Computer Methods in Applied Mechanics and Engineering 429 (2024) 117149

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Data-driven identification of stable sparse differential operators using constrained regression

Aviral Prakash *, Yongjie Jessica Zhang

Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ARTICLE INFO

Dataset link: https://github.com/CMU-CBML/Py_SLDO

Keywords:
Constrained regression
Linear stability
Differential operators
System identification
Scientific machine learning
Stencil-based sparsity

ABSTRACT

Identifying differential operators from data is essential for the mathematical modeling of complex physical and biological systems where massive datasets are available. These operators must be stable for accurate predictions for dynamics forecasting problems. In this article, we propose a novel methodology for learning differential operators that are theoretically linearly stable and have sparsity patterns of common discretization schemes. These differential operators are obtained by solving a constrained regression problem, involving local constraints to ensure the linear stability of the global dynamical system. We further extend this approach for learning nonlinear differential operators by determining linear stability constraints for linearized equations around an equilibrium point. The applicability of the proposed method is demonstrated for both linear and nonlinear partial differential equations such as 1-D scalar advection-diffusion equation, 1-D Burgers equation and 2-D advection equation. The results indicated that solutions to constrained regression problems with linear stability constraints provide accurate and linearly stable sparse differential operators.

1. Introduction

Mathematical models for predicting physical systems rely on well-defined partial differential equations (PDEs) or ordinary differential equations (ODEs) that govern the spatio-temporal dynamics. Traditionally, these equations have been defined based on physical insights from domain experts. This approach has prohibited simulations for many complex systems, such as those in biology and finance, where obtaining such equations through insights may not be possible. With advancements in machine learning techniques and the availability of large-scale datasets, there has been a widespread interest in data-driven simulation and modeling [1–6]. This interest has led to mathematical techniques that utilize such datasets with partial physical information to infer physical systems and improve existing models.

For several scenarios, such as physical and biological dynamical systems where PDEs and ODEs are not readily available, there has been growing interest in techniques for system identification [7–11]. The first step of modeling these complex systems is to determine PDEs from data [8,12,13]. The next step is solving these learned PDEs to predict system dynamics. The typical approach to solving these identified equations is using numerical methods such as finite differences [14], finite volumes [15] or finite elements [16]. These numerical methods have been commonly used for solving PDEs as these have provable theoretical estimates of accuracy and stability. However, these methods are often tailored to maintain accuracy and stability for specific problems. Furthermore, the solution of PDEs using these methods also requires the development of a comprehensive and efficient codebase, especially if the targeted application is complex and computationally expensive. When system identification techniques [8,12,13] are applied

* Corresponding author.
E-mail address: aviralp@andrew.cmu.edu (A. Prakash).

https://doi.org/10.1016/j.cma.2024.117149
Received 30 April 2024; Received in revised form 8 June 2024; Accepted 9 June 2024
Available online 26 June 2024

for identifying PDEs, accuracy and stability considerations for the spatial and temporal discretization of learned PDEs to obtain a valid solution are typically not addressed. Instead, appropriate experience and insights from users are expected to address this selection. In such situations, it is desirable to determine sparse discrete differential operators directly using data and modern machine learning methods, which will be discussed in detail in this article. These differential operators can also enable nonintrusive reduced order modeling where, despite the knowledge of underlying PDE, the knowledge and access to the discretization scheme are unavailable [17].

In contrast to approaches that directly provide full order [11,18] or reduced order [19] dynamical systems from data, the goal of this article is to determine sparse stencil-based differential operators from data, which is intimately connected to learning spatial discretization from data [20]. This research area of system identification has gained significant interest over the years. Several popular approaches use artificial neural networks (ANNs) for discrete representations of systems [20–22]. Despite their popularity and accuracy for different applications, the typical black-box nature of ANNs often discourages interpretability [23], which is desired when performing theoretical analysis to identify the accuracy and stability properties of the method. An in-depth study of these properties is essential for gauging the performance of learned differential operators for scenarios not included in the training dataset. Furthermore, these nonlinear learned differential operators could have a high evaluation cost, prohibiting the scalability of such methods for large-scale physical systems. Standard numerical methods often result in sparse linear systems with a lower computational overhead. Decades of research in solving such systems have resulted in sparse linear system solvers that provide high accuracy with a low memory footprint and enable scalability and portability to different hardware architectures. These ANN-based discrete operators may not allow the efficient use of such sparse linear system solvers, which could be required while using implicit time integration for stiffer dynamics.

In contrast to previous approaches, which learn noninterpretable differential operators, recent work [24] has focused on obtaining interpretable sparse stencils from data for linear PDEs. They posed this as a regression problem where a local solution stencil is learned from the data. In a follow-up work [25], this approach was extended for time-dependent problems and nonlinear PDEs while using a strategy [13] to identify stencils and regularization parameters for stable learned differential operators. Recent work on the adjacency-based determination of differential operators [17,26] has focused on a similar approach for identifying sparse differential operators and demonstrating their application for nonintrusive reduced order modeling. The accuracy of these methods can be adjusted by selecting the appropriate solution stencil sizes, machine learning techniques and regularization parameters. However, despite high accuracy within the training dataset, the numerical stability of these discrete solution representations is essential to ensure that numerical approximation errors do not grow in time. All these methods, including ANN-based methods, do not theoretically guarantee stability even for linear systems. Instead, feasible stable solutions may only be constrained to scenarios within the validation dataset without any stability guarantees for dynamics forecasting scenarios outside the validation dataset. For the reasons mentioned above, there is an immense need for approaches that determine differential operators from data while providing theoretical stability guarantees. Our experience with learning differential operators through state-of-the-art approaches that use regression to obtain differential operators [17,24] indicated that these methods often learn unstable differential operators for linear PDEs and may not reliably perform even within the training dataset. While this issue is partly addressed in [25] using the stability selection procedure [27], this approach can be computationally expensive for a system with many degrees of freedom and cannot theoretically guarantee stability. Recently, several research methodologies have been proposed to tackle the issue of stability in the context of dynamical systems [18,28,29] and reduced order modeling using operator inference [30,31], however, no such methodologies exist for learning stencil-based sparse differential operators that are stable.

In this article, we propose a novel approach for learning sparse differential operators that are provably linearly stable. This approach relies on a set of local conditions for differential operators derived using the stability theory for linear dynamical systems. These conditions are incorporated as inequality constraints in the regression problem to determine the unknown differential operators. The resulting constrained regression problem is solved using a sequential least squares programming optimizer. We further extend this method for learning nonlinear differential operators by formulating constraints based on linearized equations obtained using Taylor-series expansion around an equilibrium point. The applicability of the proposed method for learning stable differential operators is demonstrated by comparing the results against the standard regression-based approach for multiple linear and nonlinear PDEs: 1-D scalar advection-diffusion equation, 1-D Burgers equation and 2-D advection equation. The proposed approach targets identifying suitable differential operators while using the known form of PDE to provide stability constraints. Therefore, this approach differs from other system identification approaches that identify an unknown PDE.

The outline of this article is as follows: Section 2 discusses the relevant mathematical background on differential operators for PDEs and stability theory for linear differential operators; Section 3 first discusses the standard approach for learning sparse differential operators from data. This section also gives the mathematical details for learning stable sparse differential operators from data and an extension of this approach to nonlinear equations; Section 4 includes numerical experiments for three PDEs and demonstrates the applicability of the proposed approach for learning stable differential operators from data; Section 5 concludes this article by highlighting the main contributions and mentioning directions for future research.

2. Mathematical background

In this section, we first introduce the theory of obtaining the semi-discrete form of PDEs and then discuss the linear stability of differential operators in this semi-discrete form.

2.1. Differential operators for PDEs

We restrict the analysis and mathematical formulation to 1-D PDEs, although these concepts can be generalized to higher-dimensional PDEs. Consider a 1-D PDE of the following form:

$$\frac{\partial u}{\partial t} + \mathcal{F}(u) = 0, \tag{1}$$

where $\mathcal{F}(\cdot)$ is a continuous differential operator, $u : \Omega \times [0, T] \to \mathbb{R}$ is the PDE solution, Ω is the simulation domain and T is the final simulation time. Boundary conditions often accompany these PDEs on the boundaries Γ. Using the method of lines with a suitable spatial discretization, the semi-discrete form of this PDE is obtained as

$$\frac{d\boldsymbol{u}}{dt} + \boldsymbol{F}(\boldsymbol{u}) = 0, \tag{2}$$

where $\boldsymbol{u} \in \mathbb{R}^n$ is the discrete solution field, $\boldsymbol{F} : \mathbb{R}^n \to \mathbb{R}^n$ is a differential operator and n is the number of degrees of freedom. These degrees of freedom often correspond to different locations on the discretized domain, called the simulation grid Ω^h, depending on the spatial discretization approach. The differential operator can be linear or nonlinear, depending on the PDE under consideration. We make this distinction by decomposing the differential operator into a linear component $\boldsymbol{L}$ and a nonlinear component $\hat{\boldsymbol{N}}$ such that Eq. (2) is simplified as

$$\frac{d\boldsymbol{u}}{dt} + \boldsymbol{L}\boldsymbol{u} + \hat{\boldsymbol{N}}(\boldsymbol{u}) = 0. \tag{3}$$

The form of these linear and nonlinear operators depends on the numerical method chosen to discretize the PDE. Typical discretization methods such as finite difference, finite volume or finite element methods result in a sparse operator, implying that the linear and nonlinear operators are applied to localized degrees of freedom, commonly called solution stencil. The sparsity of these operators allows efficient storage and faster computation of the PDE solution field. The discretized PDE at the ith degree of freedom can be considered an ODE problem

$$\frac{du_i}{dt} + (\boldsymbol{L}^i)^T \boldsymbol{u}_{\Omega_i^l} + \hat{\boldsymbol{N}}^i(\boldsymbol{u}_{\Omega_i^n}) = 0, \tag{4}$$

where $\boldsymbol{L}^i \in \mathbb{R}^{s_l}$ is the local linear operator which contributes to the ith row of $\boldsymbol{L}$, superscript T indicates the transpose, s_l is the size of the local linear stencil, $\hat{\boldsymbol{N}}^i : \mathbb{R}^{s_n} \to \mathbb{R}$ is the local nonlinear operator and s_n is the size of the local nonlinear stencil. The solution stencils $\boldsymbol{u}_{\Omega_i^l}$ and $\boldsymbol{u}_{\Omega_i^n}$ are selected based on local degrees of freedom. In the context of this article, we consider polynomial nonlinearity allowing transformation of the nonlinear term in Eq. (4) to a matrix–vector product. The resulting equation is

$$\frac{du_i}{dt} + (\boldsymbol{L}^i)^T \boldsymbol{u}_{\Omega_i^l} + (\boldsymbol{N}^i)^T \boldsymbol{z}(\boldsymbol{u}_{\Omega_i^n}) = 0, \tag{5}$$

where $\boldsymbol{N}^i \in \mathbb{R}^{s_n}$ is the nonlinear differential operator in a vector form, s_n is the nonlinear stencil size and $\boldsymbol{z}(\boldsymbol{u}_{\Omega_i^n}) \in \mathbb{R}^{s_n}$ represents nonlinear products of $\boldsymbol{u}_{\Omega_i^n}$. To elucidate this notation, we consider an example of the 1-D viscous Burgers equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0 \tag{6}$$

with periodic boundary conditions. For this example, we select a 1st-order backward difference for the nonlinear term and a 2nd-order centered finite difference for the linear term for discretizing the system on n uniformly spaced grid nodes. This discretization gives the following semi-discrete form

$$\frac{du_i}{dt} + u_i\frac{u_i - u_{i-1}}{\Delta x} - \nu\frac{u_{i+1} - 2u_i + u_{i-1}}{(\Delta x)^2} = 0. \tag{7}$$

This semi-discrete form can be written as Eq. (5) with

$$\boldsymbol{N}^i = \frac{1}{\Delta x}[-1, 1, 0]^T \quad \text{and} \quad \boldsymbol{L}^i = \frac{\nu}{(\Delta x)^2}[-1, 2, -1]^T, \tag{8}$$

where $\boldsymbol{u}_{\Omega_i^l} = [u_{i-1}, u_i, u_{i+1}]^T$ and $\boldsymbol{z}(\boldsymbol{u}_{\Omega_i^n}) = [u_i u_{i-1}, u_i^2, u_i u_{i+1}]^T$. Note that the nonlinear operator and the associated stencil can also be written in other ways. In this article, stencils $\boldsymbol{u}_{\Omega_i^l}$ and $\boldsymbol{z}(\boldsymbol{u}_{\Omega_i^n})$ are taken to be symmetrical about the ith degree of freedom, which would include ghost degrees of freedom if needed for handling different boundary conditions. Typically, linear and nonlinear operators and their corresponding stencils are designed based on accuracy and stability analysis. For example, a 2nd-order centered finite difference is unstable with 1st-order forward Euler method for time integration for hyperbolic problems such as 1-D advection equation [14].

2.2. Stability of differential operators

The stability of a numerical scheme is commonly assessed by performing a stability analysis for some canonical linear problems. A well-known strategy for stability analysis of PDEs is to perform Von Neumann analysis [32,33], which involves decomposing the solution u as the sum of spectral modes and assessing the growth or decay of these modes. Another similar strategy for stability analysis is to identify a semi-discrete form of the PDE by applying the method of lines with suitable spatial discretization [14]

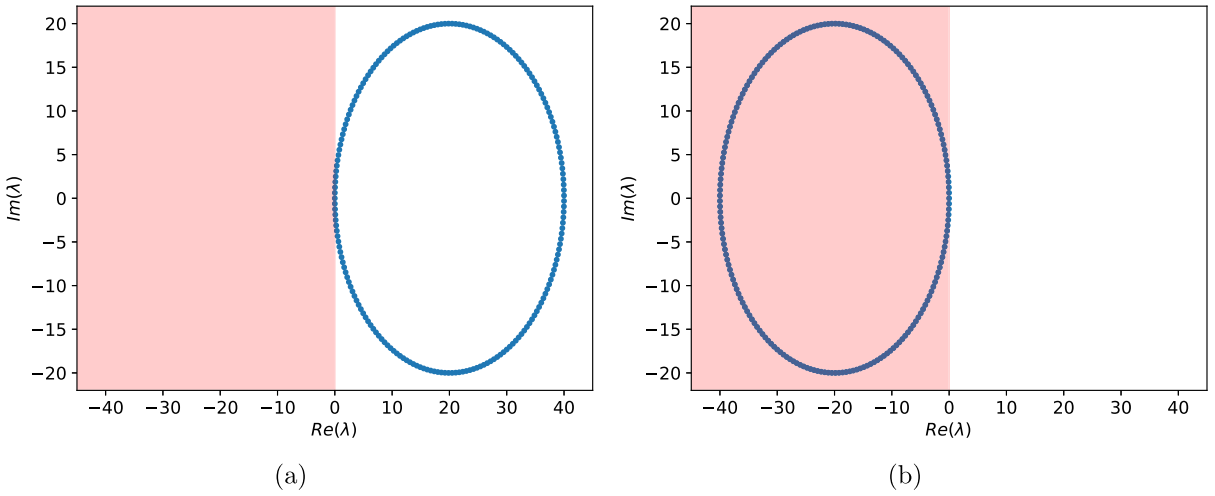

Fig. 1. Eigenvalues λ of the differential operator $\boldsymbol{L}$ obtained using (a) the 1st-order forward difference and (b) the 1st-order backward difference. The shaded area indicates the stable region.

to obtain a set of linear ODEs, which is used to perform stability analysis following common strategies in dynamical system literature [34]. We will follow the latter approach, especially as it will allow us to assess the stability property of differential operators learned from data. Consider a system of linear ODEs

$$\frac{d\boldsymbol{u}}{dt} = \boldsymbol{A}\boldsymbol{u}, \tag{9}$$

where $\boldsymbol{u} = [u_1, u_2, \cdots u_n]^T$ and $\boldsymbol{A} : \mathbb{R}^n \to \mathbb{R}^n$.

