Burgers' Physics-Informed Surrogate leveraging Proper Orthogonal Decomposition

Nuno Costa
Numerical Analysis for Machine Learning, 2023/2024
Politecnico di Milano, Italy

1 INTRODUCTION

Accurately solving partial differential equations (PDEs) that govern complex systems, such as the Burgers' equation, is both a fundamental and challenging task. Traditional numerical methods, while precise, can be computationally intensive and time-consuming, especially for high-dimensional problems and complex geometries [4]. This limitation has spurred interest in the development of surrogate models that can approximate the solution of PDEs more efficiently. A particularly promising approach in recent years has been the integration of Proper Orthogonal Decomposition (POD) with neural networks to create lower-dimensional mappings that are computationally less demanding [2].

This project explores an innovative expansion of the POD-neural network surrogate model through the lens of Physics-Informed Neural Networks (PINNs) [5]. PINNs represent a breakthrough in computational science, integrating physical laws into the learning process of neural networks to ensure that the solutions are not only data-driven but also physically plausible. By incorporating physical knowledge in the lower-dimensional space created by POD, we aim to enhance the fidelity and efficiency of surrogate models further.

The focus of our investigation is on the inviscid Burgers' equation, a fundamental nonlinear PDE in fluid mechanics that models shock waves and traffic flow, among other phenomena [3]. The inviscid case of the Burgers' equation is particularly intriguing due to its property that the solution at any time can be expressed as a function of the initial condition, a characteristic that lends itself well to the approach of using physics-informed regularization in the learning process.

This report details the methodology of combining POD with physics-informed constraints in the neural network training process to enforce physical coherence in the solutions. By evaluating the model's performance across various training regimes—supervised, unsupervised, and mixed (combining both data fidelity and physical regularization)—we aim to ascertain the effectiveness of incorporating physical knowledge at a reduced dimensionality. Our results, which compare accuracy, residual norms, and computational efficiency across these training modalities, offer insights into the potential of physics-informed surrogate models in advancing CFD and other areas of computational physics.

2 PROPER ORTHOGONAL DECOMPOSITION

2.1 Overview of POD

Proper Orthogonal Decomposition (POD), also known as Principal Component Analysis (PCA) in the context of statistics, is a mathematical technique used to reduce the dimensionality of a system while preserving its most significant features [6]. In the context of the Burgers' equation, POD serves as an essential tool for constructing a lower-dimensional representation of the fluid state, which encapsulates the bulk of the system's energy or variance [2].

The method is applied on a series of snapshots — discrete spatial configurations of the system at various time intervals. These snapshots form a high-dimensional dataset, which is often too large for practical computations, especially when dealing with real-time or iterative processes.

2.2 The POD Process

The POD process seeks to find an orthogonal basis in which this high-dimensional data can be represented succinctly. This basis is composed of the eigenvectors (also known as modes) of the data's covariance matrix, which correspond to the most energetic or most varying directions in the data. By projecting the original data onto a limited number of these modes, we obtain a set of POD coefficients that provide a compressed representation of the system. This process is formalized as:

$$\mathbf{c}_\mu := \mathbf{V}^T \mathbf{u}_\mu \tag{1}$$

where $\mathbf{V}$ is the matrix whose columns are the POD basis vectors, and $\mathbf{u}_\mu$ is the state of the system parameterized by μ.

2.3 POD-NN: The Neural Network Extension

Once the POD basis is computed, the next challenge is to create a mapping from any new set of parameters μ to the corresponding POD coefficients. This is where neural networks come into play, providing a flexible and powerful framework for learning complex mappings.

The neural network, denoted by ϕ, is trained to predict the POD coefficients from the parameters. This training leverages both the efficiency of neural networks in handling high-dimensional data and the reduced dimensionality provided by the POD. The predicted coefficients can then be used to reconstruct an approximation of the system's state using the POD basis:

$$\mathbf{u}_\mu^{ROM} := \mathbf{V}\phi(\mu) \approx \mathbf{u}_\mu \tag{2}$$

This hybrid approach, known as POD-NN, retains the advantages of dimensionality reduction from POD while utilizing the expressive power of neural networks to interpolate within the reduced space. The result is a computationally efficient surrogate model that can quickly estimate the state of the system for new parameter values, facilitating real-time predictions and analyses that would be infeasible with a full-scale simulation.

