Physics of Fluids

Memory embedded non-intrusive reduced order modeling of non-ergodic flows

Cite as: Phys. Fluids **31**, 126602 (2019); https://doi.org/10.1063/1.5128374
Submitted: 18 September 2019 . Accepted: 03 December 2019 . Published Online: 23 December 2019

Shady E. Ahmed, Sk. Mashfiqur Rahman, Omer San, Adil Rasheed, and Ionel M. Navon

View Online Export Citation CrossMark

Scilight
Highlights of the best new research in the physical sciences
LEARN MORE!
AIP Publishing

Memory embedded non-intrusive reduced order modeling of non-ergodic flows

Cite as: Phys. Fluids 31, 126602 (2019); doi: 10.1063/1.5128374
Submitted: 18 September 2019 • Accepted: 3 December 2019 •
Published Online: 23 December 2019

Shady E. Ahmed,[1] Sk. Mashfiqur Rahman,[1] Omer San,[1,a)] Adil Rasheed,[2] and Ionel M. Navon[3]

AFFILIATIONS

[1] School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, Oklahoma 74078, USA
[2] Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7465 Trondheim, Norway
[3] Department of Scientific Computing, Florida State University, Tallahassee, Florida 32306, USA

[a)] **Electronic mail:** osan@okstate.edu

ABSTRACT

Generating a digital twin of any complex system requires modeling and computational approaches that are efficient, accurate, and modular. Traditional reduced order modeling techniques are targeted at only the first two, but the novel nonintrusive approach presented in this study is an attempt at taking all three into account effectively compared to their traditional counterparts. Based on dimensionality reduction using proper orthogonal decomposition (POD), we introduce a long short-term memory neural network architecture together with a principal interval decomposition (PID) framework as an enabler to account for localized modal deformation. As an effective partitioning tool for breaking the Kolmogorov barrier, our PID framework, therefore, can be considered a key element in the accurate reduced order modeling of convective flows. Our applications for convection-dominated systems governed by Burgers, Navier-Stokes, and Boussinesq equations demonstrate that the proposed approach yields significantly more accurate predictions than the POD-Galerkin method and could be a key enabler toward near real-time predictions of unsteady flows.

I. INTRODUCTION

There are a great number of high-dimensional problems in the field of science and engineering (such as atmospheric flows) that can be efficiently modeled based on embedded low-dimensional structures or reduced order models (ROMs).[1–3] ROMs have great promise for flow control,[4–11] data assimilation,[12–22] parameter estimation,[23–25] and uncertainty quantification.[26–31] These applications typically require multiple forward simulations of the problem being investigated, and even the most powerful supercomputers might fail to perform these simulations in full order model (FOM) space due to storage and speed limitations. Also, building of digital twins[32–36] requires real-time and many-query responses. A digital twin can be defined as the virtual representation of a physical object or system across its lifecycle using real-time data[37] which requires an efficient and on-the-fly simulation emulator. The concept of using such interactive computational megamodels often pays off in terms of accelerated design cycle times, greater efficiency and safety, predictive maintenance and scheduling, more efficient and informed decision support systems, real-time monitoring, performance optimization, supervisory control to reduce energy consumption, and perhaps much beyond. With the recent wave of digitization, reduced order modeling can be viewed as one of the key enablers to bring the promise of the digital twinning concept closer to reality.[38] Therefore, there is a continuous demand for the development of accurate ROMs for complex physical phenomena. In projection-based ROMs, the most widely used technique, the discrete high-dimensional operators are projected onto a lower-dimensional space, so that the problem can be solved more efficiently in this reduced space.[39–44]

One of the earliest developed and well-known approaches to build this reduced space is Fourier analysis. However, it assumes universal basis functions (or modes) which have no specific relation to the physical system. On the other hand, snapshot-based model reduction techniques tailor a reduced space that best fits the problem by extracting the underlying coherent structures that control the major dynamical evolution we are interested in. Proper orthogonal decomposition (POD) is a very popular and well-established approach extracting the modes which contribute the most to the total variance.[45,46] In fluid dynamics applications, where we are mostly interested in the velocity field, those modes contain the largest amount of kinetic energy.[47,48] That is why POD is usually classified as an energy-based decomposition method. Another popular approach for model order reduction is the dynamic mode decomposition (DMD)[49–54] which generates a number of modes, each characterized by an oscillating frequency and growth/decay rate. In this study, we are interested in the application of POD for dimensionality reduction.

POD generates a set of spatial orthonormal basis functions, each containing a significant amount of total energy. To obtain a reduced representation of a system, the first few modes are selected, and the remaining are truncated assuming their contribution to the system's behavior is minimum (i.e., compression). The Kolmogorov n-width[55,56] provides a mathematical guideline to quantify the optimal n-dimensional linear subspace and the associated projection error (i.e., a measure of the system's linear reducibility). The Kolmogorov n-width, $d_n(\mathcal{M})$, is defined as[56–58]

$$d_n(\mathcal{M}) := \inf_{\mathcal{S}_n} \sup_{q\in\mathcal{M}} \|q - \Pi_{\mathcal{S}_n} q\|$$

where $\mathcal{S}_n$ is a linear n-dimensional subspace, $\mathcal{M}$ is the solution manifold, and $\Pi_{\mathcal{S}_n}$ is the orthogonal projector onto $\mathcal{S}_n$. The Kolmogorov n-width is a classical concept of approximation theory which describes the error, in worst-case scenario (therefore, the supremum appears), that might arise from a projection onto the best-possible linear subspace of a given dimension n (hence, the infimum is taken over all n-dimensional subspaces). That is, the Kolmogorov n-width represents a barrier on the system's linear reducibility. If the decay of the Kolmogorov n-width is fast, then employing a reduced linear subspace is feasible. Unfortunately, this is not generally the case for convection-dominated flows with severe temporal evolution or equivalently parametric situations where the solution is highly varying over the parameter space. In these situations, the decay of n-width is relatively slow, hence raising the Kolmogorov barrier (i.e., requiring more modes to be retained in the reduced-space approximation of the underlying dynamics).

Moreover, snapshots-based model reduction techniques rely, in principle, on the ergodicity hypothesis. According to this hypothesis, time average and ensemble average are equal which implies that any collection of random samples should be able to represent the whole process. That is the system's response given certain inputs that are considered to encapsulate the essential behavior and characteristics of that system. In literature, a substantial amount of studies have been devoted to investigate the ergodicity of different fluid flow systems.[59–66] We can see that the flow situations described here (e.g., convection-dominated flows) often fail to fulfill this hypothesis. As a consequence, the resulting intrinsically global POD modes cannot describe the underlying flow structures. In problems with strong convective nature, the system's state is significantly different at different time instances. Consequently, a global POD application on these systems causes averaging and deformation of POD modes in such a way that they become no more representative of any of the system's states. Principal interval decomposition (PID)[67–69] offers a treatment of this modal deformation by dividing the temporal space into a few intervals, where POD (or any other decomposition technique) is applied locally. Therefore, local POD modes are tailored to the specific flow behavior in respective time subintervals. Similarly, local POD modes can be constructed by partitioning the physical domain,[70,71] state space,[72,73] or parameter space. Indeed, this partitioning approach helps to break the Kolmogorov barrier as well. Using a single fixed global subspace would necessitate keeping a larger number of modes to meet accuracy requirements and capture a certain amount of energy. Alternatively, partitioning allows the use of several tailored, local, and lower-dimensional subspaces. Also, Lee and Carlberg[74] recently proposed a projection onto nonlinear manifolds to break the Kolmogorov barrier instead of working with linear manifolds.

POD has been often coupled with Galerkin projection to build ROMs for linear and nonlinear systems.[75–82] In Galerkin projection, the governing equations are projected onto the POD subspace. Through the orthonormality and energy-optimality characteristics of POD modes, a simpler and truncated set of coupled ordinary differential equations (ODEs) is obtained. The resulting system is low-dimensional, but it is dense. In other words, it generates small and full matrices, while the common discretization techniques often lead to large and sparse matrices. Also, the quadratic nonlinearity and triadic interactions in ROMs lead to a computational load of an order of $O(R^3)$, where R is the number of retained modes. The discrete empirical interpolation method[83] (DEIM), the discrete version of the empirical interpolation method,[84] can be used to reduce the computational complexity for nonlinear ROMs where the nonlinear term is approximated with sparse sampling through projecting it onto a separate reduced subspace, rather than directly computing it.[85–87] However, in this paper, we utilize a tensorial POD projection approach,[88] where the projected spatial variables are stored in tensors and calculated offline. Furthermore, symbolic regression techniques might serve to identify ROMs from limited sensor data.[89] Another class for ROM which is gaining popularity in recent years is the fully data-driven, nonintrusive ROM (NIROM).[90–98] NIROM is a family of methods that solely access available datasets to extract and mimic the system's dynamics, with little-to-no knowledge of the governing equations. The nonintrusive approach is sometimes called physics-agnostic modeling, in contrast to the intrusive physics-informed approach.

One of the main advantages of a nonintrusive approach is its portability, which results from the fact that it does not necessarily require the exact form of the equations and the methods used to solve them to generate the snapshots. This makes the approach applicable to experimental data where the equations are often not well established or have huge uncertainties involved in their parameters. Together with their modularity and simplicity, nonintrusive models offer a unique advantage in multidisciplinary collaborative environments. It is often necessary to share the data or the model without revealing the proprietary or sensitive information. Different departments or subcontractors can easily exchange data (with standardized I/O) or executables, securing their intangible assets and intellectual property rights. Furthermore, nonintrusive approaches are particularly useful when the detailed governing equations of the problem are unknown. This modeling approach can benefit from the enormous amount of data collected from experiments, sensor measurements, and large-scale simulations to build a robust and accurate ROM technology.

With the growing advancement of artificial neural networks (ANNs) and other machine learning (ML) techniques and the availability of massive amounts of data resources from high-fidelity simulations, field measurements, and experiments, data-driven, nonintrusive modeling approaches are currently considered some of the most promising methods across different scientific and research communities. In the past few years, there has been a significant amount of research using ANN and ML techniques dedicated to turbulence modeling.[99–108] More details on the influence of ML on fluid mechanics, specifically turbulence modeling, can be found in recent studies.[109–113] Until recently, the fully nonintrusive modeling can be considered the most attractive enabling methodology to do real-time simulation very efficiently in the context of emerging digital twin technologies.[114] In a complimentary fashion, the hybrid models[115–120] are developed by combining the intrusive and nonintrusive models in such way that the limitation of one component modeling strategy can be addressed by the other component model.

In this work, we propose a nonintrusive reduced order modeling framework that is best suited for unsteady flows, where the convective mechanisms are more predominant than the diffusive ones. The approach is based on principal interval decomposition to parse the data over time to learn more localized dominant structures. This is particularly important for problems where we observe relatively slow decay in the Kolmogorov n-width, which constrains the feasibility of reduced order approximation. Also, this partitioning helps to satisfy the ergodicity hypothesis within each local interval to provide a good approximation of the flow field. We couple this parsing technique with a long short-term memory (LSTM) neural network, which is a very efficient ML tool for time-series predictions. The PID-LSTM is compared with its counterpart based on Galerkin projection (PID-GP). PID-LSTM not only eliminates the need to access the governing equations, being solely dependent on data, but also helps to mitigate instabilities in the ROM predictions resulting from the nonlinear interaction among different fields. We tested the proposed framework with one-dimensional and two-dimensional convection-dominated problems, highlighting its benefits over the standard POD and Galerkin projection approaches.

The rest of the paper is outlined as follows. Section II provides a brief overview of the mathematical models used to test the proposed framework. Namely, we describe the one-dimensional nonlinear advective Burgers problem, a standard benchmark problem in computational fluid dynamic (CFD) studies. Also, we tested our framework using the two-dimensional Navier-Stokes equations. More specifically, we investigate the vortex merger and double shear layer problems. As a final and more complicated problem, we study the unsteady lock-exchange flow problem (also known as the Marsigli flow) governed by the two-dimensional Boussinesq equations. In Sec. III, we describe the classical proper orthogonal decomposition approach for order reduction. A generalized PID framework, an approach to construct local basis rather than global ones, is shown in Sec. IV. The application of PID with classical Galerkin projection to build intrusive ROMs is outlined in Sec. V. The proposed approach which incorporates PID while bypassing GP in a nonintrusive framework is illustrated in Sec. VI. Results obtained with the novel proposed method to illustrate its advantages are followed by relevant discussions in Sec. VII. Finally, conclusions are provided in Sec. VIII.

II. MATHEMATICAL MODELS

A. 1D Burgers equation

Our first test case is the one-dimensional (1D) viscous Burgers equation. It represents a simple form of Navier-Stokes equations in 1D setting with similar quadratic nonlinear interaction and Laplacian dissipation. It is, therefore, considered a standard benchmark for the analysis of nonlinear advection-diffusion problems.

The evolution of the velocity field $u(x, t)$, in a dimensionless form, is given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{\text{Re}}\frac{\partial^2 u}{\partial x^2}, \tag{1}$$

where Re is the dimensionless Reynolds number, defined as the ratio of inertial effects to viscous effects.

B. 2D Navier-Stokes equations

The primitive formulation of the two-dimensional (2D) Navier-Stokes equations, in the dimensionless form with the index notation, can be written as

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{2}$$

$$\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{\text{Re}}\frac{\partial^2 u_i}{\partial x_j \partial x_j}, \tag{3}$$

where u_i is the velocity in the i-th direction and p is the pressure. We note here that the index notation in the above equations follows the Einstein convention. In particular, the incompressibility condition [Eq. (2)] can be rewritten as $\nabla \cdot \mathbf{u} = 0$, where $\mathbf{u} = [u, v]^T$ is the velocity vector. Alternatively, by taking the curl of Eq. (3), the following vorticity-streamfunction formulation of the 2D Navier-Stokes equation is obtained:

$$\frac{\partial \omega}{\partial t} + \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y} = \frac{1}{\text{Re}}\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right), \tag{4}$$

where ω is the vorticity and ψ is the streamfunction. The vorticity field is defined as $\boldsymbol{\omega} = \nabla \times \mathbf{u}$. Although the vorticity is a vector field by definition, we are considering only the z-component in our analysis (i.e., $\omega = \boldsymbol{\omega} \cdot \hat{k}$, where $\hat{k}$ is a unit vector in the z-direction). Since we are dealing with 2D flows in the x-y plane, only the z-component becomes nontrivial. Streamfunction is a scalar field, defined for two-dimensional flows as $\mathbf{u} = \nabla \times \psi\hat{k}$, specifically,

$$u = \frac{\partial \psi}{\partial y}, \qquad v = -\frac{\partial \psi}{\partial x}. \tag{5}$$

The vorticity-streamfunction formulation has several computational advantages over the primitive variable formulation. It prevents the odd-even decoupling issues that might arise between pressure and velocity components. Therefore, a collocated grid can be used instead of using a staggered one without producing any spurious modes. Also, it automatically enforces the incompressibility condition. The kinematic relationship between vorticity and streamfunction is given by the following Poisson equation:

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega. \tag{6}$$

This relationship implies that the streamfunction is not a prognostic variable and can be computed from the vorticity field at each timestep. We will also use this property in our development of the intrusive ROMs with Galerkin projection in Sec. V. To shorten our notation, we can define the Jacobian, $J(f, g)$, and the Laplacian, $\nabla^2 f$, operators as follows:

$$J(f,g) = \frac{\partial f}{\partial x}\frac{\partial g}{\partial y} - \frac{\partial f}{\partial y}\frac{\partial g}{\partial x}, \tag{7}$$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \tag{8}$$

Equations (4) and (6) can be rewritten as

$$\frac{\partial \omega}{\partial t} + J(\omega,\psi) = \frac{1}{\mathrm{Re}}\nabla^2\omega, \tag{9}$$

$$\nabla^2\psi = -\omega. \tag{10}$$

This form of the 2D Navier-Stokes equations will be used in the remaining text.

C. 2D Boussinesq equations

Boussinesq equations represent a simple approach for modeling geophysical waves such as oceanic and atmospheric circulations induced by temperature differences.[121] The Boussinesq approximation enables us to solve nonisothermal flows (e.g., natural convection), without having to solve for the full compressible formulation of Navier-Stokes equations. Also, it can be equally useful for other situations, such as isothermal flows with density stratification. In this approximation, variations of all fluid properties other than the density are ignored completely. Moreover, the density dependence is ignored in all terms except for gravitational force (giving rise to buoyancy effects). As a result, the continuity equation is used in its constant density form, and the momentum equation can be simplified significantly.

The dimensionless form of the 2D incompressible Boussinesq equations on a domain Ω in vorticity-streamfunction formulation is given by the following two coupled scalar transport equations:[122,123]

$$\frac{\partial \omega}{\partial t} + J(\omega,\psi) = \frac{1}{\mathrm{Re}}\nabla^2\omega + \mathrm{Ri}\frac{\partial\theta}{\partial x}, \tag{11}$$

$$\frac{\partial \theta}{\partial t} + J(\theta,\psi) = \frac{1}{\mathrm{RePr}}\nabla^2\theta, \tag{12}$$

where θ is the temperature. In Boussinesq flow systems, there are three leading physical mechanisms, namely, viscosity, conductivity, and buoyancy. Equations (11) and (12) include three dimensionless numbers: Re, Ri, and Pr. Re is the dimensionless Reynolds number, relating viscous effects and inertial effects as defined in Sec. II A. The Richardson number, Ri, is the ratio of buoyancy force to inertial force, and the Prandtl number, Pr, is the ratio between kinematic viscosity and heat conductivity. Boussinesq approximation underlies the statement that dynamical similarity of free convective flows depends on the Grashof and Prandtl numbers,[124] where the Grashof number, Gr, is defined as Gr = RiRe^2. In natural convection heat transfer, other relevant dimensionless numbers are often used, such as Rayleigh number, Ra = GrPr, and Péclet number, Pe = RePr.

III. PROPER ORTHOGONAL DECOMPOSITION

In POD, the dominant spatial subspaces are extracted from a given dataset. In other words, POD computes the dominant coherent directions in an infinite space, which best describe the spatial evolution of a system. POD-ROM is closely related to either singular value decomposition or eigenvalue decomposition of the snapshot matrix (in the finite-dimensional case). However, in most fluid flow simulations of interest, the number of degrees of freedom (number of grid points) is often orders of magnitude larger than the number of collected datasets. This results in a tall and skinny matrix, which makes the conventional direct decomposition inefficient, as well as time and memory consuming. Therefore, we follow the method of snapshots, proposed by Sirovich,[125] to generate the POD bases efficiently. A number of snapshots (or realizations), N_s, of the flow field, denoted as $u(\mathbf{x}, t_n)$, are stored at consecutive times t_n for $n = 0, 1, 2, \ldots, N_s$. The field $u(\mathbf{x}, t_n)$ is assumed to be square-integrable in the Hilbert space, $u(\mathbf{x}, t_n) \in L^2(\mathcal{D}, \mathcal{T})$ with $\mathbf{x} \in \mathcal{R}^N, N = 1, 2, 3, \ldots$ and $t_n \in \mathcal{T} = [0, T]$. From a physical point of view, square-integrability corresponds to a finite amount of kinetic energy in the field. The time-averaged field, called "base flow," can be computed as

$$\bar{u}(\mathbf{x}) = \frac{1}{N_s}\sum_{n=0}^{N_s} u(\mathbf{x}, t_n). \tag{13}$$

The mean-subtracted snapshots, also called anomaly or fluctuation fields, are then computed as the difference between the instantaneous field and the mean field,

$$u'(\mathbf{x}, t_n) = u(\mathbf{x}, t_n) - \bar{u}(\mathbf{x}). \tag{14}$$

This subtraction has been common in ROM community, and it guarantees that ROM solution would satisfy the same boundary conditions as the full order model (FOM).[51] This anomaly field procedure can also be interpreted as a mapping of snapshot data to their origin.

