J. Fluid Mech. (2023), *vol.* 975, A41, doi:10.1017/jfm.2023.868

Physics-agnostic and physics-infused machine learning for thin films flows: modelling, and predictions from small data

Cristina P. Martin-Linares[1], Yorgos M. Psarellis[2], George Karapetsas[3], Eleni D. Koronaki[4,5,†] and Ioannis G. Kevrekidis[2,†]

[1]Department of Mechanical Engineering,Whiting School of Engineering, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA

[2]Department of Chemical and Biomolecular Engineering,Whiting School of Engineering, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA

[3]Department of Chemical Engineering, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece

[4]Interdisciplinary Center for Security, Reliability and Trust, University of Luxembourg, 29 John F. Kennedy Avenue, Luxembourg 1855, Luxembourg

[5]School of Chemical Engineering, National Technical University of Athens, Athens 15780, Greece

(Received 1 March 2023; revised 16 September 2023; accepted 9 October 2023)

Numerical simulations of multiphase flows are crucial in numerous engineering applications, but are often limited by the computationally demanding solution of the Navier–Stokes (NS) equations. The development of surrogate models relies on involved algebra and several assumptions. Here, we present a data-driven workflow where a handful of detailed NS simulation data are leveraged into a reduced-order model for a prototypical vertically falling liquid film. We develop a physics-agnostic model for the film thickness, achieving a far better agreement with the NS solutions than the asymptotic Kuramoto–Sivashinsky (KS) equation. We also develop two variants of physics-infused models providing a form of calibration of a low-fidelity model (i.e. the KS) against a few high-fidelity NS data. Finally, predictive models for missing data are developed, for either the amplitude, or the full-field velocity and even the flow parameter from partial information. This is achieved with the so-called 'gappy diffusion maps', which we compare favourably to its linear counterpart, gappy POD.

Key words: thin films, computational methods, machine learning

† Email addresses for correspondence: eleni.koronaki@uni.lu, yannisk@jhu.edu

1. Introduction

The study of multiphase flows is often limited by the computational effort involved in solving the Navier–Stokes (NS) equations (Glasser, Kevrekidis & Sundaresan 1997). One such example, the flow of thin films of liquid on inclined planes, has fascinated researchers not only because of the wide range of industrial applications but also because of the interesting dynamics of the liquid–air interface (Kalliadasis *et al.* 2012). The NS equations accurately describe the fluid motion and also the evolution of the surface but suffer from a high computational cost (Pettas *et al.* 2019*a*,*b*). To this end, significant effort has led to several approximate interface evolution equations that are much simpler to solve but are nevertheless valid under specific assumptions and limitations. Beyond their limits of validity, it is often found that they yield non-physical solutions, or even blow up (Kalliadasis *et al.* 2012), posing significant restrictions to their applicability. In more recent studies, weighted integral boundary layer models are found to compare favourably to NS simulations for moderate Reynolds numbers and even three-dimensional flows (Dietze *et al.* 2014). These surrogate models are very economical and are included even in open source codes (Rohlfs, Rietz & Scheid 2018), showing remarkable accuracy even in flow regimes exhibiting highly nonlinear waves (Chakraborty *et al.* 2014). Alternatives to models that solve partial differential equations (PDEs) are also available that compute the film height by numerical integration based on the aggregation of droplets (Bharadwaj *et al.* 2022).

The evolution of successful surrogate models has taken place over the past decades and relies on significant subject matter expertise to form meaningful assumptions and involved algebra. Our goal here is to develop novel, robust and efficient data-driven/data-assisted models, not restricted to applications of thin film flow, that combine physical and mathematical insight with machine learning strategies, in order to drastically enable computational fluid dynamics and break new barriers in flow control, uncertainty quantification and shape optimization.

The implementation of data-driven methods is gaining popularity in diverse fluid-related applications. Recently, nonlinear manifold learning and specifically isomap (Balasubramanian & Schwartz 2002) was implemented to derive a low-dimensional description of an ensemble of flow field snapshots as an alternative to proper orthogonal decomposition (POD), ultimately aiming at developing an economical surrogate (Farzamnik *et al.* 2023). Deep learning has been proposed in order to produce closures of large-eddy simulations (LES) and Reynolds-averaged NS models, by adding neural network-derived terms into the governing equations (Sirignano & MacArt 2023). Data-driven viscous damping terms have been estimated for sloshing a rectangular tank (Miliaiev & Timokha 2023), where machine learning is implemented in conjunction with an appropriately formulated loss function that relies on experimental measurements. Machine learning is implemented to derive multifidelity models (Rezaeiravesh, Mukha & Schlatter 2023), and an ensemble of neural networks is applied to develop a wall model for LES based on the assumption that the flow can be thought of as a combination of blocks (Lozano-Durán & Bae 2023). Flow control is also a popular field where machine learning is implemented (Sonoda *et al.* 2023; Zhang, Fan & Zhou 2023). Recently, artificial neural network–based nonlinear algebraic models were presented for the LES of compressible wall-bounded turbulence (Xu *et al.* 2023). A model based on a convolutional neural network is proposed so as to reconstruct the three-dimensional turbulent flows beneath a free surface using surface measurements, including the surface elevation and surface velocity (Xuan & Shen 2023). Also, data-driven modelling has recently been proposed in the context of unsteady wall pressure fields (Meloni *et al.* 2023).

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

This work presents a methodology for deriving data-driven PDEs for the film amplitude, based on a collection of NS simulation data that are not subject to restrictions and assumptions for the flow and, hence, are more general. Our work falls in the category of dynamical system identification (Rico-Martínez *et al.* 1992; Krischer *et al.* 1993; González-García, Rico-Martínez & Kevrekidis 1998; Kemeth *et al.* 2022). Recently increased interest in PDE identification has led to the development of alternative algorithmic tools, such as sparse identification of nonlinear dynamical systems using dictionaries (Brunton, Proctor & Kutz 2016; Rudy *et al.* 2017), PDE-net (Long *et al.* 2018), physics-informed neural networks (Raissi, Perdikaris & Karniadakis 2019) and others (Chen *et al.* 2018; Vlachas *et al.* 2018, 2022). Our algorithmic approach can be implemented on data from detailed PDE simulations (Psarellis *et al.* 2022), agent-based modelling (Arbabi & Kevrekidis 2021; Lee *et al.* 2022) or lattice Boltzmann simulations (Lee *et al.* 2020; Galaris *et al.* 2022), among others. Extensions of PDE identification including grey box or closure identification (such as those explored in our work) have been studied in the context of various applications (Parish & Duraisamy 2016; Pan & Duraisamy 2018; Duraisamy, Iaccarino & Xiao 2019; Lee *et al.* 2019; Lee *et al.* 2022; Psarellis *et al.* 2022; Kemeth *et al.* 2023). In the relevant literature, the Kuramoto–Sivashinsky (KS) equation, selected in this work as a low-fidelity counterpart of the NS equations, has served as a benchmark case study, due to its wealth of dynamic responses and highly nonlinear nature (González-García *et al.* 1998; Raissi & Karniadakis 2017; Vlachas *et al.* 2018; Floryan & Graham 2022; Linot *et al.* 2023).

The results of the learned PDE are compared with the ground-truth NS results and also the results of the KS equation. It is expected that past a certain limit, the KS equation will perform poorly and produce non-physical solutions. Yet it is still useful in the context of learning an accurate amplitude PDE, as will be shown, in two different ways. In the first ‘grey box’ model approach, an additive correction of the KS equation can be learned from NS data as a sort of calibration of the low-fidelity model against high-fidelity data. In this context, the data-driven model provides a measure of the discrepancy between the approximate equation and the ground truth, and serves to inform as to the actual limits of applicability of the KS equation in terms of the flow parameter, here the Reynolds number. In the second approach, which we call the ‘functional correction grey box model’, certain observations of the KS formula, such as the value of its right-hand side, its derivatives or even values in specific nearby time instances or nearby points in space, are used as inputs to the learned model.

In addition, reduced representations of the NS data, including full velocity fields and fluid film height, are further exploited for out-of-sample predictions from partial data at the small data limit. Nonlinear manifold learning, specifically diffusion maps (DMAPs) and linear methods, i.e. POD, are initially implemented in order to derive a low-order description of the high-dimensional dataset. It is then shown that efficient interpolation in the reduced space can help recover entire sets of data from partial information. Specifically, it will be demonstrated that full velocity profiles, parameter values and film height measurements can be predicted given a handful of values for the film height at specific locations. The advantages of ‘gappy’ diffusion maps over its linear counterpart, gappy POD, are discussed in relation to the parsimony of the description of the manifold that contains the data and to the location of the known measurements.

