Comparative Analysis of Wave Scattering Numerical Modeling Using the Boundary Element Method and Physics-Informed Neural Networks

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

Abstract

Purpose – This study compares the Boundary Element Method (BEM) and Physics-Informed Neural Networks (PINNs) for solving the two-dimensional Helmholtz equation in wave scattering problems. The objective is to evaluate the performance of both methods under the same conditions.

Design/methodology/approach – We solve the Helmholtz equation using BEM and PINNs for the same scattering problem. PINNs are trained by minimizing the residual of the governing equations and boundary conditions with their configuration determined through hyperparameter optimization, while BEM is applied using boundary discretization. Both methods are evaluated in terms of solution accuracy, computation time, and generalization capacity.

Findings – Numerical experiments were conducted by varying the number of integration points for BEM and the number of layers and neurons per layer for PINNs. Hyperparameter tuning identified an adequate PINN configuration for this particular problem as a network with 3 hidden layers, 25 neurons per layer, a learning rate of 10^{-2} and a Sine activation function. At comparable accuracy, the training time required for PINNs is on the order of 10^2 s, one order of magnitude longer than BEM (order 10^1 s). However, once trained, PINNs achieved evaluation times on the order of 10^{-1} s, two orders of magnitude faster than BEM. The generalization capacity was also assessed outside the PINN training domain, where the relative error increased three orders of magnitude, while BEM maintained a consistent error level in the extended region.

Originality/value – This work presents a direct comparison between BEM and PINNs for the Helmholtz equation. The analysis provides quantitative data on the performance of both methods, supporting their selection in future research on wave propagation problems and establishing future challenges and directions.

Keywords: Helmholtz equation, Wave scattering, Boundary Element Method, Physics-Informed Neural Networks, Computational methods, Numerical comparison

Introduction

Partial differential equations are central tools in mathematical modeling since they are employed to represent processes in many areas. Particularly, scattering is a physical phenomenon that arises in various fields such as acoustics, electromagnetism, geophysics, and medical imaging. Numerical methods such as the Finite Element Method (FEM) and the Boundary Element Method (BEM) are commonly used to model these problems, as they provide accurate approximations of the solution (Bermúdez et al.; Chandler-Wilde et al., 2007; 2012). However, these methods can become computationally expensive when applied to complex geometries or high-frequency regimes, as they require fine spatial discretization to maintain accuracy.

These limitations have motivated the exploration of alternative approaches, including machine learning-based methods, which aim to reduce the need for fine discretization and scale more efficiently with the dimensionality of the problem (Nganyu Tanyu et al., 2023). Machine learning has been used in the solution of forward partial differential equations. Physics-Informed Neural Networks (PINNs) incorporate the governing equations and available data into the training process (Lagaris et al.; Dissanayake and Phan-Thien; Raissi et al., 1998; 1994; 2019). PINNs have been used to approximate solutions of PDEs such as the Helmholtz equation (Escapil-Inchauspé and Ruz; Nair et al., 2023; 2024). However, their performance is often evaluated without comparisons to standard numerical methods or using limited benchmarks. A systematic review in fluid mechanics reports that 79% of machine learning solvers claiming superior performance over traditional methods do not use adequate baselines (McGreivy and Hakim, 2024).

Comparisons must be designed to avoid underestimating or overestimating the performance of each method. For example, this can be achieved by comparing solution times across methods with different accuracies, or by benchmarking against numerical methods that are not optimal for the specific problem. Therefore, a strong baseline is essential to assess the true capabilities and limitations of machine learning methods in solving PDEs. BEM can be a good reference for scattering problems governed by the Helmholtz equation. In contrast to domain-based methods such as FEM, BEM enforces the radiation condition at infinity and requires discretization only on the boundary of the scatterer. This reduces the dimensionality of the problem and eliminates the need for artificial boundary truncation. For these reasons, BEM is an appropriate baseline for evaluating the performance of PINNs.

An important aspect of machine learning methods such as PINNs is their ability to approximate solutions beyond the stabilized training domain (Bonfanti et al., 2024). This allows the model to estimate the solution at locations different from the original training points. However, unlike BEM, PINNs do not directly enforce the Sommerfeld radiation condition, which can result in inaccuracies in the far field, such as incorrect decay of the scattered wave at large distances. To address this issue, domain truncation techniques such as Perfectly Matched Layers (PML) or Absorbing Boundary Conditions (ABC) are typically employed (Berenger; Bermúdez et al.; Komatitsch and Martin, 1994; 2007; 2007). Therefore, a direct comparison between methods is necessary to evaluate their performance and to understand how each approach handles the physical constraints associated with scattering problems. This context leads to the following research question: *How can modern machine learning based approaches to solve partial differential equations be compared with traditional numerical methods?* In this work, we focus on the comparison between BEM and PINNs for solving acoustic wave scattering problems governed by the Helmholtz equation. To address this question, it is necessary to define benchmarks that incorporate accuracy, computational cost, and generalization capacity. Consistent evaluation criteria allow for the identification of the conditions under which each method is better suited to a specific problem.

This work presents a comparison between BEM and PINNs based on these criteria. Both approaches were applied to the problem of plane wave scattering by an obstacle, as introduced in Section 1. The solution was approximated using multiple configurations, and the relative error and computation time were measured. Additionally, the behavior of each method was evaluated both inside and outside the training domain, with emphasis on the asymptotic decay of the scattered field. Implementation details, the hyperparameter optimization of PINNs, and the comparison methodology are described in Section 2, while the results are analyzed in Section 3. The objective is to determine the range of conditions where each method performs best and to outline their respective advantages and limitations.

1. Wave Scattering and the Helmholtz Equation

We consider the classical problem of acoustic wave scattering by a single circular obstacle under a time-harmonic incident wave (Morse and Ingard, 1987). The problem involves solving the Helmholtz equation under boundary conditions that represent the interaction between the incident wave and the obstacle. The scattered wave field depends on the geometry of the obstacle and the properties of the surrounding medium, which in this case is unbounded. We illustrate this phenomenon in the case of a long cylinder in Figure 1.

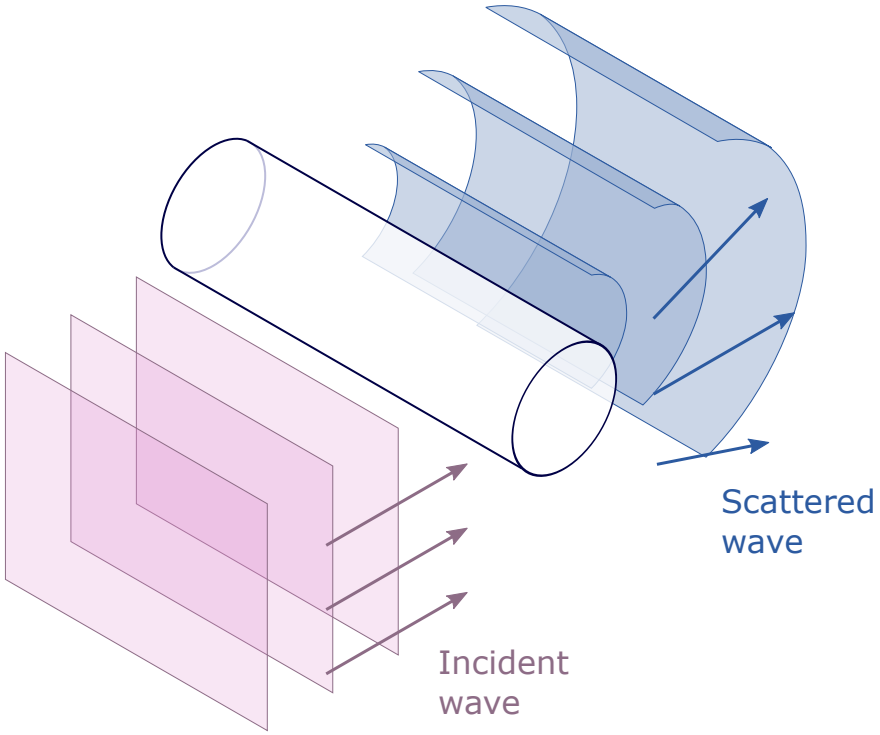

Figure 1. Schematic 3D representation of acoustic scattering by an obstacle. The incident plane wave propagates towards the obstacle, interacts with the circular interface, and produces the scattered field.

