An integrated theoretical, experimental, and numerical study of small-amplitude water waves

Lennon Ó Náraigh¹, Nicolas Farault^{1,2}, Nicola Young^{1*}

¹ School of Mathematics and Statistics, University College Dublin, Belfield, Dublin 4, Ireland and

² Polytech Lyon, 15, Boulevard Latarjet 69622 VILLEURBANNE CEDEX (Dated: July 25, 2025)

We introduce an inexpensive experimental setup for analyzing free-surface water waves: a 1 m-long tabletop flume made from perspex, driven by a variable-frequency piston wavemaker built from Lego. Using mobile-phone video capture, we collect experimental data and compare it with predictions from linear gravity-capillary wave theory and with multiphase simulations performed in OpenFOAM. We find excellent quantitative agreement across all three approaches. Our setup may be valuable for students with a background in Mathematical Modelling who lack hands-on laboratory experience. To explore this, we report on a survey of students who completed an integrated theoretical, experimental, and computational project. While students found the experience enhanced their learning of Fluid Mechanics, they also noted the need for better support in setting up and running CFD simulations.

I. INTRODUCTION

The linear theory of water waves is typically introduced to Physics, Engineering, and Applied Mathematics students after more introductory modules have been completed in Calculus of Several Variables, Vector Calculus, and Mathematical Methods. For many students taking a more theoretically-oriented programme of study, the theory remains just that, and students' theoretical understanding is often not matched by a practical lab-based experience. Equally, the simulation of water waves requires the simulation of a multiphase flow problem, which requires sophisticated interface-capturing algorithms often beyond the scope of a typical undergraduate programme. In this paper, we report on an instructional framework which combines theoretical modelling, computational modelling using readymade OpenFOAM case studies, and experiments using a practical, inexpensive flume with a custom-made wavemaker, made from Lego, thereby filling in gaps in students' learning.

A. Literature Review

The study of water waves is important for its practical applications in wave forecasting, coastal engineering [1], and for tsunami warnings. In a first approximation, linear theory is applied, wherein the wave amplitude is small compared to the water depth. Then, the relevant equations of motion (whether the Euler equations for an inviscid fluid, or the

^{*} onaraigh@maths.ucd.ie

Navier–Stokes equations for a viscous one) simplify into linear partial differential equations. Conditions at the interface typically involving the kinematic condition (namely that the interface moves with the fluid) are then used to close the equations. In many textbooks [2–4], a temporal analysis is performed, wherein an initial sinusoidal disturbance on the interface is introduced with wavenumber k. This is done via a prescribed initial condition. Then, the linear equations of motion reduce to an eigenvalue problem, in which the wave frequency is obtained as a function of k, which gives rise to the dispersion relation, $\omega = \omega(k)$. The dispersion relation may depend on other fluid parameters as well. For instance, for water waves in a fluid of finite depth h, the standard dispersion relation is [3]:

$$\omega(k) = \left[k \left(g + \frac{\gamma}{\rho} k^2 \right) \tanh(kh) \right]^{1/2}, \tag{1}$$

where g is the acceleration due to gravity, ρ is the liquid density, and γ is the surface tension.

A more advanced analysis is concerned with a time-dependent localized disturbance is introduced to the system, and the downstream propagation of the disturbance is considered. Such spatio-temporal analysis forms the basis of wavemaker theory [5], and is also important in understanding absolute and convective instability in more advanced problems [6, 7]. A spatio-temporal formulation of the water-wave problem can help to bridge the gap to these research-level problems, and is therefore introduced in this paper.

In order to validate the dispersion relation (1) (or its spatio-temporal analogue), researchers carry out experiments in a wave tank. Referring back to Equation (1), gravity waves refer to waves for which $g \gg (\gamma/\rho)k^2$ (hence, long waves), and these are often of most interest to researchers due to the aforementioned applications. To observe such waves in an experiment, long channels are required. To ensure unidirectional wave propagation, a long narrow channel is used, such that the waves are two-dimensional. Such a setup is referred to as a flume. One such example is the laboratory-scale flume at the University of Warwick [8] with dimensions 22 m long, 0.6 m wide, and 1.0 m deep. In contrast, capillary waves are short and occur when $g \ll (\gamma/\rho)k^2$. Such capillary waves appear as nonlinear, secondary disturbances which occur as a perturbation with respect to a primary gravity wave. They are also of interest in the classroom setting in so-called ripple tanks, which are small (typically, less than $0.5 \,\mathrm{m} \times 0.5 \,\mathrm{m}$ and as such, support only capillary waves [9]. In this work, we document the construction of a tabletop flume, of length 1 m, which supports waves for which both gravity and surface tension play an important role in the dispersion relation.

Numerical simulations also provide important insights into water waves. In the first instance, they can be used to validate the linear theory. They provide a surrogate for experiments, such that the flow generated by the interfacial waves can be inferred, without having to resort to experimental techniques such as PIV. Finally, simulations can be used to provide insights into nonlinear waves, including wave breaking and wave overturning, which cannot be described by linear theory. While classical water-wave theory is inviscid and relies on the Euler equations, many computational fluid dynamics (CFD) software frameworks rely on the Navier–Stokes equations, which include the effect of viscosity. However, provided the viscosity is small (in a sense to be determined), water-wave modelling with the Navier–Stokes equations will provide the same answer was water-wave modelling with the Euler equations.

B. Aim of the paper

The aim of the paper is to introduce students and instructors to an inexpensive piece of apparatus and complementary open-source fluid simulation software, which together provide a hands-on way to investigate the linear theory of free-surface waves. Crucially, with high-quality video recording now available on most mobile phones, students already carry a powerful data acquisition tool in their pockets. We show how this technology can be used to capture wave motion in the flume, enabling direct comparison between experimental observations, theoretical predictions, and numerical simulations, thereby making a complex topic more tangible for students.

C. Plan of the paper

The theoretical spatio-temporal analysis is set out in Section II for open wave tanks and in Section III for closed wave tanks. The open wave tank admits travelling waves whereas the closed wave tank admits a standing wave. As a by-product, this analysis provides a derivation of Equation (1). To illustrate the theory, experiments using a tabletop flume are carried out in Section IV. We carry out statistical analysis on the data emanating from the experiments to show that the observed wave pattern is a linear combination of travelling waves and standing waves. To illustrate the wave phenomena in more detail, and to provide a rigorous test of the dispersion relation (1), computational fluid dynamics simulations are presented in Section V. The experience of students using this three-fold approach to learning about water waves is described in Section VI. Concluding remarks are presented in Section VII.

II. SPATIO-TEMPORAL ANALYSIS OF SMALL-AMPLITUDE WATER WAVES: THE OPEN TANK

In this section we develop the spatio-temporal theory of small-amplitude water waves, for an open tank. This theory describes the linear response of the free surface to a localized forcing corresponding to a wavemaker and as such, forms the basis of wavemaker theory. The theory has already been presented in the standard reference [5] and is included here for completeness, and to provide the proper context for the subsequent experimental and computational investigations.

For this purpose, we refer to the set-up in Figure 1, and take the direction of propagation along the x-axis, and the direction of oscillation along the z-axis. The figure describes an open tank, in which $x \in [0, \infty)$. The free surface is therefore denoted by $z = \eta(x, t)$, where z = 0 represents the undisturbed free-surface height. Standard undergraduate texts describe a temporal theory [3], where the free surface is initialized to have a monochromatic sinusoidal profile $\eta(x, t = 0) \propto \sin(kx + \varphi)$ everywhere (here, φ is a constant phase term). Here, we describe in detail the spatio-temporal theory, wherein the free surface is assumed to be undisturbed initially, but to undergo a localized forcing at x = 0 corresponding to the impact of a piston wavemaker.

To understand the setup of the spatio-temporal wave propagation, we refer to Figure 1. A piston located at x = 0 generates localized, impulsive forcing. The piston oscillates according

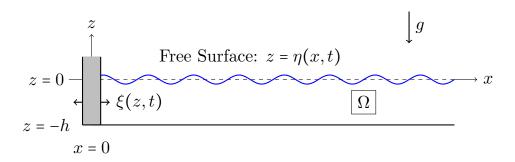


FIG. 1. Schematic diagram showing the generation of small-amplitude water waves by a piston wavemaker located at x = 0 (open tank)

to:

$$\xi(z,t) = \Re\left[-\frac{1}{\mathrm{i}\omega}f(z)\mathrm{e}^{-\mathrm{i}\omega t}\right]. \tag{2}$$

where f(z) is a shape function describing the details of the back-and-forth motion of the piston. This can be left unspecified for now. Inside the domain Ω , the flow is inviscid and irrotational, so potential theory applies:

$$\nabla^2 \Phi = 0, \qquad \boldsymbol{x} \in \Omega. \tag{3}$$

Here, Φ is the velocity potential, such that $\boldsymbol{u} = \nabla \Phi$. Also, the vector $\boldsymbol{x} = (x, z)$ is a two-dimensional vector. The boundary condition at z = -h is the no-penetration condition, w = 0, hence:

$$\frac{\partial \Phi}{\partial z} = 0, \qquad z = -h. \tag{4}$$

A. Conditions at the free surface

We next look at the boundary condition at the free surface $z = \eta$. Bernoulli's equation gives the pressure on the free surface as:

$$p = -\rho \frac{\partial \phi}{\partial t} - \frac{1}{2}\rho \mathbf{u}^2 - \rho g \eta + f(t), \tag{5}$$

where f(t) is a parameter associated with Bernoulli's principle. We assume that the wave amplitude is small in comparison to the water depth h. This introduces a small parameter $\epsilon = \max(\eta)/h$ into the problem. Thus, disturbances, whether of amplitude, pressure, velocity or streamfunction are proportional to ϵ , whereas products of disturbances (such as u^2) are proportional to ϵ^2 and can be neglected in a small-amplitude approximation. Thus, the pressure on the free surface can be approximated as:

$$p = -\rho \frac{\partial \Phi}{\partial t} - \rho g \eta + f(t). \tag{6}$$

From Reference [3], the pressure condition at the interface for an inviscid flow is:

$$p_{atm} - p = \gamma \kappa, \tag{7}$$

where $\kappa = \eta_{xx}/(1+\eta_x^2)^{3/2}$ is the mean curvature and p_{atm} is the atmospheric pressure. In the small-amplitude approximation, we have:

$$p_{atm} - p = \gamma \eta_{xx}. \tag{8}$$

Using Equation (6), this becomes:

$$\rho \frac{\partial \Phi}{\partial t} + \rho g \eta + [p_{atm} - f(t)] = \gamma \eta_{xx}, \qquad z = \eta.$$
 (9)

