Rigorously Certified Neural-Network Computation of Nonlinear Water Waves at Near-Machine Precision

Michal Shavit[1*], Yongji Wang[1] and Tristan Buckmaster[1]

[1*]Courant Institute of Mathematical Sciences, New York University, NY, 10012, USA.

*Corresponding author(s). E-mail(s): ms14479@nyu.edu;
Contributing authors: iiauthor@gmail.com; iiiauthor@gmail.com;

Abstract

We develop a computational framework that combines machine-learning-based discovery with rigorous computer-assisted numerical certification and formal verification for strongly nonlinear partial differential equations. We demonstrate the approach for periodic gravity–capillary traveling water waves. Physics-informed neural networks are used as a search mechanism, with the wave profile, wave speed and, in continuation experiments, surface tension learned directly from the governing equations without training data. A single formulation recovers three classical high-amplitude regimes: pure-gravity Stokes waves, pure-capillary Crapper waves and gravity–capillary Wilton ripples. Starting from a high-amplitude Wilton ripple, continuation in surface tension further reveals new high-amplitude candidate solutions. We freeze one such machine-learned candidate as an explicit finite representation and certify its residual using interval arithmetic, rigorously accounting for floating-point roundoff, Fourier truncation, aliasing and grid-to-continuum errors. The resulting computation provides a rigorous uniform measure of how accurately the candidate satisfies the governing equation, reaching near-machine precision. We then formalize the mathematical structure of the certification argument in Lean, which machine-checks the deductions connecting the certified numerical bounds to the final residual theorem. Both the interval certification and formalization were developed with substantial assistance from a large language model, while the validity of the result remains independent of the language model: the numerical bounds are rigorously enclosed by interval arithmetic and the subsequent mathematical deductions are checked by the Lean kernel. Together, these steps provide a pipeline from neural discovery to rigorous numerical certification and machine-checked mathematical verification, enabling

machine learning to explore strongly nonlinear regimes while retaining explicit and auditable mathematical guarantees.

Keywords: keyword1, Keyword2, Keyword3, Keyword4

1 Introduction

Machine learning can produce remarkably accurate numerical solutions of nonlinear partial differential equations, but numerical accuracy alone does not provide a mathematical guarantee of correctness. Here we develop a computational framework that bridges this gap by combining machine-learning-based discovery with rigorous numerical certification and formal verification. We demonstrate the approach for strongly nonlinear, high-amplitude water waves, where physics-informed neural networks discover candidate solutions with residuals approaching the limits of double-precision arithmetic. We then convert a selected candidate into a rigorous uniform residual certificate using interval arithmetic and formally verify the mathematical deductions underlying the certificate in Lean. The resulting pipeline provides a route from machine-learned numerical discovery to certified, machine-checkable scientific computation.

Surface gravity–capillary waves constitute a classical and enduring problem of fluid mechanics. Their governing equations are remarkably rich, describing phenomena ranging from small-amplitude capillary ripples to extreme ocean waves, yet are notoriously difficult to analyze because of their full nonlinearity and nonlocality. The subject has a long history spanning oceanography, mathematical physics and rigorous analysis, from Stokes' expansions and Zakharov's Hamiltonian formulation to modern results on well-posedness and asymptotics [1]; see [2] for a review. Despite this progress, the strongly nonlinear, high-amplitude regime remains substantially more difficult to access. In this regime, perturbative descriptions cease to apply, solution branches can develop intricate structure, and even the systematic construction and accurate numerical representation of nonlinear wave solutions becomes challenging.

This combination of strong nonlinearity, nonlocality and a rich solution structure makes water waves a stringent setting in which to test whether machine learning can move beyond numerical approximation and become a tool for mathematical discovery with rigorously certified performance. We address this question through three stages: neural discovery, computer-assisted certification by interval arithmetic, and formal verification.

First, we employ physics-informed neural networks (PINNs) [3] as a search mechanism for high-amplitude two-dimensional periodic traveling solutions of the fully nonlinear water-wave equations. The network is optimized directly to satisfy the equations in conformal variables, with the wave profile, wave speed and, in selected continuation experiments, the surface-tension parameter learned simultaneously. In this way, the network parametrizes and explores nonlinear solution branches. Building on the two-stage training procedure of ref. [4], we obtain neural representations whose

residuals approach the limits of double-precision arithmetic. Using the same formulation, we recover three canonical high-amplitude regimes: pure-gravity Stokes waves [5], pure-capillary Crapper waves [] and mixed gravity–capillary Wilton ripples. Starting from a high-amplitude Wilton ripple and continuing in surface tension, the network further discovers new high-amplitude candidate solutions of the governing equation.

Second, we turn numerical accuracy into a rigorous statement using interval arithmetic. A small residual obtained by floating-point computation does not itself constitute a rigorous certificate, because roundoff, truncation, aliasing and discretization errors enter through the nonlinear and nonlocal operations in the water-wave equations. We therefore freeze a selected machine-learned candidate as an explicit finite representation and propagate rigorous intervals through the governing operators, explicitly accounting for floating-point roundoff, Fourier truncation, aliasing errors arising from grid-based evaluation, and the error incurred in passing from a finite grid to the continuous domain. This yields a certified uniform upper bound on the residual at near-machine precision, rather than a small residual observed only at finitely many numerical evaluation points.

Third, we formally verify the mathematical structure of the certification argument in Lean, a proof assistant and functional programming language. At this level of accuracy, the numerical certificate is itself necessarily computer assisted: our interval-arithmetic proof is implemented in Julia using the Arb library and consists of hundreds of lines of code that propagate rigorous bounds through nonlinear and nonlocal operations. Although the resulting computation is mathematically rigorous, its implementation is too extensive and delicate to be audited line by line through conventional peer review. We therefore use Lean to provide a machine-checkable logical layer for the certification argument. Lean verifies the mathematical identities and deductions connecting the certified numerical bounds to the final residual theorem, substantially reducing the amount of computational reasoning that must be manually audited. The numerical interval bounds constitute the numerical trust boundary, while the subsequent mathematical deductions are checked by the Lean kernel.

Both the interval-arithmetic certification and its formalization were developed with substantial assistance from a large language model. Crucially, however, the validity of the resulting certificate does not depend on the reliability of the language model: the numerical inequalities are rigorously enclosed by interval arithmetic, while the mathematical deductions built upon them are checked by the Lean kernel. This separation allows a large language model to assist substantially in constructing a complex computer-assisted proof without becoming part of the trusted mathematical foundation of the result.

Together, these developments establish a pipeline from machine-learning-based discovery to rigorous numerical certification and machine-checked mathematical verification. Pairs of these ingredients have previously been combined—including neural discovery with computer-assisted proof and rigorous numerics with formal verification—but, to our knowledge, the complete chain has not previously been closed from a solution discovered by a neural network, through rigorous interval certification, to formal verification of the resulting certificate in a proof assistant. The framework therefore provides a route for using machine learning not only as a scientific-computing

tool, but also as an instrument of mathematical discovery: exploring strongly nonlinear regimes that are difficult to access analytically while retaining explicit, rigorous and auditable mathematical guarantees.

2 Results

Surface gravity-capillary waves are described in terms of the Euler equation for an incompressible, irrotational fluid of infinite depth with potential velocity $\mathbf{v} = \nabla\phi$ and a free surface $y = \eta(x,t)$, where x is the horizontal coordinate and y is the vertical coordinate. Incompressibility gives the Laplace equation $\Delta\phi = 0$ inside the fluid domain $\mathcal{D} = \{(x,y),\ -\infty < y \leq \eta(x,t)\}$ with the boundary conditions at the surface $y = \eta(x,t)$:

$$(\partial_t + \mathbf{v}\cdot\nabla)(y - \eta(x,t)) = 0, \tag{1}$$

$$\phi_t + \tfrac{1}{2}|\nabla\phi|^2 + g\eta = \kappa\frac{\eta_{xx}}{(1+\eta_x^2)^{3/2}}, \tag{2}$$

and $\nabla\Phi \to 0$ as $y \to -\infty$. Here g is the acceleration of gravity, κ is the surface tension coefficient and

$$\frac{\eta_{xx}}{(1+\eta_x^2)^{3/2}} = \partial_x\left(\frac{\eta_x}{\sqrt{1+\eta_x^2}}\right) \tag{3}$$

is the mean curvature of the free surface. The first boundary condition is the kinematic condition, expressing that the free surface moves with the fluid particles. The second is Bernoulli's condition, in which the pressure jump across the free surface is proportional to the mean curvature.

For periodic waves traveling in one direction, one can reformulate the equations above in terms of a conformal map $z(w,t) := x(w,t) + iy(w,t)$ that maps the lower half-plane $\{(u,v) \in [-\pi,\pi]\times(-\infty,0]\}$ to the fluid domain $\mathcal{D}$ as illustrated in Fig. 1. This approach is described in [6–8]. On $v = 0$, steady traveling waves $y = y(u - ct)$ solve the Babenko equation

$$\begin{aligned}\mathcal{B}(y) = -\mu\hat{H}y_u + \Big(y + y\,\hat{H}y_u + \hat{H}(yy_u)\Big) \\ -\beta\,\partial_u\left(\frac{y_u}{\sqrt{y_u^2 + (1+\hat{H}y_u)^2}}\right) + \beta\,\hat{H}\partial_u\left(\frac{1+\hat{H}y_u}{\sqrt{y_u^2+(1+\hat{H}y_u)^2}}\right) = 0,\end{aligned} \tag{4}$$

where μ is the squared (rescaled) wave speed, $\beta = \kappa/(g\lambda^2)$ is the Bond number encoding surface tension, gravity, and wavelength λ. $y(u)$ is the elevation of the free-surface in conformal coordinates and $\hat{H}$ is the periodic Hilbert transform defined on $L_2[-\pi,\pi]$ by linearity from the relation

$$\hat{H}e^{inu} = -i\,\mathrm{sign}(n)e^{inu}. \tag{5}$$

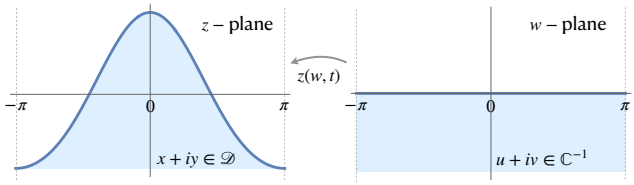

Fig. 1 The conformal mapping from the region in the w plane $(u, v) \in [-\pi, \pi] \times (-\infty, 0]$ to the domain occupied by the fluid in the (x, y) plane, with $(x, y) \in [-\pi, \pi] \times (-\infty, \eta(x, t)]$. The line $v = 0$ (blue line) is mapped onto the free surface $\eta(x, t)$.