In numerical methods based on spatial discretization, the infinite domain is typically approximated by truncating the domain at a finite distance from the obstacle and imposing special boundary conditions to minimize artificial reflections. We begin by considering the Helmholtz equation:

$$\Delta u(\mathbf{x}) + k^2 u(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega_{\mathrm{P}}\,, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^2$ is the spatial coordinate, $u(\mathbf{x})$ is the unknown acoustic pressure field, Δ is the Laplacian operator, $k = \frac{\omega}{c}$ is the wavenumber, ω is the angular frequency, c is the speed of sound in the medium, and Ω_{P} is the physical domain where the wave propagates. It is important to note that the Helmholtz equation models a time-harmonic regime. The formulation assumes a steady-state oscillation at a fixed frequency ω, and the entire problem is posed in the frequency domain. Figure 2 show the representation of the scattering problem considering the domain truncation. The acoustic pressure field is obtained by considering the incident and the scattered pressures due to the interaction with the obstacle. Therefore:

$$u(\mathbf{x}) = u_{\mathrm{inc}}(\mathbf{x}) + u_{\mathrm{sct}}(\mathbf{x})\,.$$

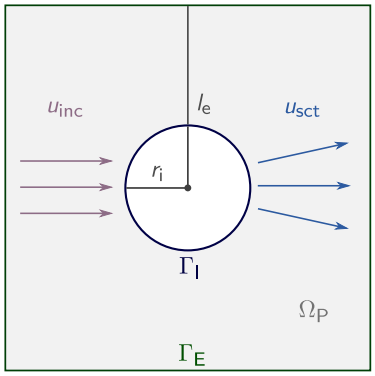

Figure 2. Schematic 2D representation of acoustic scattering by an obstacle. The scattered field u_{sct} is generated by an incident wave u_{inc} in a domain Ω_{P}, bounded internally by a circular interface Γ_{I} of radius r_{i}, and externally by a square boundary Γ_{E} of semi-length l_{e}. The incident plane wave propagates towards the obstacle, interacts with the circular interface, and produces the scattered field, which is evaluated within the computational domain Ω_{P}.

The incident wave is u_{inc}, which is typically a known function describing the incoming acoustic field (e.g., a plane wave) in polar coordinates:

$$u_{\text{inc}}(r,\theta) = Ae^{i\mathbf{k}r\cos\theta}, \tag{2}$$

where A is the amplitude and k is the wavevector. Using the Jacobi-Anger expansion, the variables of the right term can be separated and expressed in Equation 2 as:

$$Ae^{i\mathbf{k}r\cos\theta} = A\sum_{n=-\infty}^{\infty} i^n J_n(r)e^{in\theta}.$$

Here, J_n denotes the Bessel function of the first kind of order n. The scattered wave u_{sct} satisfies the Helmholtz equation in the exterior of the obstacle and is subject to a boundary condition at the obstacle surface. The specific form of the boundary condition on the surface (at $r = r_i$) depends on the type of obstacle. For a sound-hard obstacle, the appropriate condition is of Neumann type:

$$\nabla u(\mathbf{x})\cdot\mathbf{n} = 0, \quad \mathbf{x}\in\Gamma_{\text{I}}. \tag{3}$$

In this expression, $\mathbf{n}$ denotes the outward unit normal to the boundary Γ_{I}. To ensure a well-posed problem in an unbounded domain, the solution must also satisfy the Sommerfeld radiation condition (Schot, 1992):

$$\lim_{r\to\infty}\left(r^{1/2}\left(\frac{\partial u(r,\theta)}{\partial r} - iku(r,\theta)\right)\right) = 0, \tag{4}$$

where r is the radial distance from the center and $u(r,\theta)$ is the solution to the Helmholtz equation. This condition is mathematically required to ensure the uniqueness of the solution, and thus the well-posedness of the problem. In practice, the radiation condition is imposed numerically by truncating the domain with an artificial boundary Γ_{E} and applying an ABC or PML to approximate outgoing waves. The total acoustic field $u(\mathbf{x})$ must satisfy the Helmholtz equation (Equation 1) in the physical domain Ω_{P}, the Neumann boundary condition on the surface of the obstacle Γ_{I} (Equation 3), and the Sommerfeld radiation condition at infinity as $r\to\infty$ (Equation 4). Together, these three conditions fully define the mathematical formulation of the acoustic scattering problem in an unbounded domain.

2. Methods

This section presents two distinct numerical approaches for solving the exterior Helmholtz problem: the BEM and PINNs. We provide implementation details and discuss numerical strategies for both methods. For PINNs, we detail the procedure used for hyperparameter optimization in order to define a suitable network configuration. Finally, we outline the criteria used to compare the performance of both approaches in terms of accuracy and computational cost. All code used in this study is available in the public repository at github.com/oscar-rincon/comparative-bem-pinns.

2.1. Boundary Element Method Solution

To solve this problem using the BEM, it is necessary to formulate the integral equation for the exterior problem and to establish the boundary conditions for a circular obstacle under a plane incident wave (Katsikadelis; Kirkup, 2016; 2007). The following system of integral equations in the boundary is formulated:

$$\sum_{j=1}^{N} H_{ij}\, u^j \;=\; \sum_{j=1}^{N} G_{ij}\, u_n^j,$$

where the influence coefficients $\hat{H}_{ij}$ and G_{ij} are defined as

$$\hat{H}_{ij} = \int_{\Gamma_j} \frac{\partial v(p_i, q)}{\partial n_q}\, ds$$

and

$$G_{ij} = \int_{\Gamma_j} v(p_i, q)\, ds.$$

The $v(p_i, q)$ being the Green's function for the 2-D Helmholtz equation:

$$v \;=\; \frac{i}{4} H_0^{(1)}(kr),$$

and

$$\frac{\partial v}{\partial n_q} \;=\; -\frac{ik}{4} H_1^{(1)}(kr)\, \cos\phi.$$

The solution at evaluation points is computed as:

$$\frac{1}{2} u^i_{\text{pred}} = \sum_{j=1}^{N} G_{ij}\, u_n^j - \sum_{j=1}^{N} \hat{H}_{ij}\, u^j$$

The influence coefficients are estimated numerically for $i \neq j$, while for the case $i = j$, analytical corrections are applied due to the singularity of the kernel. We consider:

$$G_{jj} = \left(\int_{\Gamma_j} \frac{i}{4} H_0^{(1)}(kr) ds - \int_{\Gamma_j} \frac{1}{2\pi} \ln r ds \right) + \int_{\Gamma_j} \frac{1}{2\pi} \ln r ds, \quad \text{and} \quad H_{jj} = 0.$$

To approximate the boundary integrals in the computation of the influence coefficients G_{ij} and $\hat{H}_{ij}$, we employ Gaussian quadrature over each boundary element Γ_j. Each element is mapped to a

reference interval, $[-1, 1]$, and the integral is computed as a weighted sum of the integrand evaluated at specific quadrature points:

$$\int_{\Gamma_j} f(s)\, ds \;\approx\; \sum_{m=1}^{n_q} w_m\, f\big(s(\xi_m)\big)\, |J(\xi_m)|,$$

where ξ_m and w_m are the quadrature points and weights, respectively, and $J(\xi)$ is the Jacobian of the transformation from the reference element to the physical coordinates. First, the boundary was discretized into nodes and elements. Then, the solution was computed in the domain of interest, which consists of a uniform grid of evaluation points.

