Zenodo. https://zenodo.org/record/8381082 DOI: https://doi.org/10.5281/zenodo.8381082

Data-Driven Reduced-Order Modeling of Unsteady Nonlinear Shock Wave using Physics-Informed Neural Network (PINN) Based Solution

Amjad Ali*, Hamayun Farooq, Syeda Zahra Kazmi, Muhammad Zeeshan

Centre for Advanced Studies in Pure and Applied Mathematics (CASPAM)
Bahauddin Zakariya University (BZU), Multan, Pakistan.

*amjadali@bzu.edu.pk

Abstract

This article presents a preliminary study on data-driven reduced-order modeling (ROM) of unsteady nonlinear shock wave. A basic form of such problem can be modeled using the Burgers' equation. The physics-informed neural networks (PINN) approach is used to obtain numerical solutions to the problem at certain time steps. PINN is a cutting-edge computational framework that seamlessly integrates deep neural networks with the governing physics of the problem and is turning out to be promising for enhancing the accuracy and efficiency of numerical solutions in a wide array of scientific and engineering applications. Next, extraction of the Proper Orthogonal Decomposition (POD) modes from the solution field is carried out, providing a compact representation of the system's dominant spatial patterns. Subsequently, temporal coefficients are computed at specific time intervals, allowing for a reduced-order representation of the temporal evolution of the system. These temporal coefficients are then employed as input data to train a deep neural network (DNN) model designed to predict the temporal coefficient at various time steps. The predicted coefficient can be used to form the solution. The synergy between the POD-based spatial decomposition and the data-driven capabilities of DNN results in an efficient and accurate model for approximating the solution. The trained ANN subsequently takes the value of the Reynolds number and historical POD coefficients as inputs, generating predictions for future temporal coefficients. The study demonstrates the potential of combining model reduction techniques with machine learning approaches for solving complex partial differential equations. It showcases the use of physics-informed deep learning for obtaining numerical solutions. The idea presented can be extended to solve more complicated problems involving Navier-Stokes equations.

Keywords: Physics-Informed Neural Network (PINN), Artificial Neural Networks, Proper-Orthogonal Decomposition (POD), Data-Driven AI, Reduced-Order Modeling.

1. Introduction

In recent years, the synergy between advanced machine learning techniques and computational physics has paved the way for innovative approaches to solving complex scientific problems. One such groundbreaking methodology is the Physics-Informed Neural Network (PINN) (Raissi et al., 2019), which seamlessly integrates the principles of partial differential equations

(PDEs) with the capabilities of Artificial Neural Networks (ANNs). PINN is a machine-learning approach for solving problems involving Partial Differential Equations (PDEs). This novel framework has demonstrated remarkable success in approximating solutions to intricate PDEs, offering a powerful tool for understanding, modeling, and predicting the behavior of physical systems. PINN enables the network to learn from data while respecting underlying physical principles, making it a powerful tool for modeling and predicting intricate physical phenomena. PINNs have shown remarkable success in diverse applications, from fluid dynamics and material science to medical imaging and geophysics, offering a paradigm shift in how we approach scientific discovery and engineering optimization. By seamlessly integrating data-driven insights with the foundational laws of nature, PINNs enhance predictive accuracy and provide valuable insights into previously elusive physical phenomena, driving innovation and advancements in numerous scientific and engineering disciplines. Some valuable resources for gaining a deeper understanding of PINN are (Nascimento et al., 2020; Karniadakis et al., 2021; Cuomo et al., 2022).

Reduced Order Modeling (ROM) has emerged as a pivotal technique in the realm of computational science and engineering, revolutionizing our ability to tackle complex, high-dimensional problems efficiently and effectively. At its core, ROM aims to distill intricate systems into simplified representations without compromising accuracy, thereby offering a compelling solution to the computational bottlenecks often encountered in simulations. By leveraging various reduction strategies, ROMs enable the construction of surrogate models that capture the essential features of a system's behaviour, while significantly reducing the computational burden. ROMs find their application across diverse scientific disciplines, from fluid dynamics and structural mechanics to quantum chemistry. This versatile approach has witnessed exponential growth in recent years, driven by advancements in machine learning due to its capacity to significantly reduce CPU computational time while preserving the underlying physics (Taira et al., 2020).

Proper Orthogonal Decomposition (POD), analogous to Principal Component Analysis (PCA) in some fields, stands as a cornerstone technique in ROM. The fundamental idea behind POD is to distill the essential features of high-dimensional data into a reduced set of modes or basis functions. These modes, ranked in order of significance, represent the principal directions of variability within the data. It embodies a powerful method for capturing dominant patterns and structures within complex data sets, particularly in the context of fluid dynamics and dynamic systems (Holmes et al., 2012). Beyond fluid dynamics, POD finds extensive applications in fields such as structural mechanics, vibrational analysis, MEMS technologies, atmospheric sciences (where it's known as empirical orthogonal functions), wind engineering, acoustics, and neuroscience (Hemati et al., 2017). The method's triumph lies in its capacity to offer interpretable spatiotemporal decompositions of data, fostering a deeper understanding of complex phenomena (Berkooz et al., 1993; Chatterjee, 2000; Kerschen et al., 2002; Feeny, 2002; Kutz., 2013; Li et al., 2020). Machine learning techniques have been explored to infer POD coefficients in non-intrusive parametric ROMs (Ly et al., 2001; Bui-Thanh et al., 2004; Audouze et al., 2009; Swischuk et al., 2019). (San and Maulik, 2018) and (San et al., 2019) have utilized ANN to predict temporal coefficients of POD in a non-intrusive manner. (Wang

et al., 2018) employed long-short-term memory networks (LSTM) to predict future time steps of POD coefficients. Furthermore, ANN and LSTM have also been harnessed to develop closures for truncated POD modes (San et al., 2019; Ahmed et al., 2020). For instance, Murata et al., 2020) and (Fukami et al., 2019) adopted convolutional neural networks (CNNs) for data compression and the reconstruction of high-dimensional flow fields.

ANNs (Saeed et al., 2023) stand as a foundational pillar of modern machine learning and artificial intelligence, revolutionizing our approach to complex problem-solving. Inspired by the intricate web of neurons in the human brain, ANNs are computational models composed of interconnected nodes, or artificial neurons, organized into layers. Their inherent ability to learn and adapt from data, extracting intricate patterns and representations, has made them indispensable tools in a data-driven world. ANNs have evolved significantly over the years, spawning various architectures, including feedforward networks, recurrent neural networks (RNNs), convolutional neural networks (CNNs), and more, each tailored to specific tasks.

