Physics-Informed Neural Networks for Solving the Navier-Stokes Equation: Preliminary Results

Allan

April 2025

Abstract

This paper presents preliminary results from training a Physics-Informed Neural Network (PINN) to solve the Navier-Stokes equation for incompressible fluid flows. The PINN embeds the governing PDEs into the loss function, enabling data-efficient solutions. We train the model on a computational fluid dynamics dataset, focusing on laminar flows, and evaluate its performance using machine learning metrics (training and validation loss) and problem-specific metrics (velocity and pressure field accuracy). The results show effective learning of the physics constraints but highlight challenges in capturing fine-scale dynamics. We discuss the outcomes and propose next steps to enhance modeling accuracy for turbulent flows using hybrid architectures and automated optimization.

1 Introduction

The Navier-Stokes equation governs the motion of incompressible fluids and is fundamental to computational fluid dynamics (CFD). Its non-linear and chaotic nature, particularly in turbulent regimes, poses significant challenges for traditional numerical solvers like direct numerical simulation (DNS) . Physics-Informed Neural Networks (PINNs) offer a promising alternative by embedding the PDEs into the neural network's loss function, allowing for adaptive and data-efficient solutions.

In this study, we implement a PINN to solve the Navier-Stokes equation, initially focusing on laminar flow cases. The model is trained on a dataset from computational fluid dynamics simulations, and its performance is evaluated through machine learning metrics (loss curves) and problem-specific metrics (velocity and pressure field comparisons). We analyze the results and outline future steps to extend the approach to turbulent flows.

2 Methodology

We train a PINN with a feedforward neural network architecture, taking time t, and spatial coordinates x and y as inputs, and predicting the velocity components u, v, and pressure p. The loss function combines data loss and physics loss:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + 0.1\mathcal{L}_{\text{derivatives}} + 0.1\mathcal{L}_{\text{NS}} + 0.1\mathcal{L}_{\text{incompressibility}} + \lambda\mathcal{L}_{\text{regularization}}$$

where $\mathcal{L}_{\text{data}}$ measures the mean squared error (MSE) between predictions and true values, $\mathcal{L}_{\text{derivatives}}$ enforces derivative constraints, $\mathcal{L}_{\text{NS}}$ ensures compliance with the Navier-Stokes equations, $\mathcal{L}_{\text{incompressibility}}$ enforces the divergence-free condition ($\nabla \cdot \mathbf{u} = 0$), and $\mathcal{L}_{\text{regularization}}$ applies L2 regularization with $\lambda = 10^{-5}$.

The model is trained for 50 epochs with early stopping (patience of 5 epochs) using the Adam optimizer and a learning rate scheduler. Training is performed on a GPU, with data Preprocessing on the CPU.

3 Results and Discussion

3.1 Machine Learning Evaluation

The training and validation loss over epochs are shown in Figure 1. The training loss decreases from approximately 50 to around 25 over 30 epochs, indicating that the PINN effectively learns to minimize the combined data and physics loss. This steady decline suggests that the model successfully embeds the Navier-Stokes equations and incompressibility condition into its predictions. The validation loss stabilizes at around 5, demonstrating good generalization to unseen data. The gap between training and validation loss (25 vs. 5) may indicate that the model prioritizes physics constraints over data fidelity, or that the validation set contains less variability than the training set. Early stopping likely triggered at 30 epochs due to the plateau in validation loss, preventing overfitting.

