Commun Nonlinear Sci Numer Simulat 77 (2019) 271–287

Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

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

Research paper

An artificial neural network framework for reduced order modeling of transient flows

Omer San*, Romit Maulik, Mansoor Ahmed

School of Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK 74078, USA

ARTICLE INFO

Article history:
Received 26 July 2017
Revised 22 April 2019
Accepted 24 April 2019
Available online 25 April 2019

Keywords:
Artificial neural networks
Reduced order modeling
Proper orthogonal decomposition
Convective flows
Non-intrusive model order reduction

ABSTRACT

This paper proposes a supervised machine learning framework for the non-intrusive model order reduction of unsteady fluid flows to provide accurate predictions of non-stationary state variables when the control parameter values vary. Our approach utilizes a training process from full-order scale direct numerical simulation data projected on proper orthogonal decomposition (POD) modes to achieve an artificial neural network (ANN) model with reduced memory requirements. This data-driven ANN framework allows for a nonlinear time evolution of the modal coefficients without performing a Galerkin projection. Our POD-ANN framework can thus be considered an equation-free approach for latent space dynamics evolution of nonlinear transient systems and can be applied to a wide range of physical and engineering applications. Within this framework we introduce two architectures, namely sequential network (SN) and residual network (RN), to train the trajectory of modal coefficients. We perform a systematic analysis of the performance of the proposed reduced order modeling approaches on prediction of a nonlinear wave-propagation problem governed by the viscous Burgers equation, a simplified prototype setting for transient flows. We find that the POD-ANN-RN yields stable and accurate results for test problems assessed both within inside and outside of the database range and performs significantly better than the standard intrusive Galerkin projection model. Our results show that the proposed framework provides a non-intrusive alternative to the evolution of transient physics in a POD basis spanned space, and can be used as a robust predictive model order reduction tool for nonlinear dynamical systems.

1. Introduction

The full resolution of the dynamics of engineering flows requires a computational expense that is as yet infeasible for any practical purpose. Due to these overwhelming computational demands, model order reduction type approaches are gaining popularity. They are extensively used for many optimization and control applications [2,14,24,30–32,68,81,87,95]. Among the multitude of reduced order modeling approaches, *proper orthogonal decomposition* (POD) has emerged as a popular technique for the study of dynamical systems [3,4,20,48,52,71,99]. To clarify, POD is also documented under terminology such as the Karhunen-Loève expansion [65], principal component analysis [49] or empirical orthogonal function [66]. POD extracts the most energetic modes from a collection of high fidelity numerical simulations of the governing equations of the dynamic system being studied. These bases are then used to reduce the degrees of freedom of the governing equation to scales that

* Corresponding author.
E-mail address: osan@okstate.edu (O. San).

https://doi.org/10.1016/j.cnsns.2019.04.025
1007-5704/© 2019 Elsevier B.V. All rights reserved.

are computationally tractable. In particular, the fluid mechanics community has traditionally used POD as a method of extraction of the large scale coherent structures (represented by these modes) for the purpose of statistical pattern recognition [70]. This is because the global POD modes optimally span our physical space through a considerably truncated number of bases to resolve attractors and transients well.

The evolution equations for the lower order system are then obtained using the Galerkin projection (GP) method. The POD-GP method has been extensively used to provide fast and accurate simulations of large nonlinear systems [6,7,10,13,56,68,86,99,108]. The Galerkin projection is devised so as to generate a reduced model where the truncated linear coefficients of the POD bases are evolved through time as a set of ordinary differential equations (ODEs). This process is denoted a *projection* since it is obtained by an orthogonal projection of the governing equations onto the truncated POD modes. The result is a considerably truncated system which continues to represent the most significant characteristics of the governing equations. The small memory requirement and low computational expense of this framework make it particularly suitable for post-processing analysis, optimization and control type problems where repeated model evaluations are required over a large range of parameters. This approach is significantly effective for quasi-stationary, time-periodic and decaying problems but might be challenging for highly non-stationary, convective and nonlinear problems.

The drawbacks are generally due to the lack of adaptation of the POD modes as well as the loss of information due to the truncation being limited to the most energetic of modes [21,28]. Amongst the various efforts used to mitigate these drawbacks include modeling the effect of truncated modes on the retained ones, an approach known as closure modeling [11,82,91,105,106,109,115–117], and developing strategies to construct a more representative basis [1,14–16,50,57,89]. On the other hand, the energy balance and mass conservation properties of projection based reduced order models have also been studied for fluid flows [77]. However, the projection-based model reduction approaches have limitations especially for complex systems such as general circulation models, since there is a lack of access to the full-order model (ROM) operators or the complexity of the forward simulation codes that renders the need for obtaining the full-order operators [83]. Therefore, there is a recent interest in generating fully non-intrusive approaches without the need for access to full-order model operators to establish surrogate models [5,19,42,45,74,104,112,112,114].

Alternatively, machine learning offers promise to generate accurate parametric reduced order models. There are a broad range of opportunities for the development of new deep learning architectures for reduced order models and numerical analysis of these machine learning approaches in applications of challenging flows. Recently, a supervised machine learning approach has been used to develop feasible regions in the parameter space where the admissible target accuracy is achieved with a predefined reduced order basis [78]. Neural networks based closure schemes have also been developed by the authors to compute a stabilization term (i.e., eddy viscosity) for Galerkin projection based reduced order models [92–94]. A deep residual recurrent neural network has been introduced as an efficient model reduction technique for nonlinear dynamical systems [51]. In a context similar to this recurrent neural network architecture, we formulate our problem in low-dimensional embedded structures that capture the major portion of the energy of the system in order to take advantage of lower-dimensional coherent structures that are responsible for the bulk mass, momentum, and energy transfer instead of performing a very large-scale simulation that considers all structures. A recent investigation of a spectral POD (SPOD) spanned model order reduction framework for fluid flows has been performed by Lui and Wolf [69] using the deep feedforward neural networks. A long short-term memory (LSTM) architecture based model reduction approach has also been recently introduced to complement an imperfect reduced order model with data-streams utilizing a regularized hybrid framework [102].

The present study aims to bypass the Galerkin projection through the use of a single-layer artificial neural network (ANN) by introducing two different architectures, namely a sequential network (SN) and a residual network (RN). ANNs can be classified under the broader category of machine learning methods (i.e., systems that learn from data) and are a mathematical representation of the biological neural networks found in the human central nervous system. They have been used for a wide variety of applications such as function approximation, classification, data processing and dynamic systems control [110]. Briefly, an ANN can be used to setup a nonlinear relationship between a desired set of inputs and targets provided a large amount of benchmark data for the underlying relationship is available. This fitting to available data (also known as *training*) ideally generates a set of linear combinations of parameters and transfer functions which replicate the mean behavior of the underlying phenomena. This allows for the representation of subtle relationships which cannot be expressed explicitly in a functional form. The interested reader is directed to the excellent introductory text by Demuth et al. [23] on the development of a variety of ANN architectures and their underlying principles. We stress here that generating a robust non-intrusive reduced order modeling (ROM) approach is highly desirable for its ability to preclude model-form uncertainty in our forward simulations for reduced computational degree of freedom and is thus a highly active area of research [8,9,55,61,83,111,113]. In fact, several recent studies have investigated the suitability of ANN variants for fully non-intrusive dynamics capture in latent space [76,102,107].

With regard to the ROM methodology, we implement our framework in a context similar to the works of Narayanan et al. [80] and Khibnik et al. [53] where the POD-GP approach was replaced by an ANN-based evolution of the temporal dynamics of the ROM for the purpose of active separation control in a planar diffuser. Their dataset for ANN training was obtained both from direct numerical simulations of the Navier–Stokes equations as well as experiments. In [88], an ANN was used to devise a feedback control circuit for a transitional flow in a grooved channel. The dataset for ANN training was obtained through a POD-GP simulation itself. Both works demonstrated the unique advantages of the ANN approach, for example, the reduction of a system of governing PDEs to a no-equation dynamic system through data-driven learning. We note that

ANNs also find great utilization in the field of feedback flow control where they are used to generate a direct mapping of flow measurements to actuator control systems [26,27,29,36–38,46,58]. General machine learning based methods (of which ANN is a subset) are also growing in popularity for fluid flow based applications and represent a computationally viable alternative to the full Navier–Stokes equations [12,34,79]. An excellent source of information about some viable machine learning classes can be found in [64]. In turbulence modeling, ANN approaches have also been used in combination with reduced degree-of-freedom implementations of the Navier–Stokes framework, for instance in the quantification of errors in functional closure models [96] and for data-driven closure modeling [33,62,63,73,75,101].

This paper develops a framework for generating an ANN in order to create a very fast integrator for certain types of partial differential equation (PDE) systems (e.g., one-dimensional Burgers equation in our demonstrations). The ANN is created to represent a map for the trajectory of the POD basis coefficients evolving from time n to time $n+1$. The projection of snapshot data is used to *train* the ANN, i.e., make it choose the weights to best fit the data in the POD spanned reduced order space. Therefore, this model reduction approach is denoted as PON-ANN in our study. Using only snapshots of the state variables, our data-driven approach can thus be considered truly non-intrusive, since any prior information about the underlying governing equations is not required for generating the reduced order model. This idea can be considered as a competitor to the POD-GP, and the advantages are that this is an equation-free method, and also it can be more accurate since it can be trained with more data. Although many new model order reduction techniques perform better than standard POD-GP model, as highlighted above, we focus here on comparing the proposed approach with the POD-GP model only. Future comparative studies should be carried out to explore the feasibility of such non-intrusive models. Before proceeding with our discussion, we describe the major questions we wish to answer in this work:

- Can the POD-ANN non-intrusive ROM approach be used for transient dynamical systems such as nonlinear Burgers equation?
- How does an ANN-based evolution of the modal coefficients compare to the POD-GP approach?
- Can a trained POD-ANN (using a particular dataset) be used to interpolate or extrapolate to flow problems with slight differences in temporal evolution and physics (i.e., for different control parameter values such as Reynolds number)?
- Which POD-ANN architecture is more stable and robust in forecasting beyond the training data range?

To answer these questions, we investigate a test case given by the nonlinear viscous Burgers equation to compare both the POD-ANN and POD-GP approaches.