Definition 2.1. The ODE system in Eq. (9) is referred to as stable if and only if the real part of all eigenvalues of $\boldsymbol{A}$ are nonpositive. If any eigenvalue of $\boldsymbol{A}$ is positive, Eq. (9) is called an unstable system.

Generally speaking, $\boldsymbol{A}$ is referred to as asymptotically stable if it has negative eigenvalues. As the above definition holds for linear ODE systems, we refer to this property as linear stability. Such linear systems are also often referred to as globally stable as the stability properties hold irrespective of initial system states. For nonlinear ODE systems, which may arise after discretizing nonlinear PDEs, assessment of global stability may not be possible. Therefore, the local stability can be assessed by linearizing the system around an equilibrium point and performing a linear stability analysis of the resulting linear equations. We illustrate the applicability of stability analysis by considering the 1-D scalar advection equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \tag{10}$$

where $c > 0$ is the advection velocity. To illustrate the role of a suitable selection of spatial discretization on stability, we discretize this equation to obtain a set of ODEs of the form Eq. (9). For this example problem, we select two spatial discretization schemes, the 1st-order backward difference and the 1st-order forward difference, with uniformly spaced $n = 201$ degrees of freedom. We get a linear system with the equation for the ith interior degree of freedom as

$$\frac{du_i}{dt} + c(\boldsymbol{L}^i)^T \boldsymbol{u}_{\Omega_i^l} = 0, \tag{11}$$

where $\boldsymbol{u}_{\Omega_i^l} = [u_{i-1}, u_i, u_{i+1}]^T$, $\boldsymbol{L}^i = \frac{1}{\Delta x}[-1, 1, 0]^T$ for the 1st-order backward difference and $\boldsymbol{L}^i = \frac{1}{\Delta x}[0, -1, 1]^T$ for the 1st-order forward difference. Note that the boundary conditions will modify the definition of $\boldsymbol{L}^i$ for degrees of freedom associated with the periodic boundaries. The eigenvalues of the linear systems $-\boldsymbol{L}$ for these two spatial discretizations are shown in Fig. 1 . This figure shows the backward difference yielding a linear system with all eigenvalues in the stable region, whereas the forward difference results in eigenvalues outside the stable region. This stability plot implies that backward differences yield a stable system of equations, whereas forward differences do not. This observation and analysis is common in the literature [14]. The role of this example is solely to demonstrate how we will analyze the eigenvalues to assess the stability properties of differential operators learned from data. Even though a similar theory may not guarantee global stability for nonlinear differential equations, the local linear stability of such equations will be considered around an equilibrium point.

3. Learning differential operators from data

With the increasing availability of simulation data and advancements in modern machine-learning techniques, there is a growing interest in determining differential equations from data. Several proposed methods [20,22,23] utilize artificial neural network-based

architectures for representing discretized PDEs. On the other hand, interpretable sparse differential operators are determined from data using regression-based methods recently proposed in [24,25]. The mathematical formulation of these methods is similar to the approach explained in Section 3.1. All these methods do not theoretically guarantee stability, even for linear differential equations. This article proposes a novel approach incorporating suitable constraints to provide linearly stable differential operators from data. The mathematical formulation of this proposed approach is explained in Section 3.2.

3.1. Regression-based approach for learning differential operator

In this article, we restrict the discussion to using least squares regression as they allow for a more straightforward and interpretable representation. Using this approach, the differential operators in Eq. (4) are obtained by solving the following regression problem:

Given high-fidelity data, $\boldsymbol{u} \in \mathbb{R}^n$ and $\dot{\boldsymbol{u}} = \frac{d\boldsymbol{u}}{dt} \in \mathbb{R}^n$ at time instances $t = t_j$ for $j = 1, \cdots, n_t$, find the optimal operators $\boldsymbol{L}$ and $\boldsymbol{N}$, subject to the objective function

$$\min_{\tilde{\boldsymbol{L}},\, \tilde{\boldsymbol{N}}} \left|\left|\dot{\boldsymbol{u}}(t) + \tilde{\boldsymbol{L}}\boldsymbol{u}(t) + \tilde{\boldsymbol{N}}\boldsymbol{z}(\boldsymbol{u}(t))\right|\right|_2^2. \tag{12}$$

The cost of the regression problem scales as $O(n_t n^2)$, which makes it expensive. Furthermore, solving learned differential equations with dense matrices for $\boldsymbol{L}$ and $\boldsymbol{N}$ will have significantly higher computational overhead than solving equations obtained using standard discretization schemes, which typically have a sparser stencil. Inspired by determining more practical and efficient differential operators from data, we pose the smaller regression problems for each degree of freedom instead. The resulting regression problem is defined as follows:

Given high-fidelity data, $\boldsymbol{u} \in \mathbb{R}^n$ and $\dot{\boldsymbol{u}} = \frac{d\boldsymbol{u}}{dt} \in \mathbb{R}^n$ at time instances $t = t_j$ for $j = 1, \cdots, n_t$ and solution stencils for linear and nonlinear operators $\boldsymbol{u}_{\Omega_i^l} \in \mathbb{R}^{s_l}$ and $\boldsymbol{u}_{\Omega_i^n} \in \mathbb{R}^{s_n}$, find the optimal operators $\boldsymbol{L}^i$ and $\boldsymbol{N}^i$, subject to the objective function

$$\min_{\tilde{\boldsymbol{L}}^i,\, \tilde{\boldsymbol{N}}^i} \left|\left|\dot{u}_i(t) + (\tilde{\boldsymbol{L}}^i)^T \boldsymbol{u}_{\Omega_i^l}(t) + (\tilde{\boldsymbol{N}}^i)^T z(\boldsymbol{u}_{\Omega_i^n}(t))\right|\right|_2^2 \quad \forall\, i = 1, \cdots, n. \tag{13}$$

We can determine differential operators by solving this regression problem solely using high-fidelity solution data and its time derivative. The time derivative of the solution can further be extracted from time-resolved high-fidelity data using finite differences. This approach involves solving a regression problem for each point in the domain. Therefore, assuming data at n_t timesteps are available, then the cost of the regression problem at each degree of freedom is $O(n_t(s_l + s_n)^2)$. The complexity of determining the differential operator is $O(n_t n(s_l + s_n)^2)$. For a large stencil size, that is $n \approx s_l + s_n$, the cost can scale as $O(n_t n^3)$, making this problem computationally expensive. However, as typically linear and nonlinear operators are designed to be sparse, we can work with localized sparse stencils such that $n >> s_l + s_n$, resulting in a less expensive regression problem. Furthermore, the boundary conditions can be enforced by appropriately selecting local stencil and setting constraints on the local differential operators.

The training dataset should consist of high-fidelity data available at several time instances or simulation conditions to better pose the regression problem. However, even with large amounts of data, the model could overfit the available data and not generalize to simulation conditions outside the training dataset. Furthermore, the linear system we obtain in this regression problem could also be rank deficient, leading to undefined solutions [24]. These issues are overcome by augmenting the least squares regression problem in Eq. (13) with a regularization term. The resulting objective function is

$$\min_{\tilde{\boldsymbol{L}}^i,\, \tilde{\boldsymbol{N}}^i} \left|\left|\dot{u}_i(t) + (\tilde{\boldsymbol{L}}^i)^T \boldsymbol{u}_{\Omega_i^l}(t) + (\tilde{\boldsymbol{N}}^i)^T z(\boldsymbol{u}_{\Omega_i^n}(t))\right|\right|_2^2 + \beta_1 ||\tilde{\boldsymbol{L}}^i||_2^2 + \beta_2 ||\tilde{\boldsymbol{N}}^i||_2^2 \quad \forall\, i = 1, \cdots, n, \tag{14}$$

where β_1 and β_2 are regularization constants for the unknown linear and nonlinear operators. Similarly, if the differential equations are linear, then the objective function reduces to

$$\min_{\tilde{\boldsymbol{L}}^i} \left|\left|\dot{u}_i(t) + (\tilde{\boldsymbol{L}}^i)^T \boldsymbol{u}_{\Omega_i^l}(t)\right|\right|_2^2 + \beta_1 ||\tilde{\boldsymbol{L}}^i||_2^2 \quad \forall\, i = 1, \cdots, n. \tag{15}$$

By solving the regularized least squares regression problem, we can obtain the sparse operators $\boldsymbol{L}^i$ and $\boldsymbol{N}^i$. The choice of regularization here corresponds to the Tikhonov regularization [35] that is commonly used in the literature, resulting in a ridge regression problem. Other regression approaches, such as Lasso [36] and Elastic-net [37], can also be used. In our experience, these appear to give similar results to ridge regression, especially with a sparser solution stencil. As shown for reduced order modeling using operator inference, appropriate regularization method can improve the stability properties of the learned dynamical system [38,39]. Similarly, the choice of stencil size, regularization approach and regularization parameter significantly influence the stability properties of the learned differential operator. There is some recent but limited work on similar sparse differential operators [24,26] but without a detailed analysis of the impact of stencil size and regularization on stability. The method proposed in [25] uses statistical stability technique [13] and solves multiple similar regression problems to determine the ideal stencil size and regularization parameter that yields accurate results for a specific validation dataset window. This technique smartly selects different stencil combinations, including asymmetrical stencils, to ensure superior stability behavior. However, obtaining stable operators by solving standard unconstrained regression problems may not always be possible. Therefore, a particular combination of these parameters might yield accurate results in the validation dataset but still be theoretically unstable, yielding unphysical results for dynamics forecasting. In Section 4, we will demonstrate this behavior through several numerical experiments. Therefore, a strategy is needed to theoretically guarantee the stability of learned differential operators.

3.2. Constrained regression-based approach to learning stable differential operator

This section describes a methodology to learn differential operators from data while ensuring stability. As stability is more precisely defined for linear operators, we formulate the method for linear operators and then extend the method for nonlinear operators. Consider a discretized PDE of the form

$$\frac{d\boldsymbol{u}}{dt} + \boldsymbol{L}\boldsymbol{u} = 0 \tag{16}$$

with unknown differential operator $\boldsymbol{L}$. We can obtain a stable differential operator by formulating a constrained regression problem:

Given high-fidelity data, $\boldsymbol{u} \in \mathbb{R}^n$ and $\dot{\boldsymbol{u}} = \frac{d\boldsymbol{u}}{dt} \in \mathbb{R}^n$ at time instances $t = t_j$ for $j = 1, \cdots, n_t$, find the optimal operator $\boldsymbol{L}$, subject to the objective function

$$\min_{\tilde{\boldsymbol{L}}} \left|\left|\dot{\boldsymbol{u}}(t) + \tilde{\boldsymbol{L}}\boldsymbol{u}(t)\right|\right|_2^2 \tag{17}$$

such that

$$\boldsymbol{L} \succeq 0. \tag{18}$$

The constraint in Eq. (18) ensures that $\boldsymbol{L}$ is a positive semi-definite matrix, which implies that $\boldsymbol{L}$ has either positive or zero eigenvalues and is stable following Definition 2.1. This regression problem in Eq. (17) and Eq. (18) can be considered a semi-definite program [40], which is more expensive to compute than Eq. (12). Similar strategies for ensuring stability have been considered in research areas such as dynamical systems [18,28,29] and reduced order modeling [30,31,41]. These approaches are developed for learning dense operators, which are ensured to be positive definite using approaches like eigenvalue reassignment [41], regression constraints [18,30,31] or matrix parameterization [28,29].

Solving such a constrained regression problem will be much more expensive for large n compared to solving a standard regression problem and would not provide us $\boldsymbol{L}$ that follows the sparsity pattern obtained using standard PDE discretizations. Therefore, we pose the regression problem differently to learn a localized differential operator $\boldsymbol{L}^i$ that acts on a local solution stencil $\boldsymbol{u}_{\Omega_i^l} \in \mathbb{R}^{s_l}$ and is subject to local constraints that can ensure the stability of the global dynamical system. This formulation satisfies the objective function

$$\min_{\tilde{\boldsymbol{L}}^i} \left|\left|\dot{u}_i(t) + (\tilde{\boldsymbol{L}}^i)^T \boldsymbol{u}_{\Omega_i^l}(t)\right|\right|_2^2 \quad \forall\, i = 1, \cdots, n, \tag{19}$$

where these localized differential operators $\boldsymbol{L}^i$ are assembled to provide a differential operator $\boldsymbol{L}$ that should be positive semi-definite. To describe this regression problem mathematically, we reframe the constraint locally on $\boldsymbol{L}^i$ to ensure the assembled $\boldsymbol{L}$ is positive semi-definite. To achieve this, we utilize the well-known Gershgorin circle theorem [42].

Theorem 3.1 (*Gershgorin Circle Theorem*). *Considering a complex matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ with the ij^{th} element as a_{ij}. Every eigenvalue λ of matrix $\boldsymbol{A}$ satisfies*

$$|\lambda - a_{ii}| \leq \sum_{j\neq i} |a_{ij}| \quad \forall \quad i \in 1, 2, \cdots, n. \tag{20}$$

Applying the Gershgorin circle theorem to the differential operator $\boldsymbol{L}$, every eigenvalue λ of $\boldsymbol{L}$ must satisfy

$$|\lambda - L_i^i| \leq \sum_{j\neq i} |L_j^i| \quad \forall\, i \in 1, 2, \cdots, n, \tag{21}$$

where L_j^i is the jth element of $\boldsymbol{L}^i$ or the ij^{th} element of $\boldsymbol{L}$ in common matrix notation. This inequality indicates the bounds for eigenvalues of $\boldsymbol{L}$ relating to individual terms of local operators $\boldsymbol{L}^i$. Therefore, $\boldsymbol{L}$ is guaranteed to be positive semi-definite if

$$L_i^i \geq \sum_{j\neq i} |L_j^i| \quad \forall\, i \in 1, 2, \cdots, n, \tag{22}$$

which is a constraint on values of the local operator $\boldsymbol{L}^i$. Using this result, we can pose a constrained regression problem for each degree of freedom:

Given high-fidelity data, $\boldsymbol{u} \in \mathbb{R}^n$ and $\dot{\boldsymbol{u}} = \frac{d\boldsymbol{u}}{dt} \in \mathbb{R}^n$ at time instances $t = t_j$ for $j = 1, \cdots, n_t$ and solution stencil $\boldsymbol{u}_{\Omega_i^l} \in \mathbb{R}^{s_l}$, find the differential operator $\boldsymbol{L}^i$, subject to the objective function

$$\min_{\tilde{\boldsymbol{L}}^i} \sum_{j=1}^{n_t} \left|\left|\dot{u}_i(t) + (\tilde{\boldsymbol{L}}^i)^T \boldsymbol{u}_{\Omega_i^l}(t)\right|\right|_2^2 \tag{23}$$

such that

$$L_i^i - \sum_{j\neq i} |L_j^i| \geq 0 \quad \forall\, i = 1, \cdots, n. \tag{24}$$

This constraint ensures that we can obtain local linear operators from data while ensuring that the assembled global linear operator is stable. By solving this constrained regression problem, we obtain a differential operator that could reflect the sparsity pattern of the spatial discretization method while ensuring the stability of the assembled linear systems. To extend this approach for nonlinear

PDEs, we determine the stability constraints from the corresponding linearized equations. These linearized differential equations are obtained by applying Taylor series approximation on Eq. (4) around an equilibrium point $\boldsymbol{u}^0$. This approach provides us with a linearized ODE problem

$$\frac{d\boldsymbol{u}}{dt} + \boldsymbol{N}^L\boldsymbol{u} = 0, \tag{25}$$

where $\boldsymbol{N}^L$ is the linear term arising as a result of linearization and depends on $\boldsymbol{N}$, $\boldsymbol{L}$ and $\boldsymbol{u}^0$. The linearized equations are stable if

$$\boldsymbol{N}^L \succ 0, \tag{26}$$

implying $\boldsymbol{N}^L$ should be a positive definite matrix. Note that $\boldsymbol{N}^L$ having a zero eigenvalue may not imply linear stability as this stability characteristic would depend on the truncated terms during the linearization. With this constraint, the localized stencil regression problem transforms to:

Given high-fidelity data, $\boldsymbol{u} \in \mathbb{R}^n$ and $\dot{\boldsymbol{u}} = \frac{d\boldsymbol{u}}{dt} \in \mathbb{R}^n$ at time instances $t = t_j$ for $j = 1, \cdots, n_t$ and solution stencils for the two operators $\boldsymbol{u}_{\Omega_i^l} \in \mathbb{R}^{s_l}$ and $\boldsymbol{u}_{\Omega_i^n} \in \mathbb{R}^{s_n}$, find differential operators $\boldsymbol{L}^i$ and $\boldsymbol{N}^i$, subject to the objective function

$$\min_{\tilde{\boldsymbol{L}}^i,\, \tilde{\boldsymbol{N}}^i} \left|\left|\dot{u}_i(t) + (\tilde{\boldsymbol{L}}^i)^T\boldsymbol{u}_{\Omega_i^l}(t) + (\tilde{\boldsymbol{N}}^i)^T\boldsymbol{z}(\boldsymbol{u}_{\Omega_i^n}(t))\right|\right|_2^2 \tag{27}$$

such that

$$N_i^{L,i} - \sum_{j\neq i} |N_j^{L,i}| > 0 \quad \forall\, i = 1, \cdots, n, \tag{28}$$

which is the localized form of Eq. (26) where $N_j^{L,i}$ is the jth element of $\boldsymbol{N}^{L,i}$ and $\boldsymbol{N}^{L,i}$ is assembled to form $\boldsymbol{N}^L$.