2.2. Physics-Informed Neural Networks Solution

To solve the boundary value problem described in Section 1, we use PINNs to approximate the scattered field u_{sct}. Since the domain $\Omega_{\text{P}} \subset \mathbb{R}^2$ is unbounded, an artificial boundary Γ_{E} is introduced to truncate the computational domain. On this boundary, ABC is imposed to approximate the Sommerfeld radiation condition (Equation 4):

$$\nabla u_{\text{sct}}(\mathbf{x}) \cdot \mathbf{n} + iku_{\text{sct}}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Gamma_{\text{E}}.$$

The full problem is given by the system:

$$\Delta u_{\text{sct}}(\mathbf{x}) + k^2 u_{\text{sct}}(\mathbf{x}) = 0, \qquad \mathbf{x} \in \Omega_{\text{P}}, \tag{5}$$

$$\nabla u_{\text{sct}}(\mathbf{x}) \cdot \mathbf{n} + \nabla u_{\text{inc}}(\mathbf{x}) \cdot \mathbf{n} = 0, \qquad \mathbf{x} \in \Gamma_{\text{I}}, \tag{6}$$

$$\nabla u_{\text{sct}}(\mathbf{x}) \cdot \mathbf{n} + iku_{\text{sct}}(\mathbf{x}) = 0, \qquad \mathbf{x} \in \Gamma_{\text{E}}. \tag{7}$$

Equations (5)–(7) define the physical constraints used to construct the loss function in the PINNs method. A neural network is constructed to approximate the solution $u_{\text{sct}} \colon \mathbb{R}^2 \to \mathbb{C}$, using two input variables (x, y), and two output variables corresponding to the real and imaginary parts of the scattered wave: $\Re(u_{\text{sct}}(x, y))$ and $\Im(u_{\text{sct}}(x, y))$. The network architecture is a multilayer perceptron implemented in PyTorch, with weights initialized using the Glorot uniform initializer to promote stable training. Training is accelerated on a GeForce RTX 4060 GPU. The loss function L_{T} is constructed by enforcing the residual of each governing condition at collocation points distributed throughout the domain and its boundaries:

$$L_{\text{T}} = L_{\Omega_{\text{P}}} + L_{\Gamma_{\text{I}}} + L_{\Gamma_{\text{E}}}.$$

The loss function is defined as the sum of squared residuals evaluated at collocation points located in the interior domain Ω_{P}, on the obstacle boundary Γ_{I}, and on the artificial boundary Γ_{E}:

$$L_{\Omega_{\text{P}}} = \frac{1}{N_{\Omega_{\text{P}}}} \sum_{i=1}^{N_{\Omega_{\text{P}}}} \left\| \mathcal{R}_{\Omega_{\text{P}}}(\mathbf{x}_{\text{P},i}) \right\|^2,$$

$$L_{\Gamma_{\text{I}}} = \frac{1}{N_{\Gamma_{\text{I}}}} \sum_{i=1}^{N_{\Gamma_{\text{I}}}} \left\| \mathcal{R}_{\Gamma_{\text{I}}}(\mathbf{x}_{\text{I},i}) \right\|^2,$$

$$L_{\Gamma_{\text{E}}} = \frac{1}{N_{\Gamma_{\text{E}}}} \sum_{i=1}^{N_{\Gamma_{\text{E}}}} \left\| \mathcal{R}_{\Gamma_{\text{E}}}(\mathbf{x}_{\text{E},i}) \right\|^2,$$

where the residuals are given by:

$$\begin{aligned}\mathcal{R}_{\Omega_\mathrm{P}}(\mathbf{x}) &:= \Delta u_\mathrm{sct}(\mathbf{x}) + k^2 u_\mathrm{sct}(\mathbf{x}),\\ \mathcal{R}_{\Gamma_\mathrm{I}}(\mathbf{x}) &:= \nabla u_\mathrm{sct}(\mathbf{x}) \cdot \mathbf{n} + \nabla u_\mathrm{inc}(\mathbf{x}) \cdot \mathbf{n},\\ \mathcal{R}_{\Gamma_\mathrm{E}}(\mathbf{x}) &:= \nabla u_\mathrm{sct}(\mathbf{x}) \cdot \mathbf{n} + iku_\mathrm{sct}(\mathbf{x}).\end{aligned}$$

Here, $\mathbf{x}_{\mathrm{P},i}$, $\mathbf{x}_{\mathrm{I},i}$, and $\mathbf{x}_{\mathrm{E},i}$ denote collocation points in the interior, on the obstacle boundary, and on the artificial boundary, respectively. The collocation points are distributed using the Latin hypercube sampling strategy.

Each residual enforces the corresponding governing equation or boundary condition in weak form, contributing to the total loss minimized during training. The residual $\mathcal{R}_{\Omega_\mathrm{P}}$ enforces the Helmholtz equation for the scattered field u_sct in the physical domain Ω_P, excluding the interior of the obstacle. On the obstacle boundary Γ_I, the Neumann condition models a sound-hard obstacle. On the artificial boundary Γ_E, the ABC approximates the behavior of an outgoing wave at finite distance and allows truncation of the computational domain without introducing significant reflections. The optimization of the neural network coefficients is performed in two stages: an initial phase using the Adam optimizer to minimize the loss via stochastic gradient descent, followed by a quasi-Newton phase using the L-BFGS algorithm to improve convergence with second-order information.

Both methods differ in their sampling strategies and overall workflows used to estimate the solution. Figure 3 illustrates the implementation of BEM and PINNs for the scattering problem. In the case of BEM, the integration points are uniformly distributed along the interior boundary of the physical domain. The estimation process in BEM occurs in a single computational cycle: it begins with the selection of the number of integration and evaluation points, followed by the assembly and solution of the boundary integral equations. Finally, the solution is evaluated at the specified evaluation points. In contrast, PINNs use randomly distributed training points throughout the physical domain, including the interior, the interior boundary, and the exterior boundary. The estimation process involves multiple optimization cycles, during which the neural network coefficients are iteratively updated to minimize the residuals of the governing equations and boundary conditions.

The training process involves several hyperparameters, including the learning rate α, the number of hidden layers L, the number of neurons per layer N, and the activation function σ. These are collectively described by the hyperparameter vector

$$\lambda = [\alpha, N, L, \sigma],$$

where $\lambda \in \Lambda$ and Λ denotes the Cartesian product of the predefined hyperparameter ranges. The Adam optimizer relies on gradient updates and is highly sensitive to the choice of learning rate, which directly influences stability and convergence speed during the initial training stage. The network depth L and width N control the representational capacity of the network to approximate more complex functions. However, larger values of L and N increase the number of optimization coefficients, which makes the training process more computationally demanding and can slow down convergence if not properly regularized. Finally, the activation function σ affects both the expressiveness of the network and the propagation of gradients, thereby influencing the efficiency of the optimization process. Since it governs the type of nonlinearities introduced at each layer, its choice is particularly important in wave problems such as the Helmholtz equation, where oscillatory behavior must be represented accu-

