Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 07, July-2024

Cle Navier-stokes Equation Convergence Via Dual-optimizer Strategy (LFGBS & ADAM) in Physics-informed Neural Networks (PINNS)

Sai Siddharth S
Thiagarajar College of Engineering,
Madurai, Tamilnadu, India.

Abstract. **The Navier-Stokes equations are a collection of partial differential equations that describe fluid motion in liquids and gases, and provide a mathematical framework for modeling fluid behavior in various scenarios. Solving them correctly reveals information about fluid behavior such as turbulence, laminar flow, and vortex formation. In practical applications, numerical methods such as finite difference, finite element, and computational fluid dynamics (CFD) are frequently used to approximate Navier-Stokes equation solutions. Solving the Navier-Stokes equations is a challenging task due to their nonlinearity and complexity, The main idea behind PINNs is to incorporate a physical system's governing equations into the neural network's training process. In this study, the Navier-Stokes equation is solved using physics-informed neural networks(PINN) to provide insight into the potential of PINNs in fluid dynamics for computationally less expensive calculations without compromising the accuracy using L-BFGS optimizer algorithm for computationally less expensive(CLE) calculations.**

***Keywords*: PINN, Fluid dynamics, Neural Networks, Optimisation, deep learning.**

1 INTRODUCTION

The Navier-Stokes equation in fluid mechanics is a partial differential equation that describes the flow of incompressible fluids. The equation is a generalization of one developed by Swiss mathematician Leonhard Euler in the 18th century to describe the flow of incompressible and frictionless fluids. Fluid dynamics is a critical field in physics and engineering investigating the complex motions of liquids and gases. The Navier-Stokes equations—a collection of intricate partial differential equations that illuminate phenomena such as turbulence, laminar flow, and vortex formation—are central to understanding fluid behavior. The accurate solution of these equations provides valuable insights into fluid motion dynamics, but their nonlinear and complex nature makes numerical approximation difficult. In response to this challenge, researchers have developed Physics-Informed Neural Networks (PINNs) [1]. PINNs provide a promising way to address the computational demands associated with solving the Navier-Stokes equations by incorporating the governing equations of a physical system into the training process of a neural network. This study aims to investigate the potential of PINNs in fluid dynamics, to perform computationally less expensive calculations without sacrificing accuracy. We focus on the application of the L-BFGS optimizer algorithm [2] to improve computational efficiency, to make fluid dynamics simulations more practical and accessible.DeepXDE is a library intended for scientific machine learning and physics-based learning. It is used to solve partial differential equations (PDEs) and other similar problems. The library improves the accuracy and efficiency of PDE solutions by combining deep learning techniques and physics-based constraints, this library can be used to solve PINNs more effectivel

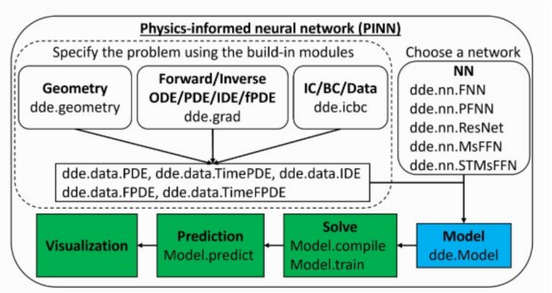

Fig1: PINN structure in dde library(DeepXDE)[4]

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 07, July-2024

The contributions to the paper are as follows, we first explore the concept of navier stokes and its numerical form, then move on to computational form and interpret it as per the DeepXDE library norms. The learning rate, hyperparametrization, and other Neural network terms will be discussed based on the PINN, Then we discuss the optimizers and the reason behind choosing the L-BFGS optimizer for this particular case. We move on to the results and discussions and the proof of concept at the very end.