This constraint ensures that eigenvalues of $\boldsymbol{N}^L$ are positive, indicating linear stability for learned nonlinear operators around an equilibrium point. For both linear and nonlinear equations, this notion of stability classifies the resulting ODEs, or the semi-discrete form of PDEs, as linearly stable or unstable. These stable ODEs will lead to stable solutions if the eigenvalues lie within the absolute stability region of the selected time integration scheme. Note that the stencils $\boldsymbol{u}_{\Omega_i^n}$ and $\boldsymbol{z}(\boldsymbol{u}_{\Omega_i^n})$ should be symmetric about the ith degree of freedom. Asymmetric stencils can also be considered. However, such a selection should be approached with caution as some asymmetric stencils cannot mathematically yield stable operators.

4. Numerical results

In this section, we assess the stability and performance of the differential operator obtained using the two approaches discussed in Section 3:

- Solving the standard regression problem Eq. (14) or Eq. (15) which gives us learned differential operators (LDOs); and
- Solving the constrained regression problem Eq. (23) or Eq. (27) which gives us stable learned differential operators (S-LDOs).

Algorithm 1 Solving Eq. (15) to obtain LDOs for 1-D scalar advection-diffusion equation. The notations follow the example in Section 4.1.

Input: Generated data ($\boldsymbol{u}(t_l)$ and $\dot{\boldsymbol{u}}(t_l)$ for $l = 1, 2, \cdots, n_t$), stencil sizes (s_{l_1} and s_{l_2}), simulation parameter (c and ν) and regularization parameter (β_1)
Output: Operators $\boldsymbol{L}_1^m$ and $\boldsymbol{L}_2^m$

for `i = 1:n` **do** ▷ Loop over all degrees of freedom
- $\boldsymbol{b} \leftarrow [\dot{u}_i(t_1)\ \dot{u}_i(t_2)\ \cdots\ \dot{u}_i(t_{n_t})]^T$
- $s_t \leftarrow s_{l_1} + s_{l_2}$ ▷ s_t is the total number of unknowns
- $s_{l_1}^h \leftarrow (s_{l_1} - 1)/2$ and $s_{l_2}^h \leftarrow (s_{l_2} - 1)/2$ ▷ Half of stencil sizes
- $\boldsymbol{A} \leftarrow$ `zeros`(n_t, s_t) ▷ Initialize $\boldsymbol{A} \in \mathbb{R}^{n_t \times s_t}$ as a zero matrix
- $A_{j,1:s_{l_1}} \leftarrow c[u_{i-s_{l_1}^h}(t_j)\ \cdots\ u_i(t_j)\ \cdots\ u_{i+s_{l_1}^h}(t_j)]$ ▷ Assign advection terms where $A_{j,k}$ is the jk^{th} element of $\boldsymbol{A}$
- $A_{j,s_{l_1}+1:s_t} \leftarrow -\nu[u_{i-s_{l_2}^h}(t_j)\ \cdots\ u_i(t_j)\ \cdots\ u_{i+s_{l_2}^h}(t_j)]$ ▷ Assign diffusion terms
- $\boldsymbol{x} \leftarrow (\boldsymbol{A}^T\boldsymbol{A} + \beta_1\boldsymbol{I})^{-1}\boldsymbol{A}^T\boldsymbol{b}$ ▷ $\boldsymbol{x}$ can be obtained by solving the linear system: $(\boldsymbol{A}^T\boldsymbol{A} + \beta_1\boldsymbol{I})\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$
- $\boldsymbol{L}_1^{i,m} \leftarrow x_{1:s_{l_1}}$ and $\boldsymbol{L}_2^{i,m} \leftarrow x_{s_{l_1}+1:s_t}$

end for

$\boldsymbol{L}_1^m \leftarrow \boldsymbol{L}_1^{i,m}$ and $\boldsymbol{L}_2^m \leftarrow \boldsymbol{L}_2^{i,m}$ ▷ Assemble local operators to give global operators or store local operators directly as a sparse matrix

The constrained regression problem for obtaining S-LDOs is solved using a sequential least squared programming optimizer available in [43]. For all the numerical experiments, the tolerance of this optimizer is set to 10^{-6} to ensure that the cost of obtaining S-LDOs is comparable to the cost of obtaining LDOs. At such lower tolerances, the performance of S-LDOs is sensitive to the stencil

Algorithm 2 Solving Eq. (23) and Eq. (24) to obtain S-LDOs for 1-D scalar advection-diffusion equation. The notations follow the example in Section 4.1.

Input: Generated data ($\boldsymbol{u}(t_l)$ and $\dot{\boldsymbol{u}}(t_l)$ for $l = 1, 2, \cdots, n_t$), stencil sizes (s_{l_1} and s_{l_2}) and simulation parameter (c and ν)
Output: Operators $\boldsymbol{L}_1^m$ and $\boldsymbol{L}_2^m$

for `i = 1:n` **do** ▷ Loop over all degrees of freedom
- $\boldsymbol{b} \leftarrow [\dot{u}_i(t_1)\ \dot{u}_i(t_2)\ \cdots\ \dot{u}_i(t_{n_t})]^T$
- $s_t \leftarrow s_{l_1} + s_{l_2}$ ▷ s_t is the total number of unknowns
- $s_{l_1}^h \leftarrow (s_{l_1} - 1)/2$ and $s_{l_2}^h \leftarrow (s_{l_2} - 1)/2$ ▷ Half of stencil sizes
- $\boldsymbol{A} \leftarrow \texttt{zeros}(n_t, s_t)$ ▷ Initialize $\boldsymbol{A} \in \mathbb{R}^{n_t \times s_t}$ as a zero matrix
- $A_{j,1:s_{l_1}} \leftarrow c[u_{i-s_{l_1}^h}(t_j)\ \cdots\ u_i(t_j)\ \cdots\ u_{i+s_{l_1}^h}(t_j)]$ ▷ Assign advection terms
- $A_{j,s_l+1:s_t} \leftarrow -\nu[u_{i-s_{l_2}^h}(t_j)\ \cdots\ u_i(t_j)\ \cdots\ u_{i+s_{l_2}^h}(t_j)]$ ▷ Assign diffusion terms
- $f(\boldsymbol{x}) \leftarrow c(x_{s_{l_1}^h+1} - \sum_{l=1, l\neq s_{l_1}^h+1}^{s_{l_1}} |x_l|)$ ▷ Stability constraint: advection term
- $g(\boldsymbol{x}) \leftarrow -\nu(x_{s_{l_1}+s_{l_2}^h+1} - \sum_{l=s_{l_1}, l\neq s_{l_1}+s_{l_2}^h+1}^{s_t} |x_l|)$ ▷ Stability constraint: diffusion term
- $\boldsymbol{J}_1 \leftarrow \partial f(\boldsymbol{x})/\partial \boldsymbol{x}$ and $\boldsymbol{J}_2 \leftarrow \partial g(\boldsymbol{x})/\partial \boldsymbol{x}$ ▷ Determine the Jacobians of constraints
- $\boldsymbol{x} \leftarrow \arg\min_{\boldsymbol{x}} ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||_2^2$, such that $f(\boldsymbol{x}), g(\boldsymbol{x}) \geq 0$ ▷ Solved using an optimizer
- $\boldsymbol{L}_1^{i,m} \leftarrow x_{1:s_{l_1}}$ and $\boldsymbol{L}_2^{i,m} \leftarrow x_{s_{l_1}+1:s_t}$

end for

$\boldsymbol{L}_1^m \leftarrow \boldsymbol{L}_1^{i,m}$ and $\boldsymbol{L}_2^m \leftarrow \boldsymbol{L}_2^{i,m}$ ▷ Assemble local operators to give global operators or store local operators directly as a sparse matrix

size which is discussed in detail for each test case. The standard regression problem that we use to obtain LDOs is similar to the most recent approach used in [17,24–26] for obtaining sparse differential operators. Therefore, this method serves as an adequate reference for comparing the stability performance of S-LDOs. Other methods of learning stable dynamical systems [29–31] also exist. However, as these approaches do not give sparse differential operators, which is the main goal of this article, these methods are not compared in this article and instead will be the subject of a future study. We consider three test cases with different PDEs: 1-D scalar advection-diffusion equation, 1-D Burgers equation and 2-D advection equation, to demonstrate the applicability of the proposed approach for learning stable differential operators from data. The pseudocodes for obtaining LDOs and S-LDOs for 1-D scalar advection-diffusion equation are shown in Algorithms 1 and 2 respectively. These pseudocodes can be suitably adjusted to incorporate different objective functions and constraints arising from other PDE problems.

4.1. 1-D scalar advection-diffusion equation

We consider 1-D scalar advection-diffusion PDE, which is one of the most commonly used validation cases for assessing the accuracy and stability characteristics of discretization techniques. This equation is of the form

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \tag{29}$$

where c is the advection velocity and ν is the diffusion coefficient. The semi-discrete form after the spatial discretization is

$$\frac{d\boldsymbol{u}}{dt} + c\boldsymbol{L}_1\boldsymbol{u} - \nu\boldsymbol{L}_2\boldsymbol{u} = 0. \tag{30}$$

The equation for each degree of freedom is

$$\frac{du_i}{dt} + c(\boldsymbol{L}_1^i)^T\boldsymbol{u}_{\Omega_i^1} - \nu(\boldsymbol{L}_2^i)^T\boldsymbol{u}_{\Omega_i^2} = 0, \tag{31}$$

where $\boldsymbol{u}_{\Omega_i^1}$ and $\boldsymbol{u}_{\Omega_i^2}$ are solution stencils for the advection and diffusion terms, whereas $\boldsymbol{L}_1^i$ and $\boldsymbol{L}_2^i$ are the local advection and diffusion differential operators that can be assembled to form $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ respectively. For scenarios where c and ν are unknown, these could be included in the definition of linear differential operators $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$. To generate the data, we solve this PDE using a 1st-order backward difference for the advective term and a 2nd-order centered difference for the diffusion term with $n = 201$ degrees of freedom. This spatial discretization, coupled with the temporal discretization mentioned later, gives accurate and stable results. With this spatial discretization, the differential operators for the chosen discretization are

$$\boldsymbol{L}_1^i = \frac{1}{\Delta x}[-1, 1, 0]^T \quad \text{and} \quad \boldsymbol{L}_2^i = \frac{1}{2\Delta x}[-1, 0, 1]^T. \tag{32}$$

In this case, the stencils for advection and diffusion terms are $\boldsymbol{u}_{\Omega_i^1} = \boldsymbol{u}_{\Omega_i^2} = [u_{i-1}, u_i, u_{i+1}]^T$. We use periodic boundary conditions for this test case. The solution of Eq. (31) not only provides us with data to determine LDOs and S-LDOs but also serves as reference results for assessing the performance of learned differential operators. This PDE exhibits a mixed hyperbolic-parabolic

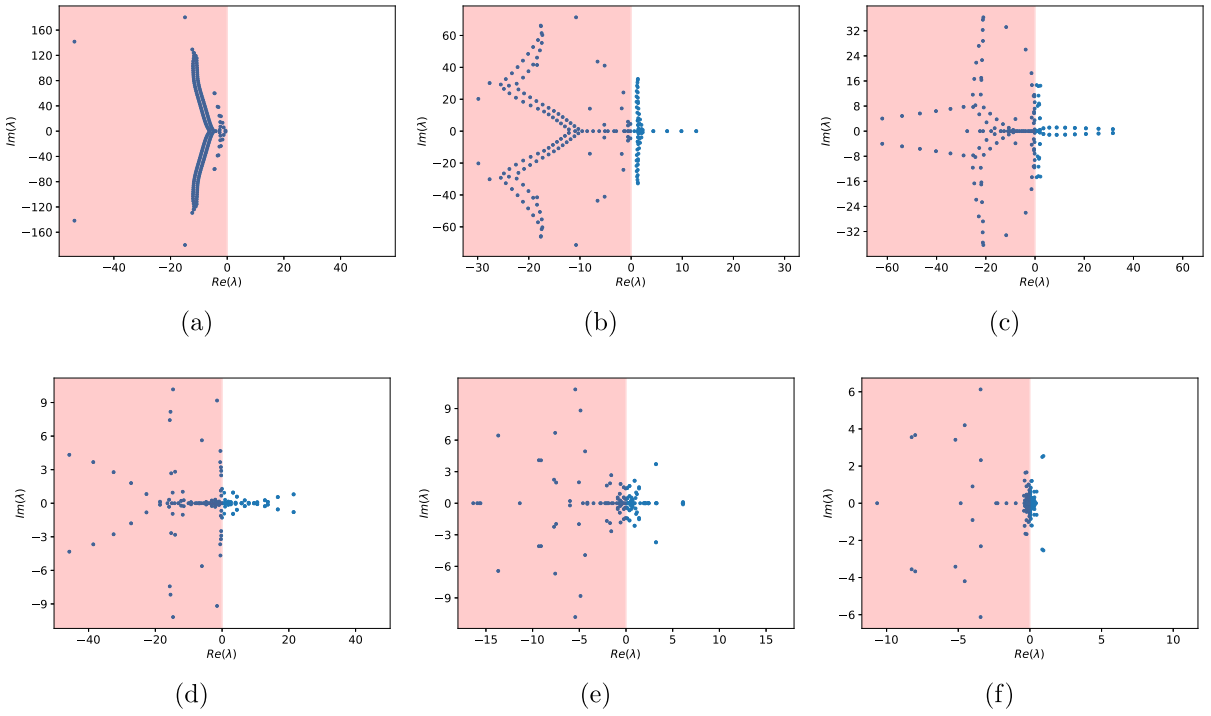

Fig. 2. Diffusion problem ($c = 0$, $\nu = 0.02$): Eigenvalues λ of LDOs ($\boldsymbol{L}_2^m$) for stencil sizes of (a) 3, (b) 5, (c) 7, (d) 11, (e) 21 and (f) 41 and regularization parameter $\beta_1 = 10^{-3}$. The shaded area indicates the stable region.

nature based on the values of c and ν. We first isolate the hyperbolic and parabolic nature to study the performance of learned differential operators for pure diffusive and advective problems. After these tests, we consider learning differential operators when both hyperbolic and parabolic terms are active.