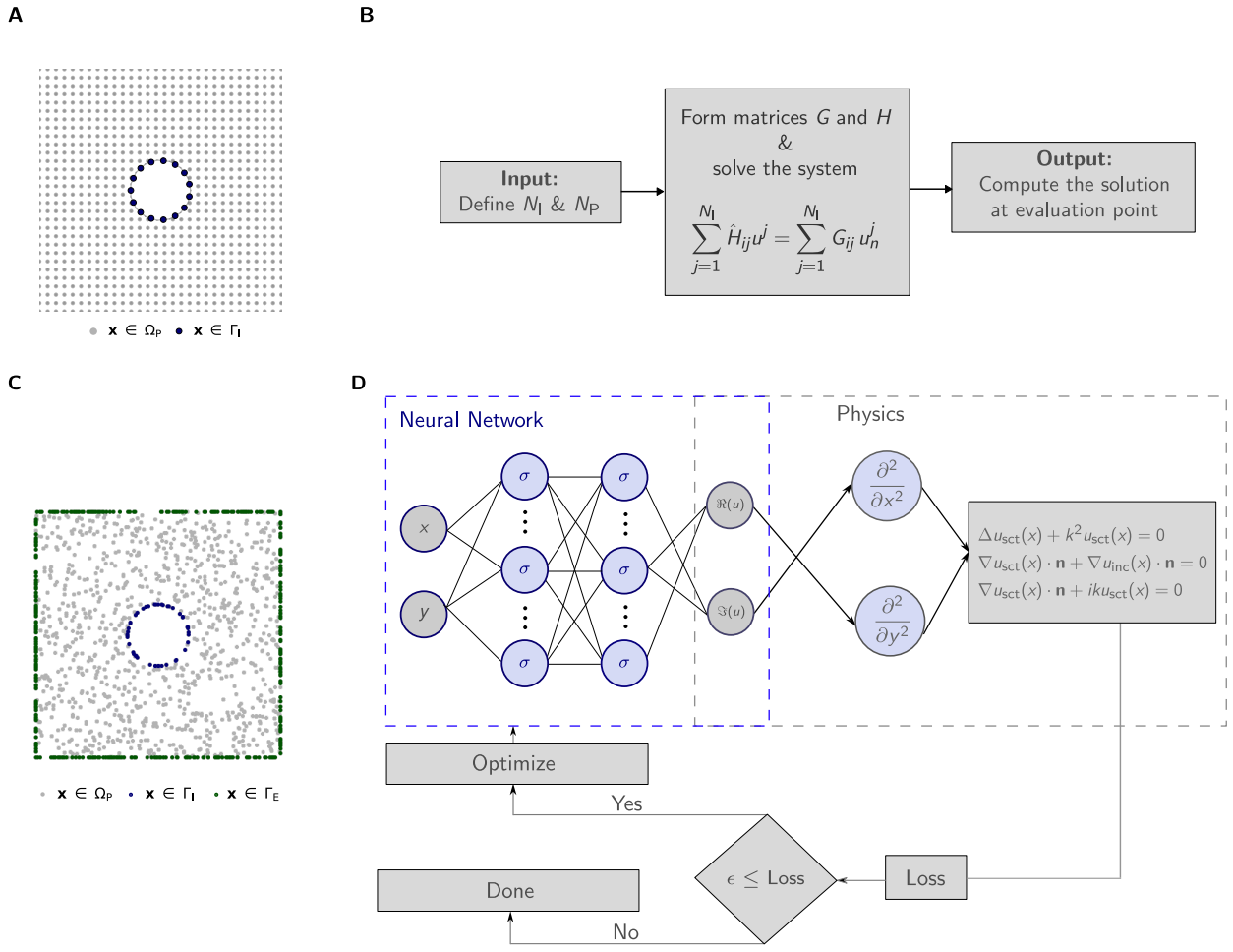

Figure 3. Illustration of the BEM and PINNs implementations for the scattering problem. (A) Distribution of boundary nodes and elements for BEM, along with the spatial arrangement of evaluation points within the domain. Gray dots represent evaluation points, while blue dots indicate integration points on the boundary Γ_I. (B) Sequential workflow of the BEM implementation. The inputs are the number of integration points for the boundary condition u_n on $\mathbf{x} \in \Gamma_I$ and the number of evaluation points N_P. The boundary integral equation is solved to obtain the scattered field, which is then evaluated at the points $\mathbf{x} \in \Omega_P$. (C) Sampling strategy in the physical domain for PINNs, including points in the interior Ω_P and on the boundaries Γ_I and Γ_E. Gray dots represent collocation points for the Helmholtz equation, while blue dots denote training points for the interior boundary condition on Γ_I. (D) Optimization process of the PINNs implementation for the Helmholtz equation. The neural network takes spatial coordinates (x, y) as input and outputs the real and imaginary parts of the scattered field u_{sct}. The loss function enforces the physics through the Helmholtz equation in Ω_P, the Neumann boundary condition on Γ_I, and the Sommerfeld radiation condition on Γ_E. The optimization loop updates the network parameters until the prescribed tolerance ϵ is reached.

rately.

The ranges for each hyperparameter were selected based on previous related studies in the PINNs literature, ensuring that they encompass configurations known to provide stable and accurate solutions for Helmholtz type problems (Escapil-Inchauspé and Ruz; Nair et al., 2023; 2024). To determine the optimal configuration, we employed Optuna, an automatic hyperparameter optimization framework

compatible with PyTorch. This procedure explored variations in the learning rate, network depth and width, activation functions, and boundary loss weights. The final configuration was selected as the one that minimized the objective loss, which is given by the relative error of the prediction.

2.3. Comparative Analysis

We conducted numerical experiments to compare the performance of the BEM and PINNs, focusing on solution accuracy and computational cost. We followed a similar approach to that used by Grossmann et al. (2024). Both methods were applied to solve the Helmholtz equation and systematic comparison using the ground truth solution for an obstacle described in Section 3.1. We evaluated the accuracy of BEM as a function of boundary discretization by varying the number of integration points. Smaller element sizes lead to higher accuracy but increase computation time. For PINNs, we controlled model capacity by adjusting the number of layers and neurons per layer, and assessed the corresponding effect on solution accuracy. The expressiveness of a neural network depends on both its depth and width.

We also estimated the computational time required for both methods. In the case of BEM, this included the time required to assemble and solve the boundary integral system, while for PINNs, we measured training and inference times. To compare computational efficiency under similar conditions, we selected the BEM configuration whose accuracy most closely matched that of the best-performing PINN model obtained through hyperparameter tuning.

We distinguish between solution time and evaluation time because the BEM and PINNs approximate the solution in different ways. In BEM, the solution is obtained by discretizing the boundary, and values in the domain are computed after the boundary representation through integral equations and its solution. In contrast, in PINNs, a neural network is trained using the collocation points distributed in the domain and on the boundaries. Once the network is trained, the evaluation at new points requires only a forward pass, which is faster than the trainning phase. For this reason, we report separately the training and evaluation times in PINNs.

To analyze the generalization ability of PINNs beyond the training domain and to assess how well each method handles the radiation condition, we evaluate the scattered field in a region that extends beyond the training domain. Specifically, we compute the numerical solution at a fixed propagation angle while varying the radial distance from the scatterer. The results obtained from both methods are compared against the analytical solution to evaluate their accuracy in the far field. We examine the radial decay of the wave amplitude and the relative error as functions of distance. Particular attention is given to the role of domain truncation in the PINN framework and its impact on the accuracy of the solution at large distances.

3. Results

In this section, we present the results obtained from the analytical solution, BEM, and PINNs for the 2D Helmholtz scattering problem. We begin by introducing the exact solution, which serves as the reference to assess numerical accuracy. Then, we describe the hyperparameter optimization process for PINNs and report the optimal configuration obtained. Using this configuration, we vary the number of integration points in BEM to achieve comparable accuracy, enabling a fair comparison of computational time. Finally, we evaluate the accuracy of both methods outside the training domain to assess their generalization capability. The results highlight the performance of each approach in estimating the scattered field within and beyond the training domain. All experiments were performed

on machines equipped with 10 CPU cores / 20 threads (13^{th} Gen Intel Core i9-13900H @ 3.40 GHz) and an NVIDIA GeForce RTX 4060 Laptop GPU.

