A LONG SHORT-TERM MEMORY EMBEDDING FOR HYBRID UPLIFTED REDUCED ORDER MODELS

Shady E. Ahmed
School of Mechanical & Aerospace Engineering,
Oklahoma State University,
Stillwater, Oklahoma - 74078, USA.
shady.ahmed@okstate.edu

Omer San
School of Mechanical & Aerospace Engineering,
Oklahoma State University,
Stillwater, Oklahoma - 74078, USA.
osan@okstate.edu

Adil Rasheed
Department of Engineering Cybernetics,
Norwegian University of Science and Technology,
N-7465, Trondheim, Norway.
adil.rasheed@ntnu.no

Traian Iliescu
Department of Mathematics,
Virginia Tech,
Blacksburg, VA 24061, USA.
iliescu@vt.edu

ABSTRACT

In this paper, we introduce an uplifted reduced order modeling (UROM) approach through the integration of standard projection based methods with long short-term memory (LSTM) embedding. Our approach has three modeling layers or components. In the first layer, we utilize an intrusive projection approach to model dynamics represented by the largest modes. The second layer consists of an LSTM model to account for residuals beyond this truncation. This closure layer refers to the process of including the residual effect of the discarded modes into the dynamics of the largest scales. However, the feasibility of generating a low rank approximation tails off for higher Kolmogorov n-width systems due to the underlying nonlinear processes. The third uplifting layer, called super-resolution, addresses this limited representation issue by expanding the span into a larger number of modes utilizing the versatility of LSTM. Therefore, our model integrates a physics-based projection model with a memory embedded LSTM closure and an LSTM based super-resolution model. In several applications, we exploit the use of Grassmann manifold to construct UROM for unseen conditions. We performed numerical experiments by using the Burgers and Navier-Stokes equations with quadratic nonlinearity. Our results show robustness of the proposed approach in building reduced order models for parameterized systems and confirm the improved trade-off between accuracy and efficiency.

Keywords Hybrid analysis and modeling · Supervised machine learning · Long short-term memory · Model reduction · Galerkin projection · Grassmann manifold.

1 Introduction

Physical models are often sought because of their reliability, interpretability, and generalizability being derived from basic principles and physical intuition. However, accurate solution of these models for complex systems usually requires the use of very high spatial and temporal resolutions and/or sophisticated discretization techniques. This limits their applications to offline simulations over a few set of parameters and short time intervals since they can be excessively computationally-demanding. Although those are valuable in understanding physical phenomena and gaining more insight, realistic applications often require near real-time and multi-query responses [1]. Therefore, cheaper numerical approximations using "adequate-fidelity" models are usually acceptable [2]. In this regard, reduced order modeling offers a viable technique to address systems characterized by underlying patterns [3–12]. This is especially true for fluid flows dominated by coherent structures (e.g., atmospheric and oceanic flows) [13–21].

Reduced order models (ROMs) have shown great success for prototypical problems in different fields. In particular, Galerkin projection (GP) coupled with proper orthogonal decomposition (POD) capability to extract the most energetic modes has been used to build ROMs for linear and nonlinear systems [22–29]. These ROMs preserve sufficient

interpretability and generalizability since they are constructed by projecting the full order model (FOM) operators (from governing equations) on a reduced subspace. Despite that, Galerkin projection ROMs (GROMs) have severe limitations in practice, especially for systems with strong nonlinearity. Most fluid flows exhibit quadratic nonlinearity, which makes the computational cost of the resulting GROMs $\sim O(R^3)$, where R is the number of retained modes in ROM approximation. As a result, R should be kept as low as possible (e.g., $O(5)$) through modal truncation for practical purposes; however, this has two main consequences. First, the solution is enforced to live in a smaller subspace which might not contain enough information to accurately represent complex realistic systems. Examples include advection-dominated flows and parametric systems where the decay of Kolmogorov n-width is slow [30, 31]. Second, due to the inherent system's nonlinearity, the truncated modes interact with the retained modes. In GROM, these interactions are simply eliminated by modal truncation, which often generates instabilities in the approximation [32–34]. Several efforts have been devoted to introduce stabilization and closure techniques [35–51] to account for the effects of truncated modes on ROM's dynamics.

In the present study, we aim to address the above problems while preserving considerable interpretability, and generalizability at the core of our uplifted ROM (UROM) approach. In UROM, we present a three-modeling layer framework. In the first layer, we use a standard Galerkin projection method based on the governing equations to model the large scales of the flow (represented by the first few R POD modes) and provide a predictor for the temporal evolution. In the next layer, we introduce a corrector step to correct the Galerkin approximation and make up for the interactions of the truncated modes (or scales) with the large ones. These large scales contribute most to the total system's energy. That is why we dedicate two layers of our approach to correctly resolve them, one of which (i.e., the Galerkin projection) is totally physics-based to promote framework generality. In the third layer, we uplift our approximation and extend the solution subspace to recover some of the flow's finer details by learning a map between the large scales (predicted using the first two layers) and smaller scales.

In particular, we choose the first $R \approx O(5)$ modes to represent the resolved largest scales and the next $Q - R$ modes to represent the resolved smaller scales, where Q is about 4 to 8 times larger than R. For the second and third layers, we incorporate memory embedding through the use of long short-term memory (LSTM) neural network architecture [52,53]. Machine learning (ML) tools (of which neural networks is a subclass) have been gaining popularity in fluid mechanics community and analysis of dynamical systems [54–61]. In particular, LSTMs have shown great success in learning maps of sequential data and time-series [62–66]. It should be noted here that UROM can be thought of as a way of augmenting physical models with data-driven tools and vice versa. For the former, an LSTM closure model (second component) is developed to correct GROM and an LSTM super-resolution model (third component) is constructed to uplift GROM and resolve smaller scales. This relaxes the computational cost of GROM to account only for a few R modes. On the other hand, LSTMs in UROM framework are fed with inputs coming from a physics-based approach. This is one way of utilizing physical information rather than fully depend on ML results.

To illustrate the UROM framework, we consider two convection-dominated flows as test cases. The first is the one-dimensional (1D) Burgers equation, which is a simplified benchmark problem for fluid flows with strong nonlinearity. As the second test case, we investigate the two-dimensional (2D) Navier-Stokes equations for a flow with interacting vortices, namely the vortex merger. We compare the UROM approach with the standard GROM approach using R and Q modes. We also investigate a fully non-intrusive ROM (NIROM) approach in these flow problems. We perform a comparison in terms of solution accuracy and computational time to show the pros and cons of UROM with respect to either GROM or NIROM approaches. The rest of the paper is outlined here. In Section 2, we introduce the POD technique for data compression and constructing lower-dimensional subspaces to approximate the solution. For an out-of-sample control parameter (e.g., Reynolds number), we describe basis interpolation via Grassmann manifold approach in Section 3. As a standard physics-informed technique for building ROMs, we introduce Galerkin projection in Section 4 with a brief description of the governing equations of the test cases as well as their corresponding GROMs. In Section 5, we present the proposed UROM framework with a description of its main features. We give results and corresponding discussions in Section 6. Finally, we draw concluding remarks and insights in Section 7.

2 Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) is one of the most popular techniques for dimensionality reduction and data compression [13, 67–70]. Given datasets, POD provides a linear subspace that minimizes the projection error between the true data and its projection compared to all possible linear subspaces with the same dimension. If a number of N_s data snapshots, $\mathbf{u}(\mathbf{x}, t_n)$, where $n \in \{1, 2, \ldots, N_s\}$, $\mathbf{x} \in \mathbb{R}^N$ (N being the spatial resolution), are collected in a snapshot matrix $\mathbf{A} \in \mathbb{R}^{N \times N_s}$, then a reduced (or thin) singular value decomposition (SVD) can be applied to this matrix as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{N \times N_s}$ is a unitary matrix whose columns are the left-singular vectors of $\mathbf{A}$, also known as spatial basis and $\mathbf{V} \in \mathbb{R}^{N_s \times N_s}$ is also a unitary matrix whose columns represent the right-singular vectors, sometimes referred to as temporal basis . $\mathbf{\Sigma} \in \mathbb{R}^{N_s \times N_s}$ is a diagonal matrix whose entries are the singular values of $\mathbf{A}$ (square-roots of the largest N_s eigenvalues of $\mathbf{AA}^T$ or $\mathbf{A}^T\mathbf{A}$). In $\mathbf{\Sigma}$, the singular values σ_i are sorted in a descending order such that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{N_s} \geq 0$.

For dimensionality reduction purposes, only the first R columns of $\mathbf{U}$ (denoted as $\widehat{\mathbf{U}}$), the first R columns of $\mathbf{V}$ (denoted as $\widehat{\mathbf{V}}$), and the upper-left $R \times R$ block sub-matrix of $\mathbf{\Sigma}$ (denoted as $\widehat{\mathbf{\Sigma}}$) are retained to provide a reduced order approximation $\widehat{\mathbf{A}}$ of $\mathbf{A}$ as

$$\widehat{\mathbf{A}} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^T. \tag{2}$$

It can be easily shown that this approximation $\widehat{\mathbf{A}}$ satisfies the following infimum [71]

$$\|\mathbf{A} - \widehat{\mathbf{A}}\|_2 = \inf_{\substack{\mathbf{B} \in \mathbb{R}^{N \times N_s} \\ rank(\mathbf{B}) \leq R}} \|\mathbf{A} - \mathbf{B}\|_2 \tag{3}$$

$$\|\mathbf{A} - \widehat{\mathbf{A}}\|_2 = \sigma_{R+1}, \tag{4}$$

where $\| \cdot \|_2$ refers to the induced ℓ_2 matrix norm. Equation 3 means that across all possible matrices $\mathbf{B} \in \mathbb{R}^{N \times N_s}$ with a rank of R (or less), $\widehat{\mathbf{A}}$ provides the closest one to $\mathbf{A}$ in the ℓ_2 sense. Moreover, the singular values σ provide a measure of the quality of this approximation as Equation 4 shows that the ℓ_2 norm between the matrix $\mathbf{A}$ and its R-rank approximation equals to σ_{R+1}. From now on, the first R columns of $\mathbf{U}$ will be referred to as the POD modes or basis function, denoted as $\mathbf{\Phi} = [\phi_1, \phi_2, \ldots, \phi_R]$.

3 Grassmann Manifold Interpolation

In recent years, Grassmann manifold has attracted great interest in various applications including model order reduction for parametric systems [72–76]. The Grassmann manifold, $\mathcal{G}(q, N)$, is a set of all q-dimensional subspaces in $\mathbb{R}^N$, where $0 \leq q \leq N$. A point $[\mathbf{\Phi}] \in \mathcal{G}(q, N)$ is given as [77]

$$[\mathbf{\Phi}] = \{\mathbf{\Phi}Q \mid \mathbf{\Phi}^T\mathbf{\Phi} = I_q, Q \in \mathcal{O}(q)\}, \tag{5}$$

where $\mathbf{\Phi} \in \mathbb{R}^{N \times q}$ and $\mathcal{O}(q)$ is the group of all $q \times q$ orthogonal matrices. This point represents a q-dimensional subspace $\mathcal{S}$ in $\mathbb{R}^N$ spanned by the columns of $\mathbf{\Phi}$. At each point $[\mathbf{\Phi}] \in \mathcal{G}(q, N)$, a tangent space $\mathcal{T}([\mathbf{\Phi}])$ of the same dimension, $N \times q$, can be defined as follows [78, 79]

$$\mathcal{T}([\mathbf{\Phi}]) = \{\mathcal{X} \in \mathbb{R}^{N \times q} \mid \mathbf{\Phi}^T\mathcal{X} = \mathbf{0}\}. \tag{6}$$

Similarly, each point $[\mathbf{\Gamma}]$ on $\mathcal{T}$ represents a subspace spanned by the columns of $\mathbf{\Gamma}$. This tangent space is a vector space with its origin at $[\mathbf{\Phi}]$. An exponential mapping from a point $[\mathbf{\Gamma}] \in \mathcal{T}([\mathbf{\Phi}])$ to $[\mathbf{\Psi}] \in \mathcal{G}(q, N)$ can be defined as

$$\mathbf{\Psi} = \left(\mathbf{\Phi}\mathbf{V}\cos{(\mathbf{\Sigma})} + \mathbf{U}\sin{(\mathbf{\Sigma})}\right)\mathbf{V}^T, \tag{7}$$

where $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}$ are obtained from the reduced SVD of $\mathbf{\Gamma}$ as $\mathbf{\Gamma} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Inversely, a logarithmic map can be defined from a point $[\mathbf{\Psi}]$ in the neighborhood of $[\mathbf{\Phi}]$ to $[\mathbf{\Gamma}] \in \mathcal{T}([\mathbf{\Phi}])$ is defined as

$$\mathbf{\Gamma} = \mathbf{U}\tan^{-1}{(\mathbf{\Sigma})}\mathbf{V}^T, \tag{8}$$

where $(\mathbf{\Psi} - \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{\Psi})(\mathbf{\Phi}\mathbf{\Psi})^{-1} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. We would like to note here that the trigonometric functions are applied element-wise on the diagonal entries.