2. Mathematical modeling

The viscous Burgers equation is used in our study as a test case to explain the proposed model reduction framework. In its original form, the Burgers equation can be written as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2}, \quad x \in [0,1], \quad t \in [0,1], \tag{1}$$

where Re is a non-dimensional Reynolds number. We must mention here that this equation is generally considered a framework for preliminary evaluation of numerical methods for analysis of fluid flow applications as it possesses the hallmarks of general nonlinear multidimensional advection-diffusion problems. We consider a convective system in our model order reduction assessments since it characterizes localized flow structures such as shock waves. The above PDE can be solved exactly to obtain an analytical formulation for time evolution of the field variable $u(x, t)$ given by [72]

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

where $t_0 = \exp(\frac{Re}{8})$. This exact expression is used to generate snapshot data for our forthcoming model order reduction analysis. The database is constituted with Eq. (2) using $N_x = 1024$ spatial collocation points at each snapshot. Our database consists of snapshots obtained by using 10 equally spaced values of Reynolds number, i.e., $Re = [100, 200, \ldots, 1000]$. Fig. 1 shows the space-time behavior of the four representative solutions at different Reynolds number.

3. Proper orthogonal decomposition

In this section we explore the POD approach for reduced order modeling to capture unsteady, convective and non-periodic dynamics of the underlying governing partial differential equations. We elaborate the procedure for calculating the orthonormal bases and corresponding coefficients for the Burgers problem and note that an extension to higher dimensions is straightforward. For a detailed discussion on this technique we refer the reader to San and Iliescu [90].

A representative POD basis can be constructed from any arbitrary scalar field variable f at different times (also known as snapshots). These snapshots are generally obtained by solving the governing equations we are attempting to model using a regular full-order model (FOM) approach. The numerical methods used to obtain this field data are explained in Section 6. In the following, we will utilize the index n to indicate a particular snapshot in time. For the POD approach we utilize a total

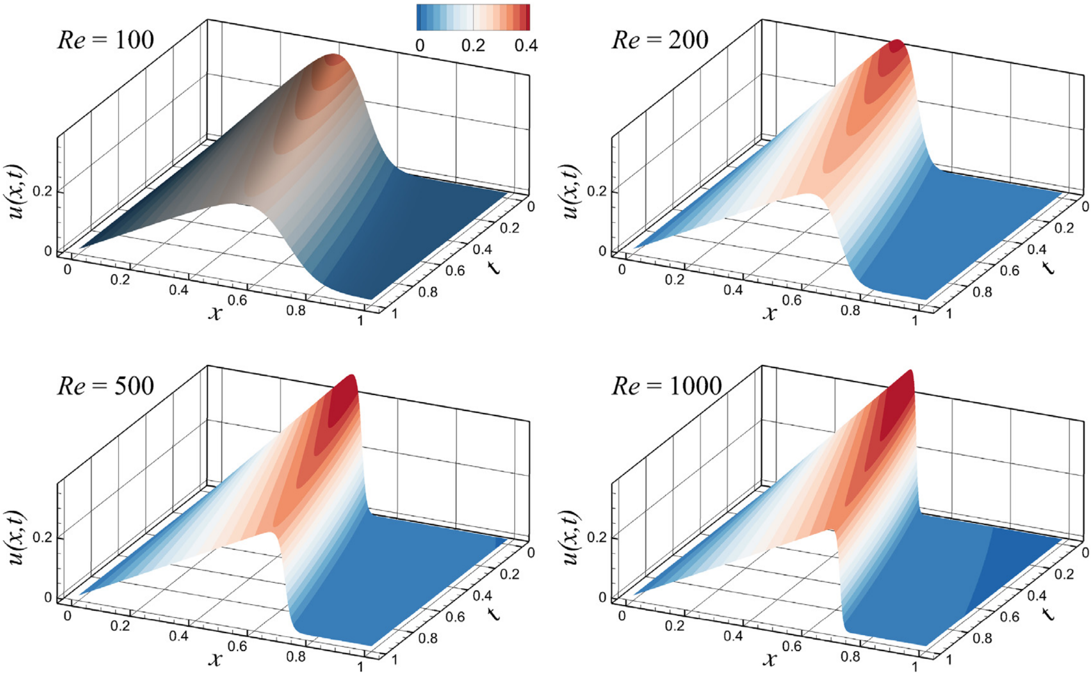

Fig. 1. Space-time solution of the Burgers equation for various Reynolds numbers within our database. Note that both the POD analysis and ANN training consider a database constructed by using 10 equally spaced values of Reynolds number between $Re = 100$ and $Re = 1000$, i.e., $Re = [100, 200, \ldots, 1000]$.

of N snapshots for the field variable, i.e., $f(\mathbf{x}, t^{(n)})$ for $n = 1, 2, \ldots, N$. The flow field data thus obtained is first decomposed into the mean and fluctuating part as follows

$$f(\mathbf{x}, t) = \bar{f}(\mathbf{x}) + f'(\mathbf{x}, t), \qquad \bar{f}(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x}, t^{(n)}), \tag{3}$$

where $\bar{f}$ implies a temporal averaging of a particular point value in the field and f' contains the fluctuating quantity. We must remark here that $\bar{f}$ is a function of space alone whereas f' is a function of both space and time. A correlation matrix may be constructed using the fluctuating components of the snapshots to give

$$C_{ij} = \int_{\Omega} f'(\mathbf{x}, t^{(i)}) f'(\mathbf{x}, t^{(j)}) d\mathbf{x}, \tag{4}$$

where Ω is the entire spatial domain and i and j refer to the ith and jth snapshots. The time correlation data matrix C is a non-negative symmetric square matrix of size $N \times N$. If we define the inner product of any two fields f_1 and f_2 as

$$(f_1, f_2) = \int_{\Omega} f_1(\mathbf{x}) f_2(\mathbf{x}) d\mathbf{x} \tag{5}$$

we may express the correlation matrix as $C_{ij} = \left(f'(\mathbf{x}, t^{(i)}), f'(\mathbf{x}, t^{(j)})\right)$. In this study, we use the well-known Simpson's 1/3 integration rule for a numerical computation of the inner products. The optimal POD basis functions are obtained by performing an eigendecomposition of the C matrix. This has been shown in detail in the POD literature (see, e.g., [48,84,97]). The eigenvalue problem can be written in the following form:

$$CW = W\Lambda, \tag{6}$$

where $\Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \ldots, \lambda_N]$ is a diagonal matrix containing the eigenvalues of this decomposition and W =[$\boldsymbol{w}^1$, $\boldsymbol{w}^2$, ..., $\boldsymbol{w}^N$]. The eigenvalues are stored in descending order, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$. Then the orthogonal POD basis functions can be written as

$$\phi_1(\mathbf{x}) = \sum_{n=1}^{N} w_n^1 f'(\mathbf{x}, t^{(n)}), \quad \phi_2(\mathbf{x}) = \sum_{n=1}^{N} w_n^2 f'(\mathbf{x}, t^{(n)}), \quad \ldots, \quad \phi_N(\mathbf{x}) = \sum_{n=1}^{N} w_n^N f'(\mathbf{x}, t^{(n)}), \tag{7}$$

where w_n^k is the nth component of the kth eigenvector $\boldsymbol{w}^k$. Therefore, we emphasize that the POD modes are ranked according to the magnitude of their eigenvalue. The eigenvectors must also be normalized in order to satisfy the condition of orthonormality between bases:

$$\left(\phi_i, \phi_j\right) = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases} \tag{8}$$

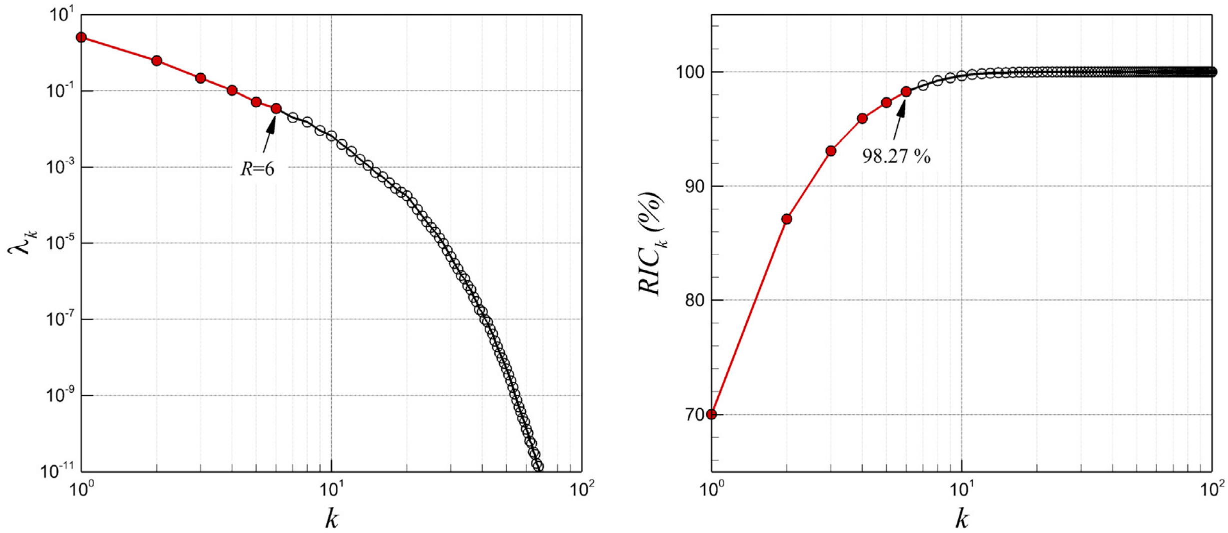

Fig. 2. Eigenvalues of the time correlation matrix (left) and the corresponding relative information content (RIC) index (right) for the space-time solution of the viscous Burgers equation. Note that we retain only the most energetic 6 POD modes in our model order reduction analysis.

Table 1
The relative information content (RIC) index that measures the percentage of captured energy for different number of POD modes.

R	RIC(%)
1	70.0273
2	87.1354
3	93.0921
4	95.9237
5	97.3327
6	98.2723
10	99.6722
20	99.9897

It can be shown that, for Eq. (8) to be true, the eigenvector $\boldsymbol{w}^k$ must satisfy the following equation:

$$\sum_{n=1}^{N} w_n^k w_n^k = \frac{1}{\lambda_k}, \tag{9}$$

where λ_k is the kth eigenvalue associated to the eigenvector $\boldsymbol{w}^k$. In practice, most of the subroutines for solving the eigensystem given in Eq. (6) return the eigenvector matrix W having all the eigenvectors normalized to unity. In that case, the orthogonal POD bases are given by

$$\phi_k(\mathbf{x}) = \frac{1}{\sqrt{\lambda_k}} \sum_{n=1}^{N} w_n^k f'(\mathbf{x}, t^{(n)}) \tag{10}$$

where $\phi_k(\mathbf{x})$ is the kth POD basis function. We remark here that the POD should be carried out independently for each field variable for multidimensional governing equations. We collect 101 snapshots between time $t = 0$ and $t = 1$ at each Reynolds number between $Re = 100$ and $Re = 1000$ using 10 different realizations (i.e., $Re = [100, 200, \ldots, 1000]$). Fig. 2 shows the eigenvalues of the time correlation matrix C and the corresponding relative information content (RIC) index, which is defined as [40,100]

$$RIC_k = \frac{\sum_{n=1}^{k} \lambda_n}{\sum_{n=1}^{N} \lambda_n} \times 100, \tag{11}$$

where the number of snapshots is set to $N = 1010$ in our study. As shown in Table 1, the RIC index measures the percentage of captured energy for different number of POD modes. It is evident that a small number of modes are enough to capture the spatial and temporal dynamics of the system well. Therefore, we present our ROM analyses using $R = 6$ largest (most energetic) POD modes capturing approximately 98.27% of the energy. Fig. 3 illustrates the associated POD basis functions utilized in this study.

