

Article

Not peer-reviewed version

Physics-Informed Neural Networks for Excited Liquid Sloshing with Beating Response in a Rectangular Tank

[Zhiqiang Luo](#)*

Posted Date: 24 March 2026

doi: 10.20944/preprints202603.1849.v1

Keywords: physics-informed neural networks; potential flow; excited sloshing; beating response; Fourier feature embedding; spectral bias; time-domain decomposition; three-dimensional sloshing



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Physics-Informed Neural Networks for Excited Liquid Sloshing with Beating Response in a Rectangular Tank

Zhiqiang Luo

School of Media Engineering, Communication University of Zhejiang, Hangzhou 310018, China; zql1009@126.com

Abstract

This paper applies physics-informed neural networks (PINNs) to laterally excited liquid sloshing in a two-dimensional rectangular tank, where near-resonant forcing ($\omega_e/\omega_1 = 0.9$) produces a multi-frequency beating response with a period of approximately $10 T_1$. Linearised potential flow theory governs the problem; the network learns the velocity potential $\varphi(x, z, t)$ while the free-surface elevation η is injected analytically. Two training obstacles specific to forced sloshing are analysed. First, a *zero-resolution trap* arises because the trivial solution $\hat{\varphi} = 0$ satisfies all equations except the free-surface conditions, whose residuals are roughly 10^{-4} times smaller than the Laplace residual; characteristic-scale normalisation combined with loss weighting ($\lambda_D = \lambda_K = 100$) breaks this trap. Second, *spectral bias* prevents standard MLPs from resolving the three co-existing frequencies ($\omega_1, \omega_e, \Delta\omega$); a Fourier time embedding that augments the input from 3 to 9 dimensions overcomes this limitation. Two additional techniques further reduce errors: a hard wall boundary condition enforced exactly via a $\cos(\pi x/B)$ spatial embedding, which eliminates wall collocation points; and a gradient-enhanced Laplace regulariser ($\|\nabla(\nabla^2 \hat{\varphi})\|^2$) that constrains velocity smoothness through third-order automatic differentiation. An ablation study shows that these four techniques progressively reduce the horizontal velocity error from $\varepsilon_u = 12.46\%$ to 0.84% . Results are validated against a viscous finite-difference benchmark. Over one beating cycle the errors are $\varepsilon_\eta = 0.15\%$, $\varepsilon_u = 0.84\%$, and $\varepsilon_w = 1.65\%$. A frequency parameter study across $\omega_e/\omega_1 = 0.5\text{--}1.1$ gives $\varepsilon_\eta < 0.25\%$ and $\varepsilon_u < 2.3\%$ for all near-resonance cases. For long-time simulation, a time-domain decomposition strategy with transfer learning partitions the domain into one-beat windows; extending to five beating cycles ($50 T_1$) yields $\varepsilon_u = 3.43\%$ and $\varepsilon_\eta = 0.30\%$ with no monotonic error accumulation across windows. The methodology is then extended to a three-dimensional rectangular tank ($B \times W \times H$) with bi-directional lateral excitation. The 3-D formulation introduces the y -dimension into the Laplace equation ($\nabla^2 \varphi = \varphi_{xx} + \varphi_{yy} + \varphi_{zz} = 0$), adds transverse wall boundary conditions ($\partial\varphi/\partial y = 0$) enforced exactly via a $\cos(\pi y/W)$ embedding, and extends the Fourier time embedding from 9 to 16 dimensions to accommodate six physical frequencies. The bi-directional excitation excites both $(m, 0)$ and $(0, n)$ modal families, producing a genuinely three-dimensional beating response. Results demonstrate that the proposed techniques transfer effectively to 3-D, with errors $\varepsilon_\eta = 0.24\%$, $\varepsilon_u = 1.31\%$, $\varepsilon_v = 1.78\%$, and $\varepsilon_w = 2.32\%$ over one beating cycle (2,499 s training time).

Keywords: physics-informed neural networks; potential flow; excited sloshing; beating response; Fourier feature embedding; spectral bias; time-domain decomposition; three-dimensional sloshing

1. Introduction

Liquid sloshing in partially filled containers arises in naval architecture, aerospace engineering, and civil engineering, where it affects the structural integrity of LNG carriers, spacecraft fuel tanks, and seismically loaded storage vessels [1,2]. Conventional computational approaches—finite-element methods [3], finite-difference methods [4], and boundary element methods—require mesh generation and time integration; for long transient simulations spanning tens of natural periods, these costs

become considerable. Reduced-order models alleviate part of the burden but are typically restricted to a pre-selected set of modes.

Physics-informed neural networks (PINNs), introduced by Raissi et al. [5], embed the governing PDEs directly into the training loss, so the network is trained without labeled solution data. Reviews by Cai et al. [6] and Cuomo et al. [7] cover applications from Navier–Stokes flows to heat transfer; the open-source library DeepXDE [8] has further broadened accessibility.

Training PINNs for oscillatory problems is difficult in practice. Wang et al. [9] showed that imbalanced multi-component losses cause gradient pathologies in which the PDE residual dominates and suppresses boundary and initial condition learning; they proposed gradient-based adaptive weighting as a remedy. Jagtap et al. [10] addressed slow convergence through adaptive activation functions. A more fundamental obstacle is spectral bias—the preference of standard fully connected networks for low-frequency components of the target function, first analyzed by Rahaman et al. [11]. For oscillatory PDEs covering many cycles this can lead to severe temporal under-resolution. Tancik et al. [12] showed that mapping inputs through sinusoidal Fourier features enables networks to learn high-frequency functions; Wang et al. [13] extended the analysis to PINNs via Neural Tangent Kernel theory. Wang et al. [14] further proposed PirateNets with causal training to enforce temporal causality and prevent PINNs from using future information to predict past states, a common pathology in time-dependent problems.

For free surface and water wave problems, several strands of work have appeared in recent years. Sheikholeslami et al. [15] examined soft and hard boundary enforcement for linear Laplace-based waves, achieving velocity errors below 3% with a periodic hard boundary layer; their trial-function approach for the bottom boundary condition, however, restricted the solution space and increased errors by up to 15 times. Li et al. [16] treated fully nonlinear wave propagation using a quasi- σ coordinate transformation that maps the time-dependent free surface to a fixed domain, with a two-stage Adam/L-BFGS optimizer. Chen et al. [17] reconstructed rotational flow beneath periodic waves from sparse gauge data using WaveNets. More recently, Ehlers et al. [18] applied PINNs to the fully nonlinear potential flow equations for phase-resolved ocean wave assimilation and prediction, inferring the velocity potential throughout the fluid volume from surface elevation measurements alone.

Closer to the sloshing problem, Wessels et al. [19] proposed the Neural Particle Method, a Lagrangian PINN that tracks particle paths to handle the moving free surface; they simulated both small- and large-amplitude sloshing in a container and the dam-break problem, with good agreement with experimental data. Huang et al. [20] developed a boundary-and-initial-conditions-free PINN (bif-PINN) in Lagrangian coordinates that hardwires all boundary and initial conditions into the formulation via nonlinear residual connections; their approach solved the small- and large-amplitude sloshing cases accurately, whereas the original PINN failed entirely for all free-surface configurations. Shao et al. [21] combined PINNs with an improved neural particle method for violent sloshing simulation, validated against SPH results.

Despite this progress, all the above free-surface PINN studies address either periodic propagating waves over a few wavelengths or free sloshing (unforced) over a handful of cycles. The forced sloshing problem—where an external excitation near resonance produces a multi-frequency beating response spanning tens of natural periods—poses additional challenges that have not been addressed: the simulation must cover an order of magnitude more periods than a single natural cycle; the weak forcing creates free surface residuals orders of magnitude smaller than the Laplace residual, opening a zero-solution trap; and the multi-frequency temporal content exceeds the effective bandwidth of a standard MLP. The present work tackles these challenges for laterally forced sloshing at $\omega_e = 0.9 \omega_1$, where beating produces a response envelope over approximately $10 T_1$. Four configurations (C–F) progressively address the above difficulties, reducing ε_u from 12.46% to 0.84%. The methodology is then extended to a three-dimensional rectangular tank with bi-directional excitation, demonstrating

that the same techniques—Fourier embedding, hard wall boundary conditions, and gradient-enhanced regularisation—transfer effectively to the higher-dimensional setting.

2. Mathematical Formulation

2.1. Problem Setup

Consider a two-dimensional rectangular tank of length L and still-water depth h . The fluid is inviscid, incompressible, and irrotational, so the flow is described by a velocity potential $\varphi(x, z, t)$. The parameters are:

$$L = 2.0 \text{ m}, \quad h = 1.0 \text{ m}, \quad n = 1 \text{ (first sloshing mode)}. \quad (1)$$

The wave number and natural frequency are:

$$k_1 = \frac{\pi}{L} = 1.5708 \text{ rad/m}, \quad (2)$$

$$\omega_1 = \sqrt{g k_1 \tanh(k_1 h)} = 3.7594 \text{ rad/s}, \quad (3)$$

with $g = 9.81 \text{ m/s}^2$. The natural period is $T_1 = 2\pi/\omega_1 = 1.6713 \text{ s}$.

The tank is subjected to a lateral harmonic acceleration:

$$a(t) = a_0 \sin(\omega_e t), \quad a_0 = 0.002 \text{ m/s}^2, \quad \omega_e = 0.9 \omega_1 = 3.3834 \text{ rad/s}. \quad (4)$$

The excitation frequency is chosen near the first natural frequency to induce a beating response with period:

$$T_{\text{beat}} = \frac{2\pi}{|\omega_1 - \omega_e|} = 16.71 \text{ s} \approx 10 T_1. \quad (5)$$

The simulation domain spans one full beating cycle: $t \in [0, T_{\text{beat}}]$.

2.2. Governing Equation

In the non-inertial reference frame attached to the tank, the Euler equations for an inviscid, incompressible fluid read:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\frac{1}{\rho} \nabla p - g \hat{\mathbf{z}} - a(t) \hat{\mathbf{x}}, \quad (6)$$

where $\mathbf{v} = (u, w)$ is the velocity vector, p is the pressure, ρ is the (constant) fluid density, g is gravitational acceleration, and $a(t) \hat{\mathbf{x}}$ is the pseudo-force due to the non-inertial frame. For irrotational flow ($\nabla \times \mathbf{v} = \mathbf{0}$), a velocity potential φ exists such that $\mathbf{v} = \nabla \varphi$. The incompressibility constraint $\nabla \cdot \mathbf{v} = 0$ then yields the Laplace equation:

$$\nabla^2 \varphi = \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial z^2} = 0, \quad 0 < x < L, \quad -h < z < 0. \quad (7)$$

Substituting $\mathbf{v} = \nabla \varphi$ into the Euler equation (6) and using the vector identity $(\mathbf{v} \cdot \nabla) \mathbf{v} = \nabla(|\mathbf{v}|^2/2)$ (valid for irrotational flow), one obtains the unsteady Bernoulli equation:

$$\frac{\partial \varphi}{\partial t} + \frac{1}{2} |\nabla \varphi|^2 + \frac{p}{\rho} + g z + a(t) x = C(t), \quad (8)$$

where $C(t)$ is a spatially uniform function absorbed into φ by redefinition. This Bernoulli equation is the starting point for deriving the dynamic free surface boundary condition.

The velocity field is recovered from $\mathbf{v} = \nabla \varphi = (u, w)$.

2.3. Boundary Conditions

KBBC (impermeable bottom):

$$\frac{\partial \varphi}{\partial z} = 0 \quad \text{at } z = -h. \quad (9)$$

KWBC (impermeable walls):

$$\frac{\partial \varphi}{\partial x} = 0 \quad \text{at } x = 0 \text{ and } x = L. \quad (10)$$

KFSBC (kinematic free surface): The free surface is defined implicitly by $F(x, z, t) = z - \eta(x, t) = 0$. The kinematic condition requires that the free surface is a material surface, i.e., $DF/Dt = 0$:

$$\frac{DF}{Dt} = \frac{\partial F}{\partial t} + u \frac{\partial F}{\partial x} + w \frac{\partial F}{\partial z} = -\frac{\partial \eta}{\partial t} - \frac{\partial \varphi}{\partial x} \frac{\partial \eta}{\partial x} + \frac{\partial \varphi}{\partial z} = 0 \quad \text{at } z = \eta(x, t), \quad (11)$$

which yields the fully nonlinear KFSBC: $\partial \eta / \partial t + (\partial \varphi / \partial x)(\partial \eta / \partial x) = \partial \varphi / \partial z$ at $z = \eta$. Under the small-amplitude assumption ($\eta/h = \mathcal{O}(\epsilon)$, $\epsilon \ll 1$), we perform a Taylor expansion about the mean free surface $z = 0$:

$$\left. \frac{\partial \varphi}{\partial z} \right|_{z=\eta} = \left. \frac{\partial \varphi}{\partial z} \right|_{z=0} + \eta \left. \frac{\partial^2 \varphi}{\partial z^2} \right|_{z=0} + \mathcal{O}(\epsilon^2). \quad (12)$$

Retaining only $\mathcal{O}(\epsilon)$ terms and noting that $(\partial \varphi / \partial x)(\partial \eta / \partial x) = \mathcal{O}(\epsilon^2)$, the linearized KFSBC becomes:

$$\frac{\partial \varphi}{\partial z} = \frac{\partial \eta}{\partial t} \quad \text{at } z = 0. \quad (13)$$

DFSBC (dynamic free surface): Setting $p = p_{\text{atm}}$ (atmospheric pressure, taken as the reference) at $z = \eta$ in the Bernoulli equation (8):

$$\left. \frac{\partial \varphi}{\partial t} \right|_{z=\eta} + \frac{1}{2} \left[\left(\left. \frac{\partial \varphi}{\partial x} \right|_{z=\eta} \right)^2 + \left(\left. \frac{\partial \varphi}{\partial z} \right|_{z=\eta} \right)^2 \right] + g \eta + a(t) x = 0. \quad (14)$$

Again applying the Taylor expansion about $z = 0$:

$$\left. \frac{\partial \varphi}{\partial t} \right|_{z=\eta} = \left. \frac{\partial \varphi}{\partial t} \right|_{z=0} + \eta \left. \frac{\partial^2 \varphi}{\partial t \partial z} \right|_{z=0} + \mathcal{O}(\epsilon^2), \quad (15)$$

and noting that $|\nabla \varphi|^2 = \mathcal{O}(\epsilon^2)$, the linearized DFSBC reads:

$$\frac{\partial \varphi}{\partial t} + g \eta + a(t) x = 0 \quad \text{at } z = 0, \quad (16)$$

where $a(t) x$ is the pseudo-pressure term arising from the non-inertial reference frame.