To demonstrate the procedure in our ROM context, for a number N_p of control parameters $\{\mu_i\}_{i=1}^{N_p}$, different sets of POD basis functions are computed corresponding to each parameter, denoted as $\{\mathbf{\Phi}_i\}_{i=1}^{N_p}$. These bases correspond to a set of points on the Grassmann manifold $\mathcal{G}(R, N)$. To perform an out-of-sample testing, the basis functions $\mathbf{\Phi}_{\text{Test}}$ for the test parameter μ_{Test} should be computed through interpolation. However, direct interpolation of the POD bases is not effective since it is an interpolation on a non-flat space and it does not guarantee that the resulting point would lie on $\mathcal{G}(R, N)$. Moreover, the optimality and orthonormality characteristics of POD are not necessarily conserved. Alternatively, the tangent space $\mathcal{T}$ is a flat space where standard interpolation can be performed effectively. First, a reference point at the Grassmann manifold is selected, corresponding to $\mathbf{\Phi}_{\text{Ref}}$. The tangent plane at this point is thus

defined using Equation 6. Then, the neighboring points on Grassmann manifold corresponding to the subspaces spanned by $\{\mathbf{\Phi}_i\}_{i=1}^{N_p}$ are mapped onto that tangent plane using the logarithmic map, defined in Equation 8 to calculate $\{\mathbf{\Gamma}_i\}_{i=1}^{N_p}$. Standard Lagrange interpolation can be performed to compute $\mathbf{\Gamma}_{\text{Test}}$ as follows

$$\mathbf{\Gamma}_{\text{Test}} = \sum_{i=1}^{N_p} \left(\prod_{\substack{j=1 \\ j \neq i}}^{N_p} \frac{\mu_{\text{Test}} - \mu_j}{\mu_i - \mu_j} \right) \mathbf{\Gamma}_i. \tag{9}$$

Finally, the point $[\mathbf{\Gamma}_{\text{Test}}] \in \mathcal{T}([\mathbf{\Phi}_{\text{Ref}}])$ is mapped to the Grassmann manifold $\mathcal{G}(R, N)$ to obtain the set of POD basis functions at the test parameter, $\mathbf{\Phi}_{\text{Test}}$, using the exponential map given in Equation 7. Hence, an interpolation on the tangent plane to Grassmann manifold provides a basis of the same dimension (i.e., $[\mathbf{\Phi}_{\text{Test}}] \in \mathcal{G}(R, N)$). Moreover, it preserves the orthonormality of the basis (i.e., the columns of $\mathbf{\Phi}_{\text{Test}}$ are orthonormal to each other). Those properties are not guaranteed if standard interpolation techniques are used directly to inpterpolate basis. The procedure for Grassmann manifold interpolation is summarized in Algorithm 1.

Algorithm 1 Grassmann manifold interpolation

1: Given a set of basis functions $\mathbf{\Phi}_1, \mathbf{\Phi}_2, \ldots, \mathbf{\Phi}_P$ corresponding to the offline simulations parameterized by $\mu_1, \mu_2, \ldots, \mu_P$.
2: Select a point $[\Phi_{\text{Ref}}] \leftarrow [\Phi_i] \in \{[\Phi_1], \ldots, [\Phi_{N_p}]\}$ corresponding to the basis function set $\mathbf{\Phi}_{\text{Ref}} \leftarrow \mathbf{\Phi}_i \in [\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_{N_p}]$ as the reference point.
3: Map each point $[\Phi_i] \in \mathcal{G}(N, R)$ to $[\mathbf{\Gamma}_i] \in \mathcal{T}([\mathbf{\Phi}_{\text{Ref}}])$ using logarithmic map

$$(\mathbf{\Phi}_i - \mathbf{\Phi}_{\text{Ref}}\mathbf{\Phi}_{\text{Ref}}^T\mathbf{\Phi}_i)(\mathbf{\Phi}_{\text{Ref}}^T\mathbf{\Phi}_i)^{-1} = \mathbf{U}_i\mathbf{\Sigma}_i\mathbf{V}_i^T, \tag{10}$$

$$\mathbf{\Gamma}_i = \mathbf{U}_i\tan^{-1}(\mathbf{\Sigma}_i)\mathbf{V}_i^T. \tag{11}$$

4: Construct matrix $\mathbf{\Gamma}_{\text{Test}}$ corresponding to the test parameter μ_{Test} using Lagrange interpolation of matrices $\mathbf{\Gamma}_i$, corresponding to $\mu_1, \ldots, \mu_{N_p}$

$$\mathbf{\Gamma}_{\text{Test}} = \sum_{i=1}^{N_p} \left(\prod_{\substack{j=1 \\ j \neq i}}^{P} \frac{\mu_{\text{Test}} - \mu_j}{\mu_i - \mu_j} \right) \mathbf{\Gamma}_i. \tag{12}$$

5: Compute the POD basis functions $\mathbf{\Phi}_{\text{Test}}$ corresponding to the test parameter μ_{Test} using the exponential map

$$\mathbf{\Gamma}_{\text{Test}} = \mathbf{U}_{\text{Test}}\mathbf{\Sigma}_{\text{Test}}\mathbf{V}_{\text{Test}}^T, \tag{13}$$

$$\mathbf{\Phi}_{\text{Test}} = \left(\mathbf{\Phi}_{\text{Ref}}\mathbf{V}_{\text{Test}}\cos(\mathbf{\Sigma}_{\text{Test}}) + \mathbf{U}_{\text{Test}}\sin(\mathbf{\Sigma}_{\text{Test}}) \right)\mathbf{V}_{\text{Test}}^T, \tag{14}$$

where the trigonometric operators apply only to the diagonal elements.

---

4 Galerkin Projection

To emulate the system's dynamics in ROM context, a Galerkin projection is usually performed. In Galerkin projection-based ROM (GROM), the solution $\mathbf{u}(\mathbf{x}, t_n)$ is approximated by $\hat{\mathbf{u}}(\mathbf{x}, t_n)$. First, $\hat{\mathbf{u}}(\mathbf{x}, t_n)$ is constrained to lie in a trial subspace $\mathcal{S}$ spanned by the basis $\mathbf{\Phi}$. In our study, this basis is computed using the POD method presented in Section 2. Then, the full-order operators are projected onto the same subspace $\mathcal{S}$. In other words, the residual of the governing ODE is enforced to be orthogonal to $\mathcal{S}$. Galerkin projection can be viewed as a special case of Petrov-Galerkin method [80–83] considering the same trial subspace as a test subspace. In the following, we present the governing equations for our test cases, namely 1D Burgers equation and 2D Navier-Stokes equations as well as their low-order approximations.

4.1 1D Burgers Equation

The one-dimensional (1D) viscous Burgers equation represents a standard benchmark for the analysis of nonlinear advection-diffusion problems in 1D setting with similar quadratic nonlinear interaction and Laplacian dissipation. The evolution of the velocity field $u(x, t)$, in a dimensionless form, is given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{\text{Re}}\frac{\partial^2 u}{\partial x^2}, \tag{15}$$

where Re is the dimensionless Reynolds number, defined as the ratio of inertial effects to viscous effects. Equation 15 can be rewritten as

$$\frac{\partial u}{\partial t} = \frac{1}{\text{Re}} \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x}. \tag{16}$$

Then, the reduced-rank approximation $\hat{u}(x,t) = \sum_{k=1}^{R} \alpha_k(t)\phi_k(x)$ (where ϕ_k are the constructed POD modes and α_k are the corresponding coefficients) is plugged into this equation and an inner product with an arbitrary basis ϕ_k is performed to give the following dynamical ODE which is the GROM for the Burgers equation

$$\frac{\mathrm{d}\alpha_k}{\mathrm{d}t} = \sum_{i=1}^{R} \mathfrak{L}_{i,k}\alpha_i + \sum_{i=1}^{R}\sum_{j=1}^{R} \mathfrak{N}_{i,j,k}\alpha_i\alpha_j, \qquad k = 1, 2, \ldots, R, \tag{17}$$

where $\mathfrak{L}$ and $\mathfrak{N}$ are the matrix and tensor of predetermined model coefficients corresponding to linear and nonlinear terms, respectively. They are precomputed as

$$\mathfrak{L}_{i,k} = \Big\langle \frac{1}{\text{Re}} \frac{\partial^2 \phi_i}{\partial x^2}; \phi_k \Big\rangle,$$

$$\mathfrak{N}_{i,j,k} = \Big\langle -\phi_i \frac{\partial \phi_j}{\partial x}; \phi_k \Big\rangle,$$

where the angle-parentheses refer to the Euclidean inner product defined as $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y} = \sum_{i=1}^{N} x_i y_i$.

4.2 2D Navier-Stokes Equations

The vorticity-streamfunction formulation of the two-dimensional (2D) Navier-Stokes equations can be written as [84]

$$\frac{\partial \omega}{\partial t} + J(\omega, \psi) = \frac{1}{\text{Re}} \nabla^2 \omega, \tag{18}$$

where ω is the vorticity and ψ is the streamfunction. The vorticity-streamfunction formulation prevents the odd-even decoupling issues that might arise between pressure and velocity components and enforces the incompressibility condition. The kinematic relationship between vorticity and streamfunction is given by the following Poisson equation,

$$\nabla^2 \psi = -\omega. \tag{19}$$

Equation 18 and Equation 19 include two operators, the Jacobian ($J(f,g)$) and the Laplacian ($\nabla^2 f$) defined as

$$J(f,g) = \frac{\partial f}{\partial x}\frac{\partial g}{\partial y} - \frac{\partial f}{\partial y}\frac{\partial g}{\partial x}, \tag{20}$$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \tag{21}$$

Similar to the 1D Burgers problem, Equation 18 can be rearranged as

$$\frac{\partial \omega}{\partial t} = \frac{1}{\text{Re}} \nabla^2 \omega - J(\omega, \psi). \tag{22}$$

The reduced-rank approximations of the vorticity and streamfunction fields can be written as follows

$$\omega(x, y, t) = \sum_{k=1}^{R} \alpha_k(t)\phi_k^{\omega}(x, y), \tag{23}$$

$$\psi(x, y, t_n) = \sum_{k=1}^{R} \alpha_k(t)\phi_k^{\psi}(x, y). \tag{24}$$

We note that the vorticity and streamfunction share the same time-dependent coefficients ($\alpha_k(t)$) since they are related through the kinematic relationship given by Equation 19 (i.e., streamfunction is not a prognostic variable). Moreover, as POD preserves linear properties, the spatial POD modes for streamfunction can be obtained from the vorticity modes by solving the following Poisson equations

$$\nabla^2 \phi_k^{\psi}(x, y) = -\phi_k^{\omega}(x, y), \quad k = 1, 2, \ldots, R. \tag{25}$$

5

The GROM for the 2D Navier-Stokes equations is given by the same ODE (Equation 17) with the following coefficients

$$\mathfrak{L}_{i,k} = \big\langle \frac{1}{\text{Re}} \nabla^2 \phi_i^\omega ; \phi_k^\omega \big\rangle,$$

$$\mathfrak{N}_{i,j,k} = \big\langle - J(\phi_i^\omega, \phi_j^\psi); \phi_k^\omega \big\rangle.$$