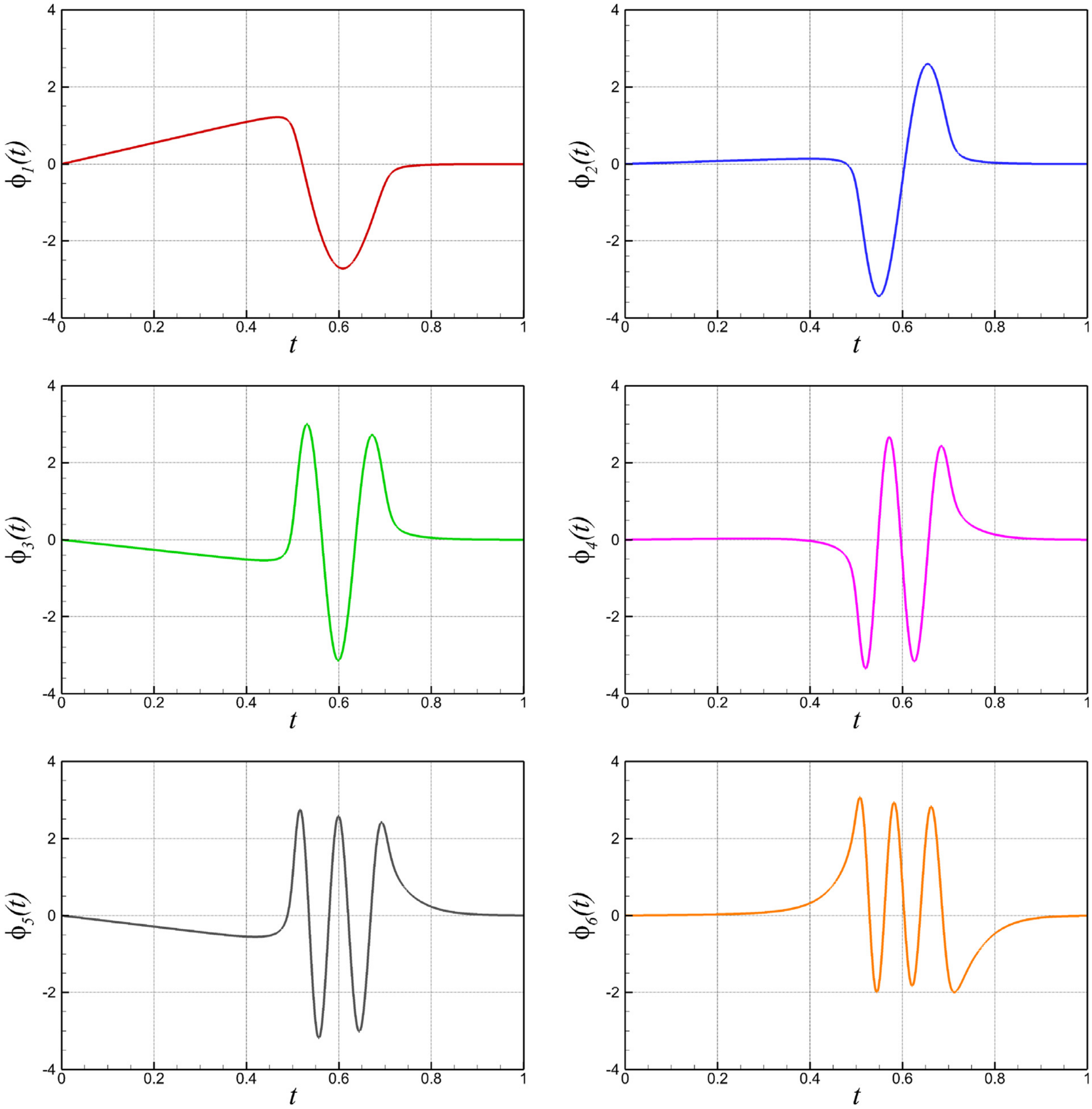

Fig. 3. Illustration of the most energetic POD basis functions generated using a total of 1010 data snapshots from $Re = 100$ and $Re = 1000$.

4. Galerkin projection methodology

The POD basis functions account for the essential dynamics of the underlying governing equations. After these data-driven empirical POD basis functions are obtained, a set of nonlinear ODEs can be derived by using a Galerkin projection. To build a projection based ROM, we truncate the system by considering the first R largest POD basis functions with $R \ll N$. These POD modes correspond to the R largest eigenvalues, λ_1, λ_2, ..., λ_R. Using these first R largest POD basis functions the field variables are then approximated.

Fig. 2 shows a collection of the modal values (i.e., the eigenvalues) of the POD-ROM analysis. The main motivation behind the construction of a POD-GP is evident when it is observed that the 6 most energetic modes retain 98.27% of the total energy of the system. We note that these bases are all orthogonal to each other. For our test case, i.e., $u(x,t) = \bar{u} + u'(x,t)$, the Galerkin projection can be carried out in the following manner

$$u'(x,t)=\sum_{k=1}^{R}a_k(t)\phi_k(x)\,, \tag{12}$$

where a_k are the time dependent coefficients, and ϕ_k are the space dependent modes. To derive the POD-ROM, we first rewrite the Burgers equation (i.e., Eq. (1)) in the following form

$$\frac{\partial u}{\partial t}=L[u]+N[u;u]\,, \tag{13}$$

where $L[f]=\frac{1}{Re}\frac{\partial^2 f}{\partial x^2}$ is the linear operator and $N[f;g]=-f\frac{\partial g}{\partial x}$ is the nonlinear operator. We emphasize that the procedure can be easily extended to the general nonlinear PDEs (e.g., Boussinesq or Navier–Stokes equations). By applying this projection to our nonlinear system (i.e., multiplying Eq. (13) with the basis functions and integrating over the domain), we obtain the Galerkin POD-ROM, denoted POD-GP:

$$\left(\frac{\partial u}{\partial t},\phi_k\right)=(L[u],\phi_k)+(N[u;u],\phi_k),\quad \text{for}\quad k=1,2,\ldots,R\,. \tag{14}$$

Substituting Eq. (12) into Eq. (14), and simplifying the resulting equation by using the condition of orthonormality given in Eq. (8), the POD-GP implementation can be written as follows:

$$\frac{da_k}{dt}=B_k+\sum_{i=1}^{R}\mathfrak{L}_{ik}a_i+\sum_{i=1}^{R}\sum_{j=1}^{R}\mathcal{N}_{ijk}a_ia_j,\quad \text{for}\quad k=1,2,\ldots,R, \tag{15}$$

where

$$B_k=(L[\bar{u}]+N[\bar{u};\bar{u}],\phi_k), \tag{16}$$

$$\mathfrak{L}_{ik}=(L[\phi_i]+N[\bar{u};\phi_i]+N[\phi_i;\bar{u}],\phi_k), \tag{17}$$

$$\mathcal{N}_{ijk}=(N[\phi_i;\phi_j],\phi_k). \tag{18}$$

The POD-GP given by Eq. (15) consists of R coupled ODEs and can be solved by a standard numerical method (such as the third-order Runge–Kutta scheme that was used in this study). The number of degrees of freedom of the system is now significantly lower. The vectors, matrices and tensors in Eqs. (16)–(18) are also precomputed quantities, which results in a dynamical system that can be solved very efficiently. To complete the dynamical system given by Eq. (15), the initial condition is given by using the following projection:

$$a_k(t=0)=(u(x,t=0)-\bar{u}(x),\phi_k)\,, \tag{19}$$

where $u(x,t=0)$ is the physical initial condition of the problem given in Eq. (2).

5. Artificial neural networks

A simple feed-forward artificial neural network consists of L layers with each layer possessing a predefined number of unit cells called neurons and is utilized for establishing nonlinear maps between two spaces of potentially different dimensionality through a supervised learning. Each of these layers may be considered an intermediate step in successive transformations between the input and output space and has an associated transfer function and each unit cell has an associated bias. Therefore, in our choice of network, any input to the neuron has a bias added to it followed by an activation through the transfer function. To describe this process using equations, we have [23]

$$s_i^l=\sum_j W_{ij}^l X_j^{l-1}, \tag{20}$$

as the state of the signal in the lth layer receiving a set of inputs from the $(l-1)$th layer, where W_{ij}^l stands for a matrix of weights linking the $l-1$ and l layers with X_j^{l-1} being the output of the $(l-1)$th layer. The output signal of the lth layer is now given by

$$X_i^l=F(s_i^l+b_i^l), \tag{21}$$

where b_i^l is the biasing parameter for the ith neuron in the lth layer. Every node (or unit cell) has an associated transfer function which acts on its input and bias to produce an output which is ‘fed forward’ in the network. The nodes which take the raw inputs of our training data set (i.e., the nodes of the first layer in the network) perform no computation (i.e., they do not have any biasing or activation through a transfer function). The next layers are a series of unit cells which have an associated bias and activation function which perform computation on their inputs. These are called the hidden layers due to the indeterminate nature of their mathematical operations. The final layer in the network is that of the outputs. The output layer generally has a linear activation function with a bias which implies a simple summation of inputs incident to a unit cell with its associated bias. A mathematical description of this mapping operation can be expressed as

$$\mathbb{M} : \{p_1, p_2, \dots, p_P\} \in \mathbb{R}^P \rightarrow \{q_1, q_2, \dots, q_Q\} \in \mathbb{R}^Q. \tag{22}$$

where P and Q are the dimensions of the input and output spaces. In this investigation, we have used one hidden layer of neurons between the set of inputs and targets with a tan-sigmoid activation function. The tan-sigmoid function can be expressed as

$$F(a) = \frac{2}{1 + \exp(-2a)} - 1. \tag{23}$$

The transfer function F calculates the neuron's output given its net input. In theory, any differentiable function can qualify as an activation function [119], however, only a small number of functions which are bounded, monotonically increasing and differentiable are used for this purpose. The choice of using ANN is motivated by its excellent performance as a forecasting tool [22,54] and its general suitability in the machine learning and function estimation domain (e.g., see [43] and references therein).

