УДК 004.89, 532.5

O. Avramenko, S. Kompan, M. Sarana

DOI: 10.18523/2617-70808202562-68

PINN-BASED MACHINE LEARNING FOR MODELING INTERNAL WAVES IN SEMI-INFINITE FLUIDS

This study investigates the application of Physics-Informed Neural Networks (PINNs) for modeling wave processes at the interface between two incompressible fluids of differing densities. As a first step, the linear formulation of the problem is considered, which admits an analytical solution based on a spectral method involving Fourier decomposition of the initial perturbation. This solution serves as a benchmark for testing and validating the accuracy of the PINN predictions.

The implementation is carried out in Python using specialized libraries such as TensorFlow, NumPy, SciPy, and Matplotlib, which provide both efficient deep learning frameworks and tools for solving mathematical physics problems numerically. The approach integrates artificial intelligence with domain-specific knowledge in hydrodynamics, enabling the construction of interpretable and physically consistent models. Particular attention is given to the organization of the computational experiment, automation of visualizations, and storage of intermediate results for further analysis. The PINN model includes a loss function that encodes the governing equations and boundary conditions, and the training is conducted on randomly sampled points across the spatio-temporal domain. The influence of network architecture and training parameters on solution accuracy is examined. Visualization of loss function evolution and predicted wave profiles provides insight into convergence behavior and physical plausibility of the solutions.

A comparative analysis between the PINN-based and analytical solutions across different time instances is presented, revealing phase shifts and amplitude deviations. The model demonstrates good agreement at early times and a gradual accumulation of errors as time progresses—an expected feature of this class of methods. The results confirm the feasibility of applying the PINN framework to linear hydrodynamic problems, laying the groundwork for future extensions to weakly and strongly nonlinear regimes, including studies of wave stability and nonlinear wave dynamics.

Keywords: physically informed neural network (PINN), loss function, neural network testing, wave profiles.

Introduction

Physics-informed neural networks (PINNs) represent a modern class of machine learning models specifically designed to solve problems while respecting physical laws formulated as nonlinear differential equations. In a seminal work, Raissi et al. introduced the foundational PINN architecture capable of addressing both forward and inverse problems, enabling effective modeling in fields such as hydrodynamics, quantum mechanics, and wave dynamics [1]. Subsequently, Lin and Chen proposed a two-stage PINN training method that incorporates additional constraints on conserved physical quantities, significantly enhancing the accuracy and generalization capability of the model in reconstructing complex localized wave solutions, including soliton molecules and their interactions [2]. Bhatnagar et al. demonstrated that PINNs can be successfully applied to three-dimensional flow and thermal processes with limited data, serving as high-fidelity surrogate models for complex engineering applications [3].

In hydromechanics, the application of PINNs gains particular importance due to the high cost of experimental measurements and numerical simulations. Sharma et al. conducted a comprehensive review highlighting the potential of physics-informed machine learning methods to improve prediction accuracy and robustness for turbulent flows, as well as to replace resource-intensive numerical models [4]. Buhendwa et al. employed PINNs to reconstruct velocity and pressure fields in two-phase flow problems based on interface motion data, effectively addressing both forward and inverse hydrodynamic problems [5].

Wave process modeling remains one of the most actively developed fields of PINN applications. Pu et al. proposed an enhanced architecture with adaptive activation functions, which enabled high-precision recovery of localized solutions to the derivative nonlinear Schrödinger equation, including various soliton and solitary wave types [6]. Wang and Yan applied a multi-level PINN framework to discover parameters and recover rogue wave solutions of the defocusing non-

linear Schrödinger equation with time-dependent potential, opening new avenues for solving inverse problems with high sensitivity to initial conditions [7]. Pan et al. demonstrated that deep neural networks could effectively reconstruct internal wave amplitudes from optical satellite images by leveraging texture features and peak-to-peak distance analysis, achieving accuracy around 3.6% compared to in-situ measurements [8]. Liu et al. introduced the hybrid LWT-PINN model, combining classical linear wave theory with PINNs to provide phase-resolved wave prediction in real time, outperforming traditional linear models in both accuracy and forecast duration [9].

