Comparative Evaluation of Periodic Boundary Condition Approaches in PINNs

(article starts on next page)

Comparative Evaluation of Periodic Boundary Condition Approaches in PINNs

Mohammad Sheikholeslami[1], Saeed Salehi[1,2], Wengang Mao[1], Arash Eslamdoost[1], Håkan Nilsson[1]
[1]Department of Mechanics and Maritime Sciences, Chalmers University of Technology, Sweden
[2]Chalmers Industriteknik, Sweden

Abstract. This study investigates the enforcement of periodic boundary conditions (PBCs) in physics-informed neural networks (PINNs), a key challenge in fluid dynamics. Both soft and hard constraint approaches are examined, comparing their theoretical foundations and practical implications. For soft PBCs, the One-Period approach is shown to be the most effective, preventing convergence to suboptimal solutions. For hard PBCs, various functions are evaluated. Additionally, the required number of neurons in the periodic layer is analyzed, confirming that at least two neurons are necessary for proper optimization and convergence across all tested functions. The linear surface wave problem serves as a benchmark, providing a comprehensive assessment of these strategies. The findings offer valuable insights optimal design choices for PINNs in periodic problems, guiding their application in fluid dynamics and related fields.
Keywords: Physics-Informed Neural Networks (PINNs), Periodic Boundary Conditions, Soft Constraints, Hard Constraints

1. Introduction

Periodic boundary conditions (PBCs) are crucial in fluid mechanics, especially for problems with repeating patterns in an infinite domain. To reduce computational costs, simulations often focus on a single representative period [1-3].

In physics-informed neural networks (PINNs) constraints such as governing equations and boundary conditions are imposed through loss terms, known as soft constraints [4-5]. PBCs can be enforced this way, but their effectiveness depends on optimization success. While soft PBCs are easy to implement without modifying the network architecture or adding trainable variables, they can lead to ill-conditioned systems, reducing solution accuracy [6-7].

PBCs can also be enforced as a hard constraint in PINNs [2,8] by incorporating a dedicated layer into the network architecture. This guarantees exact periodicity, making the enforcement independent of the optimization process. Unlike soft constraints, hard enforcement ensures that the PINN solution satisfies the PBC to machine precision. Various studies have explored different functions for hard PBC enforcement [2, 8, 9], typically by introducing a custom layer after the input layer. However, the choice of functions and the number of neurons in this layer vary across different investigations.

In this paper, we investigate the enforcement of PBCs in PINNs using two approaches for soft constraints and multiple functions for hard (exact) enforcement.

2. Periodic boundary condition as a soft constraint

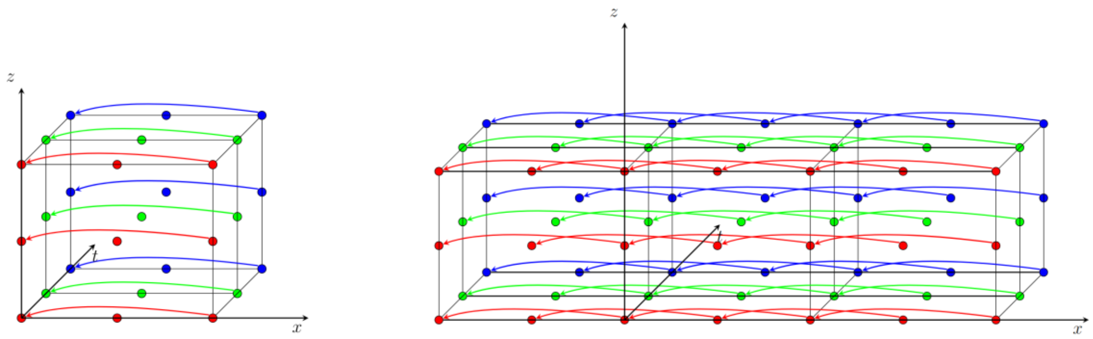
This section investigates two approaches for defining loss terms to enforce PBCs. For a target function $F(x)$ with periodicity L, a loss term can penalize mismatches at the domain boundaries. However, additional loss terms are required to address mismatches in function derivatives, making this approach inherently limited. In the One-Period approach, the domain represents a single period of the function, as illustrated in Figure 1(a), and the PBC loss is defined as