The Babenko equation (4) admits two families of known high amplitude solutions at the extreme ends of zero gravity or zero capillarity. At $\beta = 0$, the one parameter family of waves is the classical Stokes waves represented as a curve in the speed and amplitude plane (c, a) connecting small amplitude solutions with a limiting Stokes wave of extreme height with a $2\pi/3$ crest angle [5]. The amplitude is defined as the difference between the maximal and minimal height divided by the period

$$a = \frac{\max y - \min y}{2\pi}. \tag{6}$$

At zero gravity, $g = 0$, there is a family of pure capillary solutions, found by Crapper, which is represented as a curve in the speed and amplitude plane (c, a) connecting small amplitude capillary waves with a limiting Crapper wave of extreme height that encloses a bubble.

When both gravity and capillarity are present, resonant families of solutions, known as Wilton ripples, arise at special values of the surface tension. At the linear level, a Wilton ripple contains two harmonics, $\cos(mu)$ and $\cos(nu)$, whose phase speeds coincide. Although small-amplitude Wilton ripples have been computed numerically, their high-amplitude regime is much less understood both analytically and numerically.

2.1 Neural discovery and recovery of high-amplitude water waves

We use physics-informed neural networks (PINNs) [3] to discover water-wave profiles and their associated physical parameters directly from the Babenko equation (4). Rather than being trained on precomputed or experimental data, the network is optimized against the governing equation together with the physical and normalization constraints defining the solution family under consideration. The same neural architecture and residual-based optimization framework is used throughout, while the physical parameters and auxiliary constraints are adapted to select the relevant branch. The trained profile is denoted by y_N. For each run, we minimize a weighted squared residual of the Babenko equation and, when used, its differentiated form, together with auxiliary conditions controlling quantities such as vertical translation, amplitude and symmetry. The wave speed $c_N = \sqrt{\mu_N}$ is optimized simultaneously

with the profile. Depending on the experiment, gravity and surface tension are fixed to select a particular physical regime, while in the continuation experiments the surface-tension parameter is itself allowed to vary. Details of the network architecture, Hilbert-transform evaluation, loss function, constraints and optimization procedure are given in Methods.

We first use three canonical high-amplitude solution families as benchmarks for the discovery framework (Fig. 2). These regimes probe qualitatively different balances in the same nonlinear equation. In the zero-surface-tension limit, $\beta = 0$, the network reproduces the classical pure-gravity Stokes branch. In the zero-gravity limit, $g = 0$, it recovers the pure-capillary Crapper family. When both gravity and surface tension are present, the network recovers high-amplitude $(1, 2)$ Wilton ripples, for which the first and second Fourier harmonics travel at the same speed at the linear level. The three families have markedly different geometries, dominant physical balances and Fourier structures, and their recovery therefore provides complementary tests of the same computational framework. Importantly, the appropriate physical and normalization constraints differ between these regimes; what remains common is the neural representation and equation-based optimization.

The Wilton regime provides a particularly demanding test. At small amplitude, the $(1, 2)$ Wilton branch originates from a linear resonance between two Fourier modes, but at finite amplitude their contributions are coupled by the full nonlinear water-wave equation. The amplitudes of the resonant components therefore cannot in general be varied independently: nonlinear constraints determine how the harmonics combine along the solution branch. Continuing a high-amplitude Wilton ripple is consequently not equivalent to simply varying the amplitude parameter and reoptimizing an unconstrained profile. The continuation must simultaneously preserve the appropriate nonlinear constraints while allowing the profile and wave speed parameter to adjust, making the high-amplitude resonant regime a substantially more restrictive search problem.

After reproducing a high-amplitude $(1, 2)$ Wilton ripple, we use it as a seed for neural continuation. We allow the surface-tension parameter to vary together with the wave profile and speed, while imposing the amplitude and normalization constraints required to control motion through the solution space. Starting from the same high-amplitude seed, continuation in different constrained directions generates new nearby gravity–capillary profiles with small Babenko residuals (Fig. **??**). In this setting, the PINN acts as a nonlinear continuation and discovery method: it searches for profiles satisfying the governing equation.

These benchmark and continuation experiments separate two roles of the neural computation. Recovery of the Stokes, Crapper and Wilton families tests whether the framework can represent and optimize solutions across distinct nonlinear regimes when supplied with the appropriate physical constraints. Continuation away from the high-amplitude Wilton state then uses the same machinery in a discovery setting, where the resulting profiles are not prescribed in advance. The low-residual candidates obtained in this stage provide the starting point for the rigorous certification described below.

2.1.1 Continuation experiments and new solutions

After reproducing a high-amplitude $(1, 2)$ Wilton ripple, we use it as a seed for neural continuation. We allow the surface-tension parameter to vary together with the wave profile and speed, while imposing constraints on the wave amplitude and on the ratio a_1/a_2 of the two resonant Fourier components to control motion through the solution space. Starting from the same high-amplitude seed, continuation in different constrained directions produces new gravity–capillary profiles with small Babenko residuals (Fig. **??**). Although these solutions are reached by continuation from a Wilton ripple, they are not simply nearby members of the known high-amplitude Wilton family: their Fourier composition and values of β depart substantially from those of the corresponding Wilton branch, and their geometry develops qualitatively different features as the continuation proceeds. Toward one end of the computed branch, the crest develops a pronounced central depression while the trough becomes increasingly sharp, producing profiles distinct from the high-amplitude Wilton ripples used to initialize the search. To our knowledge, gravity–capillary solutions with this combination of Fourier structure, parameter values and free-surface geometry have not previously been computed.

We were also unable to continue this family toward the small-amplitude regime. Within the range accessible to our numerical continuation, the branch instead remains intrinsically finite amplitude. This observation suggests that these profiles may not arise through a conventional bifurcation from a linear water-wave mode, although establishing whether the family is genuinely disconnected from the small-amplitude spectrum requires further mathematical and numerical analysis. In this sense, the high-amplitude Wilton ripple provides an entry point into a region of the nonlinear solution landscape that is not apparent from the linearized problem.

A detailed mathematical characterization of this solution family—including its global bifurcation structure, possible limiting states, stability and connection, or lack thereof, to small-amplitude solutions—is beyond the scope of the present work. Here these solutions primarily demonstrate the exploratory capability of the computational framework: constrained neural continuation can identify previously uncharacterized high-amplitude states, which can subsequently be passed to rigorous numerical certification. Our focus in the remainder of the work is therefore on establishing this discovery-to-certification pipeline and on rigorously validating a representative machine-learned solution.

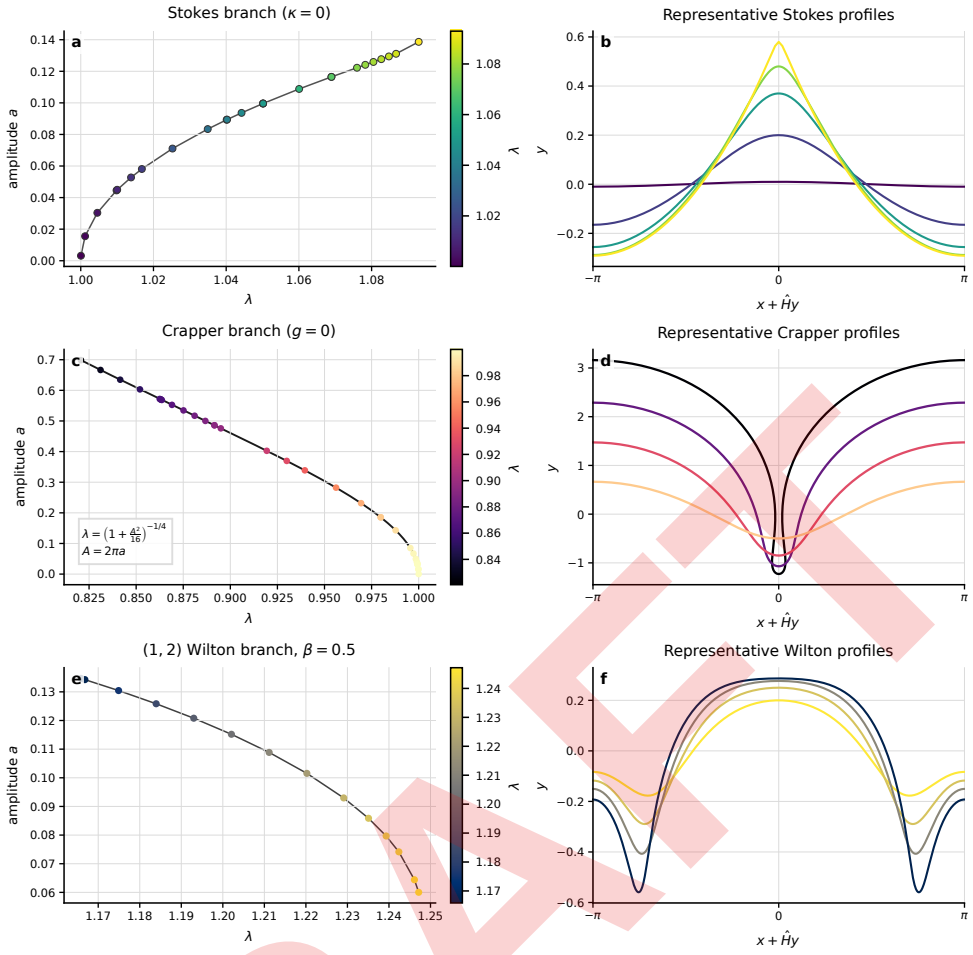

Fig. 2 Three high-amplitude regimes reproduced by the PINN. a, Pure-gravity Stokes branch with $\kappa = 0$, plotted in the (λ, a) plane, where $a = (\max y - \min y)/(2\pi)$. The panel shows the selected Stokes files, including the low-amplitude main-branch runs, that satisfy $\max|\mathcal{B}(y_N)| \leq 10^{-4}$, with the grey curve joining the retained runs. **b,** Five representative retained Stokes profiles plotted over $[-\pi, \pi]$ as $x + \hat{H}y$ versus y. **c,** Pure-capillary Crapper branch with $g = 0$. All Crapper files in the plotting folder are included as points on the branch, and the black curve shows the fit $\lambda = (1+A^2/16)^{-1/4}$ with $A = 2\pi a$. **d,** Representative Crapper profiles plotted over $[-\pi, \pi]$. **e,** Reproduced $(1, 2)$ Wilton branch at $\beta = 0.5$, again with all available branch files included. **f,** Four representative Wilton free-surface profiles, including low- and high-amplitude cases, plotted over $[-\pi, \pi]$.