The present work revolves around a comprehensive approach that combines PINNs, POD, and ANNs to predict complex spatiotemporal systems accurately. PINNs are employed to obtain an approximate solution to the Burger, setting the stage for subsequent analyses. A key focus lies in extracting dominant coherent structures within the obtained solutions. Here, the POD technique comes into play, enabling the identification of essential modes that capture critical information about the system's behavior. These temporal modes, meticulously derived through the POD framework, serve as the foundation for our data-driven modeling endeavors. The heart of this approach lies in using a deep neural network to harness the temporal information gleaned from POD, constructing a predictive model that can anticipate future temporal coefficients with remarkable accuracy.

The rest of the article is organized as follows. Section 2 discusses the PINN solution of the model problem. Section 3 describes the development/extraction of POD modes and the preparation of temporal coefficient data for data-driven ROM activity. Section 4 discusses the development of ROM using a deep neural network. The results and discussion are presented in Section 5. Finally, the concluding remarks are given in Section 6.

2. PINN Solution of the Model Problem

PINNs (Raissi et al., 2019) represent a cutting-edge fusion of physics-based modeling and machine learning techniques, promising transformative solutions to complex problems across various scientific domains. PINN is a machine-learning approach for solving problems involving Partial Differential Equations (PDEs). It leverages neural networks' remarkable capacity for function approximation (Hornik and Stinchcombe, 1989). It is a mesh-free approach that modifies the solution of governing equations into an optimization problem. It merges the data from measurements and partial differential equations (PDEs) in ANN. It embeds the PDE in the loss function of the neural network. The loss function includes the initial and boundary condition and residual of PDE at selected points in the domain. This residual term serves as a penalty to constrain the acceptable solution space. PINN approximates the PDE solution by training a neural network to minimize the loss function, as shown in Fig. 2.1.

The input of the neural network are points within the integration domain. A key innovation with PINNs is the inclusion of a residual network that encodes the governing physics equations. The general form of a time-dependent PDE can be expressed as follows,

$$\mathcal{N}(u(X,t);\ \gamma) \quad = \quad f(x,t), \quad X \in \Omega, \mathrm{t} > 0 \tag{1}$$

$$B(u) \quad = \quad g(x,t) \quad \text{for all } t \text{ and } X \in \partial\Omega \tag{2}$$

Here, $\Omega \subset \mathbb{R}^d$ is the domain with boundary $\partial\Omega$ and $X = [x_1, x_2, \cdots, x_d]$ is a d-dimensional spatial coordinate vector. γ represents the parameters associated with the underlying problem physics. The function f characterizes the data of the problem, and $\mathcal{N}$ is a nonlinear differential operator. The initial condition can be regarded as a variation of the Dirichlet boundary condition within this spatiotemporal domain. Consequently, B is the operator that defines initial or boundary conditions pertinent to the problem, and g serves as the boundary function. These boundary conditions can manifest as the Neumann, Robin, Dirichlet, or even periodic conditions.

Within the framework of the PINN approach, the prediction of u is achieved through a neural network (NN). This neural network is characterized by parameters denoted as θ, leading to the generation of an approximation.

$$\hat{u}_\theta(x,t) \quad \approx \quad u(x,t) \tag{3}$$

Here, $\hat{u}_\theta(x,t)$ is a neural network. The neural network is trained to learn the optimal values of the parameters θ. These parameters are learned through the minimization of a loss function. The loss for PINN is defined as follows,

$$C \quad \approx \quad \omega_{\mathcal{N}} C_{\mathcal{N}} + \omega_B C_B + \omega_d C_{data} \tag{4}$$

Here, $\omega_{\mathcal{N}}$, ω_B, and ω_d are the weights associated with the model equation, boundary and initial conditions, and known data, respectively.

If the MSE loss function is used, then $C_{\mathcal{N}}, C_B$, and C_{data} can be written as follows,

$$C_{\mathcal{N}} \quad = \quad MSE_{\mathcal{N}} \quad = \quad \frac{1}{N_c}\sum_{i=0}^{N_c} |f(x_i,t_i)|^2 \tag{5}$$

$$C_B \quad = \quad MSE_B \quad = \quad \frac{1}{N_B}\sum_{i=0}^{N_B} |B(\hat{u}_\theta)|^2 \tag{7}$$

$$C_{data} \quad = \quad MSE_{data} \quad = \quad \frac{1}{N_d}\sum_{i=0}^{N_d} |\hat{u}_\theta(x_i,t_i) - u_i|^2 \tag{8}$$

Here, N_c, N_B, and N_{data} are collocation points, initial and boundary points, and data points as input for the neural network, respectively.

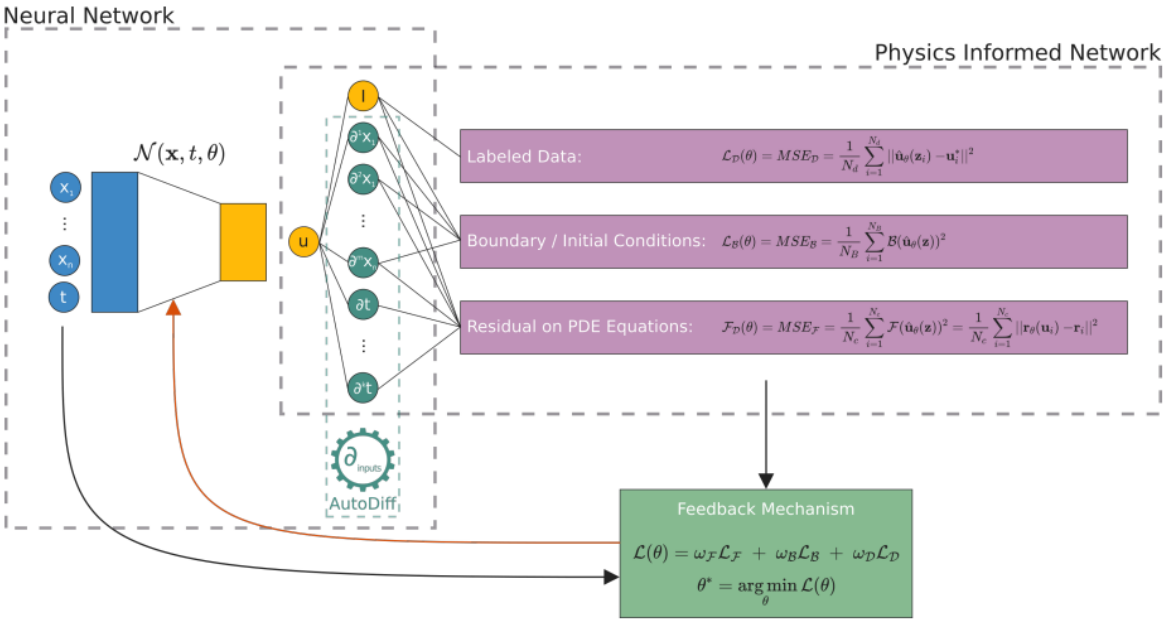


Fig. 2.1: Physics-informed neural network architecture (Cuomo et al., 2022)