$$Loss_{periodic} = \|F(0) - F(L)\|_2^2. \quad (1)$$

An alternative, the Three-Period approach, enforces periodicity by ensuring all collocation points inside the domain match their counterparts shifted by one full period forward or backward. The domain spans three consecutive periods, as shown in Figure 1(b), and the loss term is given by

$$Loss_{periodic} = \|F(x+L) - F(x)\|_2^2 + \|F(x-L) - F(x)\|_2^2. \quad (2)$$

The One-Period approach has been applied to linear free surface waves [10, 11], compressor cascade flow [12] and flow between parallel plates [13], while the Three-Period approach has been used solve the heat equation [14].



(a) (b)

Figure 1. One-Period (a) and Three-Period (b) approaches for setting soft period boundary condition

3. Periodic boundary condition as a hard constraint

Two widely used methods for enforcing PBCs in PINNs were developed by Dong and Ni [2] and Lu et al. [8]. Dong and Ni [2] proposed using the functions shown on the left side of Figure 2, where A, B, and C are trainable parameters. The parameter α is defined as $\alpha = 2\pi/L$, with L representing the function's period, and σ denotes a nonlinear activation function such as $tanh$. In contrast, Lu et al. [8] suggested constructing the periodic layer using Fourier series basis function, as illustrated on the right side of Figure 2.

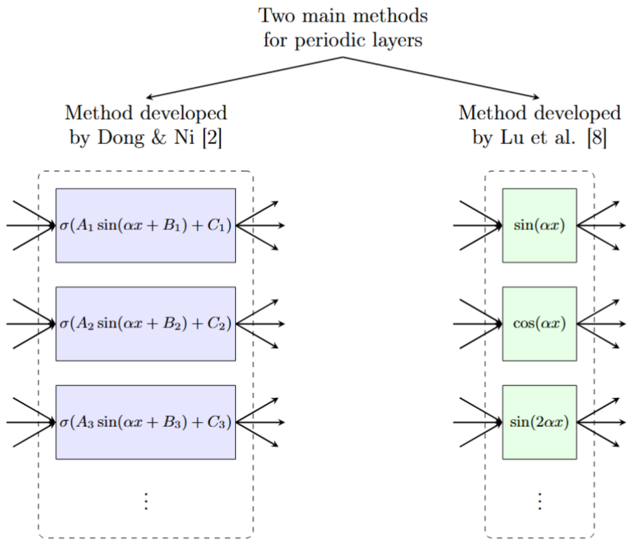


Figure 2. Methods developed for hard PBCs

The following subsections examine the number of neurons in the periodic layer and the choice of functions used within it.

3.1 Number of neurons in the periodic layer

Lu et al. [8] stated that including the first two terms of the Fourier series in the periodic layer is sufficient to maintain accuracy, although adding more neurons can be beneficial. Here, we investigate whether this holds for the function proposed by Dong and Ni [2] and its simplified variations. We begin with the simplified version, $\sin(\alpha x + B)$, instead of the full function $\sigma(A\sin(\alpha x + B) + C)$.

A PINN with a periodic layer must satisfy two key requirements:

1. **Exact Periodicity:** The PBC must be enforced to machine precision, ensuring strict periodicity.
2. **Unconstrained Periodic Solutions:** The model must be able to converge to any valid periodic solution, with the periodic layer introducing no restrictions beyond enforcing periodicity.

