Benchmarking Physics-Informed Neural Networks and Boundary Element Method: Accuracy–Efficiency Trade-offs in Wave Scattering

O. Rincón-Cardeño[1], G. Pérez-Bernal[1], S. Montoya-Noguera[2], and N. Guarín-Zapata[1,*]

[1] Mathematical Applications in Science and Engineering Research Group, School of Applied Sciences and Engineering, Universidad EAFIT, Medellín, Colombia

[2] Nature and Cities Research Group, School of Applied Sciences and Engineering, Universidad EAFIT, Medellín, Colombia

[*] Correspondence: nguarinz@eafit.edu.co

This supplementary material provides additional methodological details supporting the analyses presented in the main manuscript. First, an analytical solution of the two-dimensional Helmholtz equation for plane-wave scattering by a cylindrical obstacle is presented and used as an exact reference to validate the numerical results. Subsequently, the hyperparameter optimization procedure adopted for the Physics-Informed Neural Networks is described, detailing the selection of architectural and training parameters that ensure a consistent and reliable baseline for the comparative study. References to equations in the main manuscript are explicitly indicated as "equation $(\cdot)$ in the main manuscript".

S1. Analytical solution

The Helmholtz equation in polar coordinates is expressed as:

$$\frac{\partial^2 R(r)}{\partial r^2} + \frac{1}{r}\frac{\partial R(r)}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u(r,\theta)}{\partial \theta^2} + k^2 u(r,\theta) = 0. \tag{1}$$

Assuming the solution can be written as the product of radial and angular components:

$$u(r,\theta) = R(r)\Theta(\theta), \tag{2}$$

substituting equation (2) into equation (1) yields to

$$\frac{r^2}{R(r)}\left[\frac{\partial^2 R(r)}{\partial r^2} + \frac{1}{r}\frac{\partial R(r)}{\partial r} + k^2 R(r)\right] + \frac{1}{\Theta(\theta)}\frac{d^2\Theta(\theta)}{d\theta^2} = 0,$$

which separates into two ordinary differential equations, one for $R(r)$ and other for $\Theta(\theta)$. Since r and θ are independent variables:

$$\frac{r^2}{R(r)}\left[\frac{\partial^2 R(r)}{\partial r^2} + \frac{1}{r}\frac{\partial R(r)}{\partial r} + k^2 R(r)\right] = -\frac{1}{\Theta(\theta)}\frac{d^2\Theta(\theta)}{d\theta^2} = n^2. \tag{3}$$

From equation (3), we obtain the radial part:

$$r^2\frac{d^2R(r)}{dr^2} + r\frac{dR(r)}{dr} + \left[r^2k^2 - n^2\right]R(r) = 0.$$

Now, introducing the substitution $\hat{r} = kr$ and $\hat{R}(\hat{r}) = R(\hat{r}/k) = R(r)$:

$$\hat{r}^2\frac{d^2\hat{R}(\hat{r})}{d\hat{r}^2} + \hat{r}\frac{d\hat{R}(\hat{r})}{d\hat{r}} + \left[\hat{r}^2 - n^2\right]\hat{R}(\hat{r}) = 0. \tag{4}$$

This is Bessel's differential equation. Consequently, the solution to equation (4) is:

$$R(r) = BH_n^{(1)}(kr) + CH_n^{(2)}(kr) \tag{5}$$

where $H_n^{(1)}(kr)$ and $H_n^{(2)}(kr)$ are the Hankel functions of the first and second kinds. Additionally, from equation (3), we obtain the angular equation:

$$\frac{d^2\Theta(\theta)}{d\theta^2} + n^2\Theta(\theta) = 0.$$

Therefore, $\Theta(\theta)$ can be expressed as:

$$\Theta(\theta) = De^{in\theta}. \tag{6}$$

Substituting equations (5) and (6) into equation (2), we obtain the exact solution for $u(r, \theta)$:

$$u(r,\theta) = \sum_{n=-\infty}^{\infty} E_n J_n(kr)e^{in\theta} + F_n H_n^{(1)}(kr)e^{in\theta}.$$