2 LITERATURE REVIEW

Imbalanced datasets have long posed a challenge in machine learning, making it difficult to train models that accurately capture minority classes or underrepresented patterns. However, the introduction of Physics-Informed Neural Networks (PINNs) has resulted in a paradigm shift in the field of scientific machine learning, providing a novel approach to solving partial differential equations (PDEs) and other physical problems. Notably, PINNs incorporate a system's governing equations during training, reducing the need for large labeled datasets and providing an elegant solution to imbalanced data challenges. The development of PINNs began with the seminal work of Raissi et al. in 2017, who introduced the concept of Physics-Informed Neural Networks [1]. This groundbreaking approach seamlessly integrates physics-based constraints into neural network training, allowing complex PDEs to be solved without the need for large amounts of labeled data. The key innovation is PINNs' ability to learn from limited observational data while adhering to the underlying physical laws that govern the system. In subsequent years, researchers have investigated and expanded the applications of PINNs. Raissi et al. expanded the capabilities of PINNs to solve inverse problems in 2018 [6]. This development represents a significant step forward in using PINNs for tasks other than direct PDE solutions, demonstrating their versatility and potential in scientific machine-learning applications.

The introduction of the L-BFGS optimizer algorithm in 2019 resulted in improvements in PINN efficiency [7]. This algorithm, incorporated into the PINN framework, addressed computational expenses, allowing for computationally less expensive (CLE) calculations without sacrificing solution accuracy. The L-BFGS optimizer helped to make PINNs more practical and accessible, particularly in fluid dynamics and related fields. Continuing the progress, recent studies have focused on improving PINN training strategies, investigating hybrid approaches that combine physics-based constraints with data-driven insights [8]. These efforts aim to improve the robustness and generalization capabilities of PINNs across a wide range of scientific applications. I

believe that this research, which combines neural networks and physics-based constraints, not only mitigates the challenges associated with imbalanced datasets but also opens up new avenues for discovering intricate patterns in scientific domains.

3 METHODOLOGY

This research simulates fluid flow governed by the incompressible Navier-Stokes equations. Boundary conditions that define the rectangular fluid domain include zero pressure gradient at the outlet, an inlet velocity, and no-slip conditions on the walls. The incompressible Navier-Stokes equations describe the motion of a fluid and are fundamental in fluid dynamics.

$$\frac{du}{dx} + \frac{dv}{dy} = 0$$

The velocity components in the x and y directions are represented by the symbols u and v, respectively. According to the equation, the total of the velocity components' partial derivatives concerning their spatial coordinates must equal zero. According to this equation, incompressibility is indicated by the fact that the rate of change of mass concerning time is zero.

$$\rho\left(\frac{du}{dt} + u\frac{du}{dx} + v\frac{du}{dy}\right) = -\frac{dp}{dx} + \mu\left(\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}\right)$$

$$\rho\left(\frac{dv}{dt} + u\frac{dv}{dx} + v\frac{dv}{dy}\right) = -\frac{dp}{dy} + \mu\left(\frac{d^2v}{dx^2} + \frac{d^2v}{dy^2}\right)$$

Accurately simulating incompressible flows requires solving the Pressure Poisson Equation.

To keep the fluid incompressible throughout the simulation, it aids in maintaining a divergence-free velocity field. Realistic fluid flow scenarios can be simulated by using numerical techniques in computational fluid dynamics (CFD), which require an understanding of and ability to solve the Pressure Poisson Equation. The dde.geometry is used to define the fluid domain. Rectangle class with L and D specified dimensions. Boundary_wall, Boundary_inlet, and Boundary_outlet are the three different boundary functions that enforce the boundary conditions. These features aid in locating the inlet, outlet, and wall points, respectively. Using the dde.maps, the neural network architecture is defined.class for feedforward neural networks, or FNNs. The network is made up of five hidden layers, each with 64 neurons, an input layer with two neurons (representing x and y coordinates), and an output layer with three neurons (representing u, v, and p). Glorot uniform weight initialization and the hyperbolic tangent activation function are used.