2. Governing equations: ground truth and surrogate model

The flow of a liquid film on an inclined plane (see figure 1) is described, in two dimensions, by conservation equations for mass and momentum, written in dimensionless

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

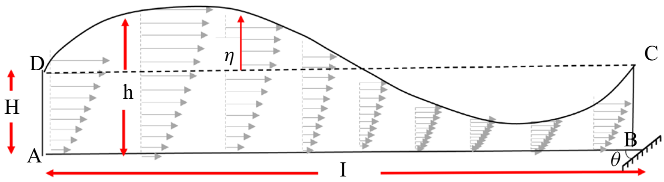

Figure 1. Cross-section of a film flowing on a plane, inclined with respect to the horizontal by angle θ. Here $H = 1$ is the dimensionless Nusselt film height.

form as

$$\left.\begin{aligned} \nabla \cdot \boldsymbol{u} &= 0, \\ R\frac{\partial \boldsymbol{u}}{\partial t} &= -R(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) - \nabla P + \nabla^2\boldsymbol{u} + F\boldsymbol{g}, \end{aligned}\right\} \tag{2.1}$$

where $\boldsymbol{u} = (u_x, u_y)^{\mathrm{T}}$ and P are the dimensionless velocity vector and pressure, respectively, and $\nabla = (\partial_x, \partial_y)$ denotes the gradient operator for Cartesian coordinates. We also define the unit gravity vector $\boldsymbol{g} = (\sin\theta, -\cos\theta)^{\mathrm{T}}$. Using the characteristic Nusselt scales for the velocity vector and all lengths,

$$U_N = \left(\frac{Q^2\rho g \sin(\theta)}{3\mu}\right)^{1/3}, \quad H_N = \left(\frac{3\mu Q}{\rho g \sin(\theta)}\right)^{1/3}, \tag{2.2a,b}$$

the dimensionless groups that emerge are the Reynolds number R, the Weber number W and the Stokes number F, defined as

$$R = \frac{\rho Q}{\mu}, \quad W = \frac{\sigma}{\rho g H_N^2}, \quad F = \frac{3}{\sin\theta}. \tag{2.3a–c}$$

Here, ρ, μ and σ are the liquid density, the viscosity and the liquid–air surface tension, respectively, while Q denotes the volumetric flow rate per unit length normal to the cross-section.

Along the liquid–air interface, a normal stress balance between capillary force and viscous stress is applied,

$$\boldsymbol{n}\cdot\boldsymbol{T} = WF\kappa\boldsymbol{n}. \tag{2.4}$$

Here, the total stress tensor is defined as $\boldsymbol{T} = -P\boldsymbol{I} + (\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{\mathrm{T}})$, $\boldsymbol{n}$ is the unit vector normal to the interface, outward with respect to the film. Note that in (2.4) the ambient pressure has been set equal to zero (datum pressure) without loss of generality. The mean curvature is $\kappa = -\nabla_s\cdot\boldsymbol{n}$, with $\nabla_s = (\boldsymbol{I} - \boldsymbol{nn})\cdot\nabla$. The rest of the boundary conditions include the no-slip condition at the liquid–solid plane interface (denoted as AB in figure 1) and periodic boundary conditions at the lateral domain boundaries (AD and BC in figure 1).

The kinematic boundary condition, which specifies that the velocity of the interface that is normal to the boundary is equal to the velocity of the fluid that is normal to the boundary, ensuring no mass transfer through the interface, completes the set of governing equations

$$\boldsymbol{n}\cdot(h_t\boldsymbol{e}_y) = \boldsymbol{n}\cdot\boldsymbol{u}. \tag{2.5}$$

The NS data, which consist our ground truth, are derived by solving the time-dependent equations in an Eulerian frame, discretized with the finite element method, using

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

COMSOL Multiphysics®. We consider the flow over a vertical plane, schematically presented in figure 1, i.e. $\theta = 90^\circ$ with a domain of dimensionless length, $l = L/H_N = 95$; L denotes the dimensional length of the domain. The value of the Weber number used in the simulations is $W = 278$. We start our simulations from an initial condition corresponding to a flat film perturbed by a sinusoidal perturbation with amplitude $\sim$3 % of the dimensionless Nusselt film height, $H = 1$. The height of the film, $h(x, t)$, is collected at each time step until a steady travelling wave is formed.

2.1. *Surrogate model – the KS equation*

The single-equation surrogate of the amplitude, selected in this study, is the KS equation. Assuming the flow over a vertical plane, the KS equation can be written as

$$\left.\begin{aligned} \phi_\tau &= -\alpha(\phi\phi_\xi + \phi_{\xi\xi}) - 4\phi_{\xi\xi\xi\xi} \quad \text{for } \xi \in [0, 2\pi], \\ \alpha &= \frac{8}{5}\frac{RF}{\epsilon^2 W}. \end{aligned}\right\} \tag{2.6}$$

The detailed derivation of the KS equation from NS equations can be found in Shlang & Sivashinsky (1982), Chang (1986) and Brown (1992) and it is summed up here for completeness. It is based on the following assumptions.

(i) The film is thin, $\epsilon = 2\pi H_N/L \ll 1$.
(ii) The waves are long, $\partial h/\partial x \ll 1$.
(iii) The mean height of the film is much larger than the deviation from the mean.

Under these assumptions, it is possible to exploit the small parameter ϵ and employ a perturbation expansion for all dependent variables, i.e. velocities, pressure and interfacial height; e.g. the interfacial height is given by $h \approx 1 + \epsilon\eta + O(\epsilon^2)$, where η denotes the deviation from the mean film height. Restricting to the case of laminar flow with $R = O(1)$, $F = O(1)$ and $W = O(\epsilon^{-2})$ and neglecting higher-order terms, the kinematic condition (written in terms of the deviation, η) may be reduced to $\eta_t + F\eta_x = 0$, indicating that waves travel with speed F. Taking this into account, a new variable $\xi = \epsilon(x - Ft)$ can be introduced to obtain constant shape waves travelling with speed F. Moreover, the amplitude is rescaled according to $\phi = (15/RF)\eta$. Finally, since it is known that the wave amplitude varies on a slow time scale compared with the travelling motion, a change in the time variable is introduced, i.e. $\tau = (\epsilon^4 WF/12)t$. In the end, the KS equation (2.6) is obtained as a function of the new defined variables.

2.2. *The NS vs KS equation: bifurcation diagram*

The KS equation will be used extensively in the discussion that will follow mostly in terms of the derivation of variations of the data-driven model. Therefore, it would be useful to briefly discuss the limitations of the KS equation before proceeding with the presentation of our results. Even though it is well known that the KS equation is valid for Reynolds number values of O(1), to the best of our knowledge, there is no direct comparison in the literature of the KS equation to the NS equation results. To clearly present the limitations of the KS equation, here we plot in figure 2 the norm of the amplitude distribution, $\|\eta\|$, with respect to both the Reynolds number and the KS parameter α.

Both equations predict very similar results for $1 \leq R \leq 3.4$ (or $4 \leq \alpha \leq 13.43$). In this range of parameter values the solution of the KS equation is a stationary wave (in a co-moving frame with speed $c = 3$), whereas the NS equation, solved in an Eulerian frame

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

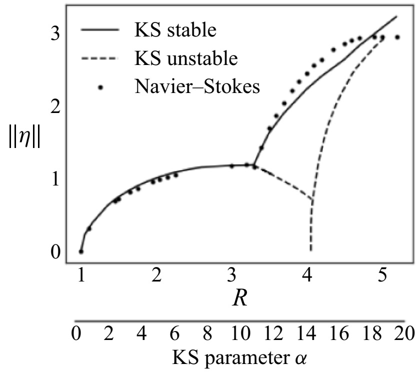

Figure 2. Bifurcation diagram of the NS and KS equations.

evolves into a travelling wave with a steady and unchanged shape and speed $c = 3$. Past that point ($R = 3.4$ and $\alpha = 13.43$), the speed of the wave becomes larger than 3 and, hence, the solution of the KS equation is also travelling (with speed $c - 3$). Up to approximately $R = 4.3$ ($\alpha = 16.5$) the KS solution gradually deviates from the NS solution. For higher values of α, i.e. $\alpha > 16.998$, the solution of the KS equation is a so-called 'pulsing' wave, as described in detail in Kevrekidis, Nicolaenko & Scovel (1990). The pulsing waves oscillate between two waveforms that are π periodic in space and are $\pi/2$ shifts of each other. Such a solution has not been reported for the NS equations.

3. Physics-agnostic and physics-infused data-driven models

3.1. *Black box model: learning the PDE*

Here the goal is to use data from the NS simulations to learn a PDE of the general form

$$\frac{\partial \eta}{\partial t} = f(\eta, \eta_x, \eta_{xx}, \eta_{xxx}, \eta_{xxxx}, R). \tag{3.1}$$

The function f is approximated by a feed forward neural network with two hidden layers with 15 neurons and a tanh activation function, implemented in Tensorflow (Abadi *et al.* 2015). The mean squared error is used as a loss function along with the Adam optimizer. The inputs to the neural network are the amplitude and four spatial derivatives of the amplitude, along with the R value, as extracted from the NS simulations. Specifically, the NS model is implemented for 20 parameter values, and snapshots, i.e. time instances of the film surface evolution are collected in equally sized time steps ($dt = 1$ is the dimensionless time unit). The spatial derivatives, up to fourth order are computed using Fourier transforms, in each point in space and time. The output of the neural network is the time derivative of the amplitude, which can be extracted directly from the NS code (although it can also be easily computed, e.g. with finite differences).

In total, 75 000 data points (pairs of local values of the inputs and output) per parameter value are used to train the neural network and once this is done, the right-hand side of the PDE in (3.1) can be used in conjunction with any method of integration in time, such as the Runge–Kutta method. The attractors that correspond to a steady travelling wave that resulted from the integration of the learned PDE are shown in figure 3 for a representative selection of parameter values. The attractor of the neural network-derived PDE, shown in black, is almost a perfect match with the ground-truth results of the NS (dashed line).

