stokes equation is identified in all the cases except for data from the middle of the channel with 100 % noise. In that case, the Euler equation is found instead. The residual for the pressure equation identified from data with 20 % noise (*c*) and for the energy equations identified from data with 10 % noise (*d*). Black (white) squares represent data collected near the edge (in the middle) of the channel.

(2019). In the noisy case, shown in figure 4, the asymptotic value of the residual is instead determined by the level of noise and is higher than in the noiseless case, as expected. It should be emphasized that physically meaningful relations can be identified in the presence of very high levels of noise, illustrating the robustness of SPIDER.

The power of the weak formulation compared with the strong form is vividly illustrated by other physical relations containing Laplacians as well. For instance, the vorticity equation can only be identified correctly in strong form for noise levels up to approximately 1 %, as shown by Rudy *et al.* (2017). The pressure-Poisson equation (2.19), which also contains second-order derivatives, can be identified in weak form for noise levels up to 20 % (500 %) for data from the middle (edge) of the domain. The energy equation (2.18) can be identified in weak form using the data from the edge of the channel with up to 10 % noise and using data from the middle of the channel with up to 3 % noise. The latter equation is more challenging to identify from noisy data due to the presence of the term $\boldsymbol{\nabla u} : \boldsymbol{\nabla u}$ which cannot be integrated by parts and was computed using second-order accurate finite differencing. However, it should be noted that a more sophisticated procedure for numerically differentiating the data would likely allow this equation to be identified even at higher noise levels.

Note that, although we used the absolute residual r in sparse regression, it is the magnitude of the relative residual η that better quantifies the accuracy of the identified relations (Reinbold *et al.* 2021). The relative residuals are indeed quite low: $4 \times 10^{-4}(2 \times 10^{-2})$ for the energy equation, $2 \times 10^{-3}(1 \times 10^{-4})$ for the two-term pressure-Poisson equation and $4 \times 10^{-5}(5 \times 10^{-5})$ for the Navier–Stokes equation identified using noiseless data from the middle (edge) of the channel. The three-term pressure equation has a relative residual of 8×10^{-4} in the middle of the channel, which is less than a half that of the pressure-Poisson equation.

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

2.2. *Learning boundary conditions*

SPIDER can also be used to discover boundary conditions. In this case, the rotational symmetry is partially broken: instead of rotations in all three spatial directions, the problem is only invariant with respect to rotations about the normal $\boldsymbol{n}$ to the boundary. The reduced symmetry group describing boundary conditions is $O(2)$. The library of terms that transform as vectors near the boundary includes $\boldsymbol{n}$ in addition to $\boldsymbol{u}$ and ∇. We exclude time derivatives, because these can be eliminated with the help of the bulk equations. We also exclude the dependence on p to keep the library to a reasonable size (this dependence is trivial to restore). Retaining terms that contain each of $\boldsymbol{u}$ and ∇ at most once ($\boldsymbol{n}$ has unit magnitude and is allowed to appear an arbitrary number of times) yields a vector library

$$\mathcal{L}_1^2 = \{\boldsymbol{u}, \boldsymbol{n}, (\boldsymbol{u}\cdot\boldsymbol{n})\boldsymbol{n}, \nabla(\boldsymbol{u}\cdot\boldsymbol{n}), (\boldsymbol{n}\cdot\nabla)\boldsymbol{u}, \boldsymbol{n}(\nabla\cdot\boldsymbol{u})\}. \tag{2.20}$$

Next, since they transform differently under rotation about the surface normal $\boldsymbol{n}$, we separate the normal and tangential components by applying the projection operators $P_\perp = \boldsymbol{nn}$ and $P_\parallel = \mathbb{1} - \boldsymbol{nn}$ to the library (2.20), where $\boldsymbol{nn}$ represents the tensor product of the normal vectors. We prune all terms which have identically vanishing projections. Furthermore, we can also prune all terms involving $\nabla\cdot\boldsymbol{u}$, since we have already identified the continuity equation (2.15). This results in two libraries for the boundary conditions

$$\mathcal{L}_\parallel^2 = \{P_\parallel\boldsymbol{u}, P_\parallel\nabla(\boldsymbol{u}\cdot\boldsymbol{n}), P_\parallel(\boldsymbol{n}\cdot\nabla)\boldsymbol{u}\}, \tag{2.21}$$

$$\mathcal{L}_\perp^2 = \{\boldsymbol{n}\cdot\boldsymbol{u}, 1, \boldsymbol{n}\cdot\nabla(\boldsymbol{u}\cdot\boldsymbol{n}), \boldsymbol{n}\cdot(\boldsymbol{n}\cdot\nabla)\boldsymbol{u}\}, \tag{2.22}$$

corresponding, respectively, to the vector and scalar irreducible representations of the symmetry group $O(2)$.