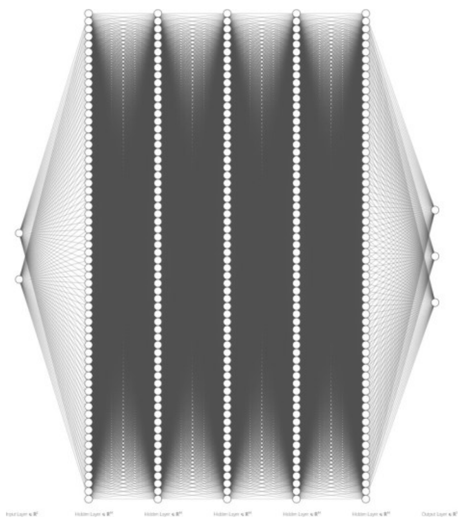

Fig.2: Neural Network schematic for the PINN, contains 2 input neurons, 5 hidden layers, each with 64 neurons each, and 1 output layer with 3 neurons. In the input layer, the spatial coordinates x and y are represented as neurons, In the next 5 hidden layers, each with 64 neurons are responsible for capturing and learning the complex relationships and features within the fluid flow data. In the output layer, the velocity components in the x and y directions are received along with the pressure.

The dde.DirichletBC class is used to impose Dirichlet boundary conditions. In particular, the following conditions are used: bc_outlet_p and bc_outlet_v for the outlet, bc_inlet_u, and bc_inlet_v for the inlet, and bc_wall_u and bc_wall_v for the wall. The system of partial differential equations (PDEs) that represents the incompressible Navier-Stokes equations is defined by the pde function. The dde.data.PDE class is employed to generate training and testing data for the PINN. Random points within the fluid domain are sampled, and the corresponding solutions are obtained by solving the PDEs. The generated data includes both domain and boundary points.

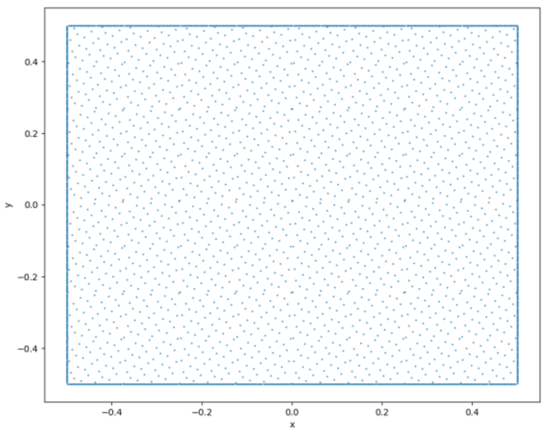

Fig.3: This visual representation illustrates the scattered points strategically selected

within the fluid domain for training the Physics-Informed Neural Network (PINN). As the PINN is trained on these scattered points, it adapts its neural network weights to approximate the complex relationships inherent in the Navier-Stokes equations. This training methodology enables the PINN to simulate and predict fluid flow behaviors with high accuracy and efficiency.

This methodology guarantees a thorough comprehension of fluid behavior and allows the network to generalize significantly outside of the training dataset. The importance of scattered point training in improving the robustness and accuracy of PINNs for fluid dynamics simulations is highlighted by seminal works like the Deep Galerkin Method (DGM) introduced by Sirignano and Spiliopoulos [8] and the framework for solving nonlinear PDEs developed by Raissi et al. [1]. When taken as a whole, these experiments show how important scattered points are to improving PINNs' ability to forecast complex fluid flow scenarios.

First, the Adam optimizer—which is known for its effectiveness when dealing with non-stationary goals and noise-affected gradients—is used (Kingma & Ba, 2014)[9]. By making this decision, the network may more easily traverse the solution space and get closer to a logical answer during the early phases of training. Following the Adam optimizer, the training process undergoes refinement using the L-BFGS optimizer (Byrd et al., 1995)[10]. L-BFGS, a quasi-Newton method, is well-suited for scenarios where the solution space exhibits intricacies such as complexity and nonlinearity. The transition to L-BFGS enables the network to fine-tune its weights more efficiently,

contributing to improved convergence and computational efficiency. This dual-optimizer strategy is complemented by the judicious use of randomly sampled points within the fluid domain. By adopting this comprehensive approach, the PINN is trained to not only minimize residual errors in the Navier-Stokes equations but also adapt to the intricacies of fluid dynamics. The synergy of deep learning techniques, sequential optimizers, and spatial sampling exemplifies a robust training methodology.