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

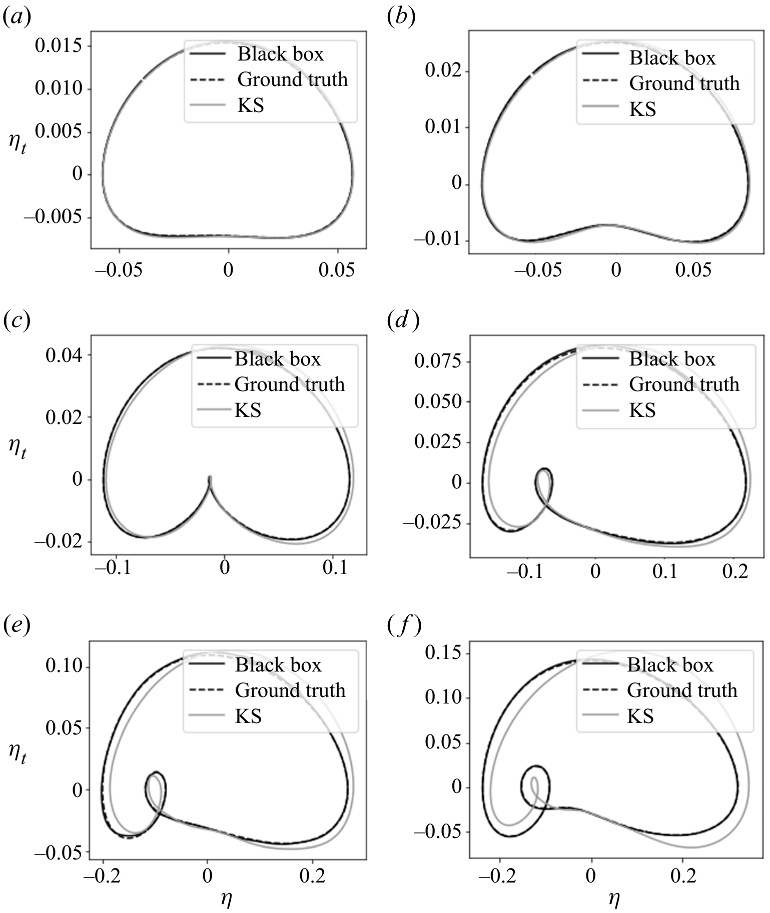

Figure 3. ‘Black box’ model performance. Attractors for different R numbers derived by (i) the NS equation (dashed line), (ii) the KS equation (grey) and (iii) the NN-derived PDE (black); all results are rescaled in the NS scaling. (*a*) $R = 1.45$, (*b*) $R = 1.95$, (*c*) $R = 3.30$, (*d*) $R = 3.70$, (*e*) $R = 4.20$ and (*f*) $R = 5.20$.

For reference and comparison, the KS results (appropriately rescaled according to the scaling factors detailed in Appendix A) are shown in the same plot (grey line). The KS equation performs well for small values of the R number ($R < 3.3$), but then progressively start to deviate quantitatively for increasing values of R. This is shown in figure 4, where a snapshot of the amplitude derived by the KS equation, the NS equations and the ‘black box’ model are shown alongside the corresponding phase portraits for $R = 1.95$ and $R = 4.2$. This figure also shows the error along time integration, which illustrates the fact that, on average, the small numerical error remains constant. These two R values are chosen to demonstrate that the black box model outperforms the KS equation even in parameter values where the latter is inaccurate ($R = 4.2$). Despite the apparent failure of the KS equation to capture the wave dynamics accurately, it still yields qualitatively good results.

It is important to note here that the black box model methodology yields accurate data-driven models, also in different flow regimes such as in the case of solitary waves, appearing in larger domains for larger Reynolds numbers (cf. figure 5). Specifically, results are presented for a flow with $R = 20$, $Ca = 0.00952$, domain length $L = 300$ and inclination angle $\theta = 6.4^\circ$, exhibiting excellent performance, even in the region

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

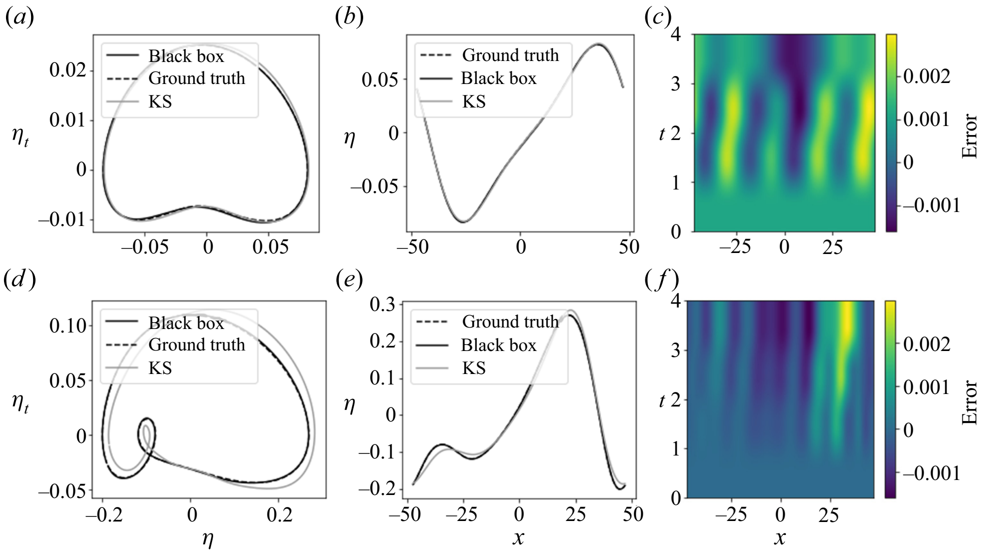

Figure 4. Black box model wave comparison. (*a*,*d*) Attractors; (*b*,*e*) time instance of the amplitude, derived by the NS equation (dashed line), the KS equation (grey line) and the NN-derived PDE (black line); (*c*,*f*) error. Results are shown for (*a–c*) $R = 1.95$, (*d–f*) $R = 4.2$; all results are rescaled in the NS scaling.

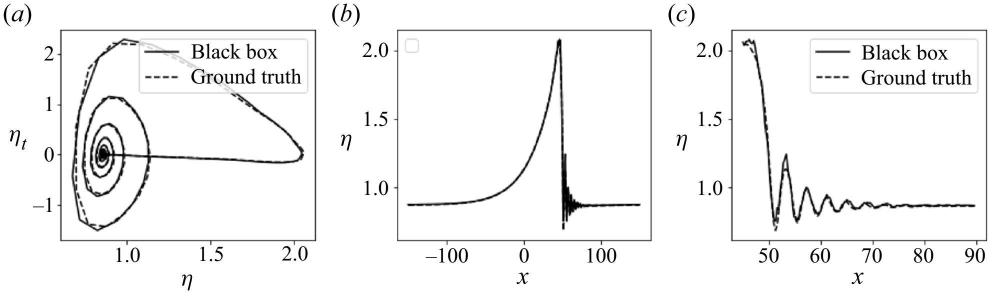

Figure 5. Black box model performance in the case of a solitary wave ($R = 20$, $L = 300$, $\theta = 6.4^\circ$). (*a*) Attractors for R numbers derived by (i) the NS equation (dashed line), (ii) the NN-derived PDE (black); all results are rescaled in the NS scaling. (*b*) Amplitude of the NS equation (dashed line) and the black box model derived at the same time instance. (*c*) Blow-up of the amplitude comparison in the capillary ripples region.

where capillary ripples develop (cf. figure 5). This shows that the applicability of the method is not restricted to low Reynolds numbers, but rather it can be applied to various flow regimes. Nevertheless, in the analysis that follows, the applications are limited to flow regimes where the KS equation is valid or mildly inaccurate, because we exploit the qualitatively good performance of the KS equation and we want to infuse physical information into the data-driven amplitude equation. This is discussed in the following paragraphs.

3.2. *Grey box model I: learning an additive correction to the KS equation*

Instead of training a neural network to learn the right-hand side of a PDE as a black box, i.e. without any physical intuition about the function, the KS equation (the approximate analytical model) is used as a foundation upon which a correction is added to make it

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

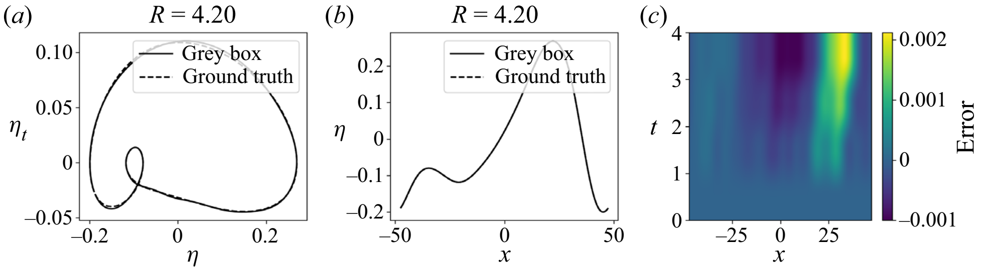

Figure 6. Additive grey box model comparison to ground truth, $R = 4.2$; attractors (*a*); wave amplitude in a specific time step (*b*); absolute error at each point in space and time between grey box I model and ground truth.

more accurate. This correction is discovered in a data-driven way, using the same data as in the black box model (described in the previous paragraph) and also the same network architecture. In this case though, the output of the neural network is not the time derivative of the amplitude but rather the difference between the actual time derivative and the time derivative of the KS equation, i.e.

$$\left(\frac{\partial \eta}{\partial t}\right)^{NS} - \left(\frac{\partial \eta}{\partial t}\right)^{KS} = f^{additive}\left(\eta, \eta_x, \eta_{xx}, \eta_{xxx}, \eta_{xxxx}, R\right), \tag{3.2}$$

which can then be thought of as a ‘residual network’ (ResNet) (He *et al.* 2016).