Due to the modal truncation and inherent nonlinearity in Equation 17, GROM no longer represents the same system (i.e., it solves a different problem). As a result, the obtained trajectory from solving the ROM deviates from the projected trajectory, as shown in Figure 1. Therefore, the optimality of the POD basis is lost. Moreover, due to the triadic nonlinear interactions, instabilities can occur in GROMs. To mitigate these problems, closure and/or stabilization techniques are usually required to obtain accurate results. Increasing the ROM dimension can improve the results. However, due to the nonlinearity of the resulting ROM, the computational cost of GROM is $O(R^3)$, which severely constrains the ROM dimension used in practical applications.
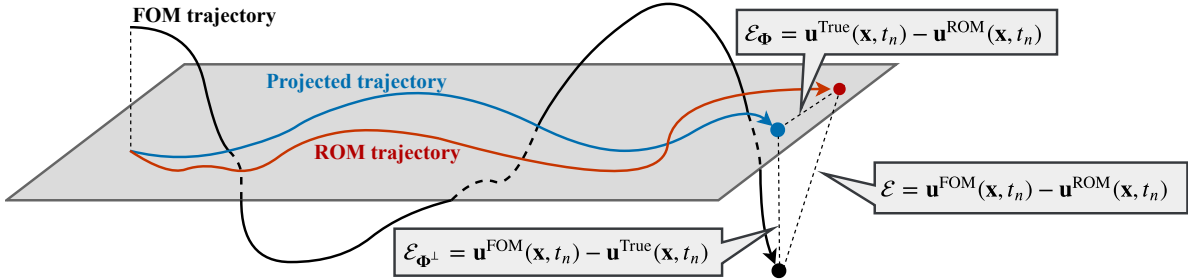
Figure 1: A representation of error sources in ROM (e.g., see [70, 85] for further details).

We note here that we are adopting the tensorial GROM approach [86], where the coefficients $\mathfrak{L}$ and $\mathfrak{N}$ are computed offline. Other approaches can include online computations while incorporating the empirical interpolation method (EIM) [87] or its discrete version, the discrete empirical interpolation method (DEIM) [88] to reduce the online computational cost.

5 Uplifted Reduced Order Modeling

As noted in Section 4, the computational cost of GROM is $O(R^3)$, which limits the number of modes to be used in the ROM. This modal truncation has two major consequences. First, the flow field variable is constrained to lie in a small subspace, spanned by the very first few modes. For convection-dominated flows or parametric problems characterized by slow decay of the Kolmogorov n-width, these few modes may be less representative of the true physical system. This significantly reduces the accuracy of the resulting ROM. This is shown as the projection error $\mathcal{E}_{\Phi^\perp}$ in Figure 1, since the truncated modes are orthogonal to the subspace spanned by Φ. Second, due to the inherent nonlinearity, the truncated modes (or scales) interact with the retained ones. Thus, this truncation simply ignores these interactions, often giving rise to numerical instabilities of solution. This error is represented as $\mathcal{E}_\Phi$ in Figure 1 since it lies in the same subspace Φ. In our uplifted reduced order modeling (UROM) framework (presented in Figure 2), we try to address these two problems.

We extend our reduced-order approximation $\hat{\mathbf{u}}$ to include the first Q modes, where $Q > R$, assuming that the first R modes account for the resolved large scales, and the next $(Q - R)$ modes represent the resolved small scales (while the remaining truncated modes account for the unresolved scales). So, the UROM approximation $\hat{\mathbf{u}}(\mathbf{x}, t_n)$ can be expanded as

$$\hat{\mathbf{u}}(\mathbf{x}, t_n) = \underbrace{\sum_{k=1}^{R} a_k(t) \phi_k(\mathbf{x})}_{\substack{\text{core} \\ \text{(resolved large scales)}}} + \underbrace{\sum_{k=R+1}^{Q} a_k(t) \phi_k(\mathbf{x})}_{\substack{\text{uplift} \\ \text{(resolved small scales)}}}, \tag{26}$$

where $\mathbf{u}$ is a general notation for the flow field of interest, ϕ_k denotes the POD modes, and a_k are the corresponding temporal coefficients. Then, standard Galerkin projection is applied to resolve the large scales (first R modes) to obtain

a predictor $\{\hat{a}_k\}_{k=1}^{R}$ as

$$\frac{\mathrm{d}\hat{a}_k}{\mathrm{d}t} = G_k, \tag{27}$$

$$\hat{a}_k(t_{n+1}) = a_k(t_n) + \Delta t \sum_{q=0}^{s} \beta_q G_k(t_{n-q}), \tag{28}$$

where s and β_q depend upon the numerical scheme used for the time integration. In the present study, we use the third-order Adams-Bashforth (AB3) method for which $s = 2$, $\beta_0 = 23/12$, $\beta_1 = -16/12$, and $\beta_2 = 5/12$. G_k is obtained by Galerkin projection (discussed in Section 4) as

$$G_k(t_n) = \sum_{i=1}^{R} \mathfrak{L}_{i,k} a_i(t_n) + \sum_{i=1}^{R} \sum_{j=1}^{R} \mathfrak{N}_{i,j,k} a_i(t_n) a_j(t_n). \tag{29}$$

In order to correct GROM results for the first R modes, closure and/or stabilization are required. In our framework, we propose the use of LSTM architecture to learn a correction term to steer the GROM prediction of the modal coefficients $\{\hat{a}_k(t_n)\}_{k=1}^{R}$ to the true values $\{a_k(t_n)\}_{k=1}^{R}$ at each timestep. In other words, an LSTM is trained to learn the map from $\{\hat{a}_k(t_n)\}_{k=1}^{R}$ as input to $\{c_k(t_n)\}_{k=1}^{R}$ as output, where c is a correction (closure) term defined as

$$c_k(t_n) = a_k(t_n) - \hat{a}_k(t_n). \tag{30}$$

It should be noted here that the introduced data driven closure takes into account the interactions of *all* the fine scales ($k = R+1, \ldots, N_s$) with the resolved large scales ($k = 1, \ldots, R$), as manifested in the data snapshots. Finally, to account for small scales, we train a second super-resolution LSTM neural network to predict the modal coefficients of the next ($Q - R$) modes, where the input of the LSTM is $\{a_k(t_n)\}_{k=1}^{R}$ and the output is $\{a_k(t_n)\}_{k=R+1}^{Q}$. To improve parametric performance of the UROM architecture and promote generality, the LSTMs' inputs are augmented with the control parameter. Therefore, the LSTM maps f and g corresponding to the closure and super-resolution models, respectively, can be written as

$$f : \begin{bmatrix} \mu \\ \hat{a}_1(t_n) \\ \vdots \\ \hat{a}_R(t_n) \end{bmatrix} \mapsto \begin{bmatrix} c_1(t_n) \\ \vdots \\ c_R(t_n) \end{bmatrix}, \qquad g : \begin{bmatrix} \mu \\ a_1(t_n) \\ \vdots \\ a_R(t_n) \end{bmatrix} \mapsto \begin{bmatrix} a_{R+1}(t_n) \\ \vdots \\ a_Q(t_n) \end{bmatrix}. \tag{31}$$

In brief, we first steer the red line in Figure 1 to the blue one (i.e., introduce data-driven closure by LSTM). Then, we reduce the projection error (difference between the the blue and black lines) by expanding our solution subspace to span Q modes rather than only R. Figure 2 demonstrates the building blocks and workflow of our proposed UROM framework in both the offline and online phases. Few merits of the proposed UROM approach can be listed as follows.

- The physics-constrained GROM is maintained to account for the large scales. This enriches the framework interpretability and generalizability across a wide range of control parameters.
- GROM acts on a few modes, minimizing the online computational cost (i.e., $O(R^3)$, where $R < Q$).
- GROM, being physics-informed, can be used as a sanity check to decide whether or not the LSTM predictions should be considered.
- Data-driven closure/correction encapsulates information from *all* interacting modes and mechanisms.
- Since both LSTMs are fed with input from a physics-based approach, UROM can be considered as a way of enforcing physical knowledge to enhance data-driven tools.
- LSTMs' inputs are augmented with the control parameter to provide a more accurate mapping (sometimes also called physics-informed mapping).

5.1 Long Short-Term Memory Embedding

To learn the maps f and g in UROM, we incorporate memory embedding through the use of LSTM architecture. LSTM is a variant of recurrent neural networks capable of learning and predicting the temporal dependencies between the given data sequence based on the input information and previously acquired information. Recurrent neural networks have been used successfully in ROM community to enhance standard projection ROMs [89] and build fully non-intrusive ROMs [90–95]. In the present study, we use LSTMs to augment the standard physics-informed ROM by introducing closure as well as super-resolution data-driven models. We utilize Keras API [96] to build the LSTMs used in our UROM approach. Details about the LSTM architecture can be found in [92, 93]. A summary of the adopted hyperparameters is presented in Table 1. We also found that the constructed neural networks are not very sensitive to hyperparameters. Meanwhile, for optimal hyperparameter selection, different techniques (e.g., gridsearch) can be used to tune them.
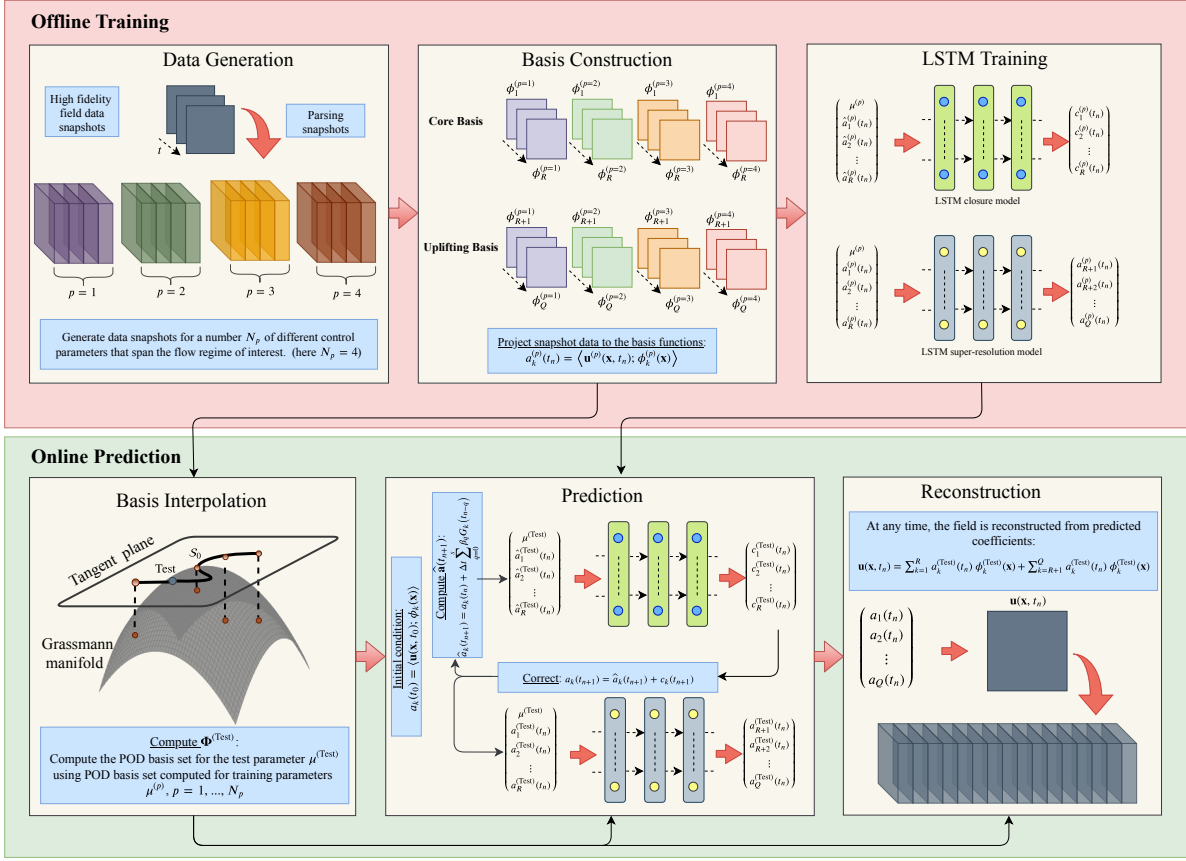
Figure 2: A schematic diagram for the workflow of UROM framework.