After reproducing the $(1, 2)$ Wilton ripple, we use it as a seed for discovery. Rather than fixing the surface tension, we train the surface-tension parameter together with the profile and speed, initializing near the high-amplitude Wilton solution. This procedure reveals nearby families of gravity–capillary waves that deform the Wilton profile

while preserving small Babenko residuals. These families form the main new solution set studied below and provide the candidates passed to the interval-arithmetic certification step.

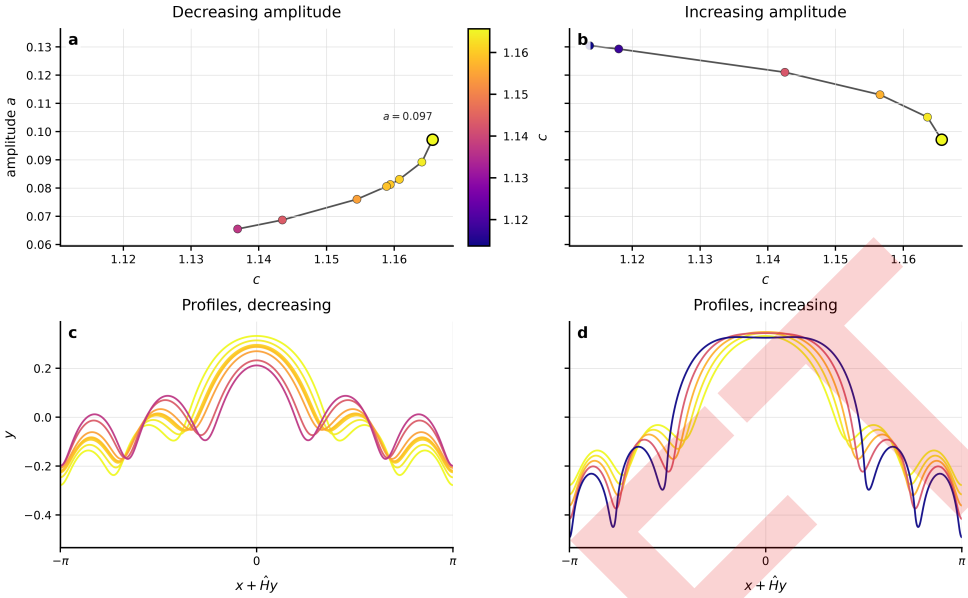

Fig. 3 Continuation away from the high-amplitude $(1, 2)$ **Wilton ripple. a,** Amplitude-decreasing continuation at $\kappa = 0.48$, starting from the shared seed with $0.09 < a < 0.1$ and plotted in the (c, a) plane. **b,** Amplitude-increasing continuation from the same seed point. **c,** Clean profiles along the amplitude-decreasing branch. **d,** Clean profiles along the amplitude-increasing branch, with colours in both profile panels matched to the same c scale.

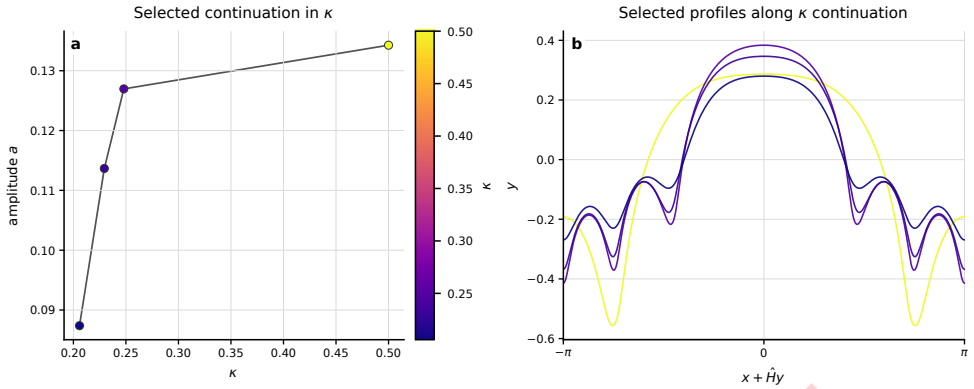

Fig. 4 Continuation near a high-amplitude Wilton ripple. **a**, Selected points on the continuation branch, shown in the amplitude–κ plane. For the three non-reference points, the stored continuation parameter is plotted as $\kappa = \beta^2$; for example, the point generated with $\beta = 0.498$ is shown at $\kappa = 0.498^2$. **b**, Corresponding free-surface profiles over one period, plotted against the conformal horizontal coordinate $x + \hat{H}y$ and colored by the same value of κ as in **a**. **c**, Certified profile at $\kappa = 0.498^2$. **d**, Pointwise residual $|\mathcal{B}(y_N)|$ for the certified profile. The dashed line marks the reported certificate level $\|\mathcal{B}(y_N)\|_\infty = 8 \times 10^{-13}$.

2.2 Second stage training and machine prescision candidate

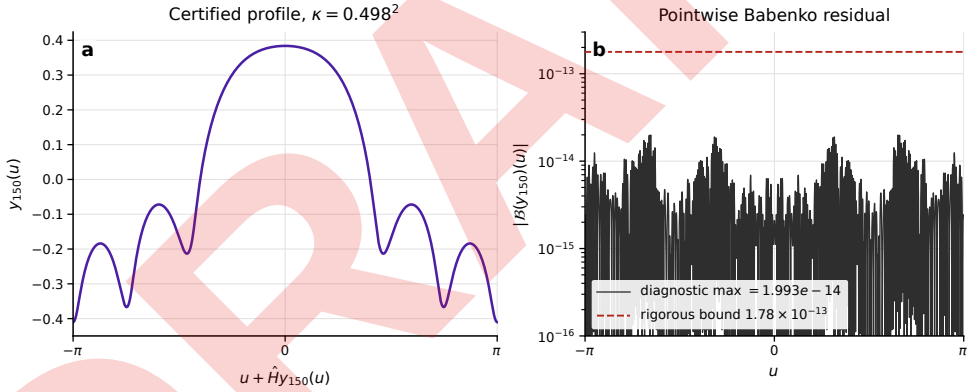

Fig. 5 Certified high-amplitude Wilton-ripple profile. **a**, Free-surface profile for the certified solution at $\beta = 0.498^2$, plotted over one period against the conformal horizontal coordinate $u + \hat{H}y$. **b**, Pointwise residual $|\mathcal{B}(y_N)|$ for the same solution. The dashed line marks the certificate level $\|\mathcal{B}(y_N)\|_\infty \le 1.78 \times 10^{-13}$ with $N = 150$.

2.3 Certified residual estimates and formal verification

We now turn from numerical discovery to rigorous certification. The neural-network approximation is used to construct a candidate profile; the object ultimately certified is an explicit finite trigonometric polynomial, with fixed coefficients and parameters.

Let

$$y_N(u) = \sum_{k=0}^{N} a_k \cos(ku), \tag{7}$$

and let $\mu, \beta > 0$ be fixed rational parameters. We write

$$R_N(u) = \mathcal{B}_{\mu,\beta}(y_N)(u) \tag{8}$$

for the residual of the gravity–capillary Babenko equation. Our objective is to establish a rigorous uniform estimate

$$\|R_N\|_{L^\infty(\mathbb{R})} = \sup_{u\in\mathbb{R}} |R_N(u)| \le \varepsilon, \tag{9}$$

for an explicit constant ε. For the candidate certified here,

$$N = 150, \qquad \varepsilon = \frac{178}{10^{15}} = 1.78 \times 10^{-13}. \tag{10}$$

This is an absolute residual bound for the specified polynomial and frozen parameters, not an estimate of the distance to an exact solution.

The certification proceeds in two distinct stages. The first is a computer-assisted numerical proof based on interval arithmetic. The second is a formal verification in Lean 4 of the logic and algorithmic correctness of the interval-arithmetic proof. The trust boundary is therefore explicit: the numerical enclosures are computed by Julia through Arblib, which uses the well-audited FLINT/Arb library for rigorous real arithmetic, while Lean verifies that the certified algorithm, together with the recorded interval endpoints, implies the stated residual and denominator bounds. This separation of rigorous numerical enclosure from machine-checked logical verification is standard in computer-assisted proofs that combine interval arithmetic with proof assistants. We now describe the two stages in turn.

This certification has both practical numerical and pure-mathematical motivations. From a numerical perspective, the objective is to establish the residual accuracy of a PINN-derived candidate through a rigorous bound, rather than through training loss, agreement with reference computations, or evaluation at finitely many points. Previous work has demonstrated numerical PINN residuals approaching the double-precision round-off floor [4], while other studies have developed a posteriori error estimates, continuous-domain residual certificates, interval-arithmetic validation, and Lean-based verification [9–12]. Numerically observing a small residual and rigorously certifying a uniform bound at the same scale are, however, distinct achievements.

Here we establish an absolute uniform residual bound of 1.78×10^{-13} for an explicit PINN-derived profile of the gravity–capillary Babenko equation, in the normalization

used throughout this work. To our knowledge, this is the first rigorous continuous-domain residual certificate at this accuracy for a PINN-derived approximate solution of a nonlinear and nonlocal equation. The certification includes the numerical evaluation of the nonlinear nonlocal capillary terms, with rigorous control of Fourier truncation, aliasing, denominator positivity, and between-grid errors. It is complemented by Lean verification of the mathematical deductions from explicitly stated numerical assumptions.

From a pure-mathematical perspective, the objective is to make PINN-based discovery usable as a high-precision tool for studying highly nonlinear and nonlocal equations. The resulting computations can serve as quantitative inputs to further mathematical analysis: the candidate is explicitly defined, its residual is rigorously controlled, and the computational and logical trust boundaries are made explicit. In particular, the residual certificate supplies a first rigorous ingredient toward a computer-assisted proof of the existence of a nearby exact solution.

We now describe the two stages in turn.

2.3.1 Rigorous certification by interval arithmetic

The first stage establishes the residual bound (9) using interval ball arithmetic. All certification computations are performed in Julia using Arb, with rigorous enclosures and outward rounding, so that computed quantities are enclosed by intervals containing their exact values. In particular, the computation accounts for floating-point roundoff, Fourier truncation and aliasing errors in the evaluation of nonlinear terms, and the passage from grid-based estimates to bounds valid between grid points.

The Hilbert transform is treated differently depending on its argument. For the trigonometric polynomial y_N and its derivatives, the Hilbert transform acts exactly mode by mode and therefore introduces no spectral truncation error; numerical evaluation of the resulting finite expressions is enclosed by interval arithmetic. By contrast, nonlinear composite quantities appearing in the capillary term are not finite trigonometric polynomials. Their Hilbert transforms are evaluated through a discrete Fourier representation, with the contribution of unresolved modes and the associated aliasing errors bounded explicitly using estimates on the Fourier tail.