5.1. POD transforms

The key idea in our learning problem is the model reduction. Instead of posing our learning problem in a full-order space, we first apply projection to the full-order data and generate coefficients on a reduced subspace, and then construct the learning on this reduced-order space. Here we briefly describe the main procedure to generate such nun-intrusive reduced order models. Lets consider we have snaphots either coming from the full-order numerical simulations or experiments. Section 3 illustrates how to obtain representative POD modes. At any snapshot, $u^{(n)}$, our encoder operator can be written using the definition of inner product as

$$a_k^{(n)} = \left(u^{(n)} - \bar{u}, \phi_k\right), \quad \text{for} \quad k = 1, 2, \dots, R, \tag{24}$$

where basis functions ϕ_k and the mean field $\bar{u}$ are already available. This can be interpreted as a forward POD transform from the FOM space to the ROM space. We then perform our supervised learning for $a_k^{(n)}$ to train a map for computing the modal coefficient trajectories. The details of the neural network architectures can be found in more detail in the following section. Finally, the decoder operator is defined as

$$u^{(n)} = \bar{u} + \sum_{k=1}^{R} a_k^{(n)} \phi_k, \tag{25}$$

which can be referred to as an inverse POD transform. Indeed this expression can be used to generate full-order space data from the modal coefficients at any time.

5.2. ANN training

The training of a desired ANN is carried out by minimizing the error between the target and the inputs to determine a set of best fit parameters. This is also known as supervised learning where labeled data is utilized for gradient based optimization. Our best fit parameters obtained from optimization are biases and linear weights that have captured the underlying relationship between the targets and inputs and may now be used to predict target data for inputs a-posteriori. The main advantage of using the ANN approach over traditional statistical regression models is that comparatively smaller data sets for training are suitable. We shall describe the individual features of the ANN architecture relevant to our test case in the sections that follow.

The training of an ANN architecture may be undertaken by a variety of optimization algorithms. For our purpose, we have employed the Bayesian regularization minimization algorithm which is one of the popular approaches to ANN training within the framework of the MATLAB `nnstart` toolbox and particularly useful for noisy data [35]. We remark that other training methods may also be used such as standard Levenberg-Marquardt [60] and nature-inspired heuristic algorithms such as particle swarm [25], differential evolution [98] or leapfrogging [85]. However, access to gradient information (through the use of a differentiable cost-function) heavily favors the use of gradient-based optimizers. The process of backpropagation, where the errors in the output of the network are propagated back through the network to determine optimal search directions in parameter space is well suited for our particular architecture which utilizes solely one hidden layer. However, a detailed hyperparamater search is beyond the scope of this study since we would like to focus on the performance assessment of the proposed architectures using default options available in `nnstart` (e.g., mean-squared error loss function).

For the purpose of tracking the training of the ANN and evaluating its performance, we utilize three different subsets of our overall data with 70% being utilized for training, and 15% being utilized for validation and testing each. The training

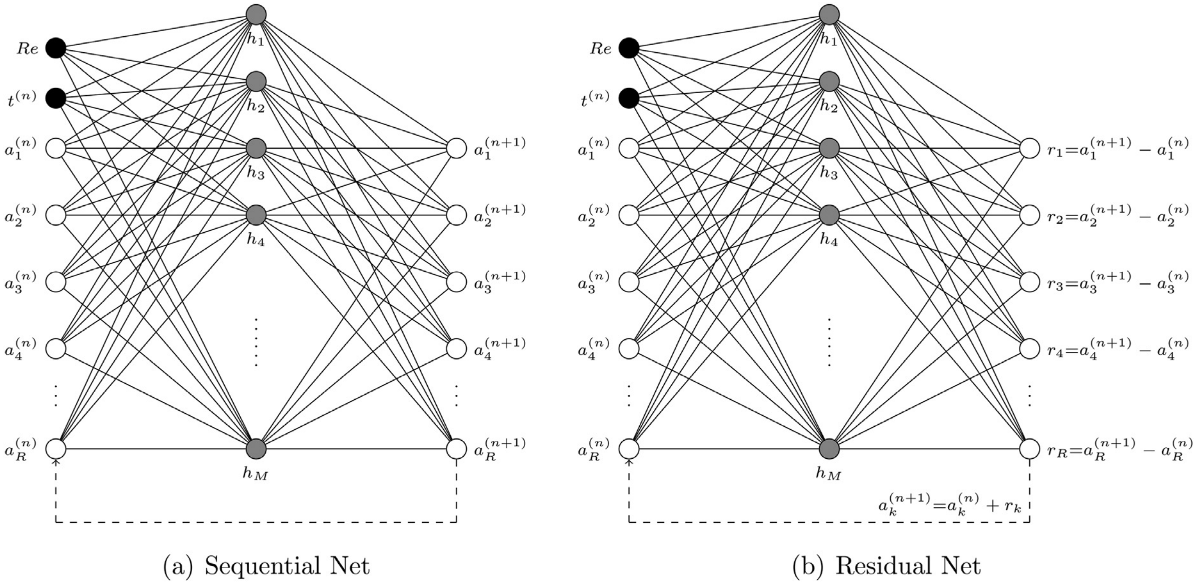

Fig. 4. The non-intrusive model order reduction enablers for transient flows with the sequential network $\mathbb{M}_1$ (left) and residual network $\mathbb{M}_2$ (right) ANN architectures. We illustrate our architectures for one physical control parameter *Re* and use only one hidden layer equipped with *M* neurons and employing *R* most energetic POD modes.

data set is used to adjust the weights of the ANN. The validation data set is used to minimize overfitting, i.e., it is used to verify whether an increase in accuracy over the training data set also yields an increased accuracy over a data set that is not revealed to the network a-priori. The validation data set is used for testing the final solution of the neural network. We note that the input and output features are all scaled between their minimum and maximum values and the scaling parameters are ported for deployment. Within the POD context, the trained ANN can be used as a ‘blackbox’ which takes the present state of the latent space dynamics as the input along with a dimensionless parameter and the present time and uses the memory of its training to determine the time evolution. Although not shown here for brevity, we note that the regression analysis shows a perfect correlation in the time history of the temporal modes and develops a system which could predict their evolution accurately.

5.3. ANN architecture for model order reduction

In the present study, we devise a framework for developing an ANN in order to create a very fast (equation-free or non-intrusive) integrator/solver for certain types of PDE systems. This idea can be considered as a competitor for POD-GP, with the advantage this is an equation-free and purely data-driven with associated benefits in computational efficiency and model-form uncertainty reduction.

Fig. 4 illustrates the ANN architectures utilized in this study. In these architectures, several inputs are provided, out of which the majority are basis coefficients at the current timestep in the POD-ROM evolution and the remaining two are Reynolds number and time. The training data is also arranged in the same manner. Our outputs are given by just the coefficient values required for a field reconstruction in the next step. The figure referred to above shows the number of inputs (modal POD coefficients, Reynolds number and time) and the number of outputs (modal POD coefficients at the next timestep). The middle layer l contains the neurons with the tan-sigmoid activation function and these are varied in our investigation. Therefore our first mapping approach, sequential network (SN) architecture, equipped by $R = 6$ modes can be given by

$$\mathbb{M}_1 : \{Re, t^{(n)}, a_1^{(n)}, a_2^{(n)}, \ldots, a_6^{(n)}\} \in \mathbb{R}^8 \to \{a_1^{(n+1)}, a_2^{(n+1)}, \ldots, a_6^{(n+1)}\} \in \mathbb{R}^6. \tag{26}$$

We denote this model as POD-ANN-SN. As discussed by Chen et al. [18], machine learning models such as residual networks [44] define a discrete sequence of finite transformations to a hidden state in the following form [41,67]:

$$a_k^{(n+1)} = a_k^{(n)} + r_k, \quad \text{for} \quad k = 1, 2, \ldots, R, \tag{27}$$

where r_k refers to the residual that we can approximate using an ANN. This can be seen as an Euler update in a discrete sense as follows (i.e., defining the residual as $r_k = \Delta t f_k$)

$$a_k^{(n+1)} = a_k^{(n)} + \Delta t f_k, \tag{28}$$

where $\Delta t = t^{(n+1)} - t^{(n)}$, and f_k is the slope

$$f_k = \frac{a_k^{(n+1)} - a_k^{(n)}}{\Delta t}. \tag{29}$$

To investigate the performance of such a residual network, a similar network shown also in Fig. 4 is also designed for $R = 6$ largest POD modes, which can be defined as

$$\mathbb{M}_2 : \{Re, t^{(n)}, a_1^{(n)}, a_2^{(n)}, \ldots, a_6^{(n)}\} \in \mathbb{R}^8 \to \{r_1, r_2, \ldots, r_6\} \in \mathbb{R}^6. \tag{30}$$

and the POD coefficients are then obtained by Eq. (27) before the consequent step. This model is denoted as POD-ANN-RN in our study. Details related to data acquistion for our network training are provided in Section 7. We highlight that the main objective of the present work is to test and evaluate the characteristics of the proposed POD-ANN approaches and compare them with the true solution obtained by a projection to the full order model (FOM). Our assessments also includes intrusive results generated by the standard POD-GP framework.

6. Numerical methods

In this section, we provide a brief description of the numerical methods employed in this study. First of all, we implement compact difference schemes to approximate the differential operators (both linear and nonlinear terms) in the POD-GP. Compact difference schemes have shown their ability to reach the objectives of high-accuracy and low computational cost for many problems in fluid dynamics [59,103].

In a compact difference scheme, the first order derivatives can be computed accordingly [59]

$$\alpha f'_{i-1} + f'_i + \alpha f'_{i+1} = a\frac{f_{i+1} - f_{i-1}}{2\Delta x} + b\frac{f_{i+2} - f_{i-2}}{4\Delta x}, \tag{31}$$

which gives rise to an α-family of tridiagonal schemes with $a = \frac{2}{3}(\alpha + 2)$, and $b = \frac{1}{3}(4\alpha - 1)$. The subscript i represents the spatial grid index in the considered direction, and Δx is the uniform grid spacing in that direction. Here, $\alpha = 0$ leads to the explicit non-compact fourth-order scheme for first derivative computations. A classical compact fourth-order scheme, which is also known as Padé scheme, is obtained by setting $\alpha = 1/4$, resulting in $b = 0$ with a compact dependence. In computing the linear diffusion term, the second derivative compact scheme is given by

$$\alpha f''_{i-1} + f''_i + \alpha f''_{i+1} = a\frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} + b\frac{f_{i+2} - 2f_i + f_{i-2}}{4\Delta x^2}, \tag{32}$$

where $a = \frac{4}{3}(1 - \alpha)$, and $b = \frac{1}{3}(10\alpha - 1)$. For $\alpha = 1/10$ the classical fourth-order Padé scheme is obtained. The high-order one-sided derivative formulas are used for the Dirichlet boundary conditions to complete a tridiagonal system of equations for both the first and second order derivative formulas [17].