In seismic modeling, PINNs address challenges related to wave propagation in semi-infinite and geologically complex domains where proper boundary condition handling and computational scalability are critical. Ren et al. developed SeismicNet, a PINN model integrating soft boundary condition enforcement and sequential temporal training, enabling efficient seismic wave simulation under various geological parameter distributions [10]. To overcome spectral bias in modeling multifrequency seismic wavefields governed by the Helmholtz equation, Song and Wang proposed the Fourier Feature PINN—a model with embedded Fourier features that demonstrates high efficiency and accuracy when processing multiple frequencies simultaneously [11].

It is therefore worthwhile to investigate the applicability of PINNs for solving weakly nonlinear problems in hydromechanics and wave processes occurring in heterogeneous fluids, which is the focus of the present study.

Problem Statement and Preliminary Results

Physics-informed neural networks (PINNs) will be employed to solve the nonlinear problem of interfacial waves between two incompressible, ideal fluids with surface tension. The main objective is to utilize this classical hydromechanical problem to test the PINN methodology, assess its effectiveness, and demonstrate its applicability to solving complex physical problems. The goal is to determine the wave profile $\eta(x,t)$ and the velocity potential fields $\phi_1(x,z,t)$ in the lower fluid half-space Ω_1 and $\phi_2(x,z,t)$ in the upper half-space Ω_2. A potential flow is considered, where ϕ_1 and ϕ_2 satisfy Laplace's equations in their respective domains.

The interfacial boundary is given by $z = \alpha\eta(x,t)$, where α is a parameter scaling nonlinear effects. On this boundary, kinematic conditions relate the surface motion to the fluid velocity, and a dynamic condition describes the pressure balance accounting for gravity, kinetic energy, and surface tension T. All nonlinear terms, including those associated with surface tension, are scaled by α. At remote boundaries along the z-direction (as $z \to \pm\infty$), fluid velocities decay to zero. The initial wave profile is prescribed at $t = 0$ as $\eta(x,0) = f(x)$.

The mathematical formulation of the problem (for $i = 1, 2$):

$$\phi_{i,xx} + \phi_{i,zz} = 0 \quad \text{in } \Omega_i, \tag{1}$$

with the kinematic boundary conditions on the interface $z = \alpha\eta(x,t)$

$$\eta_{,t} + \alpha\phi_{i,x}\eta_{,x} = \phi_{i,z}, \tag{2}$$

and the dynamic boundary condition on the interface $z = \alpha\eta(x,t)$

$$\phi_{1,t} - \rho\phi_{2,t} + (1-\rho)g\eta + \frac{\alpha}{2}\left(\phi_{1,x}^2 + \phi_{1,z}^2\right) - \frac{\alpha\rho}{2}\left(\phi_{2,x}^2 + \phi_{2,z}^2\right) - T\frac{\eta_{,xx}}{\left(1 + (\alpha\eta_{,x})^2\right)^{3/2}} = 0, \tag{3}$$

with decay conditions at infinity

$$\phi_{i,x} \to 0, \quad \phi_{i,z} \to 0 \quad \text{as } z \to \mp\infty, \tag{4}$$

and the initial condition

$$\eta(x,0) = f(x). \tag{5}$$

In the linear regime, an analytical solution exists and is detailed, for example, in [12]. Specifically, the interface deviation $\eta_{lin}(x,t)$ in the linear approximation ($\alpha \to 0$) can be represented as a sum of right- and left-propagating waves,

$$\eta_{lin}(x,t) = \eta_{lin}^{+}(x,t) + \eta_{lin}^{-}(x,t),$$

each expressed through Fourier integrals of the initial displacement $f(x)$:

$$\eta_{lin}^{\pm}(x,t) = \frac{1}{2}\int_{-\infty}^{+\infty}\Big(a_f(k)\cos(kx \mp \omega(k)t) - b_f(k)\sin(kx \mp \omega(k)t)\Big)dk, \tag{6}$$

where

$$a_f(k) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} f(\xi)\cos(k\xi)d\xi,$$

$$b_f(k) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} f(\xi)\sin(k\xi)d\xi,$$

and the dispersion relation is

$$\omega(k) = \sqrt{\frac{k(1-\rho+Tk^2)}{1+\rho}}.$$

The following sections describe the implementation of PINN modeling, with the analytical solution (6) used as a reference to verify and validate the correctness of the PINN approach.

Implementation of PINN Modeling

Model Initialization and PINN Architecture. To solve the wave dynamics problem (1)–(5), a PINN was implemented using Python. The setup relies on standard libraries: `TensorFlow` (for constructing and training neural networks), `NumPy` (for numerical operations), `Matplotlib` (for visualization), and `os` (for file system interaction). Automatic differentiation is enabled through `TensorFlow`'s eager execution mode.