The capillary term also contains the denominator

$$D(u) = \sqrt{\big(\partial_u y_N(u)\big)^2 + \big(1 + \hat{H}\partial_u y_N(u)\big)^2}. \tag{11}$$

The interval computation simultaneously establishes a strictly positive lower bound for D, ensuring that all divisions appearing in the residual are well defined and rigorously controlled.

Finally, the residual is not certified merely at a finite collection of grid points. On each grid cell, a Taylor expansion with a rigorously bounded remainder propagates the interval estimates from the grid to the entire cell. Combining these bounds over one period, and using periodicity, yields a uniform estimate for every $u \in \mathbb{R}$.

Thus, the output of the interval-arithmetic stage is a pair of explicit inequalities

$$\|\mathcal{B}_{\mu,\beta}(y_N)\|_{L^\infty(\mathbb{R})} \le \varepsilon, \qquad D(u) \ge d_{\min} > 0 \quad \text{for all } u \in \mathbb{R}, \tag{12}$$

whose numerical justification is supplied by the validated computation. The bound includes the errors in evaluating the residual itself, rather than treating the numerical Hilbert transform or the sampled residual values as exact.

2.3.2 Formal verification in Lean

The second stage concerns the logical verification of the certificate. The purpose of Lean is not to repeat the interval-arithmetic computation or the floating-point training procedure. Instead, the numerical endpoints produced by Arb are represented as exact rational data, and the corresponding numerical certification statements are admitted as explicit assumptions. Lean verifies the mathematical deductions that connect these assumptions to the stated residual theorem.

In particular, the Babenko residual is defined formally, and the identities connecting the finite cosine representation, differentiation, the periodic Hilbert transform, and the numerical storage conventions are proved within Lean. Periodicity is then used to promote the certificate on a single period to a statement on all of $\mathbb{R}$. Under the stated numerical certification assumptions, the resulting theorem has the form

$$\forall u \in \mathbb{R}, \qquad |\mathcal{B}_{\mu,\beta}(y_N)(u)| \le \varepsilon, \tag{13}$$

with the exact rational tolerance given in (10).

The numerical inequalities supplied by the interval computation constitute the numerical trust boundary of the formal proof. As in standard computer-assisted proofs based on external interval-arithmetic computations, the validity of these numerical bounds rests on the validated-numerics software and its correct application. Lean does not independently recompute the bounds; rather, it checks that, once the numerical certification statements are admitted as assumptions, the claimed theorem follows through formally verified mathematical reasoning. Representing the endpoints as exact rationals prevents additional rounding ambiguity in the formal certificate, but does not itself verify the external numerical computation. This separation makes explicit the respective roles of the two components: interval arithmetic supplies the rigorous numerical estimates, while Lean checks the mathematical structure and logical deductions built upon them.

We emphasize that neither stage establishes the existence of an exact solution near y_N. The result certified here is the uniform residual estimate (9) for the explicit polynomial, not a solution-error bound for the polynomial or the neural network. Upgrading this estimate to the existence of a nearby exact solution would additionally require quantitative invertibility estimates for the appropriately constrained linearized Babenko operator and control of the nonlinear remainder, as part of a Newton–Kantorovich or related contraction argument. That step lies beyond the scope of the present work.

3 Discussion

The results presented here demonstrate a route from machine-learning-based exploration of a nonlinear PDE to a rigorously certified and machine-checkable computational statement. The role of the neural network is primarily one of discovery: it provides a flexible representation with which to search nonlinear solution spaces and identify high-accuracy candidate solutions. The subsequent certification is deliberately separated from the training procedure. Once a candidate is selected, it is frozen as an explicit finite representation, and the mathematical claim concerns this representation rather than the neural network, optimizer or training history. This separation allows the exploratory flexibility of machine learning to be combined with the reproducibility and explicit error control of validated numerical computation.

The water-wave problem illustrates both the potential and the limitations of this approach. The same computational framework accesses qualitatively different high-amplitude regimes, including pure-gravity, pure-capillary and resonant gravity–capillary waves, but the search is not free of problem-specific mathematical information. Different solution families require appropriate physical, normalization and symmetry constraints, and continuation of high-amplitude Wilton ripples requires additional control of the relative amplitudes of the resonant Fourier modes. The neural network therefore does not replace mathematical structure; rather, it provides a flexible computational representation within which such structure can be imposed while leaving the detailed nonlinear profile to be determined by the governing equation. Developing methods that can infer useful continuation coordinates and constraints directly from the learned solution landscape would reduce this dependence on prior analytical knowledge.

A second distinction concerns numerical accuracy and mathematical existence. The certificate established here is a rigorous uniform bound on the residual of an explicit approximate solution. It shows that the reported near-machine-precision residual is not an artifact of floating-point roundoff, Fourier truncation, aliasing or finite sampling. It does not, by itself, prove the existence of an exact water-wave solution nearby. Such a result would require quantitative control of the linearized Babenko operator and of the nonlinear remainder, for example through a Newton–Kantorovich or related contraction argument. The certified residual obtained here provides one of the quantitative inputs required for such an existence proof and establishes a natural interface between machine-learning-based discovery and computer-assisted analysis.

Formal verification addresses a different source of uncertainty. At the accuracy considered here, the interval-arithmetic certification is itself a substantial computation involving hundreds of lines of validated numerical code and a sequence of analytic estimates that would be difficult to audit completely by conventional inspection. Lean makes the mathematical identities and deductions connecting the numerical certificate to the final theorem machine-checkable. The present implementation nevertheless retains an explicit numerical trust boundary: Lean does not independently reproduce the Arb computation, and the validated numerical bounds supplied by the interval-arithmetic calculation remain trusted inputs to the formal layer. A more tightly integrated system in which validated numerical computations produce proof objects that can be checked directly by the proof assistant would further reduce this boundary.

An important feature of the workflow is that substantial parts of both the rigorous numerical implementation and the formalization were developed with assistance from a large language model. This illustrates a potentially broader role for generative AI in scientific computing: not only in producing numerical candidates, but also in helping construct the machinery required to validate them. The reliability of the final mathematical statement need not inherit the reliability of the generative model, provided that model-generated computations and arguments terminate in independently checkable certificates. In the present framework, interval arithmetic provides rigorous numerical enclosures and Lean checks the subsequent logical deductions.

More broadly, the approach is not specific to water waves or to PINNs. Its essential ingredients are a computational method capable of discovering accurate candidate solutions, an explicit representation to which validated numerics can be applied, rigorous control of discretization and continuum errors, and a formal layer capable of checking the deductions made from the resulting numerical bounds. These ingredients suggest a general architecture for trustworthy machine-learning-assisted scientific computing, particularly for nonlinear problems in which analytical descriptions are incomplete and numerical exploration plays an important role. Extending this architecture from residual certification to rigorous existence, stability and bifurcation results would provide a stronger connection between machine-assisted discovery and mathematical analysis.

4 Methods

4.1 PINN architecture

4.1.1 Neural representation

The unknown free surface is represented in conformal coordinates by a neural function of the periodic variable $u \in [-\pi, \pi]$. For the symmetric runs, periodicity and evenness are imposed by feeding the network the transformed coordinate $\cos u$. The implementation used for the plotted runs consists of three scalar fully connected multilayer perceptrons with tanh activations, initialized with Xavier-scaled truncated normal weights. Each scalar network has three hidden layers of width 30. The networks are combined as

$$y_N(u) = s\, N_0(\cos u), \qquad T_{1,N}(u) = s\, N_1(\cos u) \sin u, \qquad T_{2,N}(u) = s\, N_2(\cos u), \tag{14}$$

where s is the amplitude scale used in a given continuation step. This construction makes y_N and $T_{2,N}$ even and $T_{1,N}$ odd, matching the crest-symmetric geometry. The auxiliary variables $T_{1,N}$ and $T_{2,N}$ represent the two components of the normalized tangent/curvature expression,

$$T_1 = \frac{y_u}{\sqrt{y_u^2 + (1 + \hat{H} y_u)^2}}, \qquad T_2 = \frac{1 + \hat{H} y_u}{\sqrt{y_u^2 + (1 + \hat{H} y_u)^2}}, \tag{15}$$

so that the surface-tension term can be evaluated stably through first derivatives of the auxiliary outputs.

The wave speed is optimized together with the network weights through a bounded scalar parameterization

$$c_N = c_0 + \eta_c \tanh \rho_c, \tag{16}$$

where c_0 is the continuation seed and η_c is the allowed local search radius. In the Wilton continuation runs in which surface tension is trained, the same bounded parametrization is used for β_N. Fixed-regime benchmark runs set the corresponding physical parameter to its prescribed value.

4.1.2 Residual, Hilbert transform and loss

All derivatives of the neural output are obtained by automatic differentiation in JAX using double precision. The periodic Hilbert transform is evaluated numerically on a dense nonuniform grid of $15,000$ sample points. The singular integral is split into a local contribution around the evaluation point, a close-field correction and an outer contribution evaluated by Gaussian quadrature. The same Hilbert-transform routine is used for y_N, $y_{N,u}$, derivatives of y_N^2 and derivatives of the auxiliary variables entering the surface-tension term.

Training minimizes a weighted loss

$$\mathcal{L} = w_{\mathrm{eq}} \sum_{u_i \in \mathcal{C}} \omega_i \left\| \mathcal{R}_N(u_i) \right\|^2 + w_{\mathrm{der}} \sum_{v_j \in \mathcal{D}} \nu_j \left| \partial_u \mathcal{B}(y_N)(v_j) \right|^2 + \mathcal{L}_{\mathrm{aux}}. \tag{17}$$

Here $\mathcal{R}_N = (e_1, e_2, e_3)$ consists of the Babenko residual and the two algebraic residuals defining $T_{1,N}$ and $T_{2,N}$. The auxiliary loss $\mathcal{L}_{\mathrm{aux}}$ fixes the amplitude, removes vertical translation through a mean-zero condition and, when needed, enforces or breaks prescribed symmetry conditions. The active runs use $w_{\mathrm{eq}} = 0.1$ with residual-component weight one; the normalization and mean constraints are kept on throughout training.

4.1.3 Collocation and optimization

The collocation sets are resampled during optimization. In the runs used here, each optimization step combines fixed and moving collocation points: 251 fixed equation points, 201 fixed derivative points, two groups of 71 points concentrated near the endpoints and 121 moving points for each of the equation and derivative residuals. The sampling density is biased toward regions where high-amplitude waves develop sharp features, especially near crests and troughs. After a training pass, the point distribution can be updated from the residual itself so that later passes concentrate effort where the current approximation is least accurate.

The optimizer is KFAC in JAX, with adaptive learning rate, momentum and damping, minimum damping 10^{-14} and a typical training horizon of $15,000$ iterations per continuation step. New branches are followed by warm starting from a previously trained profile and allowing the speed, and when appropriate the surface tension, to move within a prescribed local window. The trained state is saved as both a weight checkpoint and a MATLAB file containing x, y_N, $\hat{H}y_N$, derivatives, residual values and the loss history.