Since f(t) is arbitrary, we set $f(t) = p_{atm}$, leaving:

$$\rho \frac{\partial \Phi}{\partial t} + \rho g \eta = \gamma \eta_{xx}, \qquad z = \eta. \tag{10}$$

However, we may expand $\Phi(z = \eta) = \Phi(z = 0) + (\partial \Phi/\partial z)_{z=0}\eta + O(\eta^2)$. Because of the small-amplitude approximation, we can replace $\Phi(z = \eta)$ with $\Phi(z = 0)$, and similarly for derivatives, giving

$$\rho \frac{\partial \Phi}{\partial t} + \rho g \eta = \gamma \eta_{xx}, \qquad z = 0. \tag{11}$$

The difference between Equations (10) and (11) is subtle but it enables a great simplification in the foregoing analysis.

We now make the standard transformations:

$$\Phi = \Re \left[\phi(\mathbf{x}) e^{-i\omega t} \right], \tag{12a}$$

$$\eta = \Re \left[\widehat{\eta}(x) e^{-i\omega t} \right]. \tag{12b}$$

We henceforth drop the hat on $\widehat{\eta}$. Thus, we use the same symbol for η (which depends on x and t), and $\widehat{\eta}$ (which depends on x only). It should be clear from context which variable is being used. In this way, Equation (11) becomes:

$$\rho i\omega \phi = \rho g \eta - \frac{\gamma}{\rho} \eta_{xx}, \qquad z = 0. \tag{13}$$

A second interfacial condition is the kinematic condition. In the small-amplitude approximation, which states that the interface moves with the flow, hence:

$$\frac{\partial \eta}{\partial t} + u \frac{\partial \eta}{\partial x} = w, \qquad z = \eta. \tag{14}$$

As with Equation (11), we linearize this identity on to the surface z = 0, which gives:

$$\frac{\partial \eta}{\partial t} = w \qquad z = 0, \tag{15}$$

or $\partial_t \eta = \partial_z \phi$ on z = 0, hence:

$$-i\omega\eta = \frac{\partial\phi}{\partial z}, \qquad z = 0. \tag{16}$$

We combine Equations (13)–(16). First, Equation (16) gives $\eta = -1/(i\omega)\phi_z$. We substitute this into Equation (13) to obtain a single boundary condition at z = 0:

$$\omega^2 \phi = g \frac{\partial \phi}{\partial z} - \frac{\gamma}{\rho} \partial_{xx} \frac{\partial \phi}{\partial z}, \qquad z = 0.$$
 (17)

B. Solving Laplace's Equation

We solve $\nabla^2 \phi = 0$ in the linearized domain $\Omega_{\mathcal{L}} = \{(x, z) | -h < z < 0\}$. We do separation of variables to get $\phi(x, z) = X(x)Z(z)$. Following standard steps, we get:

$$\frac{X''}{X} = -\frac{Z''}{Z} = k^2. {18}$$

We look at the boundary conditions at z = 0 next. The boundary condition (17) gives:

$$\omega^2 X(x) Z(0) = \left(gX(x) - \frac{\gamma}{\rho} X''(x) \right) Z'(0). \tag{19}$$

We use the separation-of-variables condition (18) to reduce this to:

$$\omega^2 Z(0) = \left(g - \frac{\gamma}{\rho} k^2\right) Z'(0). \tag{20}$$

We further re-write this as:

$$Z'(0) = \alpha_k Z(0), \qquad \alpha_k = \frac{\omega^2}{g - \frac{\gamma}{\rho} k^2}.$$
 (21)

Putting it all together, we have to solve:

$$Z'' + k^2 Z = 0, (22a)$$

$$Z'(-h) = 0, (22b)$$

$$Z'(0) = \alpha_k Z(0). \tag{22c}$$

The solution is:

$$Z = \frac{\cos[k(z+h)]}{\cos kh},\tag{23}$$

with solvability condition $k \tan(kh) = -\alpha_k$, or:

$$k\tan(kh) = -\frac{\omega^2}{g - \frac{\gamma}{\varrho}k^2}.$$
 (24)

We label the solutions of Equation (25) as k_n , where $n \in \{0, 1, 2, \dots\}$.

C. Dispersion Relation

Equation (24) has two solution types:

• Case 1. This corresponds to n = 0, so we are dealing with k_0 . In this case, k_0 is purely imaginary, and we write $k_0 = \pm i\kappa$, where κ is real. Using the properties of trigonometric functions, Equation (25) reduces to:

$$\kappa \tanh(\kappa h) = \frac{\omega^2}{g + \frac{\gamma}{\rho} \kappa^2},\tag{25}$$

which is precisely Equation (1). In this case, however, ω is known, and κ has to be obtained by inversion. A sample dispersion curve is shown in Figure 2.

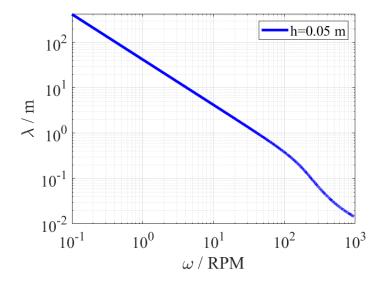


FIG. 2. The dispersion relation (25). For a given ω , there is a uniquely determined k-value, hence a uniquely determined wavelength $\lambda = 2\pi/k$. Parameter values: $h = 0.05 \,\mathrm{m}$, $\rho = 1000$, kg · m⁻³, $g = 9.8 \,\mathrm{m} \cdot \mathrm{s}^{-2}$, $\gamma = 0.072 \,\mathrm{N} \cdot \mathrm{m}^{-1}$.

• Case 2. In this case, we look at k_n , where $n \ge 1$. A standard graphical eigenvalue analysis shows in this case there are infinitely many real positive roots, confirming that $n \in \{1, 2, \dots\}$.

Putting the two cases together, we have the following set of eigenfunctions, with Z(z) being replaced by $\chi_n(z)$:

$$\chi_n(z) = \begin{cases} \frac{\cos[k_n(z+h)]}{\cos k_n h}, & n \ge 1\\ \frac{\cosh[\kappa(z+h)]}{\cosh \kappa h}, & n = 0. \end{cases}$$
 (26)

As these are eigenfunctions of a self-adjoint operator, we have an orthogonality relation

$$\int_{-h}^{0} \chi_m(z) \chi_n(z) dz = C_n \delta_{nm}. \tag{27}$$

In particular,

$$C_0 = \frac{1}{4\kappa} \frac{1}{\cosh^2(\kappa h)} \left[2\kappa h + \sinh(2\kappa h) \right]. \tag{28}$$

D. General Solution

The general solution for the velocity potential can now be written as:

$$\phi(x,z) = \sum_{n=1}^{\infty} a_n \chi_n(z) e^{-k_n x} + a_0 \chi_0(z) e^{i\kappa x}.$$
 (29)

Notice that we do not allow for a contribution proportional to $e^{-i\kappa x}$, as this would correspond to a wave travelling inward from positive infinity, which is not physical. Furthermore, for

a bounded solution, we rule out contributions that depend on $e^{k_n x}$. Thus, the Sommerfeld Radiation condition $\partial \phi / \partial x \sim ik\phi$ is satisfied as $x \to \infty$. Furthermore, at x = 0, we have:

$$\left(\frac{\partial \phi}{\partial x}\right)_{(x=0,z)} = \sum_{n=1}^{\infty} a_n \chi_n(z) (-k_n) + a_0 \chi_0(z) (\mathrm{i}\kappa). \tag{30}$$

The boundary condition at x = 0 is $\partial_x \phi = u = \partial_t \xi$, where ξ is the displacement of the wall at x = 0 (cf. Equation (2)). Thus, we obtain:

$$\sum_{n=1}^{\infty} a_n \chi_n(z) (-k_n) + a_0 \chi_0(z) (i\kappa) = f(z).$$
 (31)

Hence, the coefficients a_0 and a_n can be determined from:

$$a_0 = \frac{1}{(i\kappa)C_0} \int_{-h}^0 f(z)\chi_0(z)dz,$$

$$a_n = \frac{1}{(-k_n)C_n} \int_{-h}^0 f(z)\chi_n(z)dz, \qquad n \ge 1.$$

In particular, for a piston wavemaker with $f(z) = f_0 = \text{Const.}$, we have:

$$a_0 = \frac{f_0}{(i\kappa)C_0} \frac{1}{\kappa} \frac{\sinh(\kappa h)}{\cosh(\kappa h)}.$$
 (32)

Furthermore, in the far field, we have

$$\phi \sim a_0 \chi_0(z) e^{i\kappa x}, \qquad x \to \infty,$$
 (33)

since $e^{-k_n x} \to 0$ as $x \to \infty$, for $n \ge 1$. Only the oscillatory wave with dispersion relation (25) survives far downstream of the disturbance.

E. Results of summary calculations

By analysing the dispersion relation (25), we can see what type of wavelengths can be expected for a given forcing frequency. The wavelengths depend on depth, as shown in Table I.