Combined free surface condition. Eliminating η from the linearized KFSBC (13) and DFSBC (16) by differentiating the DFSBC with respect to t and substituting $\partial \eta / \partial t$ from the KFSBC yields:

$$\frac{\partial^2 \varphi}{\partial t^2} + g \frac{\partial \varphi}{\partial z} + \dot{a}(t) x = 0 \quad \text{at } z = 0, \quad (17)$$

where $\dot{a}(t) = a_0 \omega_e \cos(\omega_e t)$. This combined condition, while not used directly in the PINN loss, provides physical insight: the free surface acts as a dispersive wave guide with the forcing $\dot{a}(t) x$ injecting energy into the system.

2.4. Initial Condition

Since the tank starts from rest, $\varphi(x, z, 0) = 0$ and $\partial\varphi/\partial t(x, z, 0) = 0$. No terminal condition is imposed because φ does not vanish at $t = T_{\text{beat}}$; the two initial conditions together serve as temporal anchors.

3. PINN Methodology

3.1. Network Architecture

The neural network approximation $\hat{\varphi}_\theta(x, z, t)$ is a fully connected feed-forward network (multi-layer perceptron) defined by the recursive composition:

$$\mathbf{h}^{(0)} = \mathbf{x}_{\text{input}}, \quad \mathbf{h}^{(\ell)} = \tanh(\mathbf{W}^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}), \quad \hat{\varphi} = \mathbf{W}^{(L+1)}\mathbf{h}^{(L)} + b^{(L+1)}, \quad (18)$$

where $\mathbf{W}^{(\ell)} \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ and $\mathbf{b}^{(\ell)} \in \mathbb{R}^{n_\ell}$ are the weight matrix and bias vector for layer ℓ . The collection of all trainable parameters is $\theta = \{\mathbf{W}^{(\ell)}, \mathbf{b}^{(\ell)}\}_{\ell=1}^{L+1}$. The network uses $L = 16$ hidden layers with $n_\ell = 64$ neurons, Xavier normal initialization, and a single scalar output. Velocities follow from automatic differentiation:

$$u = \frac{\partial \hat{\varphi}}{\partial x}, \quad w = \frac{\partial \hat{\varphi}}{\partial z}. \quad (19)$$

In the enhanced configuration (Config E), the input is augmented with Fourier time features, increasing the input dimension from 3 to 9 (see Section 3.6). The configuration is summarized in Table 1.

Table 1. Neural network architecture.

Parameter	Value
Input	3 (x, z, t) or 9 (with Fourier features)
Output	1 ($\hat{\varphi}$)
Hidden layers	16
Neurons/layer	64
Activation	tanh
Initialization	Xavier normal
Parameters	$\sim 62,700$ (3-input) / $\sim 63,100$ (9-input)

3.2. Loss Function and Automatic Differentiation

All required derivatives of the network output $\hat{\varphi} = f_\theta(\gamma(\mathbf{x}))$ are computed via automatic differentiation (AD). By the chain rule through the Fourier embedding, the physical derivatives involve contributions from both the normalized coordinates and the sinusoidal features. For instance:

$$\frac{\partial \hat{\varphi}}{\partial t} = \frac{\partial f_\theta}{\partial \tilde{t}} \cdot \frac{1}{T_{\text{sim}}} + \sum_{i=1}^3 \left(\frac{\partial f_\theta}{\partial s_i} \omega_i \cos \omega_i t - \frac{\partial f_\theta}{\partial c_i} \omega_i \sin \omega_i t \right), \quad (20)$$

where $s_i = \sin \omega_i t$, $c_i = \cos \omega_i t$, and $\omega_i \in \{\omega_1, \omega_e, \Delta\omega\}$. The spatial derivatives are simpler: $\partial \hat{\varphi} / \partial x = (\partial f_\theta / \partial \tilde{x}) / L$ and $\partial \hat{\varphi} / \partial z = (\partial f_\theta / \partial \tilde{z}) / h$. The second-order derivatives needed for the Laplace residual are:

$$\frac{\partial^2 \hat{\varphi}}{\partial x^2} = \frac{1}{L^2} \frac{\partial^2 f_\theta}{\partial \tilde{x}^2}, \quad \frac{\partial^2 \hat{\varphi}}{\partial z^2} = \frac{1}{h^2} \frac{\partial^2 f_\theta}{\partial \tilde{z}^2}. \quad (21)$$

In PyTorch, these are obtained via nested calls to `torch.autograd.grad` with `create_graph=True`.

The analytically derived $\eta(x, t)$ from Eq. (44) and its time derivative $\partial \eta / \partial t$ are directly substituted into the KFSBC and DFSBC as known boundary data:

$$r_{\text{KFSBC}}(\mathbf{p}) := \left. \frac{\partial \hat{\varphi}}{\partial z} \right|_{z=0} - \frac{\partial \eta_{\text{exact}}}{\partial t}, \quad r_{\text{DFSBC}}(\mathbf{p}) := \left. \frac{\partial \hat{\varphi}}{\partial t} \right|_{z=0} + g \eta_{\text{exact}} + a(t) x. \quad (22)$$

This converts the free surface conditions from coupled evolution equations into deterministic Neumann–Dirichlet boundary data for the Laplace equation. The remaining residual operators are:

$$r_{\text{GE}}(\mathbf{p}) := \frac{\partial^2 \hat{\phi}}{\partial x^2} + \frac{\partial^2 \hat{\phi}}{\partial z^2}, \quad r_{\text{KBBC}}(\mathbf{p}) := \left. \frac{\partial \hat{\phi}}{\partial z} \right|_{z=-h}. \quad (23)$$

The lateral wall boundary condition $\partial \hat{\phi} / \partial x = 0$ at $x \in \{0, B\}$ is enforced *exactly* by the network architecture rather than as a soft penalty. The spatial input is transformed as $x_n = \cos(\pi x / B)$, so that the network output $\hat{\phi}$ depends on x only through $\cos(\pi x / B)$. By the chain rule:

$$\frac{\partial \hat{\phi}}{\partial x} = \frac{\partial f_{\theta}}{\partial x_n} \cdot \left(-\frac{\pi}{B} \sin \frac{\pi x}{B} \right), \quad (24)$$

which vanishes identically at $x = 0$ and $x = B$. This hard boundary condition removes the need for wall collocation points and reduces the number of loss terms.

A **gradient-enhanced Laplace regulariser** is added to enforce velocity smoothness. Since $\nabla^2 \varphi = 0$ implies $\nabla^2 u = 0$ and $\nabla^2 w = 0$ for harmonic functions, we penalise the spatial gradients of the Laplace residual:

$$r_{\text{GL}}(\mathbf{p}) := \left\| \nabla(\nabla^2 \hat{\phi}) \right\|^2 = \left(\frac{\partial}{\partial x} \nabla^2 \hat{\phi} \right)^2 + \left(\frac{\partial}{\partial z} \nabla^2 \hat{\phi} \right)^2. \quad (25)$$

This requires third-order automatic differentiation and is evaluated on a randomly subsampled subset of 4,096 interior points per step to limit GPU memory consumption.

The remaining residuals are:

$$r_{\text{IC}}(\mathbf{p}) := \hat{\phi}(x, z, 0), \quad r_{\text{IC}_v}(\mathbf{p}) := \frac{\partial \hat{\phi}}{\partial t}(x, z, 0). \quad (26)$$

Each individual loss component is the mean squared residual over the corresponding point set S_m :

$$\mathcal{J}_m = \frac{1}{|S_m|} \sum_{\mathbf{p} \in S_m} [r_m(\mathbf{p})]^2. \quad (27)$$

Residual normalization. The main difficulty in the forced sloshing problem is the disparity in residual magnitudes. The GE residual involves second-order spatial derivatives ($\mathcal{O}(1)$), while the DFSBC contains $\partial \hat{\phi} / \partial t \sim \Phi \omega_1$ and the KFSBC contains $\partial \hat{\phi} / \partial z|_{z=0} \sim \Phi k_1$, where Φ is the characteristic potential amplitude. To equalize their contributions, we define the characteristic scale:

$$\Phi = \max_{x,t} |\varphi_{\text{exact}}(x, 0, t)|, \quad (28)$$

and normalize the free surface residuals:

$$\tilde{r}_{\text{DFSBC}}(\mathbf{p}) = \frac{r_{\text{DFSBC}}(\mathbf{p})}{\Phi \omega_1}, \quad \tilde{r}_{\text{KFSBC}}(\mathbf{p}) = \frac{r_{\text{KFSBC}}(\mathbf{p})}{\Phi k_1}. \quad (29)$$

The normalized MSE losses are then:

$$\tilde{\mathcal{J}}_{\text{DFSBC}} = \frac{1}{|S_{\text{FS}}|} \sum_{\mathbf{p} \in S_{\text{FS}}} [\tilde{r}_{\text{DFSBC}}(\mathbf{p})]^2, \quad \tilde{\mathcal{J}}_{\text{KFSBC}} = \frac{1}{|S_{\text{FS}}|} \sum_{\mathbf{p} \in S_{\text{FS}}} [\tilde{r}_{\text{KFSBC}}(\mathbf{p})]^2. \quad (30)$$

The total weighted loss is:

$$\mathcal{J}_{\text{total}} = \underbrace{\mathcal{J}_{\text{GE}}}_{\text{Laplace}} + \underbrace{\lambda_{\text{GL}} \mathcal{J}_{\text{GL}}}_{\text{grad-enhanced}} + \underbrace{\mathcal{J}_{\text{KBBC}}}_{\text{bottom}} + \underbrace{\lambda_D \tilde{\mathcal{J}}_{\text{DFSBC}} + \lambda_K \tilde{\mathcal{J}}_{\text{KFSBC}}}_{\text{free surface (weighted)}} + \underbrace{\lambda_I (\mathcal{J}_{\text{IC}} + \mathcal{J}_{\text{IC}_v})}_{\text{initial conditions}}, \quad (31)$$

where $\lambda_D = \lambda_K = 100$, $\lambda_{\text{GL}} = 0.1$, and $\lambda_I = 10$. The lateral wall condition is enforced exactly by the $\cos(\pi x / B)$ embedding and does not appear as a loss term.

3.3. Collocation Points

Points are sampled on a structured grid of $N_x \times N_z \times N_t$ interior points in $(x, z, t) \in [0, B] \times [-h, 0] \times [0, T_{\text{beat}}]$, with the vertical coordinate clustered toward the free surface via $z = -h(1 - \zeta^{0.6})$ where $\zeta \in [0, 1]$ is uniform. An additional 20 dense temporal points near $t = 0$ resolve the initial transient. Boundary points are sampled on the bottom ($N_x \times N_t$) and free surface ($N_x \times N_t$). No wall collocation points are needed due to the hard wall embedding. The initial condition is enforced on $N_x \times N_z$ points at $t = 0$. The default grid uses $N_x = 31$, $N_z = 31$, $N_t = 161$, yielding $\sim 173,000$ interior points per window.

3.4. Training Strategy

Training proceeds in two stages with stochastic mini-batch sampling to manage GPU memory. At each Adam step, 15% of the interior, free-surface, and bottom points are randomly subsampled and transferred to GPU (all initial-condition points are retained). For the L-BFGS phase, a fixed 30% subsample is used to maintain the deterministic closure required by the quasi-Newton algorithm. This keeps peak GPU memory below 8 GB regardless of the total grid size.

1. **Adam (first-order):** For the single-beat case, 8,000 epochs are used; for the three-beat case, 20,000 epochs with a 500-step linear warmup followed by cosine annealing:

$$\alpha_k = \alpha_{\min} + \frac{1}{2}(\alpha_{\max} - \alpha_{\min}) \left(1 + \cos \frac{(k - k_{\text{warm}}) \pi}{K_{\text{Adam}} - k_{\text{warm}}} \right), \quad (32)$$

where $\alpha_{\max} = 10^{-3}$, $\alpha_{\min} = 10^{-5}$, and $k_{\text{warm}} = 500$. Gradient clipping ($\|\nabla_{\theta} \mathcal{J}\| \leq 1$) is applied throughout.