Since the boundary conditions only hold on the solid walls $y = \pm 1$, each projection of the relation (2.20) is integrated over rectangular $(2+1)$-dimensional domains $\varOmega_k$ of size $H_x \times H_z \times H_t$ confined to one of the walls. Correspondingly, the weight functions $\boldsymbol{w}_j$ are constructed as products of three one-dimensional functions $\tilde{w}(s)$ where $s = \bar{x}$, $\bar{z}$ or $\bar{t}$. Note that the derivatives of the data with respect to the wall-normal (y) coordinate cannot be eliminated using integration by parts in this case; instead, we evaluate them directly using finite differences, although other alternatives could be used as well. For all noise levels up to the maximum of 50 %, valid single-term boundary conditions were always identified. Specifically, for the normal component, the relation $\boldsymbol{u}\cdot\boldsymbol{n} = 0$ is identified. For the tangential component, the relation $P_\parallel\boldsymbol{u} = 0$ is identified. These can be combined into the algebraically 'simplest' boundary condition

$$\boldsymbol{u} = 0, \tag{2.23}$$

which is the well-known no-slip boundary condition.

At sufficiently low noise levels, both the correct governing equations and the boundary conditions can be identified using only a single integration domain in the bulk and its projection onto the boundary, provided sufficiently many different weight functions are used. In particular, the no-slip boundary condition $\boldsymbol{u} = 0$ and the incompressibility condition (2.15) are always correctly identified.

3. Discussion

Physical constraints – the first key ingredient of SPIDER – play an essential role in the equation inference approach described here. The procedure used to construct the libraries of terms crucially relies on the irreducible representations of the symmetry

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

group describing the physical problem. For the bulk equations, it is the orthogonal group $O(3)$ describing rotations and reflections. In particular, the libraries $\mathcal{L}_0$ and $\mathcal{L}_1$ represent two of the irreducible representations of $O(3)$ corresponding to tensors of rank 0 and 1. Other irreducible representations of $O(3)$ can be used to identify additional physical relations. For instance, the vorticity equation would require a library of antisymmetric rank-2 tensors, which are isomorphic to pseudovectors by contraction with ε_{ijk}. For the boundary conditions, the symmetry group is $O(2)$, representing rotations around the surface normal $\boldsymbol{n}$ and in-plane reflections; $\mathcal{L}_\parallel$ and $\mathcal{L}_\perp$ correspond to two different irreducible representations of $O(2)$, vectors and scalars, respectively.

Fluid flows have additional symmetries: translational invariance in space and time is responsible for all of the coefficients being constant. Furthermore, all governing equations should have Galilean invariance. We could have imposed this symmetry as a constraint from the start when constructing the libraries, which would have reduced the size of both scalar and vector libraries even further, as discussed in § 2.1. Instead, we have let the data uncover this symmetry for us: inspection of the coefficients shows that both identified equations involving temporal derivatives acquire an explicitly Galilean-invariant form

$$\left.\begin{aligned}[\partial_t + \boldsymbol{u}\boldsymbol{\cdot}\nabla]\boldsymbol{u} + \nabla p - \nu_1\nabla^2\boldsymbol{u} = 0,\\ [\partial_t + \boldsymbol{u}\boldsymbol{\cdot}\nabla]E + \boldsymbol{u}\boldsymbol{\cdot}\nabla p - \nu_2\nabla^2 E + \nu_3\nabla\boldsymbol{u} : \nabla\boldsymbol{u} = 0,\end{aligned}\right\} \tag{3.1}$$

with some positive coefficients ν_i after the incompressibility condition is applied.

Physics also dictates that all data not only transform in a particular manner under various symmetries – pressure as a scalar and velocity as a vector – but have appropriate dimensions or units. There is no need to explicitly enforce dimensional homogeneity of all the terms in the relations (2.1); this is accomplished by properly non-dimensionalizing the terms f_n and treating the coefficients c_n as dimensionless constants. However, the physical units determine the scales S_i that play an essential role in non-dimensionalization. This step is absolutely critical to the success of sparse regression, as the magnitudes of the coefficients c_n are only meaningful once the terms f_n have been non-dimensionalized using proper scales. In particular, it would be entirely unclear whether any single-term relation, such as the incompressibility condition, is appropriate without a proper scale to compare it with.