ω (RPM)	$\lambda (h = 0.05 \mathrm{m})$	$\lambda (h = 0.1 \mathrm{m})$
10	4.19	5.91
50	0.820	1.13
100	0.381	0.484
200	0.139	0.142

TABLE I. Expected wavelengths (in metres), based on the dispersion relation (25). Depths: $h = 0.05 \,\mathrm{m}$ and $0.1 \,\mathrm{m}$. Other parameters as in Figure 2.

A further key quantity of interest is the height-to-stroke ratio, which we derive now for the piston wavemaker as follows. We apply the kinematic condition (16) in the far field (for $x \to \infty$) to get

$$a_0 \left(\frac{\partial \chi_0}{\partial z} \right)_{z=0} = -i\omega \eta_0. \tag{34}$$

Here, we have decomposed $\eta(x)$ into a phase η_0 and the complex exponential $e^{i\kappa x}$, corresponding to the n=0 normal mode. We fill in for $\chi_0(z)$ (cf. Equation (26)) to get

$$\frac{f_0}{(i\kappa)C_0} \frac{\sinh^2(\kappa h)}{\cosh^2(\kappa h)} = -i\omega\eta_0.$$
 (35)

For a piston wavemaker, we have $f_0 = \omega A e^{i\varphi}$, where A is the amplitude of the back-and-forth motion of the piston (and equal to half the stroke, 2A = S), and φ is a constant phase. This gives:

$$\left| \frac{\eta}{A} \right| = \frac{1}{\kappa C_0} \frac{\sinh^2(\kappa h)}{\cosh^2(\kappa h)},\tag{36}$$

and filling in for C_0 gives:

$$\left| \frac{\eta_0}{A} \right| = \frac{4 \sinh^2(\kappa h)}{2\kappa h + \sinh(2\kappa h)}.$$
 (37)

We identify the height of the wave $H = 2|\eta_0|$, hence $|\eta/A| = |2\eta/(2A)| = H/S$. This gives the required height-to-stroke ratio in the far field, valid for a piston wavemaker:

$$\frac{H}{S} = \frac{4\sinh^2(\kappa h)}{2\kappa h + \sinh(2\kappa h)}.$$
 (38)

III. SPATIO-TEMPORAL ANALYSIS OF SMALL-AMPLITUDE WATER WAVES: THE CLOSED TANK

In this section, we introduce wavemaker theory for a closed tank, in which $x \in [0, L]$, as shown schematically in Figure 3. The notation is the same as Section II. The only difference is that a no-penetration boundary condition now applies at x = L:

$$\frac{\partial \phi}{\partial x} = 0, \qquad x = L. \tag{39}$$

The left-hand boundary condition is unchanged from Section II, namely $\partial_x \phi = \partial_t \xi$, at x = 0. For a piston wavemaker, this amounts to:

$$\frac{\partial \phi}{\partial x} = f_0, \qquad x = 0. \tag{40}$$

The other boundary conditions are unchanged from before.

A. Cosine Transform

We take Laplace's Equation $\nabla^2 \phi = 0$ in the linearized domain $\Omega_{\mathcal{L}} = \{(x, z) | -h < z < 0, 0 < x < L\}$, multiply by $\cos(n\pi x/L)$ and integrate with respect to x from x = 0 to x = L. Here, $n \in \{0, 1, \dots\}$. Applying the boundary conditions (39) and (40), we obtain:

$$\frac{\mathrm{d}^2 \widehat{\phi}_n}{\mathrm{d}z^2} - k_n^2 \widehat{\phi}_n = f_0. \tag{41}$$

We solve this equation subject to the no-penetration boundary condition $d\hat{\phi}_n/dz = 0$ at z = -h and the dynamic boundary condition (17) at z = 0. This gives:

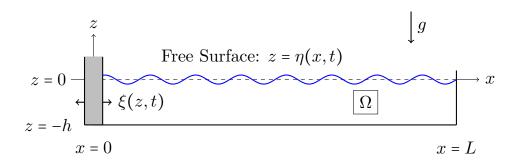


FIG. 3. Schematic diagram showing the generation of small-amplitude water waves by a piston wavemaker located at x = 0 (closed tank)

• The case n = 0:

$$\widehat{\phi}_0 = \frac{gf_0h}{\omega^2} + f_0hz + \frac{1}{2}f_0z^2, \qquad \omega \neq 0.$$
 (42)

• The case n > 0:

$$\widehat{\phi}_n = A_n \cosh[k_n(z+h)] - \frac{f_0}{k_n^2},\tag{43}$$

where

$$A_n = \frac{\omega^2 f_0}{k_n^2 \cosh(k_n h) \left\{ \omega^2 - \left[g + (\gamma/\rho) k_n^2 \right] k_n \tanh k_n \right\}}.$$
 (44)

Equations (42)–(44) are valid off-resonance, that is, provided

$$[g + (\gamma/\rho)k_n^2]k_n \tanh k_n \neq \omega^2, \tag{45}$$

and provided also that $\omega \neq 0$. On resonance, the original trial solution (12) which sets $\Phi = \Re \left[\phi(\boldsymbol{x})e^{-i\omega t}\right]$ and $\eta = \Re \left[\widehat{\eta}(x)e^{-i\omega t}\right]$, is no longer valid. In this case, the solution needs to be derived by carefully following the steps in a Laplace-transform calculation [10]. This yields a resonant solution which grows algebraically in time.

B. General Solution

We use the inverse cosine transformation to write the general solution for $\phi(x,z)$:

$$\phi(x,z) = \frac{1}{L} \left[\frac{gf_0h}{\omega^2} + f_0hz + \frac{1}{2}f_0z^2 \right] + \frac{2}{L} \sum_{n=1}^{\infty} \frac{\omega^2 f_0}{k_n^2 \cosh(k_n h) \left\{ \omega^2 - \left[g + (\gamma/\rho) k_n^2 \right] k_n \tanh k_n \right\}} \cosh[k_n(z+h)] \cos(n\pi x/L).$$
 (46)

The free surface is given by $\eta = [-1/(i\omega)\phi_z]_{z=0}$, hence:

$$\eta(x,t) = \Re\left[\frac{\mathrm{i}f_0}{\omega}\mathrm{e}^{-\mathrm{i}\omega t}\right] + \frac{2}{L}\sum_{n=1}^{\infty} \frac{\Re(\mathrm{i}f_0\omega\mathrm{e}^{-\mathrm{i}\omega t})}{k_n\cosh(k_nh)\left\{\omega^2 - \left[g + (\gamma/\rho)k_n^2\right]k_n\tanh k_n\right\}} \sinh(k_nh)\cos(n\pi x/L). \tag{47}$$

C. Approximate Solution

In this section, we introduce an approximate solution of Equation (47) which contains only one wavenumber. This is obtained simply by taking the most-dominant component of the sum in Equation (47):

$$\eta(x,t) = \Re\left[\frac{\mathrm{i}f_0}{\omega}\mathrm{e}^{-\mathrm{i}\omega t}\right] + \frac{2}{L} \frac{\Re(\mathrm{i}f_0\omega\mathrm{e}^{-\mathrm{i}\omega t})}{k_n \cosh(k_{n_0}h) \left\{\omega^2 - \left[g + (\gamma/\rho)k_{n_0}^2\right]k_{n_0} \tanh k_{n_0}\right\}} \sinh(k_{n_0}h) \cos(n_0\pi x/L), \quad (48)$$

where n_0 is the solution of

$$n_0 = \arg\min_{n \in \mathbb{N}} J_n,$$
where $J_n = k_n [\cosh(k_n h)/\sinh(k_n h)] \left\{ \left[g + (\gamma/\rho)k_n^2 \right] k_n \tanh k_n - \omega^2 \right\}.$ (49)

Referring to Figure 4, for $\gamma = 0.072 \,\mathrm{N \cdot m^{-1}}$, $\rho = 1000 \,\mathrm{kg \cdot m^{-3}}$, $g = 9.8 \,\mathrm{m \cdot s^{-2}}$, $L = 1 \,\mathrm{m}$, $h = 0.05 \,\mathrm{m}$, and $\omega = 140 \,\mathrm{RPM}$, the integer that minimizes J_n is $n_0 = 8$, corresponding to four maxima in the wave tank. Furthermore, the neighbouring values of $J_{n_0 \pm 1}$ are well separated from J_{n_0} ,

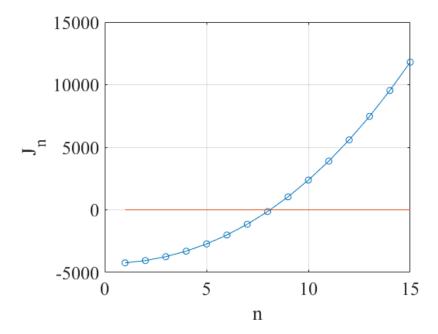


FIG. 4. Plot showing the minimization of J_n at n=8 for the case $\gamma=0.072\,\mathrm{N\cdot m^{-1}},\ \rho=1000\,\mathrm{kg\cdot m^{-3}},\ g=9.8\,\mathrm{m\cdot s^{-2}},\ L=1\,\mathrm{m},\ h=0.05\,\mathrm{m},\ \mathrm{and}\ \omega=140\,\mathrm{RPM}.$

with $|J_{n_0\pm 1}/J_{n_0}| > 7.6$, meaning that Equation (48) is a good first approximation.

IV. EXPERIMENTS USING THE TABLETOP FLUME

In this section we report on a study of waves generated using the tabletop flume described in the introduction. We set out the technical specification of the flume and the associated wavemaker, and report summary results. An in-depth statistical analysis is subsequently carried out, which produces excellent agreement with between the theory in Sections II–III and the measurements.