3 PHYSICS-INFORMED NEURAL NETWORKS

3.1 The Traditional PiNN Approach

The concept of Physics-Informed Neural Networks (PiNNs) revolutionized the integration of machine learning with physical sciences by enforcing the laws of physics directly within the neural network architecture [5]. PiNNs are designed to respect the fundamental principles that govern the behavior of physical systems. This is achieved by incorporating differential equations as regularization terms in the training process, ensuring that the network output not only fits the data but also aligns with known physical laws.

3.2 An Illustrative Example: The Damped Harmonic Oscillator

To understand how PiNNs operationalize physical laws, consider the example of a damped harmonic oscillator. The second-order differential equation describes the motion of such an oscillator:

$$m\frac{d^2u}{dt^2} + \mu\frac{du}{dt} + ku = 0 \tag{3}$$

where m is the mass of the oscillator, μ is the damping coefficient, and k is the spring constant. The position of the oscillator $u(t)$ at any time t is the solution we seek.

In a PiNN setup, the neural network approximates this position with its output $\hat{u}_\theta(t)$, and the training process involves minimizing the discrepancy between the network's predictions and the actual physical behavior. This discrepancy, known as the residual $\mathcal{R}$, is given by:

$$\mathcal{R} = m\frac{d^2\hat{u}_\theta}{dt^2} + \mu\frac{d\hat{u}_\theta}{dt} + k\hat{u}_\theta \tag{4}$$

Ideally, $\mathcal{R}$ should be zero if $\hat{u}_\theta(t)$ accurately models the oscillator's motion. The residual is a crucial component of the loss function, reflecting the degree to which the network's predictions are physically plausible.

4 PROBLEM SETUP

4.1 The Burgers' Equation as a Test Bed

In the study of fluid dynamics, Burgers' equation serves as a fundamental test bed due to its simplicity and the rich set of behaviors it can exhibit [3]. It is a primary example of a nonlinear partial differential equation (PDE) and is given by:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{5}$$

for $u(x,t)$ in the domain Ω. In this project, we defined $\Omega = (-1,1)$, complemented with Neumann boundary conditions to ensure that the spatial derivative of the solution u at the boundaries remains zero, hence preventing flux across the domain boundaries.

4.2 Selection of Initial Conditions

The initial condition for u is crucial as it sets the stage for the evolution of the solution over time. We chose the following initial conditions:

$$\lambda(x) = \left(\frac{(1-\sin(\frac{\pi x^2}{\alpha}))}{16} + \frac{1}{8}\right) + 0.5 \tag{6}$$

$$u(x,0) = \psi(x) = \max(x+1,0)\cdot\lambda(x), \quad x \in \Omega \tag{7}$$

This specific form was selected to control the impact on the solution's evolution, ensuring the input velocity is stable, and the function's range is confined between 0 and 1 within the domain of interest. Setting the maximum to 0.5 at the left boundary ensures a stable starting point for the simulation. The sinusoidal component introduces variability across different values of α, allowing us to explore the solution space more comprehensively.

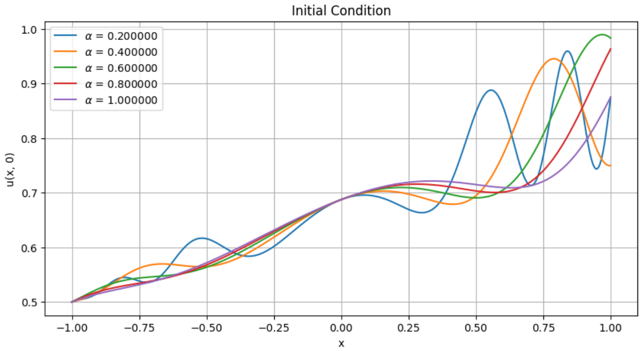

Figure 1: The initial conditions of the problem for different values of α.

4.3 Parameter Space and Data Generation

The parameter vector $\mu = [t,\alpha]$ spans a time-extended space $[0,2]\times[0.2,1]$, providing a broad range of scenarios under which the Burgers' equation behaves. This two-dimensional parameter space allows us to investigate the sensitivity of the solution to both temporal progression and the parameter α, which controls the initial condition's shape.

We leverage the computational framework provided by *phiflow* to generate high-fidelity data for training our models [7]. This approach significantly simplifies the data generation process. The generated dataset, therefore, constitutes a robust foundation for developing and testing our surrogate model, ensuring that the machine learning algorithms learn to predict solutions that are physically accurate and computationally efficient.