Then, an $N_s \times N_s$ snapshot data matrix $\mathbf{A} = [a_{ij}]$ is computed from the inner product of mean-subtracted snapshots,

$$a_{ij} = \langle u'(\mathbf{x}, t_i); u'(\mathbf{x}, t_j)\rangle, \tag{15}$$

where the angle-parenthesis denotes the inner product defined as

$$\langle q_1(\mathbf{x}); q_2(\mathbf{x})\rangle = \int_\Omega q_1(\mathbf{x}) q_2(\mathbf{x}) d\mathbf{x}. \tag{16}$$

It turns out that $\mathbf{A}$ is a nonnegative, positive-semidefinite Hermitian matrix, also called the Gramian matrix. An eigenvalue decomposition of $\mathbf{A}$ is carried out as

$$\mathbf{AV} = \mathbf{V\Lambda}, \tag{17}$$

where $\mathbf{\Lambda}$ is a diagonal matrix whose entries are the nonnegative eigenvalues λ_k of $\mathbf{A}$, and $\mathbf{V}$ is a matrix whose columns $\mathbf{v}_k$ are the corresponding eigenvectors. It should be noted that these eigenvalues need to be arranged in a descending order (i.e., $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{N_s}$) for proper selection of the POD modes. In general, the eigenvalues, λ_k, represent the respective POD mode contribution to the total variance. In the case of velocity time series, it represents the turbulent kinetic energy. The POD modes ϕ_k are then computed as

$$\phi_k(\mathbf{x}) = \frac{1}{\sqrt{\lambda_k}} \sum_{n=0}^{N_s} v_k^n u'(\mathbf{x}, t_n), \tag{18}$$

where v_k^n is the n-th component of the eigenvector $\mathbf{v}_k$. The scaling factor, $1/\sqrt{\lambda_k}$, is to guarantee the orthonormality of POD modes, i.e., $\langle \phi_i; \phi_j \rangle = \delta_{ij}$, where δ_{ij} is the Kronecker delta. Using these basis functions, we can represent our reduced-order approximation of the field as follows:

$$u(\mathbf{x}, t_n) = \bar{u}(\mathbf{x}) + \sum_{k=1}^{R} \alpha_k(t_n)\phi_k(\mathbf{x}), \tag{19}$$

$$\alpha_k(t_n) = \big\langle u(\mathbf{x}, t_n) - \bar{u}(\mathbf{x}); \phi_k(\mathbf{x}) \big\rangle, \tag{20}$$

where R is the number of retained modes ($R \ll N_s \ll N$), N_s is the number of collected snapshots, and N is the spatial dimension (number of grid points). POD is optimal in the sense that it provides the most efficient way (with respect to other linear representations) of capturing the dominant components of an infinite-dimensional process with only finite, and few, modes. The POD modes can be interpreted geometrically as the principal axes of the cloud of data points, $\{u(\mathbf{x}, t_n)\}_{n=0}^{N_s}$, in the N-dimensional vector space.

From a mathematical perspective, the set of POD modes, $\mathbf{\Phi} = \{\phi_k\}_{k=1}^{R}$, represents the solution of the following optimization problem:[45]

$$\max_{\{\phi_k\}_{k=1}^{R}} \quad \sum_{n=0}^{N_s}\sum_{k=1}^{R} \Big| \big\langle u'(\mathbf{x}, t_n); \phi_k(\mathbf{x}) \big\rangle \Big|^2,$$

$$\text{subject to} \quad \phi_k \in L^2(\mathcal{D}), \quad \big\langle \phi_i; \phi_j \big\rangle = \delta_{ij}.$$

A more in-depth discussion about mathematical aspects of POD and its optimality can be found in the rigorous discussions by Berkooz *et al.*[126] and Holmes *et al.*[127]

IV. PRINCIPAL INTERVAL DECOMPOSITION

The classical POD approach, presented in Sec. III, produces a set of modes, or bases, that contains the largest amount of snapshot energy, but in average sense. Intrinsically, this results in global modes that are most similar to the overall flow. In general, POD-based ROMs work well for relatively smooth and ergodic systems with rapid decay in the Kolmogorov n-width. For those, only the first few R modes are sufficient to represent the system with acceptable accuracy, where $R \ll N_s \ll N$, and the remaining $(N_s - R)$ modes are truncated with minimal effect. However, in nonlinear convective flow problems, this is not always the case. As a consequence, energy is widely distributed over a large number of modes. Therefore, the truncated modes possess a significant amount of total energy, and an increased number of modes need to be retained in order to describe the system in hand properly. Closure models and stabilization schemes were proven to improve the performance of ROMs via approximating the effects of truncated modes.[128–132] Despite this, the accuracy gain from closure and stabilization only is limited in these scenarios.

A major source of accuracy loss in the application of POD in convection-dominated flows comes from the global nature of the POD approach, which results in overall deformation of the obtained modes by the rapidly varying flow field states. As a result, the constructed POD modes do not resemble any of the flow states and cannot capture any dominant structure at all. Furthermore, excursions in state space that contain a small amount of energy can be overlooked by POD because their contribution to the total energy may be negligible. These excursions can, however, be of interest and have significant impact on the dynamical evolution (see Cazemier *et al.*[133] for example). To address these issues, we follow and extend the principal interval decomposition (PID) approach, first presented by IJzerman.[134] The main purpose of PID is to replace the global POD modes, with localized ones. This is accomplished by dividing the whole time domain $\mathcal{T}$ into a number N_p of nonoverlapping time windows, $\tau_1, \tau_2, \ldots, \tau_{N_p}$, where $\mathcal{T} = \cup_{p=1}^{N_p} \tau_p$. We denote $\kappa^{(p)}$ as the index of the time instance at the interface between the consecutive subintervals τ_p and τ_{p+1} (i.e., $\tau_p \cap \tau_{p+1} = t_{\kappa^{(p)}}$, $p = 1, 2, \ldots, N_p - 1$). That is,

$$\tau_p = [t_{\kappa^{(p-1)}}, t_{\kappa^{(p)}}]. \tag{21}$$

Here, we restrict ourselves to equally spaced time intervals although an adaptive partitioning approach may be performed.[135] A set of local basis functions $\mathbf{\Phi}^{(p)} = \{\phi_k^{(p)}\}_{k=1}^{R^{(p)}}$ is constructed for each time window τ_p, following the same standard procedure described in Sec. III, within each subinterval, where $\phi_k^{(p)}$ is the k-th mode in the p-th interval and $R^{(p)}$ is the number of modes in this interval. Even though the approach is applicable for different number of modes in each interval, we will continue our discussion assuming a fixed number of modes per interval [i.e., $R^{(p)} = R$, $p = 1, 2, \ldots, N_p$]. It should be noted that local mean fields are also constructed within each interval as

$$\bar{u}^{(p)}(\mathbf{x}) = \frac{1}{N_s/N_p} \sum_{n=\kappa^{(p-1)}}^{\kappa^{(p)}} u(\mathbf{x}, t_n), \tag{22}$$

where $\bar{u}^{(p)}(\mathbf{x})$ is the mean field over the subinterval τ_p. Our reduced-rank approximation becomes

$$u(\mathbf{x}, t_n) = \bar{u}^{(p)}(\mathbf{x}) + \sum_{k=1}^{R} \alpha_k^{(p)}(t_n)\phi_k^{(p)}(\mathbf{x}), \tag{23}$$

$$\alpha_k^{(p)}(t_n) = \big\langle u(\mathbf{x}, t_n) - \bar{u}^{(p)}(\mathbf{x}); \phi_k^{(p)}(\mathbf{x}) \big\rangle, \quad \forall\ t_{\kappa^{(p-1)}} \le t_n \le t_{\kappa^{(p)}}. \tag{24}$$

Although it might seem that PID would be more computationally costly (implementing the standard POD procedure N_p times), the actual time to perform PID is reduced, as reported in Sec. VII. This is caused by solving a number N_p of smaller $(N_s/N_p \times N_s/N_p)$ eigenvalue problems rather than solving one big $(N_s \times N_s)$ eigenvalue problem.[68]

V. INTRUSIVE REDUCED ORDER MODELING

In order to build intrusive reduced order models, we apply the standard Galerkin projection to our nonlinear systems presented in Sec. II. First, the governing equations [i.e., Eqs. (1), (9), (11), and (12)] need to be rearranged in semidiscretized form using linear and nonlinear operators as follows:

$$\frac{\partial q}{\partial t} = \mathcal{L} + \mathcal{N}, \tag{25}$$

where q is u in the Burgers equation, ω in the Navier-Stokes problem, and $[\omega, \theta]$ in the Boussinesq case. The linear and nonlinear operators are summarized in Table I. Then, the reduced order approximation

TABLE I. Linear and nonlinear operators for mathematical models introduced in Sec. II.

q	$\mathcal{L}$	$\mathcal{N}$
1D Burgers		
u	$\dfrac{1}{\mathrm{Re}}\dfrac{\partial^2 u}{\partial x^2}$	$-u\dfrac{\partial u}{\partial x}$
2D Navier-Stokes		
ω	$\dfrac{1}{\mathrm{Re}}\nabla^2\omega$	$-J(\omega, \psi)$
2D Boussinesq		
ω	$\dfrac{1}{\mathrm{Re}}\nabla^2\omega + \mathrm{Ri}\dfrac{\partial\theta}{\partial x}$	$-J(\omega, \psi)$
θ	$\dfrac{1}{\mathrm{RePr}}\nabla^2\theta$	$-J(\theta, \psi)$

[i.e., Eq. (19) or Eq. (23)] is plugged into Eq. (25), and Galerkin projection is applied by multiplying Eq. (25) with the basis functions $\phi_k^{(p)}$ and integrating over the domain. The orthonormality property of POD modes can be used to reduce the equations into a set of coupled ordinary differential equations (ODEs) in the POD coefficients, α_k.

A summary of the obtained ROM equations from applying the principal interval decomposition approach with Galerkin projection (PID-GP) is given below. Details of the derivation can be found elsewhere.[68,132,136]

A. 1D Burgers equation

The reduced-rank approximation for the Burgers problem along with the dynamical evolution equation for the temporal coefficients using the PID approach coupled with Galerkin projection (PID-GP) can be written as

$$u(x,t) = \bar{u}^{(p)}(x) + \sum_{k=1}^{R} \alpha_k^{(p)}(t)\phi_k^{(p)}(x) \qquad \forall\ t_{\kappa(p-1)} \le t \le t_{\kappa(p)}, \tag{26}$$

$$\frac{\mathrm{d}\alpha_k^{(p)}}{\mathrm{d}t} = \mathfrak{B}_k^{(p)} + \sum_{i=1}^{R}\mathfrak{L}_{i,k}^{(p)}\alpha_i^{(p)} + \sum_{i=1}^{R}\sum_{j=1}^{R}\mathfrak{N}_{i,j,k}^{(p)}\alpha_i^{(p)}\alpha_j^{(p)}, \quad k = 1,2,\ldots,R, \tag{27}$$

where $\mathfrak{B}$, $\mathfrak{L}$, and $\mathfrak{N}$ are predetermined model coefficients corresponding to constant, linear, and nonlinear terms, respectively. They are precomputed only once during the offline training phase as follows:

$$\begin{aligned}
\mathfrak{B}_k^{(p)} &= \left\langle \frac{1}{\mathrm{Re}}\frac{\partial^2 \bar{u}^{(p)}}{\partial x^2} - \bar{u}^{(p)}\frac{\partial \bar{u}^{(p)}}{\partial x};\, \phi_k^{(p)} \right\rangle,\\
\mathfrak{L}_{i,k}^{(p)} &= \left\langle \frac{1}{\mathrm{Re}}\frac{\partial^2 \phi_i^{(p)}}{\partial x^2} - \bar{u}^{(p)}\frac{\partial \phi_i^{(p)}}{\partial x} - \phi_i^{(p)}\frac{\partial \bar{u}^{(p)}}{\partial x};\, \phi_k^{(p)} \right\rangle,\\
\mathfrak{N}_{i,j,k}^{(p)} &= \left\langle -\phi_i^{(p)}\frac{\partial \phi_j^{(p)}}{\partial x};\, \phi_k^{(p)} \right\rangle.
\end{aligned}$$

For the sake of brevity in 2D cases, we shall drop the superscript (p) in the ROM equations below, but it should be noted that they are applicable intervalwise similar to the equations above.

B. 2D Navier-Stokes equations

In 2D Navier-Stokes equations, similar reduced-rank approximation and temporal evolution can be written as follows [after dropping the superscript (p)]:

$$\omega(x,y,t) = \bar{\omega}(x,y) + \sum_{k=1}^{R}\alpha_k(t)\phi_k^{\omega}(x,y), \tag{28}$$

$$\psi(x,y,t_n) = \bar{\psi}(x,y) + \sum_{k=1}^{R}\alpha_k(t)\phi_k^{\psi}(x,y), \tag{29}$$

$$\frac{\mathrm{d}\alpha_k}{\mathrm{d}t} = \mathfrak{B}_k + \sum_{i=1}^{R}\mathfrak{L}_{i,k}\alpha_i + \sum_{i=1}^{R}\sum_{j=1}^{R}\mathfrak{N}_{i,j,k}\alpha_i\alpha_j, \tag{30}$$

where

$$\begin{aligned}
\mathfrak{B}_k &= \left\langle -J(\bar{\omega},\bar{\psi}) + \frac{1}{\mathrm{Re}}\nabla^2\bar{\omega};\, \phi_k^{\omega}\right\rangle,\\
\mathfrak{L}_{i,k} &= \left\langle -J(\bar{\omega},\phi_i^{\psi}) - J(\phi_i^{\omega},\bar{\psi}) + \frac{1}{\mathrm{Re}}\nabla^2\phi_i^{\omega};\, \phi_k^{\omega}\right\rangle,\\
\mathfrak{N}_{i,j,k} &= \left\langle -J(\phi_i^{\omega},\phi_j^{\psi});\, \phi_k^{\omega}\right\rangle.
\end{aligned}$$

We can observe that the vorticity and streamfunction share the same time-dependent coefficients because they are related through a kinematic relationship, given by Eq. (6). Moreover, the mean field and spatial POD modes for streamfunction can be obtained from solving the following Poisson equations during the offline stage as POD preserves linear properties:

$$\nabla^2\bar{\psi}(x,y) = -\bar{\omega}(x,y), \tag{31}$$

$$\nabla^2\phi_k^{\psi}(x,y) = -\phi_k^{\omega}(x,y), \quad k = 1,2,\ldots,R. \tag{32}$$

This results in a set of basis functions for the streamfunction that are not necessarily orthonormal. The same procedure will be used in the 2D Boussinesq problem since it is also represented in vorticity-streamfunction formulation.

C. 2D Boussinesq equations

For 2D Boussinesq equations, the vorticity, streamfunction, and temperature fields can be written as

$$\omega(x,y,t) = \bar{\omega}(x,y) + \sum_{k=1}^{R}\alpha_k(t)\phi_k^{\omega}(x,y), \tag{33}$$

$$\psi(x,y,t_n) = \bar{\psi}(x,y) + \sum_{k=1}^{R}\alpha_k(t)\phi_k^{\psi}(x,y), \tag{34}$$

$$\theta(x,y,t_n) = \bar{\theta}(x,y) + \sum_{k=1}^{R}\beta_k(t)\phi_k^{\theta}(x,y), \tag{35}$$

where the temporal coefficients $\alpha_k(t)$ and $\beta_k(t)$ can be calculated from the following ODEs:

$$\frac{\mathrm{d}\alpha_k}{\mathrm{d}t} = \mathfrak{B}_k^{(\omega)} + \sum_{i=1}^{R} \mathfrak{L}_{i,k}^{(\omega,\psi)} \alpha_i + \sum_{i=1}^{R} \mathfrak{L}_{i,k}^{(\omega,\theta)} \beta_i + \sum_{i=1}^{R}\sum_{j=1}^{R} \mathfrak{N}_{i,j,k}^{(\omega,\psi)} \alpha_i \alpha_j, \tag{36}$$

$$\frac{\mathrm{d}\beta_k}{\mathrm{d}t} = \mathfrak{B}_k^{(\theta)} + \sum_{i=1}^{R} \mathfrak{L}_{i,k}^{(\theta,\psi)} \alpha_i + \sum_{i=1}^{R} \mathfrak{L}_{i,k}^{(\psi,\theta)} \beta_i + \sum_{i=1}^{R}\sum_{j=1}^{R} \mathfrak{N}_{i,j,k}^{(\theta,\psi)} \alpha_i \beta_j, \tag{37}$$

where the predetermined coefficients are calculated as

$$\begin{aligned}
\mathfrak{B}_k^{(\omega)} &= \big\langle -J(\bar{\omega},\bar{\psi}) + \frac{1}{\mathrm{Re}}\nabla^2\bar{\omega} + \mathrm{Ri}\frac{\partial\bar{\theta}}{\partial x}; \phi_k^{\omega}\big\rangle,\\
\mathfrak{B}_k^{(\theta)} &= \big\langle -J(\bar{\theta},\bar{\psi}) + \frac{1}{\mathrm{RePr}}\nabla^2\bar{\theta}; \phi_k^{\theta}\big\rangle,\\
\mathfrak{L}_{i,k}^{(\omega,\psi)} &= \big\langle \frac{1}{\mathrm{Re}}\nabla^2\phi_i^{\omega} - J(\phi_i^{\omega},\bar{\psi}) - J(\bar{\omega},\phi_i^{\psi}); \phi_k^{\omega}\big\rangle,\\
\mathfrak{L}_{i,k}^{(\omega,\theta)} &= \big\langle \mathrm{Ri}\frac{\phi_i^{\theta}}{\partial x}; \phi_k^{\omega}\big\rangle,\\
\mathfrak{L}_{i,k}^{(\theta,\psi)} &= \big\langle -J(\bar{\theta},\phi_i^{\psi}); \phi_k^{\theta}\big\rangle,\\
\mathfrak{L}_{i,k}^{(\psi,\theta)} &= \Big\langle \frac{1}{\mathrm{RePr}}\nabla^2\phi_i^{\theta} - J(\phi^{\theta},\bar{\psi}); \phi_k^{\theta}\Big\rangle,\\
\mathfrak{N}_{i,j,k}^{(\omega,\psi)} &= \big\langle -J(\phi_i^{\omega},\phi_j^{\psi}); \phi_k^{\omega}\big\rangle,\\
\mathfrak{N}_{i,j,k}^{(\theta,\psi)} &= \big\langle -J(\phi_i^{\theta},\phi_j^{\psi}); \phi_k^{\theta}\big\rangle.
\end{aligned}$$

For all cases, the initial conditions to initiate the ODE solver are obtained by projecting the initial field (mean-subtracted) onto the POD space of the first subinterval as

$$\alpha_k^{(p=1)}(t_0) = \big\langle u(\mathbf{x},t_0) - \bar{u}^{(p=1)}(\mathbf{x}); \phi_k^{(p=1)}(\mathbf{x})\big\rangle. \tag{38}$$

The only remaining part to close this section is to determine how to update the working manifold at the interface when moving from the p-th interval to the $(p+1)$-th interval [i.e., when $t = t_{\kappa(p)}$]. Once ROM solver reaches the end of the current interval, a reconstruction back to FOM space should be done. Subsequently, this reconstructed field is projected back onto the new basis functions. These two steps can be summarized as follows:

$$(1) \qquad u(\mathbf{x},t_{\kappa(p)}) = \bar{u}^{(p)}(\mathbf{x}) + \sum_{k=1}^{R} \alpha_k^{(p)}(t_{\kappa(p)})\phi^{(p)}(\mathbf{x}),$$

$$(2) \qquad \alpha_k^{(p+1)}(t_{\kappa(p)}) = \big\langle u(\mathbf{x},t_{\kappa(p)}) - \bar{u}^{(p+1)}(\mathbf{x}); \phi_k^{(p+1)}\big\rangle,$$

where the update step (manifold transfer) can be summarized as

$$\alpha_k^{(p+1)}(t_{\kappa(p)}) = \big\langle \bar{u}^{(p)}(\mathbf{x}) - \bar{u}^{(p+1)}(\mathbf{x}); \phi_k^{(p+1)}\big\rangle + \Big\langle \sum_{k=1}^{R} \alpha_k^{(p)}(t_{\kappa(p)})\phi^{(p)}(\mathbf{x}); \phi_k^{(p+1)}\Big\rangle. \tag{39}$$

This allows us to reinitiate our solver at the first timestep of the new time interval. Mathematically, this corresponds to imposing the following condition at the interface:[135]

$$\big\langle u^{(p)}(\mathbf{x},t_{\kappa(p)}) - u^{(p+1)}(\mathbf{x},t_{\kappa(p)}); \phi_k^{(p+1)}\big\rangle = 0. \tag{40}$$

VI. NONINTRUSIVE REDUCED ORDER MODELING

In this section, we devise the proposed nonintrusive PID-LSTM framework for unsteady convective flows. To illustrate the PID-LSTM framework, we depict a workflow schematic diagram in Fig. 1 for any arbitrary two-dimensional unsteady flow problem. In our two-dimensional representation of the PID-LSTM framework, **U** denotes any arbitrary two-dimensional field, for example, ω in Navier-Stokes and θ in Boussinesq test problems discussed in this study. However, with proper modification in the LSTM architecture, this framework can be utilized for any three-dimensional field data and one-dimensional field data, for example, the 1D Burgers case in our study. Also, this PID-LSTM framework is parallelization friendly (e.g., using the parareal framework[137,138]). As shown in Fig. 1, the first two stages of the offline training phase in the PID-LSTM framework are similar to the intrusive PID-GP framework described in Sec. V that we first split the stored high-fidelity field data snapshots into a desired number of intervals and then generate the basis functions as well as the true modal coefficients locally for each interval [i.e., using Eq. (24)].

For ROM dynamics, we replace the Galerkin projection of the PID-GP approach with an LSTM recurrent neural network (RNN) architecture to make the framework fully nonintrusive or data-driven. There have been a number of research efforts which showed that RNN, specifically LSTM as a variant of RNN, is capable of predicting the dependencies among temporal data sequences.[139–144] Hence, we utilize the LSTM neural network architecture to model and predict the time-varying modal coefficient data sequences for our nonintrusive ROM framework. In our PID-LSTM formulation, we train individual LSTM architectures for the modal coefficients from each interval which gives us the individual LSTM model for the respective PID interval. It should be noted that training the LSTM architectures is the computationally heaviest part of the overall framework which is done in the offline phase (the top-right box in Fig. 1). Before describing the online testing phase, we briefly describe the LSTM architecture utilized in our study.