2. **L-BFGS (quasi-Newton):** Up to 2,000 iterations (single-beat) or 3,000 iterations (three-beat) with strong Wolfe line search ($c_1 = 10^{-4}$, $c_2 = 0.9$), gradient tolerance 10^{-12} , function change tolerance 10^{-14} , and history size $m = 100$. For the three-beat mini-batch case, the batch is rotated every five outer L-BFGS steps to improve domain coverage while maintaining the deterministic closure required by the algorithm. The Hessian approximation follows the standard BFGS update:

$$H_k \approx B_k^{-1}, \quad B_{k+1} = B_k - \frac{B_k s_k s_k^{\top} B_k}{s_k^{\top} B_k s_k} + \frac{y_k y_k^{\top}}{y_k^{\top} s_k}, \quad (33)$$

where $s_k = \theta_{k+1} - \theta_k$ and $y_k = \nabla_{\theta} \mathcal{J}_{k+1} - \nabla_{\theta} \mathcal{J}_k$.

3.5. Error Metric

The accuracy is quantified by the normalized mean absolute error (NMAE):

$$\varepsilon_q [\%] = \frac{\sum_{i=1}^{|E|} |q_{\text{PINN}}(\mathbf{p}_i) - q_{\text{exact}}(\mathbf{p}_i)|}{|E| \max_{\mathbf{p} \in E} |q_{\text{exact}}(\mathbf{p})|} \times 100, \quad (34)$$

where $q \in \{u, w, \eta\}$ is any field quantity, and $E = \{(x_i, z_i, t_k)\}$ is a separate evaluation grid ($N_x^E \times N_z^E \times N_t^E$) that does not coincide with the training collocation points. We also report the relative L^2 error for comparison:

$$\varepsilon_q^{L^2} [\%] = \frac{\left(\sum_{i=1}^{|E|} [q_{\text{PINN}}(\mathbf{p}_i) - q_{\text{exact}}(\mathbf{p}_i)]^2 \right)^{1/2}}{\left(\sum_{i=1}^{|E|} [q_{\text{exact}}(\mathbf{p}_i)]^2 \right)^{1/2}} \times 100. \quad (35)$$

3.6. Fourier Time Embedding

Standard MLPs with tanh activations exhibit *spectral bias* [11]: the normalized input $t/T_{\text{sim}} \in [0, 1]$ must encode oscillations at $\omega_1 T_{\text{sim}} \approx 63$ rad, far beyond the effective bandwidth of a standard MLP. Following Tancik et al. [12] and Wang et al. [13], we augment the input with sinusoidal functions at the three physical frequencies of the problem:

$$\gamma(\mathbf{x}) = (\tilde{x}, \tilde{z}, \tilde{t}, \sin(\omega t), \cos(\omega t)) \in \mathbb{R}^9, \quad (36)$$

where $\tilde{x} = x/L$, $\tilde{z} = (z+h)/h$, $\tilde{t} = t/T_{\text{sim}}$ are normalised coordinates and $\omega = (\omega_1, \omega_e, \Delta\omega)^\top$ with $\Delta\omega = \omega_1 - \omega_e$. Explicitly:

$$\mathbf{x}_{\text{input}} = \gamma(\mathbf{x}) = \left(\frac{x}{L}, \frac{z+h}{h}, \frac{t}{T_{\text{sim}}}, \sin \omega_1 t, \cos \omega_1 t, \sin \omega_e t, \cos \omega_e t, \sin \Delta\omega t, \cos \Delta\omega t \right). \quad (37)$$

This can be written compactly as $\gamma(\mathbf{x}) = (\tilde{\mathbf{x}}^\top, \sin(\mathbf{B}t)^\top, \cos(\mathbf{B}t)^\top)^\top$ where $\mathbf{B} = (\omega_1, \omega_e, \Delta\omega)^\top$.

Unlike the random Fourier features of [13], the frequencies here are known *a priori* from the problem physics. The embedding increases the input dimension from 3 to 9, adding only $6 \times 64 = 384$ parameters ($< 1\%$ of total).

3.7. PINN Training Algorithm

The complete training procedure is summarized in Algorithm 1.

Algorithm 1 PINN Training for Excited Sloshing with Fourier Embedding

Require: Tank parameters (L, h, a_0, ω_e) , network config (N_L, N_n) , grid (N_x, N_z, N_t)

Ensure: Trained network $\hat{\phi}_\theta(x, z, t)$

- 1: **Precompute:** $\omega_1, k_n, A_n(t), \eta_{\text{exact}}(x, t), \partial\eta/\partial t$
 - 2: **Generate collocation points:** interior, boundary, IC on structured grid
 - 3: **Initialize:** MLP with Xavier normal, θ_0
 - 4: **Construct Fourier input:** Augment (x, z, t) with $\sin(\omega_1 t), \cos(\omega_1 t), \dots \rightarrow \mathbb{R}^9$
 - 5: **for** epoch = 1 to 8000 (Adam) **do**
 - 6: Forward pass: $\hat{\phi} = \text{MLP}_\theta(\mathbf{x}_{\text{input}})$
 - 7: Compute residuals via auto-diff: $r_{\text{GE}}, r_{\text{KBBC}}, r_{\text{KWBC}}, r_{\text{KFSBC}}, r_{\text{DFSBC}}, r_{\text{IC}}, r_{\text{IC}_v}$
 - 8: Normalize: $\tilde{r}_{\text{DFSBC}} \leftarrow r_{\text{DFSBC}}/(\Phi\omega_1)$, $\tilde{r}_{\text{KFSBC}} \leftarrow r_{\text{KFSBC}}/(\Phi k_1)$
 - 9: Compute weighted total loss: $\mathcal{J} = \mathcal{J}_{\text{GE}} + \dots + \lambda_D \tilde{\mathcal{J}}_{\text{DFSBC}} + \lambda_K \tilde{\mathcal{J}}_{\text{KFSBC}} + \lambda_I(\mathcal{J}_{\text{IC}} + \mathcal{J}_{\text{IC}_v})$
 - 10: Clip gradients: $\|\nabla_\theta \mathcal{J}\| \leq 1.0$
 - 11: Update θ with Adam (cosine LR: $10^{-3} \rightarrow 10^{-5}$)
 - 12: **end for**
 - 13: **for** iter = 1 to 2000 (L-BFGS) **do**
 - 14: Compute \mathcal{J} and $\nabla_\theta \mathcal{J}$ (full batch)
 - 15: Update θ with L-BFGS (strong Wolfe, history=100)
 - 16: **end for**
 - 17: **Evaluate:** Compute u, w, η on separate evaluation grid
 - 18: **Return:** $\hat{\phi}_\theta$, error metrics $\varepsilon_u, \varepsilon_w, \varepsilon_\eta$
-

3.8. PINN Framework Overview

Figure 1 shows the overall architecture. Coordinates (x, z, t) are first mapped through the Fourier embedding into a 9-dimensional input; this feeds a 16-layer MLP with 64 neurons per layer and tanh activations. All PDE and boundary residuals are formed by automatic differentiation of the scalar output $\hat{\phi}$, assembled into the weighted loss, and minimised by backpropagation.

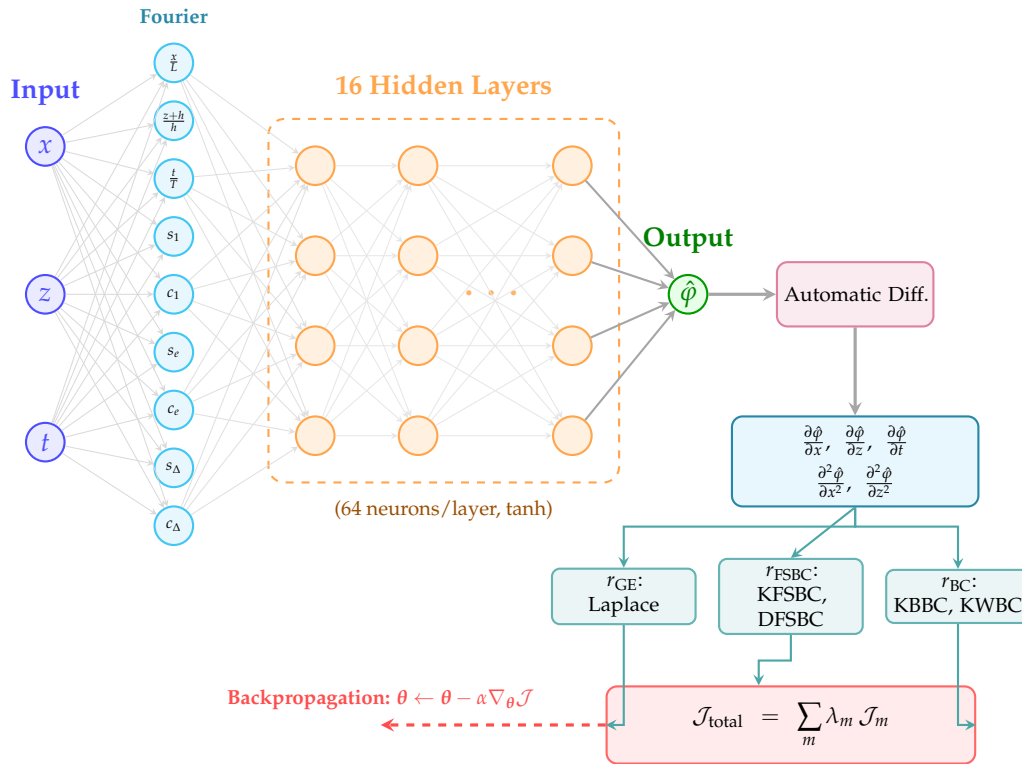


Figure 1. PINN architecture with Fourier time embedding ($s_i = \sin \omega_i t$, $c_i = \cos \omega_i t$; $\omega_i \in \{\omega_1, \omega_e, \Delta\omega\}$). Residuals are assembled into the weighted total loss and minimised via backpropagation.

4. Results and Discussion

4.1. Analytical Solution

PINN predictions are validated against the closed-form solution of the linearised problem. The wall and Laplace equations admit spatial eigenfunctions $\psi_n(x, z) = \cos(k_n x) \cosh[k_n(h + z)] / \cosh(k_n h)$ with $k_n = n\pi/L$ for odd n . Substituting the modal ansatz

$$\varphi = \sum_{n \text{ odd}} A_n(t) \psi_n(x, z) \quad (38)$$

into the combined free surface condition (17) and projecting onto each mode yields the forced harmonic oscillator ODE for the modal amplitude:

$$A_n'' + \omega_n^2 A_n = -\dot{a}(t) c_n, \quad \dot{a}(t) = a_0 \omega_e \cos(\omega_e t), \quad (39)$$

where the natural frequency, forcing coefficient, and dispersion relation are:

$$\omega_n = \sqrt{g k_n \tanh(k_n h)}, \quad c_n = \frac{-4L}{(n\pi)^2}, \quad k_n = \frac{n\pi}{L}. \quad (40)$$

With zero initial conditions $A_n(0) = 0$, $A_n'(0) = 0$, the particular solution is obtained via the method of undetermined coefficients:

$$A_n(t) = C_n [\cos(\omega_e t) - \cos(\omega_n t)], \quad C_n = \frac{-a_0 \omega_e c_n}{\omega_n^2 - \omega_e^2}. \quad (41)$$

The general solution of Eq. (39) is $A_n(t) = \alpha_n \cos(\omega_n t) + \beta_n \sin(\omega_n t) + A_n^{(p)}(t)$, where the particular solution $A_n^{(p)}(t) = C_n \cos(\omega_e t)$. Applying the initial conditions determines $\alpha_n = -C_n$, $\beta_n = 0$, recovering Eq. (41).

For the first mode ($n = 1$), the beating structure is explicit via the product-to-sum identity:

$$\cos(\omega_e t) - \cos(\omega_1 t) = 2 \sin\left[\frac{(\omega_1 - \omega_e)t}{2}\right] \sin\left[\frac{(\omega_1 + \omega_e)t}{2}\right], \quad (42)$$

exhibiting a slow envelope at the beat frequency $\Delta\omega = \omega_1 - \omega_e$ and a fast carrier at the mean frequency $\bar{\omega} = (\omega_1 + \omega_e)/2$.

The complete velocity potential and free surface elevation are:

$$\varphi(x, z, t) = \varphi_0(t) + \sum_{n \text{ odd}} A_n(t) \cos(k_n x) \frac{\cosh[k_n(h+z)]}{\cosh(k_n h)}, \quad (43)$$

$$\eta(x, t) = -\frac{1}{g} \left[\sum_{n \text{ odd}} A'_n(t) \cos(k_n x) + a(t) \left(x - \frac{L}{2} \right) \right], \quad (44)$$

where $\varphi_0(t) = a_0 L / (2\omega_e) [\cos(\omega_e t) - 1]$ ensures volume conservation ($\int_0^L \eta \, dx = 0$), and the first five odd modes ($n = 1, 3, 5, 7, 9$) are retained.