A. Specification of the Flume and the Wavemaker

The tabletop flume is shown in Figure 5 and consists of a perspex box (open at the top) of length $L=1\,\mathrm{m}$, height $H=0.1\,\mathrm{m}$, and width $W=0.16\,\mathrm{m}$. These measurements are taken with respect to the outer edges of the box. As the perspex has a thickness $4\,\mathrm{mm}$, the inner dimensions of the box are slightly smaller. For added strength, the perspex box is set in a wooden housing. To produce two-dimensional waves, the width of the flume can reduced by way of a simple width-adjuster made up of a wooden beam clamped at both ends to the flume housing. To help with measurements and analysis, rulers on the cm scale have been drawn one on the flume housing (for water depth) and one on the width adjustor (for wavelength) with permanent marker. This enables us to do quantitative video analysis at a later stage. A photograph of the setup is shown in Figure 5. A piston wave-maker

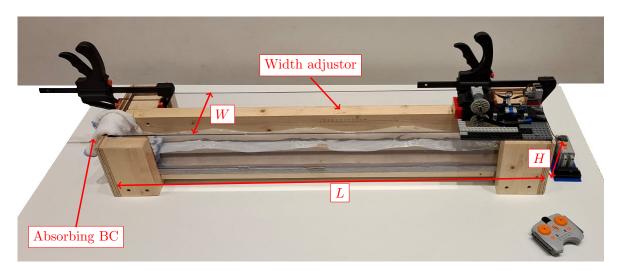


FIG. 5. Photograph of the tabletop flume with wavemaker attached

is secured to the flume housing as shown Figure 5. The back-and-forth action of the flap which generates the wave motion is shown in Figure 6. At the opposite end of the flume, an absorbing boundary condition is applied by the simple expedient of draping a cloth over the end, as shown also in the figure.

The wavemaker is a piston wavemaker made from Lego Tecnic components. Detailed instructions to make the wave-maker are provided in the accompanying GitHub repository [11]. The wave-maker consists of five parts, highlighted as follows and shown separately in Figure 7:

1. Electric Motor;



FIG. 6. A zoom-in on Figure 5 showing the back-and-forth action of the flap (yellow Lego board) which generates the wave motion.

- 2. Battery pack;
- 3. Variable RPM controller (parts (3a) and (3b) in Figure 6);
- 4. Transmission system, with 1:1 gear ratio;
- 5. Oscillating piston;
- 6. Housing.

This Lego wavemaker is screwed to the wooden housing which encases the perspex box.

B. Preliminary Results

We present here a first set of results based on the setup shown in Figure 5. The water depth is set as $h = 5.0 \,\mathrm{cm}$, correct to the nearest millimetre. Hence, to avoid spurious precision, we report final results here to two significant figures only. Intermediate measurements may be reported to higher precision, if available. A video recording was made of the experiment using a Samsung Galaxy A55. To enable further analysis and reproducibility, the video has been posted on YouTube [12]. The recording was analyzed on a frame-by-frame basis using the VideoReader function in Matlab. From this analysis, the frame rate of video is obtained: $r = 29.86 \,\mathrm{fps}$. From a representative frame (Frame 7 in the video, see Figure 8), the wavelength is measured to be $0.28 \pm 0.01 \,\mathrm{m}$. Correspondingly, $k = (2\pi/\lambda) \pm \Delta k$, where $\Delta k = k(\Delta \lambda/\lambda)$, hence $k = (22 \pm 1) \,\mathrm{m}^{-1}$.

Although the setup shown in Figures 5–7 allows for variable RPM of the piston wave-maker, given the load on the wavemaker, the RPM is not known a priori. For this reason, we have also measure the frequency of the wavemaker from the video analysis: one cycle of the piston is measured to take 13 ± 1 frames, the measurement error here coming from the uncertainty in identifying which is the last frame in one cycle of the piston. These data then give the period of the piston wavemaker as $T = \lceil (13 \pm 1)/r \rceil$ s, where ΔT is identified here

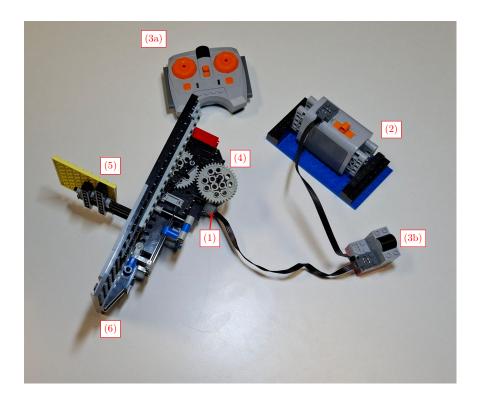


FIG. 7. Photograph showing the different components Lego wavemaker

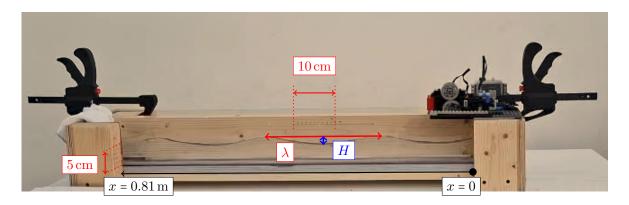


FIG. 8. Snapshot of the wavemaker experiment, Frame 7 from Reference [12]. A re-centred x-axis is shown; the location of x = 0 is at the black dot.

as (1/r)s. The frequency is therefore calculated as:

$$\omega = \frac{2\pi}{T} \left(1 \pm \frac{\Delta T}{T} \right) \text{ rad} \cdot \text{s}^{-1}, \tag{50}$$

hence $\omega = 140 \pm 10 \, \text{RPM}$.

We jump the gun slightly and endeavour to connect these measurements back to the theory of travelling waves in Section II (full justification below – in the in-depth statistical analysis). For this purpose, we refer the reader to Table II. From this table, we identify a range of wavelengths $I_{theory} = [0.22, 0.27]$ m consistent with the travelling-wave theory. Furthermore, we identity $I_{measured} = [0.27, 0.29]$ m as the range of wavelengths consistent with

the experimental measurements. Since $I_{theory} \cap I_{measured}$ is non-empty, the the experimental results are consistent with the theoretical analysis.

$\lambda(\omega = 130 \text{RPM})$	$\lambda(\omega = 140 \text{RPM})$	$\lambda(\omega = 150 \text{RPM})$
$0.27\mathrm{m}$	$0.25\mathrm{m}$	$0.22\mathrm{m}$

TABLE II. Range of wavelengths consistent with the linear theory. Water depth: $h = 0.05 \,\mathrm{m}$ and all other parameters the same as in Figure 2.

We have also measured the wave height as $H = (1.8 \pm 0.3)$ cm and the stroke length of the piston as $S = (2.4 \pm 0.2)$ cm, giving $(H/S)_{\text{inferred}} = 0.8 \pm 0.2$. From Equation (38) for travelling waves (with $\lambda = 0.28$ m), we have $(H/S)_{\text{theory}} = 1.09 \pm 0.04$. Thus, the measured value of the height-to-stroke ratio is not consistent with the theoretical value. To understand this discrepancy, we analyze our results in more detail in what follows.

C. In-depth statistical analysis

To understand the results in the video in more detail, we have digitized the first 120 frames and made a record of the interface height on a frame-by-frame basis, and stored the results in a space-time array $\eta(x,t)$. Here, $x \in [0,0.81]$ is the coordinate along the horizontal direction in the test section shown in Figure 8, and t is time. The resulting data have been made available in the accompanying GitHub repository [11]. To avoid spurious precision, we again report the results to two significant figures only. Intermediate parameter values may be reported to higher precision. Data are stored at discrete spatial coordinates x_i and discrete times t_i . We plot the results of this digitization procedure in Figure 9.

From Figure 9, a well-defined train of travelling waves can be picked out. Using non-linear least squares, we can fit a functional form

$$\eta_{model}(x,t) = h_0 + A\cos(\omega t - kx + \varphi) \tag{51}$$

to the data. The non-linear least squares problem is solved by minimizing the cost function

$$J(h_0, A, \omega, \phi) = \sum_{i} \sum_{j} [\eta_{model}(x_i, t_j) - \eta(x_i, t_j)]^2.$$
 (52)

The values that minimize the cost function are the 'fitting parameters', these are given in Table III. The upper and lower bounds are obtained by statistical bootstrapping, and correspond to the 2.5% and 97.5% confidence intervals generated using that method. The Matlab files used for the bootstrapping are made available in the accompanying GitHub repository [11]. A space-time plot of the model profile is shown in Figure 10. The fitted value of k implies $\lambda = 2\pi/k \in [0.25, 0.27]$ m = $I_{measured}$, with a best estimate $\lambda = 0.27$ m. This is in the same range as our previous summary measurement of the wavelength in Figure 8.

Next, we calculate the theoretical wavelength for travelling waves, evaluated at the best-estimate frequency $\omega = 14.7\,\mathrm{rad\cdot s^{-1}} = 140\,\mathrm{RPM}$; this is $\lambda = 0.25\,\mathrm{m}$. Using the confidence intervals in Table III, we are able to account for the spread in the estimated value of ω . These are inputted into the travelling-wave theory to produce an allowed interval of wavelengths $I_{theory} = [0.24, 0.25]\,\mathrm{m}$ (Table IV). Since the overlap $I_{theory} \cap I_{measured}$ is non-empty, the travelling-wave theory is consistent with the measurements.