4.1.1. *Diffusion problem:* $c = 0, \nu = 0.02$

We first consider the diffusion problem, which exhibits a parabolic nature. This problem is obtained by setting the advection velocity to zero, resulting in the following semi-discrete form of PDE for the ith degree of freedom:

$$\frac{du_i}{dt} - \nu(\boldsymbol{L}_2^i)^T \boldsymbol{u}_{\Omega_i^2} = 0. \tag{33}$$

In this article, we model the semi-discrete form as

$$\frac{du_i}{dt} - \nu(\boldsymbol{L}_2^{i,m})^T \boldsymbol{u}_{\Omega_i^2} = 0, \tag{34}$$

where the modeled linear differential operator $\boldsymbol{L}_2^{i,m} \in \mathbb{R}^{s_l}$ is learned from the generated data and $\boldsymbol{u}_{\Omega_i^2} \in \mathbb{R}^{s_l}$ is the solution stencil of dimensionality s_l. The modeled differential operator of the system $\boldsymbol{L}_2^m$ is assembled from $\boldsymbol{L}_2^{i,m}$. We obtain 500 snapshots of data by solving Eq. (33) using the forward Euler method with a timestep size of 0.04. This data is used to determine LDOs and S-LDOs by solving the regression problems mentioned earlier. We normalize these local regression problems by the Euclidean norm of local solution over time to ensure similar regression problems are solved for different locations in the domain.

We first assess the stability properties of the learned differential operator by analyzing the eigenvalues of $\boldsymbol{L}_2^m$. These eigenvalues for LDOs of different stencil sizes are shown in Fig. 2. All eigenvalues have a negative real part for the smallest stencil size ($s_l = 3$). On the other hand, for all other stencil sizes ($s_l > 3$), there are several eigenvalues with a positive real part. We also observe a reduction in the magnitude of eigenvalues with increased stencil size. This behavior indicates that the increase in stencil size improves the stability. Despite this improvement in stability, as several stencil sizes ($s_l > 3$) have positive eigenvalues, these learned operators are unstable. Even though the smallest stencil size ($s_l = 3$) does not follow this behavior and is found to be stable for this pure diffusion problem, we will show using the following test case that this behavior does not hold for the advection-dominated scenario.

The regression problem in Eq. (15) also involves a regularization term often tuned to improve the well-posedness of the regression problem and avoid overfitting of the model. We assess the impact of the regularization parameter β_1 on the stability characteristics of the LDO. The eigenvalues of LDOs $\boldsymbol{L}_2^m$ for different values of β_1 are shown in Fig. 3. The results indicate that the increase in β_1 reduces the positive real part of the eigenvalues. Therefore, an increase in β_1 improves the stability characteristics of the learned

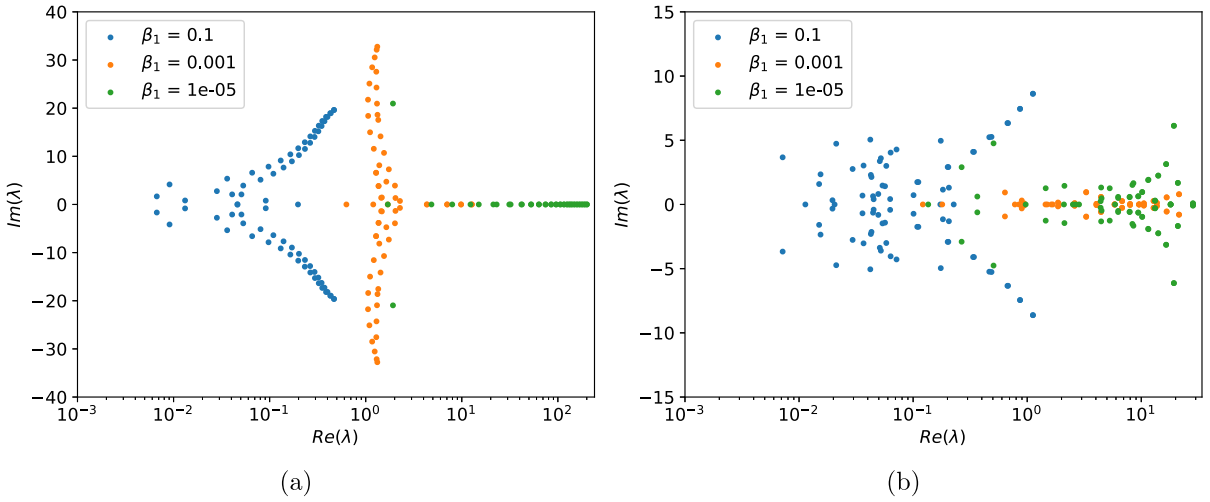

Fig. 3. Diffusion problem ($c = 0$, $\nu = 0.02$): Eigenvalues λ of LDOs ($\boldsymbol{L}_2^m$) for stencil sizes of (a) 5 and (b) 11 with several regularization parameters. The stable region is not shown as the x-axis is in the log scale.

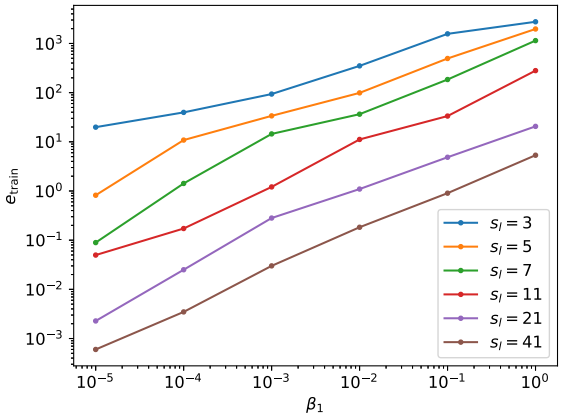

Fig. 4. Diffusion problem ($c = 0$, $\nu = 0.02$): Error in the regression objective function (e_{train} defined in Eq. (35)) for LDOs of different stencil sizes.

differential operator. However, these learned differential operators are unstable as positive real part of eigenvalues exist even at higher values of β_1. The error in the regression objective function

$$e_{\text{train}} = \sqrt{\sum_{i=0}^{n}\left(\sum_{l=0}^{n_t}\left(\frac{du_i}{dt}(t_l) - \nu(\boldsymbol{L}_2^{i,m})^T \boldsymbol{u}_{i,\Omega_i^2}(t_l)\right)^2\right)^2} \tag{35}$$

is shown in Fig. 4. The error increases with the increase in β_1 for all the stencil sizes. This observation indicates that improved stability characteristics with increased β_1 come at the cost of reduced learning accuracy. We also observe that the increase in the stencil size not only improves stability but also reduces the error while learning LDOs. Therefore, a wider stencil would imply a more accurate and improved stability of LDOs. However, such LDOs are also accompanied by a larger matrix bandwidth, which makes them more expensive due to the higher cost of matrix–vector products.

The results have indicated that LDOs for most stencil sizes are unstable even for a diffusion problem. We expect S-LDOs to resolve this issue and provide stable differential operators. We assess the stability characteristics by analyzing the eigenvalues of S-LDOs $\boldsymbol{L}_2^m$ for different stencil sizes as shown in Fig. 5. We observe that all the eigenvalues have a negative real part, implying that S-LDOs are stable for all tested stencil sizes. With the increase in stencil size, we observed a reduction in the magnitude of the eigenvalues. A larger ratio of eigenvalue magnitudes implies stiffer system behavior and possibly decreased accuracy. The results indicate that the increase in stencil size reduces the stiffness of the system. Despite the variability in stiffness behavior, the stability of the system is still guaranteed in such situations.

Analysis of eigenvalues has allowed the identification of stability characteristics of LDOs and S-LDOs. We now assess the performance of these learned operators on replicating the dynamics of true solutions. These solution dynamics are obtained by

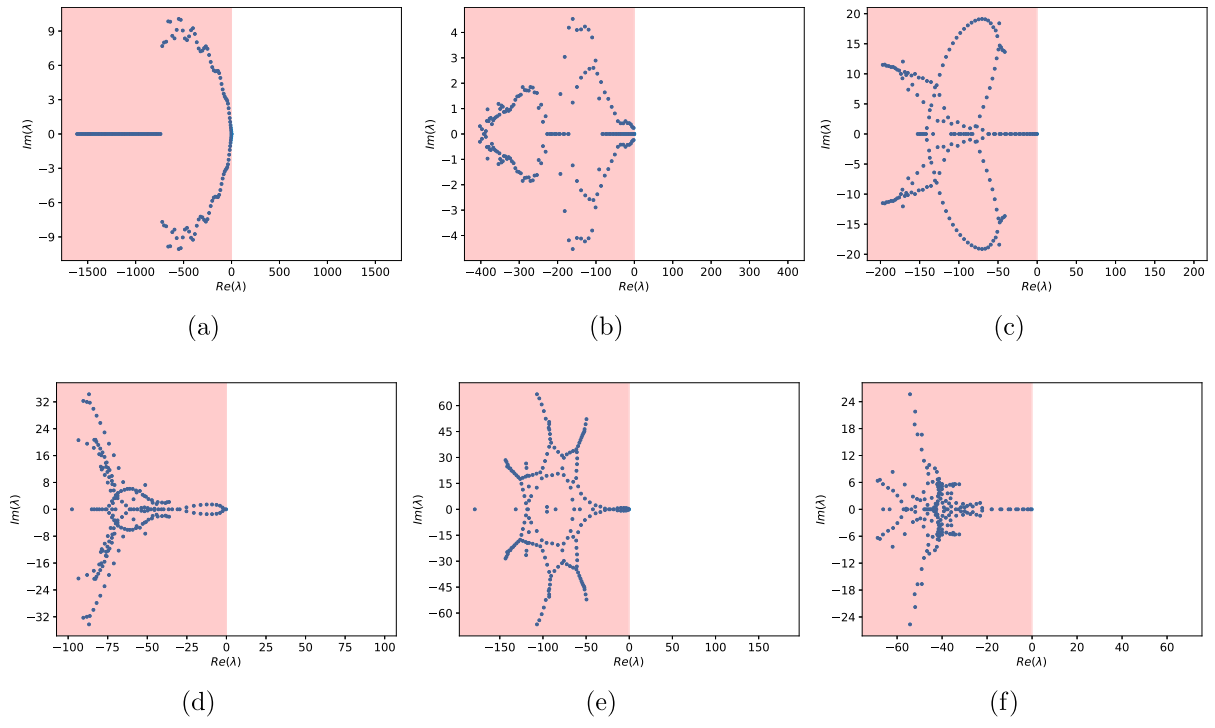

Fig. 5. Diffusion problem ($c = 0$, $\nu = 0.02$): Eigenvalues λ of S-LDOs ($\boldsymbol{L}_2^m$) of stencil size of (a) 3, (b) 5, (c) 7, (d) 11, (e) 21 and (f) 41 with a regularization parameter of $\beta_1 = 10^{-3}$. The shaded area indicates the stable region.

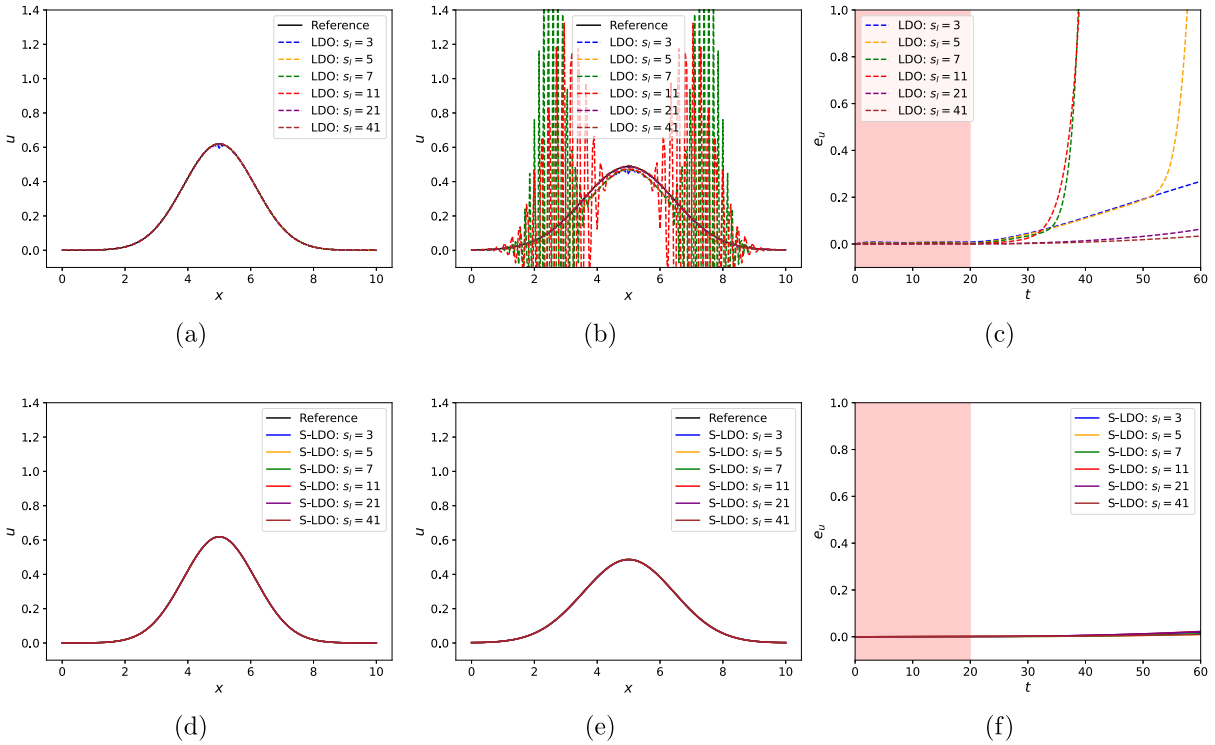

Fig. 6. Diffusion problem ($c = 0$, $\nu = 0.02$): Predicted solution u at (a) $t = 20$ s, (b) $t = 40$ s and (c) error in time (e_u defined in Eq. (36)) using LDOs (with $\beta_1 = 10^{-3}$). Predicted solution u at (d) $t = 20$ s, (e) $t = 40$ s and (f) error in time (e_u defined in Eq. (36)) using S-LDOs. The results for S-LDOs with different stencil sizes overlap. In the error plots, the unshaded region is the region of extrapolation.

solving Eq. (34) using the same time integration scheme for generating the data. We also quantify the error in solution prediction defined as

$$e_u(t) = \frac{||\boldsymbol{u}^{ref}(t) - \boldsymbol{u}(t)||_2}{||\boldsymbol{u}^{ref}(t)||_2}, \tag{36}$$

where $\boldsymbol{u}^{ref}(t)$ is the reference solution and $\boldsymbol{u}(t)$ is the solution predicted using LDOs or S-LDOs. The predicted solutions using LDOs and S-LDOs and the corresponding errors are shown in Fig. 6. Both LDOs and S-LDOs provide a good estimation of the solution at peak and tails at $t = 20$ s for all stencil sizes. The smallest stencil size ($s_l = 3$) for LDOs exhibits some oscillations at the peak for $t = 20$ s, whereas S-LDOs exhibit very close prediction to the reference results for all stencil sizes. At $t = 40$ s, we observe oscillations in solutions predicted by LDOs, which are a consequence of the unstable nature of LDOs. On the other hand, solutions predicted by S-LDOs are very accurate even at $t = 40$ s which is outside the training dataset. These observations are also reiterated by assessing the variation of error in time. Despite the low error for LDOs in the initial time interval, the data from which is used for learning differential operators, high error and unstable behavior is observed for multiple stencil sizes outside this time interval. The error is maintained at a low value for the largest stencil sizes ($s_l \geq 21$). However, this error is expected to increase in future time instances due to the unstable nature of LDOs. On the other hand, we observe that S-LDOs exhibit much smaller errors, which do not grow substantially outside the initial time interval. Furthermore, this error is low for all stencil sizes, indicating that even sparser S-LDOs are more accurate and stable than LDOs with much larger stencil sizes. These results suggest that S-LDOs are better suited for diffusion problems than LDOs.