The present study employs the viscous Burgers equation to approximate the solution using PINN. The Burger equation can be written as,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial t^2} \tag{9}$$

Here, Re is Reynold's number. The initial and boundary conditions are as follows,

$$u(x,0) = \frac{x}{1+\sqrt{\frac{1}{e^{1/8Re}} \times e^{x^2 Re/4}}} \tag{10}$$

$$u(0,t) = 0 \tag{11}$$

$$u(1,t) = \frac{1}{1+e^{3Re/16}} \tag{12}$$

3. Development of POD Modes

The Proper Orthogonal Decomposition (POD) method finds its application in the realm of Partial Differential Equations (PDEs) and constitutes an algorithmic process rooted in the principles of Singular Value Decomposition (SVD). It is quite a handy method for reducing dimensionality when analyzing spatiotemporal systems (Aubry et al., 1988; Amsallem and Farhat, 2012). Such systems often manifest as nonlinear PDEs governing the evolution of various quantities across time and space in diverse physical, engineering, and biological contexts. POD's success pivots on the widespread observation that complex systems tend to encode meaningful behaviors in low-dimensional patterns of dynamic activity. Therefore, POD

aims to capitalize on this phenomenon by generating low-rank dynamical systems that model the complete spatiotemporal behavior of the underlying complex system. ROMs utilize POD modes to project PDE dynamics onto low-rank subspaces, facilitating more efficient evaluations of the governing PDE model. These ROMs play a pivotal role in significantly enhancing computational speed, enabling tasks like computationally intensive Monte Carlo simulations of PDE systems, parameterized PDE system optimization, and real-time control of PDE-based systems. Generally, the POD modes can be derived through either the Singular Value Decomposition (SVD) or the method of snapshots (Sirovich and Kirby, 1987; Sirovich, 1989).

An ensemble of S time-discrete snapshots $(p_n = p(\mathbf{x}, t_n), t_n = t_s + (n-1)\Delta t, n = 1, 2, \cdots, S)$ serves as input for the POD analysis, focusing on properties like velocity in the flow. The time-dependent flow field p is decomposed into its mean $\bar{p}(\mathbf{x})$ and fluctuating part $p'(\mathbf{x}, t)$. The fluctuating component $p'(\mathbf{x}, t)$ is then expanded in a Galerkin fashion using temporal and spatial variables as follows:

$$p(\mathbf{x}, t) = \bar{p}(\mathbf{x}, t) + p'(\mathbf{x}, t) = \bar{p}(\mathbf{x}, t) + \sum_{j=1}^{\infty} a_j(t)\varphi_j(\mathbf{x}, t) \approx \bar{\tau}(\mathbf{x}, t) + \sum_{j=1}^{M} a_j(t)\varphi_j(\mathbf{x})$$

Here, M represents the finite number of modes in the expansion, $\varphi_j(\mathbf{x})$ are the POD modes obtained through an eigenvalue problem and $p_j(t)$ are the temporal coefficients. The objective of this study is to train an ANN model to determine the values ofa_j. Details regarding the ANN model setup can be found in section 2.3.

The instantaneous solutions at times $t_1, t_2, \cdots, t_s \in (0, T)$ of p are stored in a matrix U of size $N \times S$, where $S \ll N$ signifies the number of grid points. The goal is to identify a low-dimensional basis $\{\varphi_1, \varphi_2, \cdots, \varphi_M\}$ that minimizes the following objective while adhering to orthogonality constraints $\left(\varphi_i, \varphi_j\right)_{\mathcal{H}} = \delta_{ij}$,$1 \le i, j \le M$,

$$\min_{\varphi_j} \frac{1}{S} \sum_{i=1}^{S} \left\| \tau(., t_i) - \sum_{j=1}^{M} \left(p(., t_i), \varphi_j(.)\right)_{\mathcal{H}} \varphi_j(.) \right\|_{\mathcal{H}}^2$$

Here, δ_{ij} is the Kronecker delta function and $\mathcal{H}$ is a real Hilbert space with $(., t) \in \mathcal{H} \mid t \in (0, T)$. To address this, an eigenvalue problem is formulated:

$$\mathcal{M}v = \lambda v$$

where v_k for $k = 1, 2, \cdots, S$ are the eigenvectors, $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_S > 0$ are eigenvalues, and $U^T U = \mathcal{M} \in \mathcal{R}^{S \times S}$ is the correlation matrix that can be defined as,

$$\mathcal{M}_{ij} = \frac{1}{S}\left(p(., t_j), p(., t_i)\right)_{\mathcal{H}}$$

As demonstrated in (Sirovich, 1987), the solution can be expressed as:

$$\varphi_k(.) = \frac{1}{\sqrt{\lambda_k}} \sum_{j=1}^{S} (v_k)_j p(., t_j), \qquad 1 \le k \le M$$

4. Data-Driven Prediction of Temporal Coefficients (Development of ROM using DNN)

ANN (Saeed et al., 2023) are deep learning frameworks. ANN generates its output by mapping a set of inputs through a generalized variation of a linear function. The neurons are organized into layers, including input and output layers, with one or more hidden layers in between. The input layer takes the input data, and no computation is performed at the input layer. The subsequent layers consist of a sequence of neurons that carry out computations on their respective inputs and generate the output of hidden layers. In computation, the value of each neuron of the previous layer and weights are multiplied, and the bias term is added to generate the output neuron of a layer l. An activation function is applied to the output neuron of a layer. The hidden layers carry out computations hidden from the user's view. The last layer in the network is the output layer, typically featuring a linear activation function. Mathematically, the dot product of the neurons of the previous layer $x_i^{[l-1]}$ and weights $w_{ji}^{[l]}$ is taken and bias $b_j^{[l]}$ is added to generate the output neuron of layer l. The activation function φ is applied to the output of layer l.

$$z_j^{[l-1]} = \sum_{i=1}^{n_{l-1}} x_i^{[l-1]} . w_{ji}^{[l]} + b_j^{[l]}$$

Here, $j = 1,2,\cdots,n_l$ and n_l represents the number of neurons in a layer l.

The weights and biases are the trainable parameters of the neural network. A loss function is the distance between predicted and true values. It is a fundamental component in the training of neural networks. The primary objective during training is to minimize this loss.