3.1. Analytical Solution

The analytical solution to the 2D Helmholtz equation for the scattering of a plane wave by a circular obstacle can be expressed as a series expansion in cylindrical harmonics. The total field is written as the sum of the incident and scattered fields (Pao and Mow, 1973):

$$u_{\text{inc}}(r,\theta) = \sum_{n=-\infty}^{\infty} i^n J_n(kr) e^{in\theta}, \tag{8}$$

$$u_{\text{sct}}(r,\theta) = \sum_{n=-\infty}^{\infty} F_n H_n^{(1)}(kr) e^{in\theta}, \tag{9}$$

where J_n and $H_n^{(1)}$ denote the Bessel and Hankel functions of the first kind and order n [19]. The coefficients F_n are determined by enforcing the boundary condition on the surface of the obstacle. For a Neumann boundary condition, they are given by:

$$F_n = -\frac{i^n J_n'(kr_i)}{H_n^{(1)\prime}(kr_i)},$$

where r_i is the radius of the obstacle. The analytical expression in Equation 9 was used as the ground truth to validate the numerical results. Figure 4 shows the incident, scattered, and total displacement fields. The parameters used in all simulations were: wavenumber $k = 3$, obstacle radius $r_{\text{i}} = \frac{\pi}{4}$, and element length $l_{\text{e}} = \pi$, corresponding to a circular obstacle. This analytical solution was used to compute the relative error of the numerical methods evaluated in this study.

3.2. 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 the chosen hyperparameters. Figure 5 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 Figure 5 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 $\sigma = \text{Sine}$ and $\alpha = 10^{-2}$, yielding the lowest error in our study

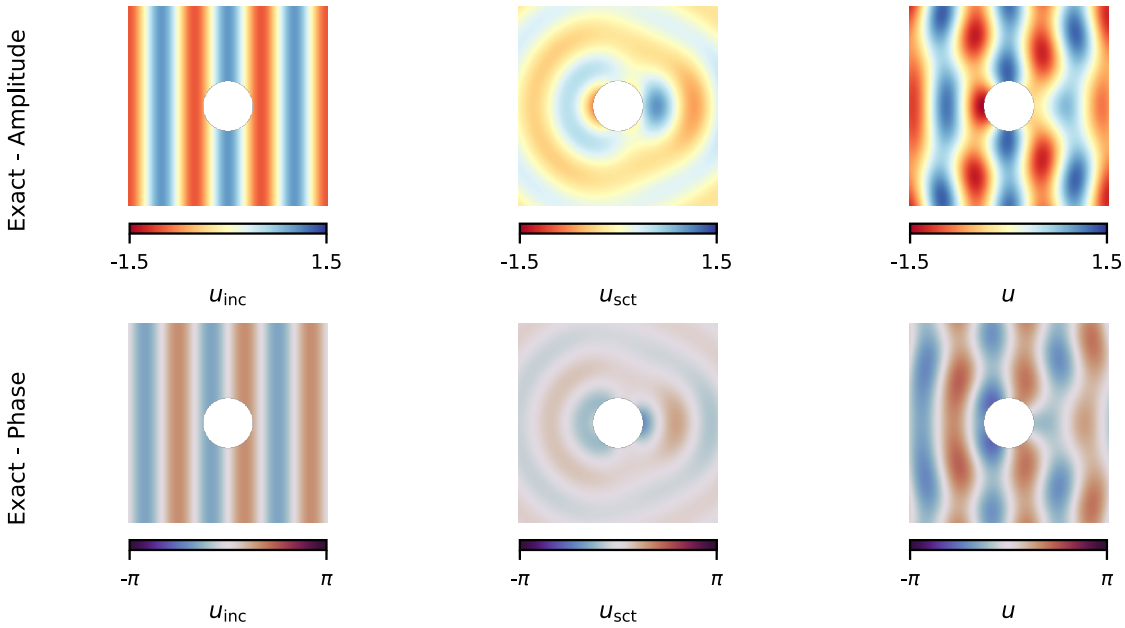

Figure 4. Analytical estimation of wave scattering. The first column shows the incident plane wave field (u_{inc}), the second column shows the corresponding scattered wave field (u_{sct}), and the third column shows their superposition (u). The top row presents the field amplitudes ($\Re(u(x, y))$), while the bottom row shows the corresponding phases ($\Im(u(x, y))$).

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.

3.3. Comparison of Accuracy and Computational Efficiency Between Methods

We conducted a comparative analysis of the BEM and PINNs for solving the two-dimensional Helmholtz equation, focusing on both solution accuracy and computational efficiency under varying parameters. To ensure reproducibility of the accuracy metrics, a fixed random seed was applied during model initialization and training. To account for runtime variability, each experiment was repeated 10 times and the average execution time was reported.

We tested BEM with the number of integration points ranging from 5 to 45 in increments of 5, and PINNs with network configurations varying from 1 to 3 hidden layers and 25 to 75 neurons per layer in increments of 25. These ranges are sufficient to capture the error reduction trends and to identify the achievable accuracy scales without making the computation excessively expensive. The remaining hyperparameters of the PINN were fixed according to the optimal configuration identified during the hyperparameter optimization stage.

