The 11th Asian-Australasian Conference on Precision Agriculture (ACPA 11)
October 14-16, 2025, Chiayi, Taiwan

A Physics-Informed Neural Network Approach for Simulating Laminar Flow

Ting-Shiuan Gung [1], Fu-An Wang [1], Chen-Kang Huang[1,3*]

[1] Department of Biomechatronics Engineering, National Taiwan University, Taiwan.
[3] Center for Intelligent Agriculture Education and Research, National Taiwan University, Taiwan

*Corresponding Author: ckhuang94530@ntu.edu.tw

ABSTRACT

Efficient and accurate modeling in agricultural fields is critical for advancing precision agriculture. These simulations, often involving the prediction of airflow, temperature, and humidity distributions, directly support decisions related to crop management, greenhouse climate control, and irrigation strategies. Computational Fluid Dynamics (CFD) has been a primary tool for decades, offering reliable and high-fidelity simulations through established numerical methods such as the finite-difference or finite-volume approach. While CFD tools have long been the standard in such fields, they are often computationally intensive and may be unsuitable for real-time applications or scenarios with limited computational resources.
On the other hand, Physics-Informed Neural Networks (PINNs) have emerged in the year of 2017 by Raissi et al. (2017) as a promising alternative to traditional CFD solvers. By incorporating governing physical equations directly into the loss functions of deep neural networks, PINNs offer a mesh-free modeling approach that combines the strengths of machine learning with physical consistency. Moreover, the use of modern GPUs has the potential to accelerate the training and inference processes of PINNs, making them attractive for certain simulation tasks.
In this study, we evaluate the performance of PINN-based solvers (with GPU acceleration) against conventional CFD methods in simplified flow problems. Our results show that, although PINNs provide flexibility in solver construction and avoid meshing in the preprocessing stage, CFD remains significantly faster and more stable for two-dimensional flow simulations. Furthermore, the performance of PINNs is highly sensitive to hyperparameter choices and physical parameters, which can lead to unstable or inaccurate predictions. These findings suggest that while PINNs have potential for future applications in precision agriculture, further development is required before they can outperform conventional CFD in practical scenarios.

Keywords: precision agriculture, CFD, PINN, mesh-free modeling, machine learning

INTRODUCTION

With the rapid improvement of GPU performance, neural networks can be significantly accelerated, making physics-informed neural networks (PINNs) a competitive alternative to the currently dominant method, computational fluid dynamics (CFD).

The physics-informed neural network (PINN) is a branch of the machine learning field. Unlike traditional neural networks proposed by Ren et al. (2020) , which compute the loss function by comparing the network outputs with actual data, PINNs compute their loss by substituting the outputs into the governing equations and boundary conditions. As a result, no experimental or simulation data are required to predict the desired physical properties.

The governing equations for fluid dynamics are the Navier–Stokes equations. In this study, we adopt the mixed-variable PINN architecture for fluid dynamic problems proposed by Rao et al. (2020), as illustrated in Fig. 1. This network generates the outputs ψ, p and σ, representing the stream function, pressure, and shear stress tensor, respectively. For two-dimensional problems, the velocity components can be directly computed from the stream function ψ, ensuring that the divergence-free condition of the flow is automatically satisfied. Moreover, directly learning the shear stress matrix avoids the need for second-order derivatives during backpropagation, thereby reducing fluctuations and allowing the network to converge more effectively. The Navier–Stokes equations used in this architecture are given by:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = \frac{1}{\rho}\nabla \cdot \sigma + \mathbf{g} \quad (1)$$

where $\mathbf{v}$ = velocity vector (m/s), ρ = fluid density (kg/m^3), σ = stress tensor (Pa), and $\mathbf{g}$ = body force per unit mass (m/s^2)

$$\sigma = -p\mathbf{I} + \mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T) \quad (2)$$

where p = pressure (Pa), $\mathbf{I}$ = identity matrix, and μ = dynamic viscosity (Pa·s)