Training a neural network is a crucial phase in its development, where the network learns to capture patterns and relationships within the provided data. During training, the network adjusts the parameters to minimize the loss function. This process involves an iterative optimization technique to update these parameters gradually. Gradient descent or its variants are used as the optimization technique. The neural network learns from its mistakes by comparing its predictions to the true values. The training process refines the ability of neural networks to make reasonably accurate predictions over time. Proper training also involves considerations such as selecting an appropriate loss function to quantify the prediction errors and fine-tuning hyperparameters, like learning rates, to optimize the training process. The training aims to enable the neural network to generalize its learned patterns to unseen data, ensuring its effectiveness in real-world applications.

The information is transmitted sequentially from consecutive layers in a unidirectional manner, progressing from input to output. The network assumes full connectivity, where every node in one layer is linked to those in the subsequent layer. The number of layers and neurons specifies the architecture of the network.

5. Results and Discussion

In the article, the Burger equation, given in Eq. (9), is solved using PINNs, implemented within the DeepXDE framework (Lu et al., 2021). The geometry of the problem is an interval in the x-domain. The time domain is specified $(0,1)$ to study the problem's evolution over time together with the Dirichlet boundary and initial conditions. The power of automatic differentiation (Raissi et al., 2019) is leveraged to compute derivatives efficiently. A feedforward neural network is employed with three hidden layers with 20 neurons in each hidden layer. The model is compiled using the Adam (Kingma and Ba, 2017) optimizer with a learning rate of $1.0\times10-3$. The training process is initiated for a specified number of iterations. Following initial training, the L-BFGS (Byrd et al., 1995) optimizer is used for further fine-tuning. We introduce an early stopping criterion to enhance convergence. The space-time solution of the Burgers equation at different values of the Reynolds number is shown in Fig. 2.2.

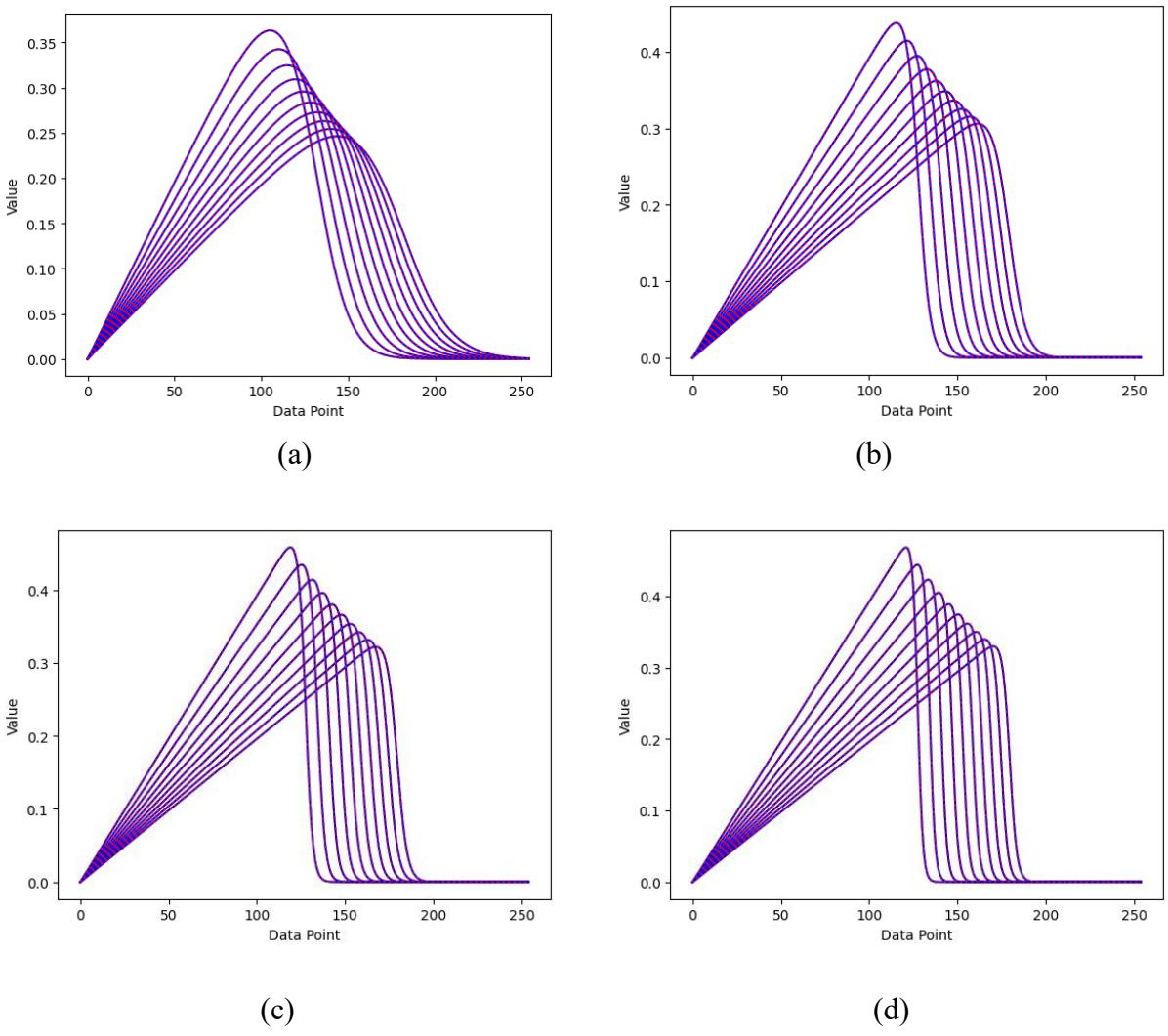


Fig. 2.2: Space-time solution of the Burgers equation for different values if the Reynolds number: (a) Re = 100, (b) Re = 300, (c) Re = 500, and (d) Re = 700

The PINN solution is used to generate POD for the viscus Burger equation. The SVD is preferred due to its computational efficiency. The POD modes are calculated using the solution of the Burger equation obtained using PINN. Ten temporal coefficients are obtained at 500 equally spaced time steps. These temporal coefficients are further used to generate the ANN-based ROM. The singular values and first four POD modes for Re = 100, 300, 700, and 900 are visualized.in Fig. 2.3 and 2.4, respectively.