Dong and Ni [2] proved that as long as the periodic layer applies a periodic function to its input, the PINN's output remains periodic. This ensures the first requirement is met, regardless of the number of neurons in the periodic layer. However, the second requirement depends on the number of neurons. Lu

et al. [8] referenced Zhang et al. [9] in stating that two neurons (sine and cosine of a single Fourier frequency) in the periodic layer allow the PINN to approximate any periodic function. We examine whether this also applies to the simplified function $\sin(\alpha x + B)$.
Lu et al. [8] argued that any function can be expressed as a nonlinear function $\sin(x)$ and $\cos(x)$. Considering a PINN with a periodic layer as a nonlinear function $F(x)$, it can be represented as

$$F(x) = F(f_1(\sin(\alpha x), \cos(\alpha x)), f_2(\sin(\alpha x), \cos(\alpha x)), \ldots, f_n(\sin(\alpha x), \cos(\alpha x))) = F(\sin(\alpha x), \cos(\alpha x)), \quad (3)$$

where each neuron is a function f_i of both sine and cosine terms, allowing independent control of their contributions during training. This concept in illustrated in Figure 3, which provides a schematic representation of a PINN incorporating a periodic layer with two neurons.

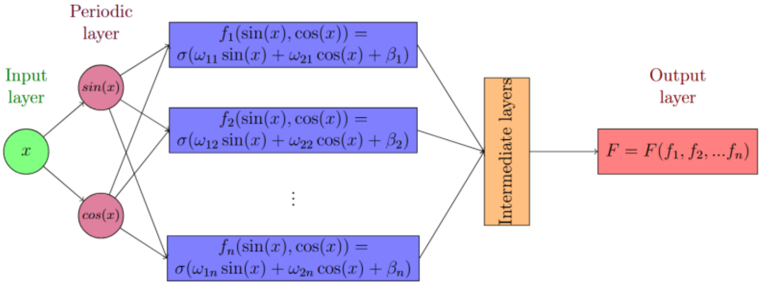


Figure 3. Schematic of a generic PINN having a periodic layer with two neurons

Now, if the periodic layer includes only one neuron using $\sin(\alpha x + B)$, the function in the subsequent hidden layer can be rewritten as

$$f(x) = f\big(\sin(\alpha x + B_1)\big) = f(D_1 \sin(\alpha x) + D_2 \cos(\alpha x)), \quad (4)$$

where:

$$D_1 = \cos(B_1), \qquad D_2 = \sin(B_1). \quad (5)$$

This introduces a constraint, which is D_1 and D_2 are dependent through $D_1^2 + D_2^2 = 1$, preventing the optimizer from independently adjusting the sine and cosine terms. Adding a second neuron with $\sin(\alpha x + B_2)$ allows the function in a neuron after of the periodic layer to be expressed as:

$$f(x) = f\left(\omega_1\big(D_1\sin(\alpha x) + D_2\cos(\alpha x)\big) + \omega_2\big(D_3\sin(\alpha x) + D_4\cos(\alpha x)\big)\right), \quad (6)$$

where

$$D_1 = \cos(B_1), \qquad D_2 = \sin(B_1), \qquad D_3 = \cos(B_2), \qquad D_4 = \sin(B_2). \quad (7)$$

As long as the determinant of the transformation matrix

$$det\begin{bmatrix} D_1 & D_3 \\ D_2 & D_4 \end{bmatrix} \neq 0, \quad (8)$$

the sine and cosine terms remain independent, ensuring that ω_1 and ω_2 can be adjusted freely by the optimizer. Thus, at least two neurons are required to allow a PINN to approximate any periodic solution. Including trainable parameters A and C in the function $\sin(\alpha x + B)$, and turning it to $A\sin(\alpha x + B) + C$ does not change the fundamental requirement, and two neurons remain essential. Additional neurons can provide more flexibility but are not strictly necessary. However, incorporating an activation function, forming $\sigma(\sin(\alpha x + B))$ or the full function $\sigma(A\sin(\alpha x + B) + C)$, complicates the analysis. Since different activation functions can influence the behaviour of the periodic layer, the impact of $\sigma = \tanh$ is explored in case studies in the following sections.