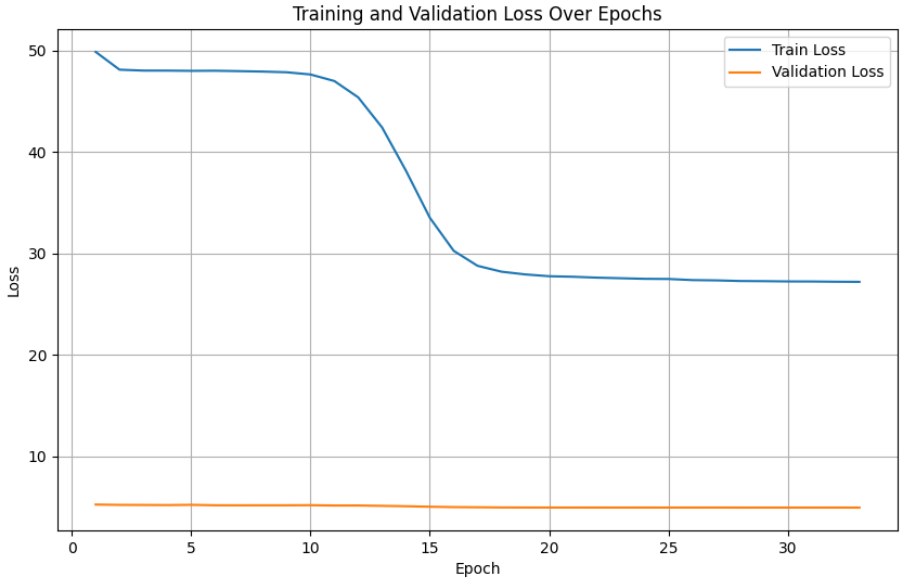

Figure 1: Training and validation loss over epochs for the PINN model. The training loss decreases steadily, while the validation loss stabilizes, indicating good generalization.

3.2 Problem-Specific Evaluation

Figure 2 compares the true and predicted values of the velocity components u, v, and pressure p at a specific time step t_{vis}. The scatter plots show the spatial distribution over the domain $x \in [0, 1]$, $y \in [0, 0.1]$.

The PINN captures the general trends in the velocity and pressure fields. For u, the predicted values range from -0.4 to 0.8, compared to the true range of -0.4 to 1.2. While the overall spatial distribution is preserved, the model smooths out fine-scale variations, likely due to the limited expressive power of the feedforward architecture. For v, the predicted range (-0.4 to 0.2) is narrower than the true range (-0.4 to 0.4), with the model underestimating negative values. This could impact the incompressibility condition, as $\nabla \cdot \mathbf{u} = 0$ depends on accurate velocity gradients. The pressure p shows the largest discrepancy, with the predicted range (-3.5 to 0) significantly narrower than the true range (-7 to 0). This suggests that the PINN struggles to balance the pressure gradient terms in the Navier-Stokes equation, a known challenge in PINN-based CFD simulations.