This approach maintains the physical insight already offered by the approximate equations, but improves its accuracy in a data-driven fashion. The predictions of this corrected model, referred to as an additive grey box model, to contrast with the black box model presented before, are visually very close to the ground truth as shown in figure 6 for $R = 4.2$. This demonstrates the accuracy of the correction achieved in a parameter value where the KS equation is inaccurate. Nevertheless, the accuracy of the approximate equation (the KS equation here) is not a prerequisite for the additive grey box approach to work. This is demonstrated in figure 7 where we examine the case of a travelling solitary wave (the approximate KS equation is not applicable in this regime) and find an excellent agreement between the grey box model and ground truth. This is not surprising, as the neural network architecture is the same: instead of learning the right-hand side of a PDE, it learns the difference between the ‘truth’ and the approximate equation. The benefit of this approach is that, on top of deriving an accurate amplitude equation, it offers a measure of the deviation from other approximate models that can potentially help determine the limits of approximate equations in different applications.

3.3. *Grey box model II: a functional correction*

Exploiting further the physical insight of the KS equation, even in parameter ranges where it is inaccurate, it is possible to use local, in space and/or in time, values of the right-hand side of the KS function, evaluated using the NS data as inputs to the function, to approximate the ‘correct’ right-hand side of the data-driven PDE. Now, a neural network is trained to predict the time derivative of the amplitude, given a few locally nearby evaluations of the function of the KS time derivative, denoted by f^{KS} in (3.3), using ground-truth data from the NS simulations as inputs. Alternatively, a few of its derivatives with respect to the dependent variable, η, or a few nearby values of its spatial partial derivatives, e.g. η_x, η_{xx}, evaluated on NS data, can also be used as inputs to the functional

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

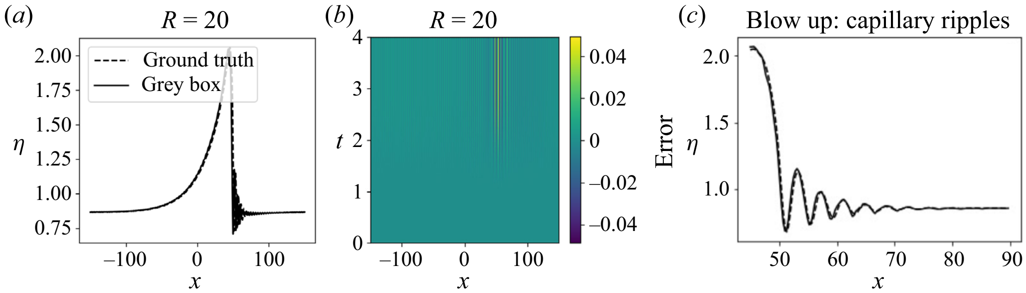

Figure 7. Additive grey box model performance in the case of a solitary wave ($R = 20$, $L = 300$, $\theta = 6.4^\circ$). (*a*) Amplitude of the NS equation (dashed line) and the grey box model derived at the same time instance (all results are rescaled in the NS scaling). (*b*) Absolute error at each point in space and time. (*c*) Blow-up of the amplitude comparison in the capillary ripples region.

model. Several flavours of this approach are implemented, i.e.

$$\left.\begin{aligned}\left(\frac{\partial\eta}{\partial t}\right)^{NN} &= f_1^{functional}(f^{KS}, \partial f^{KS}/\partial\eta, \partial f^{KS}/\partial\eta_x, R),\\ \left(\frac{\partial\eta}{\partial t}\right)^{NN} &= f_2^{functional}(\partial f^{KS}/\partial\eta, \partial f^{KS}/\partial\eta_x, \partial f^{KS}/\partial\eta_{xx}, R),\\ \left(\frac{\partial\eta}{\partial t}\right)^{NN} &= f_3^{functional}(f_j^{KS}, f_{j-1}^{KS}, f_{j+1}^{KS}, R),\\ \left(\frac{\partial\eta}{\partial t}\right)^{NN} &= f_4^{functional}(f_t^{KS}, f_{t-1}^{KS}, f_{t-2}^{KS}, R).\end{aligned}\right\} \tag{3.3}$$

In the last two examples the subscripts j, $j-1$, $j+1$ signify points in space where the value of the right-hand side of the KS equation, f^{KS}, is taken in the same time step; whereas the subscripts t, $t-1$, $t-2$ stand for different nearby points in time, where the value is taken at the same point in space. In every one of the cases mentioned, the same network architecture is implemented in Tensorflow, consisting of two hidden layers with 20 neurons, the tanh activation function, the mean squared error loss function and the Adam optimizer.

This idea is inspired by the Takens embedding theorem (Whitney 1936; Takens 1981), which allows us to recover missing variables and reconstruct the dynamics of a system, given a short time history of the variables that can be measured. Here, this short time history is replaced by values of f^{KS} and some of its partial derivatives, e.g. $\partial f^{KS}/\partial\eta$ and $\partial f^{KS}/\partial\eta_x$. Alternatively, one can consider the value of f^{KS} at the same point in time but in three different nearby spatial points, or in the same point is space but in three different nearby time steps.

The performance of the alternative functional models is shown in figure 8 for the same parameter value, $R = 4.2$. The error is within the same order of magnitude as in the previous case of the additive residual model.

4. Nonlinear manifold learning for predictions from small data

The focus here is shifted to exploiting NS data to recover missing information. Missing data is a critical problem in applications in flow measurement and monitoring.

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

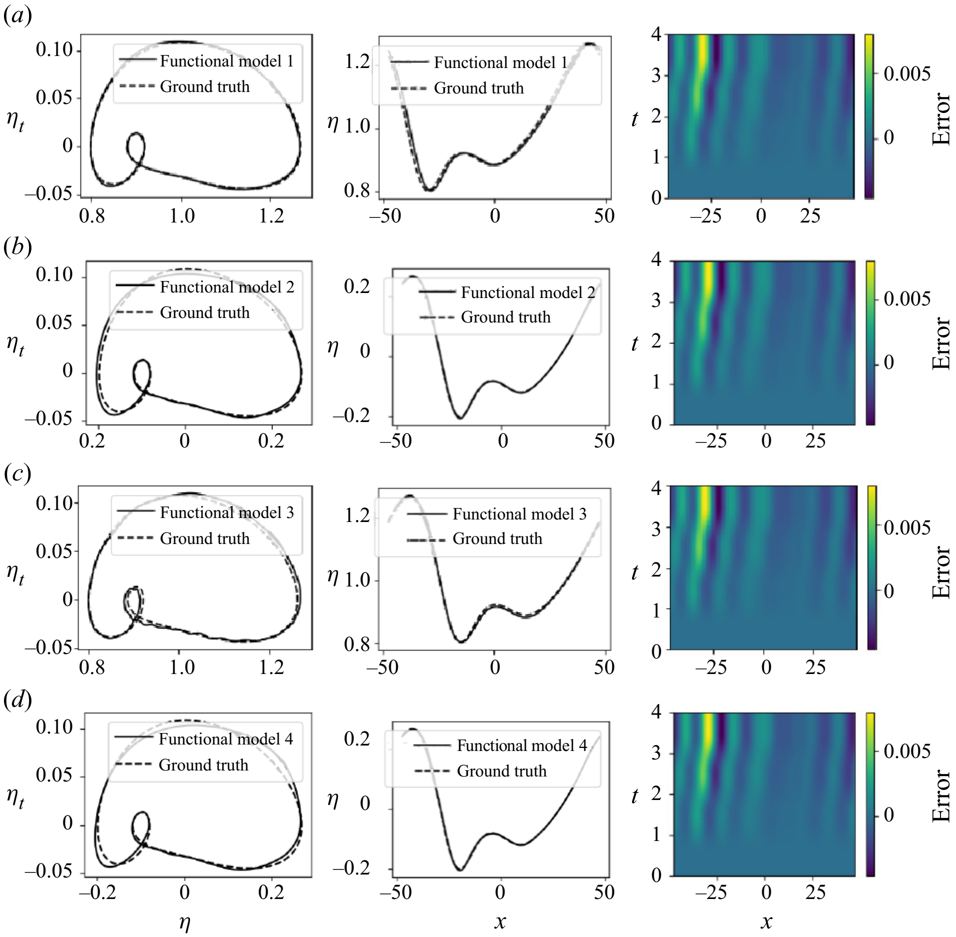

Figure 8. ‘Functional grey box’ model, $R = 4.2$ (cf. (3.3)) comparison to ground truth. (*a*) Functional model 1, (*b*) functional model 2, (*c*) functional model 3 and (*d*) functional model 4. (left) Attractor comparison; (centre) wave comparison at a specific time step; (right) absolute error at each point in space and time.

For example, in film flow applications it is often easy for an experimentalist to measure the film height, whereas being able to evaluate the detailed underlying flow field is a significantly more difficult task, if not impossible in the case of opaque liquids. The goal here is to exploit NS data to derive a predictive tool, e.g. for the full velocity field or even the flow parameter, the Reynolds number, from only partial information. More importantly, to be able to do so as efficiently as possible and without having to care too much about the sensor positions.

The proposed approach is inspired by gappy POD (Everson & Sirovich 1995), according to which it is possible to recover missing information from a vector that we know belongs to the subspace spanned by a few predetermined POD modes, by performing efficient interpolation in this reduced subspace. Here, the same concept is demonstrated, but also the notion of deriving a reduced description of the data with nonlinear manifold learning, in this case with diffusion maps (Coifman & Lafon 2006; Nadler *et al.* 2006; Coifman *et al.* 2008) (details about diffusion maps and gappy POD can be found in Appendix B). The added benefit is twofold: first, diffusion maps identify a parsimonious parametrization of the reduced subspace, which requires significantly less modes than POD, especially if

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

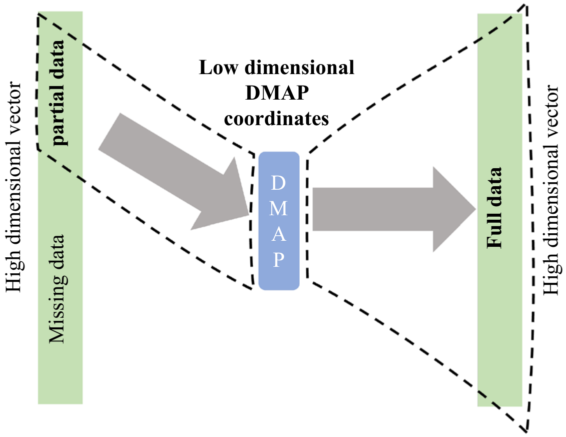

Figure 9. Schematic representation of the gappy DMAPs workflow: the DMAP coordinates of the full dataset are discovered. Then from partial data, the corresponding latent (DMAP) variables are inferred and, from them, it is possible to map back to the complete data.

the data belong to a curved manifold. The second benefit is related to the fact that in gappy POD the accuracy of the method is critically influenced by the 'location' of the known elements of the vector. The reason is purely numerical, and has to do with the condition number of the gappy matrix $M = (m{\cdot}\Phi)'{\cdot}(m{\cdot}\Phi)$ (cf. Appendix B.4 for details), with Φ the selected POD basis and m the mask matrix that defines which elements of the vector are known.