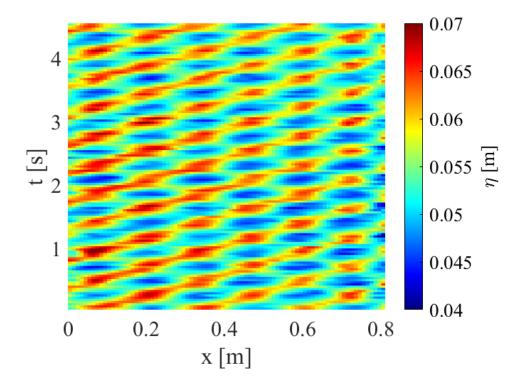


FIG. 9. Plot of the free-surface height η as a function of space and time

	h_0 [m]	A [m]	$\omega \left[\mathrm{rad} \cdot \mathrm{s}^{-1} \right]$	$k \left[\text{m}^{-1} \right]$	φ
Best Estimate	0.0500	0.0134	14.7	23.3	0.000
Lower Bound	0.0499	0.0083	14.5	23.3	0.510
Upper Bound	0.0502	0.0168	14.9	25.4	0.819

TABLE III. Optimal fitting parameters, model (51)

Notwithstanding the accurate measurement of λ obtained from fitting the model (51) to the data, there is a clear visual mismatch between Figure 9 (experiment) and 10 (model). In the experimental data, there is clear evidence of a standing wave, in addition to the travelling wave. Therefore, to obtain better agreement between the model and the data, we perform non-linear least squares again, and we fit a functional form

$$\eta_{model}(x,t) = h_0 + A_1 \cos(\omega t - k_1 x + \varphi_1) + A_2 \cos(\omega t + \varphi_2) \cos(k_2 x) + A_3 \cos(\omega t + \varphi_3)$$
 (53)

to the data. The non-linear least squares problem is solved by minimizing a cost function analogous to Equation (52). The fitting parameters are given in Table V. The upper and lower bounds are again obtained by statistical bootstrapping, and correspond to the 2.5%

$\lambda(\omega = 138 \mathrm{RPM})$	$\lambda(\omega = 140 \text{RPM})$	$\lambda(\omega = 142 \text{RPM})$
$0.25\mathrm{m}$	$0.25\mathrm{m}$	$0.24\mathrm{m}$

TABLE IV. Range of wavelengths consistent with the linear theory (updated, using the model (51)). Water depth: $h = 0.050 \,\mathrm{m}$ and all other parameters the same as in Figure 2.

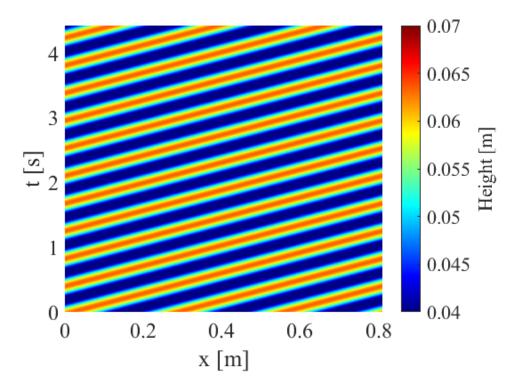


FIG. 10. Plot of the model free-surface height η_{model} (Equation (51)) as a function of space and time

	h_0 [m]	A_1 [m]	$\omega \left[\mathrm{rad} \cdot \mathrm{s}^{-1} \right]$	$k_1 \left[\mathrm{m}^{-1} \right]$	φ_1
Best Estimate	0.0500	0.0129	14.7	24.6	0.152
Lower Bound	0.0499	0.00531	14.4	23.4	0.000
Upper Bound	0.0501	0.0154	14.9	26.5	0.750

	A_2 [m]	$k_2 \left[\mathrm{m}^{-1} \right]$	φ_2	A_3 [m]	φ_3
Best Estimate	0.00503	22.19	2.74	0.00308	3.36
Lower Bound	0.00304	15.7	1.37	0.00280	3.14
Upper Bound	0.00617	28.1	5.22	0.00350	4.02

TABLE V. Best-fit parameters, model (53). The lower bounds are obtained from the 2.5% and 97.5% confidence intervals.

and 97.5% confidence intervals generated using that method. Results are shown in Figure 11. There is much better qualitative agreement between Figure 9 (experiment) and 11 (updated model), than there is between the experiment and the previous travelling-wave-only model. Therefore, the experimental evidence is that the wave profile in the tank is a linear superposition of a travelling wave (obtained by waves traveling to the end of the tank and being absorbed by the cloth), and a standing wave (obtained by waves reflected back-and-forth between both ends of the tank).

Referring to Table V, $\lambda = 2\pi/k \in [0.24, 0.27]\,\mathrm{m} = I_{measured}$, with a best estimate $\lambda =$

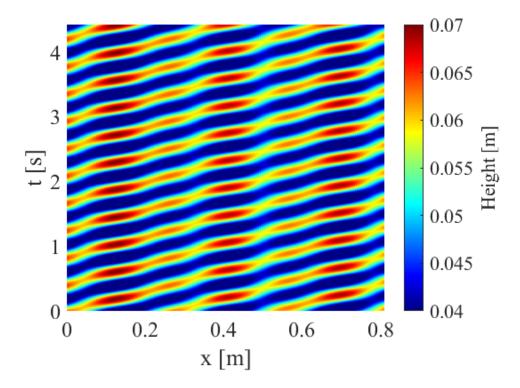


FIG. 11. Plot of the model free-surface height η_{model} (Equation (53)) as a function of space and time

0.26 m, consistent with the previous fitted model (51). Repeating earlier, similar calculations, we calculate the theoretical wavelength (travelling waves) evaluated at the best-estimate frequency $\omega = 14.7 \,\mathrm{rad \cdot s^{-1}} = 140 \,\mathrm{RPM}$; this is $\lambda = 0.25 \,\mathrm{m}$. We again account for the spread in the fitted value of ω , using the ω -confidence intervals in Table V and the results in Table IV, hence $I_{theory} = [0.24, 0.25] \,\mathrm{m}$. Again, the overlap $I_{theory} \cap I_{measured}$ is non-empty, so the measurements are consistent with the theory.

We look at the height-to-stroke formula in the context of the fitted model (53). The appropriate value of the height is now the height of the travelling-wave component, hence $H = 2A_1$. We use $S = (2.4 \pm 0.2)$ cm, as previously. Hence, the inferred height-to-stroke ratio is $(H/S)_{\text{inferred}} = 1.07$. However, allowing for the uncertainty in the fitted value of A_1 and the measured value of S, we have $(H/S)_{\text{inferred}} \in [0.41, 1.29]$. From Equation (38) with $\lambda = 0.25$ m, we have $(H/S)_{\text{theory}} = 1.21$, so the measured and theoretical values are consistent.

D. Discussion

The observed waveform in Figure 9 clearly consists of a superposition of a travelling wave and a standing wave. The theory in Section II describes a travelling wave only and can be realised by having a perfectly absorbing boundary condition at the far end of the wave tank, absorbing all outgoing waves. The opposite extreme is the theory in Section III, which describes a standing wave. This setup can be realised by having a perfectly reflecting boundary condition at the far end of the tank.

It is clear from the results in Figure 9 that the cloth placed at the end of the wave tank does a good job of absorbing outgoing waves but is not perfect. Therefore, the boundary condition at x = L is not known a priori but is such that the allowed waves in the tank are a superposition of standing and travelling waves. Thus, the two complementary theories in Section II and III are required to explain the observed waveform. This is the inspiration for the fitted model in Equation (53).

Referring back to the fitted model in Equation (53), the estimate for the standing-wave wavenumber k_2 is in the range $k_2 \in [15.7, 28.1] \,\mathrm{m}^{-1}$. Referring to the standing-wave theory in Section III, we identify this as $k_2 = n\pi/L$. This gives $n \in [5, 9]$, which is consistent with the predicted value of the most-dominant standing-wave mode in Section III $(n = 8, \mathrm{Figure} \ 4)$.

In the next section we will look at a computational model for the wave tank which allows for much more precise control of the outgoing waves and hence, a much sharper travellingwave solution in the (numerical) wave tank.

V. COMPUTATIONAL FLUID DYNAMICS USING OPENFOAM

To understand the waves generated by the tabletop flume in more detail, we perform numerical simulations using a two-phase flow algorithm in the open-source finite-volume code OpenFOAM. In this section, we report on our findings. We first give a broad description two-phase flow modelling to wave modelling, we describe the implementation of the model in OpenFOAM and then present our results.

A. Volume-of-Fluid Method

We describe here the Volume-of-Fluid (VoF) method to model two-phase flow. The method is based on a one-fluid formulation of the two-phase Navier–Stokes equations, first introduced by Brackbill [13]. In such a one-fluid formulation, one solves a single set of Navier–Stokes equations for a single fluid, the properties of which transition sharply across a zone corresponding to the interface in the classic two-fluid formulation. An advantage of this approach is that one does not have to solve separate Navier–Stokes equations in the different phases, which would be computationally challenging and would require the prescription of complicated matching conditions across the interface separating the phases.

To build such a one-fluid formulation, one starts with an indicator function $\chi(x)$ which tracks the phases:

$$\chi(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \text{ is in the liquid phase,} \\ 0 & \text{if } \boldsymbol{x} \text{ is in the gas phase,} \end{cases}$$
 (54)

Since $\chi(x)$ is a step function, it cannot be differentiated numerically. Hence, we introduce a smoothened version:

$$\alpha(\boldsymbol{x}) = \frac{1}{V} \int_{V} \chi(\boldsymbol{x}') d^{3}\boldsymbol{x}', \tag{55}$$

where V is a small test volume. Thus, in the Volume-of-Fluid formulation, $\alpha(x)$ is used to track the phases, with $\alpha = 0.5$ indicating the interphase between the phases. In this way, we can model the multiphase medium as a single fluid with variable density and viscosity

as follows:

$$\rho(\mathbf{x}) = \rho_L \alpha(\mathbf{x}) + [1 - \alpha(\mathbf{x})] \rho_G, \tag{56a}$$

$$\mu(\boldsymbol{x}) = \mu_L \alpha(\boldsymbol{x}) + [1 - \alpha(\boldsymbol{x})] \mu_G. \tag{56b}$$

Here, ρ_L and μ_L are the constant liquid density and viscosity, and similarly for ρ_G and μ_G in the gas.