6 Results

In order to demonstrate the features and merits of UROM, we present results for the two test cases at out-of-sample control parameters (interpolatory and extrapolatory). For the number of modes, we use $R = 4$ for the core dynamics and $Q = 16$ for super-resolution. We compare the accuracy of UROM prediction with the FOM results as well as the true projection of the FOM snapshots on the POD subspace (denoted as 'True' in our results), where

$$a_k^{\text{True}}(t_n) = \langle \mathbf{u}(\mathbf{x}, t_n); \phi_k(\mathbf{x}) \rangle, \tag{32}$$

$$\mathbf{u}^{\text{True}}(\mathbf{x}, t_n) = \sum_{k=1}^{Q} a_k^{\text{True}}(t_n) \phi_k(\mathbf{x}). \tag{33}$$

Since UROM can be considered as a hybrid approach between fully intrusive and fully non-intrusive ROMs, we compare it with standard Galerkin projection ROM with 4 and 16 modes, denoted as GROM(4) and GROM(16), respectively. Moreover, we show the results of a fully non-intrusive ROM approach using 16 modes (denoted as NIROM). For this NIROM, we use the same LSTM architecture presented in Section 5.1 as a time-stepping integrator. In particular, we learn a map between the values of modal coefficients at current timestep and their values at the following timestep. Also, we augment our input with the control parameter to enhance the mapping accuracy. In other words, the NIROM map h can be represented as follows

$$h : \begin{bmatrix} \mu \\ a_1(t_n) \\ \vdots \\ a_Q(t_n) \end{bmatrix} \mapsto \begin{bmatrix} a_1(t_{n+1}) \\ \vdots \\ a_Q(t_{n+1}) \end{bmatrix}. \tag{34}$$

8

Table 1: A list of hyperparameters utilized to train the LSTM network for all numerical experiments.

Variables	1D Burgers	2D Navier-Stokes
Number of hidden layers	3	3
Number of neurons in each hidden layer	60	80
Number of lookbacks	3	3
Batch size	64	64
Epochs	200	200
Activation functions in the LSTM layers	tanh	tanh
Validation data set	20%	20%
Loss function	MSE	MSE
Optimizer	ADAM	ADAM
Learning rate	0.001	0.001
First moment decay rate	0.9	0.9
Second moment decay rate	0.999	0.999

Finally, we present the CPU time for UROM, GROM(4), GROM(16), and NIROM to demonstrate the computational gain.

6.1 1D Burgers Problem

For 1D Burgers simulation, we consider the initial condition [97]

$$u(x,0) = \frac{x}{1 + \exp\left(\frac{\text{Re}}{16}\left(4x^2 - 1\right)\right)},$$
(35)

with $x \in [0, 1]$. Also, we assume Dirichlet boundary conditions: $u(0, t) = u(1, t) = 0$. The 1D Burgers equation with the above initial and boundary conditions has the following analytic solution representing a traveling wave [97]

$$u(x,t) = \frac{\frac{x}{t+1}}{1 + \sqrt{\frac{t+1}{t_0}}\exp\left(\text{Re}\frac{x^2}{4t+4}\right)},$$
(36)

where $t_0 = \exp(\text{Re}/8)$. For offline training, we obtain solution for different Reynolds numbers (Re $\in \{200, 400, 600, 800\}$). For each case, we collect 1000 snapshots for $t \in [0, 1]$ (i.e., $\Delta t = 0.001$). For online deployment, we obtain the POD basis at Re $= 500$ and Re $= 1000$ using Grassmann manifold interpolation as discussed in Section 3.

6.1.1 Re = 500: demonstrating interpolatory capability

A Reynolds number of 500 represents an interpolatory case, where we use the POD basis at Re $= 600$ as our reference point for basis interpolation. The evolution of the first 4 POD temporal coefficients using different frameworks are shown in Figure 3. It is clear that GROM(4) is incapable of capturing the true dynamics due to the severe modal truncation. On the other hand, both GROM(16) and UROM show very good results; however, GROM(16) is more computationally expensive as will be shown in Section 6.3.

For field reconstruction, we present the temporal field evolution in Figure 4 for FOM snapshots, true projection, UROM, GROM, and NIROM. It can be seen that UROM gives very good predictions for field reconstruction compared with GROM(4) and NIROM, which yield less accurate results. For better visualizations, we show the final field (i.e., at $t = 1$) in Figure 5 with a close-up view on the region characterizing the wave-shock.

6.1.2 Re = 1000: demonstrating extrapolatory capability

In order to investigate the exptrapolatory performance of UROM, we test the approach at Re $= 1000$, with the basis at Re $= 800$ as reference point for interpolation. The POD modal coefficients are shown in Figure 6, where we can see that both UROM and GROM(16) are still capable to capture the true projected trajectory. Interestingly, the NIROM predictions are less satisfactory, giving non-physical behavior at some time instants. This suggests that the physical core of UROM promotes its generality, compared to the totally data-driven NIROM approach. However, we note that the deficient behavior of NIROM can be partly due to the sub-optimal architecture for our network as we only use the
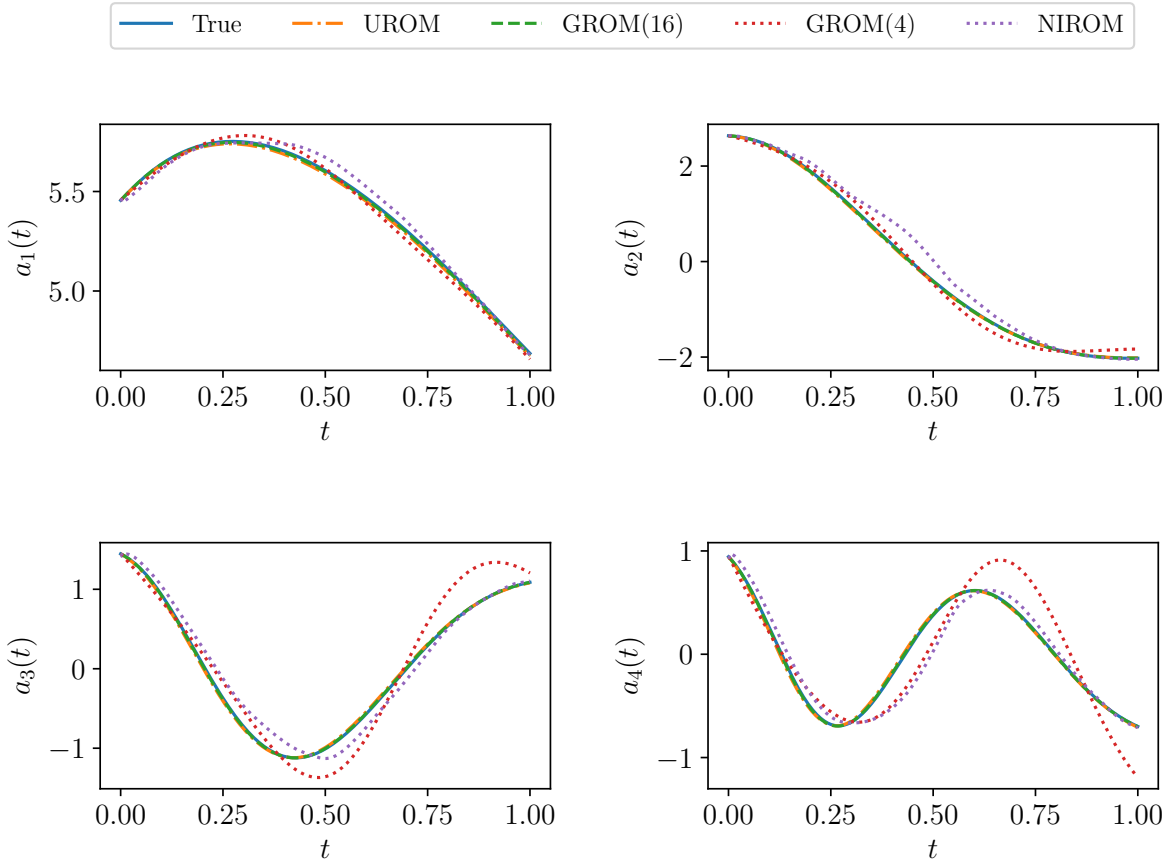
9

Figure 3: Temporal evolution of the first 4 POD modal coefficients for Burgers problem as predicted by UROM, GROM(4), GROM(16), and NIROM compared with the true values obtained by projection of FOM field on the interpolated modes at Re = 500. Note that GROM(4) and NIROM yield inaccurate results.

same hyperparameters (except for the size of input and output layers) for all simulations (as given in Table 1). More sophisticated architectures and further tuning of hyperparameters would probably improve NIROM predictions.

The temporal field evolution of flow field is shown in Figure 7, which illustrates the non-physical and unstable behavior of both GROM(4) and NIROM approaches. The final field is plotted in Figure 8 with a close-up view at the right. It can be seen that even the true projected fields do not match the FOM and show some fluctuations at the shock region. For this type of behavior, a larger subspace is required to capture most of the dynamics of the flow at Re = 1000. Alternatively, a principal interval decomposition approach can be adopted to localize POD basis and tailor more representative compact subspaces [93, 98].

6.2 2D Vortex Merger Problem

As an application for 2D Navier-Stokes equations, we examine the vortex merger problem (i.e., the merging of co-rotating vortex pair) [99]. 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 [100]. We consider an initial vorticity field of two Gaussian-distributed vortices with a unit circulation as follows,

$$
\omega(x, y, 0) = \exp\left(-\rho\left[(x - x_1)^2 + (y - y_1)^2\right]\right) \\
+ \exp\left(-\rho\left[(x - x_2)^2 + (y - y_2)^2\right]\right),
\tag{37}
$$

where ρ is an interacting constant set as $\rho = \pi$ and the vortices centers are initially located at $(x_1, y_1) = \left(\dfrac{3\pi}{4}, \pi\right)$ and $(x_2, y_2) = \left(\dfrac{5\pi}{4}, \pi\right)$. We use a Cartesian domain $(x, y) \in [0, 2\pi] \times [0, 2\pi]$ over a spatial grid of size 256^2, with
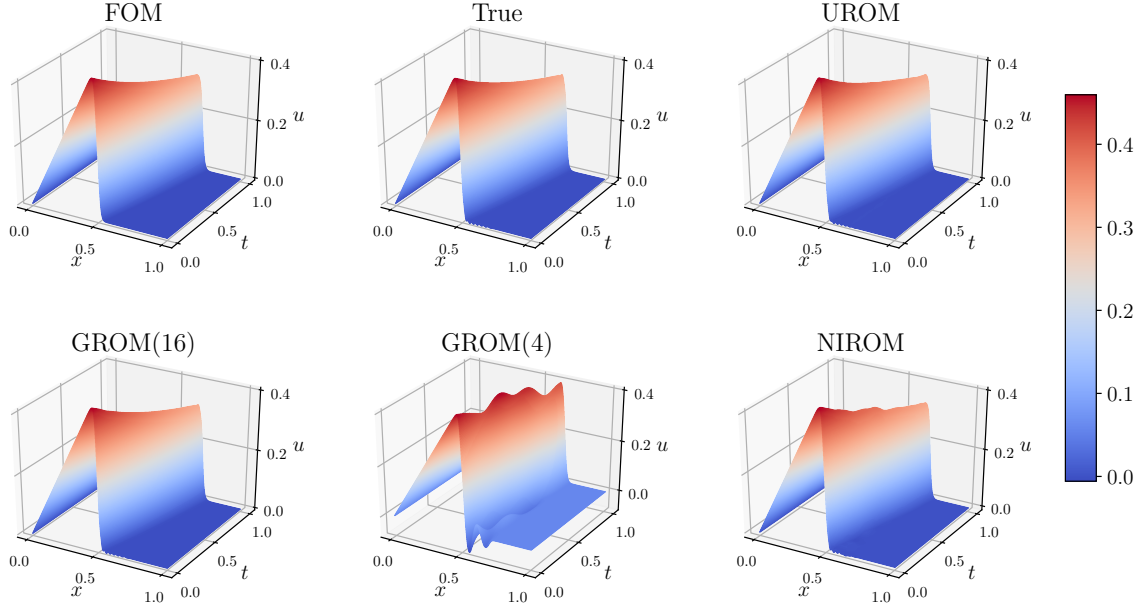
10