The incident and scattered fields $u_{\text{inc}}(r, \theta)$ and $u_{\text{sct}}(r, \theta)$ are given by:

$$u_{\text{inc}}(r,\theta) = \sum_{n=-\infty}^{\infty} E_n J_n(kr)e^{in\theta}, \tag{7}$$

and

$$u_{\text{sct}}(r,\theta) = \sum_{n=-\infty}^{\infty} F_n H_n^{(1)}(kr)e^{in\theta}. \tag{8}$$

From equation (3) of the main manuscript and 7, we have that $E_n = i^n$. By applying the Neumann boundary condition expressed in equation (4) of the main manuscript, we can determine the scattered field coefficients F_n as:

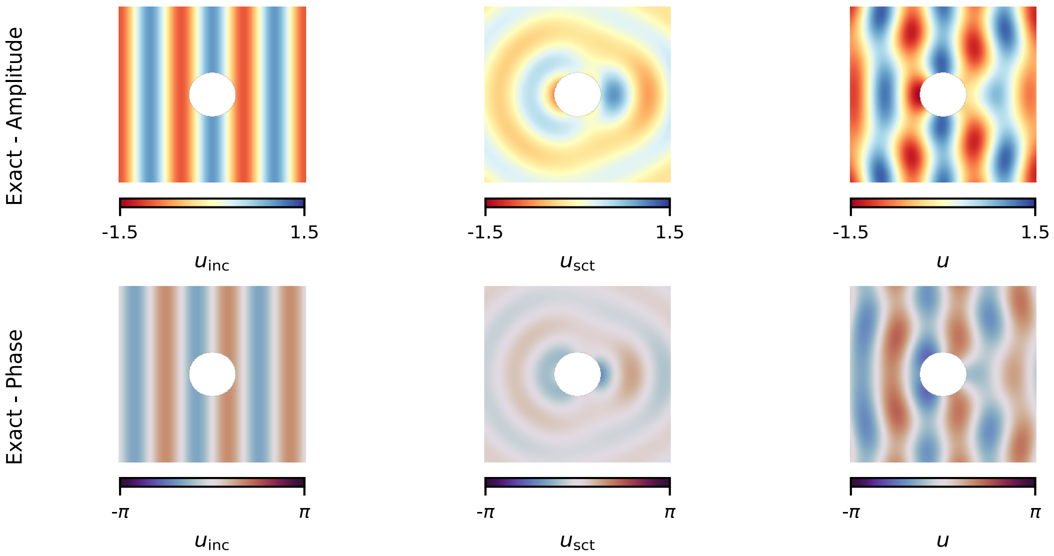
$$F_n = -\frac{E_n J_n'(kr)}{H_n^{(1)\prime}(kr)}.$$

Using equation (8), the scattered field can be computed analytically. Supplementary Figure 1 shows the resulting incident, scattered, and total fields. In our experiment, we used the parameters $k = 3\,\text{m}^{-1}$, $r_{\text{i}} = \frac{\pi}{4}\,\text{m}$, and $l_{\text{e}} = \pi\,\text{m}$.

S2. Hyperparameter optimization

The performance of the PINN depends on the choice of hyperparameters, which define the network architecture and the training procedure. In this study, the activation function σ, the number of hidden layers L, the number of neurons per layer N, and the learning rate α were tuned, following the procedure described in Escapil-Inchauspé and Ruz (2023). We specified a fixed random seed to make the hyperparameters suggested by Optuna reproducible.

The hyperparameter ranges explored were: $\alpha \in \{10^{-2}, 10^{-3}, 10^{-4}\}$, $L \in \{1, 2, 3\}$, $N \in \{25, 50, 75\}$, and $\sigma \in \{\text{Tanh, Sigmoid, Sine}\}$. This procedure involves a bi-level optimization scheme. In the outer loop, the hyperparameters are optimized, while in the inner loop the coefficients (weights and biases) of the neural network are trained. This scheme allows the network parameters to be estimated under



Supplementary Figure 1. Analytical estimation of wave scattering. The first column shows the incident plane wave field (u_{inc}), the second column shows the corresponding scattered wave field (u_{sct}), and the third column shows their superposition (u). The top row presents the field amplitudes ($\Re(u(x,y))$), while the bottom row shows the corresponding phases ($\Im(u(x,y))$).