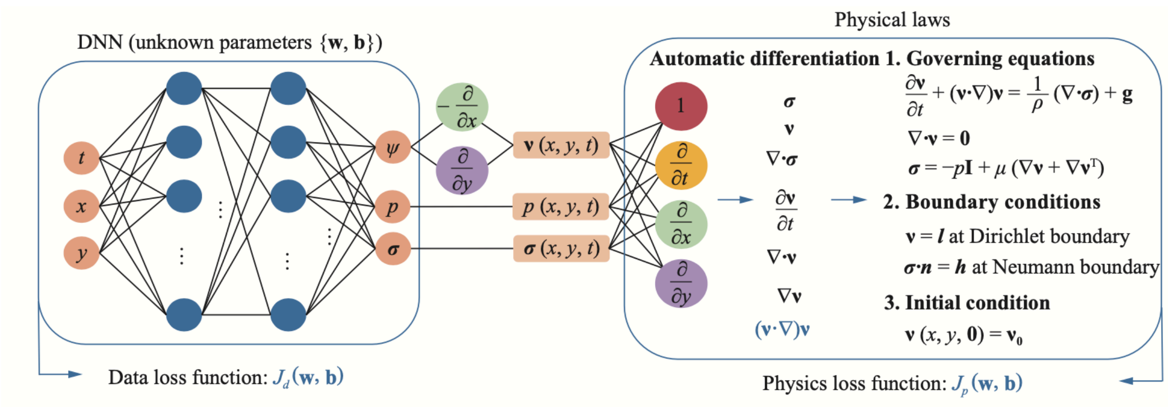
The loss function is composed of the physics loss $\mathbf{L}_g$ and the boundary condition loss $\mathbf{L}_{bc}$, defined as

$$\mathbf{L} = \mathbf{L}_g + \beta \mathbf{L}_{bc} \quad (3)$$

where $\beta(>0)$ a user-defined weighting coefficient

And the individual losses are computed as the mean square value of the residuals. In this paper, we discuss the advantages and disadvantages of PINN and CFD, compare the speed of PINN with and without GPU acceleration, examine valid parameter settings for the PINN structure, and explain the details of the PINN modeling framework.

Fig. 1. The mixed-variable architecture of physics informed neural network for fluid dynamics from Rao et al. (2020).



MATERIAL AND METHODS

CFD simulations were performed using the commercial software SolidWorks 2022, which employs the finite-volume method to discretize the Navier–Stokes equations. A two-dimensional rectangle containing a circular cylinder was constructed to mimic velocity and pressure distributions when a fluid encounters an obstacle. A parabolic inlet velocity profile was applied at the inlet using the dependency chart, while a gauge pressure condition was imposed at the outlet. No-slip boundary conditions were applied to all solid surfaces, including the walls and the cylinder. The physical properties were set as: density = 1 kg/m³, dynamic viscosity = 0.02 Pa·s, specific heat = 1 J/(kg·K), and thermal conductivity = 1 W/(m·K).

The PINN model was implemented in Python, with the core neural network architecture built using the PyTorch framework. Required packages included torch, numpy, and lhs from pyDOE, among others. Approximately 50,000 collocation points were generated for training, including 1,585 Dirichlet boundary points (cylinder, wall, inlet), 201 Neumann boundary points (outlet), and ~48,000 interior points randomly distributed using Latin hypercube sampling (LHS). Similar to the CFD setup, a parabolic velocity profile was applied at the inlet, while a gauge pressure condition was imposed at the outlet. However, unlike CFD, the PINN solution only provides the pressure scale rather than absolute values.

Training was performed using the L-BFGS optimizer, with the code simplified from Rao et al. (2020) and implemented with Python's built-in nn.Module. The weighting coefficient β was set to 5, which produced the best training results. For hardware comparison, an NVIDIA RTX 4070 GPU with CUDA cores was used to accelerate the PINN model (treatment group), while an Apple M1 CPU was used for the control group.

RESULTS AND DISCUSSION

The traditional PINN structure generates outputs u, v and p, representing the x-direction velocity, y-direction velocity, and pressure, respectively. However, this scheme often fails to reproduce accurate velocity and pressure fields. To address this, the mixed-variable architecture was employed. Fig.2. shows the velocity and pressure fields obtained using CFD, the mixed-variable PINN with a network size of 8×60, and the traditional PINN with the same size.