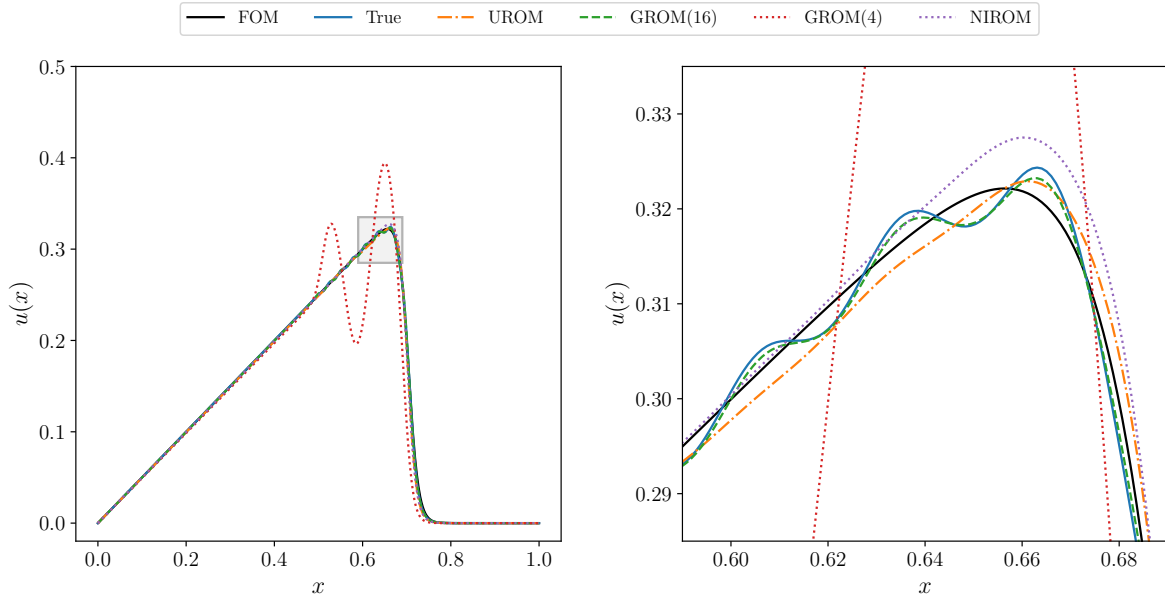Figure 4: Temporal evolution of velocity fields for Burgers problem at Re $= 500$ with $R = 4$ and $Q = 16$.

Figure 5: Final velocity fields (at $t = 1$) for Burgers problem at Re $= 500$ with a zoom-in view at the right using $R = 4$ and $Q = 16$. Note that UROM is giving smooth predictions while GROM(4) is showing significant oscillations.

periodic boundary conditions. For this 2D problem, we collect 200 snapshots for $t \in [0, 20]$, while varying Reynolds number as Re $\in \{200, 400, 600, 800\}$. Similar to the 1D Burgers problem, we test our framework at Re $= 500$ and Re $= 1000$. Also, for basis interpolations, we used reference points at Re $= 600$ and Re $= 800$, respectively.

6.2.1 Re = 500: demonstrating interpolatory capability

Figure 9 shows the temporal evolution of the first 4 POD coefficients for the vorticity field. Recall that the temporal coefficients for vorticity and streamfunction fields are the same, as discussed in Section 4.2. We can see that both UROM and GROM(16) accurately predict the true modal dynamics. For better visualizations, the final vorticity field at

11
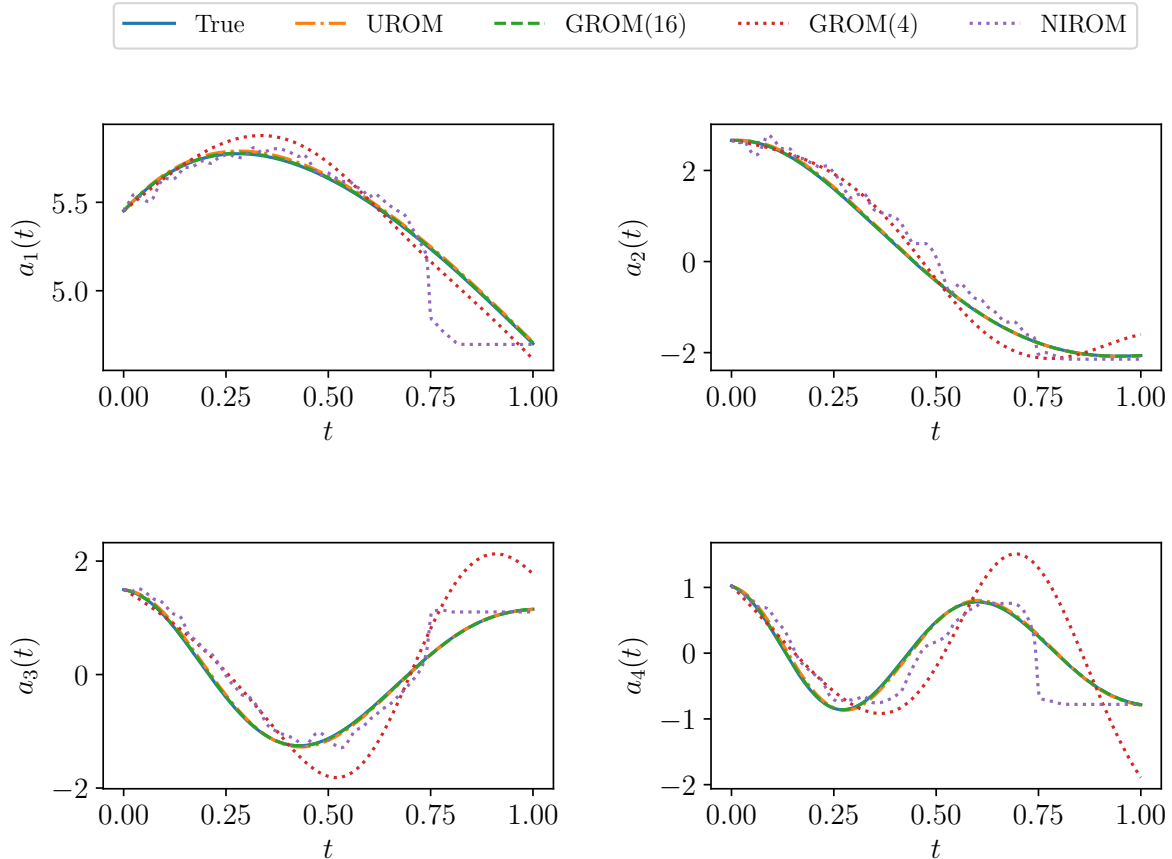
Figure 6: Temporal evolution of the first 4 POD modal coefficients for Burgers problem as predicted by UROM, GROM(4), GROM(16), and NIROM compared with the true values obtained by projection of FOM field on the interpolated modes at Re = 1000. GROM(4) deviates from true trajectory while NIROM gives non-physical predictions.

$t = 20$ is given in Figure 10, where GROM(4) is showing instabilities manifested in the reconstructed field. For this interpolatory case, NIROM is providing "acceptable" results.

6.2.2 Re = 1000: demonstrating extrapolatory capability

The investigated approaches, namely GROM, UROM and NIROM, are tested at Re = 1000 as a case that is out-of-range compared to the training set. The POD modal coefficients predicted by these approaches are given in Figure 11. It can be easily seen that as time increases, the predictions of GROM(4) and NIROM become poor. Using a neural network as time-stepping integrator in NIROM increases its sensitivity to computational noise and this recursive deployment accumulates the error until predictions totally depart from the true trajectory. This is even clearer in the reconstructed field shown in Figure 12, where the orientation of the merging vortices is not matching the true orientation. GROM(4) prediction is also suffering from severe deformation of the true flow topology. On the other hand, the field reconstruction via UROM is accurate compared to the true projection and GROM(16).

6.3 Computing Time

Finally, we report the "online" computing time for the investigated approaches. In particular, we show the computational time as well as the root mean squared error (RMSE) of reconstructed fields at final time for the two test cases at Re = 1000 in Table 2. The reported RMSE is computed as

$$\text{RMSE}(t) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{u}^{FOM}(\mathbf{x}, t) - \hat{\mathbf{u}}(\mathbf{x}, t) \right)^2}, \tag{38}$$
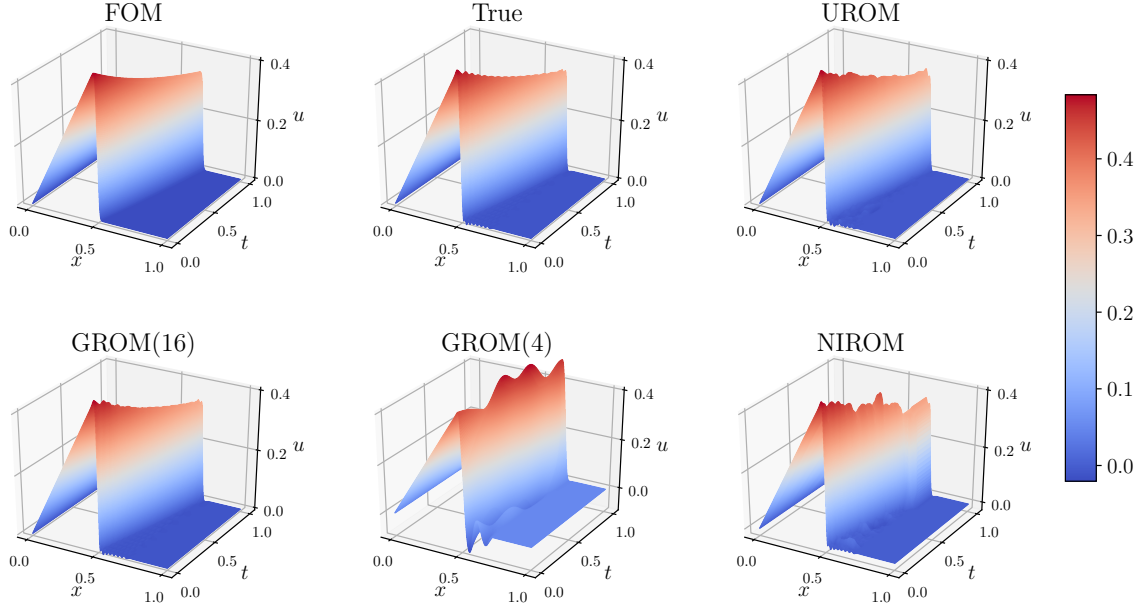
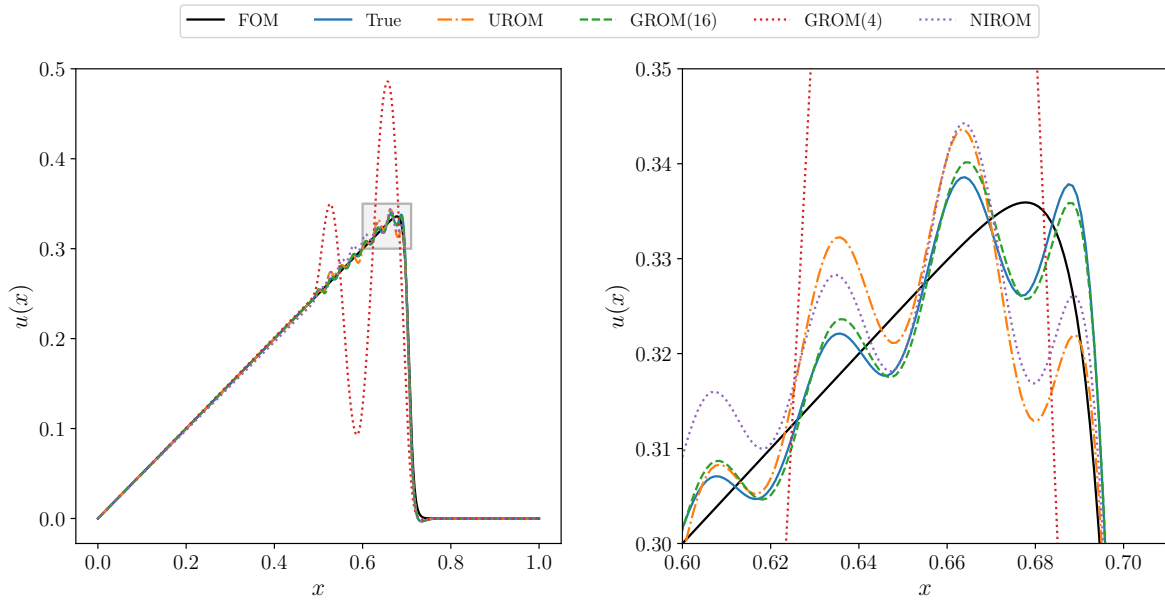Figure 7: Temporal evolution of velocity fields for Burgers problem at Re = 1000 with $R = 4$ and $Q = 16$.

