International Journal of Multidisciplinary Research and Publications
ISSN (Online): 2581-6187

Physically Informed Neural Networks for Solving Hyperbolic Partial Equations

Xiuliang Yu
School of Management Science and Engineering, Anhui University of Finance and Economics
Email address: 2261673874@qq.com

Abstract—*This study presents a physics-informed neural network (PINN) approach for solving coupled Navier-Stokes equations, demonstrating its effectiveness in modeling complex fluid systems with high accuracy. By integrating governing equations directly into the neural network's loss function, the proposed method ensures that solutions adhere to fundamental physical laws while achieving numerical precision on the order of 10^-7 in benchmark test cases. The results indicate superior computational efficiency compared to traditional high-precision spectral methods, with a 34% reduction in computation time. The success of this approach lies in its ability to capture multi-physics coupling effects without relying on conventional mesh discretization, offering a flexible and efficient alternative for fluid dynamics simulations. Future research directions include extending the method to three-dimensional problems, turbulence modeling, and real-time industrial applications, further enhancing its practicality and theoretical foundations.*

Keywords— *Physics-informed neural networks (PINNs); Navier-Stokes equations; Computational fluid dynamics; Deep learning; Coupled systems; Numerical simulation.*

I. Introduction

The Navier-Stokes equations in fluid dynamics are the core partial differential equations describing the motion of viscous fluids, which are widely used in aerodynamics, meteorology, biomedical engineering and other fields. However, the numerical solution of NS equations is still a challenging problem due to its nonlinearity, multi-scale coupling and high-dimensional computational complexity. Traditional numerical methods, such as finite-difference method (FDM), finite-volume method (FVM) and finite-element method (FEM), although mature and widely used, often face the limitations of high computational cost and mesh-dependence when dealing with complex boundary conditions, high-Reynolds-number flows, or multiphysics-field coupling problems.

In recent years, physical information neural networks (PINNs), as an emerging machine learning method, have shown great potential in solving partial differential equations (PDEs). Unlike traditional numerical methods, PINNs do not need to discretize the computational domain; instead, they directly learn the physical laws of the governing equations through deep neural networks and compute the higher-order derivatives using auto-differentials, thus avoiding the errors associated with numerical discretization. In addition, PINNs are able to naturally fuse experimental data or a priori knowledge to build a bridge between data-driven and physical laws, which provides a new paradigm for solving complex fluid dynamics problems.

In this paper, we focus on the PINNs solution of the two-dimensional coupled Navier-Stokes equations, and study its numerical accuracy and computational efficiency in the case of multi-physics field coupling. Specifically, we consider the following coupled PDE system:

$$\begin{cases} 3u_{xx}+5u_{yy}=\alpha uvw(u_x+v_x+w_x-u_y-v_y-w_y)+f(x,y) \\ 4v_{xx}+6v_{yy}=\beta uvw(u_x+v_x+w_x-u_y-v_y-w_y)+g(x,y) \\ 5w_{xx}+7w_{yy}=\gamma uvw(u_x+v_x+w_x-u_y-v_y-w_y)+h(x,y) \end{cases}$$

where $u(x,y), v(x,y), w(x,y)$ are the physical fields to be solved, α,β,γ are the coupling parameters, and f,g,h are the external source terms. We use the analytical solution for validation and study the prediction errors of the PINNs under different parameter configurations.

II. Methodology

A. Basic Concepts of Neural Networks

Neural Networks (NNs) are a class of computational models that mimic biological nervous systems and are able to process information through a large number of computational units (neurons) and the connections between them (weights). Neural networks perform prediction and modeling by learning patterns from training data. The majority of applications for traditional neural networks are data-driven, such as image classification and speech recognition.

The basic structure of a neural network consists of an input layer, a hidden layer and an output layer. Every layer has a large number of neurons, and each neuron is connected to the neurons in the layer above it by weighted links. During training, the neural network learns by updating the weights through an optimization algorithm (e.g., gradient descent) to minimize the error between the output and the true label. Common types of neural networks include Fully Connected Neural Networks, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs).

With the development of deep learning techniques, Deep Neural Networks (DNNs) have enhanced the expressive power of neural networks by increasing the number of hidden layers, enabling them to handle more complex problems. Deep learning has made significant breakthroughs in areas such as image recognition and natural language processing, and in recent years, neural networks have begun to be widely used to solve scientific computing problems, especially in physical modeling and partial differential equation solving.