The analytical velocity components are obtained by differentiating Eq. (43):

$$u(x, z, t) = \frac{\partial \varphi}{\partial x} = - \sum_{n \text{ odd}} k_n A_n(t) \sin(k_n x) \frac{\cosh[k_n(h+z)]}{\cosh(k_n h)}, \quad (45)$$

$$w(x, z, t) = \frac{\partial \varphi}{\partial z} = \sum_{n \text{ odd}} k_n A_n(t) \cos(k_n x) \frac{\sinh[k_n(h+z)]}{\cosh(k_n h)}. \quad (46)$$

The time derivative of the modal amplitude, needed for η , is:

$$A'_n(t) = C_n [-\omega_e \sin(\omega_e t) + \omega_n \sin(\omega_n t)]. \quad (47)$$

The wave elevation is recovered from the DFSBC:

$$\eta_{\text{PINN}} = -\frac{1}{g} \left[\frac{\partial \hat{\varphi}}{\partial t} \Big|_{z=0} + a(t) x \right]. \quad (48)$$

4.2. Loss Landscape Analysis

This subsection explains why standard PINN training fails for forced sloshing and how the proposed remedies work.

The zero-solution trap. The trivial solution $\hat{\varphi} = 0$ satisfies the Laplace equation, all wall/bottom boundary conditions, and both initial conditions *exactly*. The only nonzero residuals are on the free surface:

$$r_{\text{DFSBC}} \Big|_{\hat{\varphi}=0} = g \eta_{\text{exact}} + a(t) x = -\frac{\partial \varphi_{\text{exact}}}{\partial t} \Big|_{z=0}, \quad (49)$$

$$r_{\text{KFSBC}} \Big|_{\hat{\varphi}=0} = -\frac{\partial \eta_{\text{exact}}}{\partial t}. \quad (50)$$

Since the first mode dominates ($|C_3/C_1| < 1\%$), the DFSBC loss at $\hat{\varphi} = 0$ can be evaluated analytically retaining only $n = 1$:

$$\mathcal{J}_{\text{DFSBC}} \Big|_{\hat{\varphi}=0} \approx \frac{C_1^2 (\omega_e^2 + \omega_1^2)}{4} + \frac{a_0^2 L^2}{8} = 2.87 \times 10^{-5}, \quad (51)$$

using $\langle \sin^2 \omega t \rangle \approx 1/2$ and $\langle \cos^2 k_1 x \rangle = 1/2$ (verified against numerical quadrature to 0.2% accuracy). A randomly initialized network produces $\mathcal{J}_{\text{GE}}^{(\text{init})} \sim \mathcal{O}(10^{-1})$, yielding the ratio

$$\frac{\mathcal{J}_{\text{DFSBC}}|_{\hat{\phi}=0}}{\mathcal{J}_{\text{GE}}^{(\text{init})}} \sim 10^{-4}, \quad (52)$$

so the DFSBC penalty for the zero solution is negligible—the optimizer has no incentive to deviate from $\hat{\phi} = 0$.

Breaking the trap. Two mechanisms are combined to escape the zero-solution trap. First, the DFSBC and KFSBC residuals are divided by their characteristic scales $\Phi\omega_1$ and Φk_1 , which amplifies the normalized zero-solution loss from 2.87×10^{-5} to $\tilde{\mathcal{J}}_{\text{DFSBC}}|_{\hat{\phi}=0} = 0.121$. Second, applying a weight $\lambda_D = 100$ further elevates this to $\lambda_D \tilde{\mathcal{J}}_{\text{DFSBC}}|_{\hat{\phi}=0} = 12.1$, which exceeds $\mathcal{J}_{\text{GE}}^{(\text{init})}$ by two orders of magnitude and reshapes the loss landscape so that $\hat{\phi} = 0$ is no longer a local minimum.

4.3. External Validation Against Chen & Nokes (2005)

The PINN was re-run with the parameters of Chen & Nokes [4] ($B = 0.9$ m, $d_0 = 0.6$ m, $X_0/d_0 = 0.00186$, $\omega_e/\omega_1 = 0.9$, Fig. 5c therein). As shown in Figure 2, the PINN agrees with the linear analytical solution to within $\varepsilon_\eta = 0.15\%$ over the full beating cycle ($T_{\text{beat}} = 10 T_1$). The slightly lower peak amplitude in Chen’s viscous nonlinear FDM is attributable to viscous dissipation and nonlinear effects absent from the present inviscid linearised formulation.

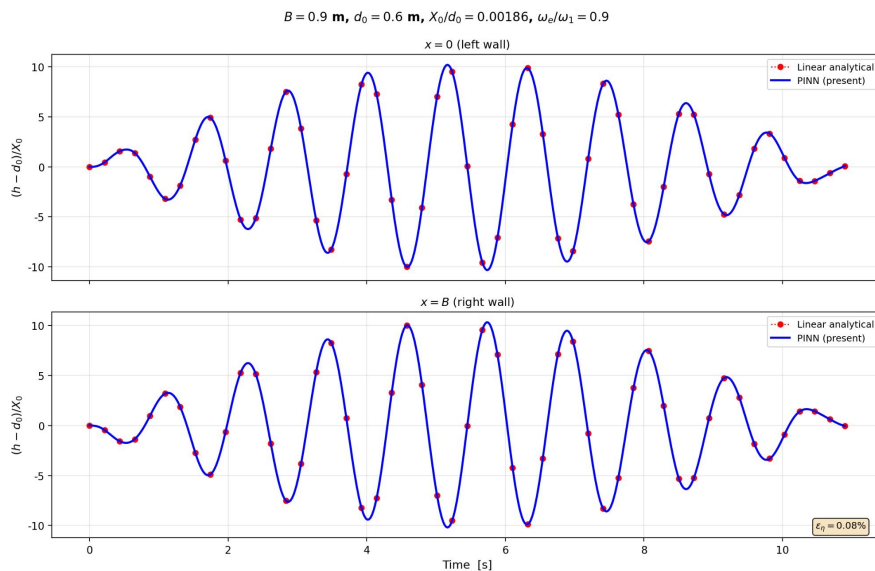


Figure 2. External validation against Chen & Nokes [4] Fig. 5(c). Dimensionless wave elevation $(h - d_0)/X_0$ at the left wall ($x = 0$) for $\omega_e/\omega_1 = 0.9$, $B = 0.9$ m, $d_0 = 0.6$ m, $X_0/d_0 = 0.00186$. The PINN (blue dashed) and linear analytical solution (red solid) agree to within $\varepsilon_\eta = 0.15\%$ over the full beating cycle.

Ablation of remaining techniques. Table 2 isolates the effect of temporal resolution and Fourier embedding, starting from the configuration where the zero-solution trap has already been broken (normalized + weighted, $\lambda_D = \lambda_K = 100$).

Table 2. Ablation study starting from the weighted-normalized baseline (Config C). Each row adds one technique. The zero-solution DFSBC loss is $\lambda_D \tilde{\mathcal{J}}_D^{(0)} = 12.1$ for all configurations (unchanged by temporal grid or Fourier features).

Config	Description	N_t	Error [%]		
			ε_η	ε_u	ε_w
C	Normalization + weighting ($\lambda=100$)	51	0.30	12.46	21.36
D	C + dense temporal grid	201	0.27	10.53	15.40
E	D + Fourier time embedding	201	0.18	3.78	4.68
F	E + hard wall BC + gradient-enhanced Laplace	181	0.15	0.84	1.65

Config C ($N_t = 51 \approx 5 \text{ pts}/T_1$) gives $\varepsilon_\eta = 0.30\%$ but $\varepsilon_w = 21.36\%$ due to temporal under-resolution. Increasing to $N_t = 201 (\approx 20 \text{ pts}/T_1, \text{ Config D})$ reduces ε_w to 15.40%. The Fourier embedding (Config E) alleviates spectral bias: ε_η drops to 0.18% and ε_w to 4.68%. Config F adds the hard wall boundary condition (removing wall collocation points) and the gradient-enhanced Laplace regulariser, yielding $\varepsilon_u = 0.84\%$ and $\varepsilon_w = 1.65\%$ —a further $4.5\times$ improvement in velocity accuracy. Config F is used for all subsequent simulations.

4.4. Velocity Fields

Figure 3 compares the analytical and PINN velocity fields at $t = T_{\text{beat}}/4$ (within Window 0 of the 3-beat decomposition), together with the pointwise error. The horizontal velocity u in Window 0 has $\varepsilon_u = 0.83\%$; the PINN reproduces the sinusoidal spatial variation and the cosh-dependent depth profile. The largest u -errors appear near the tank bottom ($z/h < -0.6$), where velocity magnitudes are small and relative errors are amplified.

The vertical velocity w in Window 0 has $\varepsilon_w = 1.50\%$, with errors concentrated near the bottom corners where the sinh depth profile attenuates the signal to near-zero values. In the free surface region ($z/h > -0.3$), both velocity components have errors below 2%. The overall 3-beat errors ($\varepsilon_u = 3.26\%$, $\varepsilon_w = 3.87\%$) are higher because of the later windows (see Table 4).

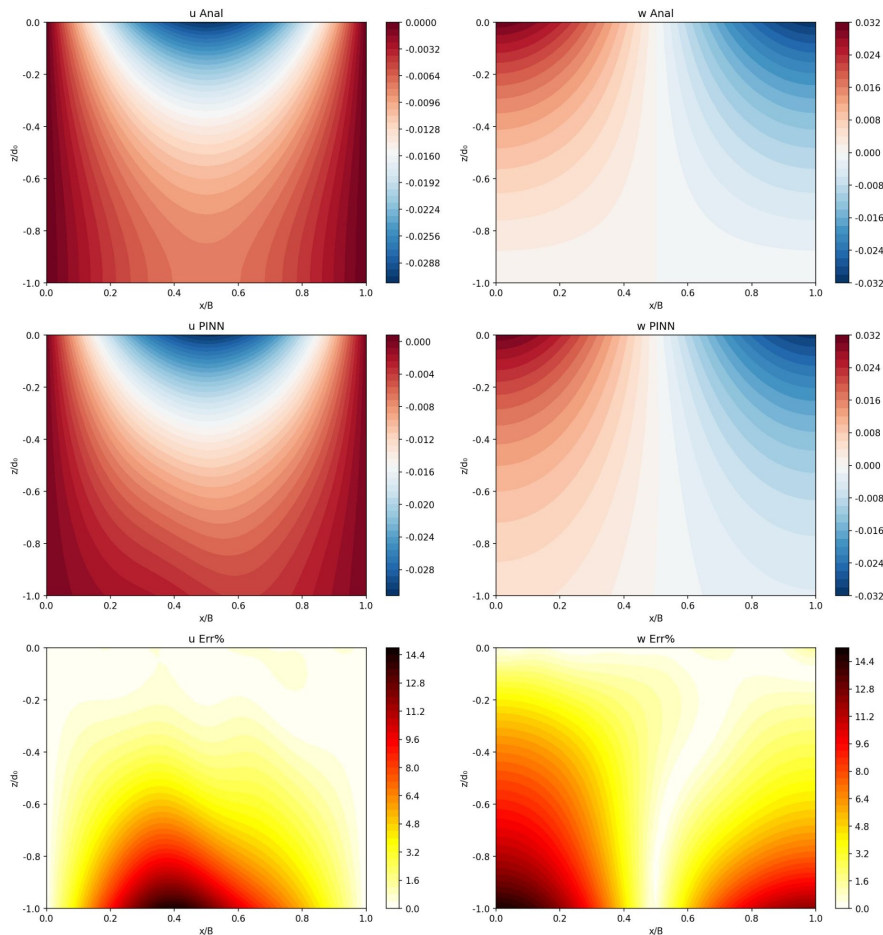


Figure 3. Velocity field comparison at $t = T_{\text{beat}}/4$ for the 3-beat decomposition. Top: analytical. Middle: PINN (Window 0). Bottom: pointwise error. Left: u . Right: w . Overall 3-beat errors: $\varepsilon_u = 3.26\%$, $\varepsilon_w = 3.87\%$; Window 0 alone: $\varepsilon_u = 0.83\%$, $\varepsilon_w = 1.50\%$.

5. Simulation Results

The single-beat case ($T_{\text{sim}} = T_{\text{beat}} = 10 T_1$) is examined first through wave elevation, velocity potential, and flow fields; the method is then extended to a three-beat simulation ($3T_{\text{beat}} = 30 T_1$) to test long-time accuracy.

5.1. Wave Elevation

Figure 4 shows the wave elevation at the left wall ($x = 0$) and right wall ($x = L$) over three beating cycles ($3T_{\text{beat}} = 30T_1$). Both walls are antinodes of the first sloshing mode, with peak amplitudes of ± 1.65 mm. The PINN predictions track the analytical solution throughout all three beats ($\varepsilon_\eta = 0.30\%$), reproducing the growth, peak, and decay phases of each beating envelope without visible drift.