As a time integrator, we use an optimal third-order accurate total variation diminishing Runge–Kutta (TVDRK3) scheme that is given as [39]

$$\begin{aligned} u_i^{(1)} &= u_i^{(n)} + \Delta t \pounds(u_i^{(n)}), \\ u_i^{(2)} &= \frac{3}{4}u_i^{(n)} + \frac{1}{4}u_i^{(1)} + \frac{1}{4}\Delta t \pounds(u_i^{(1)}), \\ u_i^{(n+1)} &= \frac{1}{3}u_i^{(n)} + \frac{2}{3}u_i^{(2)} + \frac{2}{3}\Delta t \pounds(u_i^{(2)}), \end{aligned} \tag{33}$$

where $\pounds(u)$ is the discrete operator of spatial derivatives including linear and nonlinear terms. To perform the inner products defined by Eq. (5), we compute the integral of any arbitrary function $g(x)$ over the domain Ω by using the Simpson's 1/3 rule [47]

$$\int_\Omega g(x)dx = \frac{\Delta x}{3}\sum_{i=1}^{N_x/2-1}\left(g_{2i} + 4g_{2i+1} + g_{2i+2}\right), \tag{34}$$

where N_x is the total number of grid point (e.g., an even number in order to be consistent with the Simpson's integration rule).

7. Results

In the following we provide modal coefficient evolution data for our test case and make assessments of the relative performance of the POD-ANN and POD-GP approaches. Although measuring the state evolution error is not equivalent to measuring the error in any standard function space norm or quantity of interest, we found that it simplifies our demonstration and presents an efficient way in our assessments between the proposed approaches and standard POD-GP model. In addition, we also provide details related to the calculation of POD bases and their coefficients for the use of data-driven learning and a-posteriori deployment.

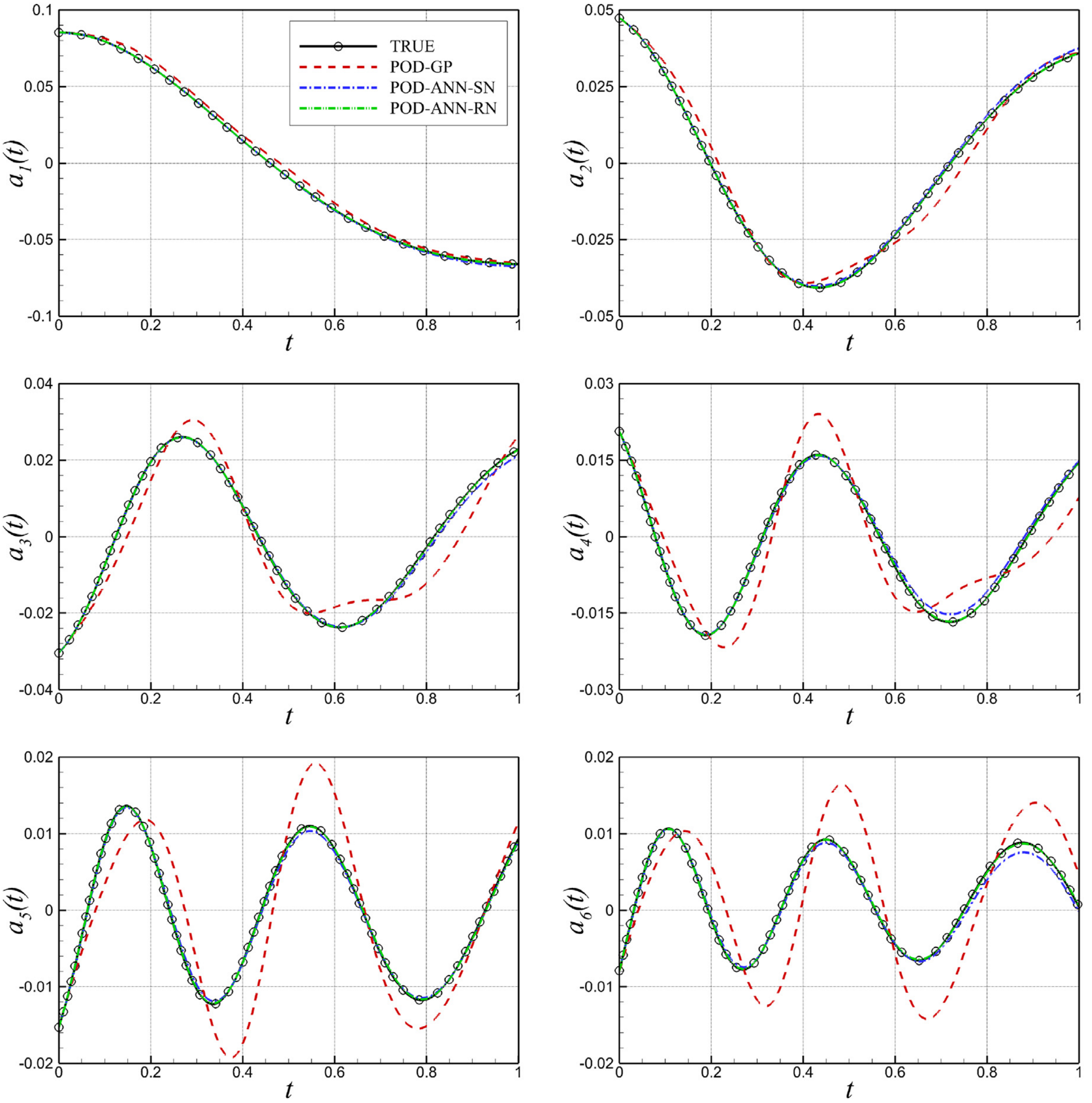

Fig. 5. Numerical assessments of the POD-GP and POD-ANN models with $R = 6$ POD modes applied to the Burgers problem for an interpolatory out-of-sample parameter $Re = 750$, where the training set includes snapshots between $Re = 100$ and $Re = 1000$. The non-intrusive POD-ANN-SN and POD-ANN-RN results are obtained using $M = 10$ neurons in the hidden layer. Note that the exact solution projected to the reduced order space is shown by the black solid line with circles.

POD bases for the viscous Burgers equation problem are generated by accumulation of snapshot data from the exact solution given by Eq. (2). As described in Section 3, snapshot sampling is done at different Reynolds numbers between 100 and 1000 at intervals of 100 each (i.e., $Re = 100$, $Re = 200$, $\ldots$, $Re = 1000$). Each choice of a Reynolds number is sampled for 101 snapshots in time between $t = 0$ and $t = 1$. This leads to a total of 1010 snapshots to construct a snapshot correlation data matrix C with the size of 1010×1010. Although we generate such 1010 snapshots from the closed form expression, we remark that these snapshots can be obtained from experiments or full order model numerical simulations in most real-life problems. The physics of these multiple snapshots for each control parameter is thus spanned by a single set of optimal bases used for subsequent ANN training or Galerkin projection. The ANN training data has been reconstructed with a timestep of $\Delta t = 0.001$ for the same database with 10 different Re from $t = 0$ to $t = 1$, which leads to 10,010 samples in our training set. We note that the timestep for POD-GP and POD-ANN deployment are kept identical in our numerical assessments. We highlight that the training of ANN to bypass the Galerkin procedure is an offline cost which utilizes approximately 45 seconds.

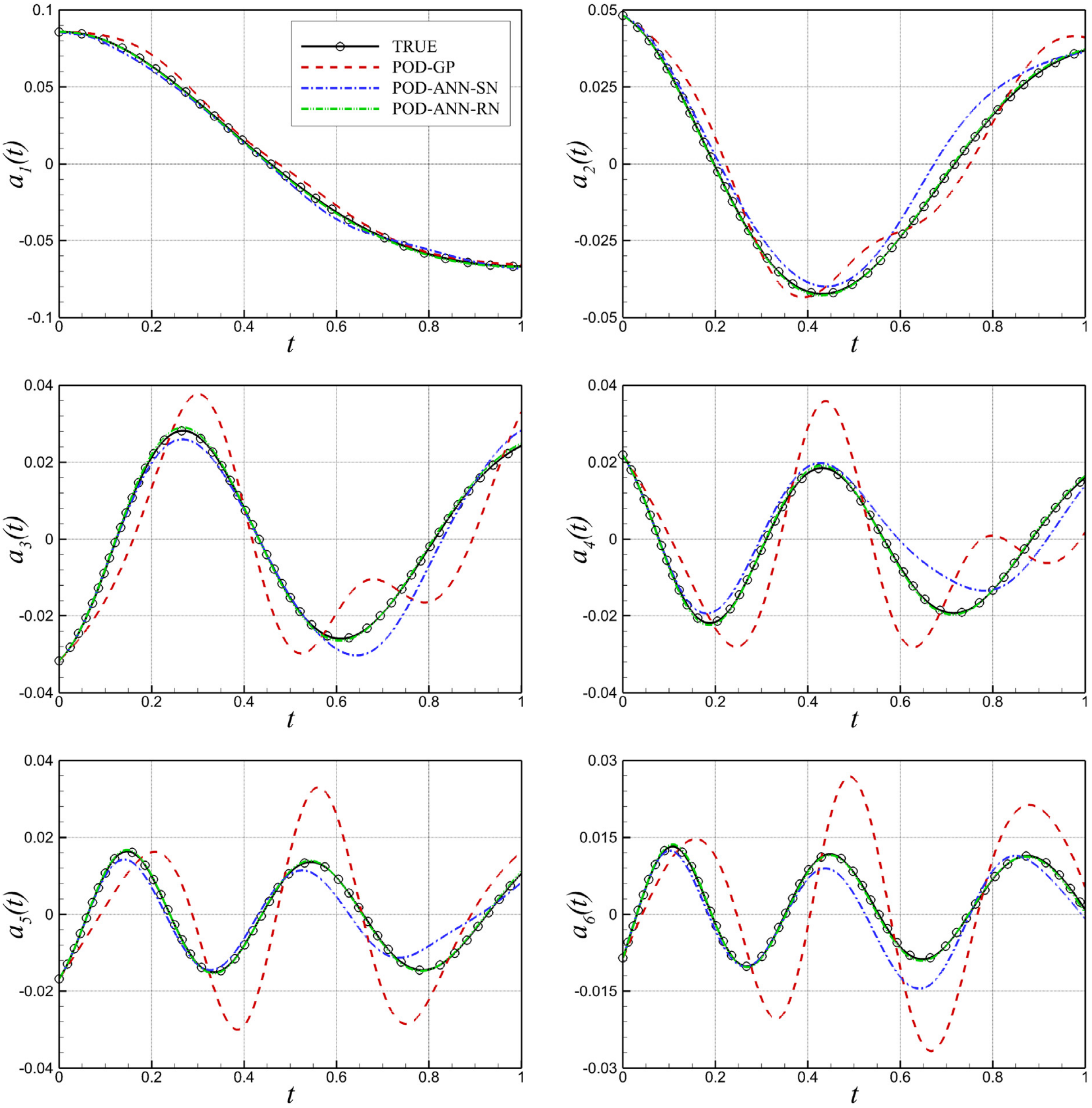

Fig. 6. Numerical assessments of the POD-GP and POD-ANN models with $R = 6$ POD modes applied to the Burgers problem at $Re = 1250$, where the training set includes snapshots between $Re = 100$ and $Re = 1000$. The non-intrusive POD-ANN-SN and POD-ANN-RN results are obtained using $M = 10$ neurons in the hidden layer. Note that the exact solution projected to the reduced order space is shown by the black solid line with circles.