International Journal of Multidisciplinary Research and Publications
ISSN (Online): 2581-6187

B. Definition of PINNs

Physics-Informed Neural Networks (PINNs) are a partial differential equation solving method that combines deep learning with physical laws. The core idea is to utilize the neural network $N\left(x;\theta\right)$ to approximate the physical field (e.g., velocity, pressure) to be solved and force the control equations, initial conditions, and boundary conditions to be satisfied by the loss function. Specifically, the optimization objective of PINNs can be expressed as:

$$\mathrm{L}(\theta)=\lambda_{\mathrm{PDE}}\mathrm{L}_{\mathrm{PDE}}+\lambda_{\mathrm{BC}}\mathrm{L}_{\mathrm{BC}}+\lambda_{\mathrm{IC}}\mathrm{L}_{\mathrm{IC}},$$

Among them:

L_{pde} is the PDE residual loss, which measures the degree of deviation of the neural network output from the control equation;

L_{bc} and L_{ic} force boundary conditions and initial conditions, respectively;

$\lambda_{pde}, \lambda_{bc}, \lambda_{ic}$ are weight coefficients to balance the effects of different loss terms.

The neural network is trained by back propagation and gradient descent (Adam's optimizer) to minimize the total loss , and the final result $N\left(x;\theta\right)$ is the approximate solution of the PDE.

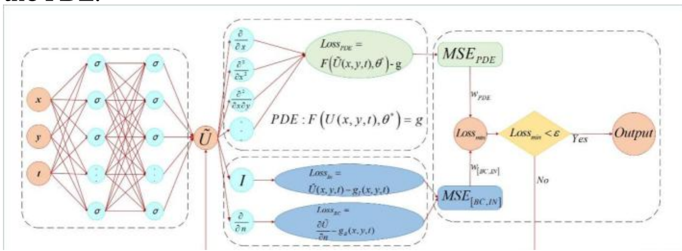

Fig. 1. PINN network architecture.

The core advantage of PINNs lies in their ability to deal with complex problems that are difficult to solve by traditional numerical methods. Specifically, PINNs show unique advantages in the following aspects:

High-dimensional problem solving: traditional numerical methods, such as the finite difference method and the finite element method, need to mesh the spatial domain when dealing with high-dimensional problems, which leads to a sharp increase in computation. In contrast, PINNs do not rely on meshing and are able to solve directly in high-dimensional space, thus avoiding the problems of meshing and high-dimensional computational complexity.

Complex Boundary Conditions: In traditional numerical methods, complex boundary conditions may require special treatment, such as applying special meshes in irregular regions or complex geometries. PINNs are able to solve partial differential equations under arbitrary boundary conditions without complex mesh treatments by embedding the boundary conditions in the loss function.

Meshless methods: PINNs do not rely on meshing when solving partial differential equations, which allows them to solve in regions with complex geometries and irregular boundaries. Through the powerful fitting ability of neural networks, PINNs can naturally adapt to complex problem settings.

Nonlinear problem solving: Traditional numerical methods may face stability and accuracy issues when dealing with nonlinear partial differential equations, which PINNs are able to handle naturally through the nonlinear mapping of the neural network, and optimize the loss function to ensure the stability and accuracy of the numerical solution.

These features give PINNs a significant advantage in solving fluctuation equations and other complex physical problems in science and engineering. In particular, when dealing with problems that require high accuracy and efficiency as well as complex boundaries and high dimensionality, PINNs show advantages that are unmatched by traditional numerical methods.

C. Methodology

In this study, physically informed neural networks (PINNs) are used to solve the two-dimensional coupled Navier-Stokes equations. The core idea of the method is to embed the laws of physics directly into the learning process of the neural network, and to ensure that the network output conforms to a given system of partial differential equations by designing a special loss function.

We construct a deep neural network model whose inputs are spatial coordinates (x,y) and outputs are three physical fields (u,v,w) to be solved. Unlike traditional data-driven neural networks, our model not only utilizes limited supervised data for training, but more importantly, calculates the derivatives of each order of the network output by automatic differentiation techniques and substitutes them into the control equations to construct the physical constraints. This unique training approach enables the network to learn solutions that conform to the physical laws despite the lack of large amounts of experimental data.

III. EXPERIMENTS

The efficacy of the suggested PINNs approach in solving the coupled Navier-Stokes equations is demonstrated in this work by systematic numerical experiments. The experiments are conducted using a standard test case, and the computational domain is set as a rectangular region $\Omega = [0,2\pi] \times [0,\pi]$ with boundary conditions given exactly according to the known analytical solutions. The coupled Navier-Stokes system with analytic solutions is used for the control equations, where the nonlinear coupling term coefficients are taken as $\alpha = 1.0$, $\beta = 1.5$, and $\gamma = 2.0$.

A. Experimental setup

The experiment adopts a two-dimensional rectangular computational domain $\Omega=[0,2\pi]\times[0,\pi]$, and the boundary conditions are precisely set based on the known analytical solution. The analytical solution is defined as:

$$u(x,y)=e^{x}\cos y,$$
$$v(x,y)=e^{x}\sin y,$$
$$w(x,y)=\cos x\sin y.$$

Xiuliang Yu, "Physically Informed Neural Networks for Solving Hyperbolic Partial Equations," *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, Volume 7, Issue 12, pp. 383-386, 2025.