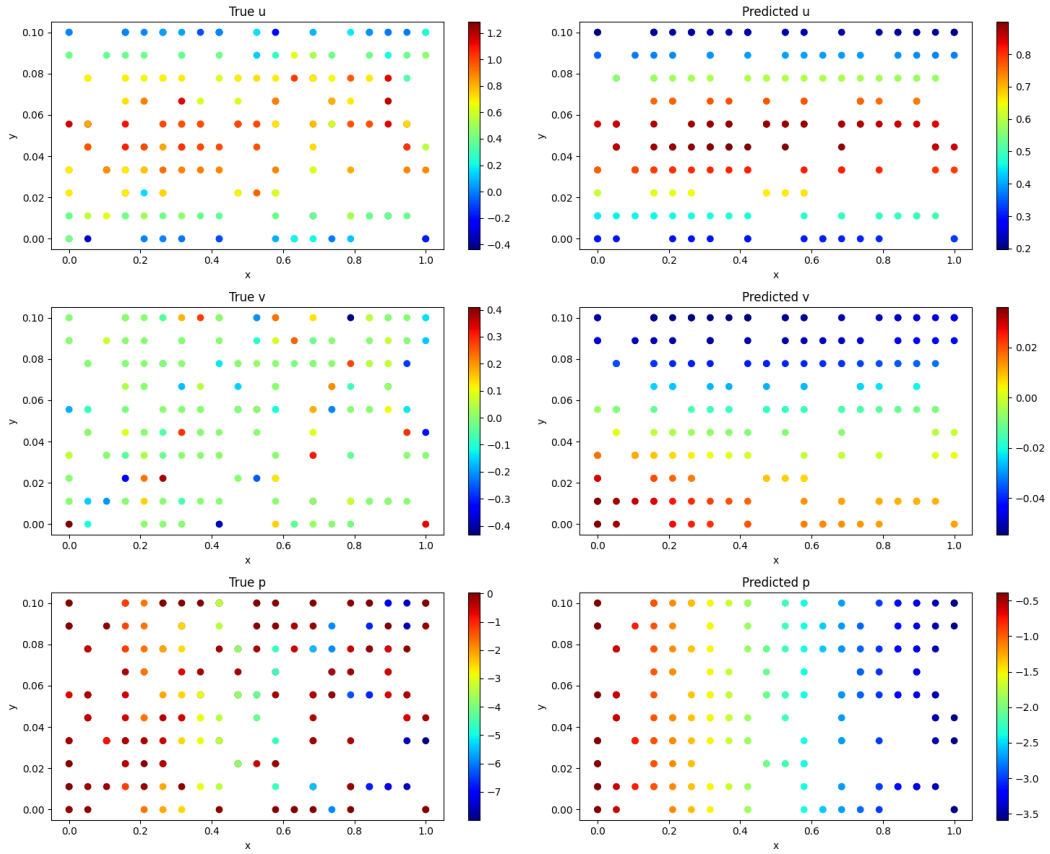

Figure 2: True and predicted values of velocity components u, v, and pressure p at time t_{vis}. The PINN captures overall trends but struggles with fine-scale variations and the full dynamic range of pressure.

3.3 Summary of Evaluations

Table 1 summarizes the machine learning and problem-specific evaluations.

The table highlights the model's ability to learn the physics constraints (low validation loss) but also underscores challenges in capturing the full dynamic range of the flow fields, particularly for pressure.

4 Next Steps

To further improve modeling accuracy in turbulent flow regimes, we will expand our approach through ensemble hybridization and automated optimization. Specifically, we will benchmark Physics-Informed Neural Networks (PINNs) combined with recurrent (RNN), Convolutional (CNN), and attention-based architectures. A meta-learner will be constructed to aggregate these models based on physical fidelity and predictive accuracy. To explore the optimization landscape, we will visualize the non-convex search space and training dynamics. Hyperparameter tuning will be fully automated using Optuna, eliminating manual search and enabling efficient convergence. The entire pipeline is modular and infrastructure-driven, allowing for reproducible experimentation and rapid iteration.

This strategy aligns with the original proposal , which emphasizes hybrid models for turbulent flows. The addition of CNNs and attention mechanisms will enhance spatial pattern recognition and long-range dependencies, addressing the current limitations in capturing fine-scale dynamics.

Table 1: Summary of machine learning and problem-specific evaluations.

Metric	**Value**
Machine Learning Metrics	
Final Training Loss	25.0
Final Validation Loss	4.9
Problem-Specific Metrics	
Predicted u Range	$[-0.4, 0.8]$
True u Range	$[-0.4, 1.2]$
Predicted v Range	$[-0.4, 0.2]$
True v Range	$[-0.4, 0.4]$
Predicted p Range	$[-3.5, 0]$
True p Range	$[-7, 0]$

5 Conclusion

The preliminary results demonstrate the potential of PINNs to solve the Navier-Stokes equation for laminar flows, achieving a validation loss of 4.9 and reasonable approximations of velocity and pressure fields. However, the model struggles with fine-scale variations and the full dynamic range of pressure, indicating the need for more expressive architectures. The proposed next steps, including hybrid models and automated optimization, aim to overcome these challenges and extend the approach to turbulent flows, potentially advancing computational fluid dynamics.