The weak formulation – the second key ingredient of our approach – imparts SPIDER with unprecedented robustness, allowing it identify correct physical relations from data with extreme levels of noise, making it indispensable for analysing experimental data. Weak formulation also allows SPIDER to identify extremely subtle physical effects such as viscous stresses near the midplane of the flow where velocity gradients are small. Note that all of the coefficients in the energy and momentum balance equations (3.1) are very close to their true values of either unity or the viscosity $\nu_0 = 5 \times 10^{-5}$ used in the numerical simulations. Table 4 shows the deviation of the learned viscosity coefficients ν_i from the actual value. In particular, the viscosity ν_1 appearing in the momentum equation is identified with remarkable precision, especially near the boundary, where the velocity gradients are large. On the contrary, the values of the viscosity ν_2 and ν_3, which appear in the energy equation, are substantially less accurate, which reflects the manner in which the energy dissipation term $\nabla\boldsymbol{u} : \nabla\boldsymbol{u}$ is computed. There is no way to move all of the derivatives contained in this term onto the weight function, so these derivatives must be calculated numerically. In this work, we used central finite differencing for all first-order derivatives that cannot be eliminated, incurring a substantial error quadratic in the grid spacing. To obtain ν_2 and ν_3 with higher accuracy from noiseless data, a higher-order differentiation scheme could, in principle, be employed.

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

	ν_1/ν_0	s_1/ν_0	ν_2/ν_0	s_2/ν_0	ν_3/ν_0	s_3/ν_0
edge	0.999992	6×10^{-6}	1.04	0.01	1.04	0.01
centre	1.0006	4×10^{-4}	0.988	0.004	0.972	0.04

Table 4. The mean values ν_i and uncertainties s_i of the coefficients corresponding to viscosity in (3.1), all normalized by the true viscosity $\nu_0 = 5 \times 10^{-5}$ of the dataset. The uncertainty is estimated by rerunning the regression $M/2 = 128$ times using a random sample of only half of the integration domains and taking the sample standard deviation of the resulting coefficient vectors.

Sparse regression is the third key ingredient of SPIDER, and our regression algorithm based on singular value decomposition of the feature matrix has several advantages compared with SINDy and its various alternatives. First of all, as mentioned previously, the important dominant balances can be identified by inspecting the smallest singular values even without performing sparsification. Furthermore, the magnitude of the smallest singular value can be used to determine, again without performing sparsification, whether the corresponding library contains any meaningful relations describing the data and, hence, whether it needs to be expanded.

For the vector library (2.3), very different dominant balances are found in the two sampled regions, as the magnitudes $\chi_n = \|c_n \boldsymbol{q}_n\|$ of different terms listed in table 1 illustrate. Near the boundary, all four terms in the Navier–Stokes equation are of comparable magnitude, so it is not surprising that the same relation is identified for all listed noise levels. For data from the middle of the channel, the dominant balance involves only the terms $\partial_t \boldsymbol{u}$ and $\boldsymbol{u} \cdot \nabla \boldsymbol{u}$. In comparison, the term ∇p is smaller by more than an order of magnitude, and the viscous term is smaller by more than four orders of magnitude. (That such a small viscous term can be identified – at noise levels that are as large as 15 % – is due mainly to the exceptional robustness of the weak formulation.) These large differences in the magnitudes of different terms explain the order in which the inviscid Burgers equation, the Euler equation and the Navier–Stokes equation are identified as the noise magnitude is decreased. All three equations accurately describe the flow in the midplane of the channel and all three equations belong to the Pareto-optimal set generated by our greedy regression algorithm. Our choice of stopping criterion is one, but far from the only, way to choose between these three equations.

Let us comment on one unexpected result pointed out previously. As figure 3(*b*) illustrates, for noiseless data from the middle of the channel, SPIDER fairly consistently identifies a spurious term α in the pressure equation

$$\nabla^2 p + \nabla \cdot [(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}] + \alpha = 0. \tag{3.2}$$

Most commonly, this term is a small constant, as shown in table 3, and its magnitude $\chi_n = O(10^{-3})$ is much less than unity. Including this extra term decreases the residual r by a factor between 1.2 to 2 depending on the sample of integration domains, which does not always surpass the threshold $\gamma = 1.3$ used in the greedy algorithm. However, (3.2) is consistently identified as the most accurate three-term relation for four of the five data subsets described in Appendix A. For one subset (centre5), the spurious term is instead consistently identified as a multiple of E, with a similarly small coefficient c_n and magnitude χ_n. Including either term does not produce a noticeable improvement in the residual for noiseless data from the edge of the channel (edge1), where the residual decreases by a much smaller factor of 1.02. Oversmoothing by the weak formulation can, in principle, lead to inaccuracy in inferred relations. However, we find