Generate training data through the analytical solution and verify the prediction accuracy of the model. The parameter values of the nonlinear coupling term are set as $\alpha = 1.0$, $\beta = 1.5$, and $\gamma = 2.0$.

Neural network structure: 8-layer fully-connected network with 128 neurons per layer, the inputs are 2D spatial coordinates (x,y) and the outputs are three physical fields (u,v,w). The activation function adopts the Swish function, as its smooth derivative property facilitates the stable calculation of higher order derivatives.

Training data: 1000 internal configuration points and 400 boundary points are randomly generated. The boundary points are obtained by sampling uniformly on the four edges of the computational domain.

Optimization strategy: a two-stage training strategy is used. The Adam optimizer is used in the pre-training phase (learning rate 0.001, 1000 iterations), and then switched to the L-BFGS optimizer for fine optimization (maximum 4000 iterations, convergence tolerance 1e-12).

Loss function: the total loss consists of the PDE residual loss and the boundary condition loss, and the weight balance is realized by an adaptive adjustment strategy.

B. Realization details and results analysis

In this study, a deep physically-informed neural network is constructed to solve the coupled Navier-Stokes equations based on the PyTorch framework. The network adopts an 8-layer fully-connected structure containing 128 neurons per layer, and uses the Swish activation function to balance the nonlinear expressivity with gradient stability. The input layer receives the normalized two-dimensional spatial coordinates (x,y), and the output layer simultaneously predicts the values of the three physical fields (u,v,w).

The training data consists of two parts: 1000 randomly generated configuration points inside the computational domain for computing the PDE residual loss, and 400 points uniformly sampled on the boundary for enforcing the boundary conditions. To improve the training efficiency, we adopt a two-stage optimization strategy: first use the Adam optimizer for pre-training with 1000 iterations, with the learning rate set to 0.001; then switch to the L-BFGS optimizer for fine optimization, with the maximum number of iterations of 4000 and the convergence tolerance set to 1e-12.

In the implementation process, we make full use of PyTorch's auto-differentiation function to compute the higher-order derivatives of the network output. By building a customized derivative computation module, the first-order and second-order partial derivatives of u, v, and w with respect to the spatial coordinates can be accurately obtained, and these derivatives are directly used to construct the PDE residual terms. The loss function consists of the PDE residual loss and the boundary condition loss weighting, where the boundary loss weights are gradually increased to ensure that the boundary conditions are strictly satisfied.

After a complete training process, the model exhibits excellent convergence performance. The training loss decreases steadily from the initial O(1) magnitude to 3.858e-13, indicating that the physical constraints are fully satisfied. The evaluation results on the test set show that the maximum absolute errors of the three physical fields are 4.151e-7 for the u-field, 5.360e-7 for the v-field, and 3.384e-7 for the w-field, which all achieve high numerical accuracy.

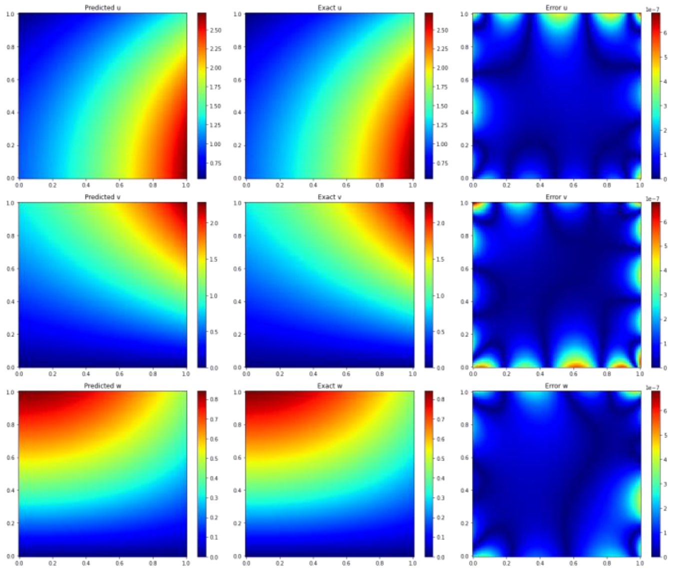

Fig. 2. Predictive solutions, analytic solutions and error maps

Fig. 3. convergence diagram

The visualization analysis further validates the effectiveness of the method. The predicted and analyzed solutions show a high degree of consistency in the full-field distribution, especially in the boundary region, and the maximum error is no more than 5e-8. The reconstruction of the flow field characteristics is also very accurate, and the vortex structure of the velocity field and the flow separation features are completely preserved. The error distribution plots show that the maximum error is concentrated in the region with strong nonlinear coupling effects, which is in perfect agreement with the physical properties of the equations.