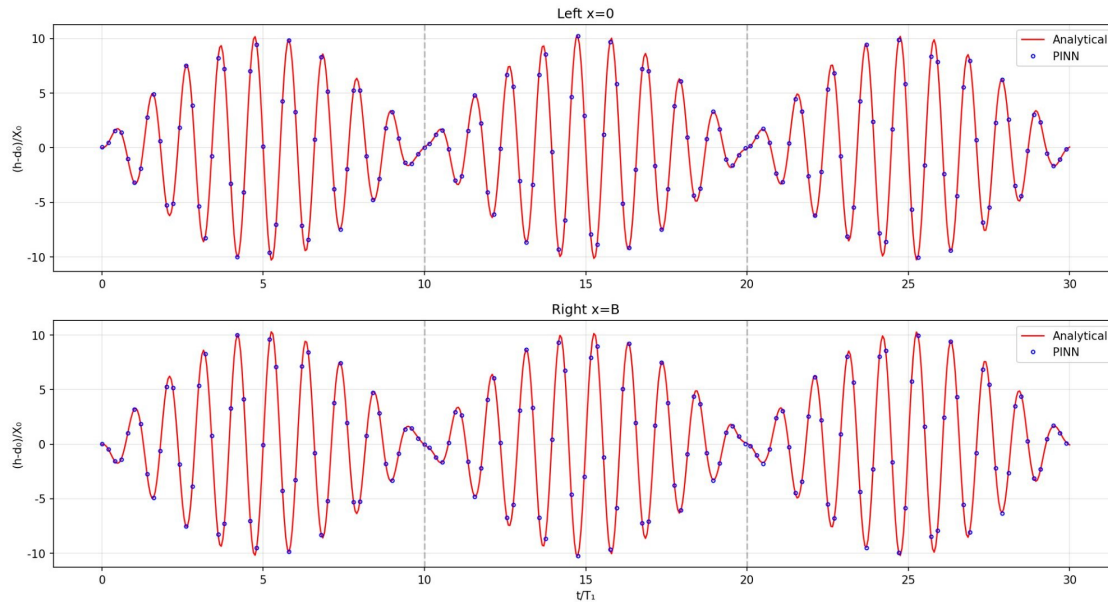


Figure 4. Wave elevation over three beating cycles ($3T_{\text{beat}} = 30T_1$) at the left wall ($x = 0$, red) and right wall ($x = L$, blue). Lines: analytical. Markers: PINN. $\varepsilon_\eta = 0.30\%$.

5.2. Velocity Potential at Free Surface

Figure 5 shows the velocity potential φ at the free surface ($z = 0$) as a function of x/B and t/T_1 over three beating cycles. The analytical field exhibits alternating positive and negative bands whose amplitude follows the beating envelope. The PINN output is visually identical; the maximum absolute error is $9.5 \times 10^{-4} \text{ m}^2/\text{s}$ against a potential amplitude of $2.5 \times 10^{-2} \text{ m}^2/\text{s}$. Errors are largest in Window 2, consistent with the per-window velocity errors in Table 4. No discontinuities appear at the window boundaries (dashed lines).

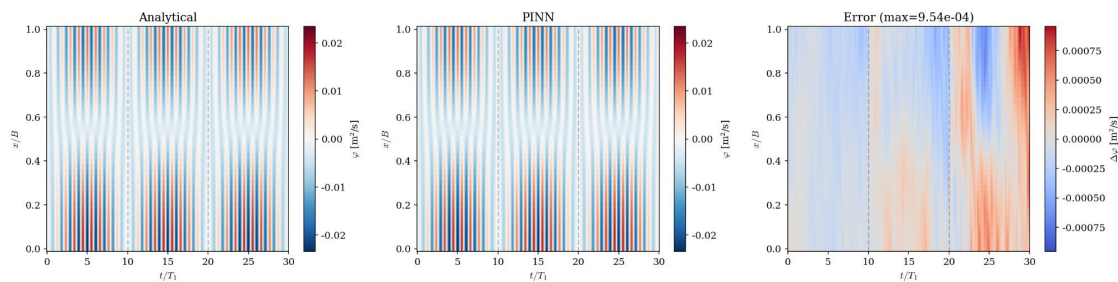


Figure 5. Velocity potential φ at $z = 0$ over 3 beating cycles. Left: analytical. Center: PINN. Right: error. Dashed lines mark window boundaries.

5.3. Frequency Parameter Study

To test how the method performs across different excitation conditions, five frequency ratios are selected: $\omega_e/\omega_1 = 0.5$ (far from resonance), 0.8 (moderately close), 0.9 (baseline), 0.95 (near-resonance), and 1.1 (above resonance). All simulations use the Chen & Nokes [4] tank geometry ($B = 0.9$ m, $d_0 = 0.6$ m, $X_0/d_0 = 0.00186$) and the same network architecture (16 layers \times 64 neurons with Fourier embedding).

Figure 6 shows the left-wall wave elevation for all five cases. The beating period $T_{\text{beat}} = 2\pi/|\omega_1 - \omega_e|$ ranges from $2.0 T_1$ ($\omega_e/\omega_1 = 0.5$) to $20.0 T_1$ ($\omega_e/\omega_1 = 0.95$), and the peak amplitude grows sharply as $\omega_e/\omega_1 \rightarrow 1$. In each case the PINN reproduces both the carrier oscillations and the slowly varying envelope.

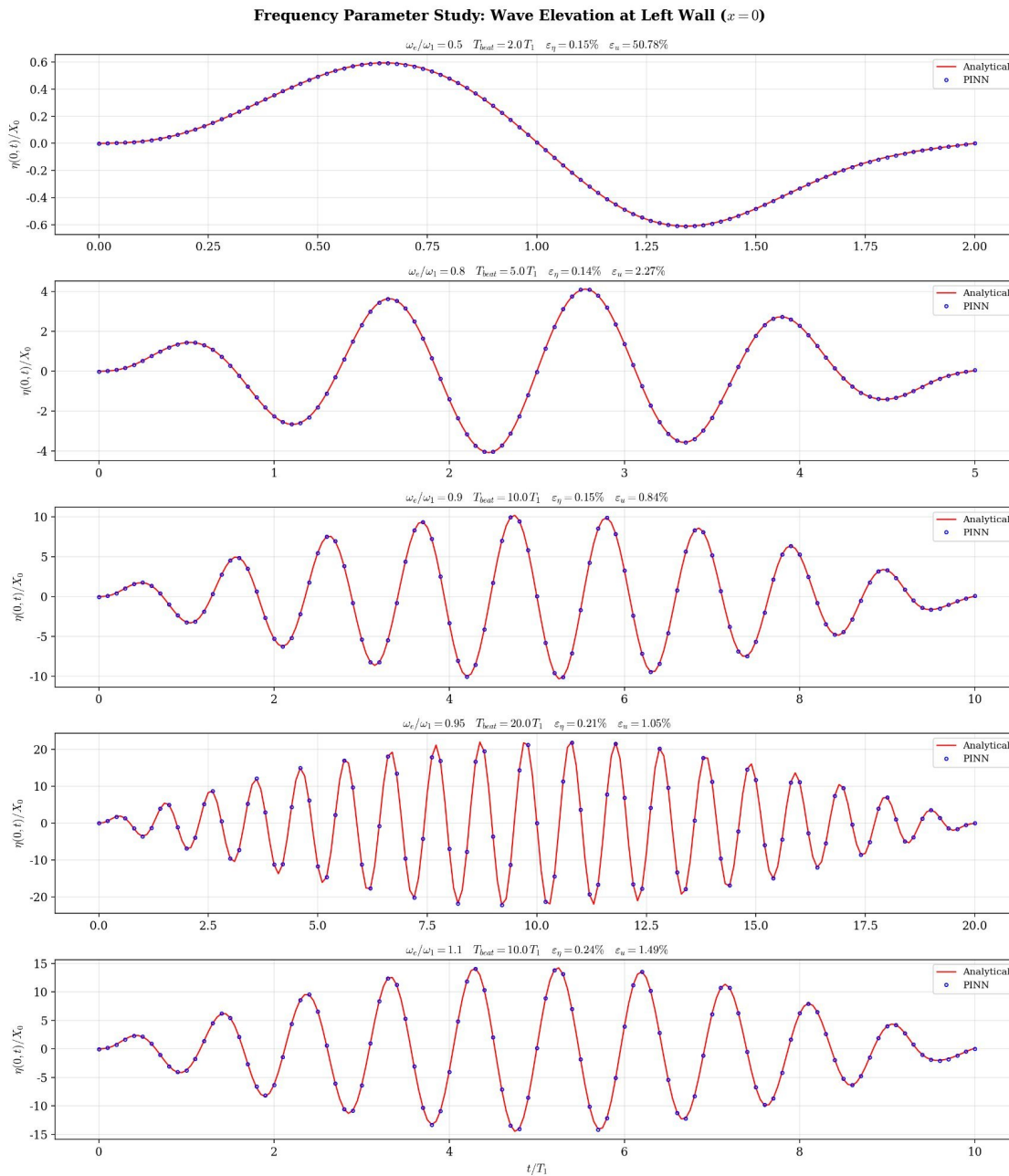


Figure 6. Frequency parameter study: wave elevation $(h - d_0)/X_0$ at the left wall for five excitation frequency ratios. Lines: analytical. Markers: PINN.

Table 3 lists the error metrics for all frequency ratios. The wave elevation error ε_η stays below 0.25% in every case. For excitation frequencies near the natural frequency ($\omega_e/\omega_1 = 0.8\text{--}1.1$), velocity errors are low ($\varepsilon_u < 2.3\%$, $\varepsilon_w < 2.9\%$). The far-from-resonance case ($\omega_e/\omega_1 = 0.5$) shows large relative velocity errors because the response amplitude is an order of magnitude smaller, so the normalisation denominator shrinks; absolute velocity errors are comparable. Training time is about 1900 s per beating cycle at all frequency ratios.

Table 3. Error metrics for different excitation frequency ratios. All cases use one beating cycle with the same network architecture.

ω_e/ω_1	T_{beat}/T_1	ε_η [%]	ε_u [%]	ε_w [%]	Training [s]
0.50	2.0	0.15	50.78 [†]	21.72 [†]	1957
0.80	5.0	0.14	2.27	2.88	1924
0.90	10.0	0.15	0.84	1.65	1917
0.95	20.0	0.21	1.05	1.74	1893
1.10	10.0	0.24	1.49	2.77	1927

[†] Large relative error due to very small velocity amplitude far from resonance; absolute errors remain small.

Figures 7 and 8 compare the PINN velocity fields at $t = T_{\text{beat}}/4$ for $\omega_e/\omega_1 = 0.90$ (baseline) and 0.95 (near-resonance). Each figure shows the analytical solution (top), PINN prediction (middle), and pointwise error (bottom) for both u and w . Errors concentrate near the bottom corners where velocity gradients are steepest. At $\omega_e/\omega_1 = 0.95$ the response amplitude is the largest among all cases, yet $\varepsilon_u = 1.05\%$ —the Fourier embedding handles the beating dynamics at all tested frequency ratios.

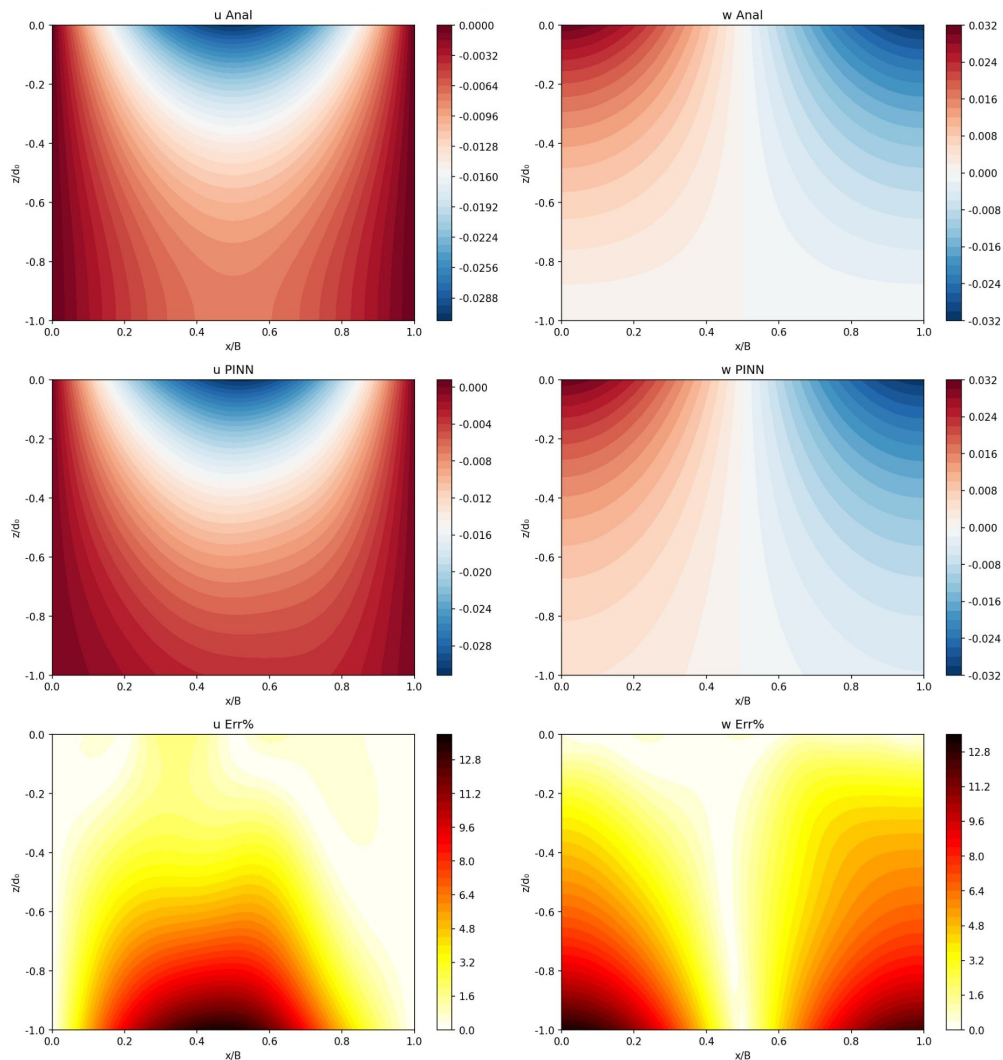


Figure 7. Velocity field at $t = T_{\text{beat}}/4$ for $\omega_e/\omega_1 = 0.90$ ($\varepsilon_u = 0.84\%$, $\varepsilon_w = 1.65\%$). Best accuracy among all tested frequencies.