In this way, we can write the Navier–Stokes equations in a single-fluid formulation as follows:

$$\rho(\boldsymbol{x}) \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) = -\vec{\nabla} p + \vec{\nabla} \cdot \left[\mu(\boldsymbol{x}) \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T \right) \right] - \rho(\boldsymbol{x}) g \widehat{\boldsymbol{z}} + \boldsymbol{F}_{ST}, \tag{57a}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{57b}$$

Here, $-\rho(x)g\hat{z}$ is the body force due to gravity, pointing in the negative z-direction. Furthermore, F_{ST} is the approximation to the surface-tension force in the VoF formulation, and is given by [13]:

$$\mathbf{F}_{ST} = -\gamma \nabla \alpha \left[\nabla \cdot \left(\frac{\nabla \alpha}{|\nabla \alpha|} \right) \right]. \tag{58}$$

The rationale behind this approximation is as follows. In a two-fluid formulation of the Navier–Stokes equations, the surface tension is represented either by interfacial matching conditions across the phases or equivalently, by a force localized to the interface and hence, expressed in terms of a delta function. In the latter formulation, one would have:

$$\mathbf{F}_{ST} = \gamma \kappa \widehat{\mathbf{n}} \delta(\mathbf{x} - \mathbf{x}_I), \tag{59}$$

where x_I denotes the interface location. Since $\alpha = 1/2$ in the volume-of-fluid formulation describes the interface location, we have:

$$\delta(\boldsymbol{x} - \boldsymbol{x}_I) = |\nabla \alpha| \delta\left(\alpha - \frac{1}{2}\right). \tag{60}$$

Hence:

$$\widehat{\boldsymbol{n}}\,\delta(\boldsymbol{x}-\boldsymbol{x}_I) = \nabla\alpha\,\delta\left(\alpha - \frac{1}{2}\right). \tag{61}$$

Furthermore, $\kappa = -\left[\nabla \cdot \left(\nabla \alpha / |\nabla \alpha|\right)\right]_{\alpha=1/2}$, hence:

$$\mathbf{F}_{ST} = -\gamma \nabla \cdot \left(\frac{\nabla \alpha}{|\nabla \alpha|} \right) \nabla \alpha \, \delta \left(\alpha - \frac{1}{2} \right). \tag{62}$$

A final step in the VoF approximation of the surface-tension is to 'smear' the force over the entire fluid volume by omitting the delta function in Equation (61). In this way, Equation (58) is recovered.

The smearing of the surface-tension force over the entire fluid volume is justified because α is approximately constant away from the interface, meaning that the expression F_{ST} in Equation (58) becomes negligible far from the interface. However, this approximation can in some applications introduce 'spurious currents' into the numerical model [14], which can be mitigated by a careful discretization of the Navier–Stokes equations, and by carrying out simulations at high resolution.

Finally, conservation of mass of the separate phases requires that:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\boldsymbol{u}\alpha) = 0. \tag{63}$$

Since the flow is incompressible, this is equivalent to:

$$\frac{\partial \alpha}{\partial t} + \boldsymbol{u} \cdot \nabla \alpha = 0. \tag{64}$$

To summarize, the relevant equations of motion to be solved are: Equations (57a)–(57b), and Equation (64). These are highly complicated coupled partial differential equations. To solve these efficiently, we use the OpenFOAM software toolbox.

B. OpenFOAM

OpenFOAM is a C++ toolbox for the development of customized numerical solvers for continuum mechanics problems, in particular Computational Fluid Dynamics (CFD). However, the toolbox comes equipped with a range of already-built solvers and tutorials, meaning that simple test-case simulations can be set up with reduced effort (i.e. compared to developing custom-built solvers). One such solver is interFOAM, which enables one to simulate two-phase flow problems based on Equations (57) and (64).

The key elements of an OpenFOAM transient CFD simulation can be explained by reference to the directory structure of a sample simulation with directory name case, shown in Figure 12. We show this here because it paves the way for subsequent discussion on the treatment of boundary and initial conditions. For the same purpose, we summarize the contents of the sub-directories as follows:

- 1. Sub-directory system contains the following files:
 - (a) controlDict: Simulation parameters are set, e.g. the end-time of the simulation, the frequency at which simulation data is outputted to files.
 - (b) fvSchemes: The algorithms for the solution Equations (57) and (64) are chosen.
 - (c) fvSolution: Further algorithms for the solution Equations (57) and (64) are chosen (e.g. the time-marching scheme).
 - (d) setFieldsDict: Initial conditions on α , u, and p are prescribed.
 - (e) blockMeshDict: Specifications for the computational domain are given. The geometry of the fluid domain is described, together with the grid resolution and any local grid refinement that is required. Boundary conditions on the various faces making up the domain are also supplied. Once the mesh parameters are specified, a mesh is generated using the blockMesh command. The resulting mesh is then copied into the directory polyMesh. All domain dimensions are given in metres.
- 2. Sub-directory constant contains the following files:
 - (a) transportProperties: The surface tension, liquid and gas densities and viscosity ate set using S.I. units.
 - (b) turbulenceProperties: Turbulence modelling, if required, is specified.
 - (c) g: The magnitude and direction of the gravity vector is specified.

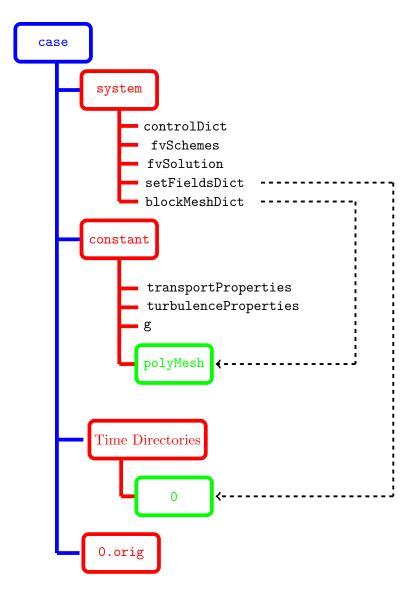


FIG. 12. Directory structure for a typical OpenFOAM simulation. The broken lines with arrow-heads show the effect of running the setFields and blockMesh commands on various directories.

3. Time Directories:

- (a) 0: An initial configuration is copied from 0.orig into 0. The initial configuration is over-written with custom initial conditions using the setFields command. This sets the state of the system $(\alpha, \mathbf{u}, \text{ and } p)$ at t = 0. The original initial conditions are kept always in 0.orig so that the simulation can be recreated from scratch, if needs be.
- (b) Further directories corresponding to the state of the system $(\alpha, \mathbf{u} \text{ and } p)$ at later times t_1, t_2, \ldots are created here when the simulation is run. The values t_1, t_2, \cdots are set in controlDict.

A simulation such as the one outlined here is executed in OpenFOAM using the interFOAM command. We do not elaborate further on the use of interFOAM here as our main purpose here is to define reference points for subsequent discussion of initial and boundary conditions.

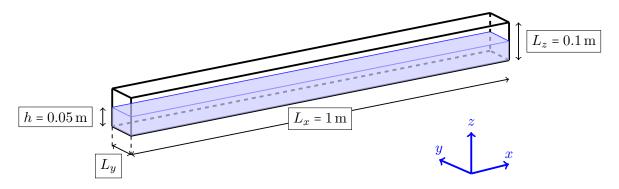


FIG. 13. Schematic diagram showing the geometry of the simulation, together with the initial condition for the water level. The unit normal $\hat{\mathbf{y}}$ referred to in the text points in the positive y-direction.

Instead, the interested reader is referred to the rich online resources on OpenFOAM for further instruction (e.g. Reference [15], specifically on interFOAM).

C. olaFLOW

We set up a simulation in a simple rectangular domain such as the one shown in Figure 13. We apply standard no-slip conditions at the bottom wall and standard atmospheric conditions at the top of the domain. Simple 'empty' boundary conditions are applied in the faces whose normals are $\pm \hat{\mathbf{y}}$, these enforce a two-dimensional flow. A simple uniform mesh is created based on this domain using the blockMesh command in OpenFOAM. However, we encounter difficulty when we seek to apply a time-varying boundary condition at the inlet and a wave-damping boundary condition at the outlet.

Rather than writing OpenFOAM code from scratch to describe an oscillating inlet, we resort to a suite of OpenFOAM cases called olaFlow which has been developed to simulate flows in wave tanks using the Volume-of-Fluid methodology. The olaFlow codes are available on an online repository [16] and a simple tutorial corresponding to a two-dimensional piston wavemaker in a flume is already set up (specifically, wavemakerFlume). Detailed instructions on how to download and execute olaFlow are provided A. To model the tabletop flume, we use the tutorial case wavemakerFlume with our own custom initial conditions and geometry, which we describe as follows.

- Geometry: The geometry of the simulation shown in Figure 13, to match the tabletop flume shown in Figures 5–7. We use $L_x = 1.0 \,\mathrm{m}$ to match the tabletop flume, as well as $h = 0.05 \,\mathrm{m}$. The domain in the z-direction is chosen to be $L_z = 10 \,\mathrm{cm}$, comprising liquid in the bottom half and gas (air) in the top half. We also use $L_y = 2 \,\mathrm{cm}$ in the y-direction. Although this L_y -value does not match the tabletop flume, this is not important, as we seek to simulate a fundamentally two-dimensional wave problem. This choice of narrow channel width in the simulation, together with 'empty' boundary conditions enforces a two-dimensional flow.
- Mesh Resolution: The computational mesh is a uniform one, with 500 cells in the x-direction, 50 cells in the z-direction, and 1 cell in the y-direction, corresponding to a two-dimensional flume. We have verified that this resolution is adequate by performing a mesh-refinement study (below).