4.2 Second-stage training

After the first PINN pass produced a low-residual profile, see Figure , we applied a refinement stage designed to remove the remaining structured residual. The guiding principle is the following elementary linearization. If $F(y) = 0$ is a nonlinear equation and y_p is an approximate solution with residual $r_p = F(y_p)$, then for a small correction h,

$$F(y_p + h) = r_p + DF(y_p)h + O(\|h\|^2).$$

Thus, as long as the correction remains small, training a correction network is approximately solving $DF(y_p)h \simeq -r_p$. We therefore wrote the refined unknown as

$$Y_\theta(u) = Y_p(u) + \varepsilon_p \Phi_\theta(u; d_f),$$

where $Y = (y, T_1, T_2)$, Y_p is the previously trained profile and auxiliary fields, and Φ_θ is a small oscillatory correction network. In the certified Wilton refinement we used $\varepsilon_p = 10^{-10}$. The oscillatory scale $d_f = 144$ was chosen from the dominant Fourier mode of the first-stage residual. Concretely, the correction network used sinusoidal activations, input $\cos u$, and a first-layer frequency multiplier d_f; the T_1 component was multiplied by $\sin u$ to preserve the odd symmetry of the tangent field.

This refinement stage also changed the residual evaluation. Instead of the nonuniform Hilbert-transform quadrature used during discovery, we evaluated the periodic Hilbert transform on a full uniform grid using the FFT rule

$$\widehat{\hat{H} f}_k = -i \operatorname{sgn}(k) \hat{f}_k,$$

with the Nyquist mode set to zero. Training batches used the full grid

$$u_j^{(s)} = -\pi + \frac{2\pi}{N}(j + s), \qquad j = 0, \dots, N-1,$$

with $N = 4096$ and shifts $s \in \{0, 0, 1/4, 1/2, 3/4\}$. Thus each residual evaluation was spectral and periodic; resampling consisted only of shifting the uniform grid. The final certified residual was then re-evaluated on a 2^{16}-point uniform grid.

The loss used the gradient-normalized residual weighting from the Stage 3 refinement code. Let

$$E(Y_\theta) = (e_1, e_2, e_3)$$

denote the three residual components: the Babenko residual and the two auxiliary equations defining T_1, T_2. At a grid point u_j, the equation residual was divided by a stopped-gradient scale

$$S_b(u_j) = \operatorname{clip}_{[0.2,100]}\left[\frac{|y_p'| + |\hat{H} y_p'| + b\left(|y_p''| + |\hat{H} y_p''|\right) + \beta^2\left(|T_{1,p}'| + |\hat{H} T_{2,p}'|\right)}{\left\langle |y_p'| + |\hat{H} y_p'| + b\left(|y_p''| + |\hat{H} y_p''|\right) + \beta^2\left(|T_{1,p}'| + |\hat{H} T_{2,p}'|\right)\right\rangle + 10^{-30}}\right],$$

where $b = 1$ for the main residual and $b = 2$ for the derivative residual. This scale was computed from the fixed base profile Y_p, not differentiated through during

optimization. The equation part of the loss was therefore

$$\mathcal{L}_{\mathrm{eq}} = \sum_{j,\ell} w_{j\ell} \left(\frac{e_\ell(Y_\theta)(u_j)}{S_1(u_j)} \right)^2 + 10^{-6} \sum_j w_j^{(d)} \left(\frac{\partial_u e_1(Y_\theta)(u_j)}{S_2(u_j)} \right)^2 .$$

The first term trains the normalized Babenko and auxiliary residuals, while the second term weakly penalizes the derivative of the main Babenko residual. Additional endpoint and hotspot weights were used only as quadrature weights: they concentrated training near the endpoints and the observed residual peaks while keeping the total quadrature mass of each residual component fixed. This combination of a small residual-correction network, FFT Hilbert transform on shifted uniform grids, and the derivative-based stopped-gradient normalization reduced the residual to the round-off scale needed for certification.

Before the second-stage refinement, the equation loss was the weighted mean-square residual of the Babenko system and, when enabled, the derivative of the main Babenko residual,

$$\mathcal{L}_{\mathrm{eq}}^{(1)} = \sum_{j,\ell} w_{j\ell} \left| e_\ell(Y_\theta)(u_j) \right|^2 + w_{\mathrm{der}} \sum_j w_j^{(d)} \left| \partial_u e_1(Y_\theta)(u_j) \right|^2 .$$

Here

$$E(Y_\theta) = (e_1, e_2, e_3),$$

where e_1 is the Babenko residual and e_2, e_3 are the auxiliary residuals enforcing the definitions of T_1 and T_2. The weights $w_{j\ell}$ and $w_j^{(d)}$ are the collocation/quadrature weights for the equation and derivative residuals, respectively. In the first-stage discovery runs the main equation loss used $w_{\mathrm{eq}} = 0.1$, and the derivative residual was used as a diagnostic or weak auxiliary term depending on the run.

4.2.1 Gradient-normalized residual weighting

The second-stage refinement uses a gradient-normalized residual loss, following the idea that a standard mean-square residual can become poorly balanced when the solution contains steep or rapidly varying regions. In a standard residual loss one minimizes

$$\sum_j |\mathcal{R}(Y_\theta)(u_j)|^2.$$

This treats the residual as an absolute error. In high-gradient regions, however, the natural scale of the equation can be much larger than in flatter parts of the profile, so those regions can dominate the optimization. The normalized loss instead minimizes a local relative residual,

$$\sum_j \left| \frac{\mathcal{R}(Y_\theta)(u_j)}{\text{local derivative scale at } u_j} \right|^2 .$$

This is the principle behind the gradient-normalized weighting used in the second-stage training.

For the Babenko equation, the residual has three components,

$$\mathcal{R}(Y_\theta) = (\mathcal{R}_1(Y_\theta), \mathcal{R}_2(Y_\theta), \mathcal{R}_3(Y_\theta)),$$

where $\mathcal{R}_1$ is the Babenko residual and $\mathcal{R}_2, \mathcal{R}_3$ enforce the auxiliary definitions of T_1, T_2. The local derivative scale was computed from the fixed profile $Y_p = (y_p, T_{1,p}, T_{2,p})$ obtained before the refinement stage. At each grid point u_j, we set

$$S_b(u_j) = \mathrm{clip}_{[0.2,100]}\left(\frac{|y_p'| + |\hat{H}y_p'| + \beta^2\left(|T_{1,p}'| + |\hat{H}T_{2,p}'|\right) + b\left(|y_p''| + |\hat{H}y_p''|\right)}{\left\langle |y_p'| + |\hat{H}y_p'| + \beta^2\left(|T_{1,p}'| + |\hat{H}T_{2,p}'|\right) + b\left(|y_p''| + |\hat{H}y_p''|\right)\right\rangle + 10^{-30}}\right).$$

Here $\langle\cdot\rangle$ denotes the grid average. The clipping operation

$$\mathrm{clip}_{[0.2,100]}(x) = \min\{100, \max\{0.2, x\}\}$$

prevents extreme reweighting: the denominator is never allowed to become so small that flat regions dominate the loss, or so large that steep regions are ignored. These bounds are numerical safeguards rather than physical parameters. The scale is treated as fixed during optimization, so the network cannot reduce the loss by changing the denominator.

The normalized equation loss used in the refinement was

$$\mathcal{L}_{\mathrm{eq}} = \sum_{j,\ell} w_{j\ell}\left|\frac{\mathcal{R}_\ell(Y_\theta)(u_j)}{S_1(u_j)}\right|^2 + 10^{-6}\sum_j w_j^{(d)}\left|\frac{\partial_u \mathcal{R}_1(Y_\theta)(u_j)}{S_2(u_j)}\right|^2.$$

The first term is the normalized loss for the three residual components. The second term weakly penalizes the derivative of the main Babenko residual, which helps suppress remaining oscillatory error. The two denominators are the same derivative scale with different boost factors:

$$S_1 = S_b\big|_{b=1}, \qquad S_2 = S_b\big|_{b=2}.$$

The larger value $b = 2$ was used for the derivative-residual term because $\partial_u \mathcal{R}_1$ is more sensitive to high-frequency and high-curvature error than $\mathcal{R}_1$ itself. The collocation grid was uniform; the derivative-residual term used equal quadrature weights, while the equation-residual weights were normalized componentwise and mildly concentrated near endpoints and observed residual hotspots.

4.3 Figure generation and filtering

For all plotted profiles, the physical free surface is shown as $(x + \hat{H}y_N, y_N)$. The amplitude used in the branch plots is

$$a = \frac{\max y_N - \min y_N}{2\pi}. \tag{18}$$

Branch panels include all files from the corresponding uploaded folders, while representative profile panels are selected from runs with $\max |\mathcal{B}(y_N)| \leq 10^{-4}$ unless otherwise stated. The plotting script records every plotted file, parameter value, amplitude and residual in a CSV summary, making the figures reproducible from the saved `.mat` outputs.

4.4 Rigorous certification of the numerical solution

To rigorously quantify the accuracy of the computed wave, we certify the residual of an explicit finite-dimensional approximation using interval arithmetic. The neural network itself is not part of the certificate. After training and continuation, the computed profile is projected onto an even cosine series,

$$y_N(u) = \sum_{k=0}^{N} a_k \cos(ku), \tag{19}$$

and the coefficients are refined by Newton iteration in coefficient space. The resulting coefficients, together with the wave parameters, are fixed before certification.

For this explicit profile we evaluate the residual

$$R_N(u) = \mathcal{B}_{\mu,\beta}(y_N)(u) \tag{20}$$

of the gravity–capillary Babenko equation using interval ball arithmetic implemented in Julia with Arb. All arithmetic operations are outward rounded, so that the resulting intervals rigorously enclose the corresponding exact quantities. The computation accounts for roundoff, Fourier truncation and aliasing errors, and the evaluation of the nonlinear and nonlocal terms in the equation.

The periodic Hilbert transform of finite trigonometric polynomials is evaluated exactly through its Fourier multiplier. For nonlinear composite quantities, which are not band limited, the Hilbert transform is computed using a discrete Fourier representation together with rigorous bounds on the unresolved Fourier tail. The computation also establishes a positive lower bound on the denominator appearing in the capillary term.

To obtain a uniform rather than pointwise certificate, the residual is first enclosed on a sufficiently fine periodic grid and then controlled between grid points using Taylor expansions with rigorous remainder bounds. This yields an explicit constant ε such that

$$\|\mathcal{B}_{\mu,\beta}(y_N)\|_{L^\infty(\mathbb{R})} \leq \varepsilon. \tag{21}$$

Details of the interval construction, Fourier-tail estimates and off-grid bounds are provided in the Supplementary Methods.