3.2 Periodic layer function selection

In order to design the periodic layer of a PINN, there are a few alternatives for the function applied in the neurons of this layer. The first is the Fourier series basis functions, proposed by Lu et al. [8]. Four variations of the function proposed by Dong and Ni [2] are investigated in this study, as mentioned in Table 1.

Table 1. Original and simplified versions of the periodic function proposed by Dong and Ni [2]

Original function	$\sigma(A\sin(\alpha x + B) + C)$
Simplified function #1	$A\sin(\alpha x + B) + C$

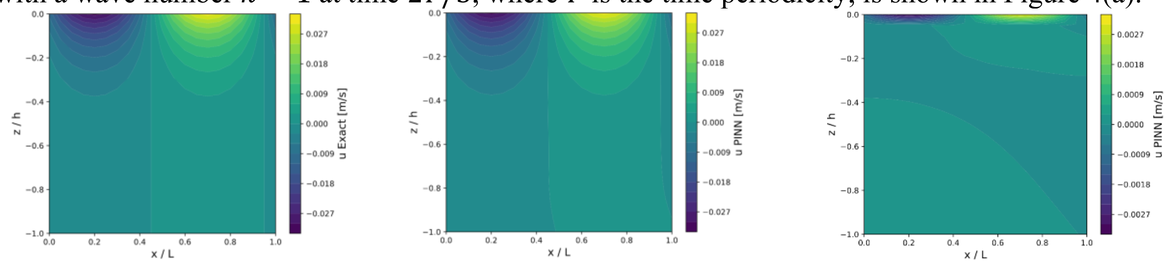
Simplified function #2	$\sin(\alpha x + B)$
Simplified function #3	$\sigma(\sin(\alpha x + B))$

The original function and simplified function #3 in Table 1 include an activation function. Given the wide variety of possible activation functions, making a general statement is difficult. Therefore, case studies will be conducted specifically for $\sigma = \tanh$.

Comparing simplified functions #1 and #2, each neuron in function #1 has two extra trainable variables, increasing computational cost. However, these additional variables do not enhance the network's flexibility. In other words, they do not expand the dimensions of the solution space. The reason is that the neuron's output is multiplied by a weight passing to the next neuron. The weight can be adjusted to produce the same effect as the variables A and C. Thus, A and C do not provide any additional flexibility to the PINN.

4. Case studies and results

This section compares different approaches for soft constraints and functions for hard constraints in enforcing PBCs in PINNs for linear water waves. A PINN with a feedforward neural network, consisting of eight hidden layers with ten neurons each, was trained to determine the velocity potential ϕ of the flow using the Adam and L-BFGS optimizers with $tanh$ as the activation function. The details of the linear wave theory, case study, and PINN hyperparameters follow the works by Sheikholeslami et al. [10, 11]. The exact distribution of $u = \partial\phi/\partial x$ from the analytical solution of the linear wave theory with a wave number $k = 1$ at time $2T/3$, where T is the time periodicity, is shown in Figure 4(a).



(a) (b) (c)

Figure 4. Exact solution (a), optimal PINN solution (b), and suboptimal PINN solution (c) of u at time $2T/3$, illustrating the possible convergence of the model.