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

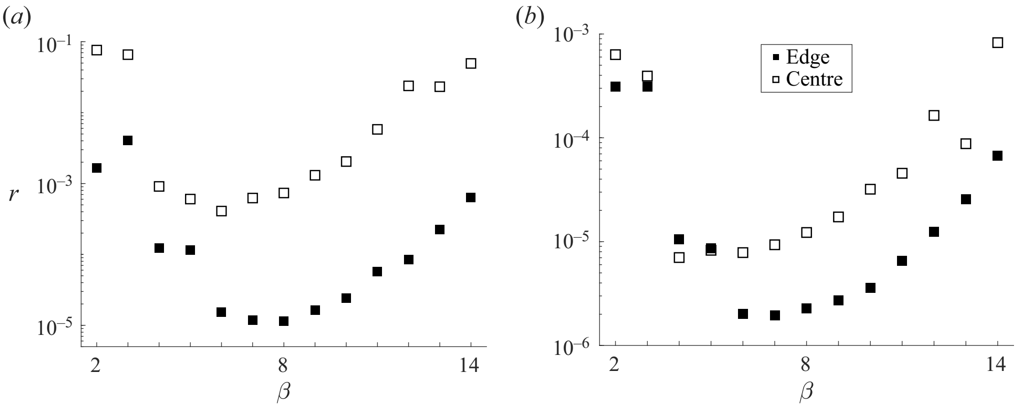

Figure 5. Dependence of residuals in (*a*) the pressure equation and (*b*) the momentum equation on the β hyperparameter. The solid (dashed) curves correspond to data near the middle (edge) of the channel.

substantial inaccuracy in the pressure-Poisson equation in strong form as well, as shown in Appendix C. The data presented there instead suggest that the presence of a spurious term likely reflects the limited resolution of the numerical solution in the middle of the channel, where the computational grid is the coarsest.

All the results presented here were obtained for the choice of the weight function exponent $\beta = 8$, which in figure 5 we find to roughly minimize the residuals of both the pressure-Poisson equation and the Navier–Stokes equation in both regions. Any choice in the range $6 \leq \beta \leq 10$ yields comparable residuals. For uniform grids, it is advantageous to use higher values of β (Gurevich *et al.* 2019); this improves the accuracy of the quadratures used in evaluating different library terms in weak from. The increase in the residual at higher values of β is due to non-uniformity (in the wall-normal direction) of the computational grid on which the data is available.

Finally, note that the approach presented here could be generalized to identify both governing equations and boundary conditions with parametric variation in space and/or time. Parametric variation can be easily detected by applying regression to subsets of data confined to small spatio-temporal volumes located at different positions. If the same functional relation is found but the coefficients differ, these variable coefficients could be replaced with a linear superposition of some basis functions and the regression repeated on the expanded library, as done by e.g. Rudy *et al.* (2019).

4. Conclusion

To summarize, we have shown that a combination of very general physical constraints, weak formulation of PDEs and sparse regression yields an extremely powerful model discovery tool, which we call SPIDER. It allows one to identify complete and easily interpretable quantitative mathematical models of continuum systems, such as the highly turbulent fluid flow considered here, from even very noisy data. Moreover, SPIDER provides information about the relative importance of different physical effects in various regimes represented in the data.

The utility of the approach presented here is not limited to fluid dynamics. This same approach can be used to identify mathematical models of numerous high-dimensional, nonlinear, non-equilibrium systems that have defied traditional first-principles modelling approaches. Some examples include high energy density plasmas, as found inside the

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

Subset	I_x	I_y	I_z	I_t
centre1	[1024,1088]	[256,320]	[750,814]	[2000,2064]
centre2	[1,65]	[256,320]	[750,814]	[1000,1064]
centre3	[1024,1088]	[256,320]	[750,814]	[500,564]
centre4	[1524,1588]	[256,320]	[850,914]	[500,564]
centre5	[1724,1788]	[256,320]	[850,914]	[500,564]
edge1	[1024,1088]	[1,65]	[768,832]	[2000,2065]

Table 5. The grid indices representing boundaries of the space–time domains from which integration domains were sampled.

stars and the interior of fusion energy devices, and excitable media such as cardiac or intestinal muscle tissue and biological neural networks. Other interesting applications include active matter systems such as animal herds, bird flocks, insect swarms, fish schools, bacterial aggregates, self-propelled particles and even collections of robots – these are formally discrete but may possess useful continuum models. Most active matter systems lack quantitative mathematical models while exhibiting interesting collective behaviours that could be better understood within the framework of such continuum 'hydrodynamic' models (Toner & Tu 1998). Initial data-driven efforts to construct such models have already been made (Messenger & Bortz 2022; Supekar *et al.* 2023).