- Initial Conditions: These are zero velocity in the water and in the air, and a flat interface, corresponding to $\alpha = 1$ in the liquid and $\alpha = 0$ in the water. The initial conditions are specified in the 0.orig directory in Figure 12 and in the setFields dictionary.
- Boundary Conditions: Boundary conditions are specified in the 0.orig directory and in the blockMeshDict dictionary. By using olaFlow, we are able to set up the inlet as a boundary wall which oscillates back and forth with a set frequency. The oscillating inlet is implemented in the olaFlow code pistonWaveGen.py. We have modified this code so that the inlet location $x = \xi$ is a simple function of time,

$$\xi(t) = (S/2)\cos(\omega t),\tag{65}$$

where S and ω are set by the user. This enables us to create a simulation which mimics the tabletop flume as closely as possible. Finally, the outlet wall is set up so as to absorb outgoing waves, using the olaFlow outlet boundary type waveAbsorption2DVelocity as boundary condition for the velocity. This absorbing boundary condition is described in the PhD thesis of Higuera [17] which provides the theoretical underpinning for olaFlow.

D. Mesh-Refinement Study

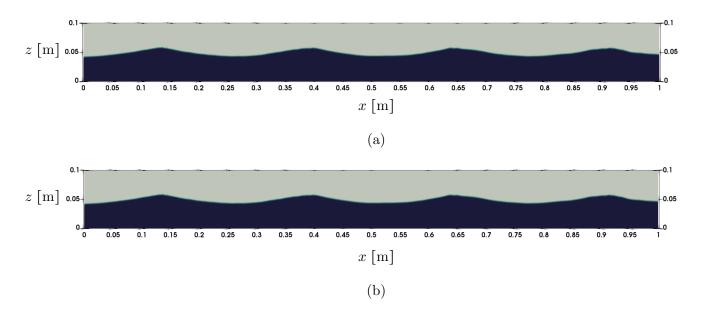


FIG. 14. Mesh-refinement study: snapshot (xz-plane) of the free surface at t = 8.5 s. Top: standard mesh (N_x, N_y, N_z) = (500, 1, 50). Bottom: fine mesh (N_x, N_y, N_z) = (1000, 1, 100). Simulation parameters: S = 1.1 cm, $\omega = 138$ RPM.

For the present purposes, the reference mesh is uniform blockmesh of size $(N_x, N_y, N_z) = (500, 1, 50)$. Here, we report very briefly on simulation results where the blockmesh has been increased to (1000, 1, 100). We carry out simulations for a stroke length S = 0.011 m and a forcing period T = 0.435 s (hence, $\omega = 138$ RPM). A comparison between the standard mesh

and the fine mesh is shown in Figure 14. There is no difference between the two results, confirming that the simulation can be deemed to have converged when using the reference mesh.

E. First results

Representative results results are shown in Figure 15, for a test case with $T = 0.45 \,\mathrm{s}$ (hence, $\omega = 133 \,\mathrm{RPM}$, close to the value in the experiment in Section IV), and $S = 0.01 \,\mathrm{m}$. The results represent a snapshot taken at $t = 8.5 \,\mathrm{s}$, such that the piston wavemaker has had sufficient time to produce a fully-developed wave train. Results have been visualized with Paraview, which interfaces very easily with OpenFOAM.

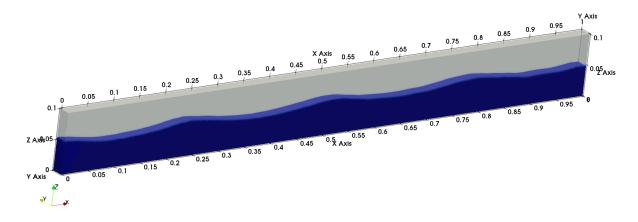


FIG. 15. Representative result: snapshot of the free surface at $t=8.5\,\mathrm{s}$. Simulation parameters: $S=1.0\,\mathrm{cm},\,\omega=133\,\mathrm{RPM}.$ Axes scales in metres.

A single monochromatic wave can be identified in Figure 15. We can visualize this more precisely by taking a two-dimensional planar slice. The plane has a normal vector in the y-direction and is centred at $L_y/2$. By extracting the free surface at each point in time, and plotting the result in a space-time plot, we confirm that the monochromatic wave is a travelling wave (Figure 16).

Remark In this section we shift the origin in the z-direction, such that the mean water level is at $z = h_0$. Thus, the free surface is represented by $\eta(x,t) = h_0 + [\text{Disturbance}]$.

To analyze the travelling wave more precisely, we again focus the snapshot at t = 8.5 s, well after transient effects have died away. A planar slice of the snapshot is shown in Figure 17. We have fitted a sinusoidal curve to the snapshot, $\eta(x) = h_0 + A\sin(kx + \varphi)$, and the result is $k = 23.12 \,\mathrm{m}^{-1}$ and $A = 0.0066 \,\mathrm{m}$, corresponding to $\lambda = 0.272 \,\mathrm{m}$. The predicted value of the wavelength according to linear theory (cf. Figure 2, $\omega = 2\pi/0.45 \,\mathrm{rad \cdot s^{-1}}$) is $\lambda = 0.265 \,\mathrm{m}$, Furthermore, the measured value of H/S is H/S = 2a/S = 1.07. This is to be compared with the linear theory (cf. Equation (38)), which predicts H/S = 1.15 at $\lambda = 0.265 \,\mathrm{m}$. The agreement here between the numerical simulations and the linear theory can be considered very close. Discrepancies between the simulations and the theory can be accounted for by the fact that the simulated wave is not perfectly monochromatic: secondary oscillations at shorter wavelength can be seen in Figure 17, especially at wave crests and troughs. This can be explained by nonlinear effects. We address this in more detail below.

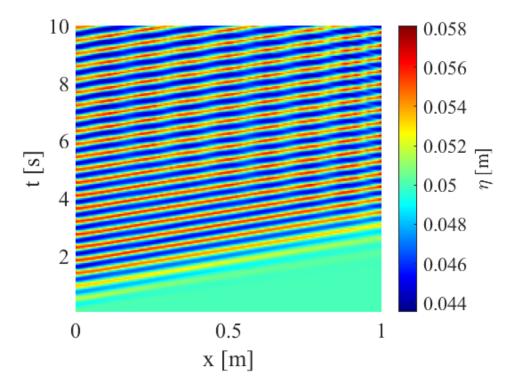


FIG. 16. Spacetime plot of the free surface $\eta(x,t)$ (simulation results). Simulation parameters: $S = 1.0 \,\mathrm{cm}$, $\omega = 133 \,\mathrm{RPM}$.

F. Height-to-stroke ratio

We have extended the study in the previous section to a range of different stroke lengths S. We have thereby computed the dependence of the wave height on S. The results are summarized in Figure 18. For small values of S, there is good agreement between the linear theory (broken red line) and the simulations. At $S \approx 14\,\mathrm{mm}$, significant wave steepening occurs, and the linear theory breaks down. Beyond this point, the waves steepen significantly, as seen in the inset in Figure 18.

VI. STUDENT EXPERIENCES

The theoretical modelling of linear water waves (similar to Section II–III) was introduced to students in ACM 40890 Advanced Fluid Mechanics at University College during the 2024–2025 academic year. This is an advanced module, taken by final-year Bachelor of Science students majoring in Theoretical Physics and Applied Mathematics, as well as by first-year PhD students. Ten students took the module in 2024-2025, in the Spring Trimester (12 weeks). Alongside lectures on theory, students completed a substantial assignment involving an experiment with the tabletop flume, as well as CFD simulations of same, using olaFlow.

To assess student perspectives on this integrated approach, a survey was administered in April 2025 via an anonymous Google Form, following approval from the University's Human Research Ethics Committee. Participation was voluntary; 5 anonymous responses

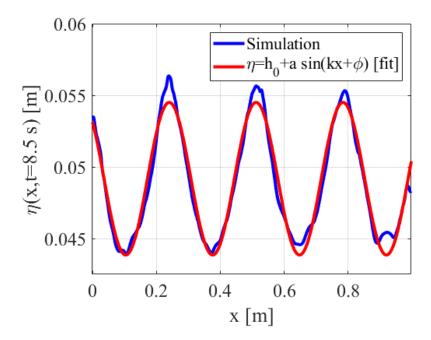


FIG. 17. Snapshot of the free-surface height at $t=8.5\,\mathrm{s}$. Simulation parameters: $S=1.0\,\mathrm{cm}$, $\omega=133\,\mathrm{RPM}$.

were obtained. While the sample size is too small for statistical analysis, the feedback offers useful qualitative insights for instructors considering similar module designs.

Overall, students reported that the integration of theory, experiment, and simulation enhanced their understanding of complex fluid mechanics concepts. One student remarked:

My perception of experiments in Fluid Mechanics is they are insightful and provide an opportunity to solidify and gain a greater understanding the various topics we studied in Fluid Mechanics. It gives an opportunity to tweak variables and see how they affect the outcome and gain a more intuitive understanding of many phenomena than studying the Mathematics alone does not achieve.

Students also highlighted the value of simulations in bridging theory and experiment:

I learned a lot from how we used the data from the experiments to help us set up the simulations and compare the theory, experiments, and simulations. As someone who never really liked experiments and always liked simulations I never really saw the connection between the two until now. I found great learning in that.