5 PINNS IN THE CONTEXT OF REDUCED SPACE

In a ROM environment, such as the POD-NN, we can apply the notions of PiNNs, resulting in a POD-PiNN. However, the direct application of PiNN principles is not straightforward due to the abstract nature of the reduced space. The derivatives within this space lack direct physical interpretation.

5.1 Residual in Reduced Space

For our inviscid Burgers' equation, the residual in the full space is derived based on the equation's intrinsic property. For a given parameter vector μ and its corresponding reduced-space prediction $\mathbf{c}_\mu$, the full-space solution $\mathbf{u}_\mu$ must satisfy:

$$\mathbf{u}(\mathbf{x},\mathbf{t},\alpha) = \psi(\mathbf{x} - \mathbf{u}(\mathbf{x},\mathbf{t},\alpha)\cdot\mathbf{t},\alpha) \tag{8}$$

The residual $\mathcal{R}$ for our problem is then:

$$\mathcal{R} = \mathbf{u}_\mu(\mathbf{x}) - \psi(\mathbf{x} - \mathbf{u}_\mu(\mathbf{x})\cdot\mu_t,\mu_\alpha) \tag{9}$$

Here, ψ represents the initial condition as a function of space x and the parameter α. We use μ_t and μ_α to denote the t and α components of the μ parametrization, respectively. By minimizing this residual, we encourage the neural network to learn a mapping that is consistent with the dynamics of the Burgers' equation despite operating in a reduced space.

5.2 Projecting the Residual Back to Full Dimensional Space

To apply physics-informed constraints in the reduced space, we first predict the POD coefficients $\mathbf{c}$ using our neural network. The predicted coefficients are then projected back to the full-dimensional physical space to obtain the predicted solution $\mathbf{u}_{pred}$ through the equation:

$$\mathbf{u}_{pred} = \mathbf{V}\mathbf{c}_{pred} \tag{10}$$

where $\mathbf{V}$ is the matrix containing the POD basis vectors.

Given the nature of the inviscid Burgers' equation, the residual $\mathcal{R}$ in the full-dimensional space is derived by comparing the predicted solution $\mathbf{u}_{pred}$ against the solution obtained by the characteristic equation of the Burgers' equation:

$$\mathcal{R} = \mathbf{u}_{pred} - \psi(\mathbf{x} - \mathbf{u}_{pred} \cdot \mu_t, \mu_\alpha) \tag{11}$$

5.3 Loss Function Incorporating Physics-Informed Regularization

For our ROM framework, the loss function is a combination of the mean squared error (MSE) between the predicted and target POD coefficients and the physics-informed residual:

$$\mathcal{L}(\theta) = \sum_i (\mathbf{c}_\theta(\mu_i) - \mathbf{c}_i^*)^2 + \alpha_0 \sum_j \mathcal{R}(\mu_j)^2 \tag{12}$$

where $\mathbf{c}_i^*$ are the target POD coefficients, $\mathbf{c}_\theta(t_i)$ are the predicted POD coefficients by the neural network, and α_0 is a hyperparameter.

In this formulation, the first term ensures the network accurately captures the dynamics in the reduced space, while the second term enforces the physical laws when projected back to the full-dimensional space. This dual-objective loss function underpins the training of our neural network, ensuring both data fidelity in the reduced space and physical plausibility in the full-dimensional space.

5.4 Connection Between Traditional and Reduced-Space PiNNs

This approach represents a valuable adaptation of the traditional PiNN framework. By ensuring that the physical integrity of the model is maintained in the full-dimensional space, we can enjoy the computational efficiency of reduced-order models while still providing solutions that are physically informed. The ability to project and apply physical constraints in full space offers a novel perspective on leveraging the power of PiNNs in scenarios where high-dimensional problems necessitate a more computationally tractable approach.

6 RESULTS

For our investigation, we selected a configuration that was both computationally efficient and capable of capturing the essential dynamics of the Burgers' equation. Specifically, we utilized 20 modes in the POD, which were found to sufficiently capture the variance in the system while keeping the dimensionality low. This would set a minimum error for the perfect reconstruction at 0.03%.

The parameter space spanned by α was discretized into 10 evenly spaced values in the interval $[0.2, 1]$. This discretization allowed us to cover a representative range of the initial condition's shape and ensured a robust evaluation of the model's performance across different physical scenarios. The simulation was, still, not perfect, and thus the minimum possible residual — the residual of the simulation results — was set at 0.0007.