Figure 8: Final velocity fields (at $t = 1$) for Burgers problem at Re = 1000 with a zoom-in view at the right using $R = 4$, and $Q = 16$. Oscillations in UROM and GROM(16) occur mainly because a subspace spanning the first 16 modes is insufficient to capture the dynamics at this Reynolds number.

where N represents the spatial resolution (i.e., $N = N_x \times N_y$). In this table, we also report the NIROM results using 4 modes in the input and output layers. Although GROM(4) is the fastest, its predictions are very poor and further corrections and stabilization might be required. Also, a subspace spanned by only the first four POD modes might be insufficient in complex applications. On the other hand, GROM(16) is the slowest. We can also observe that computation time of UROM is close to that of NIROM and much lower than GROM(16). In Figure 13, we present a bar chart for both the computing time and RMSE of reconstructed fields at final time to illustrate the time-accuracy trade-off.
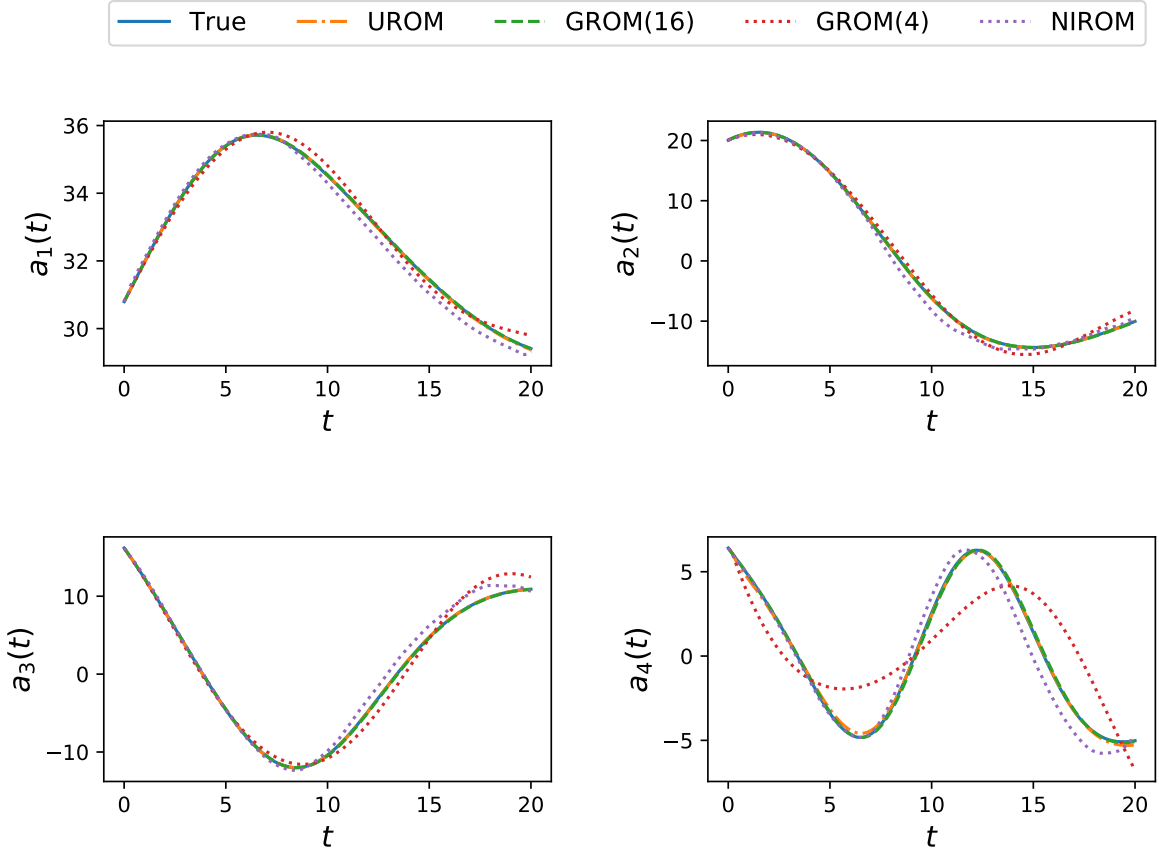
Figure 9: Temporal evolution of the first 4 POD modal coefficients of vorticity field for 2D vortex merger problem as predicted by UROM, GROM(4), GROM(16), and NIROM compared with the true values obtained by projection of FOM field on the interpolated modes at Re = 500.

We note that Table 2 and Figure 13 document the performance of our implementation rather than that of the approaches. We should emphasize that, in this paper, we are not aiming at benchmarking the computational performance of these approaches. Instead, our main objective is to demonstrate the feasibility of hybrid approaches fusing physics-based and machine learning models. Nonetheless, the runtimes in Table 2 indicate that GROM approximately (not exactly due to various other loading/writing abstractions in our Python implementations) scales with R^3. Therefore, combining NIROM and GROM, UROM yields better computational performance. We also note that, if written more optimally, we would also expect that the execution time of UROM (with 16 modes) can be reduced to the sum of the execution times of GROM (with 4 modes) and NIROM (with 16 modes). Indeed, we remark that the second LSTM in UROM (representing the map g) need not be used at all times and can be deployed only at the instant of interest. In that case, the UROM computing times become 1.31 s for Burgers case and 0.30 s for vortex merger (whic are very close to NIROM computing time).

Table 2: Computing time (in seconds) and RMSE of UROM, GROM(4), GROM(16), NIROM(4), and NIROM(16) for Re = 1000. We note that the computing time assessments documented in this table are based on Python executions.

Framework	1D Burgers		2D vortex-merger	
	time(s)	RMSE	time(s)	RMSE
UROM	2.46	$4.13E - 3$	0.54	$5.44E - 3$
GROM(16)	8.40	$3.17E - 3$	1.67	$4.17E - 3$
GROM(4)	0.17	$5.17E - 2$	0.06	$3.99E - 2$
NIROM(16)	1.16	$4.64E - 3$	0.25	$7.80E - 2$
NIROM(4)	1.07	$3.14E - 2$	0.23	$8.58E - 2$

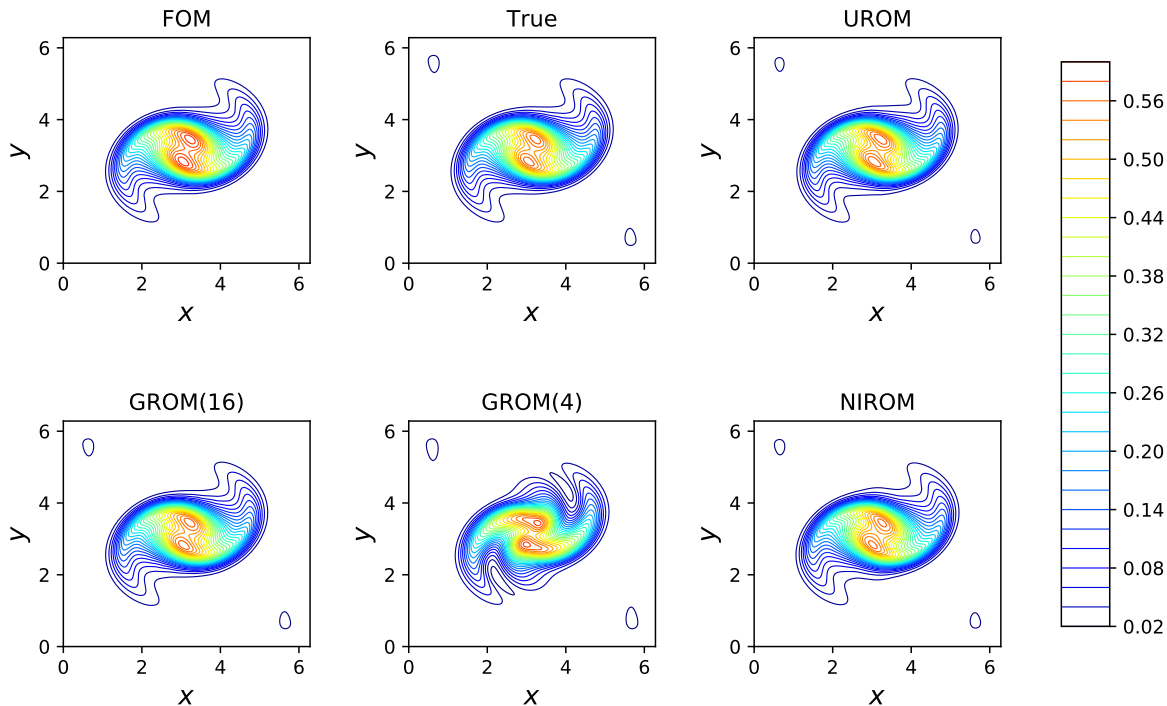Figure 10: Final vorticity fields (at $t = 20$) for 2D vortex merger problem at Re $= 500$ with $R = 4$, and $Q = 16$.

7 Concluding Remarks

In the present study, we have proposed an uplifted reduced order modeling (UROM) approach to elevate the standard Galerkin projection reduced order modeling (GROM). This approach can be considered as a hybrid approach between physics-based and purely data-driven techniques. With GROM at the core of the framework, UROM (with three modeling layers) enhances the model generalizability and interpretability. Moreover, large scales (represented by the first few modes) are given due attention since they control most of the bulk mass, momentum, and energy transfers. Therefore, two out of a total of three layers in UROM aim at predicting the dynamics of these modes as accurately as possible. Then, an uplifting layer is designed to enhance the prediction resolution (i.e., super-resolution). Performance of UROM has been compared against standard GROM and fully non-intrusive ROM (NIROM) approaches.