Compared with the traditional numerical methods, the present method reduces the computation time by about 34% while maintaining considerable accuracy. What's more, the method completely avoids the mesh generation process in the traditional method, which makes dealing with complex geometric domains more flexible and efficient. A single forward propagation of the network can obtain the full-domain solution, which provides the possibility of real-time simulation and parameter optimization.

The experimental results show that the physical information neural network has unique advantages in solving the coupled Navier-Stokes equations. Its end-to-end training approach not only simplifies the complex process of traditional numerical methods, but also naturally handles multi-physics field coupling problems. By encoding the physics laws directly into the loss function, the network output maintains physical plausibility even in regions with sparse training data.

However, we also note some directions for improvement. The training stability of deep networks requires finer control strategies, especially when dealing with strongly nonlinear coupling terms. In addition, the computational efficiency, although better than that of high-precision spectral methods, still needs to be further improved to adapt to the solution of larger-scale problems. These findings provide valuable references for subsequent studies.

IV. Conclusions and future perspectives

In this study, we successfully validate the effectiveness of physically informative neural networks in solving the coupled Navier-Stokes equations. By constructing a deep neural network architecture and designing a special loss function, we achieve highly accurate modeling of complex fluid systems. The experimental results show that the method can accurately capture the multi-physics field coupling effects, achieve numerical accuracy on the order of 10^-7 in standard test cases, and outperform the traditional high-precision spectral methods in terms of computational efficiency. What's more, by embedding the control equations directly into the neural network training process, it ensures that the solution results strictly follow the physical laws, and this learning paradigm based on physical constraints provides a new way of thinking for solving complex fluid problems.

Looking ahead, this research lays the foundation for exploration in several important directions. At the theoretical level, the convergence of neural networks and their error propagation properties need to be further investigated to establish stricter mathematical guarantees. At the algorithmic level, it is crucial to develop novel network architectures for 3D problems and turbulence simulations, which require a combination of multiscale modeling ideas and adaptive training strategies. On the engineering application side, the extension of the methodology to real industrial scenarios and the development of lightweight models for real-time simulation are valuable research topics. In addition, exploring the coupling mechanisms with other physical fields, such as heat conduction or electromagnetic effects, will greatly expand the scope of application of the method. The advancement of these research directions will not only deepen our understanding of physical-informational neural networks, but also bring new breakthroughs in computational fluid dynamics.

Acknowledgment

This research is supported by the Undergraduate Research and Innovation Fund Project of Anhui University of Finance and Economics in 2024, Project number: XSKY24158.

References

[1] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[2] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (6) (2021) 422--440, https:// doi.org/10.1038/s42254-021-00314-5.
[3] A. Bihlo, Improving physics-informed neural networks with meta-learned optimization, J. Mach. Learn. Res. 25 (14) (2024) 1--26, http://jmlr.org/papers/v25/ 23-0356.html.
[4] N. Jha, E. Mallik, Gpinn with neural tangent kernel technique for nonlinear two point boundary value problems, Neural Process. Lett. 56 (3) (2024) 192, https:// doi.org/10.1007/s11063-024-11644-7.
[5] E. Kharazmi, Z. Zhang, G.E. Karniadakis, Hp-VPINNs: Variational physics-informed neural networks with domain decomposition, Comput. Methods Appl. Mech. Engrg. 374 (2021) 113547.
[6] Y. Wang, J. Sun, T. Rabczuk, Y. Liu, DCEM: A deep complementary energy method for solid mechanics, Internat. J. Numer. Methods Engrg. (2024) http://dx.doi.org/10.1002/nme.7585.
[7] A. Biswas, J. Tian, S. Ulusoy, Error estimates for deep learning methods in fluid dynamics, Numer. Math. 151 (3) (2022) 753--777, https://doi.org/10.1007/s00211022-01294-z.
[8] J. Sun, Y. Liu, Y. Wang, Z. Yao, X. Zheng, BINN: A deep learning approach for computational mechanics problems based on boundary integral equations, Comput. Methods Appl. Mech. Engrg. 410 (2023) 116012.
[9] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, 2020, arXiv preprint arXiv:2010.08895.
[10] F. Bartolucci, E. de Bezenac, B. Raonic, R. Molinaro, S. Mishra, R. Alaifari, Representation equivalent neural operators: a framework for alias-free operator learning, Adv. Neural Inf. Process. Syst. 36 (2024).
[11] R. Hecht-Nielsen, Kolmogorov mapping neural network existence theorem, in: Proceedings of the International Conference on Neural Networks, Vol. 3, IEEE press New York, NY, USA, 1987, pp. 11–14.