Acknowledgements. The authors are grateful to C. Meneveau for his help with interpretation of the data in the Johns Hopkins turbulence database.

Funding. This material is based on work supported by the National Science Foundation under Grants Nos. CMMI-1725587 and CMMI-2028454 and the Air Force Office of Scientific Research under grant FA9550-19-1-0005. D.R.G. was also supported by the NSF Graduate Research Fellowship Program.

Declaration of interests. The authors report no conflict of interest.

Author ORCIDs.

Daniel R. Gurevich https://orcid.org/0000-0002-3659-407X;

Roman O. Grigoriev https://orcid.org/0000-0001-6220-4701.

Appendix A. The sampled data locations

The data used by SPIDER were obtained from the web cutout service of the Johns Hopkins turbulence database. The grid indices of the space–time regions of the data are summarized in table 5. Data subsets centre1 and edge1 were used to generate all the reported results, while subsets centre2 through centre5 where used to investigate spurious terms in the pressure equation (3.2).

Appendix B. The impact of noise correlation and data sampling

In this section we discuss how the accuracy of our results is affected by varying several hyperparameters in the model discovery process. The momentum equation (2.16) will be used to demonstrate the scaling of the relative error in the coefficients, $\epsilon_n = |c_n^{inf} - c_n^{the}|/|c_n^{the}|$, where the superscript refers to the inferred (*inf*) or theoretical values (*the*). To facilitate comparison, the coefficients are scaled such that $\|\boldsymbol{c}\| = 1$ in both cases.

Correlated noise with spatial correlation length $\ell_c = k_c^{-1}$ is generated by applying an inverse Fourier transform to a source $F(\boldsymbol{k})$, where $F(\boldsymbol{k}) = F^*(-\boldsymbol{k})$ is drawn randomly

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

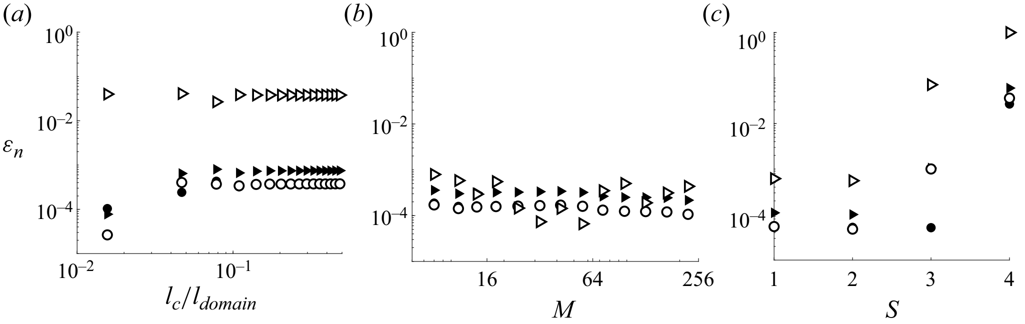

Figure 6. The relative error ε_n in the coefficients of the Navier–Stokes equation. (*a*) Dependence of ε_n on the spatial correlation length ℓ_c. (*b*) Dependence of ε_n on the number of integration domains M. (*c*) Dependence of ε_n on the subsampling factor S. In all panels, the solid (hollow) triangles correspond to the pressure gradient ∇p (viscous dissipation $\nabla^2 \boldsymbol{u}$) and the solid (hollow) circles correspond to advection $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ (time derivative $\partial_t \boldsymbol{u}$).

from a uniform distribution with zero mean for wavenumbers $k \geq k_c$ and $F(\boldsymbol{k}) = 0$ for $k < k_c$. This noise is then rescaled so that the standard deviation is unity, and independently drawn noise is added to each hydrodynamic field, weighted by the standard deviations of the respective fields. The impact of spatial correlation on the coefficients can be seen in figure 6(*a*), which shows the results computed for noise with 10 % amplitude. As expected, increasing spatial correlation leads to a (slight) decrease in the accuracy of the coefficients, with ε_n saturating for all four coefficients when ℓ_c exceeds 10 % of the size ℓ_{domain} of the integration domain. It is worth noting that weak-form regression was also found to easily handle noise characteristic of particle image velocimetry in Reinbold *et al.* (2021) and Golden *et al.* (2023).

Note that the scaling of the accuracy of weak-form regression with both the amount of available data and the resolution of the data has been investigated both empirically and theoretically in the context of the Kuramoto–Sivashinsky equation by Gurevich *et al.* (2019). We repeat the scaling analysis for the channel flow data here. To determine how the accuracy ε_n varies with the amount of available data, regression was performed by constructing a reference $\boldsymbol{Q}$ matrix with 768 rows (integration domains) and using only the first $3M$ of these rows. The results for noiseless data are shown in figure 6(*b*). Note that increasing M does not bring meaningful improvement in the accuracy of the coefficients, which reflects that, in the noiseless case, the magnitude of ε_n is mainly controlled by the resolution and/or accuracy of the data.