Two test cases, representing convection-dominated flows in 1D and 2D settings, have been used to evaluate the UROM. For testing, two out-of-sample control parameters have been used to study the interpolatory and extrapolatory performances. In all cases, UROM showed very good results, compared to GROM and NIROM. In particular, UROM$(4 + 12)$ has been shown to provide more accurate results than both GROM(4) and NIROM. In contrast to NIROM where the deployment is fully data-driven, the LSTMs in UROM take their inputs from a physics-based approach. This can be considered as one way of leveraging physical information and intuition into LSTM. On the other hand, UROM has provided significant speed-ups compared to GROM(16) with comparable accuracy. Although we have presented the results for $Q = 16$, more complex flows can require much larger Q, which makes GROM(Q) unfeasible. Finally, this UROM approach is thought to open new avenues to utilize data-driven tools to enhance existing physical models as well as use physics to inform data-driven approaches to maximize the pros of both approaches and mitigate their cons.
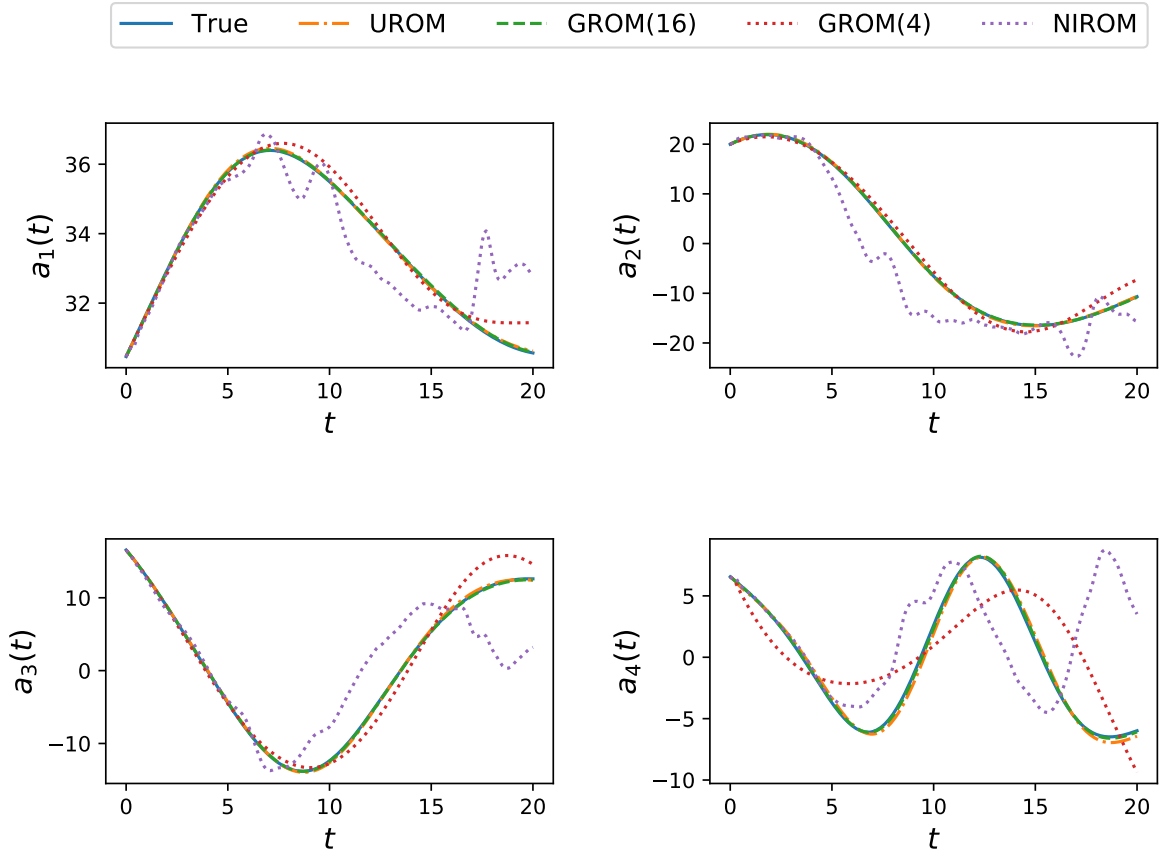
Acknowledgements

Figure 11: Temporal evolution of the first 4 POD modal coefficients of vorticity field for 2D vortex merger problem as predicted by UROM, GROM(4), GROM(16), and NIROM compared with the true values obtained by projection of FOM field on the interpolated modes at Re = 1000.

References

[1] Victor Singh and Karen E Willcox. Engineering design with digital thread. *AIAA Journal*, 56(11):4515–4528, 2018.

[2] Rossella Arcucci, Laetitia Mottet, Christopher Pain, and Yi-Ke Guo. Optimal reduced space for variational data assimilation. *Journal of Computational Physics*, 379:51–69, 2019.

[3] Zhaojun Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied Numerical Mathematics*, 43(1-2):9–44, 2002.

[4] David J Lucia, Philip S Beran, and Walter A Silva. Reduced-order modeling: new approaches for computational physics. *Progress in Aerospace Sciences*, 40(1-2):51–117, 2004.

[5] Martin Hess, Alessandro Alla, Annalisa Quaini, Gianluigi Rozza, and Max Gunzburger. A localized reduced-order modeling approach for PDEs with bifurcating solutions. *Computer Methods in Applied Mechanics and Engineering*, 351:379–403, 2019.

[6] Boris Kramer and Karen E Willcox. Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition. *AIAA Journal*, 57(6):2297–2307, 2019.
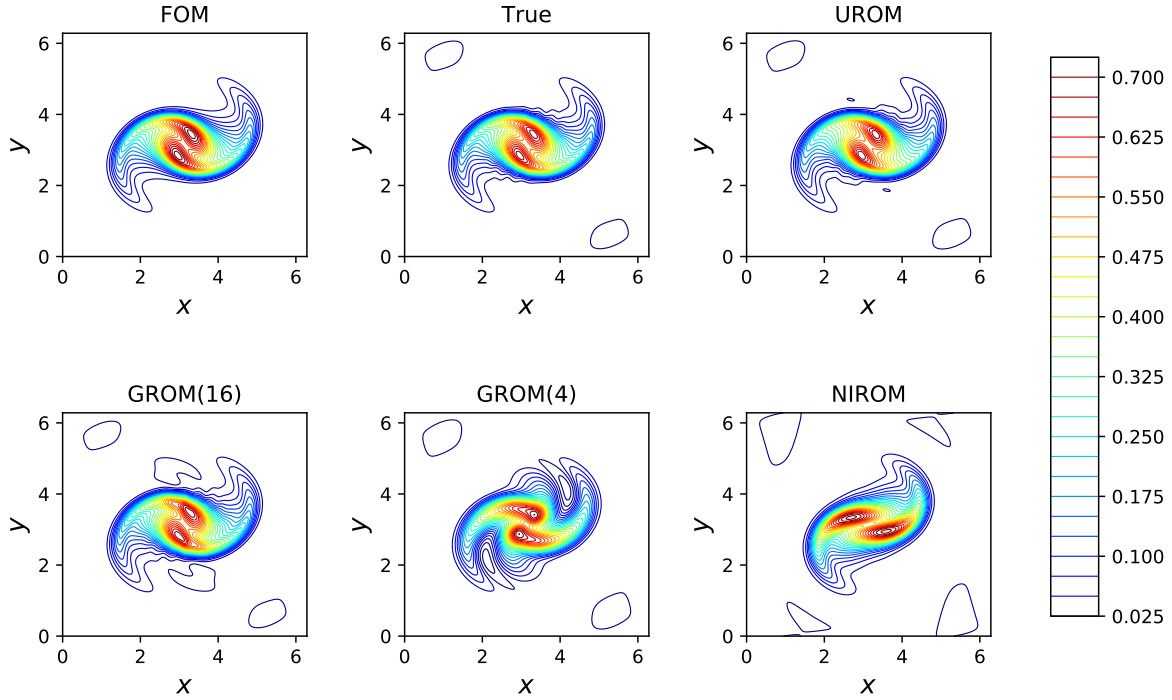
Figure 12: Final vorticity fields (at $t = 20$) for 2D vortex merger problem at Re $= 1000$ with $R = 4$, and $Q = 16$.
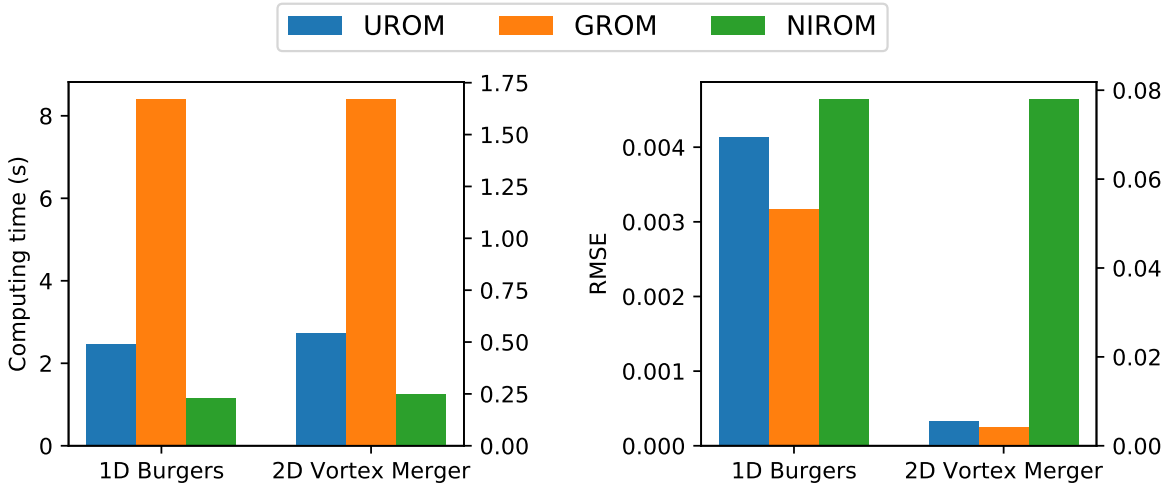

Figure 13: Computing time of testing (online) stage at Re $= 1000$ (left) and RMSE of reconstructed fields at final time for UROM $(4 + 12)$ (i.e., $R = 4$ and $Q = 16$), GROM(16), and NIROM(16).

[7] Renee Swischuk, Laura Mainini, Benjamin Peherstorfer, and Karen Willcox. Projection-based model reduction: Formulations for physics-based machine learning. *Computers & Fluids*, 179:704–717, 2019.

[8] Jake Bouvrie and Boumediene Hamzi. Kernel methods for the approximation of nonlinear systems. *SIAM Journal on Control and Optimization*, 55(4):2460–2492, 2017.

[9] Boumediene Hamzi and Eyad H Abed. Local modal participation analysis of nonlinear systems using Poincaré linearization. *Nonlinear Dynamics*, pages 1–9, 2019.

[10] Milan Korda, Mihai Putinar, and Igor Mezić. Data-driven spectral analysis of the Koopman operator. *Applied and Computational Harmonic Analysis*, page DOI:10.1016/j.acha.2018.08.002, 2018.

[11] Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

[12] Sebastian Peitz, Sina Ober-Blöbaum, and Michael Dellnitz. Multiobjective optimal control methods for the Navier-Stokes equations using reduced order modeling. *Acta Applicandae Mathematicae*, 161(1):171–199, 2019.

[13] Philip Holmes, John L Lumley, Gahl Berkooz, and Clarence W Rowley. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, Cambridge, 2012.

[14] Kunihiko Taira, Steven L Brunton, Scott TM Dawson, Clarence W Rowley, Tim Colonius, Beverley J McKeon, Oliver T Schmidt, Stanislav Gordeyev, Vassilios Theofilis, and Lawrence S Ukeiley. Modal analysis of fluid flows: An overview. *AIAA Journal*, pages 4013–4041, 2017.

[15] Kunihiko Taira, Maziar S Hemati, Steven L Brunton, Yiyang Sun, Karthik Duraisamy, Shervin Bagheri, Scott TM Dawson, and Chi-An Yeh. Modal analysis of fluid flows: Applications and outlook. *AIAA Journal*, pages 1–25, 2019.

[16] Bernd R Noack, Marek Morzynski, and Gilead Tadmor. *Reduced-order modelling for flow control*, volume 528. Springer, Berlin, 2011.

[17] Clarence W Rowley and Scott TM Dawson. Model reduction for flow analysis and control. *Annual Review of Fluid Mechanics*, 49:387–417, 2017.

[18] Nirmal J Nair and Maciej Balajewicz. Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks. *International Journal for Numerical Methods in Engineering*, 117(12):1234–1262, 2019.

[19] Eurika Kaiser, Bernd R Noack, Laurent Cordier, Andreas Spohn, Marc Segond, Markus Abel, Guillaume Daviller, Jan Östh, Siniša Krajnović, and Robert K Niven. Cluster-based reduced-order modelling of a mixing layer. *Journal of Fluid Mechanics*, 754:365–414, 2014.

[20] Bernard Haasdonk, Markus Dihlmann, and Mario Ohlberger. A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space. *Mathematical and Computer Modelling of Dynamical Systems*, 17(4):423–442, 2011.