The standard RNN architecture suffers from issues such as vanishing gradient problem[145] which led to the development of improved RNN architectures. LSTM is one of the most successful upgrades of the traditional RNN architecture which can learn and predict the temporal dependencies between the given data sequence based on the input information and previously acquired information.[146,147] The conventional LSTM operates by the cell states stored in the memory blocks and gating mechanisms to control the flow of information. Each memory block has an input gate controlling the flow of input activations into the cell, a forget gate for adaptive forgetting and resetting the cell's memory (to prevent overfitting by processing continuous inflow of input streams), and an output gate controlling the output flow of cell activations into the next cell. To demonstrate our LSTM architecture of this study, we can focus on the first interval only where the input sequential data matrix for training can be denoted as $\mathcal{X}_k$ and the output sequential data matrix $\mathcal{Y}_k$. Each sample of the input training matrix $\mathcal{X}_k$ is constructed as $\{\alpha_1^{(1)}(t_{n-\eta+1}),\ldots,\alpha_R^{(1)}(t_{n-\eta+1});\ldots;\alpha_1^{(1)}(t_{n-1}),\ldots,\alpha_R^{(1)}(t_{n-1});\alpha_1^{(1)}(t_n),\ldots,\alpha_R^{(1)}(t_n)\}$, and the corresponding output sample in the output sequential data matrix $\mathcal{Y}_k$ is

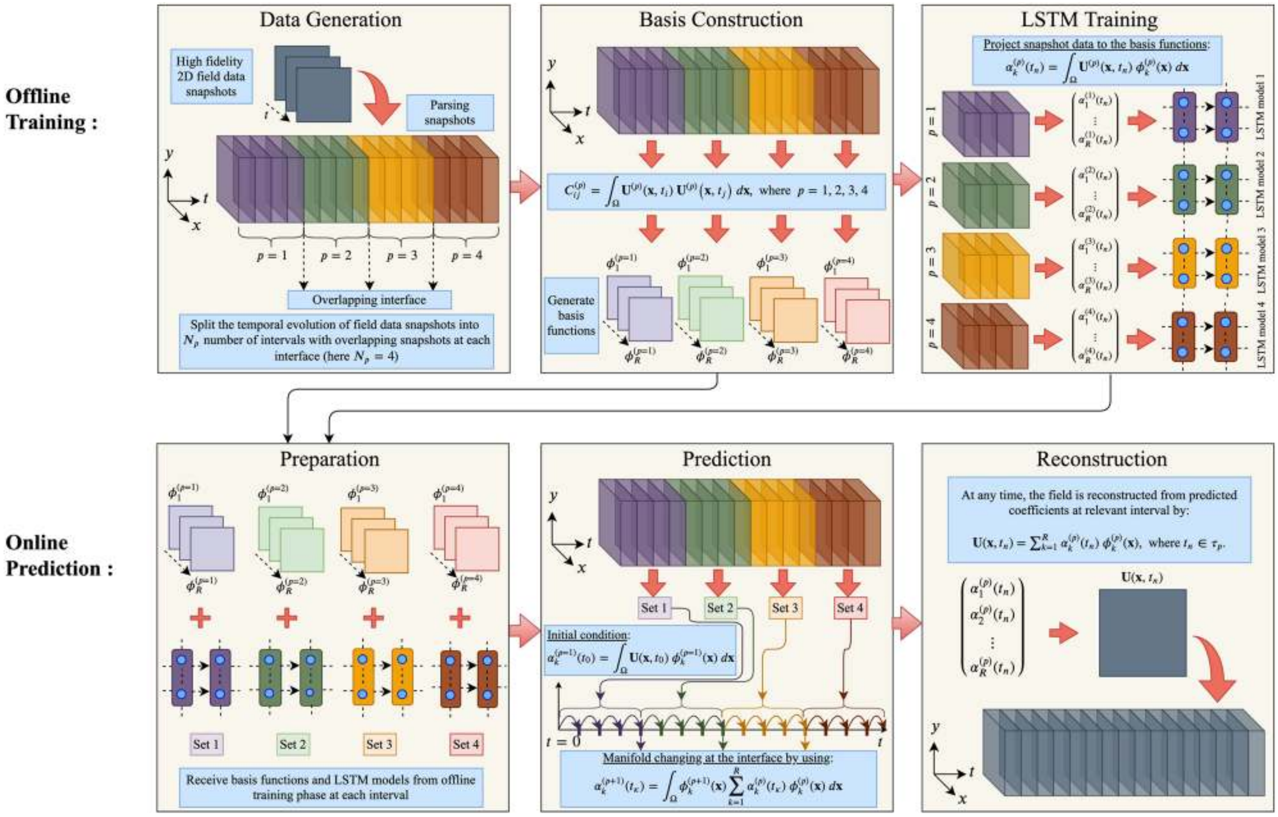
Offline Training :
Data Generation
High fidelity 2D field data snapshots
Parsing snapshots
t
y
t
x
$p = 1$
$p = 2$
$p = 3$
$p = 4$
Overlapping interface
Split the temporal evolution of field data snapshots into N_p number of intervals with overlapping snapshots at each interface (here $N_p = 4$)
Basis Construction
y
t
x
$C_{ij}^{(p)} = \int_\Omega \mathbf{U}^{(p)}(\mathbf{x}, t_i)\,\mathbf{U}^{(p)}(\mathbf{x}, t_j)\,d\mathbf{x}$, where $p = 1, 2, 3, 4$
Generate basis functions
$\phi_1^{(p=1)}$ $\phi_1^{(p=2)}$ $\phi_1^{(p=3)}$ $\phi_1^{(p=4)}$
$\phi_R^{(p=1)}$ $\phi_R^{(p=2)}$ $\phi_R^{(p=3)}$ $\phi_R^{(p=4)}$
LSTM Training
Project snapshot data to the basis functions:
$\alpha_k^{(p)}(t_n) = \int_\Omega \mathbf{U}^{(p)}(\mathbf{x}, t_n)\,\phi_k^{(p)}(\mathbf{x})\,d\mathbf{x}$
$p = 1$
$p = 2$
$p = 3$
$p = 4$
$\alpha_1^{(1)}(t_n)$ $\vdots$ $\alpha_R^{(1)}(t_n)$
$\alpha_1^{(2)}(t_n)$ $\vdots$ $\alpha_R^{(2)}(t_n)$
$\alpha_1^{(3)}(t_n)$ $\vdots$ $\alpha_R^{(3)}(t_n)$
$\alpha_1^{(4)}(t_n)$ $\vdots$ $\alpha_R^{(4)}(t_n)$
LSTM model 1
LSTM model 2
LSTM model 3
LSTM model 4
Online Prediction :
Preparation
$\phi_1^{(p=1)}$ $\phi_1^{(p=2)}$ $\phi_1^{(p=3)}$ $\phi_1^{(p=4)}$
$\phi_R^{(p=1)}$ $\phi_R^{(p=2)}$ $\phi_R^{(p=3)}$ $\phi_R^{(p=4)}$
Set 1
Set 2
Set 3
Set 4
Receive basis functions and LSTM models from offline training phase at each interval
Prediction
y
t
x
Set 1
Set 2
Set 3
Set 4
Initial condition:
$\alpha_k^{(p=1)}(t_0) = \int_\Omega \mathbf{U}(\mathbf{x}, t_0)\,\phi_k^{(p=1)}(\mathbf{x})\,d\mathbf{x}$
$t = 0$
t
Manifold changing at the interface by using:
$\alpha_k^{(p+1)}(t_\kappa) = \int_\Omega \phi_k^{(p+1)}(\mathbf{x}) \sum_{k=1}^{R} \alpha_k^{(p)}(t_\kappa)\,\phi_k^{(p)}(\mathbf{x})\,d\mathbf{x}$
Reconstruction
At any time, the field is reconstructed from predicted coefficients at relevant interval by:
$\mathbf{U}(\mathbf{x}, t_n) = \sum_{k=1}^{R} \alpha_k^{(p)}(t_n)\,\phi_k^{(p)}(\mathbf{x})$, where $t_n \in \tau_p$.
$\mathbf{U}(\mathbf{x}, t_n)$
$\alpha_1^{(p)}(t_n)$ $\alpha_2^{(p)}(t_n)$ $\vdots$ $\alpha_R^{(p)}(t_n)$
y
t
x

FIG. 1. The proposed nonintrusive principal interval decomposition LSTM framework for unsteady nonergodic flows.

$\left\{\alpha_1^{(1)}(t_{n+1}), \ldots, \alpha_R^{(1)}(t_{n+1})\right\}$. Here, η is the time history over which the LSTM model does the training and prediction recursively, called the number of lookbacks. In our study, a constant value of η is set as 5 for the test cases to avoid complexity while analyzing the results.

To illustrate the data stream flow through a standard LSTM cell, we have shown the sketch of a typical LSTM cell unit in Fig. 2. The basic LSTM equations to compute the gate functions can be given by

$$\mathcal{F}_k^{(1)}(t_n) = \xi\Big(W_{fh} h_k^{(1)}(t_{n-1}) + W_{f\mathcal{X}} \mathcal{X}_k^{(1)}(t_n) + b_f\Big), \tag{41}$$

$$\mathcal{I}_k^{(1)}(t_n) = \xi\Big(W_{ih} h_k^{(1)}(t_{n-1}) + W_{i\mathcal{X}} \mathcal{X}_k^{(1)}(t_n) + b_i\Big), \tag{42}$$

$$\mathcal{O}_k^{(1)}(t_n) = \xi\Big(W_{oh} h_k^{(1)}(t_{n-1}) + W_{o\mathcal{X}} \mathcal{X}_k^{(1)}(t_n) + b_o\Big), \tag{43}$$

where $\mathcal{I}$, $\mathcal{F}$, and $\mathcal{O}$ represent the input, forget, and output gates, respectively. The LSTM cell output activation vector or the hidden state vector is denoted as h, while ξ represents the logistic sigmoid function, b denotes the bias vectors, and W represents the weight matrices for each gate. Assuming the cell activation vector or internal cell state vector as C, the internal cell state equation can be expressed as

$$C_k^{(1)}(t_n) = \mathcal{F}_k^{(1)}(t_n) \odot C_k^{(1)}(t_{n-1}) + \mathcal{I}_k^{(1)}(t_n) \odot \widetilde{C}, \tag{44}$$

where $\widetilde{C} = \tanh\Big(W_{ch} h_k^{(1)}(t_{n-1}) + W_{c\mathcal{X}} \mathcal{X}_k^{(1)}(t_n) + b_c\Big)$ and $\odot$ is the Hadamard product of two vectors. The output state of each LSTM cell is given by

$$h_k^{(1)}(t_n) = \mathcal{O}_k^{(1)}(t_n) \odot \tanh\Big(C_k^{(1)}(t_n)\Big). \tag{45}$$

We utilize Keras API to design the LSTM architecture for our PID-LSTM framework.[148] The hyperparameters that we used in our numerical experiments implementing PID-LSTM are listed in Table II. The mean-squared error (MSE) is chosen as the loss function for weight-optimization, and an optimizer based on a variant of the stochastic gradient descent method, called ADAM,[149] is used to optimize the mean-squared loss. We utilize the tanh activation function in each LSTM layer. We select randomly 20% of the training data for validation during training. We maintain a constant hyperparameter setup to fairly compare the results for different numerical experiments. The training data are normalized by the minimum and maximum of each time series to be in between the range $[-1, +1]$.

In the online testing phase, we use the generated basis functions and trained LSTM models for each interval from the given snapshot data to recursively predict the coefficients until final time. For testing, the input of the first trained model will be the initial states of the first interval [see Eq. (38)]. When the online prediction reaches

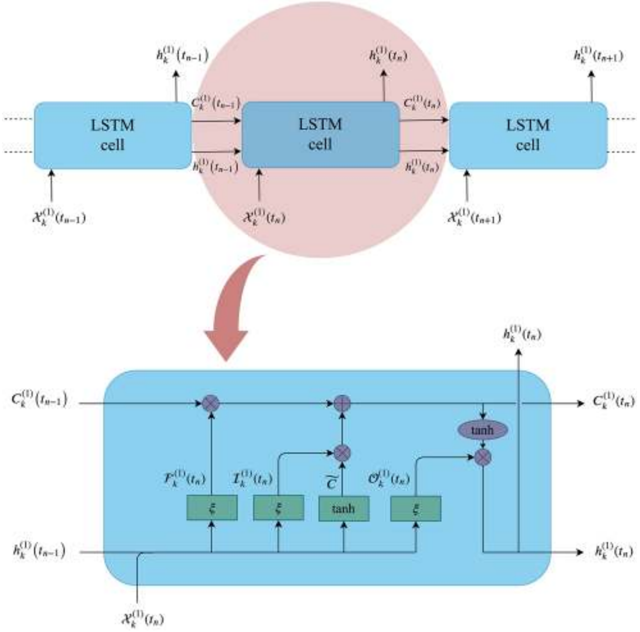

FIG. 2. Schematic of a typical LSTM cell unit.

the end of the subinterval, a manifold transfer is done at the interface by following Eq. (39) and the framework switches to the next LSTM model. By performing this procedure recursively, the modal coefficients until final time can be determined and then the field can be reconstructed at any time from the predicted coefficients at the relevant interval by using Eq. (23). Because the LSTM framework can have a wider interface than the PID-GP framework (i.e., for η > 1), a buffer zone can be defined at the interface, where Eq. (39) is applied for $t_{\kappa^{(p)}}, t_{\kappa^{(p)}-1}, \ldots, t_{\kappa^{(p)}-\eta+1}$. This would be the input for the subsequent LSTM model to predict $\alpha(t_{\kappa^{(p)}+1})$.

Before we demonstrate our framework using the test cases in Sec. VII, we would like to mention that our framework also helps in augmenting the stability of ROMs. This point is three-folds. First, regular Galerkin ROMs usually require closure/stabilization approaches to account for the effect of truncated modes due to the inherent nonlinearity. Second, for problems where more flow field variables are involved, the system becomes highly coupled and the instabilities and/or inaccuracies in any of these variables can result in further instabilities and/or inaccuracies in other flow variables. These two issues are mitigated using nonintrusive ROMs since the training/testing solely depends on the modal coefficients of the retained modes. Also, the mutual interactions between different flow variables can be diminished using different LSTMs for the variable of interest. Third, PID addresses the modal deformation and allows us to use a fewer number of modes in the ROM, which in turn reduces the modal truncation error and relevant instabilities. Indeed, we highlight that the accuracy gain due to closure in the investigated scenarios is minimum since the POD approximation with optimal modal coefficients (denoted as true POD) cannot reconstruct the high-fidelity flow fields if we just consider one global interval. So, it is basically a representability problem, rather than a closure one. Meanwhile, closure/stabilization techniques can supplement the proposed framework based on PID for even further stabilized or improved reduced order approximations. In our context, closure modeling can be referred to retaining R modes in the model and incorporating the effects of the remaining modes into the model dynamics. In this study, we generate ROMs without considering any additional stabilization or closure modeling approach. Instead, we include the true projection results, which can be considered an ultimate limit of the projection ROM with an optimal closure. This pertains to an analogy that can be made with optimal large eddy simulation methods.[150]

In other words, for example, for a fixed number of snapshots, we can consider two representation approaches. First, in a closure approach, we might compute first 6 POD modes from all snapshots and build a global model for using these modes and incorporate the effects of the remaining modes in the $R = 6$ model. If the closure model is perfect (i.e., ideal ROM), we will be able to obtain the true projection results. Alternatively, in the second approach, we might compute the first 6 POD modes from the first half of the snapshots and then another 6 POD modes from the remaining half and generate a locally partitioned $R = 6$ model by using only 6 modes for each partition. In our results, we illustrate that the second approach with

TABLE II. A list of hyperparameters utilized to train the LSTM network for all numerical experiments.

Variables	Burgers	Vortex merger	Double shear layer	Boussinesq
Number of hidden layers	3	3	4	3
Number of neurons in each hidden layer	80	80	80	80
Batch size	64	64	64	64
Epochs	100	1000	1000	100
Activation functions in the LSTM layers	tan h	tan h	tan h	tan h
Validation data set (%)	20	20	20	20
Loss function	MSE	MSE	MSE	MSE
Optimizer	ADAM	ADAM	ADAM	ADAM
Learning rate	0.001	0.001	0.001	0.001
First moment decay rate	0.9	0.9	0.9	0.9
Second moment decay rate	0.999	0.999	0.999	0.999

6 modes often yields significantly more accurate results than the first approach even if we use the perfect closure (i.e., when we model the effects of all the remaining modes into the low-dimensional model considering the first 6 modes). Furthermore, we demonstrate that the representation capability of a ROM model with 2 partitions (each having 6 modes) might be even much superior to a ROM of higher degrees of freedom (e.g., with 12 modes) with one partition in solving convective flow problems. This makes our PID-LSTM approach very attractive not only for the representation quality but also for the computational performance since a ROM with 12 modes requires 8 times more computational overhead than a ROM with 6 modes (e.g., computational run time cost of the Galerkin ROM scales with R^3 for systems with quadratic nonlinearity). It might be also interpreted that the proposed PID-LSTM framework converts the linear storage cost of saving basis functions into the cubic rate of computational speed-up of the model. Therefore, we consider the PID framework as an effective partitioning method to break the Kolmogorov barrier.

VII. RESULTS

For all test cases, we apply both the intrusive and nonintrusive frameworks, discussed in Secs. V and VI, respectively. A total number of 800 snapshots, or strobes, are collected from FOM simulations. A summary of the data generation characteristics as well as CPU time for constructing basis functions, implemented in FORTRAN, is given in Table III. As mentioned in Sec. IV, the PID algorithm reduces the computational cost for basis generation. It is evident that the CPU time for basis construction decreases with increasing N_p. The size of the matrix $\mathbf{A}$ in PID is $\left(\frac{N_s}{N_p} \times \frac{N_s}{N_p}\right)$, so increasing the number of intervals decreases the size of this matrix and makes the solution of the corresponding eigenvalue problem faster.

For numerical computations, we use a family of fourth order compact schemes for spatial derivatives[151] and a third order Runge-Kutta scheme for temporal integration.[152] The time domain is decomposed into 2, 4, 8, and 16 subintervals. The CPU times of PID-LSTM training and testing stages (in Python) are summarized in Table IV for different number of intervals. It should be noted that the most expensive stages of the PID-LSTM approach are performed offline, where the online prediction is relatively fast. Although the PID increases the computational overhead for online prediction (almost linearly with the number of intervals), speedups of several orders of magnitudes compared to FOM are still accomplished. Prediction and reconstruction accuracies are computed and compared with the case without interval decomposition (i.e., a single global interval) in order to illustrate the effects of nonergodicity and assess the PID contribution to mitigate those effects. Also, the predictive performance of the LSTM-based approach is compared with the GP-based one to demonstrate its capability to bypass the Galerkin projection step and provide accurate predictions without prior information of the underlying complex physical system.

TABLE III. A summary of data generation characteristics for the full order models and their required CPU times (in seconds). We document the corresponding speedup reached by the PID-GP model. We note that the CPU time assessments documented in this table are based on FORTRAN executions.

Variables	Burgers	Vortex merger	Double shear layer	Boussinesq
FOM relevant parameters				
Grid resolution	1024	1024 × 1024	1024 × 1024	4096 × 512
Time step, Δt	1.00×10^{-4}	1.00×10^{-3}	1.00×10^{-3}	5.00×10^{-4}
Maximum simulation time, $t_{\max}$	2.00	40.00	40.00	8.00
CPU time required for FOM simulation	3.43	1.20×10^5	1.20×10^5	7.38×10^4
Offline data preparation				
Number of snapshots collected, N_s	800	800	800	800
CPU time for generating POD bases ($N_p = 1$)	7.21×10^1	3.31×10^4	3.27×10^4	1.18×10^5
CPU time for generating POD bases ($N_p = 2$)	7.70	8.15×10^3	8.17×10^3	2.93×10^4
CPU time for generating POD bases ($N_p = 4$)	9.33×10^{-1}	2.15×10^3	2.15×10^3	7.64×10^3
CPU time for generating POD bases ($N_p = 8$)	1.53×10^{-1}	5.70×10^2	5.70×10^2	1.96×10^3
CPU time for generating POD bases ($N_p = 16$)	2.98×10^{-2}	1.22×10^2	1.25×10^2	4.55×10^2
PID-GP online phase				
Number of modes retained, R	6	6	6	6
CPU time for PID-GP [speedup] ($N_p = 1$)	0.14 [25]	0.03 [4×10^6]	0.03 [4×10^6]	0.06 [1×10^6]
CPU time for PID-GP [speedup] ($N_p = 2$)	0.13 [26]	0.13 [9×10^5]	0.12 [1×10^6]	0.43 [2×10^5]
CPU time for PID-GP [speedup] ($N_p = 4$)	0.11 [31]	0.30 [4×10^5]	0.32 [4×10^5]	1.16 [6×10^4]
CPU time for PID-GP [speedup] ($N_p = 8$)	0.07 [49]	0.63 [2×10^5]	0.64 [2×10^5]	2.61 [3×10^4]
CPU time for PID-GP [speedup] ($N_p = 16$)	0.11 [31]	1.27 [9×10^4]	1.27 [9×10^4]	4.97 [1×10^4]

TABLE IV. CPU time (in seconds) of PID-LSTM training (offline) and testing (online) stages. We note that the CPU time assessments documented in this table are based on Python executions.