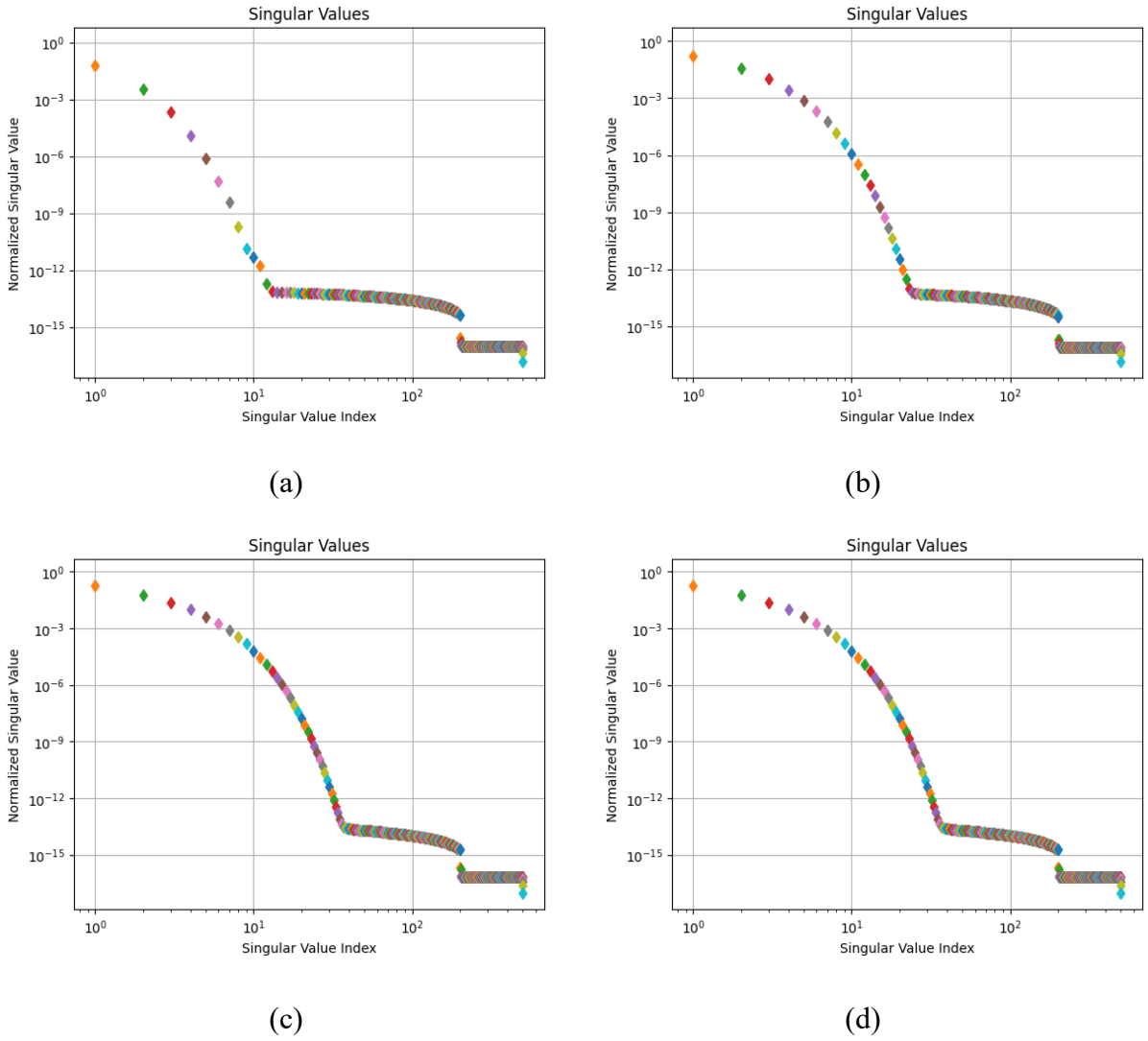
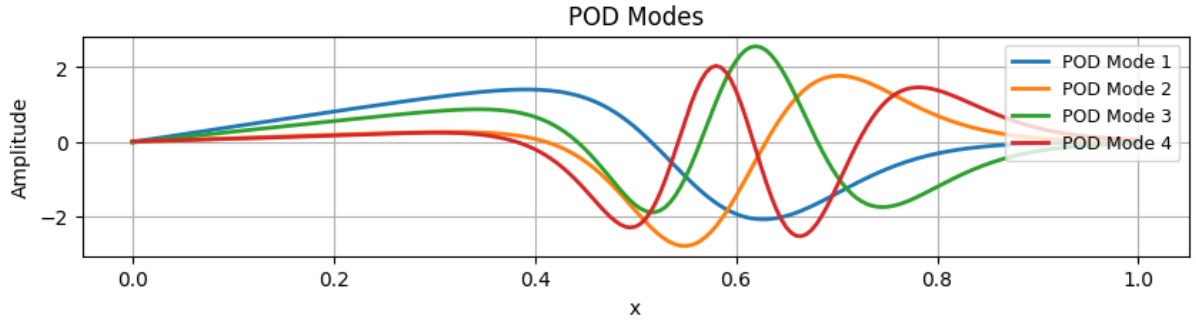
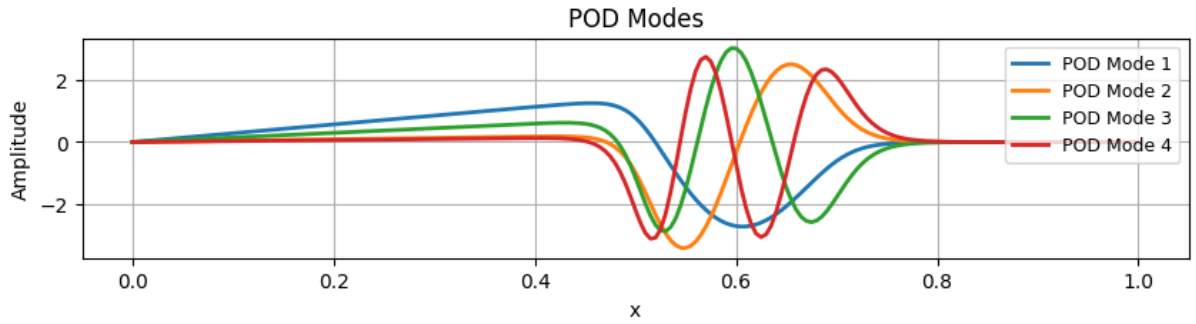