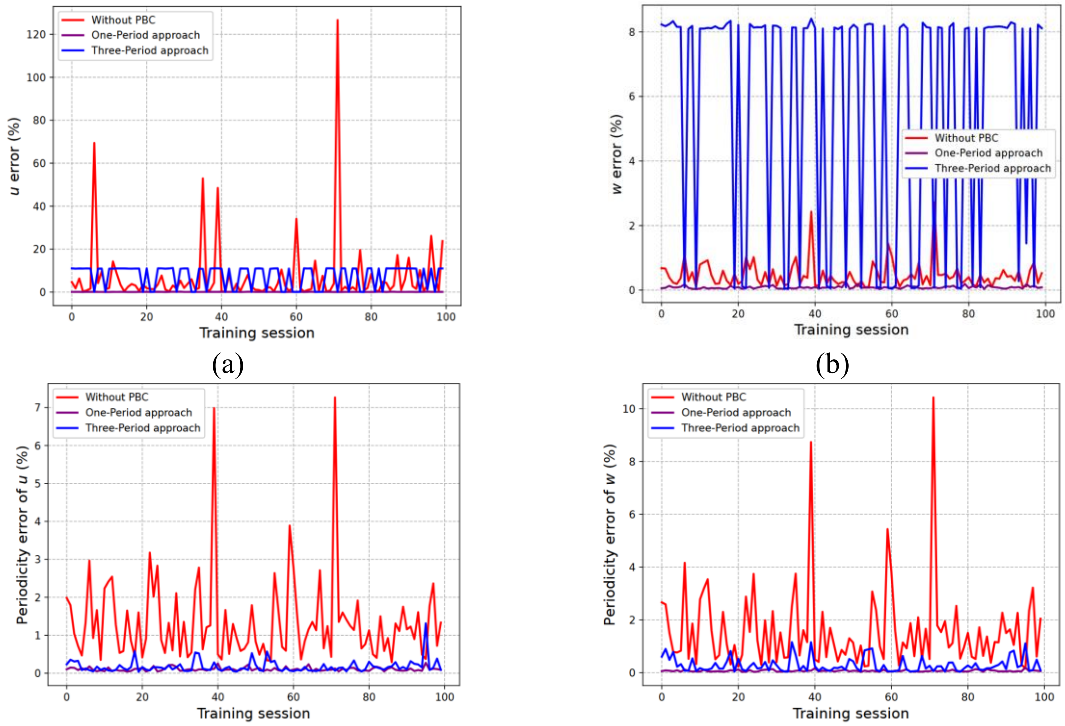
4.1 Soft periodic boundary conditions

Two approaches of enforcing PBCs as a soft constraint are compared. Both approaches are applied to the target function ϕ and its spatial derivatives, $u = \partial\phi/\partial x$, and $w = \partial\phi/\partial z$, resulting in three periodicity-related loss terms in the PINN. The One-Period and Three-Period approaches are evaluated based on their impact, as shown in Figure 5, which presents the total errors in u and w across the computational domain. Each approach is evaluated over 100 training sessions with different initializations, and the average error is analyzed. The One-Period approach reduces the average errors in u and w to 0.09 and 0.09, respectively, compared to 7.07% and 0.46% in the PINN without a PBC. Notably, these error values represent the average over 100 training sessions with each session's error being normalized based on the maximum exact solution in the domain, following the methodology outlined by Sheikholeslami et al. [10, 11].

Figure 5 also shows that the Three-Period approach leads to a suboptimal solution in 69% of training sessions, where the model consistently converges to the same incorrect pattern with an average error of 10.98%. The remaining 31% reach the correct solution with an average error of 0.12%. Figure 4 presents the exact solution alongside the optimal and suboptimal solutions at time $2T/3$, where T is time periodicity. This issue could potentially be mitigated by using transfer learning [13] or by employing adaptive weights.

Figure 5 shows that the One Period-approach reduces the average periodicity error of u and w over 100 training sessions from 1.34% and 1.58% to 0.10% and 0.07% respectively, while the Three-Period approach results in errors of 0.17% and 0.29%. These errors are calculated by comparing the values of

u and w at the domain boundaries and normalizing them with the maximum exact value of the function in the domain.