4.5 Formal verification in Lean

We additionally formalize the certification argument in Lean 4. The role of Lean is distinct from that of the interval-arithmetic computation: Lean does not repeat the numerical calculation, but verifies the mathematical identities and logical deductions that connect the certified interval bounds to the final uniform residual statement.

The finite cosine representation, differentiation, periodic Hilbert transform and Babenko residual are defined within the formalization. The numerical bounds produced by Arb are imported as exact rational endpoints, and Lean verifies that these bounds imply

$$\forall u \in \mathbb{R}, \qquad |\mathcal{B}_{\mu,\beta}(y_N)(u)| \le \varepsilon. \tag{22}$$

In particular, the formalization checks the correspondence between the analytic and computational representations of the residual and the extension of the certificate from one period to all of $\mathbb{R}$.

The numerical Arb bounds constitute the numerical trust boundary of the formal proof; all subsequent deductions are checked by the Lean kernel. Further details of the formalization and its trust assumptions are given in the Supplementary Methods.

5 Data availability

The `.mat` files used to generate the figures are stored locally in the run folders listed in the manuscript workspace, including the pure-gravity PINN runs and the gravity-capillary neural-network runs. A curated archive of the selected trained profiles, residual diagnostics and figure-generation summary table will be deposited before submission.

6 Code availability

The plotting script used for this draft is stored in the manuscript working directory together with the generated figure summary table. The PINN training notebooks and checkpoint files are currently stored with the run data and will be organized into a public repository before submission.

Declarations

Competing interests The authors declare no competing interests.
Author contributions M.S., Y.W. and T.B. designed the project. M.S. prepared the manuscript draft and figures. Y.W. and M.S. developed and ran the PINN computations. All authors contributed to the mathematical formulation and interpretation of the results.

Appendix - other exotic profiles

The pure-gravity runs also reveal a secondary family of non-symmetric double-crest Stokes waves. This family is naturally connected to the scaling symmetry of the pure-gravity Babenko equation: if $(\mu, y(u))$ is a 2π-periodic Stokes wave, then $(\mu/n, y(nu)/n)$ gives the corresponding n-crest rescaling. Along the symmetric two-crest branch, earlier numerical work identified bifurcations to waves with two unequal crests [13, 14]. We use the term "exotic branch" for the low-residual PINN candidates on this non-symmetric double-crest family. The current figure includes only representative profiles with $\max|\mathcal{B}(y_N)| \leq 10^{-4}$, while the residual diagnostic keeps the exploratory files visible.

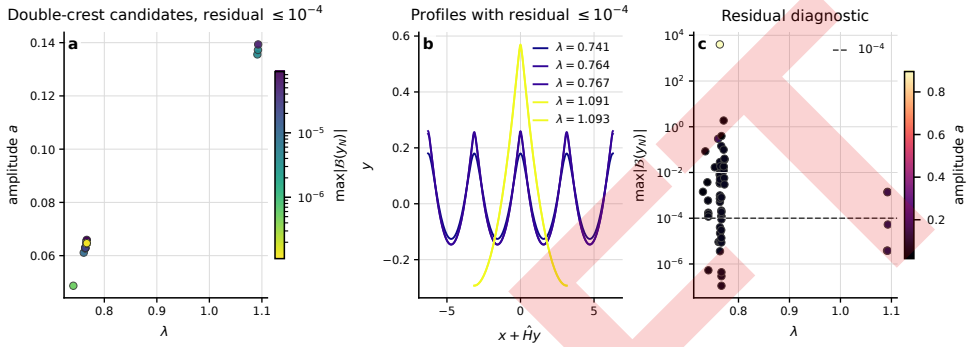

Fig. 6 Pure-gravity non-symmetric double-crest candidates. a, Low-residual candidates on the non-symmetric double-crest branch, plotted in the (λ, a) plane. **b,** Representative double-crest profiles with $\max|\mathcal{B}(y_N)| \leq 10^{-4}$. **c,** Residual diagnostic for all uploaded double-crest files; the dashed line marks 10^{-4} and separates the profiles used as representatives from exploratory runs.

6.1 Rigorous residual certification by interval arithmetic

6.1.1 Certified problem

The certification is performed on an explicit finite trigonometric polynomial, independently of the neural network used to discover the wave. Let

$$y_N(u) = \sum_{k=0}^{N} a_k \cos(ku), \qquad u \in \mathbb{R}, \tag{23}$$

and fix parameters $\mu, \beta > 0$. We define the residual

$$\begin{aligned} R_N(u) = {} & \mu \hat{H}[y_N'] - y_N - y_N \hat{H}[y_N'] - \hat{H}[y_N y_N'] \\ & + \beta \partial_u \left(\frac{y_N'}{D} \right) - \beta \hat{H} \left[\partial_u \left(\frac{1 + \hat{H}[y_N']}{D} \right) \right], \end{aligned} \tag{24}$$

where

$$D(u) = \sqrt{y'_N(u)^2 + \left(1 + \hat{H}[y'_N](u)\right)^2}. \tag{25}$$

Here $\hat{H}$ denotes the 2π-periodic Hilbert transform, with convention

$$\hat{H}[\cos(ku)] = \sin(ku), \qquad \hat{H}[\sin(ku)] = -\cos(ku), \qquad k \geq 1. \tag{26}$$

Our objective is to compute an explicit constant $\varepsilon > 0$ such that

$$\|R_N\|_{L^\infty(\mathbb{R})} = \sup_{u \in \mathbb{R}} |R_N(u)| \leq \varepsilon. \tag{27}$$

All coefficients a_k and parameters entering the certification are fixed before the rigorous computation. Thus, the statement (27) concerns a single explicit trigonometric polynomial and does not depend on the subsequent behavior of the neural network or optimization algorithm.

6.1.2 Interval ball arithmetic

All numerical estimates entering the certificate are evaluated using Arb ball arithmetic through Arblib.jl. A real quantity x is represented by an interval

$$x \in [m - r, m + r], \tag{28}$$

where m is the midpoint and $r \geq 0$ is the radius. Every arithmetic operation is performed with outward rounding. Consequently, if the input intervals contain the exact mathematical inputs, the resulting interval contains the exact value of the corresponding operation.

This enclosure property is propagated through the arithmetic operations, elementary functions, trigonometric evaluations, Fourier operations, and nonlinear expressions entering (24). The final output is therefore a rigorous bound for the residual of the fixed coefficients supplied to the certification procedure, rather than an estimate of floating-point roundoff.

Two additional sources of error must nevertheless be controlled explicitly. First, nonlinear operations generate functions that are no longer finite trigonometric polynomials, so their Hilbert transforms require a finite discrete Fourier representation and an accompanying aliasing estimate. Second, evaluation on a finite grid does not by itself control the residual between grid points. These two errors are treated separately below.

6.1.3 Exact operations on finite trigonometric polynomials

For a finite trigonometric polynomial

$$f(u) = \frac{a_0}{2} + \sum_{k=1}^{K} \left(a_k \cos(ku) + b_k \sin(ku)\right), \tag{29}$$

differentiation and the periodic Hilbert transform are performed directly on the Fourier coefficients. In particular,

$$f'(u) = \sum_{k=1}^{K} \left(k b_k \cos(ku) - k a_k \sin(ku) \right), \tag{30}$$

and

$$\hat{H} f(u) = \sum_{k=1}^{K} \left(-b_k \cos(ku) + a_k \sin(ku) \right). \tag{31}$$

Products of finite trigonometric polynomials are likewise formed by exact convolution of their Fourier coefficients.

In particular,

$$y_N'(u) = -\sum_{k=1}^{N} k a_k \sin(ku), \tag{32}$$

and hence

$$\hat{H}[y_N'](u) = \sum_{k=1}^{N} k a_k \cos(ku). \tag{33}$$

No Fourier truncation or aliasing error is introduced in these operations.

In the implemented residual pipeline, however, not every occurrence of the Hilbert transform is evaluated through this exact representation. The product $y_N y_N'$, although itself a finite trigonometric polynomial, is represented in the common Taylor-series layer and its Hilbert transform is evaluated using the certified discrete procedure described below. The same procedure is necessarily used for the capillary term involving D^{-1}, which is not band limited.

6.1.4 Taylor representation and derivative bounds

To obtain a bound that is uniform in u, all functions entering the residual are represented on the periodic grid

$$u_j = \frac{2\pi j}{N_{\text{grid}}}, \qquad j = 0, \ldots, N_{\text{grid}} - 1, \tag{34}$$

together with rigorous information about their derivatives.

For a smooth periodic function f, the computation stores interval enclosures

$$[f^{(m)}(u_j)], \qquad 0 \le m \le M, \tag{35}$$

as well as global derivative bounds

$$B_m(f) \ge \|f^{(m)}\|_{L^\infty}, \qquad 0 \le m \le M + 1. \tag{36}$$

These bounds are propagated at each derivative order rather than through a single scalar estimate. For products we use the Leibniz identity

$$(fg)^{(m)} = \sum_{\ell=0}^{m} \binom{m}{\ell} f^{(\ell)} g^{(m-\ell)}, \tag{37}$$

which gives

$$B_m(fg) \le \sum_{\ell=0}^{m} \binom{m}{\ell} B_\ell(f) B_{m-\ell}(g). \tag{38}$$

For nonlinear compositions, including reciprocals and square roots, the corresponding derivative vectors are computed using the Faà di Bruno formula. All resulting expressions are evaluated using ball arithmetic.

This construction provides simultaneously rigorous Taylor data at the grid points and the global derivative bounds needed to control the remainder between grid points.

6.1.5 Certification of the capillary denominator

Define

$$P(u) = y_N'(u)^2 + \Big(1 + \hat{H}[y_N'](u)\Big)^2, \qquad D(u) = \sqrt{P(u)}. \tag{39}$$

Since y_N' and $\hat{H}[y_N']$ are finite trigonometric polynomials, P is itself a finite trigonometric polynomial and is constructed directly from its Fourier coefficients.

Before any reciprocal involving D is evaluated, the computation certifies a strictly positive lower bound

$$D(u) \ge D_{\text{lower}} > 0 \qquad \text{for every } u \in \mathbb{R}. \tag{40}$$

The lower bound is obtained from the Taylor representation and rigorous derivative bounds for D. Once (40) has been established, D^{-1} and its derivatives can be computed rigorously using interval arithmetic and Faà di Bruno differentiation. Positivity of the denominator is therefore part of the certificate rather than an assumption of the residual computation.