To determine how the accuracy ε_n is affected by the spatial and temporal resolution of the data, the gridded hydrodynamic fields were subsampled by a constant factor S. For a subsampling factor S, every Sth gridpoint (in every direction) is used in the calculation of weak-form integrals. The physical size of integration domains was held constant, so $O((32/S)^4)$ points were used to approximate the integrals for each S considered. The results are shown in figure 6(*c*). Our results suggest that velocity data are reasonably well resolved in space and time as the accuracy for $S = 2$ is the same as that for the computational grid ($S = 1$). Decreasing the resolution further ($S \geq 3$) leads to a substantial decrease in the accuracy, especially for the coefficient c_4 representing viscosity. For $S = 4$, the Euler equation is selected instead of Navier–Stokes, which corresponds to $c_4 = 0$ and $\varepsilon_4 = 1$.

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

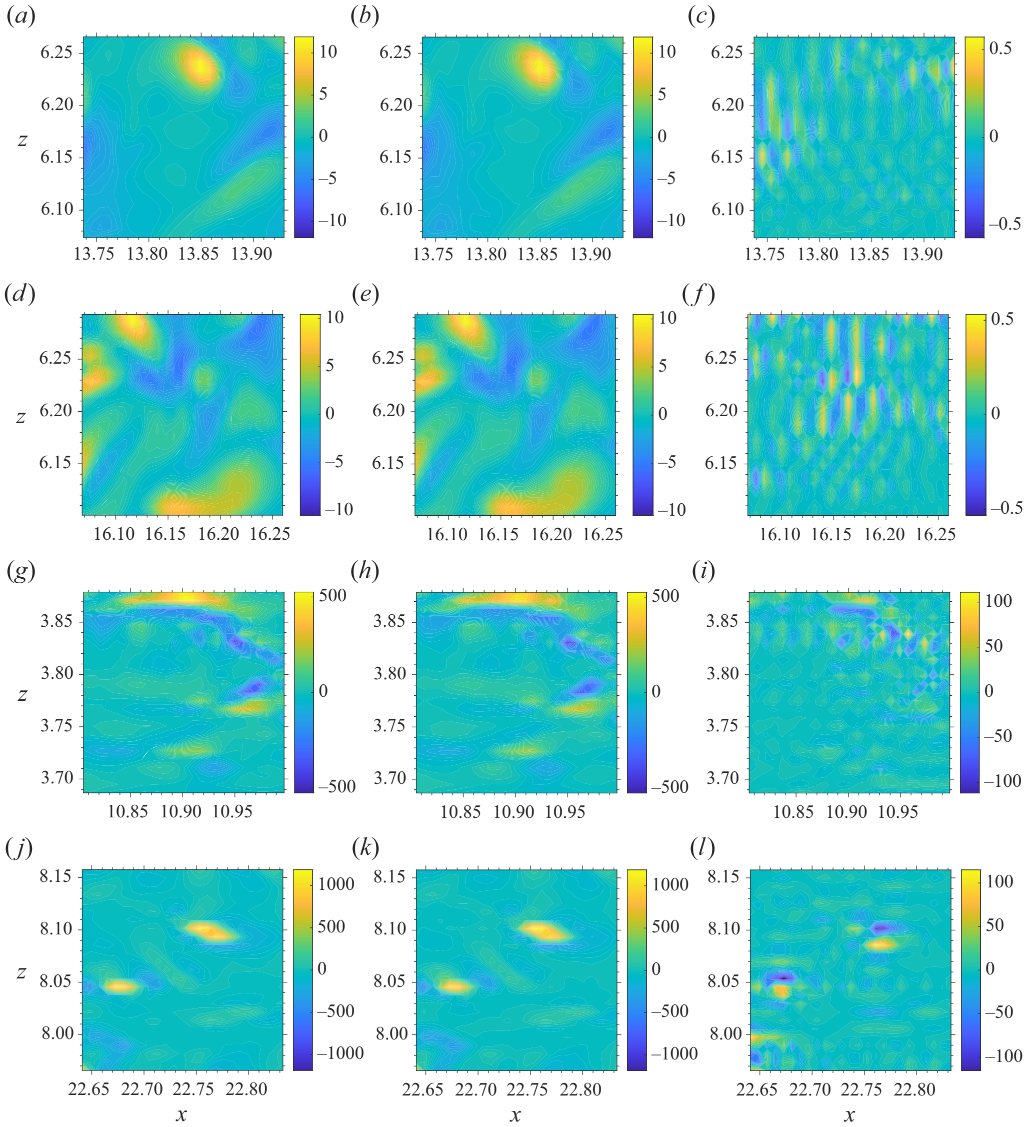

Figure 7. Individual terms in the pressure-Poisson equation and their sum at a random collection of spatial locations. The first column shows $-\nabla \cdot [(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}]$, the second column shows $\nabla^2 p$ and the third column shows the sum $\nabla^2 p + \nabla \cdot [(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}]$. (*a–c*) Correspond to $y = -0.149$ and $t = 17.084$, (*d–f*) correspond to $y = 0.089$ and $t = 11.987$, (*g–i*) correspond to $y = -0.982$ and $t = 20.543$, (*j–l*) correspond to $y = -0.989$ and $t = 12.452$. The colour bar for the sum is centred at its spatial mean, which is roughly two orders of magnitude smaller than the maximum of the sum.

Appendix C. The accuracy of the pressure field data

Spurious terms frequently identified by SPIDER in the pressure equation (3.2) raise a question about the accuracy and resolution of the pressure data contained in the channel flow database. To quantify the typical accuracy to which the pressure-Poisson equation is satisfied, we used the provided MATLAB script that computes, in random x–z planes, the terms $\nabla^2 p$, $\nabla \cdot [(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}]$ as well as their sum (or residual), $\nabla^2 p + \nabla \cdot [(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}]$, which should vanish for an exact solution. For derivatives evaluated using finite differences, the

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