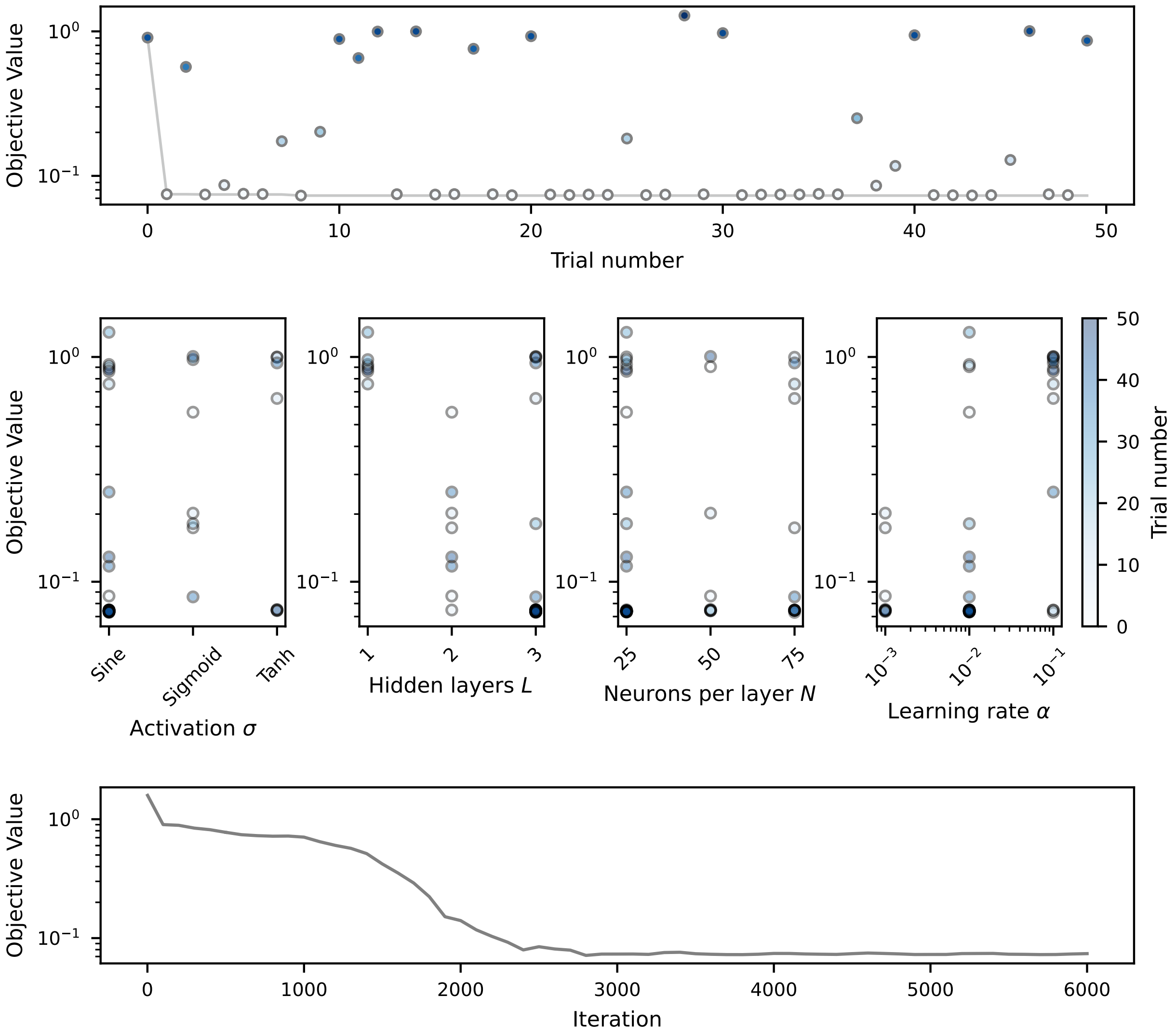
the chosen hyperparameters. Supplementary Figure 2 summarizes the optimization results across 50 trials, highlighting both the effect of each hyperparameter and the training dynamics of the selected configuration. From these results, it is possible to identify which settings lead to lower objective values.