The neural network used in the POD-PiNN framework consisted of an input layer, one hidden layer with 100 neurons, and an output layer. The hidden layer was equipped with a ReLU activation function to introduce non-linearity into the model, which is crucial for capturing the complex dynamics governed by the Burgers' equation. The neural network was implemented leveraging the *dlroms* package [1], which is a deep learning library for model order reduction that leverages on packages such as *Pytorch* and *FEniCS*.

6.1 Training Time

From the training time chart present in Figure 2, there is a discernible disparity between the supervised approach and the other methods that integrate residual calculations into the loss function.

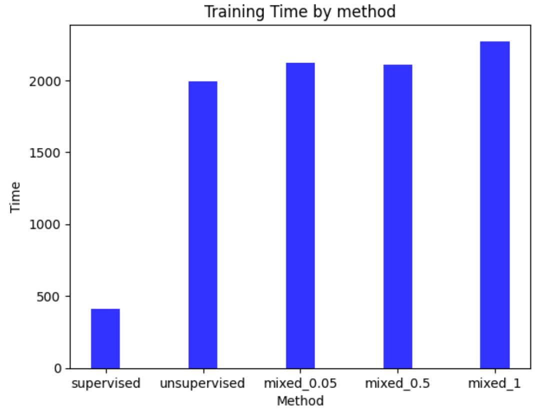

Figure 2: Time for the Solely supervised, Unsupervised, and some mixed approaches, with varying weights, of POD-PiNN.

The supervised method's markedly shorter training time corroborates the expectation that excluding the residual calculation expedites the training process. This aligns with the understanding that incorporating the physics-informed component into the training regime significantly extends training duration due to the additional complexity of calculating the residuals. Notably, the unsupervised method registers slightly faster training times compared to the mixed methods, which is anticipated because the unsupervised approach does not compute the MSE term. The discrepancies in training times among the mixed methods are likely due to variability in computational processing, and no definitive conclusion can be drawn about the influence of different weights on the training time.

6.2 Accuracy

The accuracy plot in Figure 3 reveals that the supervised method attains a plateau in accuracy swiftly, indicating an early convergence at around 400 epochs. However, the introduction of regularization transforms the optimization landscape, introducing complexity and variability, particularly noticeable in the mixed and unsupervised approaches. The mixed approach with $\alpha_0 = 1$ ultimately achieves the highest accuracy, implying that a heavier weighting on the physical regularization could guide the network towards a more accurate representation of the underlying physics. Conversely, the mixed approach with $\alpha_0 = 0.05$ illustrates the intricacies of PiNN optimization, showcasing erratic progress that suggests potential for further accuracy improvements beyond the observed epochs.

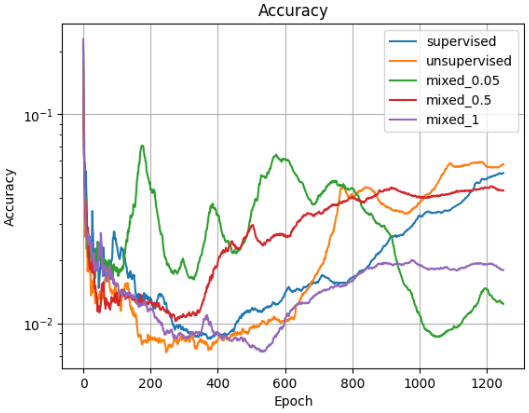

Figure 3: Validation Accuracy for the Solely supervised, Mixed and Unsupervised approaches of POD-PiNN.

6.3 Residual Error

The residual error plot in Figure 4 offers a window into the learning stability of each method. The supervised approach exhibits an initial decline in residual error, reaching its minima around the 400th epoch, after which the error starts to increase. This upsurge in residual error could signify overfitting to the training data, with the network potentially losing its generalization capability over the physical laws governing the system. In stark contrast, the unsupervised and mixed methods, particularly with higher weights, show a decreasing or stabilizing trend in residual error. The mixed method with $\alpha_0 = 1$ not only reaches the minimum residual more swiftly than its counterparts but also maintains a downward trajectory, suggesting ongoing optimization and learning of the physical dynamics. The fluctuations observed in the mixed approach with $\alpha = 0.05$ are emblematic of the challenges in PiNN optimization, where the model struggles to reconcile the competing demands of data fidelity and physical law adherence.