We note that the results shown in this subsection are for Reynolds number values which were not within the training dataset range for the ANN. Throughout our study we use a single-layer neural network with $M = 10$ neurons and employing $R = 6$ POD modes. Fig. 5 shows a comparison at $Re = 750$ on predicting the POD basis coefficients. Our numerical assessment with Reynolds number of 750 represents the interpolative abilities of both POD-GP and POD-ANN approaches. It is clear that the POD-ANN approach preserves state accuracy for a greater duration as compared to the POD-GP approach. At this *Re*, the performance of the both POD-ANN approaches is remarkable. Indeed for the first mode, the POD-ANN approaches preserve the amplitude and phase behavior for the coefficient very well. Coefficients for higher modes tend to show a slight variation in amplitude when compared to DNS data but this variation is quite negligible when compared along the POD-GP approach which shows extensive loss of phase and amplitude at higher modes.

Our next test at $Re = 1250$ represents the extrapolative abilities of both POD-GP and POD-ANN models, in which the training data is generated from snapshots between $Re = 100$ and $Re = 1000$. Fig. 6 shows a similar study where we have retained only 6 POD modes to predict the flow field at higher *Re*. It can be easily seen that both non-intrusive POD-ANN-

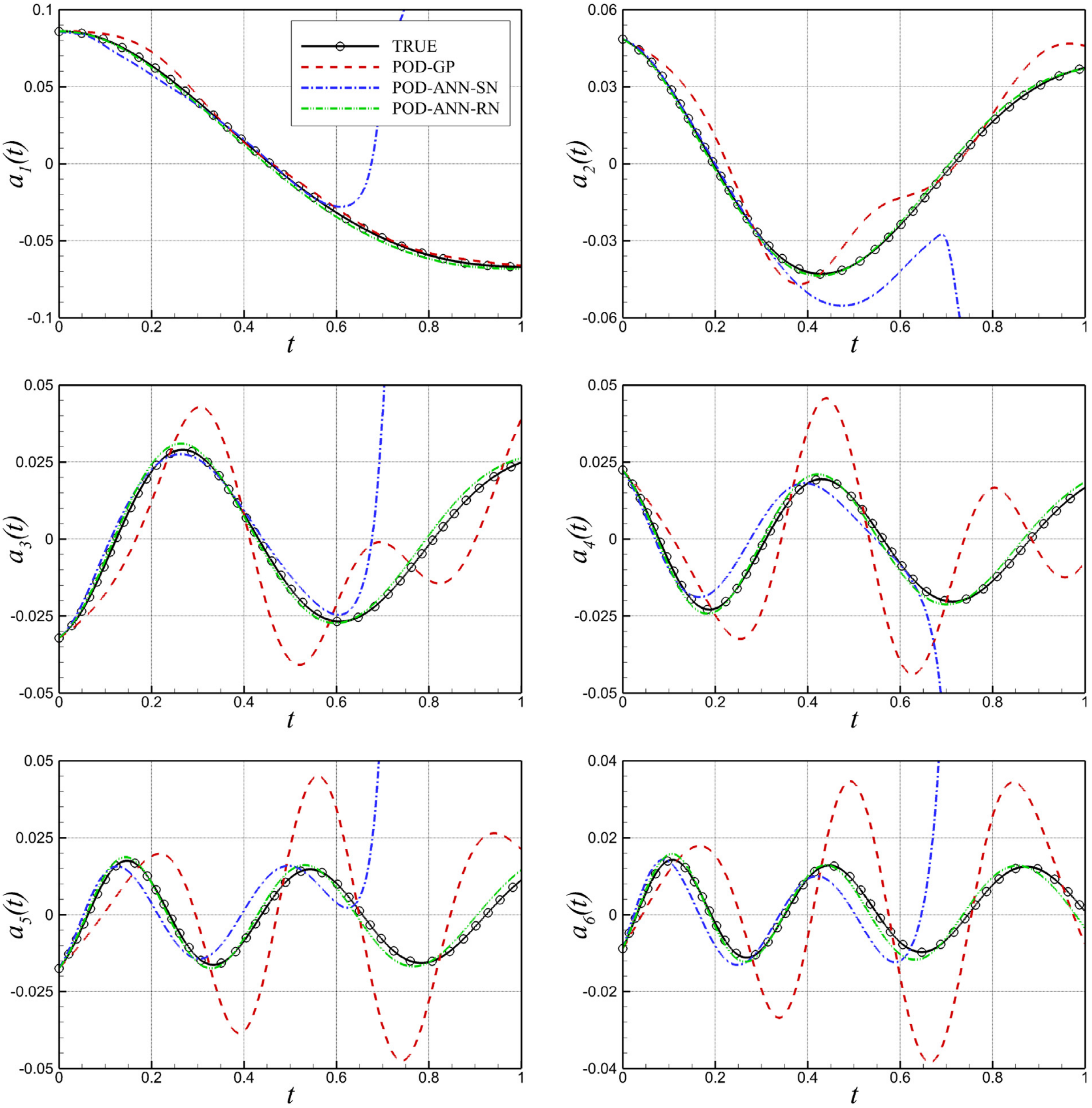

Fig. 7. Numerical assessments of the POD-GP and POD-ANN models with $R = 6$ POD modes applied to the Burgers problem at $Re = 2000$, where the training set includes snapshots between $Re = 100$ and $Re = 1000$. The non-intrusive POD-ANN-SN and POD-ANN-RN results are obtained using $M = 10$ neurons in the hidden layer. Note that the exact solution projected to the reduced order space is shown by the black solid line with circles.

SN and POD-ANN-RN approaches yield stable and accurate results. Furthermore, one can immediately notice a significant inaccuracy in the POD-GP approach. Considering only six largest POD modes might not be rich enough to represent the underlying transient behavior. However, the POD-ANN approaches still perform well in terms of capturing the amplitude of the first six modes even though slight variations in the phase for the highest modes are seen in the $Re = 1250$ case as shown in Fig. 6. Although the POD-GP method shows a considerable deviation from the truth as compared to the POD-ANN approaches, when examining the difference between the POD-ANN-SN and POD-ANN-RN non-intrusive methodologies, we can see that the POD-ANN-RN provides more accurate estimates as compared to the POD-ANN-SN approach which does a decent job at capturing amplitude and phase behavior at this Reynolds number.

To further investigate the behavior of the proposed ANN architectures at higher Reynolds number, we test the proposed non-intrusive frameworks for Re = 2000. To this end, Fig. 7 presents our numerical assessments at $Re = 2000$ that is far beyond the parameter range utilized in the training database. This examination points out that the POD-ANN-RN is more stable than the POD-ANN-SN. This figure also demonstrates significantly more accurate behavior shown by the POD-ANN-RN approach. In terms of computational costs, the POD-ANN approaches equipped by $M = 10$ neurons do not require the use of Runge–Kutta based time stepping and is thus observed to be two orders of magnitude faster than the POD-GP. However, we note that the running time is significantly less than a second for both non-intrusive POD-ANN and intrusive POD-GP approaches with $R = 6$ modes. Indeed, we observe that the computational performance of both POD-GP and POD-ANN codes is dominated by data-writing for post-processing.

We stress here that bypassing the computation of the reduced system obtained through Galerkin projection and truncation ensures that our non-intrusive frameworks are *purely* data-driven. The non-intrusive nature of the POD-ANN methodology and the lack of a system of equations that needs to be solved for discretely in time thus leads to an attractive alternative to the POD-GP implementation. Although we illustrate that a robust surrogate model can be obtained by a residual network based ANN architecture without a need to access to the governing equations, we highlight that the intrusive or any other models generated from the first principles allow one to prove important properties like numerical stability, convergence, energy conservation, etc., which are all seemingly lost in blackbox ANN models. However, we refer the readers to Xu and Darve [118] for an excellent discussion of the key issues in utilizing neural network approaches to inverse problems in differential equations such as universal approximation, regularization, curse of dimensionality and computational efficiency.

8. Concluding remarks

In this study, we have investigated the predictive performance of two different POD-enabled model order reduction approaches for the time evolution of the state. First, we look at the classical Galerkin projection approach where a system of ODEs is developed for the evolution of each state coefficient (corresponding to a scalar field variable). Second, we have developed an equation-free artificial neural network (ANN) framework through data-driven learning which predicts the evolution of the modes without the explicit time stepping approach used in the Galerkin projection. We give a demonstration of the proposed ANN framework for two architectures: (i) a sequential network (SN) where we compute the trajectory of POD coefficients directly, and (ii) a residual network (RN) where we train our model with a discrete residual information, rather than using the sequential state variables. Our numerical assessments are presented for the viscous Burgers equation benchmark problem at various Reynolds number. We notice that the POD-ANN approaches outperforms POD-GP consistently across the different testing environments and proves excellent in both interpolating and extrapolating its learning from obtained data sets for simulations with different control parameters and underlying physics. Furthermore, we demonstrate that POD-ANN-RN yields significantly more stable results especially for the extrapolatory parameter range that are outside of the training range. We revisit the questions we posed in the introduction to conclude our investigation:

- We can conclude with confidence that the POD-ANN approach is superior to the POD-GP approach for the Burgers equation especially for highly truncated systems (i.e., systems represented by a few modes only).
- We can also state that POD-ANN is a viable tool for extrapolation and interpolation beyond the data sets used to train its learning. While it is expected to provide good results within the bounds of the training data set, the Burgers case detailed its capacity to predict the coefficient evolution of different Reynolds numbers in an accurate manner with much greater accuracy than POD-GP.
- We find that the residual network formulation is significantly more stable than the sequential network formulation.