[21] Markus A Dihlmann and Bernard Haasdonk. Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems. *Computational Optimization and Applications*, 60(3):753–787, 2015.

[22] Kazufumi Ito and Sivaguru S Ravindran. A reduced-order method for simulation and control of fluid flows. *Journal of Computational Physics*, 143(2):403–425, 1998.

[23] Angelo Iollo, Stéphane Lanteri, and J-A Désidéri. Stability properties of POD-Galerkin approximations for the compressible Navier-Stokes equations. *Theoretical and Computational Fluid Dynamics*, 13(6):377–396, 2000.

[24] Clarence W Rowley, Tim Colonius, and Richard M Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1-2):115–129, 2004.

[25] René Milk, Stephan Rave, and Felix Schindler. pyMOR–generic algorithms and interfaces for model order reduction. *SIAM Journal on Scientific Computing*, 38(5):S194–S216, 2016.

[26] Vladimir Puzyrev, Mehdi Ghommem, and Shiv Meka. pyROM: A computational framework for reduced order modeling. *Journal of Computational Science*, 30:157–173, 2019.

[27] Michel Bergmann, C-H Bruneau, and Angelo Iollo. Enablers for robust POD models. *Journal of Computational Physics*, 228(2):516–538, 2009.

[28] M Couplet, C Basdevant, and P Sagaut. Calibrated reduced-order POD-Galerkin system for fluid flow modelling. *Journal of Computational Physics*, 207(1):192–220, 2005.

[29] Karl Kunisch and Stefan Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90(1):117–148, 2001.

[30] A Kolmogoroff. Über die beste Annäherung von Funktionen einer gegebenen Funktionenklasse. *Annals of Mathematics*, 37(1):107–110, 1936.

[31] Allan Pinkus. *N-widths in approximation theory*, volume 7. Springer-Verlag, Berlin, 1985.

[32] Toni Lassila, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Model order reduction in fluid dynamics: challenges and perspectives. In *Reduced Order Methods for Modeling and Computational Reduction*, pages 235–273. Springer, 2014.

[33] D Rempfer. On low-dimensional Galerkin models for fluid flow. *Theoretical and Computational Fluid Dynamics*, 14(2):75–88, 2000.

[34] Bernd R Noack, Konstantin Afanasiev, Marek Morzynski, Gilead Tadmor, and Frank Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.

[35] Sk M Rahman, Shady E Ahmed, and Omer San. A dynamic closure modeling framework for model order reduction of geophysical flows. *Physics of Fluids*, 31(4):046602, 2019.

[36] Sirod Sirisup and George E Karniadakis. A spectral viscosity method for correcting the long-term behavior of POD models. *Journal of Computational Physics*, 194(1):92–116, 2004.

[37] Omer San and Jeff Borggaard. Basis selection and closure for POD models of convection dominated Boussinesq flows. In *21st International Symposium on Mathematical Theory of Networks and Systems*, volume 5, 2014.

[38] Bartosz Protas, Bernd R Noack, and Jan Östh. Optimal nonlinear eddy viscosity in Galerkin models of turbulent flows. *Journal of Fluid Mechanics*, 766:337–367, 2015.

[39] Laurent Cordier, Bernd R Noack, Gilles Tissot, Guillaume Lehnasch, Joel Delville, Maciej Balajewicz, Guillaume Daviller, and Robert K Niven. Identification strategies for model-based control. *Experiments in Fluids*, 54(8):1580, 2013.

[40] Jan Östh, Bernd R Noack, Siniša Krajnović, Diogo Barros, and Jacques Borée. On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-reynolds-number flow over an Ahmed body. *Journal of Fluid Mechanics*, 747:518–544, 2014.

[41] M Couplet, P Sagaut, and C Basdevant. Intermodal energy transfers in a proper orthogonal decomposition–Galerkin representation of a turbulent separated flow. *Journal of Fluid Mechanics*, 491:275–284, 2003.

[42] Virginia L Kalb and Anil E Deane. An intrinsic stabilization scheme for proper orthogonal decomposition based low-dimensional models. *Physics of Fluids*, 19(5):054106, 2007.

[43] I Kalashnikova and MF Barone. On the stability and convergence of a Galerkin reduced order model (ROM) of compressible flow with solid wall and far-field boundary treatment. *International Journal for Numerical Methods in Engineering*, 83(10):1345–1375, 2010.

[44] Xuping Xie, Muhammad Mohebujjaman, Leo G Rebholz, and Traian Iliescu. Data-driven filtered reduced order modeling of fluid flows. *SIAM Journal on Scientific Computing*, 40(3):B834–B857, 2018.

[45] Muhammad Mohebujjaman, Leo G Rebholz, and Traian Iliescu. Physically constrained data-driven correction for reduced-order modeling of fluid flows. *International Journal for Numerical Methods in Fluids*, 89(3):103–122, 2019.

[46] Zhu Wang, Imran Akhtar, Jeff Borggaard, and Traian Iliescu. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Computer Methods in Applied Mechanics and Engineering*, 237:10–26, 2012.

[47] Imran Akhtar, Zhu Wang, Jeff Borggaard, and Traian Iliescu. A new closure strategy for proper orthogonal decomposition reduced-order models. *Journal of Computational and Nonlinear Dynamics*, 7(3):034503, 2012.

[48] Maciej Balajewicz and Earl H Dowell. Stabilization of projection-based reduced order models of the Navier–Stokes. *Nonlinear Dynamics*, 70(2):1619–1632, 2012.

[49] David Amsallem and Charbel Farhat. Stabilization of projection-based reduced-order models. *International Journal for Numerical Methods in Engineering*, 91(4):358–377, 2012.

[50] Omer San and Traian Iliescu. A stabilized proper orthogonal decomposition reduced-order model for large scale quasigeostrophic ocean circulation. *Advances in Computational Mathematics*, 41(5):1289–1319, 2015.

[51] M Gunzburger, T Iliescu, M Mohebujjaman, and M Schneier. An evolve-filter-relax stabilized reduced order stochastic collocation method for the time-dependent Navier–Stokes equations. *SIAM/ASA Journal on Uncertainty Quantification*, 7(4):1162–1184, 2019.

[52] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[53] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12:2451–2471, 1999.

[54] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, 2020.

[55] J Nathan Kutz. Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1–4, 2017.

[56] Paul A Durbin. Some recent developments in turbulence closure modeling. *Annual Review of Fluid Mechanics*, 50:77–103, 2018.

[57] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019.

[58] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

[59] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018.

[60] Samuel E Otto and Clarence W Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.

[61] Kookjin Lee and Kevin T Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, page 108973, 2019.

[62] Kyongmin Yeo and Igor Melnyk. Deep learning algorithm for data-driven simulation of noisy dynamical system. *Journal of Computational Physics*, 376:1212–1231, 2019.

[63] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[64] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[65] Pantelis R Vlachas, Wonmin Byeon, Zhong Y Wan, Themistoklis P Sapsis, and Petros Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2213):20170844, 2018.

[66] Haşim Sak, Andrew Senior, and Françoise 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.

[67] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, 1987.

[68] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1):539–575, 1993.

[69] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, pages 808–817, 2000.

[70] Muruhan Rathinam and Linda R Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2003.

[71] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. SIAM, Philadelphia, 1997.

[72] David Amsallem and Charbel Farhat. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal*, 46(7):1803–1813, 2008.

[73] David Amsallem, Julien Cortial, Kevin Carlberg, and Charbel Farhat. A method for interpolating on manifolds structural dynamics reduced-order models. *International Journal for Numerical Methods in Engineering*, 80(9):1241–1258, 2009.

[74] Ralf Zimmermann, Benjamin Peherstorfer, and Karen Willcox. Geometric subspace updates with applications to online adaptive nonlinear model reduction. *SIAM Journal on Matrix Analysis and Applications*, 39(1):234–261, 2018.

[75] Ralf Zimmermann. Manifold interpolation and model reduction. *arXiv preprint arXiv:1902.06502*, 2019.

[76] Mourad Oulghelou and Cyrille Allery. A fast and robust sub-optimal control approach using reduced order model adaptation techniques. *Applied Mathematics and Computation*, 333:416–434, 2018.

[77] M Oulghelou and C Allery. Non intrusive method for parametric model order reduction using a bi-calibrated interpolation on the Grassmann manifold. *arXiv preprint arXiv:1901.03177*, 2018.

[78] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[79] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Riemannian geometry of grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematica*, 80(2):199–220, 2004.

[80] Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.

[81] Kevin Carlberg, Matthew Barone, and Harbir Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693–734, 2017.

[82] Youngsoo Choi and Kevin Carlberg. Space–time least-squares Petrov–Galerkin projection for nonlinear model reduction. *SIAM Journal on Scientific Computing*, 41(1):A26–A58, 2019.

[83] D Xiao, F Fang, J Du, CC Pain, IM Navon, AG Buchan, Ahmed H Elsheikh, and G Hu. Non-linear Petrov–Galerkin methods for reduced order modelling of the Navier–Stokes equations using a mixed finite element pair. *Computer Methods In Applied Mechanics and Engineering*, 255:147–157, 2013.

[84] M.D. Gunzburger. *Finite element methods for viscous incompressible flows*. Computer Science and Scientific Computing. Academic Press Inc., Boston, MA, 1989. A guide to theory, practice, and algorithms.

[85] David Amsallem. *Interpolation on manifolds of CFD-based fluid and finite element-based structural reduced-order models for on-line aeroelastic predictions*. PhD thesis, Stanford University, 2010.

[86] Răzvan Ştefănescu, Adrian Sandu, and Ionel M Navon. Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations. *International Journal for Numerical Methods in Fluids*, 76(8):497–521, 2014.

[87] 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. *Comptes Rendus Mathematique*, 339:667–672, 2004.

[88] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[89] J Nagoor Kani and Ahmed H Elsheikh. Reduced-order modeling of subsurface multi-phase flow models using deep residual recurrent neural networks. *Transport in Porous Media*, 126(3):713–741, 2019.

[90] Arvind T Mohan and Datta V Gaitonde. A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks. *arXiv preprint arXiv:1804.09269*, 2018.

[91] Zheng Wang, Dunhui Xiao, Fangxin Fang, Rajesh Govindan, Christopher C Pain, and Yike Guo. Model identification of reduced order fluid dynamics systems using deep learning. *International Journal for Numerical Methods in Fluids*, 86(4):255–268, 2018.

[92] Sk. M. Rahman, S. Pawar, O. San, A. Rasheed, and T. Iliescu. Nonintrusive reduced order modeling framework for quasigeostrophic turbulence. *Physical Review E*, 100:053306, 2019.

[93] Shady E Ahmed, Sk Rahman, Omer San, Adil Rasheed, Ionel M Navon, et al. Memory embedded non-intrusive reduced order modeling of non-ergodic flows. *arXiv preprint arXiv:1910.07649*, 2019.

[94] Steffen Wiewel, Moritz Becher, and Nils Thuerey. Latent space physics: Towards learning the temporal evolution of fluid flow. In *Computer Graphics Forum*, volume 38, pages 71–82. Wiley Online Library, 2019.

[95] D Xiao, F Fang, J Zheng, CC Pain, and IM Navon. Machine learning-based rapid response tools for regional air pollution modelling. *Atmospheric Environment*, 199:463–473, 2019.

[96] François Chollet et al. Keras. <https://keras.io>, 2015.

[97] Montri Maleewong and Sirod Sirisup. On-line and off-line POD assisted projective integral for non-linear problems: A case study with Burgers' equation. *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*, 5(7):984–992, 2011.

[98] Markus Dihlmann, Martin Drohmann, and Bernard 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*, pages 156—167, 2011.

[99] James D Buntine and DI Pullin. Merger and cancellation of strained vortices. *Journal of Fluid Mechanics*, 205:263–295, 1989.

[100] J Von Hardenberg, JC McWilliams, A Provenzale, A Shchepetkin, and JB Weiss. Vortex merging in quasi-geostrophic flows. *Journal of Fluid Mechanics*, 412:331–353, 2000.