6.1.6 The nonlinear capillary quantity

Introduce

$$G(u) = \frac{1 + \hat{H}[y_N'](u)}{D(u)}. \tag{41}$$

Unlike y_N and $\hat{H}[y_N']$, the function G is not generally a finite trigonometric polynomial because D^{-1} contains infinitely many Fourier modes.

The Taylor representation of G is constructed from the numerator and denominator using the product, reciprocal, and composition rules described above. In this way the computation obtains rigorous interval enclosures of

$$G^{(m)}(u_j) \tag{42}$$

at every grid point, together with global bounds on the corresponding derivatives.

The capillary residual contains

$$\hat{H}[\partial_u G]. \tag{43}$$

We use the standard commutation identity

$$\partial_u^m \hat{H}[f] = \hat{H}[\partial_u^m f] \tag{44}$$

for periodic smooth functions. Thus the Taylor data for $\hat{H}[f]$ can be constructed derivative by derivative by applying the certified discrete Hilbert transform to $f^{(m)}$.

6.1.7 Certified discrete Hilbert transform

Let f be a smooth 2π-periodic function sampled at an even number N_{fft} of equally spaced points. The discrete Hilbert transform used in the computation is evaluated as a circular convolution with the periodic discrete Hilbert kernel

$$h_n = \frac{2}{N_{\mathrm{fft}}} \cot\left(\frac{\pi n}{N_{\mathrm{fft}}}\right) \tag{45}$$

for odd offsets n, with the complementary entries equal to zero. The discrete Fourier multiplier of this kernel is the spectral Hilbert multiplier

$$-i \operatorname{sgn}(k) \tag{46}$$

on the resolved modes, with the zero and Nyquist modes set to zero.

The circular convolution is carried out in complex Arb arithmetic. All rounding errors arising from this discrete computation are therefore included automatically in the output balls. The remaining difference between the discrete transform and the Hilbert transform of the underlying continuous function is the Fourier aliasing error.

Suppose that, for some integer $p \geq 2$,

$$\|f^{(p)}\|_{L^\infty} \leq C_p. \tag{47}$$

Repeated integration by parts in the Fourier coefficient formula gives

$$|\widehat{f}_k| \leq \frac{C_p}{|k|^p}, \qquad k \neq 0. \tag{48}$$

Modes beyond the resolved band are folded onto lower frequencies by sampling. The difference between the exact Hilbert multiplier and the multiplier applied after aliasing is bounded in magnitude by 2. Accounting also for positive and negative Fourier modes gives

$$\varepsilon_{\mathrm{alias}} \leq 4C_p \sum_{k \geq N_{\mathrm{fft}}/2} \frac{1}{k^p}. \tag{49}$$

Using the integral estimate

$$\sum_{k\geq K} k^{-p} \leq \frac{(K-1)^{1-p}}{p-1}, \qquad p>1, \tag{50}$$

we obtain

$$\boxed{\varepsilon_{\text{alias}} \leq \frac{4C_p}{(p-1)\,(N_{\text{fft}}/2-1)^{p-1}}.} \tag{51}$$

This error radius is added to every ball returned by the discrete Hilbert transform. Consequently, the resulting interval contains the value of the continuous periodic Hilbert transform at the corresponding grid point, including both numerical roundoff and unresolved Fourier modes.

The procedure is applied independently at each derivative level. In combination with (44), this provides rigorous grid enclosures for

$$(\hat{H}f)^{(m)}(u_j), \qquad 0\leq m\leq M. \tag{52}$$

6.1.8 Global derivative bounds for the Hilbert transform

In addition to grid values, the Taylor representation requires global derivative bounds. These are obtained directly from Fourier decay.

For a smooth 2π-periodic function q, two integrations by parts imply

$$|\widehat{q}_k| \leq \frac{\|q''\|_{L^\infty}}{k^2}, \qquad k\neq 0. \tag{53}$$

Since the Hilbert multiplier has modulus one on every nonzero Fourier mode,

$$\begin{aligned} \|\hat{H}q\|_{L^\infty} &\leq \sum_{k\neq 0} |\widehat{q}_k| \\ &\leq 2\|q''\|_{L^\infty} \sum_{k=1}^{\infty} \frac{1}{k^2} \\ &= \frac{\pi^2}{3}\|q''\|_{L^\infty}. \end{aligned} \tag{54}$$

Applying this estimate to $q = f^{(m)}$ yields

$$\boxed{\|(\hat{H}f)^{(m)}\|_{L^\infty} \leq \frac{\pi^2}{3}\|f^{(m+2)}\|_{L^\infty}.} \tag{55}$$

Accordingly, two additional derivative bounds are carried beyond the highest derivative order appearing in the Taylor polynomial. These extra bounds provide the global sup-norm information required after each discrete Hilbert-transform call.

6.1.9 Assembly of the residual

The residual is assembled in the common Taylor representation as

$$\begin{aligned} R_N = \mu\hat{H}[y'_N] - y_N - y_N\hat{H}[y'_N] - \hat{H}[y_N y'_N] \\ + \beta\partial_u\left(\frac{y'_N}{D}\right) - \beta\hat{H}[\partial_u G]. \end{aligned} \tag{56}$$

The terms involving y_N, y'_N, and $\hat{H}[y'_N]$ are obtained directly from the finite Fourier representation. Products and nonlinear operations in the Taylor representation are propagated using the exact Leibniz and Faà di Bruno formulas. The terms $\hat{H}[y_N y'_N]$ and $\hat{H}[\partial_u G]$ are evaluated using the certified discrete Hilbert transform and therefore include the aliasing enclosure (51).

As a result, for every grid point u_j and every derivative order needed for the subsequent Taylor estimate, the computation produces an interval containing the exact value of

$$R_N^{(m)}(u_j). \tag{57}$$

It simultaneously produces global bounds

$$B_m(R_N) \geq \|R_N^{(m)}\|_{L^\infty}. \tag{58}$$

6.1.10 From the finite grid to a uniform bound

A rigorous residual estimate at the grid points alone does not establish (27). We therefore use Taylor's theorem to control the residual throughout each grid cell.

Let

$$h = \frac{2\pi}{N_{\text{grid}}}, \tag{59}$$

and let u lie in the cell associated with u_j. Writing

$$u = u_j + \delta, \qquad |\delta| \leq \frac{h}{2}, \tag{60}$$

Taylor's theorem gives

$$R_N(u) = \sum_{m=0}^{M} \frac{R_N^{(m)}(u_j)}{m!}\delta^m + \mathcal{E}_{M+1}(u), \tag{61}$$

where

$$|\mathcal{E}_{M+1}(u)| \leq \frac{B_{M+1}(R_N)}{(M+1)!}|\delta|^{M+1}. \tag{62}$$

It follows that, throughout the cell,

$$|R_N(u)| \leq \sum_{m=0}^{M} \frac{|R_N^{(m)}(u_j)|}{m!}\left(\frac{h}{2}\right)^m + \frac{B_{M+1}(R_N)}{(M+1)!}\left(\frac{h}{2}\right)^{M+1}. \tag{63}$$

Every quantity on the right-hand side is represented by a rigorous interval. Taking the maximum of the resulting upper endpoints over all grid cells yields an explicit number ε satisfying

$$|R_N(u)| \le \varepsilon \qquad \text{for every } u \in [0, 2\pi]. \tag{64}$$

Finally, periodicity implies

$$\|R_N\|_{L^\infty(\mathbb{R})} \le \varepsilon. \tag{65}$$

Thus the final estimate is a continuum bound and not a maximum of sampled residual values.

6.1.11 Certified computation for the reported wave

For the candidate reported in the main text, the explicit profile is truncated at

$$N = 150. \tag{66}$$

The rigorous calculation uses

$$N_{\text{grid}} = 2{,}000{,}000, \qquad M = 6, \tag{67}$$

at 128-bit Arb precision.

The computation yields

$$\boxed{\|\mathcal{B}_{\mu_0,\beta_0}(y_{150})\|_{L^\infty(\mathbb{R})} \le 1.78 \times 10^{-13}} \tag{68}$$

and, independently,

$$\boxed{D(u) \ge 0.3446588 > 0 \qquad \text{for every } u \in \mathbb{R}.} \tag{69}$$

The bound (68) includes the propagated Arb rounding error, the discrete-Hilbert aliasing error, the nonlinear derivative bounds, and the Taylor remainder controlling the intervals between grid points. It is therefore a rigorous uniform residual certificate for the explicit polynomial y_{150}.

7 garbage

7.1 Certified error estimates and formal verification

We seek a computer-assisted proof (CAP) of the following statement:

Let $y(u) = \sum_{k=0}^{N} a_k \cos(ku)$ be a fixed finite even cosine series and let $\mu, \beta > 0$ be given rational parameters. Define the residual of the Babenko equation with surface tension by

$$R(u) = \mathcal{B}(\dagger(\sqcap)) = \ldots + D + \ldots\ldots \tag{70}$$

Then $\|R(y_N)\|_\infty = \max_u |R(u)| \le \epsilon$ for an explicit certified constant ϵ.

The proof is split between rigorous arithmetic via interval ball arithmetic (Julia) and logical scaffolding that imports the certificate and deduces the bound (Lean 4).

Neither component is trusted alone: Julia performs all the hard computation with certified error enclosures; Lean checks the logic that turns those enclosures into a theorem.

Scope. The deliverable is the residual sup-norm bound above — nothing more. The bound is the quantitative input one would need for a downstream Newton–Kantorovich / contraction argument to upgrade "small residual" into "existence of an exact gravity-capillary Stokes wave nearby" — but inverting the linearisation of the Babenko operator (an unbounded infinite-dimensional pseudodifferential operator) is a separate, paper-level problem and is out of scope here.

Why a rigorous residual bound? Formal perturbation theory gives approximate solutions (e.g., truncated Stokes expansions) but no rigorous certificate of how good the approximation is. A rigorous residual bound is the first half of a computer-assisted existence proof: it provides the quantitative ϵ in $|R(u)| \leq \epsilon$ that a downstream Newton–Kantorovich argument consumes. For gravity-capillary waves the parameter space (μ, β) has a rich bifurcation structure where purely analytic arguments are incomplete, so even a residual bound (without the existence upgrade) is of independent value as a quantitative diagnostic. Last, it provides a reliability certificate for PINN's as small residual numerical method for approximating solutions of integro-differential non-linear equations.

What the residual bound delivers, and what it doesn't:

yes. A certified explicit constant ϵ such that $\|R(y_N)\|_\infty \leq \epsilon$ for the user's chosen (a_k, μ, β) yes. Lean-side guarantee that ϵ is free of unverified numerical claims (all heavy arithmetic is in Arb interval-ball form on the Julia side; Lean checks the universal analytic lemmas).