The architecture includes three fully connected neural networks, each responsible for approximating one of the key physical quantities: the interfacial displacement $\eta(x,t)$ and the velocity potentials $\phi_1(x,z,t)$ and $\phi_2(x,z,t)$ in the lower and upper fluid domains, respectively. The networks use `tanh` as activation functions and consist of multiple hidden layers with configurable depth and width.

The initial profile of the interface is modeled as a Gaussian wave packet characterized by its amplitude, width, and position. Physical parameters such as fluid density ratio, surface tension, and domain size are defined according to the linear regime considered in the first stage of the experiment.

Loss Function Implementation. The loss function, which drives the training of the PINN model, is constructed to reflect how well the neural network approximations satisfy the governing physical equations, initial condition, and boundary conditions. Contributions from different regions of the domain—including the fluid bulk, interface, boundaries at infinity, and initial time—are evaluated using randomly sampled collocation points generated via `tf.random.uniform`.

Each component of the loss corresponds to a different aspect of the problem: satisfaction of Laplace's equation in each fluid layer, enforcement of the kinematic and dynamic boundary conditions at the interface, decay behavior at spatial infinity, and adherence to the initial condition. Derivatives involved in these conditions are computed using `TensorFlow`'s `GradientTape` mechanism for automatic differentiation.

To ensure symmetry in the predicted solutions—expected due to the symmetric form of the initial condition—a dedicated term is added to penalize asymmetry with respect to the spatial variable. This improves both accuracy and the physical consistency of the resulting wave profiles.

All components are combined into a weighted sum, which forms the total loss minimized during training. Monitoring of individual loss terms is used to evaluate convergence and guide potential refinements to the network design or training strategy.

Training and Loading of Network Parameters. Training of the neural networks is carried out iteratively using the `Adam` optimizer with a fixed learning rate. At the start, the system checks for previously saved model weights; if such files are found, training is skipped and the weights are loaded. Otherwise, training begins from scratch.

The process uses a fixed number of training epochs and a predefined number of collocation points per step. Automatic differentiation is performed via `TensorFlow`'s `GradientTape`, which computes gradients of the total loss with respect to the model parameters. These gradients are then used to update the weights during optimization.

Training progress is monitored in real time through periodic plotting of the predicted wave profile $\eta(x,t)$ at selected time snapshots, along with printing loss values to the console. All key metrics, including the evolution of total and individual loss components, as well as the learning rate history, are stored for later analysis.

After training, the learned weights are saved to disk, enabling reuse without retraining. To improve convergence and training stability, standard learning rate scheduling strategies such as `ExponentialDecay` or `ReduceLROnPlateau` may be applied, allowing adaptive adjustment of the learning rate based on loss evolution.

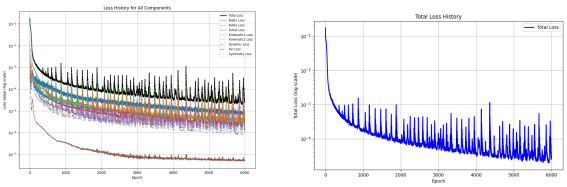
Convergence Analysis and Visualization of PINN Modeling Results. The convergence behavior of the model is analyzed through visualization of loss histories and predicted wave fields. If available, stored data on loss values from the training process is used to generate plots showing the evolution of individual loss terms and the total loss, typically on a logarithmic scale. These plots help evaluate the relative contributions of physical constraints and the model's convergence behavior.

Upon completion of training or loading of pretrained weights, predicted solutions are visualized and saved. This includes interface displacement profiles $\eta(x,t)$ at a series of fixed time points, which are plotted together and compared with the initial condition to assess accuracy and physical consistency.

A three-dimensional plot of $\eta(x,t)$ provides a global view of wave packet dynamics over space and time. Additionally, contour plots of the velocity potentials $\phi_1(x,z,t)$ and $\phi_2(x,z,t)$ at a given time are generated to examine flow structure in each fluid layer, with the interface between them clearly marked.