4.1.2. *Advection problem:* $c = 1.25, \nu = 0$

For the diffusion problem, although LDOs do not yield as good performance as S-LDOs, errors do not rapidly blow up due even with the unstable system behavior for large stencil sizes. This behavior is peculiar to this test case as diffusive problems have fewer stability issues than other advection-dominated problems. Therefore, to assess the performance of LDOs and S-LDOs for hyperbolic PDEs, we consider the advection problem with the following semi-discrete form of PDE for the ith degree of freedom:

$$\frac{du_i}{dt} + c(\boldsymbol{L}_1^i)^T \boldsymbol{u}_{\Omega_i^1} = 0. \tag{37}$$

In this article, this semi-discrete form is modeled as

$$\frac{du_i}{dt} + c(\boldsymbol{L}_1^{i,m})^T \boldsymbol{u}_{\Omega_i^1} = 0, \tag{38}$$

where the modeled local differential operator $\boldsymbol{L}_1^{i,m} \in \mathbb{R}^{s_l}$ is learned from generated data and $\boldsymbol{u}_{\Omega_i^1} \in \mathbb{R}^{s_l}$ is the solution stencil of dimensionality s_l. The modeled differential operator of the system $\boldsymbol{L}_1^m$ is assembled from $\boldsymbol{L}_1^{i,m}$. The details of the time integration for data generation are the same as those for the diffusion problem.

We assess the stability of the LDOs by analyzing the eigenvalues of $-\boldsymbol{L}_1^m$ for different stencil sizes as shown in Fig. 7. These results echo those obtained for the diffusion problem, where LDOs were observed to be unstable for all stencil sizes while improvement in stability characteristics was observed for larger stencils. Similar to the diffusion problem results, LDOs exhibited an improvement in stability behavior using a higher value of regularization parameter β_1 but at the cost of larger errors in the regression objective function e_{train}. The figures showing these results are not included in the article for brevity.

The results indicated that LDOs with different stencil sizes and regularization parameters are unstable for the advection problem. We now assess the stability characteristics of S-LDOs by analyzing the eigenvalues of $-\boldsymbol{L}_1^m$ as shown in Fig. 8. The eigenvalues for different stencil sizes are very similar and overlapping. We observe that all the eigenvalues have a negative real part, implying that S-LDOs are stable. This behavior indicates that adding the constraint in the regression formulation exhibits the desired effect and enables learning a stable differential operator from data.

We now assess the performance of these learned operators on replicating the dynamics of true solutions. These solution dynamics are obtained by solving Eq. (38) using the same time integration scheme used for generating the data. The solutions obtained using LDOs and S-LDOs for different stencil sizes and corresponding errors are shown in Fig. 9. The results indicate that LDOs exhibit oscillations near the tails, especially for the coarsest stencil size, even at an early time $t = 0.4$ s. The solutions predicted by LDOs at later instances are not shown because they exhibit large oscillations for all stencil sizes. The variation of prediction errors in time solidifies this point as we observe a rapid increase in errors for all stencil sizes at an early time, which is within the dataset used for determining LDOs. This behavior is also exhibited at other regularization parameters that were not discussed for brevity. Conversely, S-LDOs exhibit very high accuracy without any instability in the solution. Even though LDOs become unstable and yield wildly inaccurate solutions at later time instances, S-LDOs provide accurate predictions even at $t = 20$ s. The error plots indicate that S-LDOs exhibit a much smaller error in time, which grows linearly in time even outside the initial time region used to determine S-LDOs. This error is maximum when the smallest stencil is used and becomes lower when larger solution stencils are selected. An exception to this behavior is $s_l = 5$, which gives the lowest errors that could be attributed to initialization and tolerance of the optimizer.

These results indicate the importance of adding stability constraints in the regression problem for learning differential operators. Without such constraints, the learned operator may not be stable and fail to perform accurately even within the training data period. These observations for stability and accuracy can also be explained by looking at learned differential operators (scaled by Δx and spatially averaged) using both approaches as shown in Table 1. S-LDOs are observed to be similar to differential operators corresponding to the 1st-order backward difference, which explains the stable behavior of these operators. With increased stencil size,

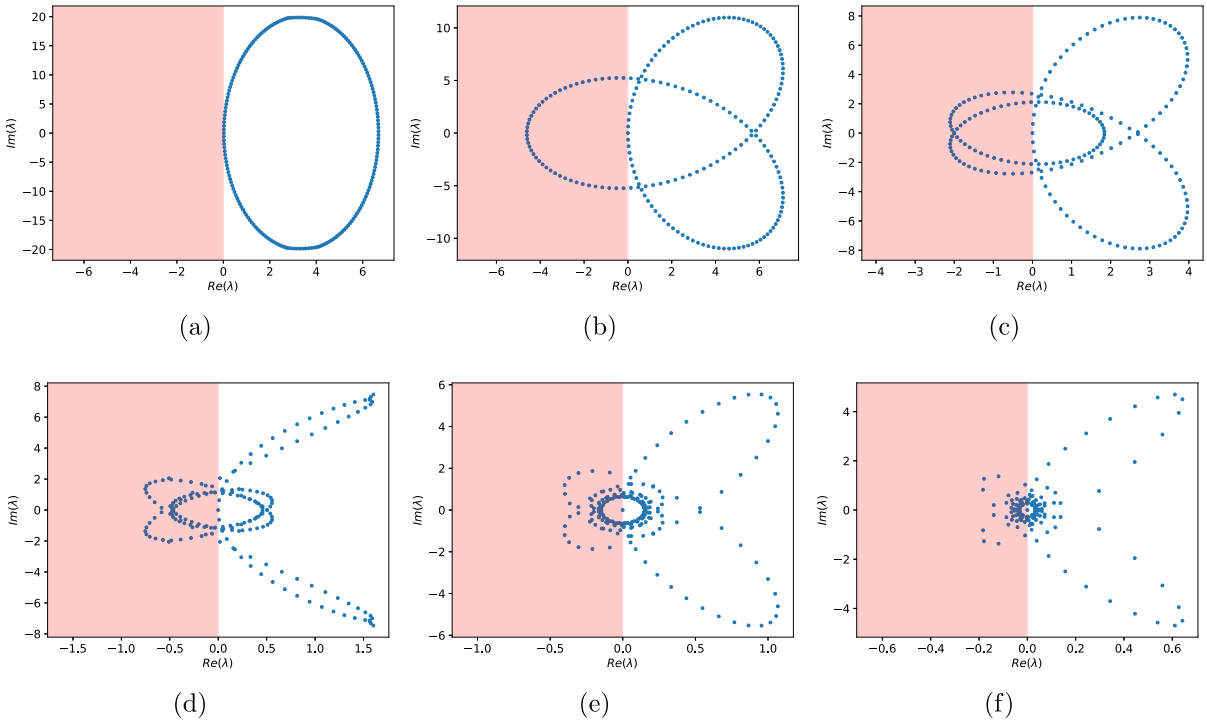

Fig. 7. Advection problem ($c = 1.25$, $\nu = 0$): Eigenvalues λ of LDOs ($-\boldsymbol{L}_1^m$) for stencil size of (a) 3, (b) 5, (c) 7, (d) 11, (e) 21 and (f) 41 with a regularization parameter of $\beta_1 = 10^{-3}$. The shaded area indicates the stable region.

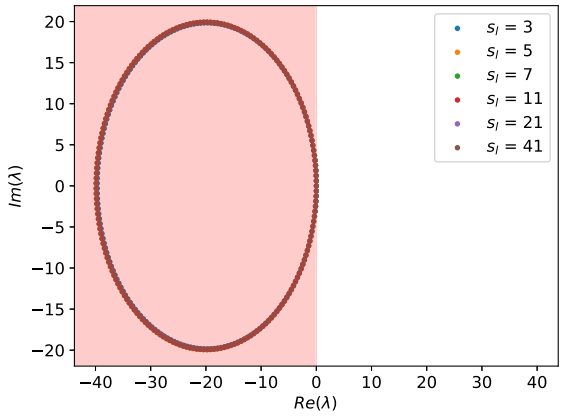

Fig. 8. Advection problem ($c = 1.25$, $\nu = 0$): Eigenvalues λ of S-LDOs ($-\boldsymbol{L}_1^m$) for various stencil sizes. The shaded area indicates the stable region. All the tested stencil sizes lead to overlapping eigenvalues.

this behavior is still retained, and the solution stencil does not correspond to a higher-order backward difference stencil. Therefore, despite providing better accuracy, a larger stencil does not guarantee a higher order of accuracy for the learned operator. Instead, it recovers the true differential operator with a better accuracy. On the other hand, LDOs do not directly resemble any standard finite difference stencil, especially for larger stencils.

4.1.3. Advection-diffusion problem: $c = 0.2$, $\nu = 0.02$

In this third scenario, we explore a mixed hyperbolic-parabolic regime with active advection and diffusion terms. Common scalar transport equations often have this form. The resulting differential equations for the ith degree of freedom is given in Eq. (31). In this article, we model the semi-discrete form as

$$\frac{du_i}{dt} + c(\boldsymbol{L}_1^{i,m})^T \boldsymbol{u}_{\Omega_i^1} - \nu(\boldsymbol{L}_2^{i,m})^T \boldsymbol{u}_{\Omega_i^2} = 0, \tag{39}$$

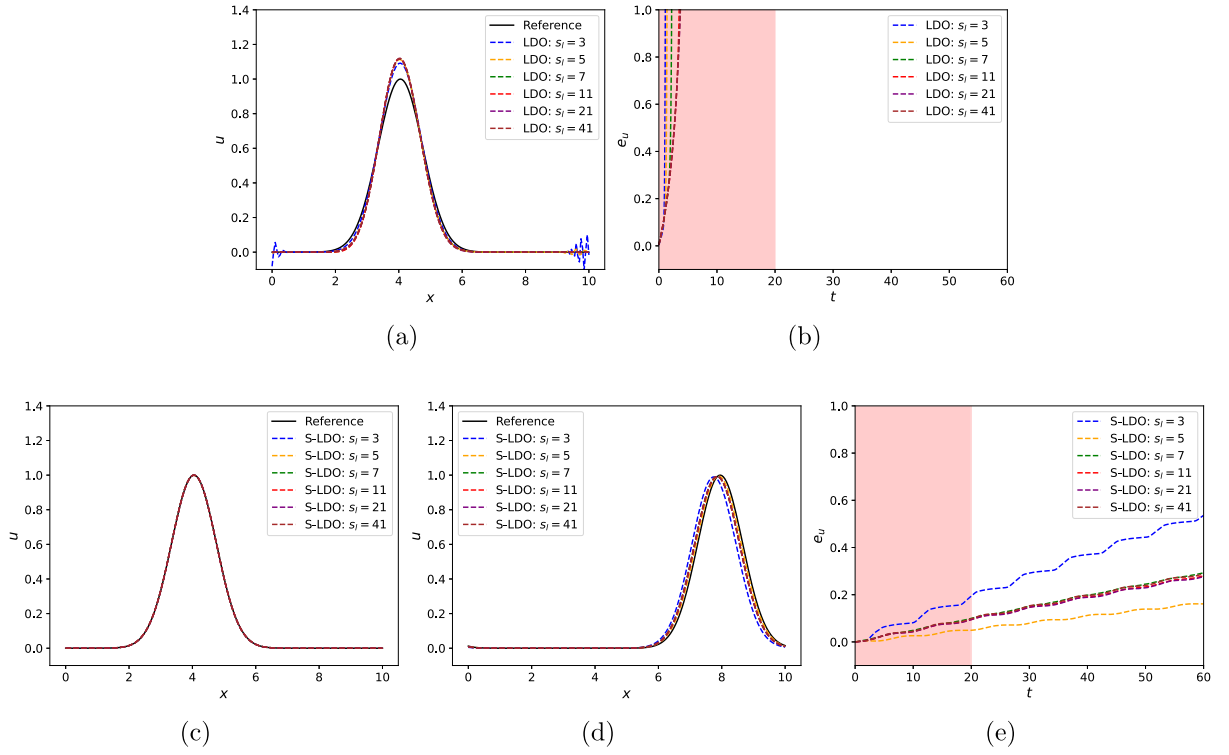

Fig. 9. Advection problem ($c = 1.25$, $\nu = 0$): Predicted solution u at (a) $t = 0.4$ s and (b) corresponding errors (e_u defined in Eq. (36)) using LDO (with $\beta_1 = 10^{-3}$). Predicted solution u at (c) $t = 0.4$ s, (d) $t = 20$ s and (e) corresponding errors (e_u defined in Eq. (36)) using S-LDOs of different stencil sizes. In the error plots, the unshaded region is the region of extrapolation.

Table 1
Advection problem ($c = 1.25$, $\nu = 0$): Spatially averaged coefficients for different solution stencils for LDOs (with $\beta_1 = 10^{-3}$) and SLDOs compared to the 1st-order backward difference (BD).

Approach	Stencil size	Coefficients						
		u_{i-3}	u_{i-2}	u_{i-1}	u_i	u_{i+1}	u_{i+2}	u_{i+3}
LDO	$s_l = 3$	–	–	−0.415	−0.167	0.582	–	–
	$s_l = 5$	–	−0.084	−0.160	−0.116	0.044	0.315	–
	$s_l = 7$	−0.041	−0.080	−0.079	−0.046	0.010	0.081	0.156
SLDO	$s_l = 3$	–	–	−0.993	0.993	≈ 0	–	–
	$s_l = 5$	–	≈ 0	−0.998	0.998	≈ 0	–	–
	$s_l = 7$	≈ 0	≈ 0	−0.996	0.996	≈ 0	≈ 0	≈ 0
BD		–	–	−1	1	–	–	–

where the modeled linear differential operators $\boldsymbol{L}_1^{i,m} \in \mathbb{R}^{s_{l_1}}$ and $\boldsymbol{L}_2^{i,m} \in \mathbb{R}^{s_{l_2}}$ are learned from generated data. The local velocity stencils are denoted by $\boldsymbol{u}_{\Omega_i^1} \in \mathbb{R}^{s_{l_1}}$ and $\boldsymbol{u}_{\Omega_i^2} \in \mathbb{R}^{s_{l_2}}$, where s_{l_1} and s_{l_2} denote the dimensionality of these stencils. The ideal stability constraint for this problem is

$$c\boldsymbol{L}_1^m - \nu\boldsymbol{L}_2^m \succeq 0. \tag{40}$$

However, for the ease of implementation and better convergence of least squares optimizer, a localized version of a more conservative constraint

$$\boldsymbol{L}_1^m \succeq 0 \quad \text{and} \quad \boldsymbol{L}_2^m \preceq 0 \tag{41}$$

is used. We obtain 1000 snapshots of data by solving Eq. (33) using the forward Euler method with a timestep size of 0.04. Assessment of stability properties by identifying eigenvalues yielded similar results and discussion to those observed for pure advection and diffusion cases. Even though we are not discussing them for brevity, the results indicated that LDOs are unstable. In contrast, S-LDOs result in a stable system as we ensure that eigenvalues for $-c\boldsymbol{L}_1^m + \nu\boldsymbol{L}_2^m$ are negative by solving a constrained regression problem. Instead, we assess the performance of these learned operators on replicating the dynamics of the true solution, which is referred to as the reference solution in the figures. The predicted dynamics are obtained by solving Eq. (39) using the same time integration scheme used for generating the data.