N_p	Training time	Testing time
1D Burgers		
1	17.62	2.04
2	30.18	1.98
4	61.88	3.78
8	132.70	11.28
16	304.40	41.09
2D vortex merger		
1	125.51	2.03
2	248.57	4.52
4	533.28	11.00
8	1399.11	35.80
16	2379.21	79.74
2D double shear layer		
1	161.76	2.40
2	333.11	5.51
4	809.51	15.17
8	1435.85	31.36
16	3655.18	94.85
2D Boussinesq		
1	17.36	2.61
2	36.01	7.55
4	82.86	20.55
8	238.22	57.90
16	347.09	116.52

A. 1D Burgers problem

For 1D Burgers simulation, we consider the following initial condition:[153]

$$u(x,0) = \frac{x}{1+\exp\left(\frac{\text{Re}}{16}(4x^2-1)\right)}, \tag{46}$$

with $x \in [0, 1]$. Also, we assume Dirichlet boundary conditions, where $u(0, t) = u(1, t) = 0$. We solve the problem for the Reynolds number, Re = 1000, using 1024 grid spacings in the x-direction and a timestep of 10^{-4} for $t \in [0, 2]$. We would like to mention here that the 1D Burgers equation with the above initial and boundary conditions has an analytic solution,[153] but we prefer to solve it numerically for consistent comparison with ROMs. The solution $u(x, t)$ represents a traveling wave along a flat horizontal bottom, shown at different times in Fig. 3.

In order to demonstrate the benefits of the PID approach constructing localized basis functions, we provide the true projection of the final field (at $t = 2$) and compared to the FOM solution in Fig. 4. We can easily observe that using only a single global interval (i.e., $N_p = 1$) gives inaccurate solution with oscillations that do not exist at any instance of the flow. This is due to the deformation and smoothing-out of the global modes by the rapidly evolving flow. This also supports the claim that the accuracy gain due to closure techniques in the investigated convection-dominated flows would be minimum as the true projection fields represent the maximum attainable accuracy with closure. The procedures of intrusive and nonintrusive ROMs are applied to evaluate the temporal coefficients $\alpha(t)$. Results are shown in Figs. 5 and 6 with a zoomed-in view in Fig. 7. Similar observations can be obtained regarding the effects of the number of intervals on the accuracy of the ROM approximation. However, it can be easily seen that the oscillations are

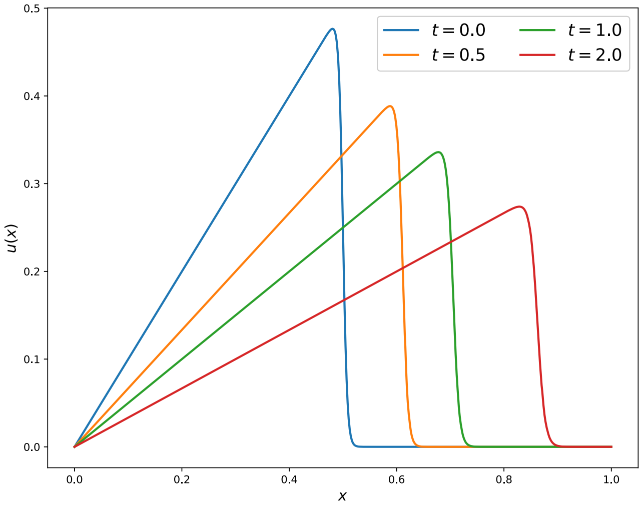

FIG. 3. Velocity field at different time instances for the Burgers equation for Re = 1000 using a 1024 grid and $\Delta t = 0.0001$.

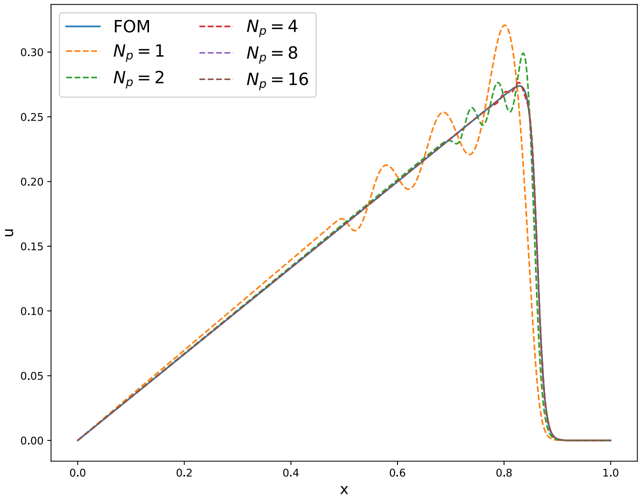

FIG. 4. Final velocity field (i.e., at $t = 2$) for the Burgers problem from true projection using different number of intervals compared to FOM solution.

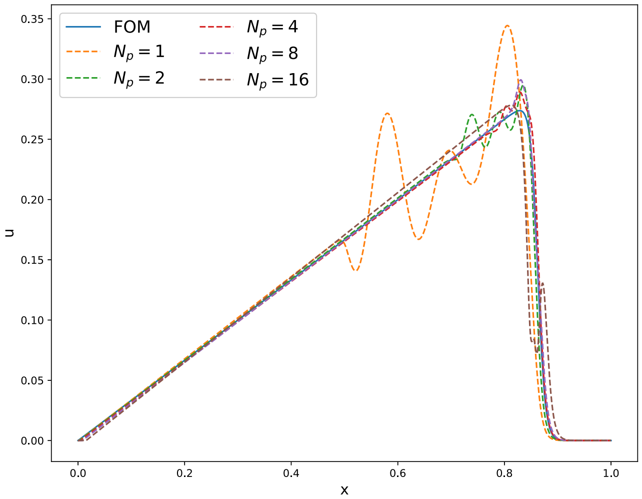

FIG. 5. Final velocity field (i.e., at t = 2) for the Burgers problem from PID-GP prediction using different number of intervals compared to FOM solution.

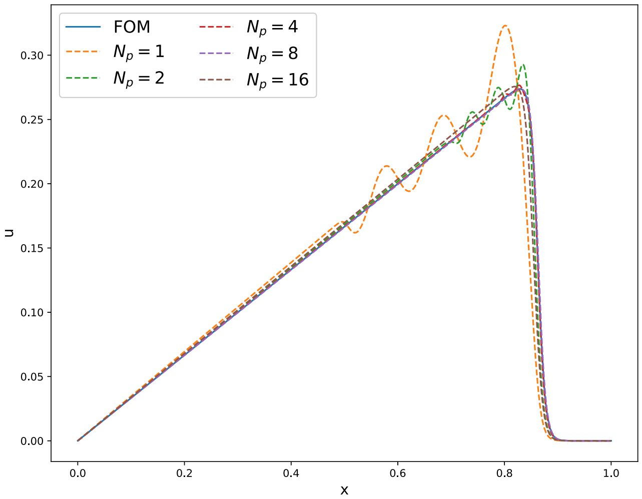

FIG. 6. Final velocity field (i.e., at t = 2) for the Burgers problem from PID-LSTM prediction using different number of intervals compared to FOM solution.

amplified in PID-GP results compared to PID-LSTM results. This is mainly due to the nonlinear interactions in the governing equation [i.e., Eq. (1)], where the lower modes interact strongly with the higher modes. Truncation of the lower modes simply ignores these interactions in the projection-based ROM, given as Eq. (27), and results in a deviation of the ROM dynamics from the true one. Hence, the nonintrusive character of the proposed framework mitigates these instability issues.

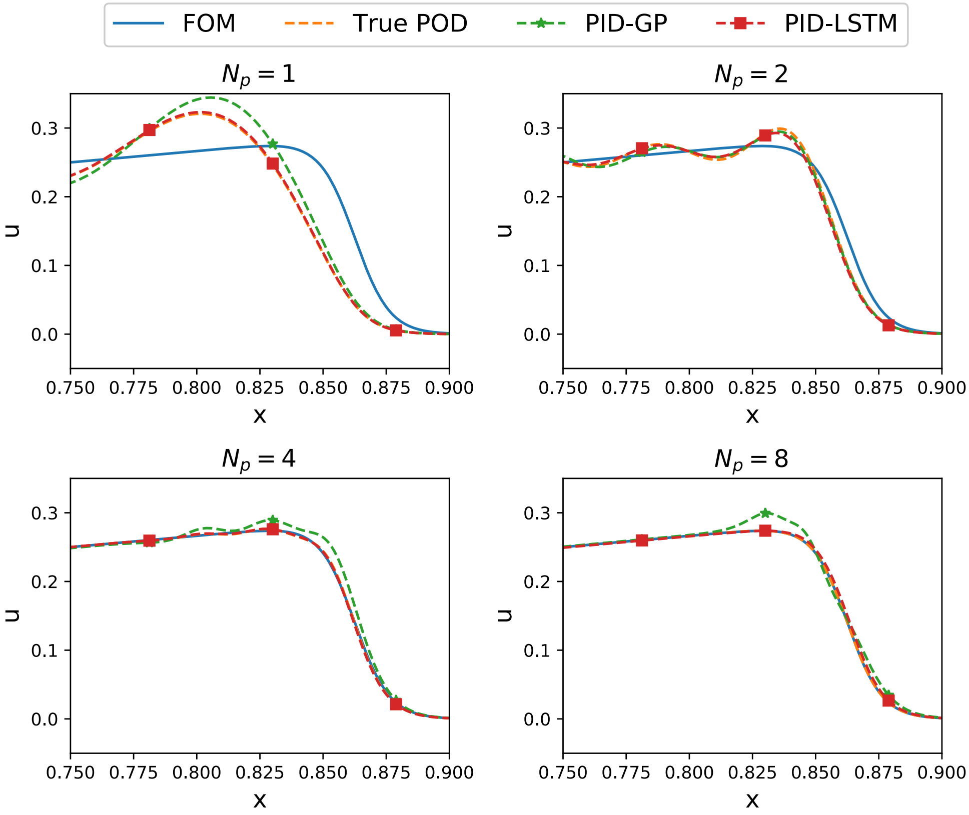

FIG. 7. Comparison between the final velocity fields for the Burgers problem obtained from different approaches using different number of intervals.

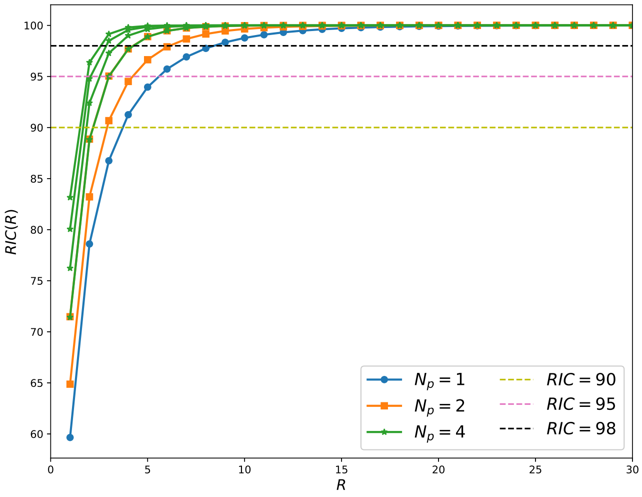

FIG. 8. Relative information content (*RIC*) for the Burgers problem, using one, two, and four intervals.

A simple eigenvalue analysis of the Burgers problem can help in demonstrating the idea behind interval decomposition. In POD, the percentage modal energy is computed using the following relative information content (*RIC*) formula:[154]

$$RIC(R) = \left(\frac{\sum_{j=1}^{R} \lambda_j}{\sum_{j=1}^{N_s} \lambda_j} \right) \times 100. \tag{47}$$

The *RIC* plot is shown in Fig. 8 for different number of intervals. One can easily observe that this interval decomposition produces local POD modes with more concentrated energy content, compared to a single interval giving global modes with more distributed energies. For example, if we are interested in capturing 98% of the total energy (snapshot variance), we will need at least 8 POD modes in the case of using a single wider interval. On the other hand, if we decompose our interval into two partitions, we will need 6 modes, and if we decompose into 4 subintervals, 4 modes will be more than enough. Although this might imply more memory requirements, significant computational gains can be obtained. For instance, if we follow the classical POD-GP approach, the computational cost is $O(R^3)$. Therefore, using 4 modes instead of 8 modes would be around 8 times faster.

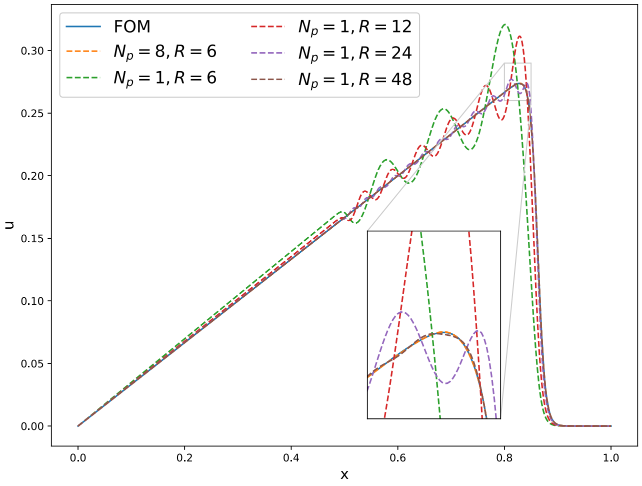

FIG. 9. Comparison of true projection of the final velocity field onto global POD modes (i.e., $N_p = 1$) with different number of modes vs local modes with $N_p = 8$ and $R = 6$.

To demonstrate the benefit of the PID framework, Fig. 9 illustrates a comparison of ideal/optimal predictive performance of a ROM (with $N_p = 8$ and $R = 6$) against a single partitioned global ROM with higher degrees of freedom ($R = 6$, 12, 24, and 48). It can be seen easily from these true projection results that a ROM built by only $R = 6$ modes using the PID framework yields more accurate results than a standard projection ROM with $R = 48$ modes. We also highlight that the PID ROM ($N_p = 8$ and $R = 6$) will be approximately 8^3 times faster than the global ROM ($N_p = 1$ and $R = 48$) when the Galerkin projection model is used.

B. 2D vortex merger problem

We expand our framework testing into two-dimensional cases. As an application for 2D Navier-Stokes equations, we consider the vortex merger problem (i.e., the merging of a corotating vortex pair).[155] The merging process occurs when two vortices of the same sign with parallel axes are within a certain critical distance from each other, ending as a single, nearly axisymmetric, final vortex.[156]

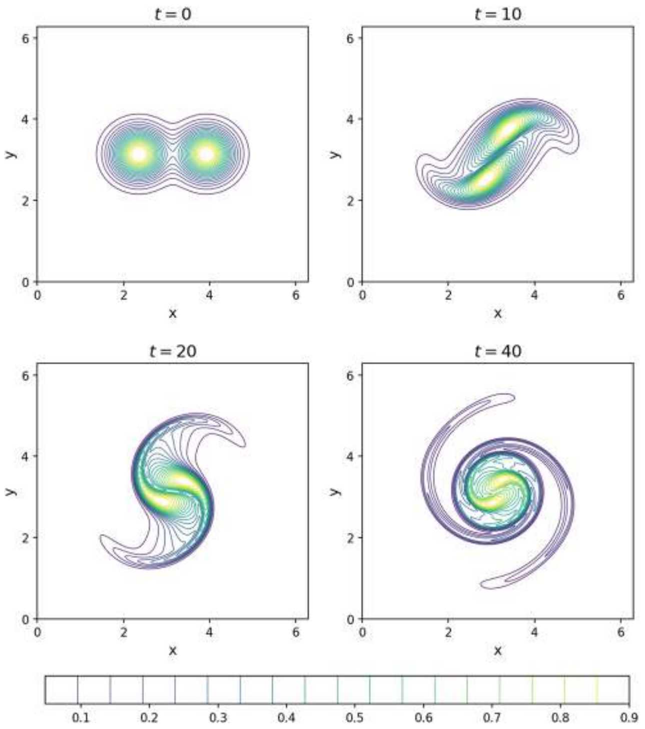

FIG. 10. Vorticity field at different time instances for the vortex merger problem using a 1024^2 grid and $\Delta t = 0.001$.

It is a two-dimensional process and is one of the fundamental processes of fluid motion and occurs in many fields such as astrophysics, meteorology, and geophysics. For example, in two-dimensional turbulence, like-sign vortex merger is the main factor affecting the evolution of the vortex population.[156] Vortex merging also plays an important role in the context of aircraft trailing wakes.[157] We consider an initial vorticity field of two Gaussian-distributed vortices with a unit circulation as follows:

$$\omega(x,y,0) = \exp\left(-\rho\left[(x-x_1)^2+(y-y_1)^2\right]\right) + \exp\left(-\rho\left[(x-x_2)^2+(y-y_2)^2\right]\right), \tag{48}$$

where ρ is the interacting constant set as $\rho = \pi$ and the vortices centers are initially located at $(x_1, y_1) = \left(\frac{3\pi}{4}, \pi\right)$ and $(x_2, y_2) = \left(\frac{5\pi}{4}, \pi\right)$. We use a Cartesian domain $(x, y) \in [0, 2\pi] \times [0, 2\pi]$, with periodic boundary conditions. We perform our simulations solving Eq. (4) with a Reynolds number of 10 000 using a 1024^2 spatial grid and a timestep of 0.001. The evolution of the two vortices from time $t = 0$ to $t = 40$ is shown in Fig. 10. Details of the numerical schemes and computations can be found in a previous study.[158]

We compare the final field, characterizing the merging of two vortices into a single vortex at the center of the 2D domain. Figure 11 shows true projection, while Figs. 12 and 13 illustrate the results obtained using the PID-GP framework and PID-LSTM framework, respectively. Similar to the 1D Burgers results, we note that increasing the number of intervals improves the results significantly. For example, if we look at the obtained field using a single interval, we will find a vorticity field that does not resemble the true one except for a very poor approximation of the merging phenomenon. However, the external field (away from the core) is very different from the

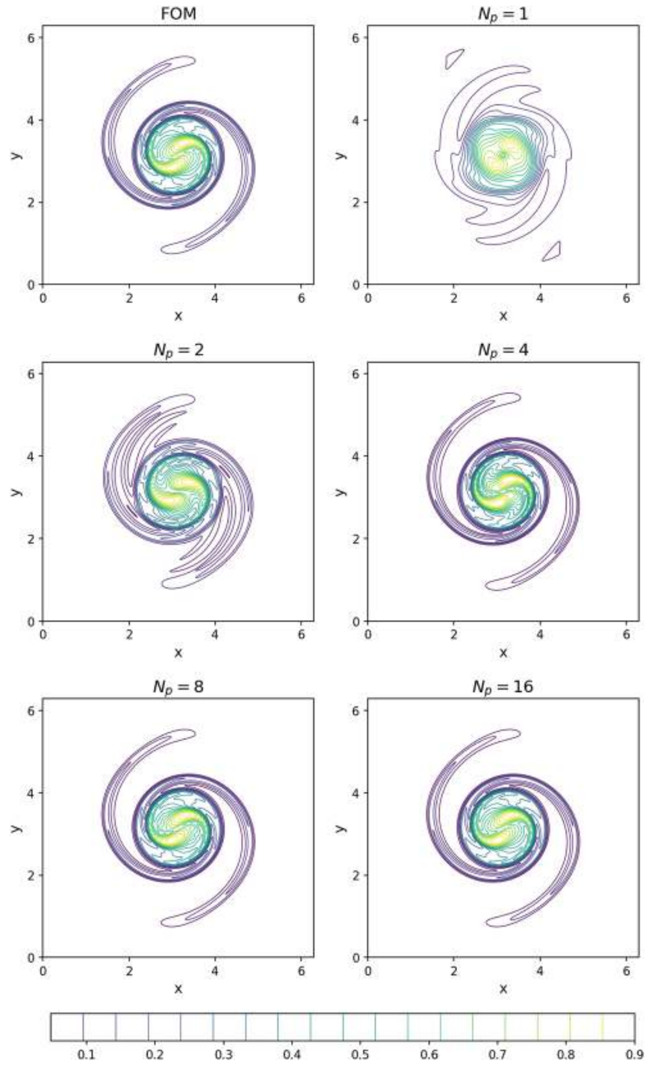

FIG. 11. Final vorticity contours (i.e., at t = 40) for the vortex merger problem from true projection using different number of intervals compared to FOM solution.