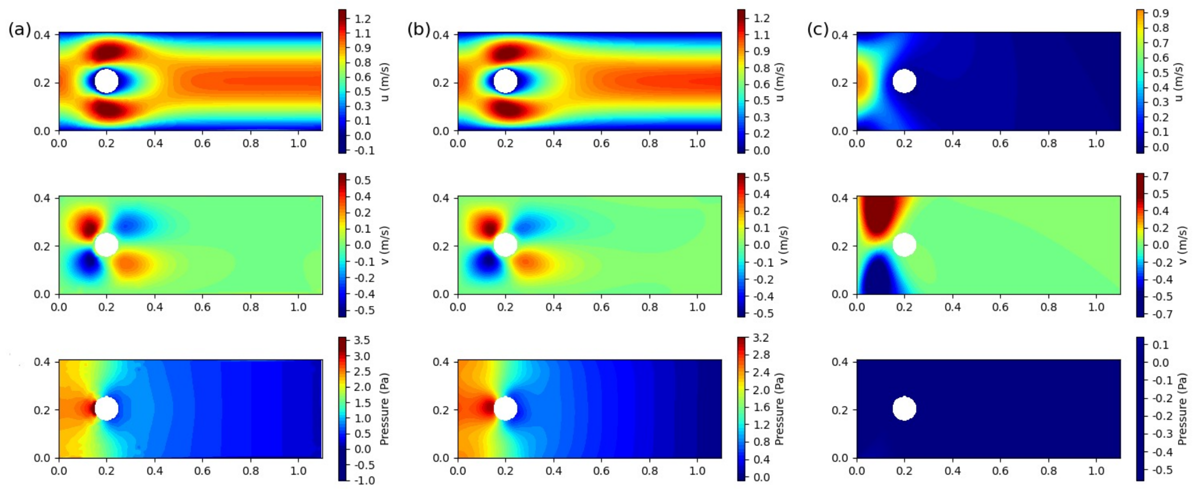


Fig. 2. Velocities and pressure felids of the steady flow passing through a circular cylinder. (a) Reference solution generated from CFD with level 5 mesh. (b) Mixed-variable scheme solution with 8×60 network size. (c) Traditional scheme with 8×60 network size.

Table 1 compares the final total loss, error in the u-field, and average training time per 100 epochs with and without GPU acceleration. The final total loss was taken from the last epoch, while the error was calculated as the mean absolute error of the x-direction velocity:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | U_{PINN,i} - U_{CFD,i} | \quad (4)$$

The results indicate that a network with 60 neurons and 8 layers achieved the smallest loss, with an error below 5% (Table 1). With GPU acceleration, training time was reduced by approximately 25×. Unlike most neural networks, which commonly use the Adam optimizer, this study employed L-BFGS. The L-BFGS optimizer dynamically adjusts its learning rate and stops either when the loss converges or when the user-defined maximum number of epochs is reached. However, the convergence rate strongly depends on the initialization of the network weights and biases. Therefore, the total training time and the number of epochs cannot be precisely determined; instead, the time per 100 epochs is reported as the average over 10,000 epochs.In contrast, CFD required less than 5 seconds to solve this two-dimensional problem, which is significantly faster than PINN.

Table 1. Comparison of final total loss, error of the u-field, and average training time per 100 epochs with GPU and CPU acceleration for different network sizes.

Depth×width	Final total loss(10^{-3})	Error	GPU time(s)/100 epoch	CPU time/ 100 epoch
8×40	42.50	0.1200	6.00	148.05
8×50	4.56	0.0800	7.23	186.35
8×60	3.87	0.0328	8.48	213.99

CONCLUSION

Even with GPU acceleration, PINNs cannot yet outperform CFD for this class of problems. Furthermore, the PINN structure is highly sensitive to hyperparameters: small changes in coefficients such as β or even in physical parameters like viscosity μ and density ρ can lead to inaccurate results.Nevertheless, the mesh-free nature and flexibility of PINNs suggest that, with further methodological improvements, they may become a viable complement to CFD in future agricultural modeling applications

ACKNOWLEDGEMENTS

This research was supported by the Ministry of Agriculture under grant 114AS-1.6.1-AS-28

REFERENCE

Raissi, M., P. Perdikaris, and G. E. Karniadakis. 2017. Physics-informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations.

Rao, C., H. Sun, and Y. Liu. 2020. Physics-informed deep learning for incompressible laminar flows.

Ren, P., C. Rao, Y. Liu, J. Wang, and H. Sun. 2020. PhyCRNet: Physics-informed convolutional-recurrent network for solving spatiotemporal PDEs.