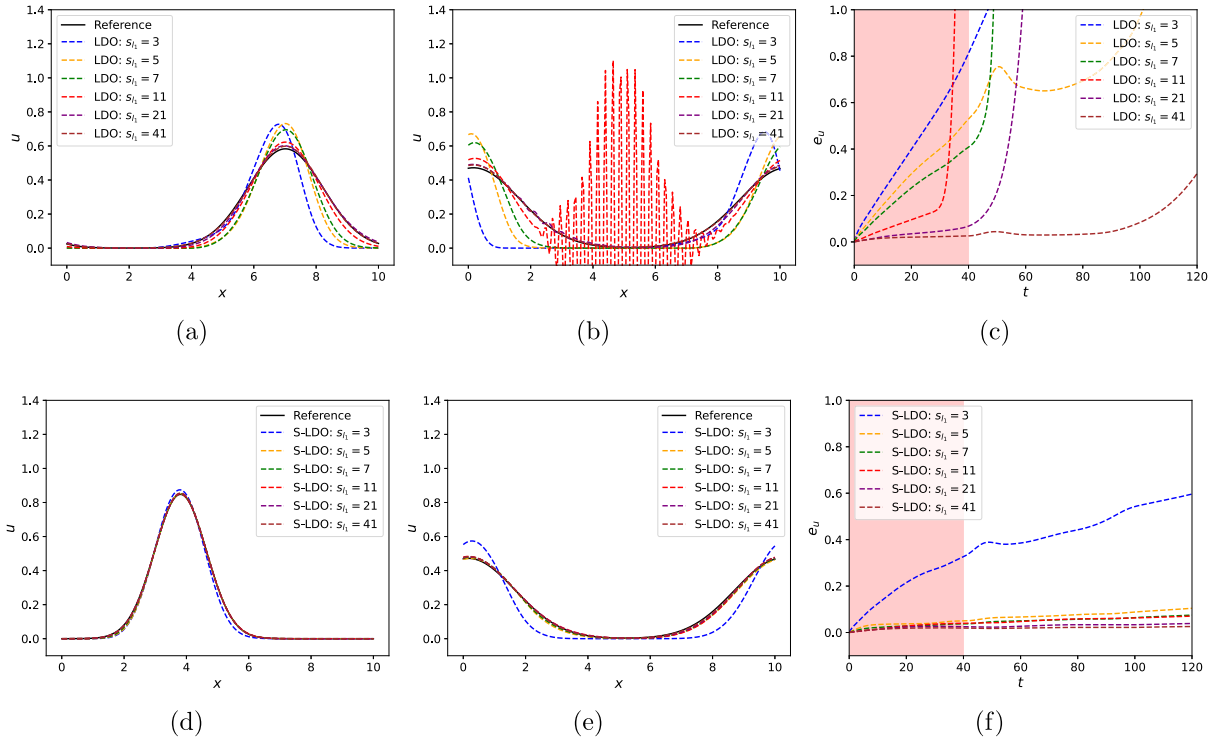

Fig. 10. Advection-diffusion problem ($c = 0.2$, $\nu = 0.02$): Predicted solutions u at (a) $t = 20$ s, (b) $t = 36$ s and (c) errors in time (e_u defined in Eq. (36)) using LDOs (with $\beta_1 = 10^{-3}$). Predicted solutions u at (d) $t = 20$ s, (e) $t = 36$ s and (f) errors in time (e_u defined in Eq. (36)) using SLDOs of several stencil sizes s_{l_1} and fixed $s_{l_2} = 3$. In the error plots, the unshaded region is the region of extrapolation.

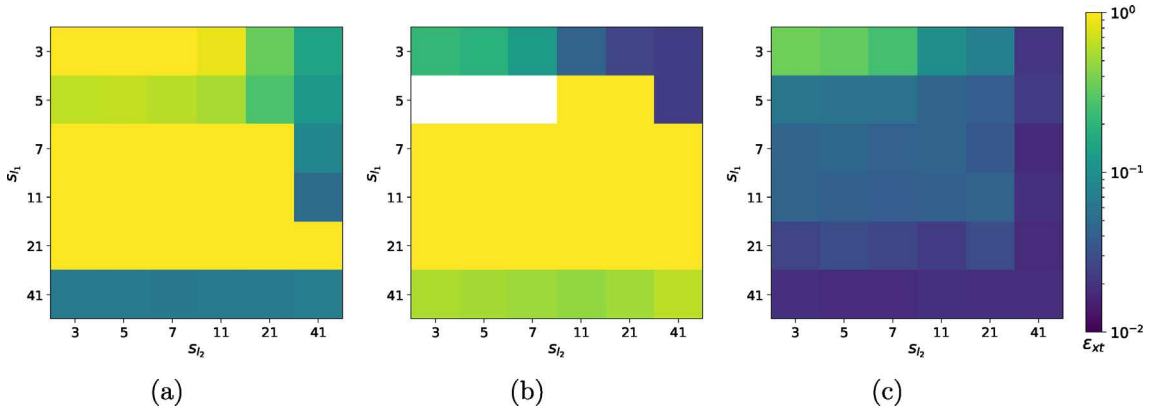

Fig. 11. Advection-diffusion problem ($c = 0.2$, $\nu = 0.02$): Total error in the solution (ε_{xt} defined in Eq. (42)) obtained using (a) LDOs (with $\beta_1 = 10^{-3}$), (b) LDOs (with $\beta_1 = 10^{-6}$) and (c) S-LDOs. The white regions indicate a very high error that is an undefined number.

The predicted solutions obtained using LDOs and S-LDOs, along with the corresponding errors, are compared to the reference data in Fig. 10 for different stencil sizes of the advection term. LDOs exhibit incorrect solution predictions for most stencil sizes at $t = 20$ s. Smaller stencil sizes lead to a higher error in prediction and exhibit wildly unstable solution time instances even at $t = 36$ s, which is within the dataset used for learning these operators. This error is reduced with increased stencil sizes and LDOs using the largest solution stencil, which exhibits good accuracy for $t = 20$ s and 36 s. This behavior is also reflected by a large growth of errors in solutions predicted by LDOs for $s_{l_1} \leq 21$. In contrast, S-LDOs exhibit a much higher accuracy where all but the smallest stencil size $s_{l_1} = 3$ yield close predictions to the reference result. This behavior holds for both time instances, indicating that S-LDOs are robust and less prone to stability issues. The temporal variation of error confirms this behavior as S-LDOs exhibit a very low error for all the stencil sizes except the smallest one. As errors remain low even for time instances not used to determine the operators, these results demonstrate the applicability of S-LDOs for long-time dynamics forecasting.

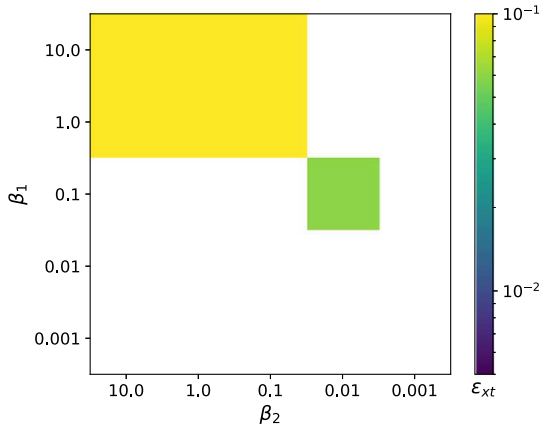

Fig. 12. 1-D Burgers problem: Total error in the solution (ϵ_{xt} defined in Eq. (42)) obtained for LDOs with $s_{l_1} = 5$ and $s_{l_2} = 5$. The white regions indicate a very high error that is an undefined number.

In the results until now, the stencil size of the diffusion operator was fixed at $s_{l_2} = 3$. To obtain a holistic picture of the operator performance for several combinations of advection and diffusion stencils, we compare the total error in both space and time

$$\epsilon_{xt} = \frac{||\boldsymbol{u}^{ref}(\cdot,\cdot) - \boldsymbol{u}(\cdot,\cdot)||_F}{||\boldsymbol{u}^{ref}(\cdot,\cdot)||_F}, \tag{42}$$

where $\boldsymbol{u}^{ref}$ is the reference solution and $\boldsymbol{u}$ is the solution predicted using LDOs or S-LDOs. This error metric is obtained by taking the L_2 norm in space and time, which corresponds to the Frobenius norm $||\cdot||_F$ if the solution is stored as a 2-D matrix. The total error for different advection and diffusion term stencil sizes is shown in Fig. 11. We observe a high error for LDOs for most combinations of stencil sizes except when $s_{l_1} = 41$ or $s_{l_2} = 41$. These results indicate that LDOs require larger stencils for low error. Even in such scenarios, the system can become unstable and yield high errors for future time instances. This behavior is also observed at different regularization parameters as shown in the figure. Alternately, S-LDOs exhibit a very low error for most combinations of stencil sizes except when $s_{l_1} = 3$ and $s_{l_2} \leq 7$. The accuracy appears to increase with both advection and diffusion stencil sizes. The accuracy of S-LDOs for small stencil sizes can be further improved by adjusting the initial guess or selecting a lower tolerance for the optimizer. In this study, we kept this strategy the same for all stencils for a fairer comparison. These results indicate that S-LDOs perform much better than LDOs and highlight the importance of ensuring stability while learning differential operators to maintain stable and accurate results for dynamics forecasting.

4.2. 1-D Burgers equation

Having demonstrated the applicability of learned differential operators for linear PDEs, we now assess the validity of this approach for nonlinear PDEs. Therefore, we consider the 1-D Burgers equation in Eq. (6). Unlike the scalar advection-diffusion problem, this equation has a nonlinear transport term. Using appropriate spatial discretization schemes, we obtain the semi-discrete form of the equations

$$\frac{d\boldsymbol{u}}{dt} + \boldsymbol{N}\boldsymbol{z}(\boldsymbol{u}) + \nu\boldsymbol{L}\boldsymbol{u} = 0, \tag{43}$$

where $\boldsymbol{N}$ is the nonlinear operator, $\boldsymbol{L}$ is the linear diffusion operator and $\boldsymbol{z}(\boldsymbol{u})$ is composed of quadratic products of $\boldsymbol{u}$ and negative sign of the viscous term is absorbed within the operator $\boldsymbol{L}$. We generate the data by using the 2nd-order centered difference for both nonlinear and linear diffusion terms with 129 degrees of freedom. The differential equation for the ith degree of freedom is

$$\frac{du_i}{dt} + (\boldsymbol{N}^i)^T \boldsymbol{z}_{\Omega_i^n} + \nu(\boldsymbol{L}^i)^T \boldsymbol{u}_{\Omega_i^l} = 0, \tag{44}$$

where

$$\boldsymbol{N}^i = \frac{1}{2\Delta x}[-1, 0, 1]^T \quad \text{and} \quad \boldsymbol{L}^i = \frac{1}{(\Delta x)^2}[-1, 2, -1]^T. \tag{45}$$

The stencil for the nonlinear term is $\boldsymbol{z}_{\Omega_i^n} = u_i[u_{i-1}, u_i, u_{i+1}]^T$, whereas it is $\boldsymbol{u}_{\Omega_i^l} = [u_{i-1}, u_i, u_{i+1}]^T$ for the linear diffusion term. In this article, we model the semi-discrete form as

$$\frac{du_i}{dt} + (\boldsymbol{N}^{i,m})^T \boldsymbol{z}_{\Omega_i^n} + \nu(\boldsymbol{L}^{i,m})^T \boldsymbol{u}_{\Omega_i^l} = 0, \tag{46}$$

where $\boldsymbol{N}^{i,m} \in \mathbb{R}^{s_{l_1}}$ and $\boldsymbol{L}^{i,m} \in \mathbb{R}^{s_{l_2}}$ are nonlinear and linear operators determined from the data, while $\boldsymbol{z}_{\Omega_i^n} \in \mathbb{R}^{s_{l_1}}$ and $\boldsymbol{u}_{\Omega_i^l} \in \mathbb{R}^{s_{l_2}}$ are the solution stencils of dimensionality s_{l_1} and s_{l_2}. For nonlinear problems such as the 1-D Burgers equation, a single stable

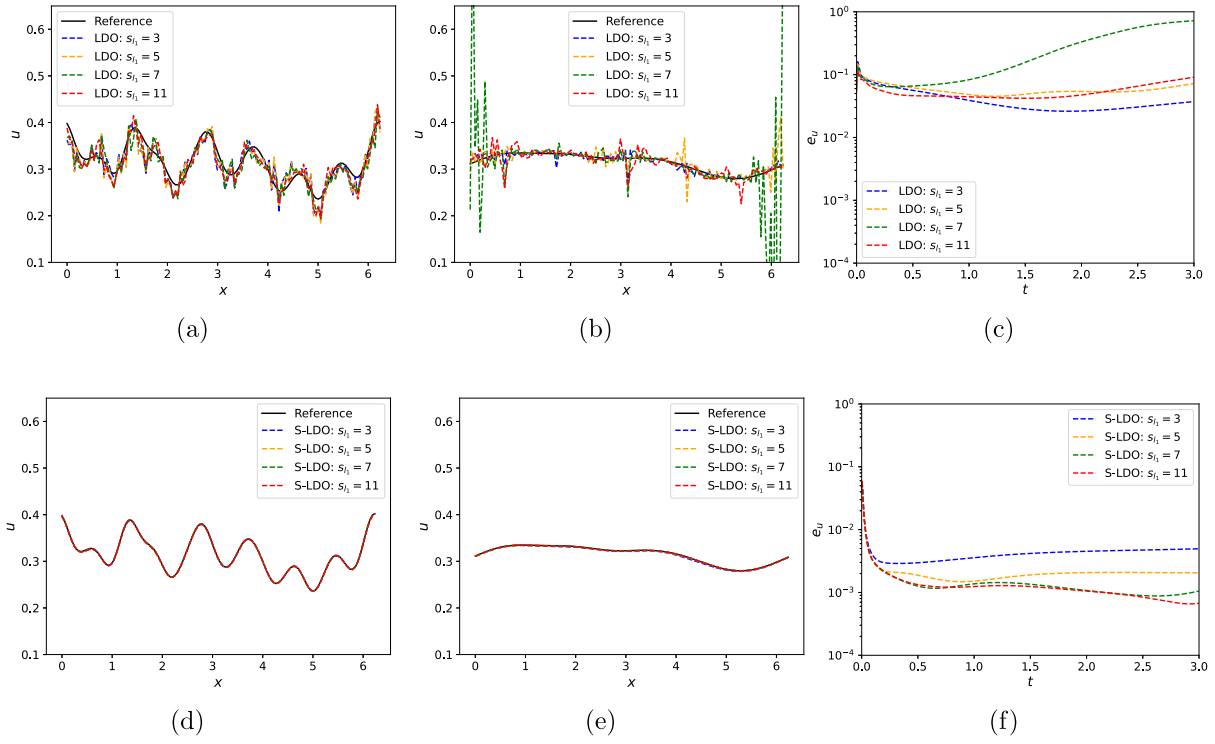

Fig. 13. 1-D Burgers problem: Predicted solutions u at (a) $t = 0.2$ s, (b) $t = 2$ s and (c) corresponding errors (e_u defined in Eq. (36)) for LDOs (with $\beta_1 = 0.1$ and $\beta_2 = 0.01$). Predicted solutions u at (d) $t = 0.2$ s, (e) $t = 2$ s and (f) corresponding errors (e_u defined in Eq. (36)) for S-LDOs. These are evaluated for different s_{l_1} while keeping $s_{l_2} = 5$.

equilibrium point does not exist as any constant solution is a valid equilibrium point. Due to the viscous term, the solution decays to a stable equilibrium close to the initial condition. As the modeled semi-discrete form in Eq. (46) is nonlinear, we cannot guarantee the global stability of these equations. Instead, we focus on local linear stability around the equilibrium point $\boldsymbol{u}^0$ by considering the 1st-order Taylor series approximation to the equation and obtaining the linearized equation

$$\frac{du_i}{dt} + (\boldsymbol{N}^{i,m,L})^T \boldsymbol{u}_{\Omega_i^n} = 0, \tag{47}$$

with

$$\boldsymbol{N}^{i,m,L} = \left| \frac{\partial (\boldsymbol{N}^{i,m})^T \boldsymbol{z}_{\Omega_i^n}}{\partial \boldsymbol{u}_{\Omega_i^n}} + \nu \frac{\partial (\boldsymbol{L}^{i,m})^T \boldsymbol{u}_{\Omega_i^l}}{\partial \boldsymbol{u}_{\Omega_i^n}} \right|_{\boldsymbol{u}^0}, \tag{48}$$

where $|\cdot|_{\boldsymbol{u}^0}$ implies the term is evaluated at equilibrium point $\boldsymbol{u}^0$ and $\boldsymbol{N}^{i,m,L}$ is the local linearized operator that is assembled to give the global linearized operator $\boldsymbol{N}^{m,L}$. This linearized equation is used in Eq. (28) to determine linear stability constraints for the regression problem in Eq. (27) for obtaining S-LDOs.