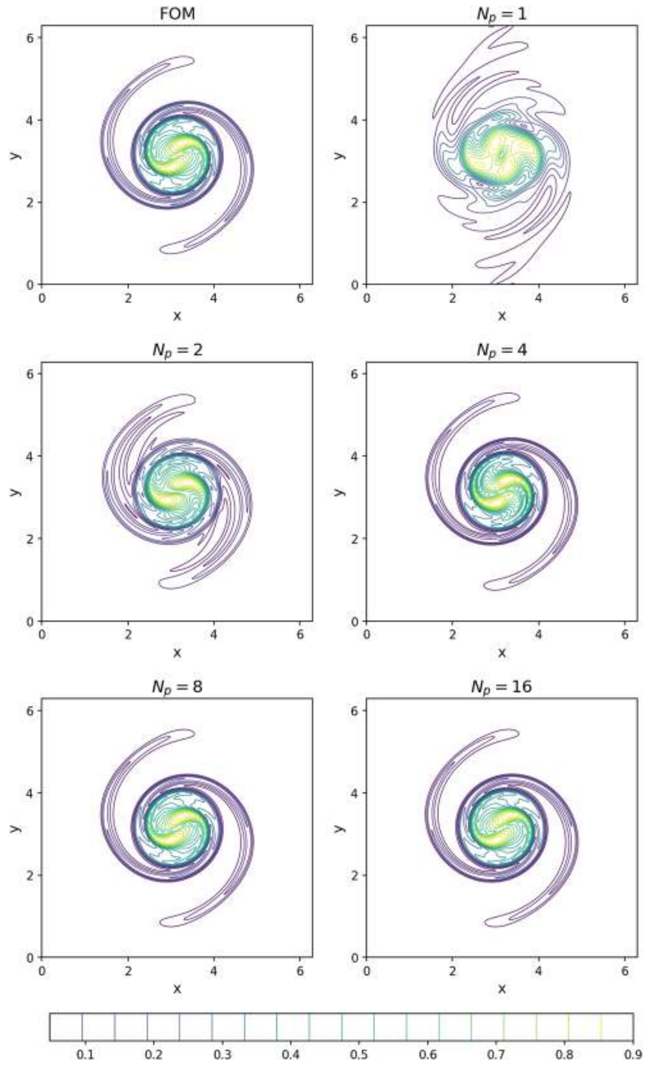

FIG. 12. Final vorticity contours (i.e., at t = 40) for the vortex merger problem from PID-GP prediction using different number of intervals compared to FOM solution.

true physical spiral motion. Moreover, Fig. 12 shows larger deformation in the field compared to both the true projection and PID-LSTM framework. Again, this is due to the nonlinear interactions affecting the truncated ROM equations, requiring stabilization schemes to mitigate these effects.

C. 2D double shear layer problem

Another application for 2D Navier-Stokes equations is the double shear layer problem, introduced by Bell *et al.*[159] We consider a square domain of side length 2π with the following initial field:[160]

$$\omega(x,y,0)=\begin{cases}\delta\cos(x)-\sigma\cosh^{-2}(\sigma[y-\dfrac{\pi}{2}]) & \text{if } y\le\pi,\\ \delta\cos(x)+\sigma\cosh^{-2}(\sigma[\dfrac{3\pi}{2}-y]) & \text{if } y>\pi.\end{cases} \quad (49)$$

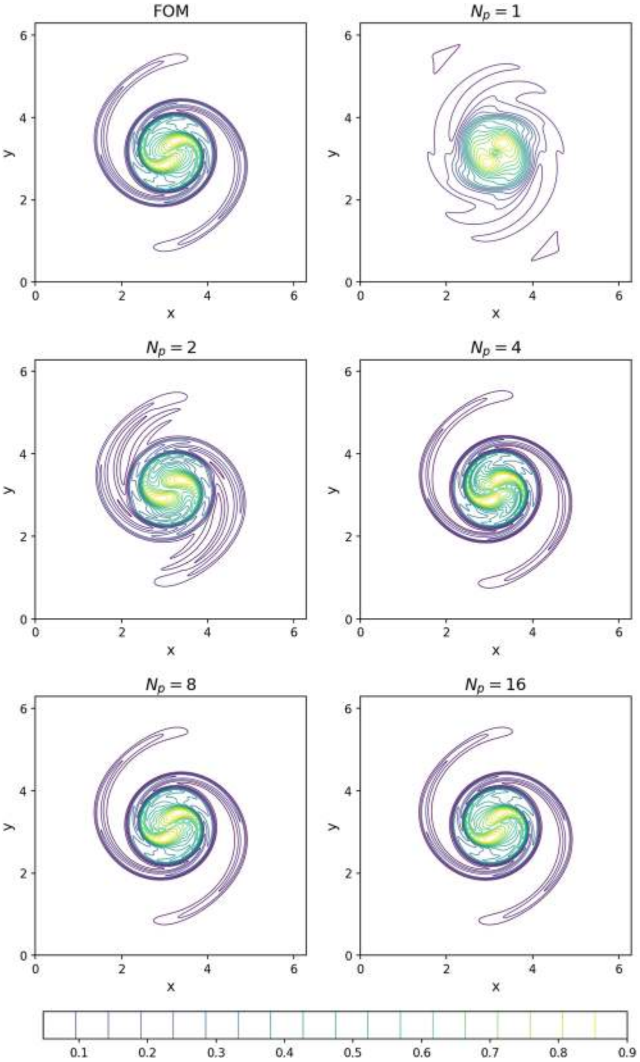

FIG. 13. Final vorticity contours (i.e., at $t = 40$) for the vortex merger problem from PID-LSTM prediction using different number of intervals compared to FOM solution.

This field represents a horizontal shear layer of finite thickness (determined by δ), perturbed by a small amplitude vertical velocity, where σ determines the amplitude of this initial perturbation. In this study, we adopt values of $\delta = 0.05$ and $\sigma = 15/\pi$. Similar to the vortex merger setup, we use Re = 10 000 over a grid of 1024^2 and $\Delta t = 0.001$. Indeed, the same numerical solver is used for both the vortex merger and double shear layer problem, with only different initial conditions. The evolution of the double shear layer from time $t = 0$ to $t = 40$ is shown in Fig. 14, where the top and bottom shear layers evolve into a periodic array of large vortices, and the layer between the rolls becomes thinner and thinner.

Similar results are obtained for the double shear problem, as shown in Figs. 15–17. As can be seen in Fig. 15, significant details of the shear layers cannot be captured using a single global interval ($N_p = 1$), even from the direct projection of the FOM field on the ROM space. The situation is even worse in PID-GP, where the vortex at the center of the domain is deformed. Interestingly, the PID-LSTM performs much better than PID-GP, almost similar to the true projection fields.

D. 2D Boussinesq problem

The two-dimensional Boussinesq problem is one-step more complex than the 2D Navier-Stokes equations, solving the energy equation along with the momentum equations. We consider a strong-shear flow exhibiting the Kelvin-Helmholtz instability,

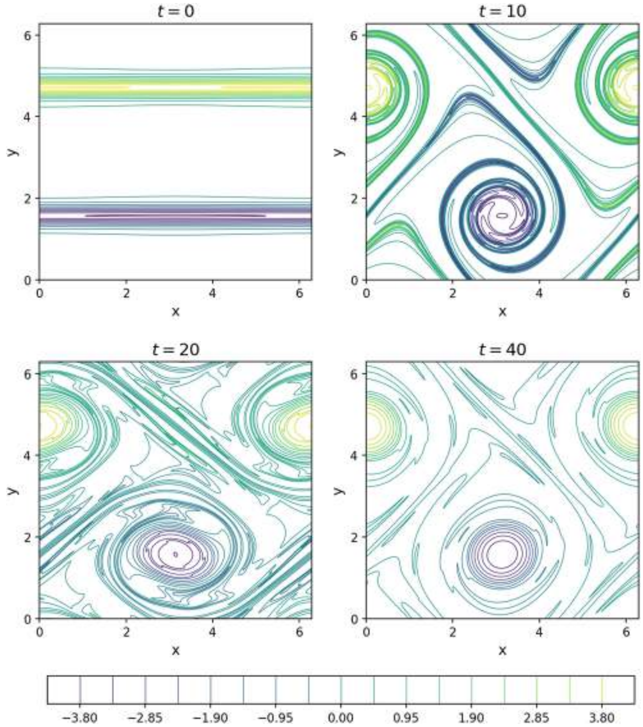

FIG. 14. Vorticity field at different time instances for the double shear layer problem using a 1024^2 grid and $\Delta t = 0.001$.

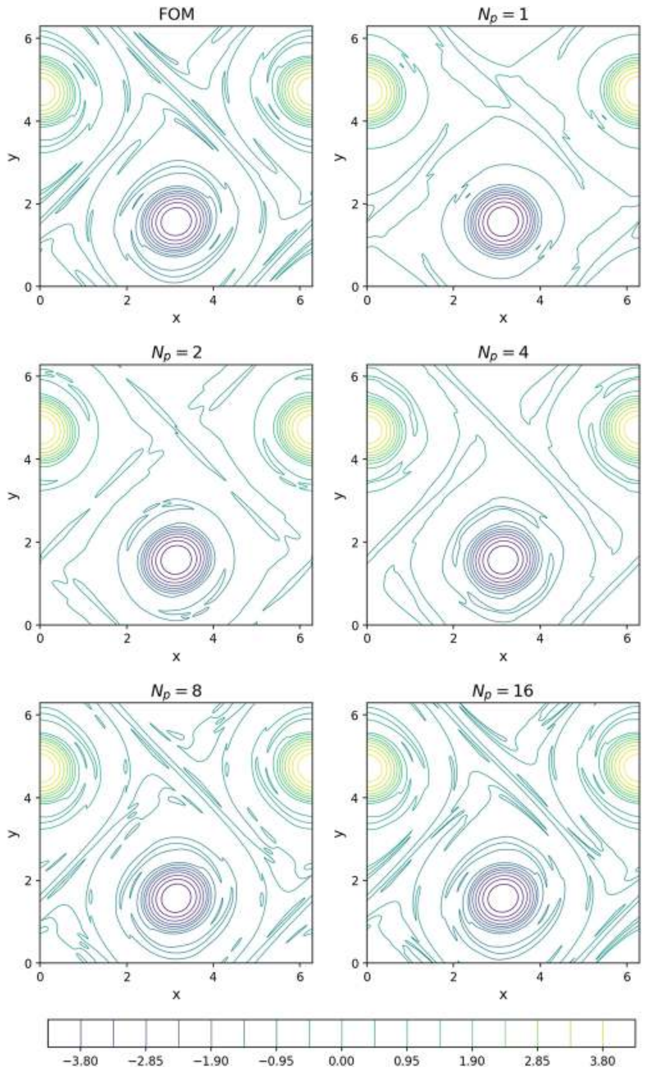

FIG. 15. Final vorticity contours (i.e., at t = 40) for the double shear layer problem from true projection using different number of intervals compared to FOM solution.

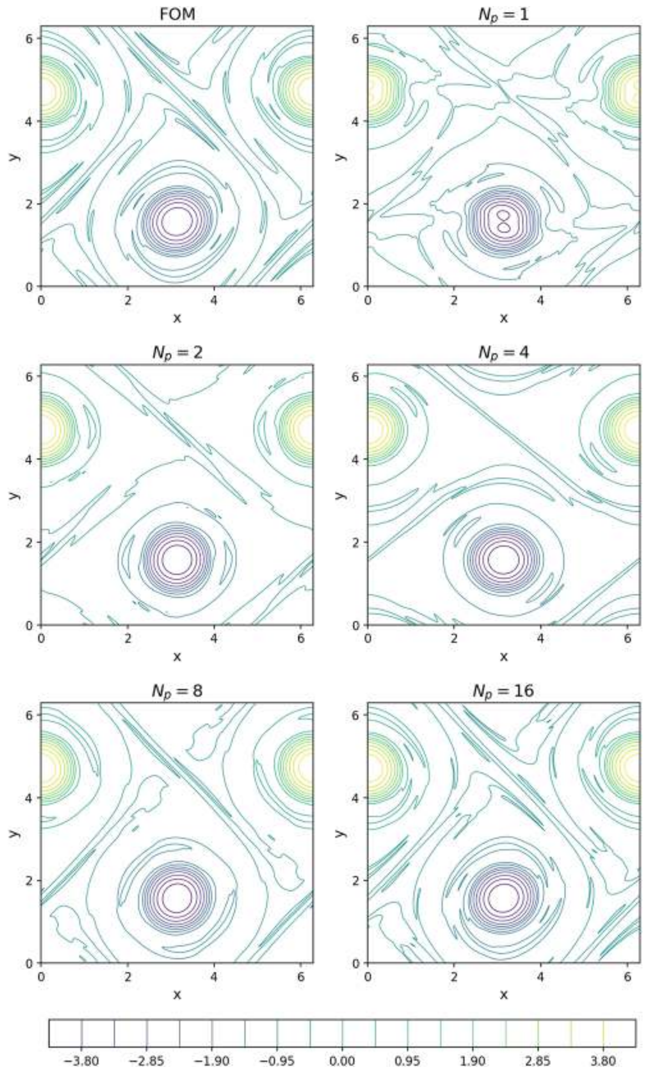

FIG. 16. Final vorticity contours (i.e., at t = 40) for the double shear layer problem from PID-GP prediction using different number of intervals compared to FOM solution.

known as the Marsigli flow or lock-exchange problem. The physical process in this flow problem explains how differences in temperature/density can cause currents to form in the ocean, seas and natural straits. For example, Marsigli discovered that the Bosporus currents are a consequence of the different water densities in the Black and Mediterranean seas.[161] Basically, when fluids of two different densities meet, the higher density fluid slides below the lower density one. This is one of the primary mechanisms by which ocean currents are formed.[162]

We consider two fluids of different temperatures, in a rectangular domain $(x, y) \in [0, 8] \times [0, 1]$. A vertical barrier divides the domain at $x = 4$, keeping the temperature, θ, of the left half at 1.5 and the temperature of the right half at 1. Initially, the flow is at rest [i.e., $\omega(x, y, 0) = \psi(x, y, 0) = 0$], with uniform temperatures at the right and left regions [i.e., $\theta(x, y, 0) = 1.5 \ \forall \ x \in [0, 4]$ and $\theta(x, y, 0) = 1 \ \forall \ x \in (4, 8]$]. No-slip boundary conditions are assumed for the flow field, and adiabatic boundary conditions are prescribed for the temperature field. The Reynolds number of Re = 10^4, the Richardson number of Ri = 4, and the Prandtl number of Pr = 1 are set in Eqs. (11) and (12). A Cartesian grid of 4096 × 512 and a timestep of $\Delta t = 5 \times 10^{-4}$ are used for the FOM simulations. The evolution of the temperature field is shown in Fig. 18 at t = 0, 2, 4, 8. At time zero, the barrier is removed, instantaneously triggering the lock-exchange problem. Due to the temperature difference (causing density difference), buoyancy forces start to emerge. The higher density fluid (on the right) slides below the lower density fluid (on the left), causing an undercurrent flow moving from right to left. Conversely, an upper current flow moves from left to right, causing a

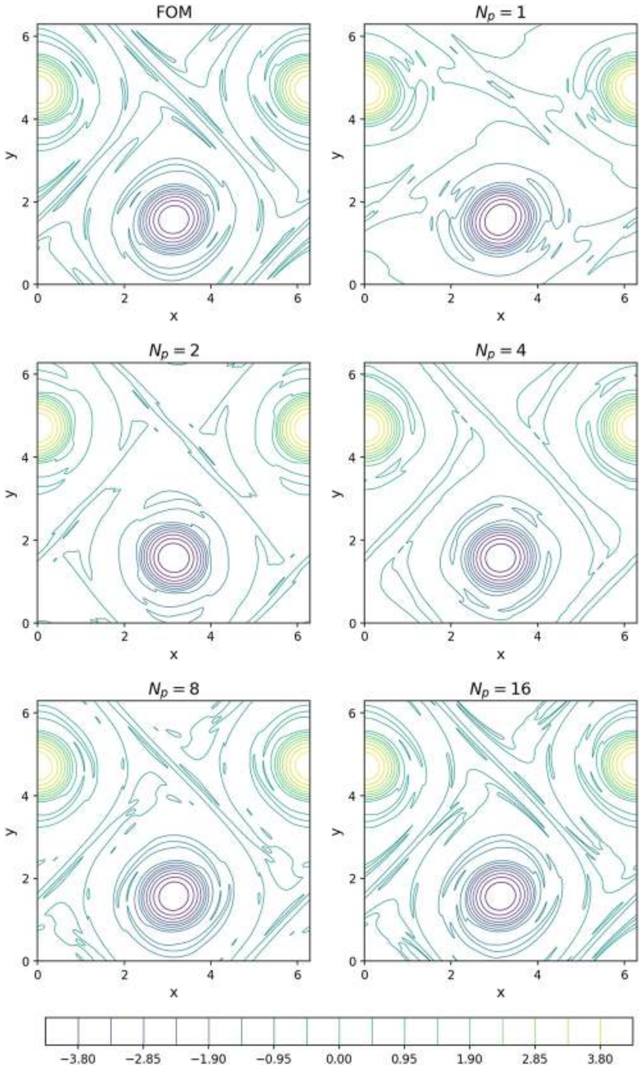

FIG. 17. Final vorticity contours (i.e., at t = 40) for the double shear layer problem from PID-LSTM prediction using different number of intervals compared to FOM solution.

strong shear layer between the countercurrent flows. As a result, vortex sheets are produced, exhibiting the Kelvin-Helmholtz instability. This problem is challenging, even for direct numerical simulation (DNS) calculations,[122] making it a good benchmark for POD/PID comparison.

In Fig. 19, we show the true projection of the final temperature field on the POD/PID space using different number of intervals. Although the overall structure is represented nicely using a single interval (corresponding to the standard POD), the small-scale structures are not captured. If we investigate the contour lines carefully, we can see that the standard POD smoothens the field. As the number of intervals is increased, more details can be captured using local basis functions. Also, we compare the final temperature field predictions from the standard Galerkin projection and LSTM frameworks.

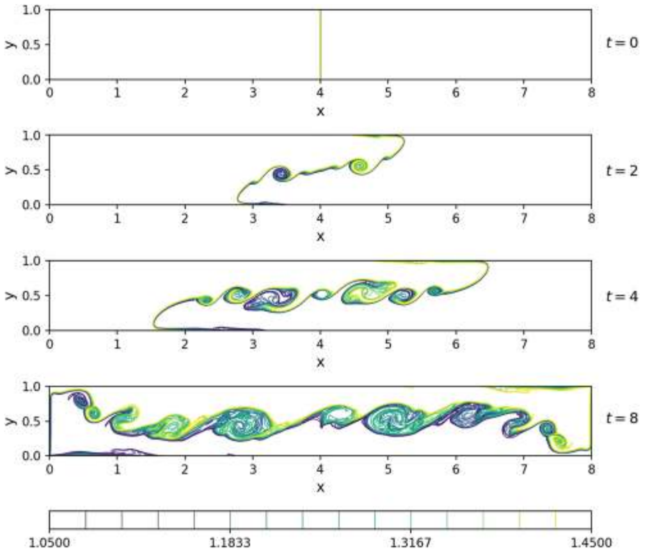

FIG. 18. Temperature field at different time instances for the 2D Boussinesq problem using a 4096 × 512 grid and Δt = 0.0005.

Contour plots for the final temperature field are shown in Figs. 20 and 21 at different number of intervals, N_p. Similar to previous cases, PID-GP is adding more deformation to the results, and instabilities are amplified. This is due to the fact that the eigenvalues of

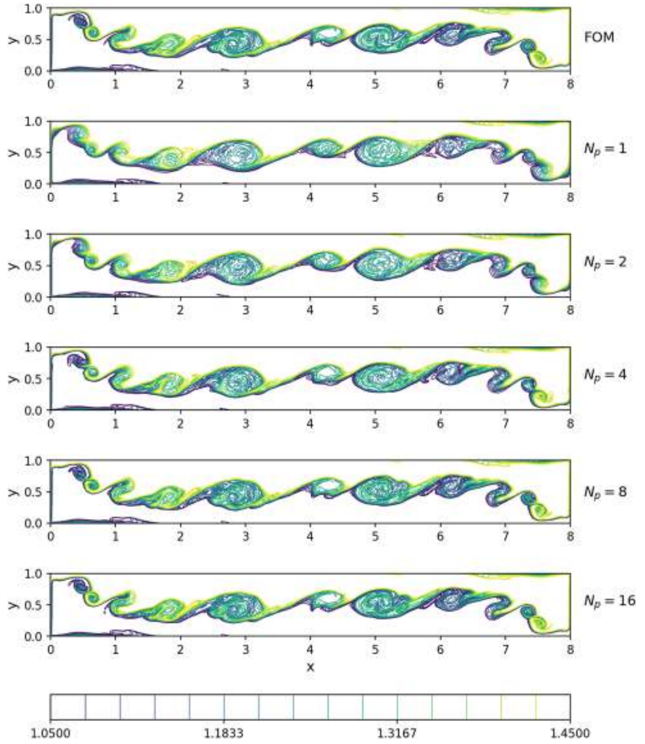

FIG. 19. Final contours (i.e., at t = 8) for the Boussinesq problem from true projection using different number of intervals compared to FOM solution.