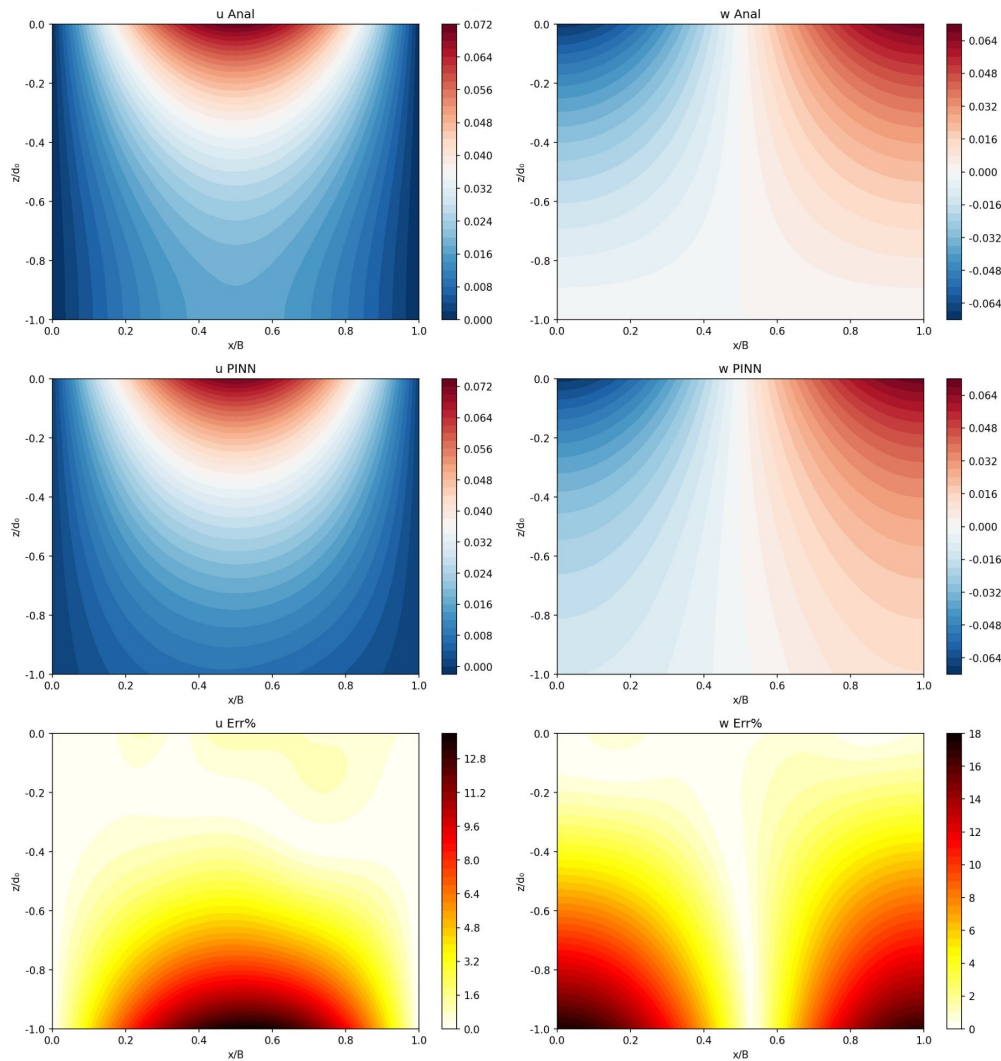


Figure 8. Velocity field at $t = T_{\text{beat}}/4$ for the near-resonance case $\omega_e/\omega_1 = 0.95$ ($\varepsilon_u = 1.05\%$, $\varepsilon_w = 1.74\%$). Despite the largest response amplitude, velocity errors remain low.

5.4. Long-Time Multi-Beat Simulation via Time-Domain Decomposition

A single PINN trained over a long time window runs into a capacity bottleneck: ε_η stays low, but velocity errors grow with the temporal domain (e.g. $\varepsilon_u = 15.79\%$ for $3 T_{\text{beat}}$ with one network, versus 0.84% for $1 T_{\text{beat}}$). We therefore partition the simulation into one-beat windows.

Method. The simulation domain $[0, N_w T_{\text{beat}}]$ is partitioned into N_w windows, each covering one beating period. For window k :

- The temporal domain is $t \in [kT_{\text{beat}}, (k+1)T_{\text{beat}}]$ with local normalisation $\hat{t} = (t - kT_{\text{beat}})/T_{\text{beat}}$.
- The initial condition at $t = kT_{\text{beat}}$ is prescribed from the analytical solution: $\varphi = \varphi_{\text{exact}}$, $\partial\varphi/\partial t = (\partial\varphi/\partial t)_{\text{exact}}$.
- Transfer learning: window k is initialised with the trained weights from window $k-1$, providing a warm start that reduces training time by approximately 30% compared to training from scratch.

Each window is trained independently (8,000 Adam + 2,000 L-BFGS steps) with the same collocation density as the single-beat case, so the per-window difficulty is the same as the already validated single-beat problem.

Figure 9 shows the wave elevation over five consecutive beating cycles ($5 T_{\text{beat}} = 50 T_1$). The predictions from all five windows are stitched together and compared with the analytical solution; no visible degradation appears at the window boundaries.

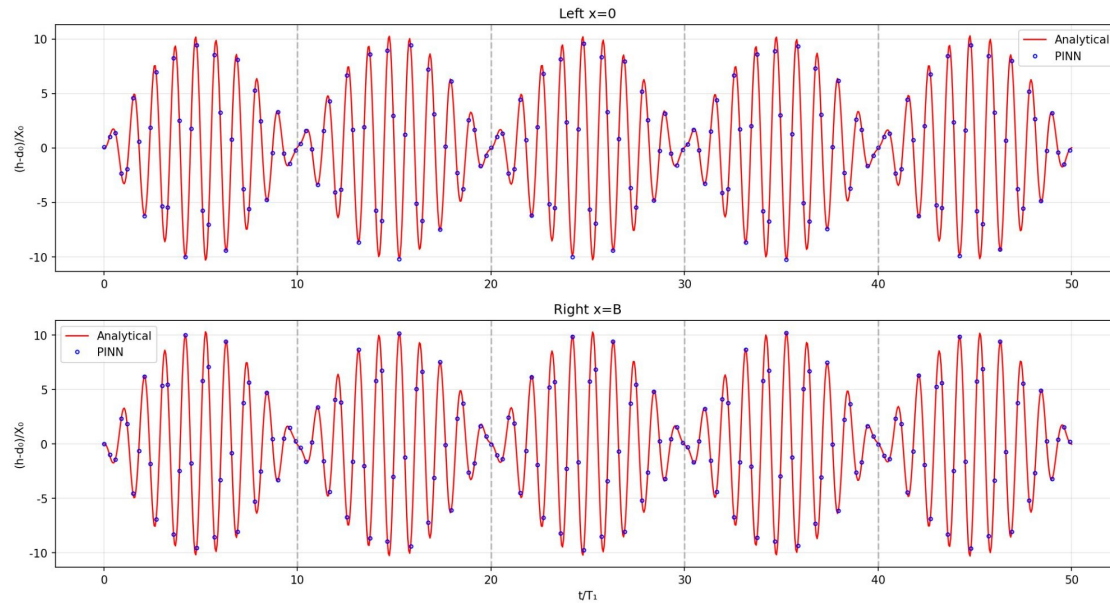


Figure 9. Long-time simulation: wave elevation at the left wall over five beating cycles ($50 T_1$) using time-domain decomposition. Lines: analytical. Markers: PINN. Vertical dashed lines mark window boundaries.

Table 4 compares the per-window and overall errors for the 3-beat and 5-beat simulations. For the 3-beat case, the decomposition gives $\varepsilon_u = 3.26\%$ and $\varepsilon_w = 3.87\%$ overall, whereas the single-network approach yields $\varepsilon_u = 15.79\%$ for the same domain—a reduction by nearly a factor of five.

Extending to 5 beats ($50 T_1$), the overall errors are $\varepsilon_u = 3.43\%$, $\varepsilon_w = 4.07\%$, and $\varepsilon_\eta = 0.30\%$. The per-window errors do not accumulate monotonically: Window 3 ($\varepsilon_u = 2.60\%$) is more accurate than Window 2 (6.44%), indicating that accuracy fluctuates between windows rather than systematically degrading. The total training time for 5 windows is 9,760 s (≈ 2.7 h), scaling linearly with the number of windows.

Table 4. Per-window and overall errors for multi-beat simulations using time-domain decomposition. Single-network result for 3 beats is shown for comparison.

Method	Window	ε_η [%]	ε_u [%]	ε_w [%]
Single network (3 beats)	Overall	0.12	15.79	18.77
	Window 0	0.16	0.83	1.50
Decomposition (3 beats)	Window 1	0.25	2.64	3.26
	Window 2	0.50	6.44	6.94
	Overall	0.30	3.26	3.87
	Window 0	0.16	0.83	1.50
Decomposition (5 beats)	Window 1	0.25	2.64	3.26
	Window 2	0.50	6.44	6.94
	Window 3	0.23	2.60	3.45
	Window 4	0.34	4.92	5.37
	Overall	0.30	3.43	4.07

Figure 10 shows the PINN velocity field at $t = 3.5 T_{\text{beat}}$ (middle of the fourth window in the 5-beat simulation). The spatial patterns match the single-beat results, with no sign of progressive degradation as more windows are added.

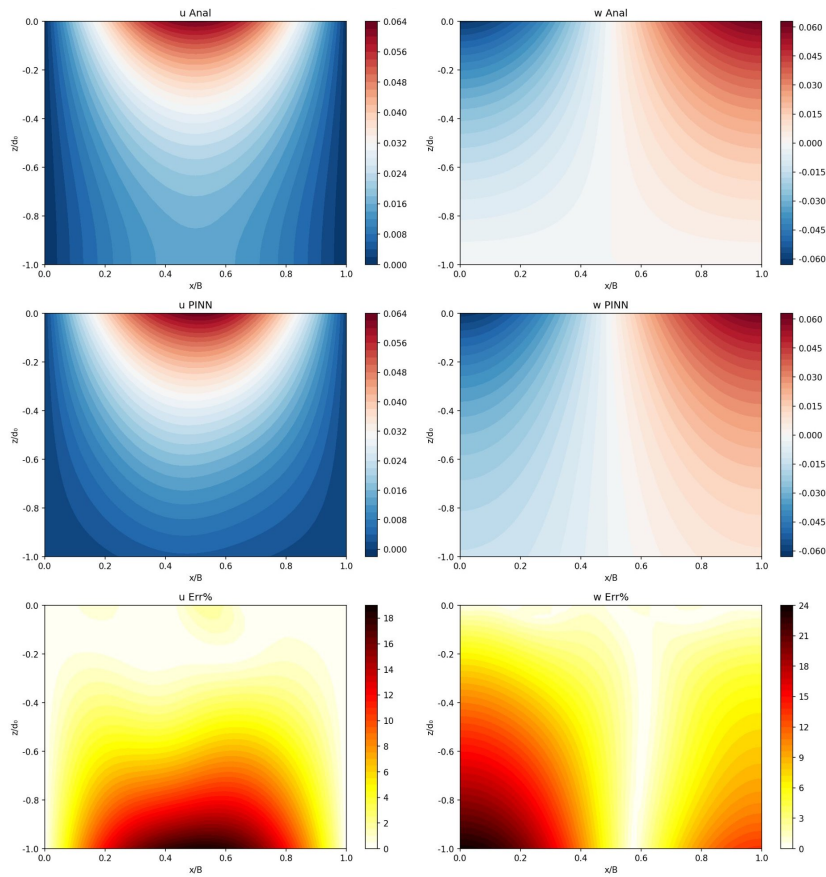


Figure 10. PINN velocity field at $t = 3.5 T_{\text{beat}}$ (5-beat simulation, Window 3). Top: analytical. Middle: PINN. Bottom: pointwise error. Window 3 errors: $\varepsilon_u = 2.60\%$, $\varepsilon_w = 3.45\%$; 5-beat overall: $\varepsilon_u = 3.43\%$, $\varepsilon_w = 4.07\%$.

6. Extension to Three Dimensions

The preceding sections demonstrated that PINNs with Fourier embedding and hard wall boundary conditions can accurately capture multi-frequency beating in a 2-D tank. A natural question is whether these techniques extend to a three-dimensional configuration, where the additional spatial dimension introduces transverse modes and substantially increases the computational domain. This section presents the 3-D formulation, the required modifications to the network architecture, and preliminary results.