The reference data is used to learn differential operators and assess the performance of LDOs and S-LDOs. This data is obtained by integrating Eq. (44) in time using a 1st-order explicit time integration scheme and a time step size of 0.002. This test case is initialized with Gaussian random perturbations with a mean of 0.3 and a standard deviation of 0.2. For this test case, we can assess the eigenvalues of the operators $\boldsymbol{N}^{m,L}$, which can provide us with the notion of linear stability of the system around an equilibrium point. This analysis aligns with one for the advection-diffusion equation, which showed that S-LDOs are linearly stable, whereas LDOs are not. A detailed discussion on this aspect is excluded from this article for brevity. Instead, we discuss the consequences of this behavior on the evolution of the solution over time.

We first assess the impact of regularization parameters β_1 and β_2 on the predicted solution for LDOs with a fixed stencil size of $s_{l_1} = s_{l_2} = 5$. The total error in the predicted solution for LDOs is shown in Fig. 12. The results indicate the sensitivity of LDOs to selected regularization parameters and typically give predictions with very high errors. As $\beta_1 = 0.1$ and $\beta_2 = 0.01$ lead to reasonable errors where solutions are well defined, we select these regularization parameters for LDOs for further comparison.

The predicted solutions obtained using LDOs and S-LDOs and corresponding errors are compared to the reference data in Fig. 13. We observe that even at an early time $t = 0.2$ s, LDOs for multiple stencil sizes exhibit oscillations. These oscillations persist and grow larger as evident from the results at $t = 2$ s. These results correspond to the best selection of the stencil sizes for a given regularization parameter, and still, LDOs do not provide stable and accurate results. This behavior is also highlighted in the error

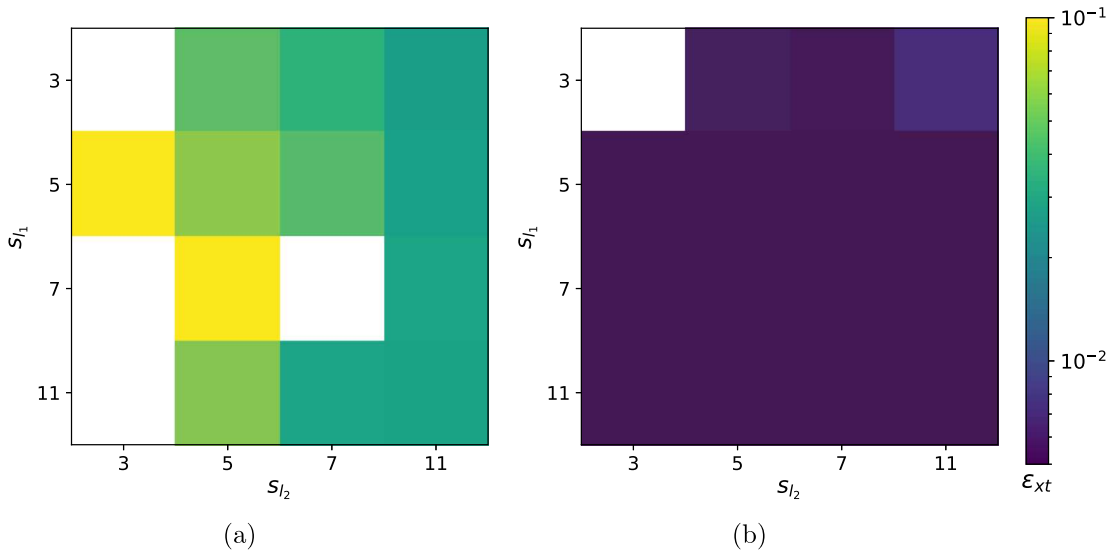

Fig. 14. 1-D Burgers problem: Total error in the solution (ε_{xt} defined in Eq. (42)) obtained using (a) LDOs (with $\beta_1 = 0.1$ and $\beta_2 = 0.01$) and (b) S-LDOs. The white regions indicate a very high error that is an undefined number.

plots, which indicate high errors for LDOs for different stencil sizes. On the contrary, S-LDOs for all stencil sizes exhibit high accuracy at both time instances as the results are close to the reference data. Furthermore, the errors in solution predicted by S-LDOs are consistently low for all stencil sizes, even at longer times. These results indicate that S-LDOs exhibit superior stability properties and provide feasible, stable and physically accurate solutions.

The matrix plot with total errors for different combinations of stencil sizes for LDOs and S-LDOs is shown in Fig. 14. We observe that LDOs exhibit high errors for all combinations of stencil sizes. The total error is lower for the largest stencil sizes, similar to the behavior observed for the advection-diffusion problem. However, this error is still much larger compared to errors exhibited by S-LDOs. For the smallest stencil size for S-LDOs, some instability in the results is observed, which gives high errors. As S-LDOs can only ensure linear stability, whereas the system is nonlinear, there is no guarantee that all stencil sizes will always have stable solutions. Unstable behavior is also observed at several stencil sizes larger than those considered in this study. The inability of differential operators to perform well in such scenarios is not a big issue as larger stencil sizes may not even be considered due to higher computational costs. This behavior highly depends on the linearization strategy and selection of the equilibrium point. The errors for most combinations of stencil sizes are very low, demonstrating the applicability of S-LDOs in giving accurate solutions. Although S-LDOs are designed to provide linearly stable operators, they perform remarkably well for nonlinear problems, as demonstrated using this test case. These results highlight that adding linear stability constraints while learning nonlinear operators can be a viable approach even for nonlinear PDEs.

4.3. 2-D scalar advection equation

We select the 2-D scalar advection equation for the third test case to demonstrate the applicability of learned differential operators to operators for 2-D hyperbolic PDEs. The equation is given as follows:

$$\frac{\partial u}{\partial t} + \boldsymbol{c} \cdot \nabla u = 0, \tag{49}$$

where $\boldsymbol{c} = [c_x \; c_y]^T$ is the advection velocity. Using an appropriate spatial discretization scheme, the PDE can be converted to a set of ODEs

$$\frac{d\boldsymbol{u}}{dt} + c_x \boldsymbol{L}^x \boldsymbol{u} + c_y \boldsymbol{L}^y \boldsymbol{u} = 0, \tag{50}$$

where $\boldsymbol{L}^x$ and $\boldsymbol{L}^y$ are the linear operators on the solution components. For data generation, we use a 1st-order backward difference for approximating these operators with 101 degrees of freedom in each direction. This selection implies that the differential equations for the ith degree of freedom is

$$\frac{du_i}{dt} + c_x (\boldsymbol{L}^{i,x})^T \boldsymbol{u}_{\Omega_i^x} + c_y (\boldsymbol{L}^{i,y})^T \boldsymbol{u}_{\Omega_i^y} = 0, \tag{51}$$

where

$$\boldsymbol{L}^{i,x} = \frac{1}{\Delta x}[-1, 1, 0]^T \quad \text{and} \quad \boldsymbol{L}^{i,y} = \frac{1}{\Delta y}[-1, 1, 0]^T. \tag{52}$$

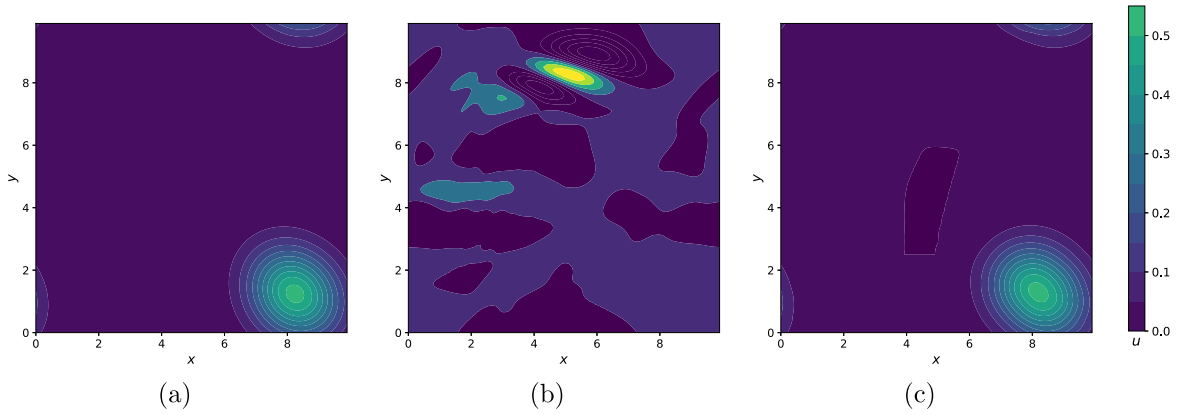

Fig. 15. 2-D advection problem: Solutions u at $t = 12.5$ s for (a) reference data, (b) the LDO with $\beta = 10^{-2}$ and (c) the S-LDO with stencil sizes $s_{l_x} = 7$ and $s_{l_y} = 7$.

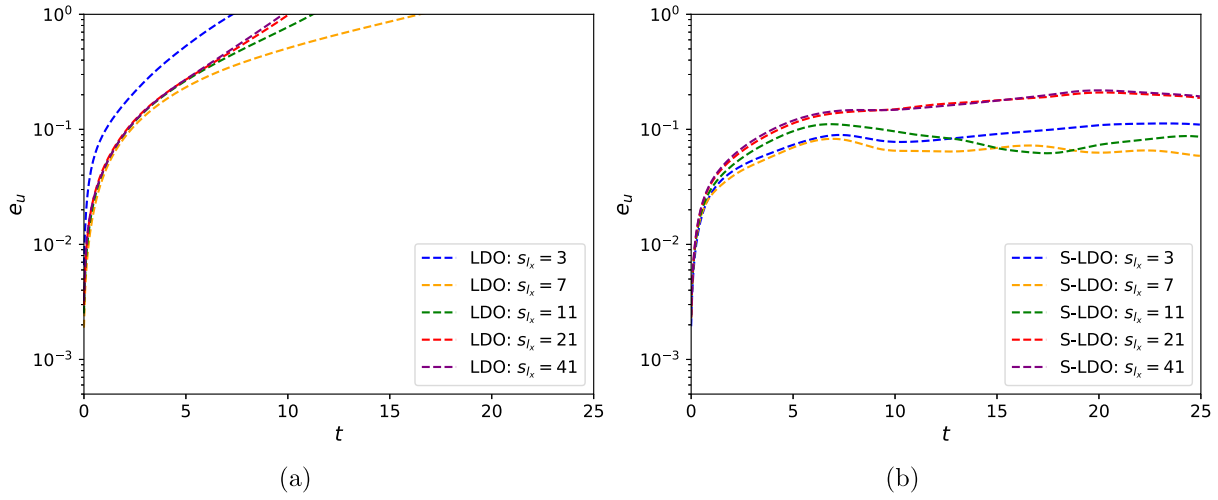

Fig. 16. 2-D advection problem: Temporal variation of errors (e_u defined in Eq. (36)) in the predicted solution obtained using (a) LDOs (with $\beta_1 = 10^{-2}$) and (b) S-LDOs of several stencil sizes s_{l_x} while keeping $s_{l_y} = 7$.

The solution stencil for the $\boldsymbol{L}^{i,x}$ operator is denoted by $\boldsymbol{u}_{\Omega_i^x}$, whereas the stencil for the $\boldsymbol{L}^{i,y}$ operator is denoted by $\boldsymbol{u}_{\Omega_i^y}$. This test case is initialized with the initial condition

$$u(x, y, 0) = \exp\left(-(x-2)^2 + (y-5)^2\right) \tag{53}$$

and integrated in time using a 1st-order forward Euler method with a timestep of 0.05. The generated data is used to learn the differential operators and as a reference result for assessing the performance of LDOs and S-LDOs. The advection velocity is chosen to be $c_x = 0.5$ and $c_y = 0.5$ where x and y components of the velocity are aligned to Cartesian x and y directions respectively. In this article, we determine the unknown operators $\boldsymbol{L}^{i,x,m}$ and $\boldsymbol{L}^{i,y,m}$ from data. Therefore, Eq. (51) is modeled as

$$\frac{du_i}{dt} + c_x(\boldsymbol{L}^{i,x,m})^T \boldsymbol{u}_{\Omega_i^x} + c_y(\boldsymbol{L}^{i,y,m})^T \boldsymbol{u}_{\Omega_i^y} = 0, \tag{54}$$

where $\boldsymbol{L}^{i,x,m} \in \mathbb{R}^{s_{l_x}}$ and $\boldsymbol{L}^{i,y,m} \in \mathbb{R}^{s_{l_y}}$ are the modeled linear operators. The stencils for these operators $\boldsymbol{u}_{\Omega_i^x} \in \mathbb{R}^{s_{l_x}}$ and $\boldsymbol{u}_{\Omega_i^y} \in \mathbb{R}^{s_{l_y}}$ have a dimensionality of s_{l_x} and s_{l_y} respectively. The constraints in S-LDO formulation ensure a positive real part for the eigenvalues of $c_x\boldsymbol{L}_x^m + c_y\boldsymbol{L}_y^m$. A detailed analysis showing this behavior is excluded for brevity, although the behavior is similar to the one observed for the 1-D advection equation.