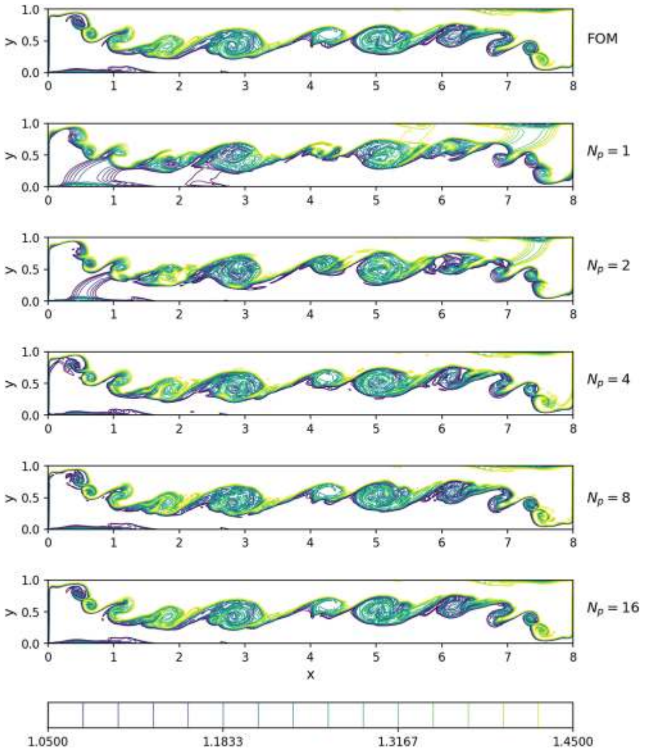

FIG. 20. Final contours (i.e., at t = 8) for the Boussinesq problem from PID-GP prediction using different number of intervals compared to FOM solution.

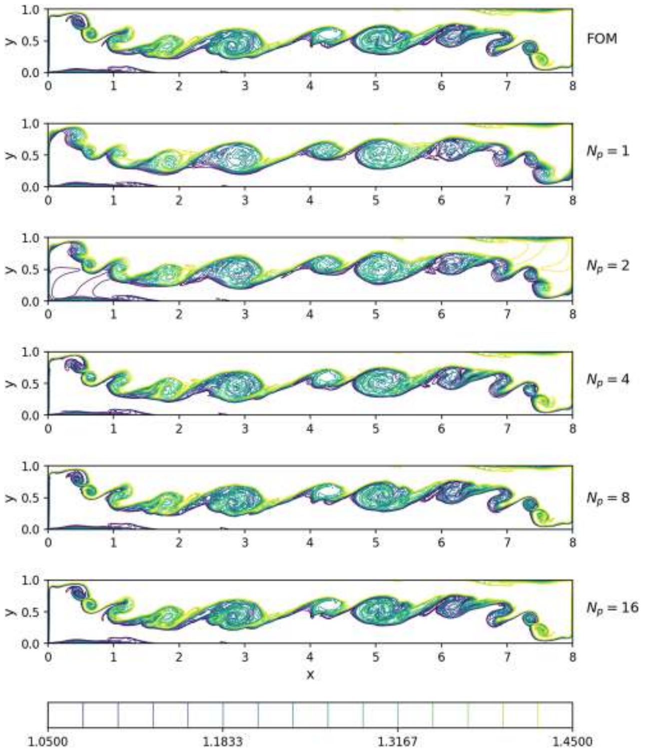

FIG. 21. Final contours (i.e., at t = 8) for the Boussinesq problem from PID-LSTM prediction using different number of intervals compared to FOM solution.

this flow problem are decaying slowly, especially for such a high Re used in the current study. In this study, we use just 6 modes (i.e., R = 6), corresponding to the *RIC* of only 60.97% for vorticity and 88.27 for temperature fields using one interval. Due to the dependence of Galerkin projection on the governing equations, the resulting ROMs strongly couple temperature and vorticity. Therefore, inaccuracies in vorticity predictions often affect temperature predictions and vice versa since they are coupled with each others in the ROM-GP equations. As a result, even if 6 modes with N_p = 4 can capture more than 90% of the variance in the temperature field, the results will be affected by the low energy captured from the vorticity fields (less than 70%), and instabilities in the predicted fields can easily occur. On the other hand, LSTM predictions do not have this dependency, which enables us to use the most accurate datasets and fields with the maximum reducibility without being affected by other irrelevant fields. This is another merit of nonintrusive ROMs for reducing instabilities with multivariate flow fields.

We note here that our results prove that the PID approach is an appropriate technique for model order reduction of convection-dominated flows. It is clear from the four test cases that these systems undergo significant evolution over time as the convective effects are more dominant than the dissipative ones. Due to variation of the system's states at different time instants, the system becomes less reducible and the standard approaches fail to extract any meaningful underlying structures. The resulting modes are largely smoothed out and deformed to give global approximation of the varying flow field. As a result, the energy is distributed over more modes rather than concentrated in a few ones. Alternatively, the PID approach aims to split the time windows over which the standard techniques (e.g., POD) are applied. This splitting helps in clustering different flow regimes/behaviors in different partitions, so that the snapshots in each partition almost represent a single regime making order reduction feasible. The Appendix provides some visualizations of the basis functions at different number of intervals. They clearly show the deficiency of global application of POD to extract the underlying patterns.

Finally, in order to quantify the results in a more quantitative way, we use the root mean square error (*RMSE*) as an error measure, defined as

$$RMSE(t) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(u^{FOM}(\mathbf{x},t) - u(\mathbf{x},t)\right)^2}, \qquad (50)$$

where N is the spatial resolution, as defined in Sec. III. For 1D cases, it is simply N_x, and in 2D cases, it is $(N_x \times N_y)$. We calculate the RMSE at the final field (i.e., at t = T) using different approaches. Results are given in Table V and illustrated graphically using a bar chart in Fig. 22, confirming our earlier findings about the accuracy gain due to time decomposition as well as PID-LSTM being superior to the PID-GP framework.

TABLE V. *RMSE* for the predicted field at final time from the true projection, PID-GP framework, and PID-LSTM framework compared to FOM results.

N_p	True projection	PID-GP	PID-LSTM
1D Burgers			
1	2.31×10^{-2}	2.82×10^{-2}	2.32×10^{-2}
2	6.62×10^{-3}	7.02×10^{-3}	6.88×10^{-3}
4	7.24×10^{-4}	4.50×10^{-3}	7.93×10^{-4}
8	2.16×10^{-5}	4.23×10^{-3}	1.39×10^{-3}
16	2.19×10^{-7}	2.18×10^{-2}	1.03×10^{-2}
2D vortex merger			
1	4.18×10^{-2}	6.27×10^{-2}	4.20×10^{-2}
2	2.74×10^{-2}	2.79×10^{-2}	2.75×10^{-2}
4	9.08×10^{-3}	9.57×10^{-3}	9.14×10^{-3}
8	1.09×10^{-3}	2.35×10^{-3}	1.61×10^{-3}
16	4.80×10^{-5}	1.65×10^{-4}	3.29×10^{-3}
2D double shear layer			
1	1.37×10^{-1}	1.87×10^{-1}	1.78×10^{-1}
2	1.16×10^{-1}	1.51×10^{-1}	1.68×10^{-1}
4	1.03×10^{-1}	1.79×10^{-1}	1.06×10^{-1}
8	6.63×10^{-2}	8.69×10^{-2}	6.83×10^{-2}
16	2.56×10^{-2}	2.97×10^{-2}	2.67×10^{-2}
2D Boussinesq			
1	6.30×10^{-2}	1.03×10^{-1}	6.30×10^{-2}
2	5.28×10^{-2}	7.28×10^{-2}	5.45×10^{-2}
4	3.73×10^{-2}	4.11×10^{-2}	3.73×10^{-2}
8	2.03×10^{-2}	2.22×10^{-2}	2.03×10^{-2}
16	8.24×10^{-3}	8.94×10^{-3}	8.67×10^{-3}

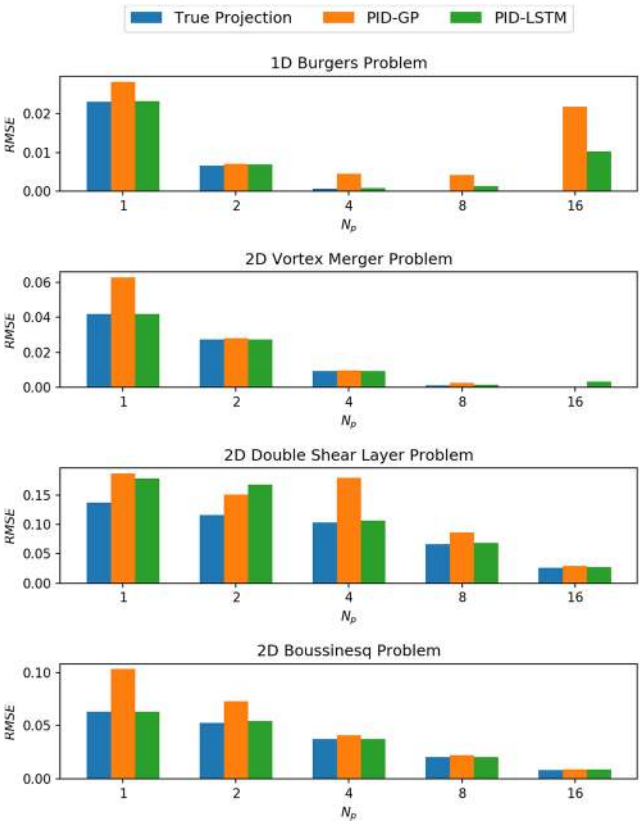

FIG. 22. A bar chart for *RMSE* of the predicted field at final time.

An important observation from Table V and Fig. 22 is the significant increase in *RMSE* for the Burgers problem at $N_p = 16$. To express this behavior, we plot the eigenspectrum for these four problems as shown in Fig. 23. Interestingly, we can see that the decay of eigenvalues for the 1D Burgers case is faster than other cases, while the decay of eigenvalues in the 2D Boussinesq problem is the slowest. This implies that the higher the decay rate is, the fewer the required intervals are. That is, for the 1D Burgers case, 4 or 8 subintervals are enough to capture local dynamics, and a further increase in the number of partitions leads to an increase in *RMSE*. This might be caused by the degradation of field quality due to successive reconstructions/projections at the interface. After a finite number of these interface treatments, the accuracy gain due to localization is surpassed by that successive degradation. This can be mitigated by applying closure and/or regularization techniques at the interface to enhance the reconstructed field and account for truncated modes before transferring into the subsequent manifold. On the other hand, for more challenging problems when the rate of decay is slow (such as 2D double shear and 2D Boussinesq cases), more localization helps to increase predictive accuracy of PID. This implies that for a higher Kolmogorov n-width barrier (i.e., lower decay rate), a larger number of subintervals N_p is required and the PID approach can offer a viable solution in such situations.

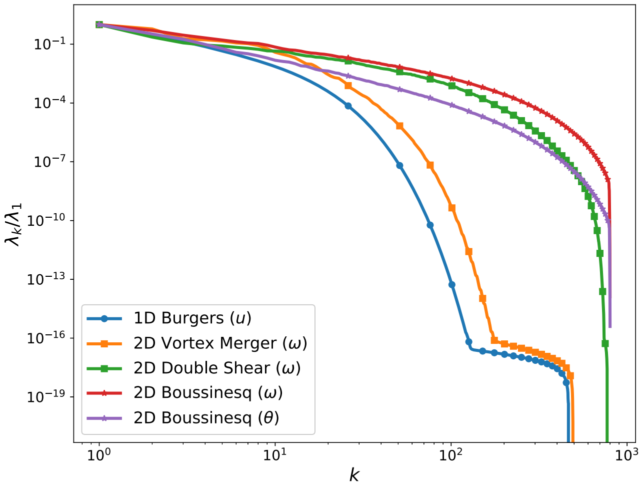

FIG. 23. Eigenspectrum plot for the investigated tested cases [normalized with respect to the first (largest) eigenvalue].

VIII. CONCLUSIONS

In the current study, we present a fully data-driven nonintrusive framework for convection-dominated problems. Most model reduction techniques depend on the ergodicity hypothesis which implies that any ensemble of realizations would carry the average statistical properties of the entire process. However, in convective flow problems (such as those investigated in this study), the convective mechanisms are more dominant than diffusive ones. The ergodicity hypothesis is therefore violated, making the application of the standard model reduction algorithms infeasible. Moreover, the Kolmogorov barrier constrains the reducibility of such systems. We address these issues by employing a splitting technique, based on principal interval decomposition. We divide the time domain into a set of equidistant partitions and apply the POD locally in each of them as our compression approach. For the system's dynamics (encapsulated in temporal coefficients), we train the corresponding LSTM models at each zone equipped with consistent interface conditions. PID-LSTM results are compared with the standard Galerkin projection framework. It is found that PID-GP provides less accurate results than PID-LSTM. This is particularly evident in problems where multiple fields are coupled, such as the 2D Boussinesq case. On the other side, PID-LSTM enables us to separate the quantity of interest and deploy our prediction on the most relevant problem-specific ones. Therefore, for example, the temperature field can be inferred without explicitly constructing the vorticity fields because of the nonintrusive nature of the predictive modeling framework. We also observe that for convection-dominated problems, the optimal number of intervals is dependent on the decay rate of the Kolmogorov *n*-width. Hence, we suggest that adaptive and automated partitioning or clustering techniques would be useful in this context. Finally, since the most expensive stages of PID-LSTM (decomposition and training) are performed offline, it is capable of providing near real-time responses during the online stage. This can serve as a key enabler for developing digital twin technologies, a topic that we plan to cover more in detail in the future.

ACKNOWLEDGMENTS

This material is based on work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Award No. DE-SC0019290. O.S. gratefully acknowledges their support. Direct numerical simulations for this project were performed using resources of the Oklahoma State University High Performance Computing Center.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

APPENDIX: BASIS FUNCTIONS

Here, we visualize the constructed basis functions of POD and PID approaches using different number of intervals. Specifically, we present the first 4 modes of POD application on the whole time interval. This illustrates the deformation of obtained modes by the severely varying system states with time. Also, we present the first mode computed locally in the first and last intervals (i.e., $\phi_1^{(1)}$ and $\phi_1^{(N_p)}$, respectively).

1. Burgers problem

Figure 24 shows the first four global functions calculated from the classical POD approach over the whole time interval. It can be observed that the shock is smoothed out as it is moving with time. As a result, none of these modes resembles the actual state of the flow and not much information about the location and characteristics of these shocks can be inferred from these global modes.

On the other hand, the application of PID results in local modes which give much better information about the shock characteristics. For example, the first mode in the first subinterval provides more accurate information about the initial shock location as emphasized in Fig. 25. As the number of intervals increases, the detection of the shock wave is improved. Similar results are obtained in Fig. 26, where the shock wave at the final time is captured.

2. Vortex merger problem

The contour plots for the global POD modes for the vortex merger problem are shown in Fig. 27. We can easily observe the

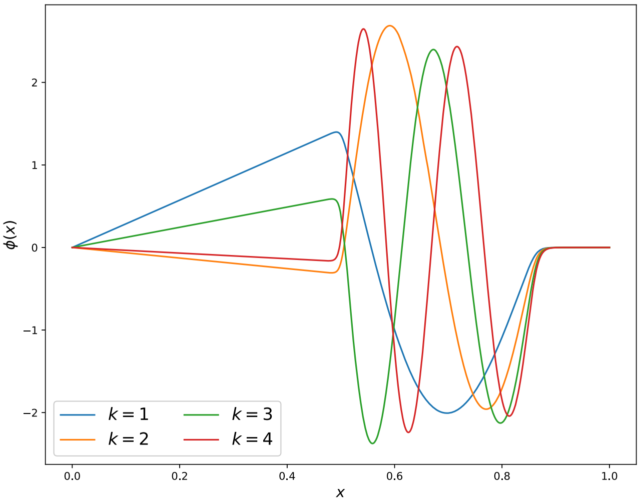

FIG. 24. The first 4 global basis functions from POD application over the whole time domain (i.e., for $0 \leq t \leq 2$) for the Burgers problem.

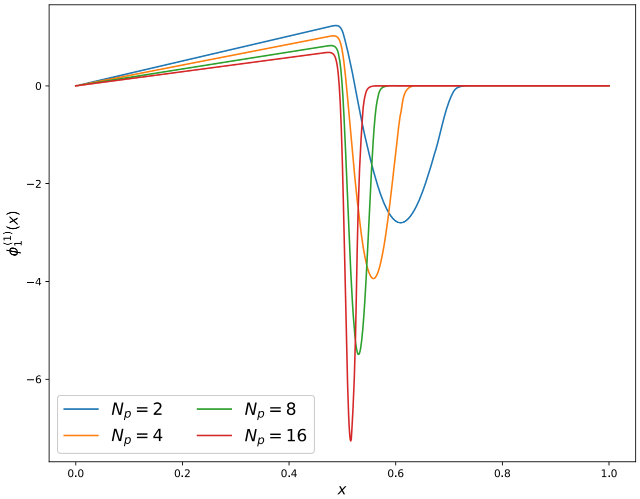

FIG. 25. The first local basis function from PID application over the first subinterval (i.e., for $0 \le t \le t_{\kappa^{(1)}}$) for the Burgers problem using different number of intervals.

modal deformation in such a way that the obtained modes give just an overview of the merging process without much detail about the growth of the two vortices. On the other hand, we can get more insights about the initial and final stages of the merging process from Figs. 28 and 29, respectively.

3. Double shear layer problem

Similar results are obtained for the double shear layer problem, where the global deformed POD modes are shown in Fig. 30. It is clear that the final field structures are captured by higher modes, where the first mode barely carries any information about the final

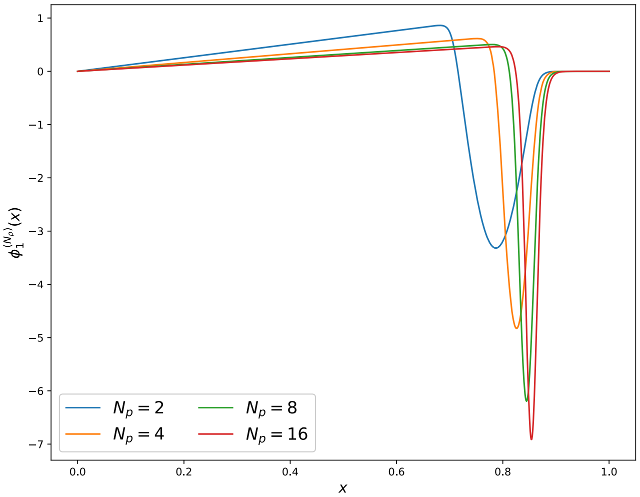

FIG. 26. The first local basis function from PID application over the last subinterval (i.e., for $t_{\kappa^{(N_p-1)}} \le t \le 2$) for the Burgers problem using different number of intervals.

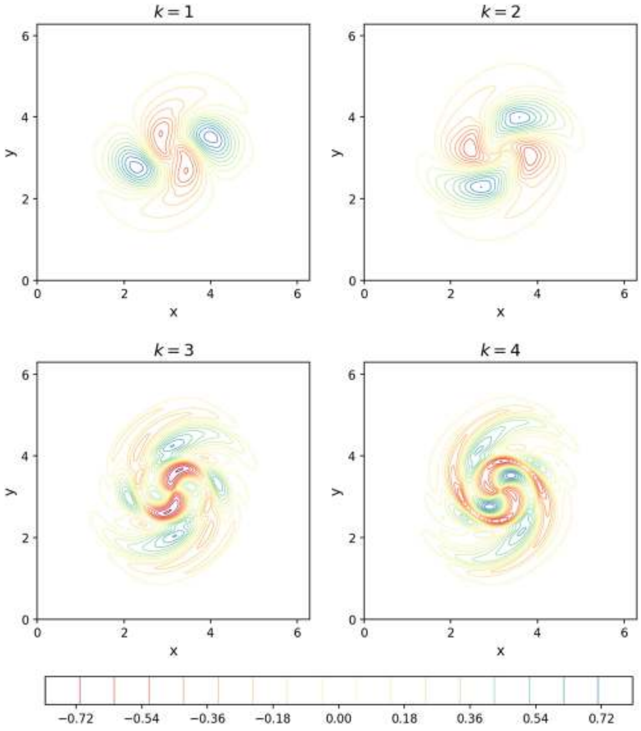

FIG. 27. The first 4 global basis functions for the vorticity field from POD application over the whole time domain (i.e., for $0 \le t \le 40$) for the vortex merger problem.

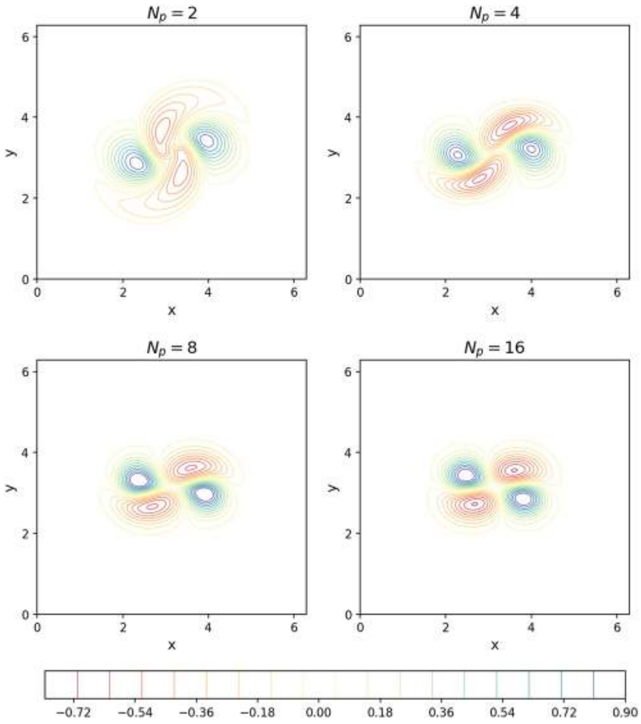

FIG. 28. The first local basis function for the vorticity field from PID application over the first subinterval [i.e., for $0 \le t \le t_{\kappa^{(1)}}$] for the vortex merger problem using different number of intervals.