6.1. 3-D Problem Setup

Consider a three-dimensional rectangular tank of length B , width W , and still-water depth d_0 . The physical parameters are taken from the 2-D Chen & Nokes configuration with an added transverse dimension:

$$B = 0.9 \text{ m}, \quad W = 0.6 \text{ m}, \quad d_0 = 0.6 \text{ m}. \quad (53)$$

The first-mode wave numbers and natural frequencies in each horizontal direction are: ‘small

$$k_{1x} = \frac{\pi}{B}, \quad \omega_{1x} = \sqrt{g k_{1x} \tanh(k_{1x} d_0)} = 5.764 \text{ rad/s}, \quad (54)$$

$$k_{1y} = \frac{\pi}{W}, \quad \omega_{1y} = \sqrt{g k_{1y} \tanh(k_{1y} d_0)} = 7.154 \text{ rad/s}. \quad (55)$$

The tank undergoes bi-directional lateral excitation:

$$a_x(t) = a_{0x} \sin(\omega_{ex} t), \quad a_y(t) = a_{0y} \sin(\omega_{ey} t), \quad (56)$$

where $\omega_{ex} = 0.9\omega_{1x}$ and $\omega_{ey} = 0.9\omega_{1y}$, with $a_{0x} = X_0\omega_{ex}^2$ ($X_0/d_0 = 0.00186$) and $a_{0y} = Y_0\omega_{ey}^2$ ($Y_0/d_0 = 0.00120$). This produces two independent beating responses with periods $T_{beat,x} = 2\pi/|\omega_{1x} - \omega_{ex}| \approx 10 T_{1x}$ and $T_{beat,y} = 2\pi/|\omega_{1y} - \omega_{ey}| \approx 10 T_{1y}$.

6.2. 3-D Governing Equations

The velocity potential $\varphi(x, y, z, t)$ satisfies the three-dimensional Laplace equation:

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = 0, \quad 0 < x < B, \quad 0 < y < W, \quad -d_0 < z < 0. \quad (57)$$

The boundary conditions are extended as follows:

- **KBBC** ($z = -d_0$): $\partial\varphi/\partial z = 0$.
- **KWBC** (x-walls): $\partial\varphi/\partial x = 0$ at $x = 0$ and $x = B$.
- **KWBC** (y-walls): $\partial\varphi/\partial y = 0$ at $y = 0$ and $y = W$.
- **KFSBC** ($z = 0$): $\partial\varphi/\partial z = \partial\eta/\partial t$.
- **DFSBC** ($z = 0$): $\partial\varphi/\partial t + g\eta + a_x(t)x + a_y(t)y = 0$.
- **IC** ($t = 0$): $\varphi = 0, \partial\varphi/\partial t = 0$.

The DFSBC now includes a y -direction forcing term $a_y(t)y$, and the new KWBC on the y -walls is the only structurally new equation compared to the 2-D case.

6.3. 3-D Analytical Solution

Under linearised potential flow, the x - and y -excitations decouple into independent modal families. The x -excitation drives $(m, 0)$ modes (odd m) and the y -excitation drives $(0, n)$ modes (odd n):

$$\varphi = \varphi_{0x}(t) + \sum_{m \text{ odd}} A_m(t) \cos(k_{mx}x) \frac{\cosh[k_{mx}(d_0 + z)]}{\cosh(k_{mx}d_0)} + \varphi_{0y}(t) + \sum_{n \text{ odd}} B_n(t) \cos(k_{ny}y) \frac{\cosh[k_{ny}(d_0 + z)]}{\cosh(k_{ny}d_0)}, \quad (58)$$

where $A_m(t) = C_m^x [\cos(\omega_{ex}t) - \cos(\omega_m t)]$ and $B_n(t) = C_n^y [\cos(\omega_{ey}t) - \cos(\omega_n t)]$ follow the same forced-oscillator structure as the 2-D case. The free-surface elevation is:

$$\eta(x, y, t) = -\frac{1}{g} \left[\sum_{m \text{ odd}} A'_m(t) \cos(k_{mx}x) + a_x(t) \left(x - \frac{B}{2} \right) + \sum_{n \text{ odd}} B'_n(t) \cos(k_{ny}y) + a_y(t) \left(y - \frac{W}{2} \right) \right]. \quad (59)$$

The velocity components are $u = \partial\varphi/\partial x$, $v = \partial\varphi/\partial y$, $w = \partial\varphi/\partial z$. Numerical verification confirms that this solution satisfies all governing equations and boundary conditions to within finite-difference truncation error ($\sim 10^{-5}$). Importantly, the y -modes contribute approximately 42% of the velocity potential at the tank corners, confirming that the 3-D problem is not a trivial extrusion of the 2-D case.

6.4. 3-D Network Architecture

The 3-D network retains the same MLP structure (16 hidden layers, 64 neurons, tanh) with the following modifications:

Hard wall BC in both directions. The spatial inputs are embedded as $x_n = \cos(\pi x/B)$ and $y_n = \cos(\pi y/W)$. By the chain rule:

$$\frac{\partial \hat{\varphi}}{\partial x} = \frac{\partial f_\theta}{\partial x_n} \cdot \left(-\frac{\pi}{B} \sin \frac{\pi x}{B} \right), \quad \frac{\partial \hat{\varphi}}{\partial y} = \frac{\partial f_\theta}{\partial y_n} \cdot \left(-\frac{\pi}{W} \sin \frac{\pi y}{W} \right), \quad (60)$$

which vanish identically at $x \in \{0, B\}$ and $y \in \{0, W\}$ respectively. This eliminates all wall collocation points in both horizontal directions.

Extended Fourier embedding. The input is augmented from 9 to 16 dimensions to accommodate six physical frequencies:

$$\gamma_{3D}(x) = \left(\cos \frac{\pi x}{B}, \cos \frac{\pi y}{W}, \frac{z+d_0}{d_0}, \frac{t}{T_0}, \sin \omega_{1x} t, \cos \omega_{1x} t, \sin \omega_{ex} t, \cos \omega_{ex} t, \sin \Delta \omega_x t, \cos \Delta \omega_x t, \sin \omega_{1y} t, \cos \omega_{1y} t, \sin \omega_{ey} t, \cos \omega_{ey} t, \sin \Delta \omega_y t, \cos \Delta \omega_y t \right). \quad (61)$$

where $\Delta\omega_x = \omega_{1x} - \omega_{ex}$ and $\Delta\omega_y = \omega_{1y} - \omega_{ey}$. The additional 7 input features add only $(16 - 9) \times 64 = 448$ parameters ($< 1\%$ of total). Table 5 compares the 2-D and 3-D architectures.

Table 5. Network architecture comparison: 2-D vs. 3-D.

Parameter	2-D	3-D
Input dimension	9	16
Output	1 ($\hat{\phi}$)	1 ($\hat{\phi}$)
Hidden layers	16	16
Neurons/layer	64	64
Physical frequencies	3 ($\omega_1, \omega_e, \Delta\omega$)	6 ($\omega_{1x}, \omega_{ex}, \Delta\omega_x, \omega_{1y}, \omega_{ey}, \Delta\omega_y$)
Hard wall dims	x only	x and y
Parameters	$\sim 63,100$	$\sim 63,550$
Velocity outputs	u, w	u, v, w

6.5. 3-D Loss Function

The loss function retains the same structure and weights as the 2-D case, with the following modifications:

- **3-D Laplace:** $r_{GE} = \hat{\phi}_{xx} + \hat{\phi}_{yy} + \hat{\phi}_{zz}$.
- **3-D gradient-enhanced Laplace:** $r_{GL} = (\partial_x \nabla^2 \hat{\phi})^2 + (\partial_y \nabla^2 \hat{\phi})^2 + (\partial_z \nabla^2 \hat{\phi})^2$.
- **DFSBC:** $r_{DFSBC} = \partial_t \hat{\phi} + g\eta + a_x(t)x + a_y(t)y$ at $z = 0$.

The loss weights ($\lambda_D = \lambda_K = 100$, $\lambda_{GL} = 0.1$, $\lambda_I = 10$) and training strategy (8,000 Adam + 2,000 L-BFGS) are unchanged from the 2-D case.

6.6. 3-D Collocation Grid

A structured grid of $N_x \times N_y \times N_z \times N_t$ interior points is used, with the same vertical clustering ($z = -d_0(1 - \zeta^{0.6})$) and early-time densification as the 2-D case. The default 3-D grid uses $N_x = 21$, $N_y = 11$, $N_z = 21$, $N_t = 101$, yielding approximately 490,000 interior points per window. The 15% mini-batch strategy keeps per-step GPU memory comparable to the 2-D case.

6.7. 3-D Results

The 3-D PINN was trained on a single NVIDIA RTX 5090 GPU. Training for one beating cycle took 2,499 s (≈ 42 min), comparable to the 2-D training time of $\sim 1,920$ s on an A100.

6.7.1. Wave Elevation

Figure 11 shows the wave elevation time history at two tank corners: $(x, y) = (0, 0)$ and $(x, y) = (B, 0)$, where both x - and y -modal families contribute. The PINN tracks the analytical solution over one beating cycle with $\varepsilon_\eta = 0.24\%$.

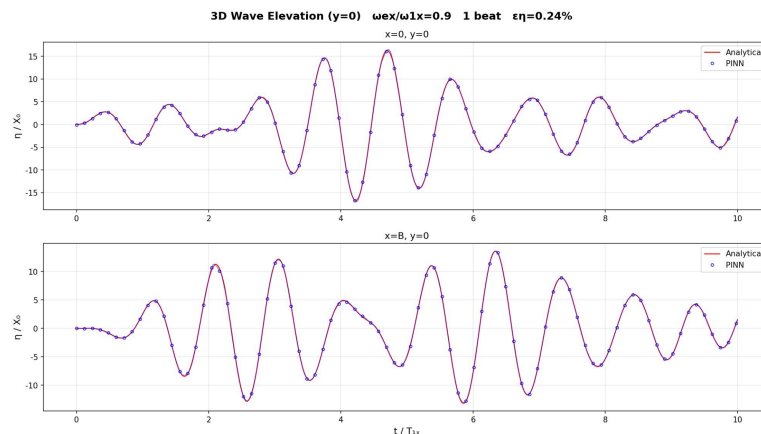


Figure 11. 3-D sloshing: wave elevation at the tank corner $(x=0, y=0)$ and $(x=B, y=0)$ over one beating cycle. Lines: analytical. Markers: PINN.

6.7.2. Free-Surface Pattern

Figure 12 shows the three-dimensional free-surface elevation $\eta(x, y)$ at four time instants spanning one beating cycle ($t/T_{\text{beat}} = 0.10, 0.25, 0.40, 0.50$). The analytical solution is rendered as a coloured wireframe mesh, while the PINN predictions are plotted as scatter points at the same grid nodes. The points sit precisely on the mesh surface at all four instants, confirming sub-percent accuracy. The bi-directional excitation produces a genuinely two-dimensional wave pattern: the free surface tilts in both x and y simultaneously, with peak amplitudes of ~ 12 mm at the corners growing from ~ 5 mm at $t = 0.10 T_{\text{beat}}$ to the maximum at $t = 0.25 T_{\text{beat}}$, then decreasing as the beating envelope passes through zero.

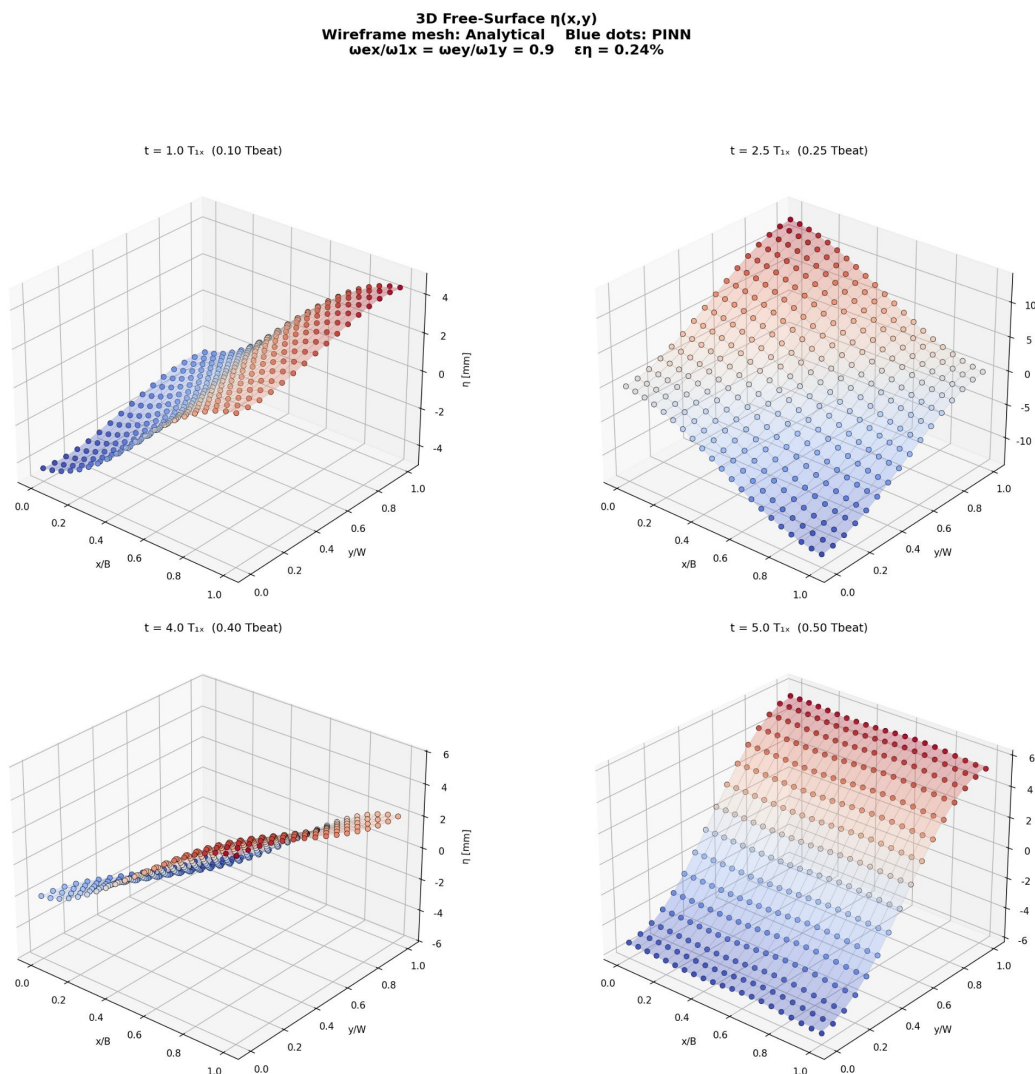


Figure 12. 3-D free-surface elevation $\eta(x, y)$ at four time instants. Wireframe mesh: analytical solution. Coloured dots: PINN prediction at the same grid nodes. The PINN points lie on the analytical surface at all times ($\epsilon_\eta = 0.24\%$).