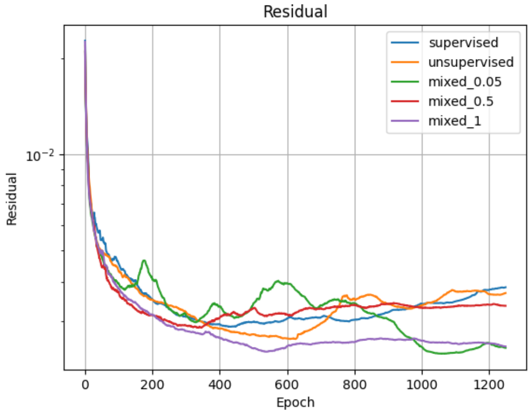

Figure 4: Residual for the Solely supervised, Mixed and Unsupervised approaches of POD-PiNN.

6.4 Overall Synthesis

Synthesizing these insights, it becomes evident that each training approach offers a unique balance of advantages and challenges. The supervised approach is efficient but may be prone to deviation from physical accuracy over prolonged training. The unsupervised method, reliant on physical laws, could potentially provide better generalization but at the cost of longer and less stable training. The mixed methods navigate these trade-offs, with the weight parameter playing a crucial role in dictating the optimization behavior and learning efficacy.

Ultimately, these results highlight the nuanced interplay between computational efficiency, optimization stability, and physical accuracy. They underscore the importance of carefully calibrating the training process to harness the full potential of PiNNs for solving complex PDEs. The insights point towards ongoing refinement of the weight parameter within the mixed training regime, aiming to optimize both the accuracy and the residual behavior for robust model performance.

7 FINAL REMARKS AND FUTURE WORK

The exploration into the realm of ROMs for solving the inviscid Burgers' Equation delivered some very valuable insights. Our investigation has demonstrated the benefits of embedding physical knowledge into the training of neural networks. Through the use of a physics regularizer, we've seen a non-negligible enhancement in model fidelity. This improvement, however, does come with the trade-off of prolonged training durations. Nonetheless, the mixed-method approaches, especially with a carefully calibrated weighting of the physics-regularizer, emerge as a particularly compelling strategy. These approaches adeptly balance accuracy and physical fidelity, presenting a viable path forward in the development of machine learning models for scientific applications.

This balance affirms the integral value of domain knowledge in machine learning. In areas where the precision of physical laws cannot be overlooked, such as fluid dynamics and thermodynamics, this methodology has great potential, as it offers a blueprint for constructing models that are not just data-efficient but also informed by the underlying physical principles they seek to emulate.

Looking ahead, one intriguing avenue for further exploration is the development of a post-processing pipeline that leverages the unique characteristics of the inviscid Burgers' Equation. Given the equation's property that the solution at any given time can be expressed as a function of the initial condition, we could recognize instances where the model's fidelity might be compromised and apply proper corrections. With this, we can potentially refine the predictions of our PiNN models further, resulting in an even better ROM.

Such a pipeline could significantly enhance the utility of ROMs in practical scenarios. By systematically addressing discrepancies between the predicted and expected solutions, this post-processing step could render the model's outputs not just more accurate but also more trustworthy from a physical standpoint. It would effectively serve as a bridge between the raw predictive power of neural networks and the nuanced requirements of physical systems modeling.

8 ACKOWLEDGEMENTS

I, Nuno Costa, extend my heartfelt gratitude to Dr. Nicola Rares Franco for his invaluable support in shaping this project and assisting me in overcoming implementation challenges.

REFERENCES

[1] Nicola Rares Franco. 2023. DLROMs: a Python package for Deep Learning based Reduced Order Models. https://github.

com/NicolaRFranco/dlroms.

[2] Carmen Gräßle, Michael Hinze, and Stefan Volkwein. 2020. Model Order Reduction by Proper Orthogonal Decomposition. arXiv:1906.05188 [math.NA]

[3] Peter J. Olver. 2020. *Introduction to partial differential equations*. Springer.

[4] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. 2006. *Numerical Mathematics*. Springer Science & Business Media.

[5] M. Raissi, P. Perdikaris, and G. E. Karniadakis. 2019. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* 378 (Feb. 2019), 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

[6] Gilbert Strang. 2019. *Linear Algebra and learning from data*. Wellesley-Cambridge Press.

[7] TUM. 2023. phiflow. https://github.com/tum-pbs/PhiFlow.