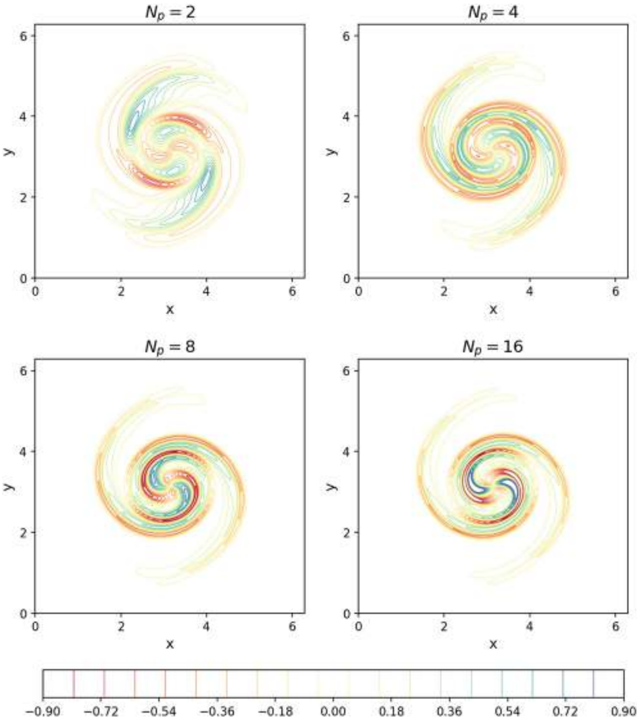

FIG. 29. The first local basis function for the vorticity field from PID application over the last subinterval [i.e., for $t_{\kappa^{(N_p-1)}} \leq t \leq 40$] for the vortex merger problem using different number of intervals.

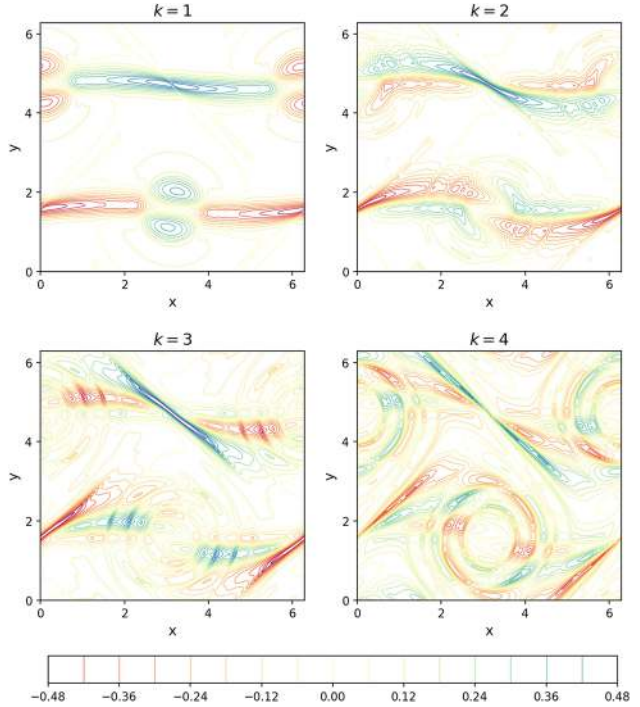

FIG. 30. The first 4 global basis functions for the vorticity field from POD application over the whole time domain (i.e., for $0 \leq t \leq 40$) for the double shear layer problem.

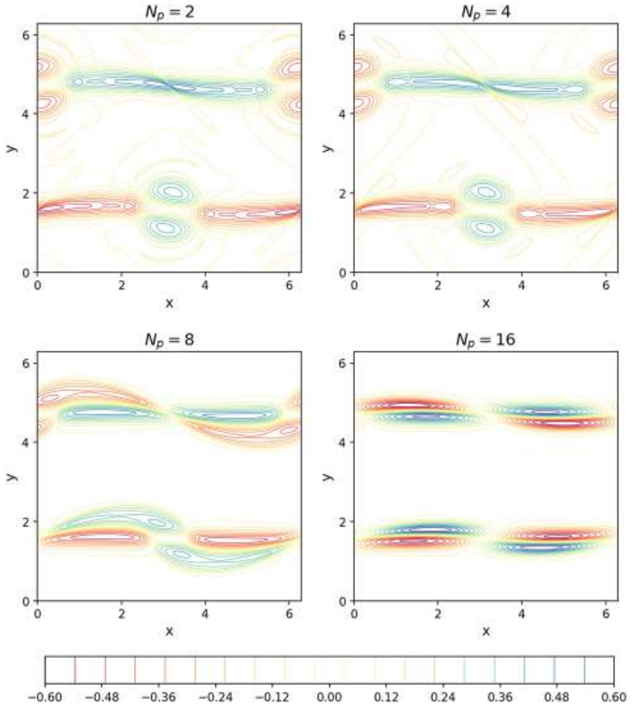

FIG. 31. The first local basis function for the vorticity field from PID application over the first subinterval [i.e., for $0 \leq t \leq t_{\kappa^{(1)}}$] for the double shear layer problem using different number of intervals.

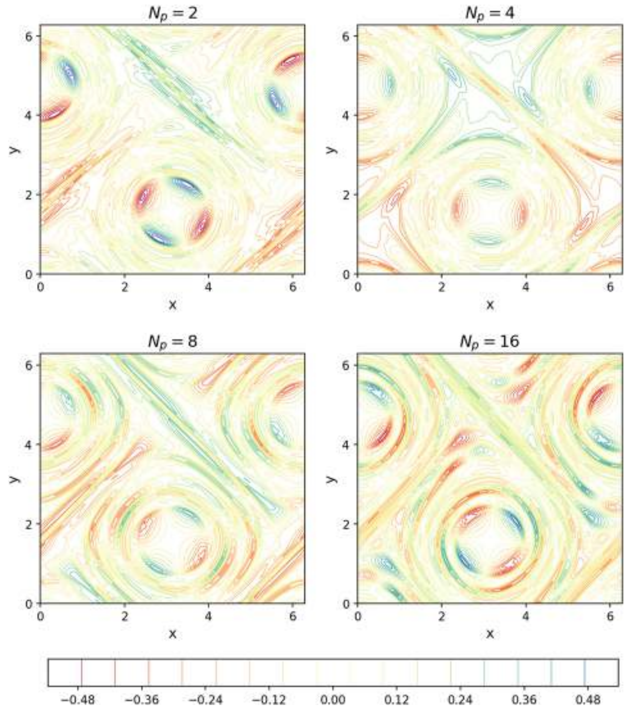

FIG. 32. The first local basis function for the vorticity field from PID application over the last subinterval [i.e., for $t_{\kappa^{(N_p-1)}} \leq t \leq 40$] for the double shear layer problem using different number of intervals.

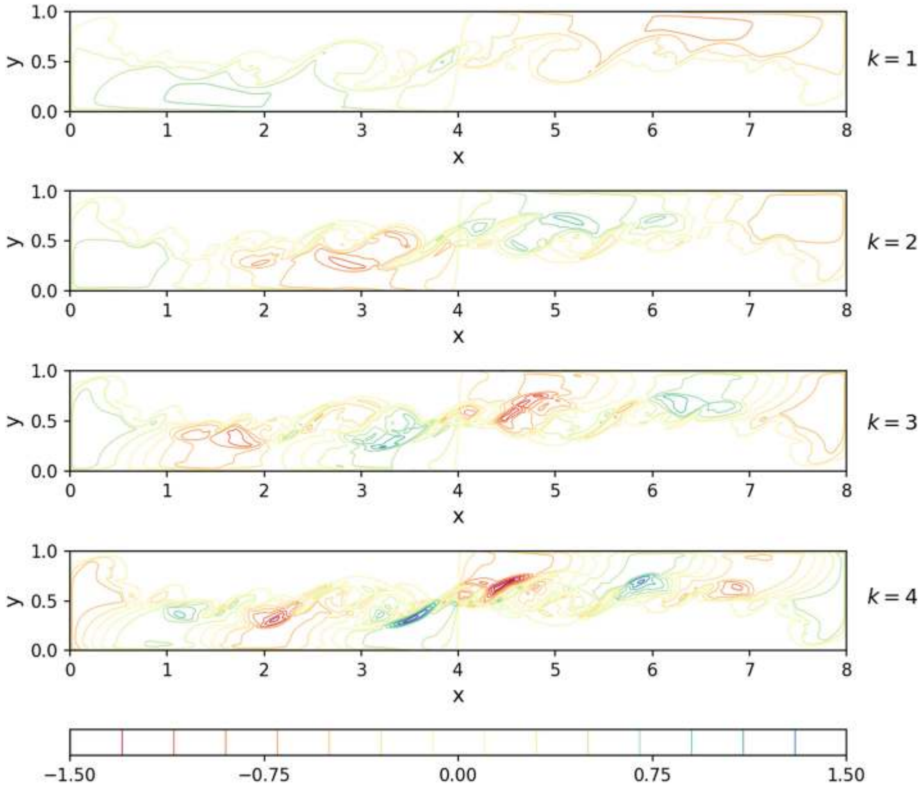

FIG. 33. The first 4 global basis functions for the temperature field from POD application over the whole time domain (i.e., for $0 \leq t \leq 8$) for the Boussinesq problem.

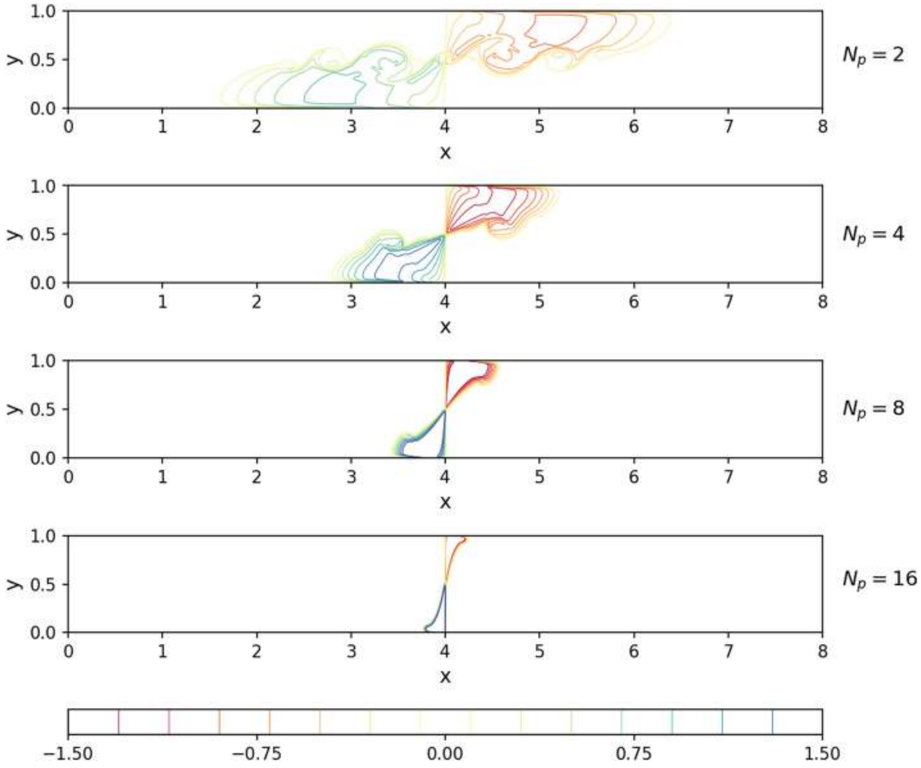

FIG. 34. The first local basis function for the temperature field from PID application over the first subinterval [i.e., for $0 \leq t \leq t_{\kappa(1)}$] for the Boussinesq problem using different number of intervals.

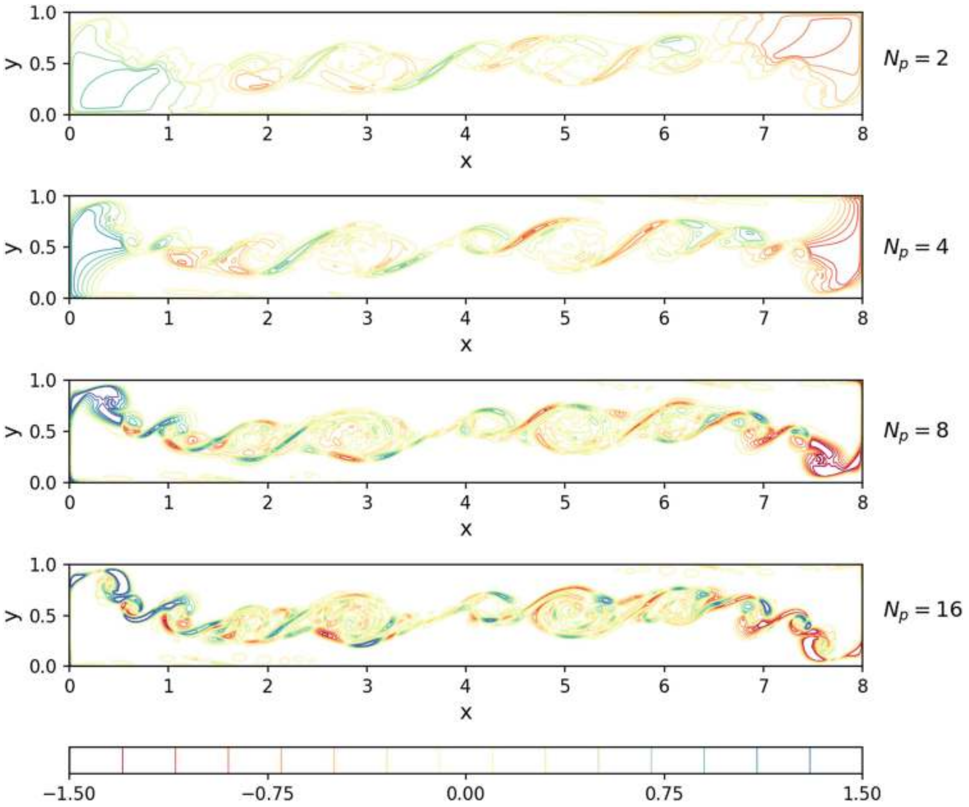

FIG. 35. The first local basis function for the temperature field from PID application over the last subinterval [i.e., for $t_{\kappa^{(N_p-1)}} \le t \le 8$] for the Boussinesq problem using different number of intervals.

field. Therefore, more modes need to be retained in the ROM. On the other hand, the local basis function for the first subinterval is similar to the initial field as shown in Fig. 31, while the first basis function in the last subinterval captures the main dynamics of the final field as depicted in Fig. 32.

4. Boussinesq problem

In the Boussinesq problem, we provide the first four POD modes for temperature fields in Fig. 33, which obviously demonstrates the averaging nature of POD. We can see that none of these modes looks like the initial or final fields. They just give an average image of the whole process. On the other hand, applying PID allows us to explore more about the details of the dynamical evolution and underlying instabilities. In Fig. 34, the initial state of the system can be identified by decomposing the domain into multiple local zones and investigating the first interval where the initial field is located. Similarly, information about the final temperature field can be acquired from narrowing the last partition (i.e., by increasing the number of intervals) as shown in Fig. 35.

REFERENCES

[1] B. R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, and F. Thiele, "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake," J. Fluid Mech. **497**, 335–363 (2003).

[2] D. J. Lucia, P. S. Beran, and W. A. Silva, "Reduced-order modeling: New approaches for computational physics," Prog. Aerosp. Sci. **40**, 51–117 (2004).

[3] A. Quarteroni, G. Rozza *et al.*, *Reduced Order Methods for Modeling and Computational Reduction* (Springer, Switzerland, 2014), Vol. 9.

[4] B. R. Noack, M. Morzynski, and G. Tadmor, *Reduced-Order Modelling for Flow Control* (Springer-Verlag, 2011), Vol. 528.

[5] K. Ito and S. S. Ravindran, "Reduced basis method for optimal control of unsteady viscous flows," Int. J. Comput. Fluid Dyn. **15**, 97–113 (2001).

[6] A. McNamara, A. Treuille, Z. Popović, and J. Stam, "Fluid control using the adjoint method," ACM Trans. Graphics **23**, 449–456 (2004).

[7] M. Bergmann and L. Cordier, "Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models," J. Comput. Phys. **227**, 7813–7840 (2008).

[8] S. S. Ravindran, "A reduced-order approach for optimal control of fluids using proper orthogonal decomposition," Int. J. Numer. Methods Fluids **34**, 425–448 (2000).

[9] W. Graham, J. Peraire, and K. Tang, "Optimal control of vortex shedding using low-order models. Part I—Open-loop model development," Int. J. Numer. Methods Eng. **44**, 945–972 (1999).

[10] W. Graham, J. Peraire, and K. Tang, "Optimal control of vortex shedding using low-order models. Part II—Model-based control," Int. J. Numer. Methods Eng. **44**, 973–990 (1999).

[11] S. L. Brunton and B. R. Noack, "Closed-loop turbulence control: Progress and challenges," Appl. Mech. Rev. **67**, 050801 (2015).

[12] D. N. Daescu and I. M. Navon, "Efficiency of a POD-based reduced second-order adjoint model in 4D-Var data assimilation," Int. J. Numer. Methods Fluids **53**, 985–1004 (2007).

[13] I. M. Navon, "Data assimilation for numerical weather prediction: A review," in *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications*, edited by S. K. Park and L. Xu (Springer, New York, 2009), pp. 21–65.

[14] Y. Cao, J. Zhu, I. M. Navon, and Z. Luo, "A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition," Int. J. Numer. Methods Fluids **53**, 1571–1583 (2007).

[15] J. He, P. Sarma, and L. J. Durlofsky, "Use of reduced-order models for improved data assimilation within an EnKF context," in *SPE Reservoir Simulation Symposium* (Society of Petroleum Engineers, Texas, 2011).

[16] P. L. Houtekamer and H. L. Mitchell, "Data assimilation using an ensemble Kalman filter technique," Mon. Weather Rev. **126**, 796–811 (1998).

[17] P. L. Houtekamer and H. L. Mitchell, "A sequential ensemble Kalman filter for atmospheric data assimilation," Mon. Weather Rev. **129**, 123–137 (2001).

[18] A. F. Bennett, *Inverse Modeling of the Ocean and Atmosphere* (Cambridge University Press, New York, 2005).

[19] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter* (Springer-Verlag, Berlin, Heidelberg, 2009).

[20] K. J. Law and A. M. Stuart, "Evaluating data assimilation algorithms," Mon. Weather Rev. **140**, 3757–3782 (2012).

[21] V. Buljak, *Inverse Analyses with Model Reduction: Proper Orthogonal Decomposition in Structural Mechanics* (Springer-Verlag, Berlin, Heidelberg, 2011).

[22] D. Xiao, J. Du, F. Fang, C. Pain, and J. Li, "Parameterised non-intrusive reduced order methods for ensemble Kalman filter data assimilation," Comput. Fluids **177**, 69–77 (2018).

[23] H. Fu, H. Wang, and Z. Wang, "POD/DEIM reduced-order modeling of time-fractional partial differential equations with applications in parameter identification," J. Sci. Comput. **74**, 220–243 (2018).

[24] M. Boulakia, E. Schenone, and J.-F. Gerbeau, "Reduced-order modeling for cardiac electrophysiology. Application to parameter identification," Int. J. Numer. Methods Biomed. Eng. **28**, 727–744 (2012).

[25] B. Kramer, B. Peherstorfer, and K. Willcox, "Feedback control for systems with uncertain parameters using online-adaptive reduced models," SIAM J. Appl. Dyn. Syst. **16**, 1563–1586 (2017).

[26] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas, "Non-linear model reduction for uncertainty quantification in large-scale inverse problems," Int. J. Numer. Methods Eng. **81**, 1581–1608 (2010).

[27] L. Biegler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, L. Tenorio, B. V. B. Waanders, K. Willcox, and Y. Marzouk, *Large-Scale Inverse Problems and Quantification of Uncertainty* (Wiley Online Library, Chichester, 2011).

[28] L. Mathelin, M. Y. Hussaini, and T. A. Zang, "Stochastic approaches to uncertainty quantification in CFD simulations," Numer. Algorithms **38**, 209–236 (2005).

[29] R. C. Smith, *Uncertainty Quantification: Theory, Implementation, and Applications* (SIAM, Philadelphia, 2014).

[30] H. N. Najm, "Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics," Annu. Rev. Fluid Mech. **41**, 35–52 (2009).

[31] M. J. Zahr, K. T. Carlberg, and D. P. Kouri, "An efficient, globally convergent method for optimization under uncertainty using adaptive model reduction and sparse grids," SIAM/ASA J. Uncertainty Quantif. **7**, 877–912 (2019).