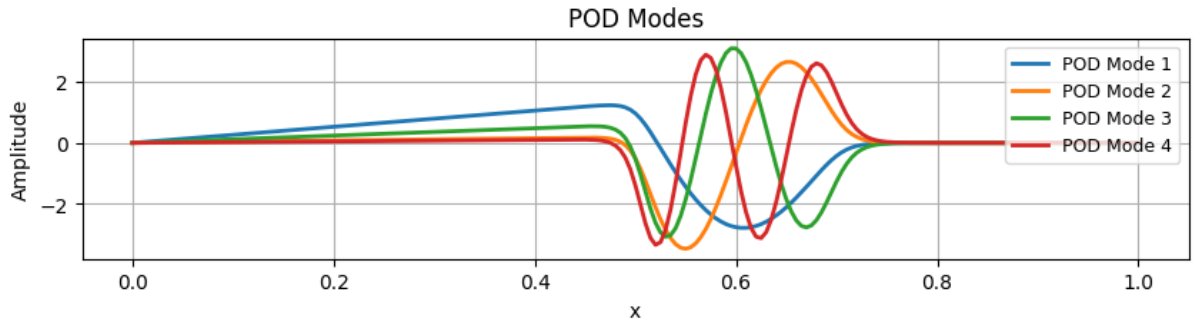
Fig. 2.3: Singular Values (a) Re = 100 (b) Re = 300 (c) Re = 500 (d) Re = 700



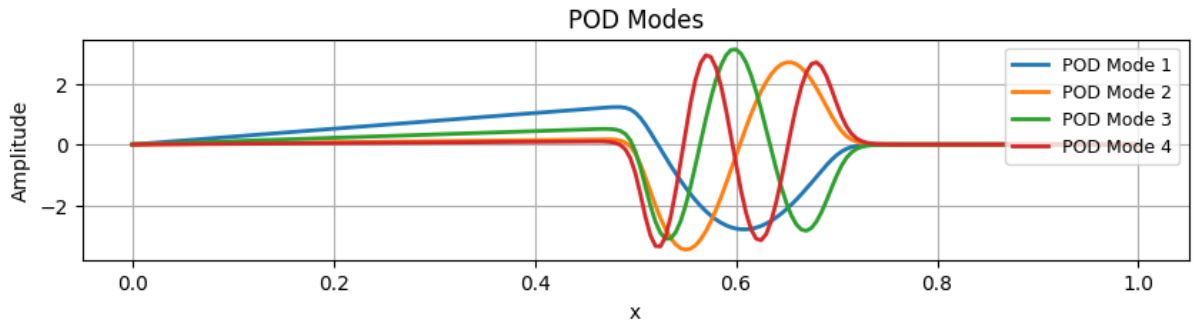
(a)



(b)



(c)

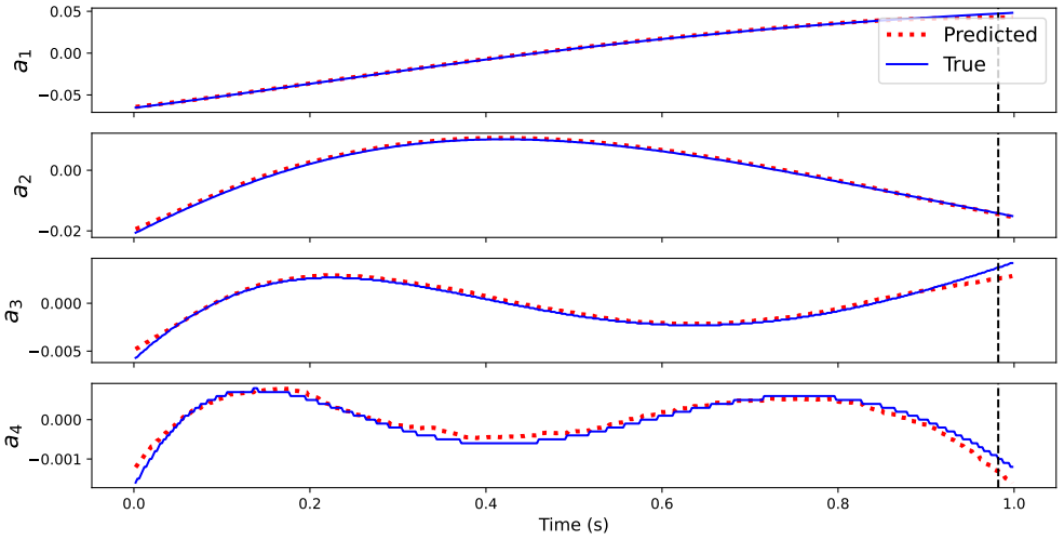


(d)

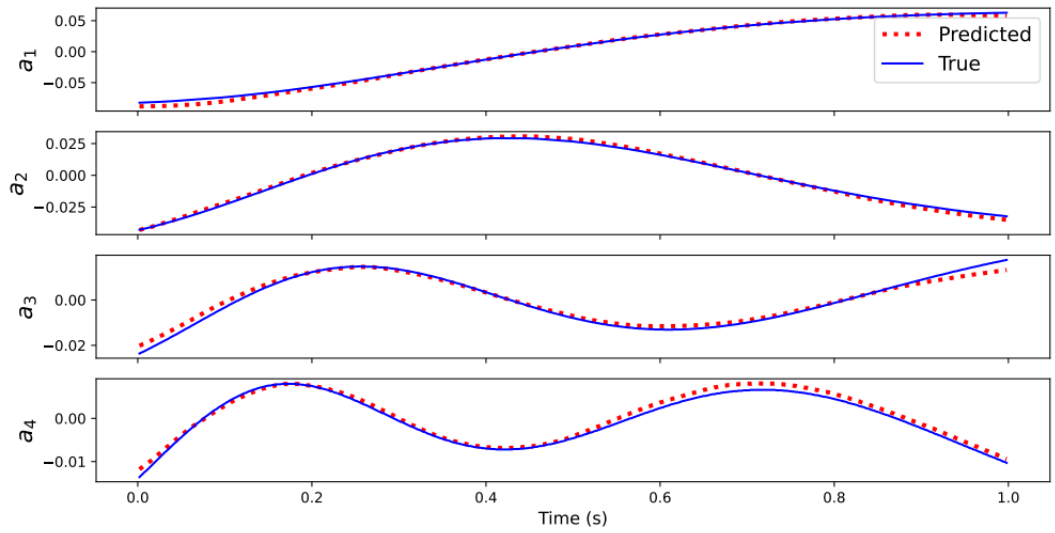
Fig. 2.4: First four POD modes: (a) Re = 100, (b) Re = 300, (c) Re = 500, and (d) Re = 700

In this study, a feedforward neural network (FNN) is used to establish a mapping from previous time steps to future time steps. The input data for the neural network comprises the temporal coefficients derived. The architecture consists of two hidden layers with 512 neurons in each layer. The ReLU activation function is applied in the hidden layer. The training process was executed using TensorFlow (Abadi et al., 2015), wherein the neural network was optimized to minimize the error between predicted and target temporal coefficients. The optimization procedure employed the Adam optimization algorithm (Kingma and Ba, 2017). To identify the optimal architecture for the ANN, Autokeras (Jin et al., 2019) was harnessed, which leverages Bayesian Optimization (Wu et al., 2019) to explore configurations such as the number of