On a cautionary note, the students noted the steep learning curve involved in setting up the CFD simulations, particularly given the lack of dedicated technical support. One student reflected:

CFD simulations require a broad skill-set; learning to use various applications (many of which lack documentation), programming, data analysis of large data sets, and understanding of the underlying workings of algorithms to understand when/why it does not behave the way you expect and if it is a syntax issue,

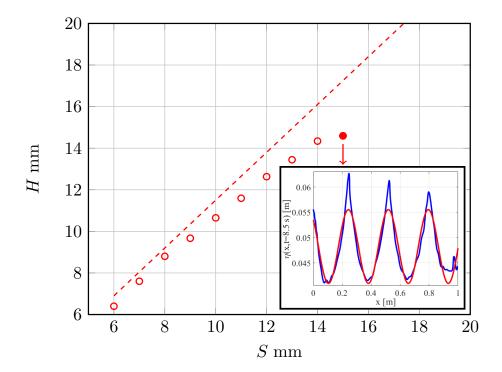


FIG. 18. Dependence of wave height H on the stroke length S of the piston wavemaker. Circles: numerical simulations. Red line: H/S = 1.15, from linear theory, with $\lambda = 0.265 \,\mathrm{m}$. The filled data-point corresponds to $S = 15 \,\mathrm{mm}$, when the linear theory clearly breaks down. The inset shows a snapshot of the free-surface height at the same stroke length, and exhibits significant wave steepening.

physics issue, a known limitation or a bug. It is a steep learning curve and working through these issues comes comes with learning and you usually don't make the same error twice. Though to make my point, I found various other hurdles in conducting CFD simulations limited the ability to focus to the Fluids Mechanics of the scenarios being studied.

This was echoed by other students in offline feedback. Overall, the students felt that more formal lectures in CFD (involving detailed cases studies) would be beneficial, as opposed to relying on a mixture of theory-heavy face-to-face lectures and posted online CFD tutorials.

Finally, the instructor observed in the completed reports on the project that while students proved adept at extracting summary information from the experimental data, few were able to see beyond the summary information, and to go down to the level of detail required for a complete understanding of the observed wave forms. While some of this must certainly be due to time constraints on the students' part, in future, the instructor should place more emphasis on data analysis techniques, to help students develop deeper insights from experimental results.

VII. CONCLUSIONS

Summarizing, we have described in detail the theory of small-amplitude water waves in the limit where the equations of motion can be linearized. The theory is presented is spatio-temporal, as it describes the response of the system in space and time to localized inlet forcing. As a by-product the classical dispersion relation for linear water waves (Equation (1)) is recovered.

As part of an integrated approach to instruction in the topic, we have developed – and present here – an inexpensive 1 m-long tabletop flume and variable-RPM Lego wavemaker, which can be used to gather experimental data. With the advent of high-quality mobile-phone cameras such data can be readily recorded and analysed. Hence, we report on a representative experiment. Using nonlinear least squares fitting to perform a statistical analysis on the data, we have found excellent agreement between the theory and the experiments, once the presence of both travelling waves and standing waves is accounted for. Future student work on the flume and the wave-maker could focus on the boundary conditions – for instance, on engineering a purely reflective boundary condition, which would also open up the possibility of finding resonant modes.

To complement the theoretical and experimental work, we have performed numerical simulations of the wave tank using OpenFOAM. In the simulations, it is much more straightforward to provide an absorbing boundary condition, and hence to produce a train of clearly defined travelling waves, whose parameters agree with the predictions of the linear theory. For small stroke lengths of the piston wavemaker, the results of the numerical simulations for the height-to-stroke ratio are also in good agreement with the theory.

Finally, we have reported also on the results of incorporating such an integrated approach – involving theory, experiments, and numerical simulation in the classroom setting. Feedback from the students showed that while students found the experience enhanced their learning, they also noted the need for better support in setting up and running CFD simulations.

Acknowledgments

LÓN thanks Maria Meehan for advice on making a submission to the UCD Human Research Ethics Committee. NY acknowledges support by the UCD School of Mathematics and Statistics through a funded summer research placement.

Appendix A: Installation of olaFlow

In this work, we use the olaFlow to simulate the generation of water waves. These are a suite of codes which can be run in the OpenFOAM computational framework. Downloading and installing olaFlow is not trivial, as successful implementation of olaFlow relies on the use of an earlier version of OpenFOAM. Hence, in this Appendix we outline the steps necessary to install olaFlow and execute simple test cases. We describe this for users using a Windows Operating System, however, most of the commands for these tasks will be the same in Ubuntu.

First, we install an Ubuntu virtual machine which will run in a Windows environment, this is done using the Windows Power Shell and the command:

```
wsl --install -d Ubuntu-18.04
```

We now initialize Ubuntu 18.04 by typing the following line into the Windows Power Shell terminal:

```
wsl -d Ubuntu-18.04
```

Next, we install the appropriate version of OpenFOAM, which is OpenFOAM6. The appropriate commands are obtained from https://openfoam.org/download/6-ubuntu/ and are repeated here:

```
sudo sh -c "wget -0 - https://dl.openfoam.org/gpg.key | apt-key
   add -"
sudo add-apt-repository http://dl.openfoam.org/ubuntu
```

Update the apt package list:

```
sudo apt-get update
```

Install OpenFOAM6:

```
sudo apt-get -y install openfoam6
```

Update the installation:

```
sudo apt-get update
sudo apt-get upgrade
```

At the end of the installation, we must append the following line to our bashrc file (located in Ubuntu-18.04\home\username):

```
% . /opt/openfoam6/etc/bashrc
source /opt/openfoam6/etc/bashrc
```

We close the terminal and open a new one. We then start a new virtual Ubuntu session. By virtue of having modified the bashrc, OpenFoam 6 is automatically loaded. This can be checked by typing:

```
simpleFoam -help
```

We are now able to download olaFlow. This can be done using the clone command in GitHub. Alternatively, the installation can be downloaded as a zip file from:

```
https://github.com/phicau/olaFlow.
```

This will produce the zip file olaFlow-master.zip, which we move in to the relevant directory and unzip. We next modify the permissions on the unzipped folder:

```
sudo chmod -R 777 olaFlow-master
```

We install 'make':

```
sudo apt-get install make
```

We can now use the OpenFOAM command 'all make' to generate the appropriate olaFlow executables. First:

```
cd olaFlow-master ./allMake
```

Then:

```
cd genAbs
./allMake
```

To run the various codes that generate transient boundary conditions in olaFlow, we require Python. Hence, we install pip:

```
sudo apt install python-pip
```

and finally, numpy:

```
sudo pip install numpy
```

Once these installations are complete, olaFlow is ready to be used. For instance, to run a simple flume model from the tutorials, we change into the relevant directory:

```
    cd tutorials

    cd wavemakerFlume
```

The relevant OpenFOAM case can be run using the appropriate file. For instance, for the piston wave maker, we type:

```
. runCasePiston
```

The details concerning this case (mesh generation, initial conditions, etc.) can be found by opening the file of the same name.

- [1] Ole Secher Madsen. Waves generated by a piston-type wavemaker. In *Coastal Engineering* 1970, pages 589–607. 1970.
- [2] Richard P Feynman, Robert B Leighton, Matthew Sands, and Everett M Hafner. The Feynman lectures on Physics; Vol. i. *American Journal of Physics*, 33(9):750–752, 1965.
- [3] David J Acheson. Elementary fluid dynamics. Oxford University Press, 1990.
- [4] CS Jog. Fluid Mechanics: Volume 2: Foundations and Applications of Mechanics. Cambridge University Press, 2015.
- [5] Robert G Dean and Robert A Dalrymple. Water Wave Mechanics for Engineers and Scientists, volume 2. World Scientific Publishing Company, 1991.
- [6] Patrick Huerre and Peter A Monkewitz. Local and global instabilities in spatially developing flows. *Annual review of fluid mechanics*, 22(1):473–537, 1990.
- [7] Lennon Ó Náraigh, Peter DM Spelt, and Stephen J Shaw. Absolute linear instability in laminar and turbulent gas—liquid two-layer channel flow. *Journal of Fluid Mechanics*, 714:58–94, 2013.
- [8] S Dong, M Salauddin, Soroush Abolfathi, and Jonathan M Pearson. Improved prediction of wave overtopping rates at vertical seawalls with recurve retrofitting. *Ocean Engineering*, 302:117647, 2024.
- [9] Institute of Physics teaching guidance: Ripple tank and accessories. https://spark.iop.org/ripple-tank-and-accessories. [Online; accessed 31-December-2024].
- [10] Jaw-Fang Lee, Jonq-Ren Kuo, and Chung-Pan Lee. Transient wavemaker theory. *Journal of hydraulic research*, 27(5):651–663, 1989.

- [11] Lennon Ó Náraigh. waveFlume: GitHub Repository. Zenodo. doi: 10.5281/zenodo.15920448, 2025.
- [12] Lennon Ó Náraigh. Tabletop Flume. https://youtu.be/ahXxWVJAG-E?si=94A29LsMp5fun8jn, 2024. [Online; accessed 7-July-2025].
- [13] Jeremiah U. Brackbill, Douglas B. Kothe, and Charles Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335–354, 1992.
- [14] Zlatko Solomenko, Peter DM Spelt, Lennon Ó Náraigh, and Pascal Alix. Mass conservation and reduction of parasitic interfacial waves in level-set methods for the numerical simulation of two-phase flows: A comparative study. *International Journal of Multiphase Flow*, 95:235–256, 2017.
- [15] József Nagy. Multiphase (VOF) simulation project (OpenFOAM tutorials). https://youtube.com/playlist?list=PLcOe4WUSsMkF93STOampeX-U8hhgbFRbm&feature=shared, 2017. [Online; accessed 20-December-2024].
- [16] Pablo Higuera. phicau/olaFlow: CFD for waves. Zenodo. doi: 10.5281/zenodo.1297013, 2018.
- [17] Pablo Higuera Caubilla. Aplicación de la dinámica de fluidos computacional a la acción del oleaje sobre estructuras / Application of computational fluid dynamics to wave action on structures. PhD thesis, Universidad de Cantabria, 2015.