residual of the pressure-Poisson equation is found to be in the range of 5 %–20 % of the magnitude of either term. Figure 7 shows some representative examples both in the middle of the channel (*a–f*) and near the bottom boundary (*g–l*). Each plot corresponds to a region of 25 × 25 grid points, which is comparable to the size of our integration domains. The spatial structure of the residual suggests that the pressure solution is not fully resolved, which is entirely consistent with the results of SPIDER. The magnitude of the spurious term identified by SPIDER is two orders of magnitude smaller than the maximal size of the residual of the pressure-Poisson equation but is comparable to the mean of the residual, providing further evidence that the spurious terms reflect the limited resolution of the pressure data. As indicated in the README documents accompanying the data, the simulations used spectral differentiation. However, for the finite grid resolution used in the simulations and for storing the data, evaluation of equation terms that use different formulations for the differentiations (e.g. finite differencing or weak formulations as used here) is expected to lead to non-negligible errors, especially for terms involving higher-order derivatives such as the pressure Laplacian.

References

Alves, E.P. & Fiuza, F. 2022 Data-driven discovery of reduced plasma physics models from fully kinetic simulations. *Phys. Rev. Res.* **4** (3), 033192.

Bär, M., Hegger, R. & Kantz, H. 1999 Fitting partial differential equations to space-time dynamics. *Phys. Rev.* E **59** (1), 337.

Biggio, L., Bendinelli, T., Neitz, A., Lucchi, A. & Parascandolo, G. 2021 Neural symbolic regression that scales. In *International Conference on Machine Learning*, pp. 936–945. PMLR.

Bongard, J. & Lipson, H. 2007 Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl Acad. Sci.* **104** (24), 9943–9948.

Brunton, S.L., Proctor, J.L. & Kutz, J.N. 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci.* **113** (15), 3932–3937.

Crutchfield, J.P. & McNamara, B.S. 1987 Equation of motion from a data series. *Complex Syst.* **1** (3), 417–452.

Ferreira, C. 2001 Gene expression programming: a new adaptive algorithm for solving problems. Preprint, arXiv:cs/0102027.

Golden, M., Grigoriev, R.O., Nambisan, J. & Fernandez-Nieves, A. 2023 Physically informed data-driven modeling of active nematics. *Sci. Adv.* **9** (27), eabq6120.

Gurevich, D.R., Reinbold, P.A.K. & Grigoriev, R.O. 2019 Robust and optimal sparse regression for nonlinear PDE models. *Chaos* **29** (10), 103113.

Joshi, C., Ray, S., Lemma, L.M., Varghese, M., Sharp, G., Dogic, Z., Baskaran, A. & Hagan, M.F. 2022 Data-driven discovery of active nematic hydrodynamics. *Phys. Rev. Lett.* **129** (25), 258001.

Kaheman, K., Kutz, J.N. & Brunton, S.L. 2020 SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proc. R. Soc.* A **476** (2242), 20200279.

Karpatne, A., Atluri, G., Faghmous, J.H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N. & Kumar, V. 2017 Theory-guided data science: a new paradigm for scientific discovery from data. *IEEE Trans. Knowl. Data Engng* **29** (10), 2318–2331.