The scatter plots in the middle row of Supplementary Figure 2 illustrate the influence of each hyperparameter on the objective value. Among the activation functions, both Sine and Tanh achieve similar errors on the order of 10^{-2}. This level of accuracy is obtained with different combinations of the remaining hyperparameters. Among these, the configuration with three layers and 25 neurons achieved the lowest error with the fewest coefficients (2,026 coefficients). Consequently, for the configuration with $L = 3$ and $N = 25$, an optimal choice was $\sigma =$ Sine and $\alpha = 10^{-2}$, yielding the lowest error in our study of 7.326×10^{-2}. This result indicates that the selected configuration provides sufficient accuracy to represent the governing equations and serves us as a reliable baseline for the subsequent comparative analysis.

The bottom panel shows the training dynamics of the selected configuration using two optimization methods: Adam followed by L-BFGS. Adam rapidly reduced the objective value during the first 1,000 iterations. Subsequent optimization with L-BFGS over the next 5,000 iterations further decreased the objective value, demonstrating the effectiveness of combining first-order and quasi-Newton methods for training PINNs.

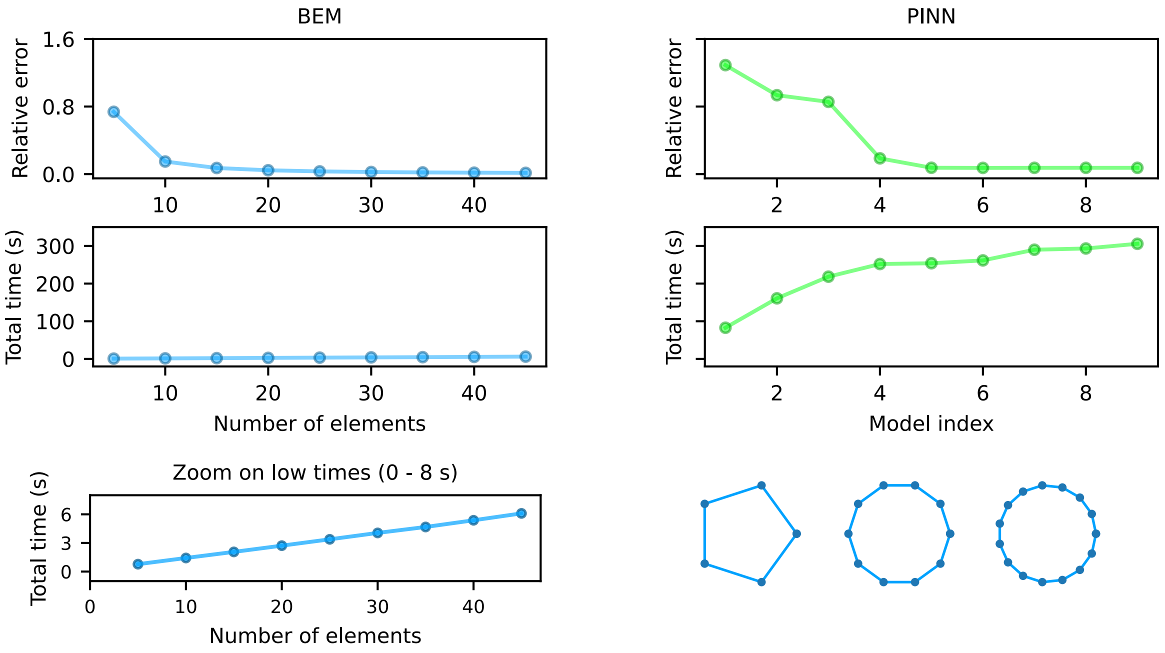
S3. Convergence analysis and computational time