[32] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and US air force vehicles," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA* (AIAA, 2012), p. 1818.

[33] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems* (Springer, 2017), pp. 85–113.

[34] S. Boschert and R. Rosen, "Digital twin—The simulation aspect," in *Mechatronic Futures* (Springer, 2016), pp. 59–74.

[35] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," Procedia CIRP **61**, 335–340 (2017).

[36] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," Int. J. Adv. Manuf. Technol. **94**, 3563–3576 (2018).

[37] M. Grieves, *Virtually Perfect: Driving Innovative and Lean Products through Product Lifecycle Management* (Space Coast Press, 2011).

[38] D. Hartmann, M. Herz, and U. Wever, "Model order reduction a key technology for digital twins," in *Reduced-Order Modeling (ROM) for Simulation and Optimization: Powerful Algorithms as Key Enablers for Scientific Computing*, edited by W. Keiper, A. Milde, and S. Volkwein (Springer, 2018).

[39] J. Roychowdhury, "Reduced-order modeling of time-varying systems," IEEE Trans. Circuits Syst. **46**, 1273–1288 (1999).

[40] M. Buffoni, S. Camarri, A. Iollo, and M. V. Salvetti, "Low-dimensional modelling of a confined three-dimensional wake flow," J. Fluid Mech. **569**, 141–150 (2006).

[41] S. Lakshmivarahan and Y. Wang, "On the relation between energy-conserving low-order models and a system of coupled generalized Volterra gyrostats with nonlinear feedback," J. Nonlinear Sci. **18**, 75–97 (2008).

[42] F. Fang, C. Pain, I. Navon, G. Gorman, M. Piggott, P. Allison, P. Farrell, and A. Goddard, "A POD reduced order unstructured mesh ocean modelling method for moderate Reynolds number flows," Ocean Modell. **28**, 127–136 (2009).

[43] R. Milk, S. Rave, and F. Schindler, "pyMOR—Generic algorithms and interfaces for model order reduction," SIAM J. Sci. Comput. **38**, S194–S216 (2016).

[44] V. Puzyrev, M. Ghommem, and S. Meka, "pyROM: A computational framework for reduced order modeling," J. Comput. Sci. **30**, 157–173 (2019).

[45] S. Volkwein, "Proper orthogonal decomposition: Theory and reduced-order modelling," in *Lecture Notes* (University of Konstanz, 2013), Vol. 4, pp. 1–29.

[46] C. Gräßle, M. Hinze, and S. Volkwein, "Model order reduction by proper orthogonal decomposition," preprint arXiv:1906.05188 (2019).

[47] K. Taira, S. L. Brunton, S. T. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, "Modal analysis of fluid flows: An overview," AIAA J. **55**, 4013–4041 (2017).

[48] C. W. Rowley and S. T. Dawson, "Model reduction for flow analysis and control," Annu. Rev. Fluid Mech. **49**, 387–417 (2017).

[49] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," J. Fluid Mech. **656**, 5–28 (2010).

[50] P. J. Schmid, L. Li, M. Juniper, and O. Pust, "Applications of the dynamic mode decomposition," Theor. Comput. Fluid Dyn. **25**, 249–259 (2011).

[51] K. K. Chen, J. H. Tu, and C. W. Rowley, "Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses," J. Nonlinear Sci. **22**, 887–915 (2012).

[52] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: Theory and applications," J. Comput. Dyn. **1**, 391–421 (2014).

[53] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems* (SIAM, Philadelphia, 2016).

[54] D. A. Bistrian and I. M. Navon, "The method of dynamic mode decomposition in shallow water and a swirling flow problem," Int. J. Numer. Methods Fluids **83**, 73–89 (2017).

[55] A. Kolmogoroff, "Über die beste annäherung von funktionen einer gegebenen funktionenklasse," Ann. Math. **37**, 107–110 (1936).

[56] A. Pinkus, *N-Widths in Approximation Theory* (Springer Science & Business Media, 2012), Vol. 7.

[57] T. Taddei, "A registration method for model order reduction: Data compression and geometry reduction," preprint arXiv:1906.11008 (2019).

[58] C. Greif and K. Urban, "Decay of the Kolmogorov N-width for wave problems," Appl. Math. Lett. **96**, 216–222 (2019).

[59] F. Flandoli and B. Maslowski, "Ergodicity of the 2-D Navier-Stokes equation under random perturbations," Commun. Math. Phys. **172**, 119–141 (1995).

[60] B. Ferrario, "Ergodic results for stochastic Navier-Stokes equation," Stochastics Stochastic Rep. **60**, 271–288 (1997).

[61] J. C. Mattingly, "Ergodicity of 2D Navier–Stokes equations with random forcing and large viscosity," Commun. Math. Phys. **206**, 273–288 (1999).

[62] G. Da Prato and A. Debussche, "Ergodicity for the 3D stochastic Navier–Stokes equations," J. Math. Pures Appl. **82**, 877–947 (2003).

[63] M. Hairer and J. C. Mattingly, "Ergodic properties of highly degenerate 2D stochastic Navier–Stokes equations," C. R. Math. **339**, 879–882 (2004).

[64] M. Hairer and J. C. Mattingly, "Ergodicity of the 2D Navier-Stokes equations with degenerate stochastic forcing," Ann. Math. **164**, 993–1032 (2006).

[65] T. G. Shepherd, "Non-ergodicity of inviscid two-dimensional flow on a beta-plane and on the surface of a rotating sphere," J. Fluid Mech. **184**, 289–302 (1987).

[66] A.-B. Cruzeiro and P. Malliavin, "Nonergodicity of Euler fluid dynamics on tori versus positivity of the Arnold–Ricci tensor," J. Funct. Anal. **254**, 1903–1925 (2008).

[67] J. Borggaard, A. Hay, and D. Pelletier, "Interval-based reduced order models for unsteady fluid flow," Int. J. Numer. Anal. Model. **4**, 353–367 (2007).

[68] O. San and J. Borggaard, "Principal interval decomposition framework for POD reduced-order modeling of convective Boussinesq flows," Int. J. Numer. Methods Fluids **78**, 37–62 (2015).

[69] M. Ahmed and O. San, "Stabilized principal interval decomposition method for model reduction of nonlinear convective systems with moving shocks," Comput. Appl. Math. **37**, 6870–6902 (2018).

[70] T. Taddei, S. Perotto, and A. Quarteroni, "Reduced basis techniques for nonlinear conservation laws," ESAIM: Math. Modell. Numer. Anal. **49**, 787–814 (2015).

[71] D. Xiao, C. Heaney, F. Fang, L. Mottet, R. Hu, D. Bistrian, E. Aristodemou, I. Navon, and C. Pain, "A domain decomposition non-intrusive reduced order model for turbulent flows," Comput. Fluids **182**, 15–27 (2019).

[72] D. Amsallem, M. J. Zahr, and C. Farhat, "Nonlinear model order reduction based on local reduced-order bases," Int. J. Numer. Methods Eng. **92**, 891–916 (2012).

[73] B. Peherstorfer and K. Willcox, "Online adaptive model reduction for nonlinear systems via low-rank updates," SIAM J. Sci. Comput. **37**, A2123–A2150 (2015).

[74] K. Lee and K. T. Carlberg, "Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders," J. Comp. Phys. (published online 2019).

[75] K. Ito and S. Ravindran, "A reduced-order method for simulation and control of fluid flows," J. Comput. Phys. **143**, 403–425 (1998).

[76] A. Iollo, S. Lanteri, and J.-A. Désidéri, "Stability properties of POD–Galerkin approximations for the compressible Navier–Stokes equations," Theor. Comput. Fluid Dyn. **13**, 377–396 (2000).

[77] C. W. Rowley, T. Colonius, and R. M. Murray, "Model reduction for compressible flows using POD and Galerkin projection," Physica D **189**, 115–129 (2004).

[78] R. Pinnau, "Model reduction via proper orthogonal decomposition," in *Model Order Reduction: Theory, Research Aspects and Applications*, edited by W. H. A. Schilders, A. H. van der Vorst, and J. Rommes (Springer, Berlin, Heidelberg, 2008), pp. 95–109.

[79] W. Stankiewicz, M. Morzyński, B. R. Noack, and G. Tadmor, "Reduced order Galerkin models of flow around NACA-0012 airfoil," Math. Modell. Anal. **13**, 113–122 (2008).

[80] E. W. Sachs and S. Volkwein, "POD-Galerkin approximations in PDE-constrained optimization," GAMM-Mitt. **33**, 194–208 (2010).

[81] I. Akhtar, A. H. Nayfeh, and C. J. Ribbens, "On the stability and extension of reduced-order Galerkin models in incompressible flows," Theor. Comput. Fluid Dyn. **23**, 213–237 (2009).

[82] M. F. Barone, I. Kalashnikova, D. J. Segalman, and H. K. Thornquist, "Stable Galerkin reduced order models for linearized compressible flow," J. Comput. Phys. **228**, 1932–1946 (2009).

[83] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," SIAM J. Sci. Comput. **32**, 2737–2764 (2010).

[84] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations," C. R. Math. **339**, 667–672 (2004).

[85] R. Ştefănescu and I. M. Navon, "POD/DEIM nonlinear model order reduction of an ADI implicit shallow water equations model," J. Comput. Phys. **237**, 95–114 (2013).

[86] H. Antil, M. Heinkenschloss, and D. C. Sorensen, "Application of the discrete empirical interpolation method to reduced order modeling of nonlinear and parametric systems," in *Reduced Order Methods for Modeling and Computational Reduction* (Springer, 2014), pp. 101–136.

[87] Z. P. Stanko, S. E. Boyce, and W. W.-G. Yeh, "Nonlinear model reduction of unconfined groundwater flow using POD and DEIM," Adv. Water Resour. **97**, 130–143 (2016).

[88] R. Ştefănescu, A. Sandu, and I. M. Navon, "Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations," Int. J. Numer. Methods Fluids **76**, 497–521 (2014).

[89] J.-C. Loiseau, B. R. Noack, and S. L. Brunton, "Sparse reduced-order modelling: Sensor-based dynamics to full-state estimation," J. Fluid Mech. **844**, 459–490 (2018).

[90] X. Xie, G. Zhang, and C. G. Webster, "Non-intrusive inference reduced order model for fluids using deep multistep neural network," Mathematics **7**, 757 (2019).

[91] D. Xiao, F. Fang, C. Pain, and I. Navon, "Towards non-intrusive reduced order 3D free surface flow modelling," Ocean Eng. **140**, 155–168 (2017).

[92] Z. Lin, D. Xiao, F. Fang, C. Pain, and I. M. Navon, "Non-intrusive reduced order modelling with least squares fitting on a sparse grid," Int. J. Numer. Methods Fluids **83**, 291–306 (2017).

[93] D. Xiao, F. Fang, C. Pain, and I. Navon, "A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications," Comput. Methods Appl. Mech. Eng. **317**, 868–889 (2017).

[94] D. Xiao, Z. Lin, F. Fang, C. C. Pain, I. M. Navon, P. Salinas, and A. Muggeridge, "Non-intrusive reduced-order modeling for multiphase porous media flows using Smolyak sparse grids," Int. J. Numer. Methods Fluids **83**, 205–219 (2017).

[95] D. Xiao, P. Yang, F. Fang, J. Xiang, C. C. Pain, and I. M. Navon, "Non-intrusive reduced order modelling of fluid–structure interactions," Comput. Methods Appl. Mech. Eng. **303**, 35–54 (2016).

[96] D. Xiao, F. Fang, C. Pain, and G. Hu, "Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation," Int. J. Numer. Methods Fluids **79**, 580–595 (2015).

[97] D. Xiao, F. Fang, A. Buchan, C. Pain, I. Navon, and A. Muggeridge, "Non-intrusive reduced order modelling of the Navier–Stokes equations," Comput. Methods Appl. Mech. Eng. **293**, 522–541 (2015).

[98] B. Peherstorfer, "Sampling low-dimensional Markovian dynamics for pre-asymptotically recovering reduced models from data with operator inference," preprint arXiv:1908.11233 (2019).

[99] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys. **378**, 686–707 (2019).

[100] R. Maulik and O. San, "A neural network approach for the blind deconvolution of turbulent flows," J. Fluid Mech. **831**, 151–181 (2017).

[101] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Multistep neural networks for data-driven discovery of nonlinear dynamical systems," preprint arXiv:1801.01236 (2018).

[102] C. Lee, J. Kim, D. Babcock, and R. Goodman, "Application of neural networks to turbulence control for drag reduction," Phys. Fluids **9**, 1740–1747 (1997).

[103] R. Maulik, O. San, A. Rasheed, and P. Vedula, "Data-driven deconvolution for large eddy simulations of Kraichnan turbulence," Phys. Fluids **30**, 125109 (2018).

[104] W. E. Faller and S. J. Schreck, "Unsteady fluid mechanics applications of neural networks," J. Aircr. **34**, 48–55 (1997).

[105] O. San and R. Maulik, "Neural network closures for nonlinear model order reduction," Adv. Comput. Math. **44**, 1717–1750 (2018).

[106] Q. Wang, J. S. Hesthaven, and D. Ray, "Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem," J. Comput. Phys. **384**, 289–307 (2019).

[107] A. Moosavi, R. Stefanescu, and A. Sandu, "Efficient construction of local parametric reduced order models using machine learning techniques," preprint arXiv:1511.02909 (2015).

[108] J. N. Kani and A. H. Elsheikh, "DR-RNN: A deep residual recurrent neural network for model reduction," preprint arXiv:1709.00939 (2017).

[109] S. Brunton, B. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," Annu. Rev. Fluid Mech. **52**, 477–508 (2019).

[110] J. N. Kutz, "Deep learning in fluid dynamics," J. Fluid Mech. **814**, 1–4 (2017).

[111] P. A. Durbin, "Some recent developments in turbulence closure modeling," Annu. Rev. Fluid Mech. **50**, 77–103 (2018).

[112] K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence modeling in the age of data," Annu. Rev. Fluid Mech. **51**, 357–377 (2019).

[113] J. C. B. Gamboa, "Deep learning for time-series analysis," preprint arXiv:1701.01887 (2017).

[114] S. Pawar, S. Rahman, H. Vaddireddy, O. San, A. Rasheed, and P. Vedula, "A deep learning enabler for nonintrusive reduced order modeling of fluid flows," Phys. Fluids **31**, 085101 (2019).

[115] V. M. Krasnopolsky and M. S. Fox-Rabinovitz, "Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction," Neural Networks **19**, 122–134 (2006).

[116] V. M. Krasnopolsky and M. S. Fox-Rabinovitz, "A new synergetic paradigm in environmental numerical modeling: Hybrid models combining deterministic and machine learning components," Ecol. Modell. **191**, 5–18 (2006).

[117] S. Rahman, O. San, and A. Rasheed, "A hybrid approach for model order reduction of barotropic quasi-geostrophic turbulence," Fluids **3**, 86 (2018).

[118] Z. Y. Wan, P. Vlachas, P. Koumoutsakos, and T. Sapsis, "Data-assisted reduced-order modeling of extreme events in complex dynamical systems," PLoS One **13**, e0197704 (2018).

[119] X. Xie, M. Mohebujjaman, L. G. Rebholz, and T. Iliescu, "Data-driven filtered reduced order modeling of fluid flows," SIAM J. Sci. Comput. **40**, B834–B857 (2018).

[120] J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, "Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model," Chaos **28**, 041101 (2018).

[121] A. Majda, *Introduction to PDEs and Waves for the Atmosphere and Ocean* (American Mathematical Society, Providence, 2003), Vol. 9.

[122] J.-G. Liu, C. Wang, and H. Johnston, "A fourth order scheme for incompressible Boussinesq equations," J. Sci. Comput. **18**, 253–285 (2003).

[123] A. Nicolás and B. Bermúdez, "2D thermal/isothermal incompressible viscous flows," Int. J. Numer. Methods Fluids **48**, 349–366 (2005).

[124] D. J. Tritton, *Physical Fluid Dynamics* (Oxford University Press, Oxford, 1988).

[125] L. Sirovich, "Turbulence and the dynamics of coherent structures. I. Coherent structures," Q. Appl. Math. **45**, 561–571 (1987).

[126] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," Annu. Rev. Fluid Mech. **25**, 539–575 (1993).

[127] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry* (Cambridge University Press, Cambridge, 2012).

[128] V. L. Kalb and A. E. Deane, "An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models," Phys. Fluids **19**, 054106 (2007).

[129] M. Bergmann, C.-H. Bruneau, and A. Iollo, "Enablers for robust POD models," J. Comput. Phys. **228**, 516–538 (2009).

[130] J. Borggaard, T. Iliescu, and Z. Wang, "Artificial viscosity proper orthogonal decomposition," Math. Comput. Modell. **53**, 269–279 (2011).

[131] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu, "Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison," Comput. Methods Appl. Mech. Eng. **237**, 10–26 (2012).

[132] S. M. Rahman, S. E. Ahmed, and O. San, "A dynamic closure modeling framework for model order reduction of geophysical flows," Phys. Fluids **31**, 046602 (2019).

[133] W. Cazemier, R. Verstappen, and A. Veldman, "Proper orthogonal decomposition and low-dimensional models for driven cavity flows," Phys. Fluids **10**, 1685–1699 (1998).

[134] W. IJzerman, "Signal representation and modeling of spatial structures in fluids," Ph.D. thesis, University of Twente, 2000.

[135] M. Dihlmann, M. Drohmann, and B. Haasdonk, "Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning," in *International Conference on Adaptive Modeling and Simulation (ADMOS 2011)* (International Center for Numerical Methods in Engineering, 2011), pp. 156–167.

[136] O. San and T. Iliescu, "Proper orthogonal decomposition closure models for fluid flows: Burgers equation," Int. J. Numer. Anal. Model. **5**, 217–237 (2014).

[137] M. J. Gander and S. Vandewalle, "Analysis of the parareal time-parallel time-integration method," SIAM J. Sci. Comput. **29**, 556–578 (2007).

[138] K. Carlberg, L. Brencher, B. Haasdonk, and A. Barth, "Data-driven time parallelism via forecasting," SIAM J. Sci. Comput. **41**, B466–B496 (2019).

[139] K. Yeo and I. Melnyk, "Deep learning algorithm for data-driven simulation of noisy dynamical system," J. Comput. Phys. **376**, 1212–1231 (2019).

[140] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," Science **304**, 78–80 (2004).

[141] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**, 436 (2015).

[142] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks," Proc. R. Soc. A **474**, 20170844 (2018).

[143] A. T. Mohan and D. V. Gaitonde, "A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks," preprint arXiv:1804.09269 (2018).

[144] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in Fifteenth Annual Conference of the International Speech Communication Association, 2014.

[145] Y. Bengio, P. Simard, P. Frasconi *et al.*, "Learning long-term dependencies with gradient descent is difficult," IEEE Trans. Neural Networks **5**, 157–166 (1994).

[146] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," Neural Comput. **12**, 2451–2471 (1999).

[147] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput. **9**, 1735–1780 (1997).

[148] F. Chollet *et al.*, Keras, 2015, https://keras.io.

[149] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), 2015.

[150] A. Das and R. D. Moser, "Optimal large-eddy simulation of forced Burgers equation," Phys. Fluids **14**, 4344–4351 (2002).

[151] S. K. Lele, "Compact finite difference schemes with spectral-like resolution," J. Comput. Phys. **103**, 16–42 (1992).

[152] S. Gottlieb and C.-W. Shu, "Total variation diminishing Runge-Kutta schemes," Math. Comput. **67**, 73–85 (1998).

[153] M. Maleewong and S. Sirisup, "On-line and off-line POD assisted projective integral for non-linear problems: A case study with Burgers' equation," Int. J. Math., Comput., Phys., Electr. Comput. Eng. **5**, 984–992 (2011).

[154] M. D. Gunzburger, *Flow Control* (Springer-Verlag, New York, 2012), Vol. 68.

[155] J. D. Buntine and D. Pullin, "Merger and cancellation of strained vortices," J. Fluid Mech. **205**, 263–295 (1989).

[156] J. Von Hardenberg, J. McWilliams, A. Provenzale, A. Shchepetkin, and J. Weiss, "Vortex merging in quasi-geostrophic flows," J. Fluid Mech. **412**, 331–353 (2000).

[157] P. Meunier, S. Le Dizès, and T. Leweke, "Physics of vortex merging," C. R. Phys. **6**, 431–450 (2005).

[158] O. San and A. E. Staples, "A coarse-grid projection method for accelerating incompressible flow computations," J. Comput. Phys. **233**, 480–508 (2013).

[159] J. B. Bell, P. Colella, and H. M. Glaz, "A second-order projection method for the incompressible Navier-Stokes equations," J. Comput. Phys. **85**, 257–283 (1989).

[160] O. San and A. E. Staples, "High-order methods for decaying two-dimensional homogeneous isotropic turbulence," Comput. Fluids **63**, 105–127 (2012).

[161] B. Soffientino and M. E. Q. Pilson, "The Bosporus strait: A special place in the history of oceanography," Oceanography **18**, 16–23 (2005).

[162] A. E. Gill, *Atmosphere-Ocean Dynamics* (Academic Press, San Diego, California, USA, 1982), Vol. 30.