Long, Z., Lu, Y., Ma, X. & Dong, B. 2018 PDE-Net: learning PDEs from data. In *International Conference on Machine Learning*, pp. 3208–3216. PMLR.

Ma, W. & Zhang, J. 2022 Dimensional homogeneity constrained gene expression programming for discovering governing equations from noisy and scarce data. Preprint, arXiv:2211.09679.

Mangan, N.M., Brunton, S.L., Proctor, J.L. & Kutz, J.N. 2016 Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Trans. Mol. Biol. Multi-Scale Commun.* **2** (1), 52–63.

Martius, G. & Lampert, C.H. 2016 Extrapolation and learning equations. Preprint, arXiv:1610.02995.

Messenger, D.A. & Bortz, D.M. 2021 Weak sindy for partial differential equations. *J. Comput. Phys.* **443**, 110525.

Messenger, D.A. & Bortz, D.M. 2022 Learning mean-field equations from particle data using WSINDy. *Phys.* D*: Nonlinear Phenom.* **439**, 133406.

Miettinen, K. 2012 *Nonlinear Multiobjective Optimization*, vol. 12. Springer Science & Business Media.

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press

RAISSI, M. & KARNIADAKIS, G.E. 2018 Hidden physics models: machine learning of nonlinear partial differential equations. *J. Comput. Phys.* **357**, 125–141.

RAISSI, M., PERDIKARIS, P. & KARNIADAKIS, G.E. 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707.

REINBOLD, P.A.K. & GRIGORIEV, R.O. 2019 Data-driven discovery of partial differential equation models with latent variables. *Phys. Rev.* E **100** (2), 022219.

REINBOLD, P.A.K., GUREVICH, D.R. & GRIGORIEV, R.O. 2020 Using noisy or incomplete data to discover models of spatiotemporal dynamics. *Phys. Rev.* E **101** (1), 010203.

REINBOLD, P.A.K., KAGEORGE, L.M., SCHATZ, M.F. & GRIGORIEV, R.O. 2021 Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression. *Nat. Commun.* **12** (1), 1–8.

RUDY, S., ALLA, A., BRUNTON, S.L. & KUTZ, J.N. 2019 Data-driven identification of parametric partial differential equations. *SIAM J. Appl. Dyn. Syst.* **18** (2), 643–660.

RUDY, S.H., BRUNTON, S.L., PROCTOR, J.L. & KUTZ, J.N. 2017 Data-driven discovery of partial differential equations. *Sci. Adv.* **3** (4), e1602614.

SAHOO, S., LAMPERT, C. & MARTIUS, G. 2018 Learning equations for extrapolation and control. In *International Conference on Machine Learning*, pp. 4442–4450. PMLR.

SCHAEFFER, H. 2017 Learning partial differential equations via data discovery and sparse optimization. *Proc. R. Soc.* A*: Math. Phys. Engng Sci.* **473** (2197), 20160446.

SCHAEFFER, H., TRAN, G. & WARD, R. 2018 Extracting sparse high-dimensional dynamics from limited data. *SIAM J. Appl. Maths* **78** (6), 3279–3295.

SCHMIDT, M. & LIPSON, H. 2009 Distilling free-form natural laws from experimental data. *Science* **324** (5923), 81–85.

STEPHANY, R. & EARLS, C. 2022 PDE-LEARN: using deep learning to discover partial differential equations from noisy, limited data. Preprint, arXiv:2212.04971.

STEPHANY, R. & EARLS, C. 2023 Weak-PDE-LEARN: a weak form based approach to discovering PDEs from noisy, limited data. Preprint, arXiv:2309.04699.

SUPEKAR, R., SONG, B., HASTEWELL, A., CHOI, G.P.T., MIETKE, A. & DUNKEL, J. 2023 Learning hydrodynamic equations for active matter from particle simulations and experiments. *Proc. Natl Acad. Sci.* **120** (7), e2206994120.

TONER, J. & TU, Y. 1998 Flocks, herds, and schools: a quantitative theory of flocking. *Phys. Rev.* E **58** (4), 4828.

XING, H., ZHANG, J., MA, W. & WEN, D. 2022 Using gene expression programming to discover macroscopic governing equations hidden in the data of molecular simulations. *Phys. Fluids* **34** (5), 057109.

XU, D. & KHANMOHAMADI, O. 2008 Spatiotemporal system reconstruction using Fourier spectral operators and structure selection techniques. *Chaos* **18** (4), 043122.

https://doi.org/10.1017/jfm.2024.813 Published online by Cambridge University Press