A convergence analysis with respect to the number of integration points is presented in Fig. 3. This analysis is performed to assess the accuracy and computational cost of the numerical integration scheme used in the BEM formulation. As shown in Fig. 3(top-left), the relative error decreases as the number of integration points increases from 5 to 45, demonstrating a convergence behavior. The most



Supplementary Figure 2. Hyperparameter optimization results obtained with Optuna. The top panel shows the optimization history, i.e., the evolution of the objective value across trials. The middle panels present slice plots, illustrating the relationship between each hyperparameter—the activation function σ, the number of hidden layers L, the number of neurons per layer N, and the learning rate α of the Adam optimizer—and the corresponding objective value. The bottom panel displays the training history, with Adam applied during the first phase (1,000 iterations) and L-BFGS during the second phase (5,000 iterations).

significant reduction in error is observed for coarse discretizations, while for $n \gtrsim 30$ the solution begins to stabilize, indicating that convergence is effectively achieved. In addition to accuracy, Fig. 3(bottom-left) shows the total computational time, defined as the sum of solution and evaluation times. As expected, the computational cost increases moderately with the number of integration points. For comparison, Fig. 3(top-right)–(bottom-right) present the corresponding results for the PINN models. In this case, the horizontal axis represents different network architectures (Models 1–9), ordered by increasing model complexity, defined in terms of the number of hidden layers and hidden units. The PINN results exhibit an initial decrease in error with increasing model complexity, followed by a clear saturation, where further increases in model size do not lead to significant improvements.



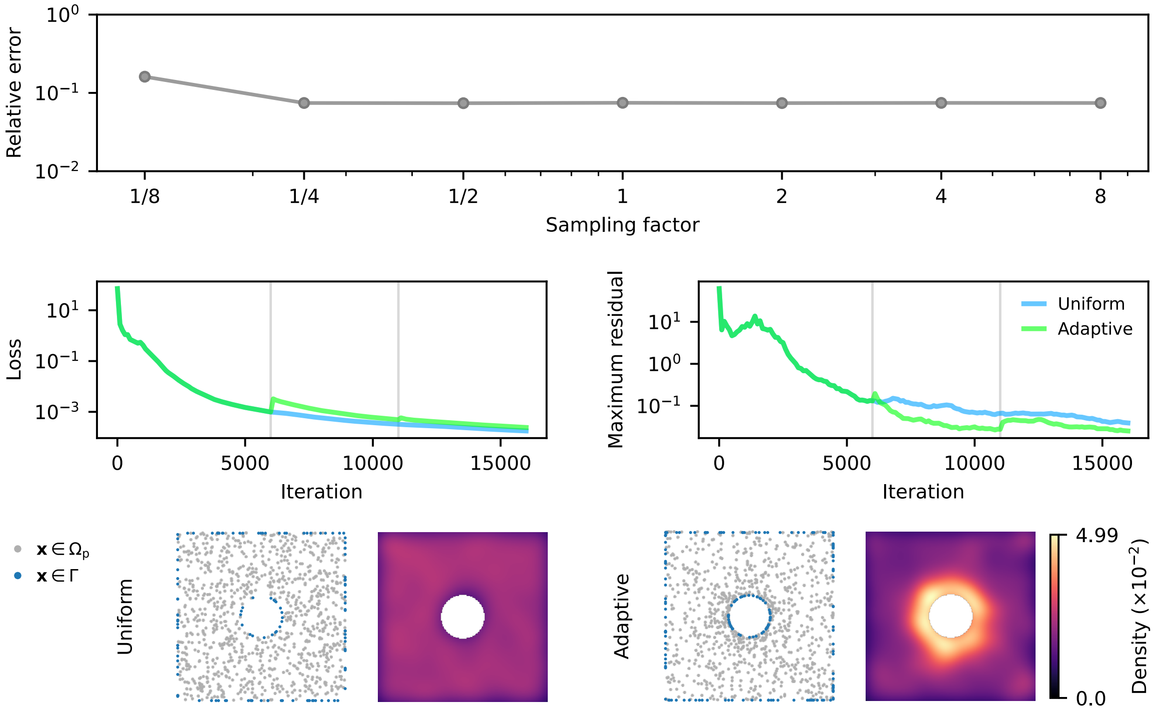
Supplementary Figure 3. Comparison of accuracy and total computational time for BEM and PINN approaches. The top row shows the relative error. For BEM, it is presented as a function of the number of boundary elements, exhibiting convergence with mesh refinement, while for PINNs it is shown across different model configurations (Models 1–9, corresponding to architectures with 1–3 hidden layers and 25–75 neurons per layer). The middle row presents the total computational time. For BEM, it is defined as the sum of assembly and evaluation times, while for PINNs it corresponds to the sum of training and evaluation times. The bottom row provides additional insight. On the left, a zoomed-in view of the BEM computational time in the range 0–8 s highlights its growth behavior. On the right, the boundary discretizations illustrate the refinement of the geometry representation.