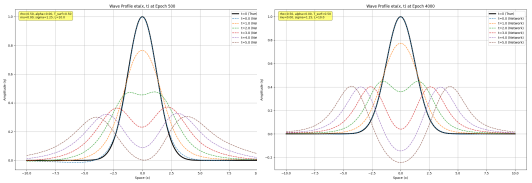
Below, several visualizations are presented to demonstrate the performance of the program, obtained using a PINN model with architecture L7_N25, consisting of 7 layers with 25 neurons each.

Figure 1 presents the dynamics of the loss function during the training process. The vertical axes of both plots are shown on a logarithmic scale, which enhances the visualization of changes in the loss values, particularly in the early stages of training. It can be observed that during the initial epochs, the total loss (Figure 1b) and its components (Figure 1a) decrease rapidly, followed by a phase of slower reduction accompanied by oscillations—a typical pattern for neural network optimization.

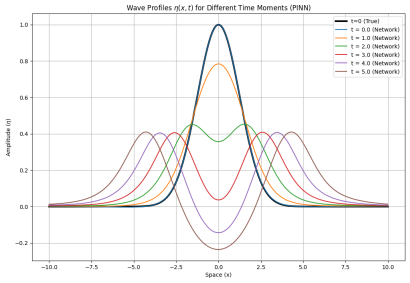


(a) (b)

Figure 1. Loss function history during training: (a) individual components; (b) total loss.



(a) (b)



(c)

Figure 2. Wave profiles $\eta(x,t)$ at $t = 0, 1, 2, 3, 4, 5$: (a) after 2000 epochs; (b) after 4000 epochs; (c) final predictions after training.

Figures 2a and 2b illustrate the evolution of wave profiles $\eta(x,t)$ at intermediate training stages (epochs 2000 and 4000). Dashed lines represent the network predictions, while the bold black line shows the true initial condition. These plots demonstrate the gradual formation and propagation of the wave packet, converging toward physically consistent behavior. Figure 2c shows the results obtained after the final stage of training for the PINN architecture L7_N25. The plot illustrates the temporal evolution of the wave profile $\eta(x,t)$ at several discrete time moments. The results demonstrate that the trained neural network is capable of capturing the propagation and dispersion of the internal wave packet, with visible agreement to the expected physical behavior across the time axis.

Figure 3 presents a three-dimensional visualization of the final trained solution for the wave profile $\eta(x,t)$ over the full spatiotemporal domain (x,t). The 3D surface plot highlights the dynamic evolution of internal waves and clearly reveals the wave packet's trajectory, amplitude variations, and dispersive spreading as learned by the PINN model after complete training.

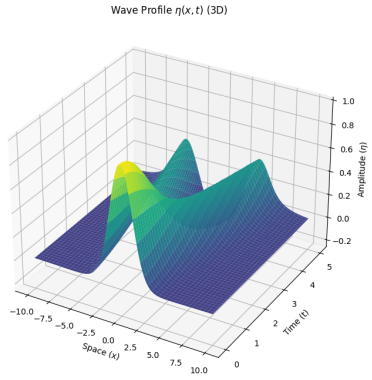


Figure 3. Three-dimensional visualization of the wave profile $\eta(x,t)$ over the entire space-time domain.

Finally, Figure 4 illustrates the distribution of velocity potentials $\phi_1(x,z,t)$ and $\phi_2(x,z,t)$ in the two fluid layers at a fixed time moment. The dashed line marks the interface between fluids, allowing assessment of the solution's continuity across the boundary.

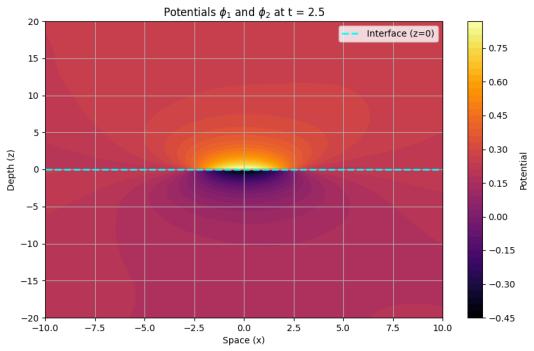


Figure 4. Velocity potentials ϕ_1 and ϕ_2 in the (x,z) plane at time $t = 2.5$. The dashed line indicates the fluid interface.

PINN Model Testing

Analytical Solution and Its Computational Implementation. The analytical solution described by equation (6) was computed using direct numerical integration, rather than relying on fast Fourier transform routines. To evaluate the Fourier coefficients explicitly, the `quad` function from the `scipy.integrate` module was employed. This method ensures high-accuracy evaluation of the integrals defining the spectral decomposition of the initial perturbation.