6.7.3. Velocity Fields

Figure 13 compares the velocity fields at the $y = W/2$ mid-plane at $t = T_{\text{beat}}/4$. All three velocity components (u, v, w) are shown. The transverse velocity $v = \partial\phi/\partial y$ is a purely 3-D quantity with no 2-D counterpart; at $y = W/2$ the transverse modes have $\sin(n\pi y/W) = \pm 1$ (odd n), so v reaches its maximum magnitude at this plane, providing a stringent test of the network's ability to learn the y -modal contributions.

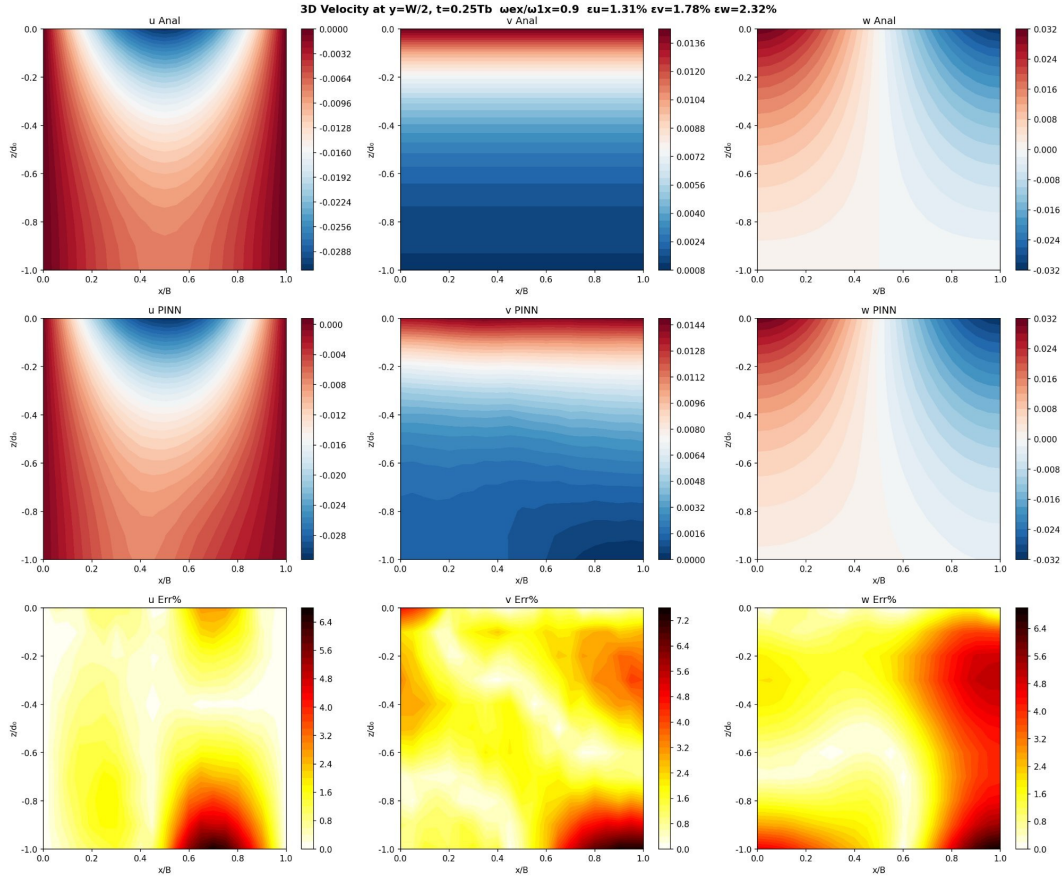


Figure 13. 3-D velocity field at $y = W/2$, $t = T_{\text{beat}}/4$. Top row: analytical. Middle row: PINN. Bottom row: pointwise error. Left to right: u , v , w .

6.7.4. Error Metrics

Table 6 summarises the 3-D PINN accuracy over one beating cycle. The errors are computed over the full 3-D volume, not on a single slice.

Table 6. 3-D PINN error metrics over one beating cycle (full-volume NMAE).

	ε_η [%]	ε_u [%]	ε_v [%]	ε_w [%]
3-D (1 beat)	0.24	1.31	1.78	2.32

Comparing with the 2-D results ($\varepsilon_\eta = 0.15\%$, $\varepsilon_u = 0.84\%$, $\varepsilon_w = 1.65\%$), the 3-D errors are expected to be moderately higher due to the increased input dimensionality and the additional PDE terms. The transverse velocity error ε_v is reported for the first time and reflects the network's ability to learn the y -modal family.

7. Conclusions

This paper has applied PINNs to near-resonant excited sloshing with a beating response in both two-dimensional and three-dimensional rectangular tanks. For the 2-D case, two training obstacles specific to forced sloshing are identified and addressed. Residual normalization and loss weighting ($\lambda_D = \lambda_K = 100$) eliminate the zero-solution trap; Fourier time embedding at the three physical frequencies (ω_1 , ω_e , $\Delta\omega$) resolves spectral bias. A hard wall boundary condition via the $\cos(\pi x/B)$ embedding enforces the lateral wall condition exactly, and a gradient-enhanced Laplace regulariser ($\nabla(\nabla^2\varphi) = 0$) improves velocity accuracy. Together, these techniques yield $\varepsilon_\eta = 0.15\%$, $\varepsilon_u = 0.84\%$, and $\varepsilon_w = 1.65\%$ over one beating cycle, validated against the viscous finite-difference results of Chen & Nokes [4].

A frequency parameter study covers $\omega_e/\omega_1 = 0.5\text{--}1.1$, including the near-resonance regime. A time-domain decomposition strategy with transfer learning partitions the domain into one-beat windows. For the 3-beat case ($30 T_1$), decomposition reduces velocity errors from $\varepsilon_u = 15.79\%$ (single network) to 3.26%. Extending to 5 beats ($50 T_1$), overall accuracy is maintained at $\varepsilon_u = 3.43\%$ and $\varepsilon_\eta = 0.30\%$ with no monotonic error accumulation.

The extension to three dimensions demonstrates that the same techniques—Fourier embedding, hard wall boundary conditions, and gradient-enhanced regularisation—transfer effectively to higher-dimensional sloshing. The 3-D formulation introduces bi-directional excitation that excites both $(m, 0)$ and $(0, n)$ modal families, producing a genuinely three-dimensional free-surface response with y -modes contributing approximately 42% of the velocity potential at the tank corners. The 16-dimensional Fourier embedding accommodating six physical frequencies, combined with hard wall enforcement in both x and y directions, enables the network to capture the additional modal family with errors of $\varepsilon_\eta = 0.24\%$, $\varepsilon_u = 1.31\%$, $\varepsilon_v = 1.78\%$, and $\varepsilon_w = 2.32\%$ over one beating cycle.

Future work will address nonlinear large-amplitude sloshing, complex tank geometries, broadband excitation spectra, and the coupling of time-domain decomposition with the 3-D formulation for long-time multi-beat simulations.

Author Contributions: Conceptualization, Z.L.; methodology, Z.L.; software, Z.L.; validation, Z.L.; formal analysis, Z.L.; investigation, Z.L.; writing—original draft preparation, Z.L.; writing—review and editing, Z.L.; visualization, Z.L. The author has read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The implementation uses PyTorch with automatic differentiation. The 2-D simulations were executed on an NVIDIA A100 GPU provided by Alibaba Cloud; the 3-D extension was trained on an NVIDIA GeForce RTX 5090.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Faltinsen, O.M.; Timokha, A.N. *Sloshing*; Cambridge University Press: Cambridge, UK, 2009.
2. Ibrahim, R.A. *Liquid Sloshing Dynamics: Theory and Applications*; Cambridge University Press: Cambridge, UK, 2005.
3. Wu, G.X.; Ma, Q.W.; Eatock Taylor, R. Numerical simulation of sloshing waves in a 3D tank based on a finite element method. *Appl. Ocean Res.* **1998**, *20*, 337–355.
4. Chen, B.-F.; Nokes, R. Time-independent finite difference analysis of fully non-linear and viscous fluid sloshing in a rectangular tank. *J. Comput. Phys.* **2005**, *209*, 47–81.
5. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707.
6. Cai, S.; Mao, Z.; Wang, Z.; Yin, M.; Karniadakis, G.E. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mech. Sinica* **2021**, *37*, 1727–1738.
7. Cuomo, S.; Schiano Di Cola, V.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *J. Sci. Comput.* **2022**, *92*, 88.
8. Lu, L.; Meng, X.; Mao, Z.; Karniadakis, G.E. DeepXDE: A deep learning library for solving differential equations. *SIAM Rev.* **2021**, *63*, 208–228.
9. Wang, S.; Teng, Y.; Perdikaris, P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J. Sci. Comput.* **2021**, *43*, A3055–A3081.

10. Jagtap, A.D.; Kawaguchi, K.; Karniadakis, G.E. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *J. Comput. Phys.* **2020**, *404*, 109136.
11. Rahaman, N.; Baratin, A.; Arpit, D.; Draxler, F.; Lin, M.; Hamprecht, F.; Bengio, Y.; Courville, A. On the spectral bias of neural networks. In *Proc. ICML 2019*; PMLR: Long Beach, CA, USA, 2019; pp. 5301–5310.
12. Tancik, M.; Srinivasan, P.P.; Mildenhall, B.; Fridovich-Keil, S.; Ramamoorthi, R.; Barron, J.T.; Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. In *Proc. NeurIPS 2020*; Vol. 33; pp. 7537–7547.
13. Wang, S.; Wang, H.; Perdikaris, P. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Comput. Methods Appl. Mech. Eng.* **2021**, *384*, 113938.
14. Wang, S.; Sankaran, S.; Wang, H.; Perdikaris, P. An expert's guide to training physics-informed neural networks. *J. Mach. Learn. Res.* **2024**, *25*, 1–51.
15. Sheikholeslami, M.; Salehi, S.; Mao, W.; Eslamdoost, A.; Nilsson, H. Physics-informed neural networks with hard and soft boundary conditions for linear free surface waves. *Phys. Fluids* **2025**, *37*, 087158.
16. Lu, H.; Wang, Q.; Tang, W.; Liu, H. Physics-informed neural networks for fully non-linear free surface wave propagation. *Phys. Fluids* **2024**, *36*, 062106.
17. Chen, L.; Li, B.; Luo, C.; et al. WaveNets: Physics-informed neural networks for full-field recovery of rotational flow beneath large-amplitude periodic water waves. *Eng. Comput.* **2024**, *40*, 2819–2839.
18. Ehlers, S.; Hoffmann, N.; Tang, T.; Callaghan, A.H.; Cao, R.; Padilla, E.M.; Fang, Y.; Stender, M. Physics-informed neural networks for phase-resolved data assimilation and prediction of nonlinear ocean waves. *Phys. Rev. Fluids* **2025**, *10*, 094901.
19. Wessels, H.; Weißenfels, C.; Wriggers, P. The neural particle method—An updated Lagrangian physics informed neural network for computational fluid dynamics. *Comput. Methods Appl. Mech. Eng.* **2020**, *368*, 113127.
20. Huang, C.; Liu, C.; Dettmer, W. Solving free-surface problems for non-shallow water using boundary and initial conditions-free physics-informed neural network (bif-PINN). *J. Comput. Phys.* **2023**, *479*, 112003.
21. Shao, K.; Wu, Y.; Jia, S. An improved neural particle method for complex free surface flow simulation using physics-informed neural networks. *Mathematics* **2023**, *11*, 1805.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.