(a) (b)

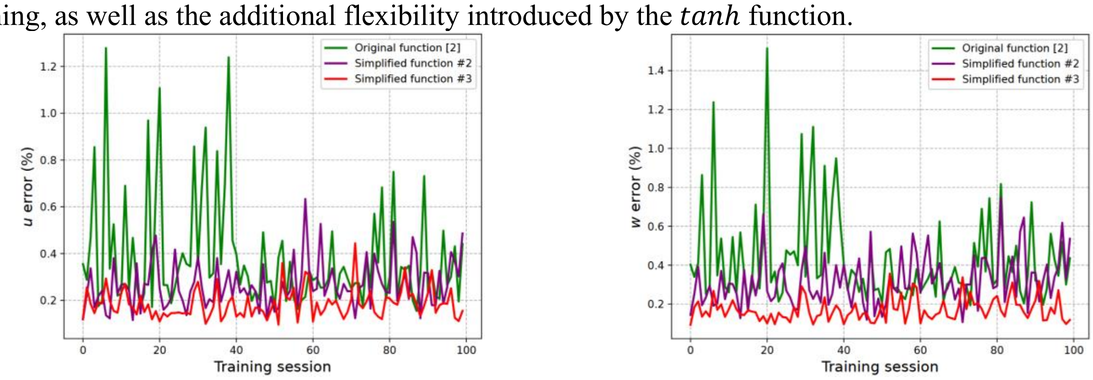
(c) (d)

Figure 5. Errors in u (a) and w (b) without PBC and with PBCs using the One-Period and Three-Period approaches

4.2 Hard periodic boundary conditions

This section examines the effects of different functions used in the periodic layer for enforcing PBCs as a hard constraint. Figure 6 presents the errors for various functions from Table 1, excluding simplified function #1, which was shown in section 3.2 to offer no advantage over simplified function #2. Each function is evaluated over 100 training sessions with different initializations. The error values for each training session are normalized based on the maximum value of the exact solution in the domain, following the approach of Sheikholeslami et al. [10, 11]. All PINNs used in this study included two neurons in the periodic layer, regardless of the function employed. The results showed that in all cases, two neurons were sufficient for the PINNs to converge to the correct solution.

The results indicate that simplified function #3 achieved the lowest average total errors over 100 training sessions, with 0.18% for u and 0.17% for w, compared to 0.38% and 0.43% for the original function and 0.27% and 0.32% for simplified function #2. This improvement can be attributed to the absence of unnecessary trainable variables A and C, which may require a more complex architecture for effective training, as well as the additional flexibility introduced by the $tanh$ function.



(a) (b)

Figure 6. Errors in u (a) and w (b) with different functions presented in Table 1

5. Conclusion

This study investigated the enforcement of periodic boundary conditions (PBCs) in physics-informed neural networks (PINNs) through both soft and hard constraints. It was demonstrated analytically and confirmed through case studies that the function proposed by Dong and Ni [2] and its simplified variations require at least two neurons in the periodic layer to ensure proper optimization and convergence. This finding aligns with the conclusion of Lu et al. [8], who also stated that at least two neurons are necessary for enforcing periodicity in their method.

For soft PBCs, the One-Period approach proved to be the most effective, preventing the model from getting trapped in local minima while also reducing computational cost. This approach significantly improved accuracy, lowering the errors in u and w from 7.07% and 0.46 to 0.09% and 0.09%, respectively, in the studied linear wave problem.

For hard PBCs, various functions were tested, with $tanh(sin(\alpha x + B))$ yielding the lowest errors for u and w in the studied linear wave problem, reducing the total error to 0.18% for u and 0.17% for w. These results offer valuable insights into optimal design choices for PINNs in periodic problems, providing guidance for future applications in fluid dynamics and related fields.

6. Acknowledgments

The work was financed by the Department of Mechanics and Maritime Sciences at Chalmers University of Technology. It was carried out as a part of the "Swedish Centre for Sustainable Hydropower – SVC".