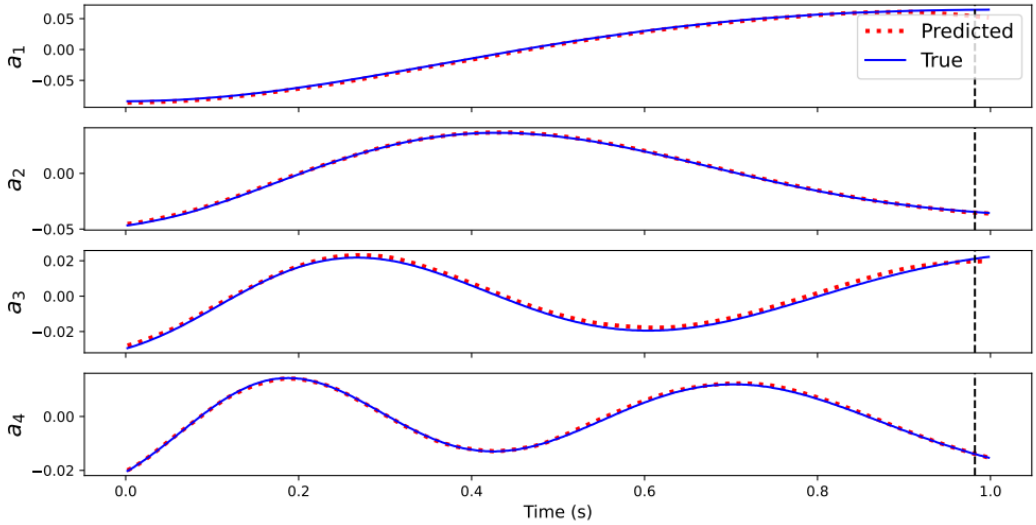
hidden layers and neurons. The training dataset is comprised of 490 time steps for each Reynolds number value at 100, 500, and 700, with an additional 30 time steps reserved for testing purposes. Notably, time coefficients associated with the Reynolds number at 300 were excluded from training. The model's capability to predict out-of-sample POD coefficients for Re = 300 is evaluated. Given a Reynolds number and past time steps, it can forecast future time steps. Furthermore, Fig. 3 illustrates the first four coefficients, demonstrating a strong agreement between the predictions and the actual data.



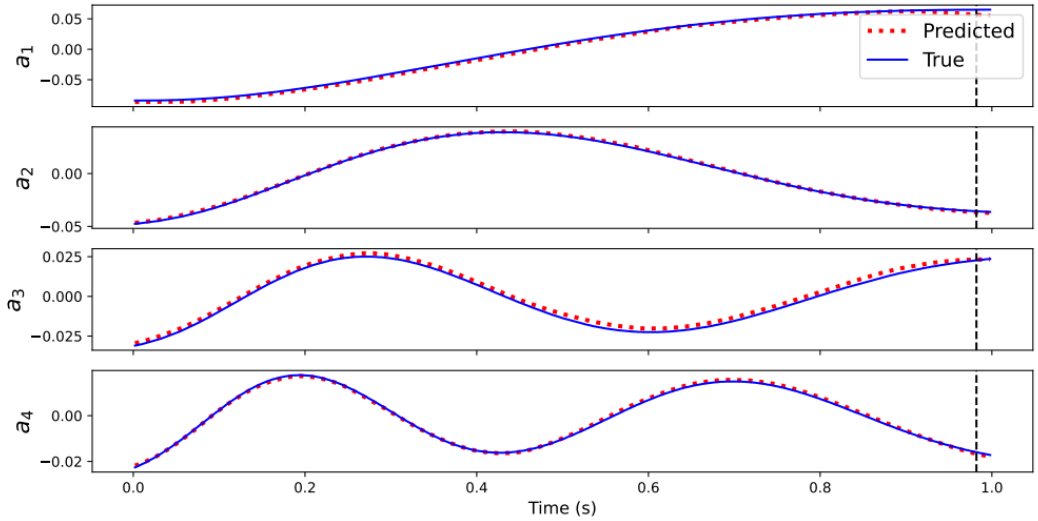
(a)



(b)



(c)



(d)

Fig. 3: First four temporal coefficients at different values of the Reynolds number: (a) Re = 100, (b) Re = 300, (c) Re = 500, and (d) Re = 700

6. Conclusion

In conclusion, this preliminary study highlights the promising synergy between Data-Driven Reduced-Order Modeling (ROM) techniques and the Physics-Informed Neural Networks (PINN) approach in tackling the Burgers' equation, which features a shock wave. In this study, the numerical solution is obtained using PINN at different values of Reynolds number. Next, our methodology involves the extraction of Proper Orthogonal Decomposition (POD) modes from the solution field, enabling a concise representation of the system's dominant spatial patterns. By subsequently computing temporal coefficients at specific time instants in the interval (0,1), we establish a reduced-order framework for capturing the temporal evolution of

the system. The utilization of these extracted POD coefficients as input data for training a deep neural network (DNN) model results in a powerful tool for predicting the solution of the model problem at various time steps. Additionally, the interpolation for out-of-sample data is conducted, specifically at the value of Reynolds number 300. Our findings reveal that the ANN model outperforms the conventional POD-ROM. This fusion of POD-based spatial decomposition and the data-driven capabilities of DNN proves efficient and accurate in approximating the solution, even in the presence of complex shock waves. Furthermore, our study underscores the broader potential of integrating model reduction techniques with machine learning approaches for solving intricate partial differential equations. The presented framework offers a promising avenue for addressing complex phenomena while balancing computational efficiency and predictive accuracy in a wide range of scientific and engineering applications.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J. M., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., . . . Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. arXiv (Cornell University). http://datascienceassn.org/sites/default/files/TensorFlow%20-%20Large-Scale%20Machine%20Learning%20on%20Heterogeneous%20Distributed%20Systems.pdf

Ahmed, S. E., San, O., Rasheed, A., & Iliescu, T. (2020). A long short-term memory embedding for hybrid uplifted reduced order models. Physica D: Nonlinear Phenomena, 409, 132471. https://doi.org/10.1016/j.physd.2020.132471

Amsallem, D., & Farhat, C. (2012). Stabilization of projection-based reduced-order models. International Journal for Numerical Methods in Engineering, 91(4), 358–377. https://doi.org/10.1002/nme.4274