4.1. *Out-of-sample predictions: a gappy DMAP approach*

The implementation of gappy diffusion maps, schematically represented in figure 9, is presented in detail in Koronaki *et al.* (2023) and summarized here for completeness.

The workflow starts by identifying a parametrization of the manifold, where the original, full dataset belongs. In this implementation, for every value of the parameter R, each data point contains the wave amplitude distribution and the velocity distributions of the fluid, perpendicular to the solid–liquid interface at different points along its length l (schematically represented with grey arrows in figure 1). Here for comparison purposes, 80 % of the dataset is used for training and the rest for testing. To derive a parsimoniously low-dimensional description of the data, the diffusion maps algorithm is implemented, as detailed in Appendix B.1. It is found by implementing the local linear regression algorithm introduced in Dsilva *et al.* (2018) and summarized in Appendix B.1, that three diffusion coordinates are enough to describe any vector in the dataset.

Then, a second round of diffusion maps is implemented, in conjunction to geometric harmonics interpolation (details can be found in Appendix B and in Evangelou *et al.* 2022), in order to map from any point on the reduced space to the high-dimensional ambient space.

Having established the methods for mapping between the ambient and reduced space, it is now possible, given partial information, to find first the corresponding reduced coordinates and then the entire ambient vector, including the missing information.

The accurate performance of this workflow is demonstrated in figure 10, where three cases are examined: (i) 80 points along the interface are known, from which the velocity and parameter value is recovered, with a maximum error of 4 %; (ii) 8 points, evenly distributed along the interface are known, from which again the velocity values and the

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

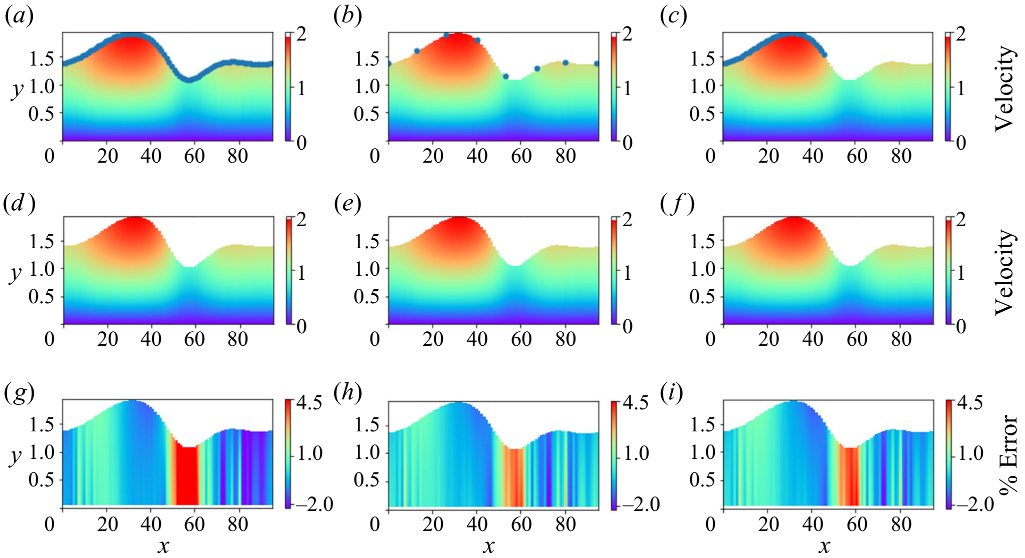

Figure 10. Double DMAPS; $R = 3.5$. Each column of figures presents the actual velocity contours (*a–c*), predicted velocity contours (*d–f*) and error (*g–i*). The dots in the top figures signify points where the value of the amplitude is considered known. (*a*,*d*,*g*) The value of the amplitude at 80 points is considered known and the maximum error is 4 %. (*b*,*e*,*h*) The value at 8 equidistant points is considered known and the maximum error is 4 %. (*c*,*f*,*i*) The value at 40 points in the first half of the wave is considered known and the maximum error is again close to 4 %.

parameter is recovered with a maximum error of 4 %; (iii) 40 points are known belonging to only half of the interface shape, leading to prediction of the velocities and the parameter with a 4 % error.

The same computational experiments are conducted with gappy POD and the results are shown in figure 11. First, a POD basis is determined, based on the error (1.5 %, the same as the reconstruction error achieved by DMAPs) between the reconstructed and the original test dataset, which leads to a basis with six POD vectors. The number of POD vectors required to describe the data is double the number of DMAP coordinates, suggesting that the underlying manifold is curved. The same points along the interface as before are considered known: (i) with 80 points along the interface, the maximum prediction error for the unknown velocity and parameter values is 10 %; (ii) with 8 equidistant points along the interface, the maximum prediction error is approximately the same; (iii) when 40 points along half of the interface are considered, then the maximum error soars to 35 % and the predicted wave shape and velocity distribution is visibly inaccurate.

Gappy DMAPs are also successfully implemented in the case of a travelling solitary wave ($R = 20$, $Ca = 0.00952$, $L = 300$ and $\theta = 6.4^{\circ}$) (cf. figure 12). Due to the complicated surface geometry of the solitary wave, with the presence of capillary ripples at the leading edge of the main humb, the dimensionality of the latent space, parameterized by the diffusion coordinates, is larger (four DMAPs) than in the low Reynolds case. Once the low-dimensional representation of the data is determined, it is possible to map between partial observations (i.e. interface height measurements) and the high-dimensional ambient space that includes the velocity distribution of the fluid. The prediction accuracy for the solitary wave is comparable to the low Reynolds case, leading to a maximum 1.5 % error, when 80 equidistant points are considered on the surface, which

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

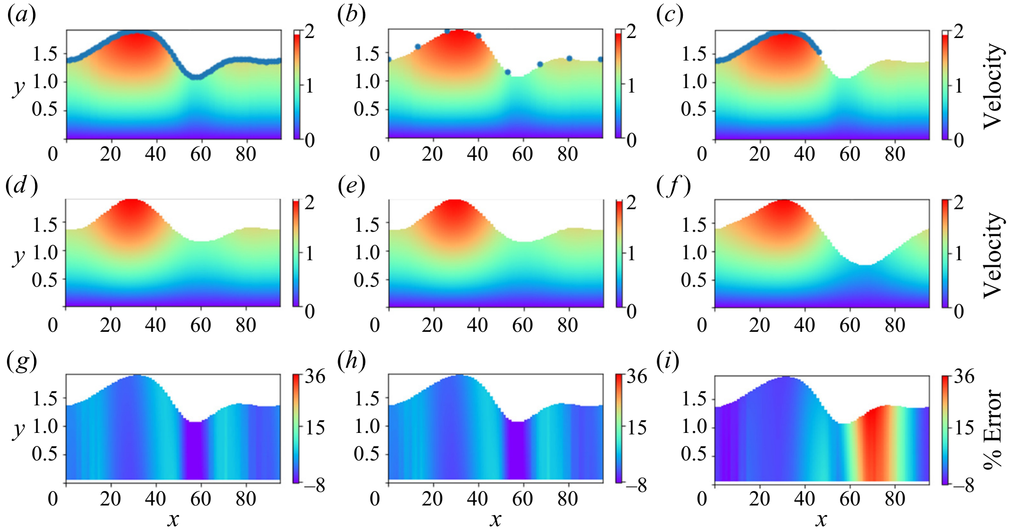

Figure 11. Gappy POD; $R = 3.5$. Each column of figures presents the actual velocity contours (*a–c*), predicted velocity contours (*d–f*) and error (*g–i*). The dots in the top figures signify points where the value of the amplitude is considered known. (*a*,*d*,*g*) The value of the amplitude at 80 points is considered known and the maximum error is 10 %. (*b*,*e*,*h*) The value at 8 equidistant points is considered known and the maximum error is 10 %. (*c*,*f*,*i*) The value at 40 points in the first half of the wave is considered known and the maximum error is close to 35 %.