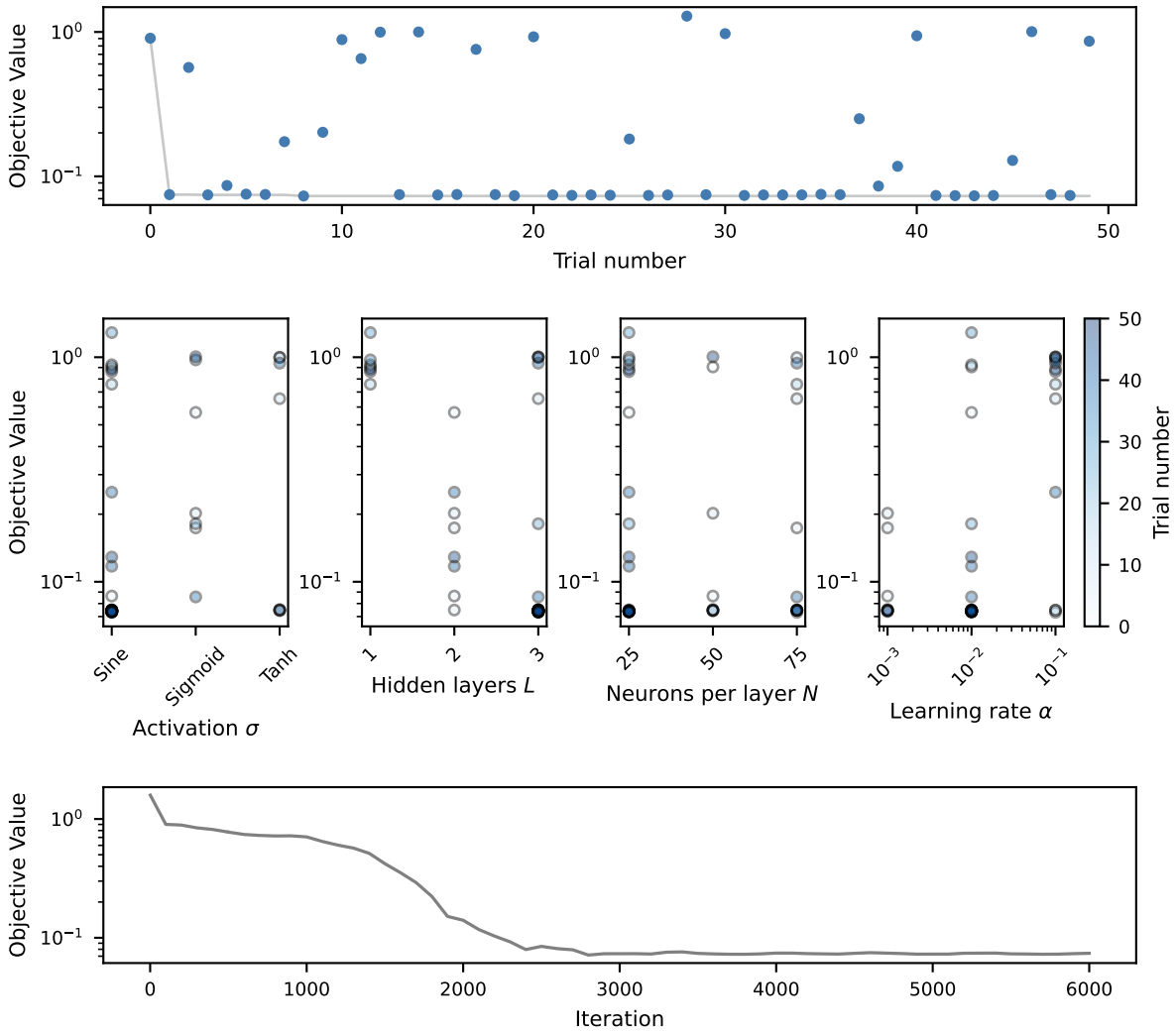

Figure 5. Hyperparameter optimization results obtained with Optuna. The top panel shows the optimization history, i.e., the evolution of the objective value across trials. The middle panels present slice plots, illustrating the relationship between each hyperparameter—the activation function σ, the number 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).

Figure 6 (top) illustrates the relation between relative error and computational time for BEM and PINNs. As expected, finer discretization in BEM improves accuracy but also increases computational cost. The computational time for BEM spans from the order of 10^0 to 10^1 s across the tested values of n. For PINNs, training times varied between 10^1 and 10^2 s, while evaluation times ranged on the order of 10^{-2} s.

For the computational time comparison, we selected the BEM configuration that achieved the closest accuracy to the optimal PINN solution. Specifically, the PINN with 3 layers and 25 neurons reached a relative error of 7.44×10^{-2}, while the closest BEM result was obtained with $n = 15$, yielding 7.25×10^{-2}. Figure 6 (bottom) shows the displacement field error for both methods using

these configurations. For BEM, the error tends to increase near the interior boundary of the obstacle, whereas for PINNs, errors are mainly concentrated in regions closer to the exterior boundary, where the domain was truncated.

In the selected case, BEM required a total solution time on the order of 10^1 s, making it approximately 10 times faster than the training time of the PINN, which was on the order of 10^2 s. This highlights the significant initial computational cost of PINNs due to the optimization process during training. However, once trained, the PINN model can be evaluated in about 10^{-1} s, which is around 100 times faster than the BEM solution time, making it well suited for applications requiring repeated simulations or real-time inference.

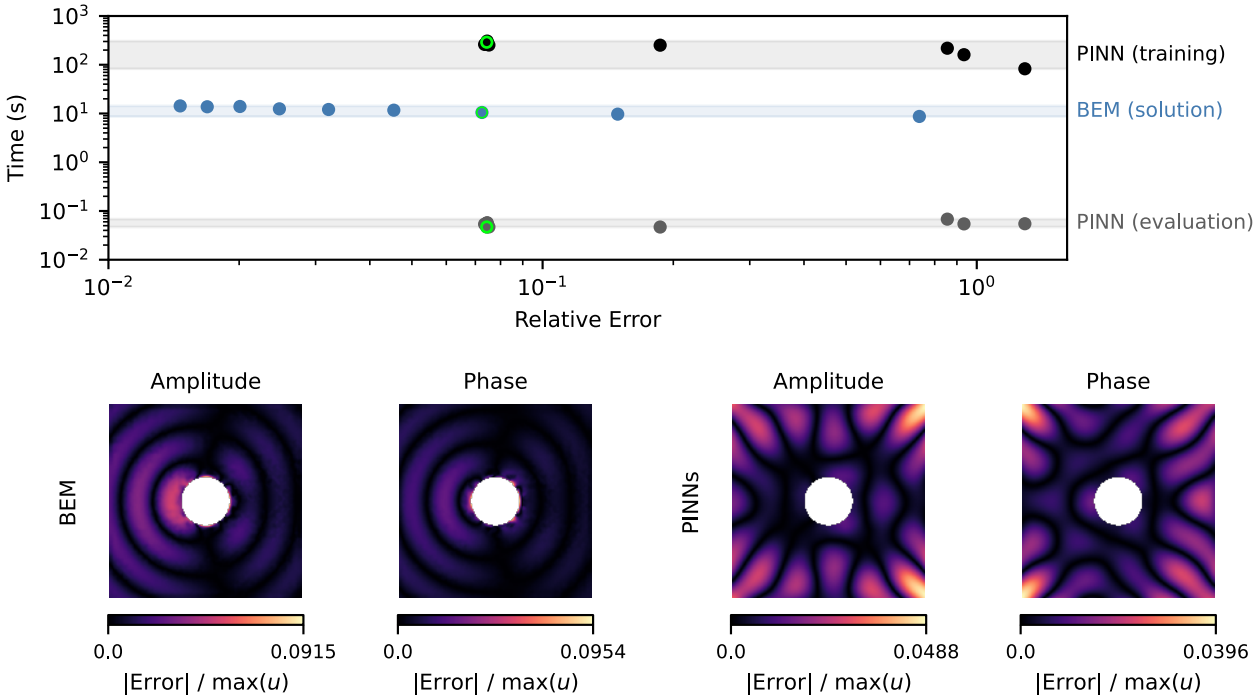

Figure 6. Performance evaluation of BEM and PINNs. (Top) Relative error versus solution time for BEM with varying numbers of integration points and for PINNs with different network architectures. Shaded horizontal bands indicate the ranges between the minimum and maximum times for each method. Labels at the right of the plot identify the three groups: PINN (training), BEM (solution), and PINN (evaluation). The green highlighted points correspond to the selected configurations, $(L = 3, N = 75)$ for the PINN and $n = 15$ for the BEM, which were used for the direct comparison of computational times, where L is the number of hidden layers, N the number of neurons per layer, and n the number of integration points. (Bottom) Spatial distributions of the relative error in amplitude and phase are shown for the selected BEM and PINN configurations.

3.4. Generalization Beyond the Training Domain

To evaluate the generalizability of both methods in an extended region of the domain, we computed the solution beyond the original training area. Figure 7 shows the amplitude and phase of the predicted fields, as well as the spatial distribution of the relative error. For the BEM, the relative error increased to 7.27×10^{-2} in the extended region. The error remained uniformly distributed, suggesting that the method maintains consistent accuracy even far from the obstacle. Additionally, an exponential decay

of the field amplitude was observed with increasing distance from the scatterer.

This behavior reflects the global nature of BEM, which inherently enforces the boundary conditions over the entire domain. In contrast, PINNs experienced a marked reduction in accuracy when evaluated outside the training domain. The relative error increased to 7.09, and the error distribution showed a clear concentration in the region outside the training domain.