The calculation involves integrating sine and cosine transforms of the initial profile over a finite spatial domain. The integration range, resolution in wave number space, and the density of sampled points are chosen to capture the essential features of the wave spectrum and accurately reflect dispersive dynamics. Integration precision is controlled through parameters specifying the allowed subdivision of intervals, which is important for resolving localized features of the initial condition.

Analysis and Comparison of Wave Profiles. To compare the results of the PINN model with the analytical solution, a custom Python script was used. The script loads precomputed data and generates visualizations to assess modeling accuracy and evaluate different network configurations.

The analysis includes two types of plots. The first presents individual comparisons at selected time instances, highlighting the evolution of the wave profile as predicted by the PINN and by the analytical solution. The second provides a global view, combining all profiles on a single graph. Here, color is used to represent time, while different line styles distinguish between analytical and PINN-based results. This visualization strategy enables clear identification of phase shifts, amplitude discrepancies, error accumulation over time.

To illustrate the program's capabilities in comparison and visualization, sample plots are provided below.

Figure 5 shows a detailed comparison of wave profiles $\eta(x,t)$ for the analytical solution (solid black line) and the PINN model with architecture L10_N50 with 10 layers, each containing 50 neurons, (dashed blue line) at six different time points ($t = 0.0$ to $t = 5.0$). Each subplot corresponds to a specific moment in time, allowing visual inspection of discrepancies or agreement between solutions during wave packet evolution. At early times ($t = 0.0$, $t = 1.0$), the predictions closely match the analytical solution. However, as time progresses, small deviations emerge, visible in phase shifts and amplitude differences, especially in the wave packet tails.

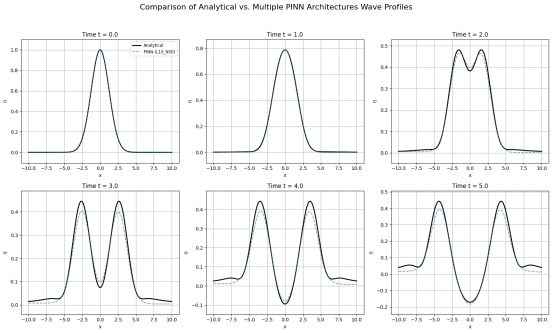


Figure 5. Comparison between analytical solution and PINN predictions (L10_N50) for wave profiles $\eta(x,t)$ at different time points.

Figure 6 presents a combined plot where all wave profiles for both the analytical solution (solid lines) and the PINN model (dashed lines) are displayed together. Line colors encode time, enabling tracking of temporal evolution. Good initial agreement is observed, which gradually deteriorates over time, indicating an accumulation of errors in the PINN model compared to the analytical solution. Differences in phase velocity and dispersion become more pronounced at later stages.

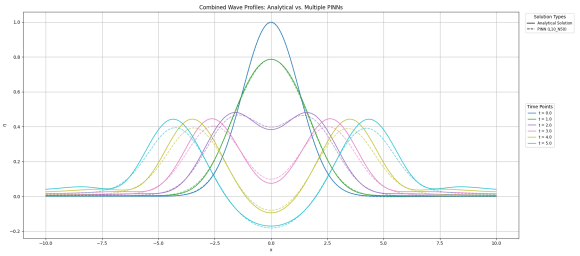


Figure 6. Wave profile evolution at different time points: the analytical solution and PINN (L10_N50) with 10 layers, each containing 50 neurons.

The obtained testing results demonstrate that the PINN model accurately reproduces the wave dynamics at early stages; however, error accumulation becomes noticeable over time, highlighting the need for further tuning of the architecture and training parameters.

Conclusion and Future Research Directions

The results of the first stage of experiments on modeling the linear interfacial wave problem between two fluid layers using PINNs demonstrate satisfactory agreement between the obtained numerical solutions and the analytical results. This confirms the fundamental applicability and effectiveness of the PINN method for problems of this class.

However, in order to improve the model's accuracy and robustness, further training is necessary, along with an investigation of the impact of various learning rate optimization strategies (e.g., `ExponentialDecay` or `ReduceLROnPlateau`) and modifications of the neural network architecture.