Not included: a Lean-side fixed-point argument upgrading $\|R(y_N)\|_\infty \leq \epsilon$ to the rigouruous existence of $y^\star$, $R(y^\star) = 0$. That is the Newton–Kantorovich step proper, and is left for a follow-up.

The Hilbert transform appears in four places in the residual. The treatment differs depending on whether the input is a trig polynomial (exact) or a nonlinear function (approximated by DFT).

7.1.1 A rigorous residual certificate for the discovered wave

From the trained Wilton-ripple solution at $(c_0, \beta_0) = (1.16878648,\ 0.498^2)$ we extract cosine coefficients by Fourier transform, polish them by a Newton iteration in coefficient space, and truncate at wavenumber $N = 150$. The result is an explicit trigonometric polynomial

$$y_{150}(u) = \sum_{k=0}^{150} a_k \cos(ku), \tag{71}$$

whose 151 coefficients are double-precision numbers and therefore exact dyadic rationals. Every statement below is a statement about this finite list of rational numbers; the network, the optimizer and the training run play no further role.

The certificate is a uniform bound on the Babenko residual $\mathcal{B}(y_{150})$, with the wave speed and surface-tension coefficient frozen. Every arithmetic operation is performed in ball arithmetic (Arb), in which each quantity is carried as a midpoint–radius pair

rigorously enclosing the true value, with outward rounding. The Hilbert transform is handled by two separate pathways: on trigonometric polynomials, such as y_{150} and its derivative, $\hat{H}$ acts exactly mode by mode and contributes no error at all; on the capillary composite $G = (1 + \hat{H}\partial_u y)/D$, which is not a trigonometric polynomial because $D = \sqrt{(\partial_u y)^2 + (1 + \hat{H}\partial_u y)^2}$ has infinitely many Fourier modes, $\hat{H}$ is evaluated by a certified discrete transform whose aliasing error is bounded explicitly from the decay of the Fourier tail. Because the capillary term divides by D, the computation simultaneously certifies a positive lower bound for it. Finally, the residual is not merely sampled: on each grid subinterval a Taylor expansion of order $M = 6$ with a certified remainder converts grid information into a bound valid for *every* $u \in \mathbb{R}$. Details of each step are given in Methods.

Running this pipeline on a grid of $N = 2{,}000{,}000$ points at 128-bit precision yields the rigorous bounds

$$\|\mathcal{B}(y_{150})\|_{L^\infty(\mathbb{R})} \;\le\; 1.78 \times 10^{-13}, \qquad D(u) \;\ge\; 0.34465 \;>\; 0 \quad \text{for all } u. \tag{72}$$

In this sense the headline figure means what it says: the discovered profile satisfies the fully nonlinear gravity–capillary equation to 10^{-13}, rigorously and uniformly in u.

7.1.2 Machine-checked verification of the certificate

Interval-arithmetic proofs are rigorous but notoriously hard to referee: the argument is distributed over thousands of elementary inequalities executed by code that the reader is implicitly asked to trust. The final stage of the pipeline addresses this by making the certificate machine-checkable in the proof assistant Lean 4. The exported certificate contains no floating-point numbers at all: the 151 coefficients and the two Arb output bounds are serialized as exact rational constants, and the certified run is byte-reproducible from the archived computation.

Within Lean, the residual operator is defined from first principles — Fourier sums, the periodic Hilbert transform, the curvature denominator — and the final theorem, checked by the Lean kernel, states

$$\forall\, u \in \mathbb{R}, \qquad |\mathcal{B}(y_{150})(u)| \;\le\; \frac{178}{10^{15}}, \tag{73}$$

together with the companion bound $D(u) \ge 0.3446588$. Several parts of this statement are themselves proved rather than assumed. The formalized residual is proved to coincide (up to overall sign, an identity established symbolically) with the Babenko operator exactly as written in equation (4); the mode-by-mode cosine expansion used for $\hat{H}\partial_u y$ is proved equal to the true periodic Hilbert transform; the storage conventions of the numerical and formal layers are reconciled by a proved identity rather than by trusted bookkeeping; and the bound, certified by the computation on one period, is extended to all of $\mathbb{R}$ by a proved periodicity argument. Alongside these instance-specific facts, the analytic ingredients of the certification pipeline — the aliasing bound for the discrete Hilbert transform, the Taylor-remainder estimate, and the Faà di Bruno,

Leibniz and Fourier-tail inequalities — are formalized as general theorems from the standard axioms of Lean's library.

The theorem (73) is not, however, an independent replay of the interval computation, and we state its trust boundary precisely. Lean's axiom checker reports that the result depends on the three standard axioms of the Lean/Mathlib foundation plus exactly two project-specific axioms: one recording the Arb upper endpoint for the residual on one period, and one recording the Arb lower bound for D (the latter taken as an explicit hypothesis by the former, since positivity of D underlies the capillary chain). These two rational numbers are the entire numerical trust boundary; everything between them and the final theorem is machine-checked logic. The resulting division of trust — the Lean kernel and its library at the logical level, the FLINT/Arb library at the arithmetic level — is the one used in most computer-assisted proofs that combine interval arithmetic with a proof assistant, here made auditable by the axiom report itself. We emphasize the scope of the claim: the theorem certifies the residual of an explicit polynomial, and neither asserts nor requires the existence of an exact solution nearby; upgrading residual bounds to an existence theorem requires control of the linearized operator and is discussed below.

8 Lean theorem results

The installed certificate is the Lean module `CertificatePhase4ScalarN150Phase420260827HalfDC`. It records $N_{\mathrm{grid}} = 2{,}000{,}000$, $M = 6$, the half-DC-aligned coefficient vector, and the exact rational Arb endpoints. The project was rebuilt successfully after installation, and the certificate file was verified byte-for-byte against the downloaded output of the Crunchy run.

Theorem 1 (exact constant-mode alignment).
For the source zeroth coefficient a_0, the certificate coefficient satisfies

$$\widehat{a}_0 = a_0/2 \tag{74}$$

as an exact rational identity. The Lean declaration is

`phase4_n150_dc_coefficient_is_source_half`.

Together with the generic half-DC lemmas, it proves that Julia and Lean evaluate the same profile rather than two series differing by a constant offset.

Theorem 2 (uniform positivity of the denominator).
The corresponding Lean declaration is

`D_bounded_below_by_phase4_n150_half_dc_decimal`.

It states

$$\forall u \in \mathbb{R}, \qquad D(a_{\mathrm{cert}})(u) \geq 0.3446588 > 0. \tag{75}$$

The displayed decimal is a slightly weakened exact-rational corollary of the Arb lower endpoint in Eq. (**??**).

Theorem 3 (exact uniform residual certificate).
The principal declaration is

`residual_bounded_by_phase4_n150_half_dc_certificate`.

Its mathematical statement is

$$\forall u \in \mathbb{R}, \qquad |\mathcal{R}(a_{\mathrm{cert}}, \mu_{\mathrm{cert}}, \beta_{\mathrm{cert}}; u)| \le \varepsilon_{0,\mathrm{cert}}, \tag{76}$$

where $\varepsilon_{0,\mathrm{cert}}$ is the exact rational encoding of the Arb upper endpoint in Eq. (**??**).

Corollary 4 (paper-scale residual bound).
The Lean declaration

`residual_bounded_by_phase4_n150_half_dc_decimal`

proves the more readable inequality

$$\boxed{\forall u \in \mathbb{R}, \qquad |\mathcal{R}(a_{\mathrm{cert}}, \mu_{\mathrm{cert}}, \beta_{\mathrm{cert}}; u)| \le \frac{178}{10^{15}} = 1.78 \times 10^{-13}.} \tag{77}$$

The comparison between the exact Arb endpoint and $178/10^{15}$ is proved inside Lean by exact rational normalization. The axiom audit for Eqs. (76)–(77) contains precisely the two numerical anchors in Eqs. (**??**)–(**??**), in addition to the three standard Mathlib logical principles listed above.

9 methods

References

[1] Nalimov, V.I.: The cauchy-poisson problem. Dinamika Sploshn. Sredy **18**, 104–210 (1974)

[2] Lannes, D.: The Water Waves Problem: Mathematical Analysis and Asymptotics vol. 188. American Mathematical Society, Providence, RI (2013)

[3] 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. Journal of Computational Physics **378**, 686–707 (2019) https://doi.org/10.1016/j.jcp.2018.10.045

[4] Wang, Y., Léger, T., Lai, C.-Y., Buckmaster, T.: Resolving Sharp Gradients of Unstable Singularities to Machine Precision via Neural Networks (2025). https://doi.org/10.48550/arXiv.2511.22819 . https://arxiv.org/abs/2511.22819

[5] Toland, J.F.: Stokes waves. Topological Methods in Nonlinear Analysis **7**, 1–48 (1996)

[6] Ovsyannikov, L.V.: Dynamika sploshnoi sredy, lavrentiev institute of hydrodynamics. Sib. Branch Acad. Sci. USSR **15**, 104 (1973)

[7] Tanveer, S.: Singularities in water waves and rayleigh-taylor instability. Proceedings of the Royal Society of London. Series A **435**(1893), 137–158 (1991)

[8] Dyachenko, A.I., Zakharov, V.E., Kuznetsov, E.A.: Nonlinear dynamics of the free surface of an ideal fluid. Plasma Physics Reports **22**(10), 829–840 (1996)

[9] Hillebrecht, B., Unger, B.: Rigorous a posteriori error bounds for PDE-defined PINNs. IEEE Transactions on Neural Networks and Learning Systems **36**(1), 1583–1593 (2025) https://doi.org/10.1109/TNNLS.2023.3335837 arXiv:2210.03426

[10] Eiras, F., Bibi, A., Bunel, R.R., Dvijotham, K.D., Torr, P., Kumar, M.P.: Efficient error certification for physics-informed neural networks. In: Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 12318–12347. PMLR, ??? (2024). https://proceedings.mlr.press/v235/eiras24a.html

[11] Tanaka, K., Yatabe, K.: Learn and Verify: A Framework for Rigorous Verification of Physics-Informed Neural Networks (2026). https://doi.org/10.48550/arXiv.2601.19818 . https://arxiv.org/abs/2601.19818

[12] George, R.J., Cruden, J., Adkisson, W., Zhong, X., Zhang, H., Anandkumar, A.: TorchLean: Formalizing Neural Networks in Lean. Version 2 (2026). https://doi.org/10.48550/arXiv.2602.22631 . https://arxiv.org/abs/2602.22631v2

[13] Chen, B., Saffman, P.G.: Numerical evidence for the existence of new types of gravity waves of permanent form on deep water. Studies in Applied Mathematics **62**(1), 1–21 (1980)

[14] Semenova, A.: Two-crested stokes waves. arXiv preprint arXiv:2411.15184 (2024)