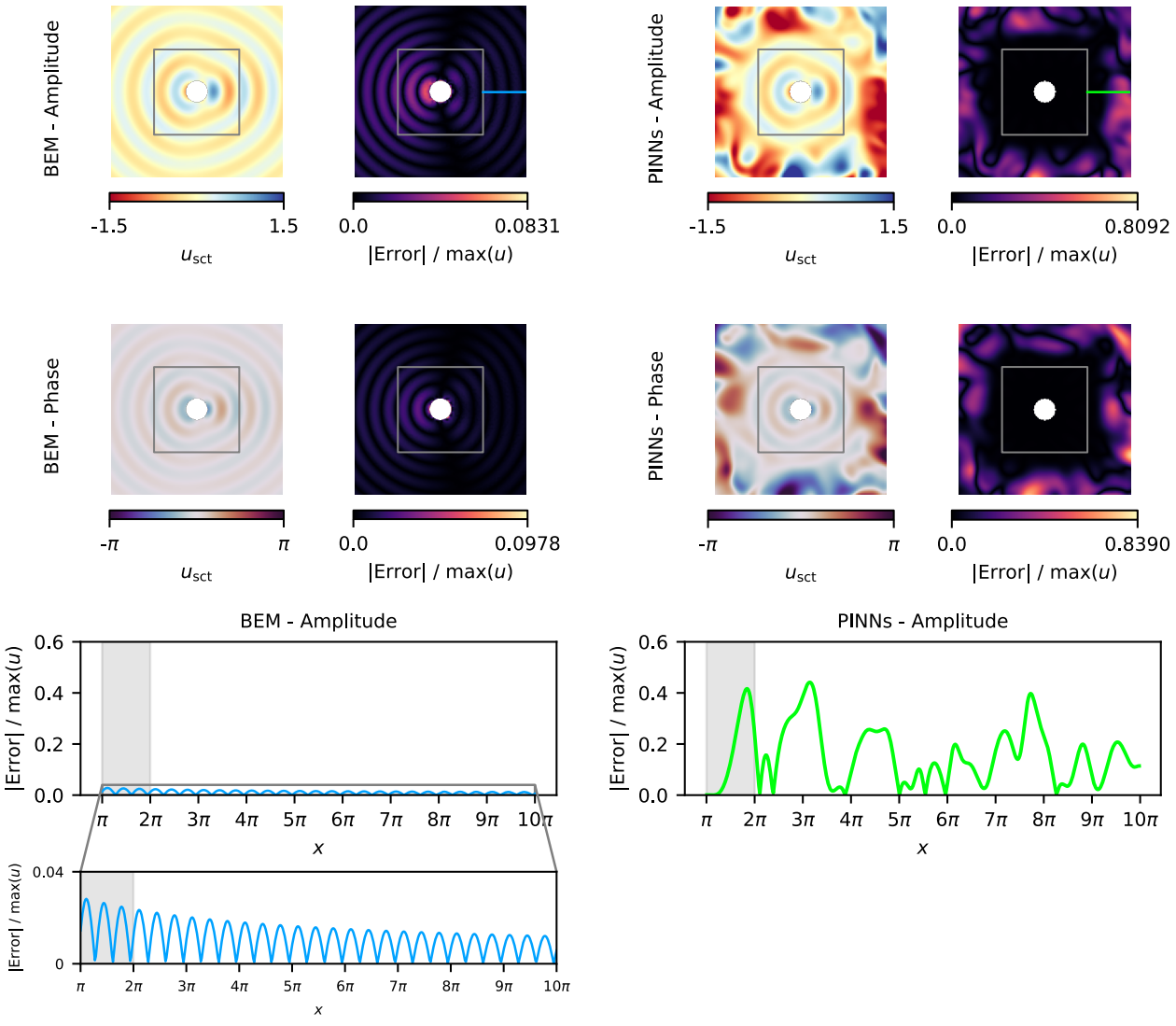

Figure 7. Scattered field computed by BEM and PINNs outside the training region. The top two rows display the amplitude (top) and phase (middle) of the scattered field resulting from an incident wave interacting with the obstacle. The square indicates the training domain, corresponding to $l_e = \pi$. For each method, the left panel shows the predicted field, while the right panel presents the relative error, normalized by the maximum of the reference solution. The bottom row shows one-dimensional slices of the relative amplitude error along the line $y = 0$, from $x = \pi$ to $x = 10\pi$. The shaded region corresponds to the training domain, while the unshaded region highlights the generalization behavior of both methods.

Discussion

A central objective of this work was to compare the performance and characteristics of traditional numerical methods and ML-based approaches for solving wave scattering problems using a fair baseline. In comparing BEM to PINNs, we addressed two common pitfalls identified in the literature on ML-based PDE solvers (McGreivy and Hakim, 2024).

First, we ensured that both methods were evaluated at comparable accuracy levels. The hyperparameters of the PINN model were tuned to achieve optimal performance, and subsequently, the number of integration points in the BEM solution was increased until the accuracy matched that of the PINN. Once comparable accuracy was achieved, we assessed the computational cost, thereby satisfying the requirement of comparing methods at either equal accuracy or equal computational cost.

Second, to avoid misleading conclusions due to the use of suboptimal numerical baselines, we employed BEM as the reference method, a highly efficient and well-established solver for wave scattering problems. BEM is particularly effective for problems governed by the Helmholtz equation, as it allows the exact enforcement of the Sommerfeld radiation condition through the use of fundamental solutions. Additionally, BEM reduces the dimensionality of the problem by requiring discretization only on the boundary, which is computationally advantageous in unbounded domains. This choice ensures that the traditional method is not only accurate but also representative of state-of-the-art efficiency for the class of problems considered.

By fulfilling these conditions, our comparative study provides a fair and informative assessment of the trade-offs between both solvers. While BEM demonstrates superior performance in terms of runtime for forward problems, PINNs offer a flexible framework that can be applied to a wide range of scientific and engineering problems, particularly due to the availability of open-source implementations. PINNs employ a mesh-free formulation in which the residuals of the governing equations are minimized at randomly sampled collocation points, allowing flexibility in spatial resolution. In contrast, classical methods such as BEM require mesh refinement to achieve higher accuracy. However, while PINNs are capable of solving forward problems, their training process is often computationally expensive due to long training times, as shown in Figure 6. Nonetheless, in scenarios that require repeated evaluations of the same PDE, such as in inverse problems, the relatively fast evaluation time of trained PINNs can be advantageous.

It is also important to acknowledge a limitation of this comparison regarding the measurement of computational cost. In this work, efficiency was assessed primarily through wall-clock time, which, although widely adopted in the literature, inherently depends on the characteristics of the hardware and system configuration used to perform the experiments. This makes the results sensitive to differences across machines and may reduce the generalizability of the reported results. A hardware independent approach would be relevant to complement wall-clock time with algorithmic complexity analysis or with estimates of the number of floating-point operations (FLOPs), thereby providing a clearer separation between the intrinsic efficiency of the methods and the capabilities of the underlying hardware. Nevertheless, reporting execution time remains a practical and frequently used benchmark for empirical comparisons, and thus it was employed in this study.

The hyperparameter tuning performed provides further insight into potentially suitable sets of hyperparameters for this problem. However, it is important to note that the performance of PINNs also depends on additional hyperparameters not considered in this work, such as the total number of

training iterations and the relative weights of the loss terms. Even with dedicated tuning procedures, the high dimensionality of the design space introduces variability and limits the reproducibility and comparability of results.

These factors imply that comparisons between BEM and PINNs should be interpreted in the context of specific problem setups and computational constraints. While BEM yields consistent solutions with relatively low sensitivity to parameters, PINNs require additional effort in model selection and optimization to achieve competitive accuracy. A possible alternative to further improve accuracy would be the incorporation of Fourier features, which have been found appropriate to mitigate spectral bias; however, the purpose of this work was to implement the method as originally proposed. An additional possible approach would be to adopt a multi-objective perspective, taking into account not only accuracy but also computational complexity.

An important distinction between BEM and PINNs lies in how each method handles the radiation condition, which governs the behavior of the scattered field at large distances. In BEM, this condition is satisfied exactly by construction through the use of the fundamental solution in the boundary integral formulation. This allows for an accurate computation of the far-field response without requiring additional assumptions. In contrast, PINNs do not explicitly enforce the radiation condition. Instead, they approximated it using domain truncation techniques, such as the ABC used in this work, which may not fully capture the correct asymptotic behavior of outgoing waves.