The next key stage of this research involves transitioning to the modeling of the nonlinear version of the problem. This will require adapting the PINN model to incorporate the full nonlinear terms in the governing equations and boundary conditions, which poses a more complex and physically rich challenge. Investigating nonlinear wave dynamics will allow for a deeper exploration of phenomena such as soliton formation and wave packet instabilities, as well as an evaluation of the potential of PINNs in solving problems that lack simple analytical solutions.

Acknowledgments. Author expresses gratitude to the Research Council of Lithuania for the support in preparing this article.

References

1. M. Raissi, P. Perdikaris and G. E. Karniadakis, Journal of Computational Physics. **378**, 686–707 (2019).
2. S. Lin and Y. Chen, Journal of Communicational Physics. **457**, 111053 (2022).
3. S. Bhatnagar, A. Comerford and F. Banaeizadeh, Journal of Machine Learning for Modeling and Computing. **5** (1), 39–67 (2024).
4. P. Sharma, W. T. Chung, B. Akoush and M. A. Ihme, Energies. **16**, 2343 (2023).
5. A. B. Buhendwa, S. Adami and N. A. Adams, Machine Learning with Applications. **4**, 100029 (2021).
6. J. Pu, J. Li and Y. Chen, Nonlinear Dynamics. **105**, 1723–1739 (2021).
7. L. Wang and Z. Yan, Physics Letters A. **404**, 127408 (2021).
8. X. Pan, J. Wang, X. Zhang, Y. Mei, L. Shi and G. A. Zhong, International Journal of Remote Sensing. **39** (3), 607–618 (2018).
9. Y. Liu, X. Zhang, Q. Dong, G. Chen and X. Li, Applied Energy. **335**, 121602 (2024).
10. P. Ren, C. Rao, S. Chen, J.-X. Wang, H. Sun and Y. Liu, Computer Physics Communications. **295**, 109010 (2024).
11. C. Song and Y. Wang, Geophysical Journal International. **232**, 1503–1514 (2022).
12. O. Avramenko, Mohyla Mathematical Journal. **7**, 51–56 (2024).

Авраменко О. В., Компан С. В., Сарана М. П.

МАШИННЕ НАВЧАННЯ НА ОСНОВІ PINN ДЛЯ МОДЕЛЮВАННЯ ВНУТРІШНІХ ХВИЛЬ У НАПІВНЕСКІНЧЕННИХ РІДИНАХ

У роботі розглядається застосування фізично інформованих нейронних мереж (Physics-Informed Neural Networks, PINNs) для моделювання хвильових процесів на межі поділу двох нестисливих рідин з різною густиною. На першому етапі дослідження вивчається лінійна постановка задачі, яка допускає аналітичний розв'язок на основі спектрального методу з розкладом початкового збурення у ряд Фур'є. Це розв'язання використовується для тестування та валідації точності передбачень моделі PINN.

Програмну реалізацію виконано мовою Python із використанням спеціалізованих бібліотек TensorFlow, NumPy, SciPy та Matplotlib, що забезпечують як ефективне створення архітектур глибинного навчання, так і чисельне розв'язання задач математичної фізики. Запропонований підхід поєднує можливості штучного інтелекту з галузевими знаннями в галузі гідродинаміки, що дає змогу будувати інтерпретовані та фізично обґрунтовані моделі. Особливу увагу приділено організації експерименту, автоматизації візуалізації та збереженню проміжних результатів для подальшого аналізу. Реалізація PINN включає формулювання функції втрат, яка відображає фізичні рівняння та граничні умови, а навчання нейромережі здійснюється на випадковій вибірці точок у просторово-часовій області. Проаналізовано вплив архітектури моделі та параметрів навчання на точність розв'язання. Візуалізація історії втрат і передбачених профілів хвиль дозволяє оцінити збіжність та фізичну адекватність отриманого розв'язання.

Наведено порівняння результатів моделювання PINN з аналітичним розв'язком у різні моменти часу, виявлено особливості фазових і амплітудних відхилень. Зафіксовано високу відповідність розв'язків на початкових етапах і поступове накопичення похибок у часі, що є типовим для подібних моделей. Отримані результати підтверджують придатність підходу PINN для задач лінійної

гідродинаміки, закладаючи підґрунтя для подальшого поширення на слабко- та сильнонелінійні режими, дослідження стійкості та динаміки нелінійних хвиль.

Ключові слова: фізично інформована нейронна мережа (PINN), функція втрат, тестування нейромережі, профілі хвиль.

Матеріал надійшов 19.08.2025