The solution predicted using an LDO and an S-LDO is compared to the reference data in Fig. 15. We observe that prediction by the LDO is very different than the reference data as the unstable behavior of these operators leads to large deviations in the results. The predicted solution by the S-LDO exhibits a similar behavior and magnitude as the reference data. This behavior echoes other test cases, where S-LDOs consistently gave much more accurate results than LDOs. The temporal variation of errors in the solutions predicted by LDOs and S-LDOs for different stencil sizes s_{l_x} is shown in Fig. 16. We observe that LDOs exhibit a large

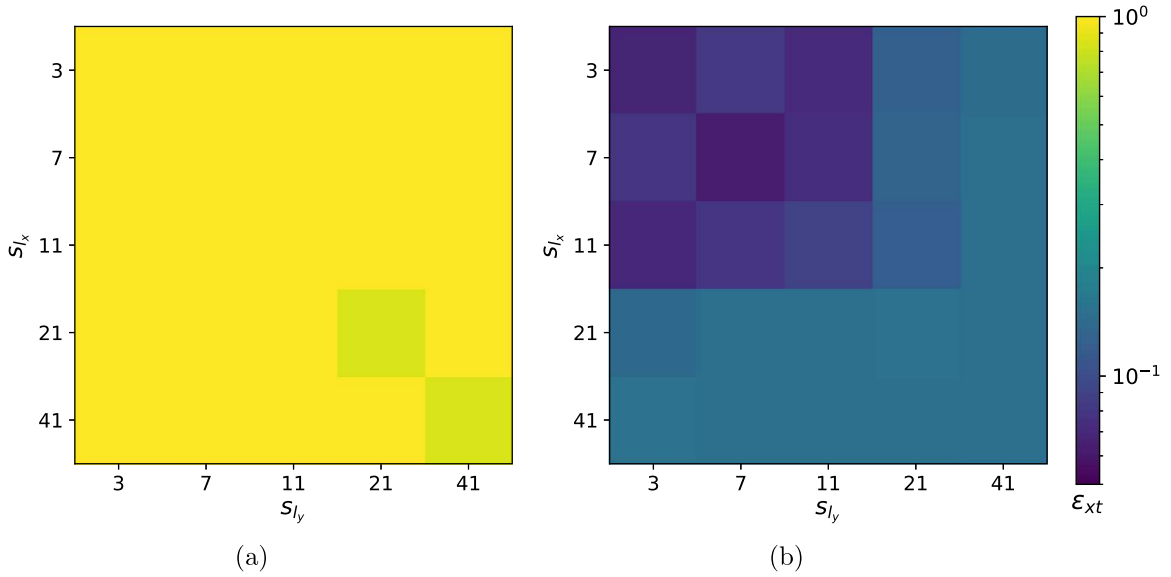

Fig. 17. 2-D advection problem: Total error in the solution (ϵ_{xt} defined in Eq. (42)) obtained using (a) LDOs (with $\beta_1 = 10^{-2}$) and (b) S-LDOs of different stencil sizes s_{l_x} and s_{l_y}.

error and grow quickly in time for different stencil sizes. These results confirm that LDOs are ill-equipped to infer operators for advection-dominated problems, as already assessed from the 1-D advection case. Conversely, S-LDOs exhibit a low error for all the stencil sizes considered in this study. After an initial rise, the error reaches a near-constant value, which does not change drastically as time increases. This behavior also numerically verifies the stability properties of S-LDOs and demonstrates its applicability for stable prediction of 2-D solution fields. Although these results are shown for different values of stencil sizes s_{l_x} while keeping $s_{l_y} = 7$, this behavior also holds other values of s_{l_y}.

The total error in the predicted solution for LDOs and S-LDOs of different stencil sizes is shown in Fig. 17. We observe that the total error is very high for LDOs for all combinations of stencil sizes. This behavior renders LDOs impractical for their use in dynamics predictions. On the contrary, S-LDOs exhibit a comparatively low error for all the stencil sizes. The error appears to be lower for the smaller stencil sizes and becomes higher for larger stencils. This behavior is different from those observed for the 1-D advection case, where the error appeared to decrease with a larger stencil. Nevertheless, the low prediction error and theoretical linear stability make S-LDOs suitable for dynamics forecasting.

5. Conclusions

Several applications, such as system identification and nonintrusive reduced order modeling, motivate the need to identify discrete approximations of PDEs from data. Most studies have focused on modern machine learning techniques to obtain accurate approximations of a PDE. However, these approaches often yield noninterpretable and dense representations, which may not be scalable for large-scale applications. Even fewer studies have addressed stability concerns for learned semi-discrete differential equations.

In this article, we propose a novel methodology for determining sparse semi-discrete approximations of PDEs from data while ensuring the linear stability of learned approximations. This approach is inspired by common spatial discretizations that have a sparse structure for computational efficiency and are stable for accurate dynamics prediction. We demonstrate that the standard regression approach, which is common in the literature, does not yield theoretically stable differential operators even for simple 1-D PDEs. We overcome this drawback by identifying stability conditions on local differential operators using dynamical system theory. These conditions are then added to the regression problem as constraints on the learned differential operators. Solving these constrained regression problems yields theoretically stable differential operators for linear PDEs. We also extend this approach to nonlinear PDEs by formulating constraints using linearized differential equations. The applicability of the proposed approach is demonstrated using three examples: 1-D scalar advection-diffusion equation, 1-D Burgers equation and 2-D advection equation. The numerical experiments indicated that differential operators learned using the standard regression approach yielded unstable differential operators for different combinations of stencil sizes and regularization parameters. Consequently, the learned differential operator often produces highly oscillatory results in the initial time window and blows up in finite time, even for linear problems. In contrast, the learned differential operators obtained using the proposed constrained regression approach yielded stable and highly accurate results for linear and nonlinear PDEs considered in this study.

The proposed approach to incorporate constraints for ensuring stability while learning differential operators can be extended to several practical applications. These learned differential operators are ideally suited for developing nonintrusive reduced

order models. We plan to demonstrate the proposed approach for this application while comparing it with other nonintrusive approaches [19,44–46]. A key benefit of the proposed approach is the possibility of enabling nonintrusive reduced order modeling for standard projection-based techniques such as those that rely on stabilization [47] and closure modeling [48,49]. We also plan to extend this approach to determine stable coarse-grained discretizations from data, enabling high-fidelity simulations at a lower computational cost. As the proposed approach relies on a local stencil, the method is directly applicable to unstructured grids, provided that a relevant grid data structure is available. Similarly, extension to other boundary conditions would involve selectively identifying constraints for boundary degrees of freedom. We plan to demonstrate this applicability in future works while considering applications to nonintrusive reduced order modeling and coarse-grained discretization determination. The strategy of ensuring stability based on linearized nonlinear equations, which is used in this article, implies that the energy of the linearized equations decays in time, which may not be a suitable assumption for several complex problems. Other strategies [29–31] for ensuring the stability of nonlinear systems have focused on adding constraints to ensure global conservation or dissipation of energy for the nonlinear system. However, these strategies are directly applicable to situations where a global operator is determined and not when we need to determine local differential operators. A suitable avenue for future work includes formulating local constraints that can ensure global energy conservation or dissipation for more robust identification of sparse differential operators for nonlinear equations.

CRediT authorship contribution statement

Aviral Prakash: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Yongjie Jessica Zhang:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

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

Data availability

Software programs for all numerical experiments considered in this article are available on the GitHub repository: https://github.com/CMU-CBML/Py_SLDO.

Acknowledgments

The authors would like to acknowledge the support from the National Science Foundation (NSF) grant, CMMI-1953323, for the funds used towards this project. The research in this paper was also sponsored by the Army Research Laboratory, United States and was accomplished under Cooperative Agreement Number W911NF-20-2-0175. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] J.T. Oden, K.R. Diller, C. Bajaj, J.C. Browne, J. Hazle, et al., Development of a computational paradigm for laser treatment of cancer, in: Computational Science – ICCS 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 530–537.
[2] J.T. Oden, K.R. Diller, C. Bajaj, J.C. Browne, J. Hazle, et al., Dynamic data-driven finite element models for laser treatment of cancer, J. Numer. Methods Partial Differ. Equ. 23 (4) (2007) 904–922.
[3] C. Bajaj, J.T. Oden, K.R. Diller, J.C. Browne, J. Hazle, et al., Using cyber-infrastructure for dynamic data driven laser treatment of cancer, in: Computational Science – ICCS 2007, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 972–979.
[4] S.L. Brunton, J.N. Kutz, Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control, Cambridge University Press, 2019.
[5] A. Prakash, K.E. Jansen, J.A. Evans, Invariant data-driven subgrid stress modeling in the strain-rate eigenframe for large eddy simulation, Comput. Methods Appl. Mech. Engrg. 399 (2022) 115457.
[6] A. Prakash, K.E. Jansen, J.A. Evans, Invariant data-driven subgrid stress modeling on anisotropic grids for large eddy simulation, Comput. Methods Appl. Mech. Engrg. 422 (2024) 116807.
[7] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. USA 113 (15) (2016) 3932–3937.
[8] H. Schaeffer, Learning partial differential equations via data discovery and sparse optimization, Proc. R. Soc. A: Math., Phys. Eng. Sci. 473 (2197) (2017) 20160446.
[9] M. Raissi, G.E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, J. Comput. Phys. 357 (2018) 125–141.
[10] S. Zhang, G. Lin, Robust data-driven discovery of governing physical laws with error bars, Proc. R. Soc. A: Math., Phys. Eng. Sci. 474 (2217) (2018) 20180305.
[11] H. Schaeffer, G. Tran, R. Ward, Extracting sparse high-dimensional dynamics from limited data, SIAM J. Appl. Math. 78 (6) (2018) 3279–3295.
[12] S.H. Rudy, S.L. Brunton, J.L. Proctor, J.N. Kutz, Data-driven discovery of partial differential equations, Sci. Adv. 3 (4) (2017) e1602614.
[13] S. Maddu, B.L. Cheeseman, I.F. Sbalzarini, C.L. Müller, Stability selection enables robust learning of differential equations from limited noisy data, Proc. R. Soc. A: Math., Phys. Eng. Sci, 478 (2262) (2022).

[14] R.J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations, Society for Industrial and Applied Mathematics, 2007.
[15] R.J. LeVeque, Finite volume methods for hyperbolic problems, in: Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.
[16] T.J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover Publications, Inc., 2000.
[17] L. Gkimisis, T. Richter, P. Benner, Adjacency-based, non-intrusive model reduction for vortex-induced vibrations, Comput. & Fluids 275 (2024) 106248.
[18] B. Boots, G.J. Gordon, S. Siddiqi, A constraint generation approach to learning stable linear dynamical systems, in: Advances in Neural Information Processing Systems, vol. 20, Curran Associates, Inc., 2007.
[19] B. Peherstorfer, K.E. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, Comput. Methods Appl. Mech. Engrg. 306 (2016) 196–215.
[20] Y. Bar-Sinai, S. Hoyer, J. Hickey, M.P. Brenner, Learning data-driven discretizations for partial differential equations, Proc. Natl. Acad. Sci. 116 (31) (2019) 15344–15349.
[21] Z. Long, Y. Lu, X. Ma, B. Dong, PDE-Net: Learning PDEs from data, in: Proceedings of the 35th International Conference on Machine Learning, vol. 80, PMLR, 2018, pp. 3208–3216.
[22] S. Maddu, D. Sturm, B.L. Cheeseman, C.L. Müller, I.F. Sbalzarini, STENCIL-NET for equation-free forecasting from data, Sci. Rep. 13 (1) (2023) 1–15.
[23] Z. Long, Y. Lu, B. Dong, PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network, J. Comput. Phys. 399 (2019) 108925.
[24] Y. Schumann, P. Neumann, Towards data-driven inference of stencils for discrete differential operators, in: Proceedings of the Platform for Advanced Scientific Computing Conference, New York, NY, USA, 2022.
[25] Y. Schumann, P. Neumann, On linear models for discrete operator inference in time dependent problems, J. Comput. Appl. Math. 425 (2023) 115022.
[26] L. Gkimisis, T. Richter, P. Benner, Adjacency-based, non-intrusive reduced-order modeling for fluid-structure interactions, Proc. Appl. Math. Mech. 23 (4) (2023).
[27] N. Meinshausen, P. Bühlmann, Stability selection, J. R. Stat. Soc. Ser. B Stat. Methodol. 72 (4) (2010) 417–473.
[28] P. Goyal, I.P. Duff, P. Benner, Inference of continuous linear systems from data with guaranteed stability, 2023, arXiv:2301.10060.
[29] P. Goyal, I.P. Duff, P. Benner, Guaranteed Stable Quadratic Models and their Applications in SINDy and Operator Inference, 2023, arXiv:2308.13819.
[30] N. Sawant, B. Kramer, B. Peherstorfer, Physics-informed regularization and structure preservation for learning stable reduced models from data with operator inference, Comput. Methods Appl. Mech. Engrg. 404 (2023) 115836.
[31] T. Koike, E. Qian, Energy-preserving reduced operator inference for efficient design and control, in: AIAA SCITECH 2024 Forum, 2024.
[32] J.G. Charney, R. Fjortoft, J. V. Neumann, Numerical integration of the barotropic vorticity equation, Tellus 2 (4) (1950) 237–254.
[33] E. Isaacson, H.B. Keller, Analysis of Numerical Methods, Dover Publications, Inc., 1994.
[34] W.J. Rugh, Linear System Theory, second ed., Prentice Hall, 1996.
[35] A.N. Tikhonov, V.Y. Arsenin, Solutions of Ill-Posed Problems, Winston & Sons, 1977.
[36] R. Tibshirani, Regression shrinkage and selection via the Lasso, J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1) (1996) 267–288.
[37] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. Ser. B Stat. Methodol. 67 (2) (2005) 301–320.
[38] C.H. S. A. McQuarrie, K.E. Willcox, Data-driven reduced-order models via regularised operator inference for a single-injector combustion process, J. R. Soc. New Zealand 51 (2) (2021) 194–211.
[39] M. Guo, S.A. McQuarrie, K.E. Willcox, Bayesian operator inference for data-driven reduced-order modeling, Comput. Methods Appl. Mech. Engrg. 402 (2022) 115336.
[40] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[41] I. Kalashnikova, B.V. Waanders, S. Arunajatesan, M. Barone, Stabilization of projection-based reduced order models for linear time-invariant systems via optimization-based eigenvalue reassignment, Comput. Methods Appl. Mech. Engrg. 272 (2014) 251–270.
[42] S.A. Gershgorin, Über die abgrenzung der eigenwerte einer matrix, Bull. de l'Acad. des Sci. de l'URSS, Classe des Sci. Math. et na 6 (1931) 749–754.
[43] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, et al., SciPy 1.0: Fundamental algorithms for scientific computing in Python, Nat. Methods 17 (2020) 261–272.
[44] C. Audouze, F. D. Vuyst, P.B. Nair, Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations, Numer. Methods Partial Differential Equations 29 (5) (2013) 1587–1628.
[45] J.S. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, J. Comput. Phys. 363 (2018) 55–78.
[46] V. Puri, A. Prakash, L.B. Kara, Y.J. Zhang, SNF-ROM: Projection-based nonlinear reduced order modeling with smooth neural fields, 2024, arXiv:2405.14890.
[47] K. Carlberg, C. Bou-Mosleh, C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, Internat. J. Numer. Methods Engrg. 86 (2) (2011) 155–181.
[48] S.E. Ahmed, S. Pawar, O. San, A. Rasheed, T. Iliescu, B.R. Noack, On closures for reduced order models - A spectrum of first-principle to machine-learned avenues, Phys. Fluids 33 (9) (2021).
[49] A. Prakash, Y.J. Zhang, Projection-based reduced order modeling and data-driven artificial viscosity closures for incompressible fluid flows, Comput. Methods Appl. Mech. Engrg. 425 (2024) 116930.