7. References

[1] S. Dong, "Turbulent flow between counter-rotating concentric cylinders: a direct numerical simulation study," Journal of Fluid Mechanics, vol. 615, pp. 371–399, 2008. https://doi.org/10.1017/S0022112008003716

[2] S. Dong, N. Ni, "A method for representing periodic functions and enforcing exactly periodic boundary conditions with deep neural networks," Journal of Computational Physics, vol. 435, 110242, 2021.

[3] S. Dong, G.E. Karniadakis, A. Ekmekci, D. Rockwell, "A combined direct numerical simulation–particle image velocimetry study of the turbulent near wake," Journal of Fluid Mechanics, vol. 569, pp. 185–207, 2006. https://doi.org/10.1017/S0022112006002606

[4] Z. Chen, S.K. Lai, Z. Yang, Y.Q. Ni, Z. Yang, K.C. Cheung, "AT-PINN-HC: A refined time-sequential method incorporating hard-constraint strategies for predicting structural behavior under dynamic loads," Computer Methods in Applied Mechanics and Engineering, vol. 436, p. 117691, 2025. https://doi.org/10.1016/j.cma.2024.117691

[5] H. Chen, G.E. Constante Flores, C. Li, "Physics-informed neural networks with hard linear equality constraints," Computers & Chemical Engineering, vol. 189, p. 108764, 2024. https://doi.org/10.1016/j.compchemeng.2024.108764

[6] B. Hao, U. Braga-Neto, C. Liu, L. Wang, M. Zhong, "Structure Preserving PINN for Solving Time Dependent PDEs with Periodic Boundary," arXiv preprint, 2024. https://arxiv.org/abs/2404.16189

[7] L. D. McClenny, U. M. Braga-Neto, "Self-adaptive physics-informed neural networks," *Journal of Computational Physics*, vol. 474, p. 111722, 2023. DOI: 10.1016/j.jcp.2022.111722.

[8] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, S.G. Johnson, "Physics-Informed Neural Networks with Hard Constraints for Inverse Design," SIAM Journal on Scientific Computing, vol. 43, no. 6, pp. B1105-B1132, 2021. https://doi.org/10.1137/21M1397908

[9] D. Zhang, L. Guo, G. E. Karniadakis, "Learning in Modal Space: Solving Time-Dependent Stochastic PDEs Using Physics-Informed Neural Networks," *arXiv preprint*, 2019. Available: arXiv:1905.01205.

[10] M. Sheikholeslami, S. Salehi, W. Mao, A. Eslamdoost, and H. Nilsson, "Physics-Informed Neural Networks for Modeling Linear Waves," International Conference on Offshore Mechanics and Arctic Engineering (OMAE), Volume 9: Philip Liu Honoring Symposium on Water Wave Mechanics and Hydrodynamics; Blue Economy Symposium, Singapore, June 2024. DOI: 10.1115/OMAE2024-125048.

[11] M. Sheikholeslami, S. Salehi, W. Mao, A. Eslamdoost, H. Nilsson, Physics-Informed Neural Networks with Hard and Soft Boundary Conditions for Linear Free Surface Waves. Available at SSRN: https://ssrn.com/abstract=5156693

[12] Z. Li, F. Montomoli, S. Sharma, "Investigation of Compressor Cascade Flow Using Physics-Informed Neural Networks with Adaptive Learning Strategy," AIAA Journal, vol. 62, no. 4, pp. 1400-1410, 2024. https://doi.org/10.2514/1.J063562

[13] S. Shah, N.K. Anand, "Physics-informed neural networks for periodic flows," Physics of Fluids, vol. 36, no. 7, p. 073620, 2024. https://doi.org/10.1063/5.0216266.

[14] M. Ngom, O. Marin, "Fourier neural networks as function approximators and differential equation solvers," Statistical Analysis and Data Mining: The ASA Data Science Journal, vol. 14, no. 6, pp. 647–661, 2021. DOI: 10.1002/sam.11531.