Furthermore, we present the following observations which were made during this study:

- Once trained, the POD-ANN can be considered to be an ‘equation-free’ ROM architecture.
- Although the POD-GP gives us usually better approximations while increasing the retained number modes, the POD-ANN is better suited to more radically truncated systems since an ANN with higher degrees of freedom may show statistical errors on time evolution.
- POD-ANN represents a viable framework for modeling highly nonlinear systems when data is available incrementally and constant updates are required. It is also very well suited to situations where model interpretability is not a major concern.
- While our training algorithm utilized the standard Bayesian regularization algorithm, it must be noted that there exists many training methodologies which may better preserve the underlying statistical relationships of the inputs and outputs. The effect of training approaches is a future direction of study for this work.
- We also recognize that the POD-ANN, despite its obvious benefits, also leads to the drawback of reduced interpretability due to its blackbox nature. In addition, the a-priori requirement of training implies poor generalizability of the POD-ANN framework to completely different physical regimes.
- Additionally we note that the use of more advanced regularization techniques in the ANN training (an aspect not considered in this study) would potentially lead to more stable non-intrusive ROM methodologies.
- Training with noisy data may also be studied for obviating the need for closure modeling in the POD-GP framework.

An overview of our conclusions are provided in Table 2 which contrasts the difference between POD-GP and POD-ANN type methods for ROMs. Although our study utilizes a shallow neural network architecture, we believe that the integration of machine learning tools such as deep learning into the mainstream reduced order modeling ideology would represent a

Table 2
A comparison of intrusive and non-intrusive methodologies describing strengths and weaknesses for each framework.

Intrusive ROMs (POD-GP)		Non-intrusive ROMs (POD-ANN)	
+	Solid foundations based on physics and first principles (high interpretability)	-	Mostly blackbox solvers and packages (low interpretability)
-	Need access to the governing equations	+	Only need observed/measured data
+	Generalizes well to new problems with similar physics	-	Poor generalization on unseen problems
-	Implementations are mostly problem dependent	+	Extremely convenient for tensor based computing
+	Numerical analysis is easier	-	Hard to establish rules for its convergence, stability, and energy conservation

significant increase in its viability for both theoretical and application oriented goals. We would like to conclude that POD-ANN-RN would be a viable ROM procedure for convective flows where the response of complex system dynamics is desired to be represented accurately by a few representative modes.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC0019290. OS gratefully acknowledges their support. Disclaimer: 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.

References

[1] Ahmed M, San O. Stabilized principal interval decomposition method for model reduction of nonlinear convective systems with moving shocks. Comput Appl Math 2018;37(5):6870–902.

[2] Akhtar I, Borggaard J, Burns JA, Imtiaz H, Zietsman L. Using functional gains for effective sensor location in flow control: a reduced-order modelling approach. J Fluid Mech 2015;781:622–56.

[3] Amsallem D, Farhat C. Stabilization of projection-based reduced-order models. Int J Numer Methods Eng 2012;91(4):358–77.

[4] Aubry N, Holmes P, Lumley JL, Stone E. The dynamics of coherent structures in the wall region of a turbulent boundary layer. J Fluid Mech 1988;192(1):115–73.

[5] Audouze C, De Vuyst F, Nair PB. Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations. Numer Methods Partial DifferEquations 2013;29(5):1587–628.

[6] Benner P, Gugercin S, Willcox K. A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev 2015;57(4):483–531.

[7] Berkooz G, Holmes P, Lumley JL. The proper orthogonal decomposition in the analysis of turbulent flows. Ann Rev Fluid Mech 1993;25(1):539–75.

[8] Bistrian D, Navon I. An improved algorithm for the shallow water equations model reduction: dynamic mode decomposition vs POD. Int J Numer Methods Fluids 2015;78(9):552–80.

[9] Bistrian DA, Navon IM. Randomized dynamic mode decomposition for nonintrusive reduced order modelling. Int J Numer Methods Eng 2017;112(1):3–25.

[10] Borggaard J, Hay A, Pelletier D. Interval-based reduced order models for unsteady fluid flow. Int J Numer AnalModel. 2007;4(3–4):353–67.

[11] Borggaard J, Iliescu T, Wang Z. Artificial viscosity proper orthogonal decomposition. Math Comput Modell 2011;53(1–2):269–79.

[12] Bright I, Lin G, Kutz JN. Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements. Phys Fluids 2013;25(12):127102.

[13] Brunton SL, Noack BR. Closed-loop turbulence control: progress and challenges. Appl Mech Rev 2015;67(5):050801.

[14] Buffoni M, Camarri S, Iollo A, Salvetti MV. Low-dimensional modelling of a confined three-dimensional wake flow. J Fluid Mech 2006;569:141–50.

[15] Bui-Thanh T, Willcox K, Ghattas O, van Bloemen Waanders B. Goal-oriented, model-constrained optimization for reduction of large-scale systems. J Comput Phys 2007;224(2):880–96.

[16] Carlberg K, Farhat C. A low-cost, goal-oriented ‘compact proper orthogonal decomposition’ basis for model reduction of static systems. Int J Numer Methods Eng 2011;86(3):381–402.

[17] Carpenter MH, Gottlieb D, Abarbanel S. Stable and accurate boundary treatments for compact, high-order finite-difference schemes. Appl Numer Math 1993;12(1):55–87.

[18] Chen TQ, Rubanova Y, Bettencourt J, Duvenaud DK. Neural ordinary differential equations. In: Advances in neural information processing systems; 2018. p. 6571–83.

[19] Chen W, Hesthaven JS, Junqiang B, Qiu Y, Yang Z, Tihao Y. Greedy nonintrusive reduced order model for fluid dynamics. AIAA J 2018;56(12):4927–43.

[20] Cizmas PG, Richardson BR, Brenner TA, OBrien TJ, Breault RW. Acceleration techniques for reduced-order models based on proper orthogonal decomposition. J Comput Phys 2008;227(16):7791–812.

[21] Cordier L, Noack BR, Tissot G, Lehnasch G, Delville J, Balajewicz M, et al. Identification strategies for model-based control. Exp Fluids 2013;54(8):1580.

[22] Dawson CW, Wilby R. An artificial neural network approach to rainfall-runoff modelling. Hydrol Sci J 1998;43(1):47–66.

[23] Demuth HB, Beale MH, De Jess O, Hagan MT. Neural network design. Martin Hagan, OK, USA; 2014.

[24] Dyke S, Spencer Jr B, Sain M, Carlson J. Modeling and control of magnetorheological dampers for seismic response reduction. Smart Mater Struct 1996;5(5):565.

[25] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science. IEEE; 1995. p. 39–43.