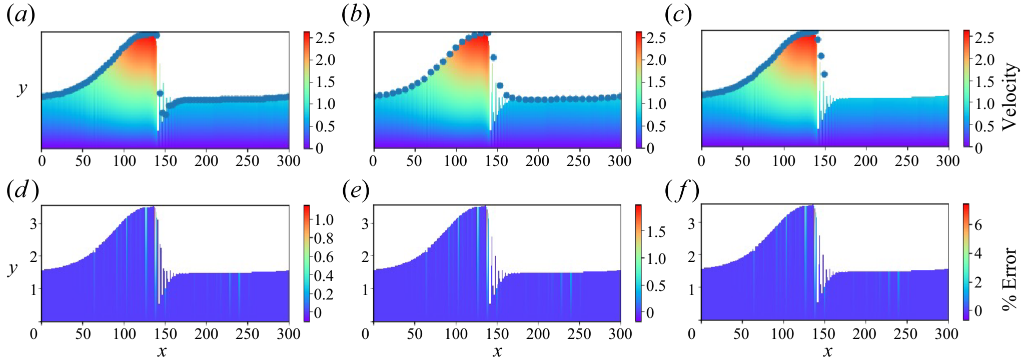

Figure 12. Gappy DMAPS; $R = 20$, $Ca = 0.00952$, $L = 300$, $\theta = 6.4^{\circ}$. Each column of figures presents the predicted velocity contours (*a–c*) and error (*d–f*). The dots in the top figures signify points where the value of the amplitude is considered known. (*a*,*d*) The value of the amplitude at 80 points is considered known and the maximum error is 1.5 %. (*b*,*e*) The value at 40 equidistant points is considered known and the maximum error is 3 %. (*c*,*f*) The value at 40 points in the first half of the wave is considered known and the maximum error is again close to 7 %.

rises to 3 % when 40 equidistant points are considered known. When 40 points along half of the domain are considered known, then the error increases to 7 %.

5. Discussion

In this work, three different strategies for deriving accurate and economical surrogates are presented along with their application to the amplitude evolution of falling thin films.

The first option is purely data driven and physics agnostic, and relies on learning, as a black box, an amplitude PDE from observed NS data over a range of R values. In essence, the right-hand side of the PDE is substituted by an artificial neural network, which can then be integrated in time for various times, different initial conditions and parameter values.

As an alternative, we propose using a low-fidelity model, here the KS equation, in order to infuse physical intuition into the learned model. This is achieved in two different ways: the first, the additive grey box approach, uses a few high-fidelity data (results of the NS) to calibrate the KS equation, by learning an additive correction, by way of a neural network, rather than the entire right-hand side.

The second strategy, the so-called functional grey box approach, is inspired by Takens' embedding theory, and proposes learning the accurate amplitude dynamics from a few observations of the right-hand side of an innacurate equation, such as the KS equation. Four different options are presented, utilizing either the right-hand side of the KS equation and some of its derivatives, or the value of the right-hand side operator at three points in space (at the same time instance) or in time (for the same point in space). This last approach is a demonstration of how a model that is qualitatively close to the ground truth, but quantitatively off, can be leveraged, in the data-driven era, into a more accurate and efficient learned model.

Finally, we presented a gappy DMAP methodology, the nonlinear counterpart of gappy POD, which allows us to infer quantities that are inaccessible to measuring devices, such as the velocity profile of the fluid below the surface, when only some measurements are known, such as the height of the thin film at certain points. This may be trivial for low R values, since the interface height is 'slaved' to the velocity. Nevertheless, for slightly higher R values, the amplitude is no longer a function of just the position; hence, surrogate models with more than one equation become necessary in this flow regime (Chang 1986; Oron & Gottlieb 2002; Kalliadasis *et al.* 2012; Shklyaev & Nepomnyashchy 2017).

The benefits of the proposed approach are twofold: nonlinear manifold learning methods, such as DMAPs, yield a more parsimonious description of the manifold, requiring only a few coordinates to accurately reconstruct the original data. In contrast, we demonstrate that gappy POD requires higher-dimensional hyperplanes to span the data. The second advantage of gappy DMAPs is related to the choice of known values. Some consideration is necessary for choosing points that contain rich enough information in order to achieve accurate reconstruction; it is, nevertheless, less sensitive to the position of the provided measurements than its linear counterpart.

Funding. This work was partially supported by the US AFOSR and by the US DOE (IGK). CML received the support of a 'la Caixa' Foundation Fellowship (ID 100010434), code LCF/BQ/AA19/11720048. EDK received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 890676 - DataProMat.

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

Author ORCIDs.

George Karapetsas https://orcid.org/0000-0002-3486-7484;

Eleni D. Koronaki https://orcid.org/0000-0002-5229-4157.

Author contributions. C.M.L. performed computational analyses, interpreted the data and wrote the manuscript, Y.P. performed computational analyses and wrote the manuscript, G.K. developed the computational fluid dynamics model, contributed to mathematical analyses and wrote the manuscript, E.D.K. performed computational analyses, interpreted the data and wrote the manuscript and I.G.K. conceived the research, supervised the study and wrote the manuscript.

Appendix A. Transformations between NS and KS equations scales

The (x, t) NS frame of reference is mapped on the KS (ξ, τ) through the expressions

$$\xi = \epsilon(x - Ft) \quad \text{and} \quad \tau = \frac{\epsilon^4 WF}{12} t, \tag{A1a,b}$$

while the interfacial height, $h(x, t)$, is related to the amplitude $\phi(\xi, \tau)$ as

$$h(x, t) \approx 1 + \epsilon \frac{RF}{15} \phi(\xi, \tau). \tag{A2}$$

To be able to compare the results between NS and KS equations, we employ the chain rule and derive the appropriate transformations for the time and spatial derivatives. To transform NS data to the KS formulation, the following expressions can be used:

$$\left.\begin{aligned} \frac{\partial^\beta \phi}{\partial \xi^\beta} &= \frac{1}{\epsilon^{\beta+1}} \frac{15}{RF} \frac{\partial^\beta h}{\partial x^\beta}, \\ \frac{\partial \phi}{\partial \tau} &= \frac{180}{\epsilon^5 WRF^2} \frac{\partial h}{\partial t} + \frac{180}{\epsilon^5 WRF} \frac{\partial h}{\partial x}. \end{aligned}\right\} \tag{A3}$$

Inversely, to map KS data to the NS formulation, the following expressions can be used:

$$\left.\begin{aligned} \frac{\partial^\beta h}{\partial x^\beta} &= \epsilon^{\beta+1} \frac{RF}{15} \frac{\partial^\beta \phi}{\partial \xi^\beta}, \\ \frac{\partial h}{\partial t} &= -\frac{\epsilon^2 RF^2}{15} \frac{\partial \phi}{\partial \xi} + \frac{\epsilon^5 WRF^2}{180} \frac{\partial \phi}{\partial \tau}. \end{aligned}\right\} \tag{A4}$$

Appendix B. Diffusion maps and gappy POD

B.1. *Diffusion maps*

Diffusion maps (Coifman & Lafon 2006; Nadler *et al.* 2006; Coifman *et al.* 2008) are a framework that can (based upon diffusion processes) facilitate discovering meaningful low-dimensional intrinsic geometric descriptions of datasets, even when the data are high dimensional, nonlinear and/or corrupted by (relatively small) noise. The method is based on the construction of a Markov transition probability matrix, corresponding to a random walk, on a graph whose vertices are the data points, with transition probabilities being the local similarities between pairs of data points. The leading few eigenvectors of the Markov matrix can be used as data-driven coordinates that provide a reparametrization of the data. To construct a low-dimensional embedding for a dataset $\boldsymbol{X}$ of M individual points (represented as d-dimensional real vectors $x_1, \ldots, x_M$), a similarity measure d_{ij} between each pair of vectors x_i, x_j is computed. The standard Euclidean distance or the Euclidean norm may be considered to this end. By using this similarity measure, an affinity matrix is constructed. A popular choice is the Gaussian kernel

$$w(i,j) = \exp\left[-\left(\frac{\|x_i - x_j\|}{\epsilon_\delta}\right)^2\right], \tag{B1}$$

where ϵ_δ defines a scale hyperparameter that quantifies the local similarity for each data point. To recover a parametrization regardless of the sampling density, the normalization

$$\tilde{\boldsymbol{W}} = \boldsymbol{P}^{-\alpha} \boldsymbol{W} \boldsymbol{P}^{-\alpha} \tag{B2}$$

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

is performed, where $P_{ii} = \sum_{j=1}^{M} W_{ij}$ and $\alpha = 1$ to factor out the density effects. A second normalization applied on $\tilde{\boldsymbol{W}}$,

$$\boldsymbol{K} = \boldsymbol{D}^{-1}\tilde{\boldsymbol{W}}, \tag{B3}$$

gives a $M \times M$ Markov matrix $\boldsymbol{K}$, where $\boldsymbol{D}$ is a diagonal matrix, collecting the row sums of matrix $\tilde{\boldsymbol{W}}$. The stochastic matrix $\boldsymbol{K}$ has a set of real eigenvalues $1 = \lambda_1 \geq \cdots \geq \lambda_M$ with corresponding eigenvectors ϕ_i. To check if model (variable) reduction can be achieved, the number of retained eigenvectors has to be appropriately truncated. In practice, it is useful to consider that not all obtained eigenvectors parametrize independent directions, but rather most of them can be considered as spanning the same directions with different frequencies. Eigenvectors that parametrize the same directions in this context are called harmonics and those that parametrize independent directions non-harmonics. A minimal representation of the DMAP space is made possible by carefully selecting the non-harmonic coordinates, which do not necessarily correspond to the most dominant eigenmodes of the Markov matrix. This is a stark difference between diffusion maps and its linear counterpart, POD or principal component analysis, where the dominant modes are retained for the truncated representation of the data. If the number of the non-harmonic eigenvectors is less than the number of ambient space dimensions, then model (variable) reduction is achieved.