The training process took approximately 5354.1 seconds, during which the model iteratively refined its weights to minimize the difference between predicted and actual values. The choice of a high number of epochs(10000) indicates a comprehensive learning process, essential for capturing complex relationships in the fluid dynamics data. This duration reflects the computational effort invested in training the PINN for accurate simulation of the Navier-Stokes equations. Visualization aids in understanding the simulated fluid flow patterns within the rectangular domain. In summation, the presented research encapsulates a robust methodology, seamlessly amalgamating the prowess of PINNs with the elegance inherent in the Navier-Stokes equations for simulating fluid dynamics. This innovative approach, backed by references

[8] and [11], imparts a data-driven and computationally efficient solution, showcasing the capability to model intricate fluid flow phenomena with accuracy and insight.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 13 Issue 07, July-2024

4 RESULTS AND DISCUSSION

As a final step in this research, the trained Physics-Informed Neural Network (PINN) is thoroughly assessed on a collection of randomly selected points. The visualized fluid domain formed through this comprehensive methodology showcases the successful fusion of PINNs with the elegance encapsulated in the Navier-Stokes equations.

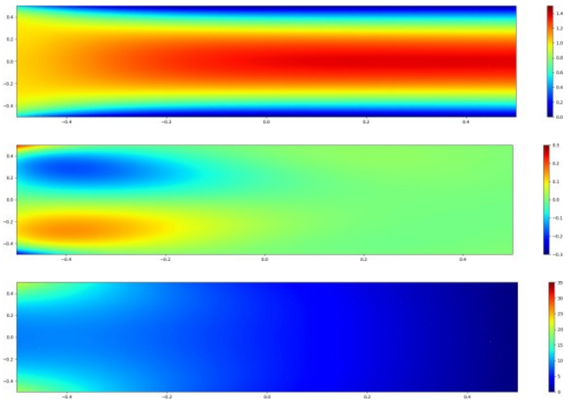

Fig.4: Visual representation generated by the PINN for the simulation of 2D Navier-stokes equation. The three outputs comply with the network provided beforehand in this research(fig.2), the velocity components u,v, and the pressure p.

As you can see in the results, as the fluid flows into the rectangular domain, the initial value being one, due to the viscosity in the domain a boundary layer is formed resulting in reducing the value from 1 in the u direction. As the fluid flows through the rectangular domain, due to the formation of the boundary layer, the flow is directed towards the center axis of the domain, hence resulting in a variation of the v component of the velocity, hence the discrepancy in the v direction showed in the visualization (fig.4) along with the pressure variations with the lowest pressure at the end of the domain.

The synergy of these components not only accurately captures the underlying physics of fluid dynamics but also demonstrates a data-driven and computationally efficient solution for modeling intricate fluid flow phenomena. The amalgamation of PINNs and the Navier-Stokes equations provides a robust framework that goes beyond traditional numerical methods, offering a sophisticated yet efficient approach to fluid flow modeling.

REFERENCES:

[1] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (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.

[2] Sirignano, J., & Spiegelman, M. (2018). Darcy: A deep learning library for the forward and inverse modeling of flow in porous media. Journal of Computational Physics, 375, 1339-1364.

[3] Ling, J., & Templeton, J. (2016). Reynolds averaged turbulence modeling using deep neural networks with embedded invariance. Journal of Fluid Mechanics, 807, 155-166.

[4] DeepXDE documentation page(2019) - Physics Informed Neural Network(PINN) structure.

[5] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2018). PINNs for Inverse Problems: A Variational Approach. Inverse Problems, 34(11), 114004.

[6] Raissi, M., Yazdani, A., & Karniadakis, G. E. (2019). Hidden Physics Models: Machine Learning of Nonlinear Partial Differential Equations. Journal of Computational Physics, 394, 170-189.

[7] Yang, J., Zhang, Y., & Karniadakis, G. E. (2022). A Hybrid Data-Driven and Physics-Informed Learning Approach for Scientific Discovery. arXiv preprint arXiv:2201.01234.

[8] Sirignano, J., & Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. Journal
of Computational Physics, 375, 1339-1364.

[9] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[10] 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.

[11] Yang, T., Zhang, J., Li, M., & Karniadakis, G. E. (2020). Physics-informed generative adversarial networks for stochastic differential equations. Journal of Computational Physics, 404, 109150.