[26] Efe M, Debiasi M, Yan P, Ozbay H, Samimy M. Control of subsonic cavity flows by neural networks-analytical models and experimental validation. In: 43rd AIAA aerospace sciences meeting and exhibit; 2005. p. 294.
[27] Efe MO, Debiasi M, Ozbay H, Samimy M. Modeling of subsonic cavity flows by neural networks. In: Proceedings of the IEEE international conference on mechatronics. IEEE; 2004. p. 560–5.
[28] El Majd BA, Cordier L. New regularization method for calibrated POD reduced-order models. Math Modell Anal 2016;21(1):47–62.
[29] Faller WE, Schreck SJ. Unsteady fluid mechanics applications of neural networks. J Aircr 1997;34(1):48–55.
[30] Fang F, Pain C, Navon I, Gorman G, Piggott M, Allison P, et al. A POD reduced order unstructured mesh ocean modelling method for moderate Reynolds number flows. Ocean Modell 2009;28(1):127–36.
[31] Fortuna L, Nunnari G, Gallo A. Model order reduction techniques with applications in electrical engineering. Springer Science & Business Media; 2012.
[32] Freund RW. Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation. In: Applied and computational control, signals, and circuits. Springer; 1999. p. 435–98.
[33] Gamahara M, Hattori Y. Searching for turbulence models by artificial neural network. Phys Rev Fluids 2017;2(5):054604.
[34] Gautier N, Aider J-L, Duriez T, Noack B, Segond M, Abel M. Closed-loop separation control using machine learning. J Fluid Mech 2015;770:442–57.
[35] Gençay R, Qi M. Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. IEEE Trans Neural Netw 2001;12(4):726–34.
[36] Gillies E. Low-dimensional characterization and control of non-linear wake flows. PhD. Dissertation, University of Glasgow, Scotland; 1995.
[37] Gillies E. Low-dimensional control of the circular cylinder wake. J Fluid Mech 1998;371:157–78.
[38] Gillies E. Multiple sensor control of vortex shedding. AIAA J 2001;39(4):748–50.
[39] Gottlieb S, Shu CW. Total variation diminishing Runge–Kutta schemes. Math Comput 1998;67(221):73–85.
[40] Gunzburger MD. Flow control. Springer, Berlin; 2012.
[41] Haber E, Ruthotto L. Stable architectures for deep neural networks. Inverse Prob 2017;34(1):014004.
[42] Hampton J, Fairbanks HR, Narayan A, Doostan A. Practical error bounds for a non-intrusive bi-fidelity approach to parametric/stochastic model reduction. J Comput Phys 2018;368:315–32.
[43] Haykin S. Neural networks and learning machines. Third edition. Upper Saddle River, NJ, USA: Pearson Education, Inc.; 2009.
[44] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–8.
[45] Hesthaven JS, Ubbiali S. Non-intrusive reduced order modeling of nonlinear problems using neural networks. J Comput Phys 2018;363:55–78.
[46] Hocevar M, Sirok B, Grabec I. Experimental turbulent field modeling by visualization and neural networks. J Fluids Eng 2004;126:316–22.
[47] Hoffman JD, Frankel S. Numerical methods for engineers and scientists. CRC Press, NY, USA; 2001.
[48] Holmes P, Lumley JL, Berkooz G. Turbulence, coherent structures, dynamical systems and symmetry. Cambridge University Press, NY, USA; 1998.
[49] Hotelling H. Analysis of a complex of statistical variables into principal components. J Educ Psychol 1933;24(6):417–41.
[50] Iollo A, Lanteri S, Désidéri J-A. Stability properties of POD–Galerkin approximations for the compressible Navier–Stokes equations. Theor Comput Fluid Dyn 2000;13(6):377–96.
[51] Kani J.N., Elsheikh A.H. DR-RNN: A deep residual recurrent neural network for model reduction. arXiv:170900939v1 2017.
[52] Kerschen G, Golinval J-c, Vakakis AF, Bergman LA. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. Nonlinear Dyn 2005;41(1–3):147–69.
[53] Khibnik A, Narayanan S, Jacobson C, Lust K. Analysis of low dimensional dynamics of flow separation. In: Continuation methods in fluid dynamics, 74. Vieweg; 2000. p. 167–78.
[54] Kim T-W, Valdés JB. Nonlinear model for drought forecasting based on a conjunction of wavelet transforms and neural networks. J Hydrol Eng 2003;8(6):319–28.
[55] Kramer B, Grover P, Boufounos P, Nabi S, Benosman M. Sparse sensing and DMD-based identification of flow regimes and bifurcations in complex flows. SIAM J Appl Dyn Syst 2017;16(2):1164–96.
[56] Kunisch K, Volkwein S. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. SIAM J Numer Anal 2002;40(2):492–515.
[57] Kunisch K, Volkwein S. Optimal snapshot location for computing POD basis functions. ESAIM Math Modell NumerAnal 2010;44(3):509–29.
[58] Lee C, Kim J, Babcock D, Goodman R. Application of neural networks to turbulence control for drag reduction. Phys Fluids 1997;9(6):1740–7.
[59] Lele SK. Compact finite difference schemes with spectral-like resolution. J Comput Phys 1992;103(1):16–42.
[60] Levenberg K. A method for the solution of certain non-linear problems in least squares. Q Appl Math 1944;2(2):164–8.
[61] Lin Z, Xiao D, Fang F, Pain C, Navon IM. Non-intrusive reduced order modelling with least squares fitting on a sparse grid. Int J Numer Methods Fluids 2017;83(3):291–306.
[62] Ling J, Jones R, Templeton J. Machine learning strategies for systems with invariance properties. J Comput Phys 2016;318:22–35.
[63] Ling J, Kurzawski A, Templeton J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. J Fluid Mech 2016;807:155–66.
[64] Ling J, Templeton J. Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty. Phys Fluids 2015;27(8):085103.
[65] Loève M. Probability theory; foundations, random sequences. New York: D. Van Nostrand Company, NY, USA; 1955.
[66] Lorenz E.N. Empirical orthogonal functions and statistical weather prediction1956;.
[67] Lu Y., Zhong A., Li Q., Dong B. Beyond finite layer neural networks: bridging deep architectures and numerical differential equations. arXiv:171010121v1 2017.
[68] Lucia DJ, Beran PS, Silva WA. Reduced-order modeling: new approaches for computational physics. Prog Aerosp Sci 2004;40(1):51–117.
[69] Lui H.F., Wolf W.R. Construction of reduced order models for fluid flows using deep feedforward neural networks. arXiv:190305206v1 2019.
[70] Lumley J. The structures of inhomogeneous turbulent flow. In: Yaglom A, Tatarski V, editors. Atmospheric turbulence and radio wave propagation; 1967. p. 160–78.
[71] Ly HV, Tran HT. Modeling and control of physical processes using proper orthogonal decomposition. Math Comput Modell 2001;33(1–3):223–36.
[72] Maleewong M, Sirisup S. On-line and off-line POD assisted projective integral for non-linear problems: a case study with Burgers' equation. Int J MathComput Phys Electr Comput Eng 2011;5(7):984–92.
[73] Maulik R, San O. A neural network approach for the blind deconvolution of turbulent flows. J Fluid Mech 2017;831:151–81.
[74] Mignolet MP, Przekop A, Rizzi SA, Spottswood SM. A review of indirect/non-intrusive reduced order modeling of nonlinear geometric structures. J Sound Vibr 2013;332(10):2437–60.
[75] Milano M, Koumoutsakos P. Neural network modeling for near wall turbulent flow. J Comput Phys 2002;182(1):1–26.
[76] Mohan A.T., Gaitonde D.V. A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks. arXiv:180409269v1 2018.
[77] Mohebujjaman M, Rebholz LG, Xie X, Iliescu T. Energy balance and mass conservation in reduced order models of fluid flows. J Comput Phys 2017;346:262–77.
[78] Moosavi A., Stefanescu R., Sandu A. Efficient construction of local parametric reduced order models using machine learning techniques. arXiv:151102909v1 2015.
[79] Müller S, Milano M, Koumoutsakos P. Application of machine learning algorithms to flow modeling and optimization. Center for Turbulence Research Annual Research Briefs, Stanford University, Stanford, CA; 1999. p. 169–78.

[80] Narayanan S, Khibnik A, Jacobson C, Kevrekedis Y, Rico-Martinez R, Lust K. Low-dimensional models for active control of flow separation. In: Proceedings of the 1999 IEEE international conference on control applications, 2. IEEE; 1999. p. 1151–6.
[81] Noack BR, Morzynski M, Tadmor G. Reduced-order modelling for flow control, 528. Springer Verlag, Berlin; 2011.
[82] Östh J, Noack BR, Krajnović S, Barros D, Borée J. On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high--Reynolds-number flow over an Ahmed body. J Fluid Mech 2014;747:518–44.
[83] Peherstorfer B, Willcox K. Data-driven operator inference for nonintrusive projection-based model reduction. Comput Methods Appl MechEng 2016;306:196–215.
[84] Ravindran SS. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. Int J Numer Methods Fluids 2000;34(5):425–48.
[85] Rhinehart RR, Su M, Manimegalai-Sridhar U. Leapfrogging and synoptic leapfrogging: a new optimization approach. Comput Chem Eng 2012;40:67–81.
[86] Rowley CW, Dawson ST. Model reduction for flow analysis and control. Annu Rev Fluid Mech 2017;49:387–417.
[87] Roychowdhury J. Reduced-order modeling of time-varying systems. IEEE Trans Circuits Syst II 1999;46(10):1273–88.
[88] Sahan R, Koc-Sahan N, Albin D, Liakopoulos A. Artificial neural network-based modeling and intelligent control of transitional flows. In: Proceedings of the 1997 IEEE international conference on control applications. IEEE; 1997. p. 359–64.
[89] San O, Borggaard J. Principal interval decomposition framework for POD reduced-order modeling of convective Boussinesq flows. Int J Numer Methods Fluids 2015;78(1):37–62.
[90] San O, Iliescu T. Proper orthogonal decomposition closure models for fluid flows: Burgers equation. Int J Numer AnalModel 2014;5:217–37.
[91] San O, Iliescu T. A stabilized proper orthogonal decomposition reduced-order model for large scale quasigeostrophic ocean circulation. Adv Comput Math 2015;41(5):1289–319.
[92] San O, Maulik R. Extreme learning machine for reduced order modeling of turbulent geophysical flows. Phys Rev E 2018;97(4):042322.
[93] San O, Maulik R. Machine learning closures for model order reduction of thermal fluids. Appl Math Modell 2018;60:681–710.
[94] San O, Maulik R. Neural network closures for nonlinear model order reduction. Adv Comput Math 2018;44:1717–50.
[95] Silveira LM, Kamon M, White J. Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-D interconnect structures. IEEE Trans Compon PackagManuf Technol Part B 1996;19(2):283–8.
[96] Singh AP, Duraisamy K. Using field inversion to quantify functional errors in turbulence closures. Phys Fluids 2016;28(4):045110.
[97] Sirovich L. Turbulence and the dynamics of coherent structures. I-Coherent structures. II-Symmetries and transformations. III-Dynamics and scaling. Q Appl Math 1987;45:561–71.
[98] Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 1997;11(4):341–59.
[99] Taira K, Brunton SL, Dawson ST, Rowley CW, Colonius T, McKeon BJ, et al. Modal analysis of fluid flows: an overview. AIAA J 2017;55:4013–41.
[100] Tallet A, Allery C, Leblond C. Optimal flow control using a POD-based reduced-order model. Numer Heat Transfer Part B 2016;70(1):1–24.
[101] Tracey B, Duraisamy K, Alonso J. Application of supervised learning to quantify uncertainties in turbulence and combustion modeling. In: 51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition; 2013. p. 259.
[102] Wan ZY, Vlachas P, Koumoutsakos P, Sapsis T. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. PloS One 2018;13(5):e0197704.
[103] Wang C, Liu J-G. Analysis of finite difference schemes for unsteady Navier–Stokes equations in vorticity formulation. Numerische Mathematik 2002;91(3):543–76.
[104] Wang Q, Hesthaven JS, Ray D. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. J Comput Phys 2019;384:289–307.
[105] Wang Z, Akhtar I, Borggaard J, Iliescu T. Two-level discretizations of nonlinear closure models for proper orthogonal decomposition. J Comput Phys 2011;230(1):126–46.
[106] Wang Z, Akhtar I, Borggaard J, Iliescu T. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. Comput Methods Appl MechEng 2012;237:10–26.
[107] Wang Z, Xiao D, Fang F, Govindan R, Pain CC, Guo Y. Model identification of reduced order fluid dynamics systems using deep learning. Int J Numer Methods Fluids 2018;86(4):255–68.
[108] Weller J, Lombardi E, Bergmann M, Iollo A. Numerical methods for low-order modeling of fluid flows based on POD. Int J Numer Methods Fluids 2010;63(2):249–68.
[109] Wells D, Wang Z, Xie X, Iliescu T. An evolve-then-filter regularized reduced order model for convection-dominated flows. Int J Numer Methods Fluids 2017;84(10):598–615.
[110] Widrow B, Rumelhart DE, Lehr MA. Neural networks: applications in industry, business and science. Commun ACM 1994;37(3):93–106.
[111] Xiao D, Fang F, Buchan A, Pain C, Navon I, Muggeridge A. Non-intrusive reduced order modelling of the Navier–Stokes equations. Comput Methods Appl MechEng 2015;293:522–41.
[112] Xiao D, Fang F, Pain C, Hu G. Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation. Int J Numer Methods Fluids 2015;79(11):580–95.
[113] Xiao D, Fang F, Pain C, Hu G. Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation. Int J Numer Methods Fluids 2015;79(11):580–95.
[114] Xiao D, Heaney C, Fang F, Mottet L, Hu R, Bistrian D, et al. A domain decomposition non-intrusive reduced order model for turbulent flows. Comput Fluids 2019;182:15–27.
[115] Xie X, Mohebujjaman M, Rebholz L, Iliescu T. Data-driven filtered reduced order modeling of fluid flows. SIAM J Sci Comput 2018;40(3):B834–57.
[116] Xie X, Wells D, Wang Z, Iliescu T. Approximate deconvolution reduced order modeling. Comput Methods Appl MechEng 2017;313:512–34.
[117] Xie X, Wells D, Wang Z, Iliescu T. Numerical analysis of the Leray reduced order model. J Comput Appl Math 2018;328:12–29.
[118] Xu K., Darve E. The neural network approach to inverse problems in differential equations. arXiv:190107758v1 2019.
[119] Zhang G, Patuwo BE, Hu MY. Forecasting with artificial neural networks: the state of the art. Int J Forecasting 1998;14(1):35–62.