A proposed algorithm for identifying the non-harmonic eigenvectors is presented in Dsilva *et al.* (2018), based on local linear regression. In a nutshell, a local linear function is used in order to fit the DMAP coordinate ϕ_k as a function, f, of the previous vectors $\tilde{\Phi}_{k-1} = [\phi_1, \phi_2, \ldots, \phi_{k-1}]$. If ϕ_k can be accurately expressed as a function of the other DMAP coordinates, then it does not represent a new direction on the dataset, and is omitted for dimensionality reduction. On the contrary, if ϕ_k cannot be expressed as a function of the previous eigenvectors then ϕ_k is a new independent eigendirection that must be retained for a parsimonious representation of the data. To quantify the accuracy of the fit, the following metric is used:

$$r_k = \sqrt{\frac{\displaystyle\sum_{i=1}^{n}(\phi_k(i) - f(\tilde{\Phi}_{k-1}(i)))^2}{\displaystyle\sum_{i=1}^{n}(\phi_k(i))^2)}}. \tag{B4}$$

A small value of r_k is associated with a ϕ_k that is a harmonic function of the previous eigenmodes, whereas a higher value of r_k signifies that ϕ_k is a new independent direction on the data manifold. It has been shown in Dsilva *et al.* (2018) that selecting only the eigenvectors that correspond to higher values of r_k leads to a parsimonious representation of the data. Eventually, the vector x_i is mapped to a vector whose first component is the ith component of the first selected non-trivial eigenvector, whose second component is the ith component of the second selected non-trivial eigenvector, and so on.

To map a new point, x_{new}, from the ambient space to DMAP space, a mathematically elegant approach known as the Nyström extension, introduced in (Coifman & Lafon 2006) is used, summarized here for completeness. The starting point of the Nyström extension is to compute the distances, $d(\cdot, x_{new})$, between the new point, x_{new}, and the M data points in the original dataset, the same normalizations used for DMAP need to also be applied here.

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

The Nyström extension formula reads

$$\phi_j(x_{new}) = \lambda_j^{-1} \sum_{i=1}^{M} \tilde{k}(x_i, x_{new})\phi_j(x_i), \tag{B5}$$

where λ_j is the jth eigenvector and $\phi_j(x_i)$ is the ith component of the jth eigenvector.

B.2. *Geometric harmonics*

Geometric harmonics was introduced in Coifman & Lafon (2006), inspired by the Nyström extension as a scheme for extending functions defined on data $\boldsymbol{X}$, $f(\boldsymbol{X}) : \boldsymbol{X} \to \mathbb{R}$, for $x_{new} \notin \boldsymbol{X}$. This out-of-sample extension is achieved by using a particular set of basis functions called geometric harmonics. These functions are computed as eigenvectors of the symmetric $M \times M$ $\boldsymbol{W}$ matrix. The eigendecomposition of the symmetric and positive semidefinite matrix $\mathbf{W}$ leads to a set of orthonormal eigenvectors $\psi_1, \psi_2, \ldots, \psi_M$ with non-negative eigenvalues $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M \geq 0$.

From this set of eigenvectors, to avoid numerical issues, we consider a truncated subset $S_\delta = (\alpha : \sigma_\alpha \geq \delta\sigma_1)$ where $\delta > 0$. The extension of f for a new point x_{new} is accomplished by first projecting the function of interest in the (truncated) computed set of eigenvectors

$$f \to P_\delta f = \sum_{\alpha \in S_\delta} \langle f, \psi_\alpha \rangle \psi_\alpha \tag{B6}$$

and then extending the function f for $x_{new} \notin \boldsymbol{X}$,

$$(Ef)(x_{new}) = \sum_{\alpha \in S_\delta} \langle f, \psi_\alpha \rangle \Psi_\alpha(x_{new}), \tag{B7}$$

where

$$\Psi_\alpha(x_{new}) = \lambda_\alpha^{-1} \sum_{i=1}^{M} w(x_{new}, x_i)\psi_\alpha(x_i) \tag{B8}$$

and

$$w(x_{new}, x_i) = \exp\left[- \left(\frac{d_i}{\tilde{\epsilon}} \right)^2 \right], \quad d_i = \|x_{new} - x_i\|_2. \tag{B9}$$

B.3. *Double diffusion maps and their latent harmonics*

A slight twist of the geometric harmonics (Evangelou *et al.* 2022) is presented in this section. As discussed above, geometric harmonics constructs an input–output mapping between the ambient coordinates $\boldsymbol{X}$ and a function of interest f defined on $\boldsymbol{X}$. However, it is possible, if the data are lower dimensional, to construct a map in terms of only the non-harmonic eigenvectors. This is achieved similar to the traditional geometric harmonics, by firstly constructing an affinity matrix

$$\bar{w}(i, j) = \exp\left[- \left(\frac{\|\phi_i - \phi_j\|}{\epsilon^\star} \right)^2 \right]. \tag{B10}$$

In this case, the affinity matrix is constructed in terms of only the non-harmonic DMAPs coordinates. To distinguish the notation between geometric harmonics and double

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

diffusion maps, we will use $\bar{\psi}$. As in the traditional geometric harmonics, the function f is projected to a truncated set of the obtained eigenvectors

$$f \rightarrow P_\delta f = \sum_{\beta \in \bar{S}_\delta} \langle f, \overline{\psi}_\beta \rangle \overline{\psi}_\beta. \tag{B11}$$

The extension of f for ϕ_{new} is achieved by firstly extending the values of the geometric harmonic functions Ψ_β for ϕ_{new},

$$\overline{\Psi}_\beta(\boldsymbol{\phi}_{new}) = \overline{\sigma}_\beta^{-1} \sum_{i=1}^{M} \bar{w}(\phi_{new}, \phi_i) \overline{\psi}_\beta(\boldsymbol{\phi}_i), \tag{B12}$$

and then estimating the value of f at ϕ_{new},

$$(Ef)(\boldsymbol{\phi}_{new}) = \sum_{\beta \in \bar{S}_\delta} \langle f, \overline{\psi}_\beta \rangle \overline{\Psi}_\beta(\boldsymbol{\phi}_{new}). \tag{B13}$$

B.4. *Gappy POD*

In this section the gappy POD method is summarized for completeness. Consider a dataset $\boldsymbol{X}$ of M vectors (represented as d-dimensional real vectors $x_1, \ldots, x_M$). A POD basis, $\varPhi \in \Re^{N \times M}$, of $\boldsymbol{X}$ is computed, such that $\boldsymbol{X}$ can be approximated as a linear combination of $\boldsymbol{p}$ vectors, i.e.

$$\tilde{X} = \sum_{j=1}^{p} c^j \varPhi^j \tag{B14}$$

or in matrix-vector format,

$$\tilde{X} = \varPhi \cdot c. \tag{B15}$$

The size of the truncated POD basis $\varPhi$ is selected based on the error between the actual vector $\boldsymbol{X}$ and the reconstructed approximation $\tilde{\boldsymbol{X}}$,

$$\text{Reconstruction error } = \|\boldsymbol{X} - \tilde{\boldsymbol{X}}\|. \tag{B16}$$

Consider now a vector X' that is spanned by the same basis $\varPhi$ and that only m values of this vector are known, so that the partial vector $X'_{partial}$ can be defined as

$$X'_{partial} = m \cdot X', m \in \Re^{m \times N}. \tag{B17}$$

The goal is to find coefficients c', such that an approximation $\widetilde{X'}$ of the vector X' can be defined as

$$\widetilde{X'} = X' \cdot c'; \tag{B18}$$

then

$$X'_{partial} \approx m \cdot X' \cdot c'. \tag{B19}$$

Finding the values of c' that satisfy the above leads to an optimization problem solved through the linear system

$$M \cdot c' = (m \cdot \varPhi)' \cdot X'_{partial}, \tag{B20}$$

with $M = (m \cdot \varPhi)' \cdot (m \cdot \varPhi)$.

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

References

Abadi, M., *et al.* 2015 TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Arbabi, H. & Kevrekidis, I.G. 2021 Particles to partial differential equations parsimoniously. *Chaos* **31** (3), 033137.

Balasubramanian, M. & Schwartz, E.L. 2002 The isomap algorithm and topological stability. *Science* **295** (5552), 7–7.

Bharadwaj, A.S., Kuhnert, J., Bordas, S.P.A. & Suchde, P. 2022 A discrete droplet method for modelling thin film flows. *Appl. Math. Model.* **112**, 486–504.

Brown, H.S. 1992 A computer-assisted, nonlinear dynamic study of instabilities and pattern formation for interfacial waves. PhD thesis, Princeton University.

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.

Chakraborty, S., Nguyen, P.-K., Ruyer-Quil, C. & Bontozoglou, V. 2014 Extreme solitary waves on falling liquid films. *J. Fluid Mech.* **745**, 564–591.

Chang, H.-C. 1986 Nonlinear waves on liquid film surfaces: I. Flooding in a vertical tube. *Chem. Engng Sci.* **41** (10), 2463–2476.

Chen, R.T.Q., Rubanova, Y., Bettencourt, J. & Duvenaud, D. 2018 Neural ordinary differential equations. *Adv. Neural Inform. Proc. Syst.* **31**.

Coifman, R.R., Kevrekidis, I.G., Lafon, S., Mauro, M. & Nadler, B. 2008 Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems. *Multiscale Model. Simul.* **7** (2), 842–864.

Coifman, R.R. & Lafon, S. 2006 Diffusion maps. *Appl. Comput. Harmon. Anal.* **21** (1), 5–30.

Dietze, G.F., Rohlfs, W., Nährich, K., Kneer, R. & Scheid, B. 2014 Three-dimensional flow structures in laminar falling liquid films. *J. Fluid Mech.* **743**, 75–123.

Dsilva, C.J., Talmon, R., Coifman, R.R. & Kevrekidis, I.G. 2018 Parsimonious representation of nonlinear dynamical systems through manifold learning: a chemotaxis case study. *Appl. Comput. Harmon. Anal.* **44** (3), 759–773.

Duraisamy, K., Iaccarino, G. & Xiao, H. 2019 Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51** (1), 357–377.