Aubry, N., Holmes, P., Lumley, J. L., & Stone, E. (1988). The dynamics of coherent structures in the wall region of a turbulent boundary layer. Journal of Fluid Mechanics, 192, 115–173. https://doi.org/10.1017/s0022112088001818

Audouze, C., De Vuyst, F., & Nair, P. B. (2009). Reduced-order modeling of parameterized PDEs using time-space-parameter principal component analysis. International Journal for Numerical Methods in Engineering, 80(8), 1025–1057. https://doi.org/10.1002/nme.2540

Berkooz, G., Holmes, P., & Lumley, J. L. (1993). The proper orthogonal decomposition in the analysis of turbulent flows. Annual Review of Fluid Mechanics, 25(1), 539–575. https://doi.org/10.1146/annurev.fl.25.010193.002543

Bui-Thanh, T., Damodaran, M., & Willcox, K. (2004). Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. AIAA Journal, 42(8), 1505–1516. https://doi.org/10.2514/1.2159

Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing, 16(5), 1190–1208. https://doi.org/10.1137/0916069

Chatterjee, A. (2000). An introduction to the proper orthogonal decomposition. Current Science, 78(7), 808–817. http://www.iisc.ernet.in/currsci/apr102000/tutorial2.pdf

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next. Journal of Scientific Computing, 92(3). https://doi.org/10.1007/s10915-022-01939-z

Feeny, B. F. (2002). ON PROPER ORTHOGONAL CO-ORDINATES AS INDICATORS OF MODAL ACTIVITY. Journal of Sound and Vibration, 255(5), 805–817. https://doi.org/10.1006/jsvi.2001.4120

Fukami, K., Fukagata, K., & Taira, K. (2019). Super-resolution reconstruction of turbulent flows with machine learning. Journal of Fluid Mechanics, 870, 106–120. https://doi.org/10.1017/jfm.2019.238

Hemati, M. S., Rowley, C. W., Deem, E. A., & Cattafesta, L. N. (2017). De-biasing the dynamic mode decomposition for applied Koopman spectral analysis of noisy datasets. Theoretical and Computational Fluid Dynamics, 31(4), 349–368. https://doi.org/10.1007/s00162-017-0432-2

Holmes, P., Lumley, J. L., & Berkooz, G. (1996). Turbulence, coherent structures, dynamical systems and symmetry. https://doi.org/10.1017/cbo9780511622700

Hornik, K., Stinchcombe, M. B., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Jin, H., Song, Q., Hu, X. (2019). Auto-Keras: An Efficient Neural Architecture Search System. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, pp. 1946–1956.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Liu, Y. (2021). Physics-informed machine learning. Nature Reviews Physics, 3(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5

Kerschen, G., & Golinval, J. (2002). PHYSICAL INTERPRETATION OF THE PROPER ORTHOGONAL MODES USING THE SINGULAR VALUE DECOMPOSITION. Journal of Sound and Vibration, 249(5), 849–865. https://doi.org/10.1006/jsvi.2001.3930

Kingma, D.P., Ba, J. (2017). Adam: A Method for Stochastic Optimization, arXiv:1412.6980v9. https://arxiv.org/abs/1412.6980v9

Kutz, J. N. (2013). Data-Driven Modeling & Scientific Computation: Methods for complex systems & big data. http://ci.nii.ac.jp/ncid/BB17934337

Li, Z., Kovachki, N. B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A. M., & Anandkumar, A. (2020). Neural Operator: Graph kernel network for partial differential equations. arXiv (Cornell University). https://arxiv.org/pdf/2003.03485

Lu, L., Meng, X., Mao, Z., & Karniadakis, G. E. (2021). DeepXDE: a deep learning library for solving differential equations. Siam Review, 63(1), 208–228. https://doi.org/10.1137/19m1274067

Ly, H. V., & Tran, H. T. (2001). Modeling and control of physical processes using proper orthogonal decomposition. Mathematical and Computer Modelling, 33(1–3), 223–236. https://doi.org/10.1016/s0895-7177(00)00240-5

Murata, T., Fukami, K., & Fukagata, K. (2020). Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. Journal of Fluid Mechanics, 882. https://doi.org/10.1017/jfm.2019.822

Nascimento, R. G. D., Fricke, K., & Viana, F. a. C. (2020). A tutorial on solving ordinary differential equations using Python and hybrid physics-informed neural network. Engineering Applications of Artificial Intelligence, 96, 103996. https://doi.org/10.1016/j.engappai.2020.103996

Raissi, M., Perdikaris, P., & Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Saeed, T. R., Kazmi, S. Z., Ali, A., & Zafar, S. (2023). Demystifying ANN with Mathematical and Graphical Insights: An Algorithmic Review for Beginners. Zenodo (CERN European Organization for Nuclear Research). https://doi.org/10.5281/zenodo.8353418

San, O., & Maulik, R. (2018). Neural network closures for nonlinear model order reduction. Advances in Computational Mathematics, 44(6), 1717–1750. https://doi.org/10.1007/s10444-018-9590-z

San, O., Maulik, R., & Ahmed, M. (2019). An artificial neural network framework for reduced order modeling of transient flows. Communications in Nonlinear Science and Numerical Simulation, 77, 271–287. https://doi.org/10.1016/j.cnsns.2019.04.025

Sirovich, L. (1989). Chaotic dynamics of coherent structures. Physica D: Nonlinear Phenomena, 37(1–3), 126–145. https://doi.org/10.1016/0167-2789(89)90123-1

Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. Journal of the Optical Society of America, 4(3), 519. https://doi.org/10.1364/josaa.4.000519

Swischuk, R. C., Mainini, L., Peherstorfer, B., & Willcox, K. (2019). Projection-based model reduction: Formulations for physics-based machine learning. Computers & Fluids, 179, 704–717. https://doi.org/10.1016/j.compfluid.2018.07.021

Taira, K., Hemati, M. S., Brunton, S. L., Sun, Y., Duraisamy, K., Bagheri, S., Dawson, S. T. M., & Yeh, C. (2020). Modal Analysis of Fluid Flows: Applications and Outlook. AIAA Journal, 58(3), 998–1022. https://doi.org/10.2514/1.j058462

Wang, Z., Xiao, D., Fang, F., Govindan, R., Pain, C. C., & Guo, Y. (2018). Model identification of reduced order fluid dynamics systems using deep learning. International Journal for Numerical Methods in Fluids, 86(4), 255–268. https://doi.org/10.1002/fld.4416

Wu, J., Chen, X.Y., Zhang, H., Xiong, L.D., Lei, H., Deng, S.H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. J. Electron. Sci. Technol. 17, 26–40.

Conflicts of Interest: The authors have *no conflicts* of interest to declare that are relevant to the content of this article.