S4. PINNs sampling points analysis

To further investigate the influence of the training points distribution on the performance of PINNs, two complementary experiments were conducted (Supplementary Figure 4). The effect of increasing the number of training points was analyzed to determine whether the prediction accuracy can be systematically improved. In this setting, the baseline number of collocation points in the interior domain ($N_{\Omega_\mathrm{P}} = 10{,}000$) and on the boundaries ($N_{\Gamma_\mathrm{I}} = 100$ and $N_{\Gamma_\mathrm{E}} = 1{,}000$) was selected based on values previously reported as effective for scattering problems [2]. The results show that reducing the number of training points by a factor of $1/8$ leads to a reduction in accuracy, with the relative error increasing to the order of 10^{-1}. Increasing the number of points by factors of 4 and 8 does not resulted in a significant improvement, as the relative error remains on the order of 10^{-2}.

This behavior indicates that, for the chosen hyperparameter configuration, insufficient sampling negatively impacts accuracy, while increasing the number of points can improve prediction accuracy, as it enhances the enforcement of the governing equations. However, this improvement is limited by other hyperparameters, including the expressiveness of the network.

A comparison between a uniform sampling strategy and an adaptive sampling approach was performed to evaluate their impact on prediction accuracy. The adaptive sampling strategy was based on the work of Wu et al., specifically the residual-based adaptive distribution proposed in [3]. Re-



Supplementary Figure 4. Sampling in PINNs. Top: Relative error as a function of the sampling factor ($1/8$, $1/4$, 1, 4, 8). Middle: Training loss (left) and maximum residual (right) as functions of the training iterations for uniform and adaptive strategies. Vertical gray lines indicate the iterations at which resampling is performed in the adaptive approach. Bottom: Spatial distribution of collocation points and corresponding sampling densities.

garding the training procedure, both strategies were trained for a total of 15,000 L-BFGS iterations following an initial stage of 1,000 Adam iterations. In the case of uniform sampling, the same fixed set of collocation points was used throughout the entire training process. In contrast, the adaptive approach employed a resampling strategy, where the collocation points were updated during training. Specifically, after the initial 1,000 Adam iterations, the model was trained for 5,000 L-BFGS iterations, followed by two additional cycles of 5,000 L-BFGS iterations with intermediate resampling of the collocation points using the adaptive method.

This procedure ensures that the total number of L-BFGS iterations remains the same for both strategies. The adaptive strategy concentrates points in regions with high residuals (e.g., near the inner boundary), resulting in similar values of the training loss and maximum residual, as well as a more uniform spatial distribution of the residuals.

References

[1] Paul Escapil-Inchauspé and Gonzalo A. Ruz. Hyper-parameter tuning of physics-informed neural networks: Application to Helmholtz problems. *Neurocomputing*, 561:126826, December 2023. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.126826.

[2] Siddharth Nair, Timothy F. Walsh, Greg Pickrell, and Fabio Semperlotti. Multiple scattering simulation via physics-informed neural networks. *Engineering with Computers*, July 2024. ISSN

1435-5663. doi: 10.1007/s00366-024-02038-3.

[3] Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *arXiv preprint arXiv:2207.10289*, 2022. URL `https://arxiv.org/abs/2207.10289`.