Evangelou, N., Dietrich, F., Chiavazzo, E., Lehmberg, D., Meila, M. & Kevrekidis, I.G. 2022 Double diffusion maps and their latent harmonics for scientific computations in latent space. arXiv:2204.12536

Everson, R. & Sirovich, L. 1995 Karhunen–Loeve procedure for gappy data. *JOSA A* **12** (8), 1657–1664.

Farzamnik, E., Ianiro, A., Discetti, S., Deng, N., Oberleithner, K., Noack, B.R. & Guerrero, V. 2023 From snapshots to manifolds – a tale of shear flows. *J. Fluid Mech.* **955**, A34.

Floryan, D. & Graham, M.D. 2022 Data-driven discovery of intrinsic dynamics. *Nat. Mach. Intell.* **4** (12), 1113–1120.

Galaris, E., Fabiani, G., Gallos, I., Kevrekidis, I. & Siettos, C. 2022 Numerical bifurcation analysis of PDEs from lattice Boltzmann model simulations: a parsimonious machine learning approach. *J. Sci. Comput.* **92** (2), 34.

Glasser, B.J., Kevrekidis, I.G. & Sundaresan, S. 1997 Fully developed travelling wave solutions and bubble formation in fluidized beds. *J. Fluid Mech.* **334**, 157–188.

González-García, R., Rico-Martínez, R. & Kevrekidis, I.G. 1998 Identification of distributed parameter systems: a neural net based approach. *Comput. Chem. Engng* **22**, S965–S968.

He, K., Zhang, X., Ren, S. & Sun, J. 2016 Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV*, pp. 770–778.

Kalliadasis, S., Ruyer-Quil, C., Scheid, B. & Velarde, M.G. 2012 *Falling Liquid Films*, Applied Mathematical Sciences, vol. 176. Springer London.

Kemeth, F.P., Bertalan, T., Thiem, T., Dietrich, F., Moon, S.J., Laing, C.R. & Kevrekidis, I.G. 2022 Learning emergent partial differential equations in a learned emergent space. *Nat. Commun.* **13** (1), 3318.

Kemeth, F.P., Alonso, S., Echebarria, B., Moldenhawer, T., Beta, C. & Kevrekidis, I.G. 2023 Black and gray box learning of amplitude equations: application to phase field systems. *Phys. Rev. E* **107** (2), 025305.

Kevrekidis, I.G., Nicolaenko, B. & Scovel, J.C. 1990 Back in the saddle again: a computer assisted study of the Kuramoto–Sivashinsky equation. *SIAM J. Appl. Math.* **50** (3), 760–790.

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

KORONAKI, E.D., EVANGELOU, N., PSARELLIS, Y.M., BOUDOUVIS, A.G. & KEVREKIDIS, I.G. 2023 From partial data to out-of-sample parameter and observation estimation with diffusion maps and geometric harmonics. *Comput. Chem. Engng* **178**, 108357.

KRISCHER, K., RICO-MARTÍNEZ, R., KEVREKIDIS, I.G., ROTERMUND, H.H., ERTL, G. & HUDSON, J.L. 1993 Model identification of a spatiotemporally varying catalytic reaction. *AIChE J.* **39** (1), 89–98.

LEE, S., DIETRICH, F., KARNIADAKIS, G.E. & KEVREKIDIS, I.G. 2019 Linking gaussian process regression with data-driven manifold embeddings for nonlinear data fusion. *Interface Focus* **9** (3), 20180083.

LEE, S., KOOSHKBAGHI, M., SPILIOTIS, K., SIETTOS, C.I. & KEVREKIDIS, I.G. 2020 Coarse-scale PDEs from fine-scale observations via machine learning. *Chaos* **30** (1), 013141.

LEE, S., PSARELLIS, Y.M., SIETTOS, C.I. & KEVREKIDIS, I.G. 2022 Learning black- and gray-box chemotactic PDEs/closures from agent based Monte Carlo simulation data. *J. Math. Biol.* **87** (1), 15.

LINOT, A.J., BURBY, J.W., TANG, Q., BALAPRAKASH, P., GRAHAM, M.D. & MAULIK, R. 2023 Stabilized neural ordinary differential equations for long-time forecasting of dynamical systems. *J. Comput. Phys.* **474**, 111838.

LONG, Z., LU, Y., MA, X. & DONG, B. 2018 PDE-net: learning PDEs from data. In *Proceedings of the 35th International Conference on Machine Learning* (ed. Jennifer Dy & Andreas Krause), Proceedings of Machine Learning Research, vol. 80, pp. 3208–3216. Stockholmsmässan, Stockholm, Sweden: PMLR.

LOZANO-DURÁN, A. & BAE, H.J. 2023 Machine learning building-block-flow wall model for large-eddy simulation. *J. Fluid Mech.* **963**, A35.

MELONI, S., CENTRACCHIO, F., DE PAOLA, E., CAMUSSI, R. & IEMMA, U. 2023 Experimental characterisation and data-driven modelling of unsteady wall pressure fields induced by a supersonic jet over a tangential flat plate. *J. Fluid Mech.* **958**, A27.

MILIAIEV, A. & TIMOKHA, A. 2023 Viscous damping of steady-state resonant sloshing in a clean rectangular tank. *J. Fluid Mech.* **965**, R1.

NADLER, B., LAFON, S., COIFMAN, R.R. & KEVREKIDIS, I.G. 2006 Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Appl. Comput. Harmon. Anal.* **21** (1), 113–127.

ORON, A. & GOTTLIEB, O. 2002 Nonlinear dynamics of temporally excited falling liquid films. *Phys. Fluids* **14** (8), 2622–2636.

PAN, S. & DURAISAMY, K. 2018 Data-driven discovery of closure models. *SIAM J. Appl. Dyn. Syst.* **17** (4), 2381–2413.

PARISH, E.J. & DURAISAMY, K. 2016 A paradigm for data-driven predictive modeling using field inversion and machine learning. *J. Comput. Phys.* **305**, 758–774.

PETTAS, D., KARAPETSAS, G., DIMAKOPOULOS, Y. & TSAMOPOULOS, J. 2019*a* Viscoelastic film flows over an inclined substrate with sinusoidal topography. I. Steady state. *Phys. Rev. Fluids* **4** (8), 083303.

PETTAS, D., KARAPETSAS, G., DIMAKOPOULOS, Y. & TSAMOPOULOS, J. 2019*b* Viscoelastic film flows over an inclined substrate with sinusoidal topography. II. Linear stability analysis. *Phys. Rev. Fluids* **4** (8), 083304.

PSARELLIS, Y.M., LEE, S., BHATTACHARJEE, T., DATTA, S.S., BELLO-RIVAS, J.M. & KEVREKIDIS, I.G. 2022 Data-driven discovery of chemotactic migration of bacteria via machine learning. arXiv:2208.11853.

RAISSI, M. & KARNIADAKIS, G. 2017 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.

REZAEIRAVESH, S., MUKHA, T. & SCHLATTER, P. 2023 Efficient prediction of turbulent flow quantities using a Bayesian hierarchical multifidelity model. *J. Fluid Mech.* **964**, A13.

RICO-MARTÍNEZ, R., KRISCHER, K., KEVREKIDIS, I.G., KUBE, M.C. & HUDSON, J.L. 1992 Discrete- vs. continuous-time nonlinear signal processing of Cu electrodissolution data. *Chem. Engng Commun.* **118** (1), 25–48.

ROHLFS, W., RIETZ, M. & SCHEID, B. 2018 Wavemaker: the three-dimensional wave simulation tool for falling liquid films. *SoftwareX* **7**, 211–216.

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.

SHKLYAEV, S. & NEPOMNYASHCHY, A. 2017 *Longwave Instabilities and Patterns in Fluids*. Birkhäuser.

SHLANG, T. & SIVASHINSKY, G.I. 1982 Irregular flow of a liquid film down a vertical column. *J. Phys.* **43** (3), 459–466.

SIRIGNANO, J. & MACART, J.F. 2023 Deep learning closure models for large-eddy simulation of flows around bluff bodies. *J. Fluid Mech.* **966**, A26.

https://doi.org/10.1017/jfm.2023.868 Published online by Cambridge University Press

Sonoda, T., Liu, Z., Itoh, T. & Hasegawa, Y. 2023 Reinforcement learning of control strategies for reducing skin friction drag in a fully developed turbulent channel flow. *J. Fluid Mech.* **960**, A30.

Takens, F. 1981 *Detecting Strange Attractors in Turbulence*, pp. 366–381. Springer Berlin Heidelberg.

Vlachas, P.R., Arampatzis, G., Uhler, C. & Koumoutsakos, P. 2022 Multiscale simulations of complex systems by learning their effective dynamics. *Nat. Mach. Intell.* **4** (4), 359–366.

Vlachas, P.R., Byeon, W., Wan, Z.Y., Sapsis, T.P. & Koumoutsakos, P. 2018 Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proc. R. Soc.* A*: Math. Phys. Engng Sci.* **474** (2213), 20170844.

Whitney, H. 1936 Differentiable manifolds. *Ann. Math.* **37** (3), 645–680.

Xu, D., Wang, J., Yu, C. & Chen, S. 2023 Artificial-neural-network-based nonlinear algebraic models for large-eddy simulation of compressible wall-bounded turbulence. *J. Fluid Mech.* **960**, A4.

Xuan, A. & Shen, L. 2023 Reconstruction of three-dimensional turbulent flow structures using surface measurements for free-surface flows based on a convolutional neural network. *J. Fluid Mech.* **959**, A34.

Zhang, B.F., Fan, D.W. & Zhou, Y. 2023 Artificial intelligence control of a low-drag Ahmed body using distributed jet arrays. *J. Fluid Mech.* **963**, A3.