This limitation directly affects the generalization capacity of PINNs beyond the training region. In the current implementation, collocation points are sampled randomly and uniformly within the physical domain. While this strategy is straightforward, it does not prioritize regions where the radiation condition is most critical, particularly near the artificial boundary. As a result, PINNs may not accurately represent the far-field response, especially outside the domain used during training. This lack of generalization is a known limitation of neural network based models, which approximate solutions only within the region where they are trained. Alternative sampling strategies, such as boundary-focused distributions or adaptive collocation schemes, could help mitigate this issue and improve the enforcement of far-field behavior.

In summary, while BEM ensures accuracy through the direct satisfaction of physical conditions and dimensional reduction, PINNs offer flexible implementation through the configuration of their loss function, but require careful hyperparameter tuning and strategic sampling to achieve reliable results. Although BEM outperforms PINNs when considering solution time and accuracy, the strengths of each method are complementary, motivating the exploration of hybrid approaches that combine the robustness of boundary-based formulations with the adaptability of neural networks.

Conclusions

Recent advances in computational modeling have integrated neural networks as function approximators, supported by automatic differentiation and GPU acceleration. These machine learning techniques offer advantages such as mesh-free implementation, efficient inference, and flexibility to incorporate data into the solution process.

Our comparative analysis demonstrates that PINNs require significantly longer training times than BEM to achieve comparable accuracy. This computational cost limits their practical use for forward simulations. However, hybrid approaches that combine the strengths of both frameworks may offer

improved performance, particularly in inverse problems, where repeated model evaluations and parameter estimation are required.

In future work, we aim to explore a hybrid methodology that integrates the physical accuracy of BEM with the flexibility of neural networks, following recent developments by Nagy-Huber and Roth; Qu et al. (2024; 2024). Specifically, we propose to replace the differential operator in the PINN loss function with a boundary integral formulation. For the Helmholtz equation, this approach enables the direct incorporation of the Sommerfeld radiation condition into the training process, improving the model's ability to generalize beyond the training domain and ensuring accurate predictions in unbounded regions. Moreover, this formulation is particularly advantageous in inverse problems, where enforcing correct far-field behavior is essential for achieving stable and physically consistent reconstructions.

References

[1] A. Bermúdez, L. Hervella-Nieto, A. Prieto, and R. Rodrı´guez. An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems. *Journal of Computational Physics*, 223(2):469–488, May 2007. ISSN 0021-9991. doi: 10.1016/j.jcp.2006.09.018. URL https://www.sciencedirect.com/science/article/pii/S0021999106004487.

[2] Simon N. Chandler-Wilde, Ivan G. Graham, Stephen Langdon, and Euan A. Spence. Numerical-asymptotic boundary integral methods in high-frequency acoustic scattering. *Acta Numerica*, 21:89–305, May 2012. ISSN 1474-0508, 0962-4929. doi: 10.1017/S0962492912000037. URL https://www.cambridge.org/core/journals/acta-numerica/article/abs/numericalasymptotic-boundary-integral-methods-in-highfrequency-acoustic-scattering/5FDA12EF93DD25E89AF7B57A01F8CAEE.

[3] Derick Nganyu Tanyu, Jianfeng Ning, Tom Freudenberg, Nick Heilenkötter, Andreas Rademacher, Uwe Iben, and Peter Maass. Deep learning methods for partial differential equations and related parameter identification problems. *Inverse Problems*, 39(10):103001, August 2023. ISSN 0266-5611. doi: 10.1088/1361-6420/ace9d4. URL https://dx.doi.org/10.1088/1361-6420/ace9d4. Publisher: IOP Publishing.

[4] I.E. Lagaris, A. Likas, and D.I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, September 1998. ISSN 1941-0093. doi: 10.1109/72.712178. URL https://ieeexplore.ieee.org/document/712178. Conference Name: IEEE Transactions on Neural Networks.

[5] M. W. M. G. Dissanayake and N. Phan-Thien. Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, 10(3):195–201, 1994. ISSN 1099-0887. doi: 10.1002/cnm.1640100303. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.1640100303. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.1640100303.

[6] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019. doi: 10.1016/J.JCP.2018.10.045. Publisher: Academic Press.

[7] 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. URL https://www.sciencedirect.com/science/article/pii/S0925231223009499.

[8] 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. URL https://doi.org/10.1007/s00366-024-02038-3.

[9] Nick McGreivy and Ammar Hakim. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations, July 2024. URL http://arxiv.org/abs/2407.07218. arXiv:2407.07218 [physics].

[10] Andrea Bonfanti, Roberto Santana, Marco Ellero, and Babak Gholami. On the generalization of PINNs outside the training domain and the hyperparameters influencing it. *Neural Computing and Applications*, 36(36):22677–22696, December 2024. ISSN 1433-3058. doi: 10.1007/s00521-024-10178-2. URL https://doi.org/10.1007/s00521-024-10178-2.

[11] Jean-Pierre Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185–200, October 1994. ISSN 0021-9991. doi: 10.1006/jcph.1994.1159. URL https://www.sciencedirect.com/science/article/pii/S0021999184711594.

[12] Dimitri Komatitsch and Roland Martin. An unsplit convolutional Perfectly Matched Layer improved at grazing incidence for the seismic wave equation. *Geophysics*, 72, September 2007. doi: 10.1190/1.2757586.

[13] Philip McCord Morse and K Uno Ingard. *Theoretical Acoustics*. Princeton University Press, 1987. ISBN 978-0-691-02401-1.

[14] Steven H Schot. Eighty years of Sommerfeld's radiation condition. *Historia Mathematica*, 19 (4):385–401, November 1992. ISSN 0315-0860. doi: 10.1016/0315-0860(92)90004-U. URL https://www.sciencedirect.com/science/article/pii/031508609290004U.

[15] John T. Katsikadelis. *The Boundary Element Method for Engineers and Scientists, Second Edition: Theory and Applications*. Academic Press, London, 2nd edition edition, July 2016. ISBN 978-0-12-804493-3.

[16] Stephen Kirkup. *The Boundary Element Method in Acoustics*, volume 8. Integrated Sound Software, January 2007. ISBN 978-0-9534031-0-3. Journal Abbreviation: Journal of Computational Acoustics Publication Title: Journal of Computational Acoustics.

[17] Tamara G Grossmann, Urszula Julia Komorowska, Jonas Latz, and Carola-Bibiane Schönlieb. Can physics-informed neural networks beat the finite element method? *IMA Journal of Applied Mathematics*, 89(1):143–174, January 2024. ISSN 0272-4960. doi: 10.1093/imamat/hxae011. URL https://doi.org/10.1093/imamat/hxae011.

[18] Yih-Hsing Pao and Chao-Chow Mow. *Diffraction of Elastic Waves and Dynamic Stress Concentrations*. A Rand Corporation Research Study. Russak, New York, 1973. ISBN 978-0-85274-244-0 978-0-8448-0155-1.

[19] DLMF. *NIST Digital Library of Mathematical Functions*. https://dlmf.nist.gov/, Release 1.2.4 of 2025-03-15, 2025. URL https://dlmf.nist.gov/. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.

[20] Monika Nagy-Huber and Volker Roth. Physics-informed boundary integral networks (PIBI-Nets): A data-driven approach for solving partial differential equations. *Journal of Computational Science*, 81:102355, September 2024. ISSN 1877-7503. doi: 10.1016/j.jocs.2024.102355. URL https://www.sciencedirect.com/science/article/pii/S1877750324001480.

[21] Wenzhen Qu, Yan Gu, Shengdong Zhao, Fajie Wang, and Ji Lin. Boundary integrated neural networks and code for acoustic radiation and scattering. *International Journal of Mechanical System Dynamics*, 4(2):131–141, 2024. ISSN 2767-1402. doi: 10.1002/msd2.12109. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/msd2.12109. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/msd2.12109.