Coupling 3D Liquid Simulation with 2D Wave Propagation for Large Scale Water Surface Animation Using the Equivalent Sources Method

Camille Schreck, Chris Wojtan

EUROGRAPHICS 2022 / R. Chaine and M. H. Kim
(Guest Editors)

Volume 41 (2022), Number 2

Coupling 3D Liquid Simulation with 2D Wave Propagation for Large Scale Water Surface Animation Using the Equivalent Sources Method

Camille Schreck[1,2] and Chris Wojtan[1]

[1]IST Austria
[2] Université de Lorraine, CNRS, Inria, LORIA, France

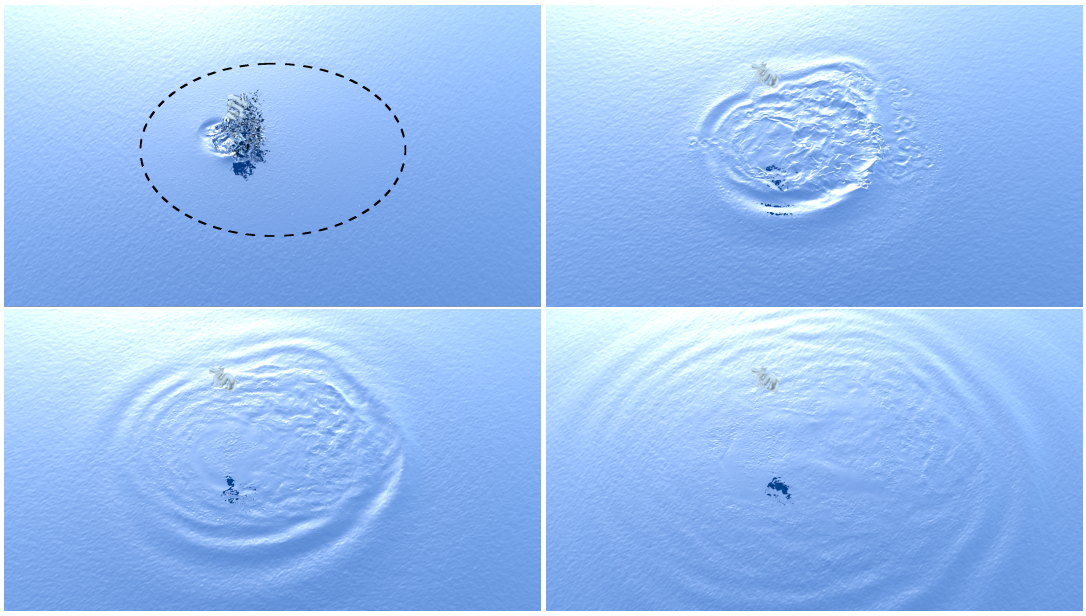


Figure 1: *A dragon springing out of water. Our method couples a FLIP fluid simulation and 2D wave propagation. The dotted ellipse shows the limit between the two.*

Abstract

This paper proposes a method for simulating liquids in large bodies of water by coupling together a water surface wave simulator with a 3D Navier-Stokes simulator. The surface wave simulation uses the equivalent sources method (ESM) to efficiently animate large bodies of water with precisely controllable wave propagation behavior. The 3D liquid simulator animates complex non-linear fluid behaviors like splashes and breaking waves using off-the-shelf simulators using FLIP or the level set method with semi-Lagrangian advection.

We combine the two approaches by using the 3D solver to animate localized non-linear behaviors, and the 2D wave solver to animate larger regions with linear surface physics. We use the surface motion from the 3D solver as boundary conditions for 2D surface wave simulator, and we use the velocity and surface heights from the 2D surface wave simulator as boundary conditions for the 3D fluid simulation. We also introduce a novel technique for removing visual artifacts caused by numerical errors in 3D fluid solvers: we use experimental data to estimate the artificial dispersion caused by the 3D solver and we then carefully tune the wave speeds of the 2D solver to match it, effectively eliminating any differences in wave behavior across the boundary. To the best of our knowledge, this is the first time such a empirically driven error compensation approach has been used to remove coupling errors from a physics simulator.

Our coupled simulation approach leverages the strengths of each simulation technique, animating large environments with seamless transitions between 2D and 3D physics.

1. Introduction

Progress in fluid simulation has made water scenes in animated movies and games increasingly realistic. However, the simulation of very large scenes with convincing visual details is still extremely expensive and challenging. This paper proposes an efficient method for simulating large bodies of water like a sea by splitting the domain into regions of complex and simple motions: We simulate complex fluid motions like splashes, breaking waves, and droplets with a fully 3D Navier-Stokes solver, while we simulate the calmer areas of water with a 2D heightfield solver based on fundamental solutions of the Helmholtz equation [SHW19].

Unlike previous methods for coupling 3D and 2D fluid animation techniques, our approach avoids the use of a fixed grid for wave simulation, which allows us to simulate arbitrarily detailed waves and infinitely large bodies of water. This 2D solver is based on Airy wave theory, which faithfully reproduces the relative speeds of waves in any water depth, not just shallow water. We then couple the simulations together using state-of-the-art techniques for limiting artificial wave reflections where the 2D and 3D domains meet.

Despite our measures to minimize numerical artifacts when coupling the simulations together, we show that differences in numerical errors between wave simulations can to produce distracting numerical artifacts. To solve this problem, we introduce a novel systematic approach to compensate for numerical errors between wave simulations, by matching the effective wave speeds of a 3D fluid solver. Our contributions are the following:

- A method to couple any 3D liquid simulation with a 2D wave propagation method based on fundamental solutions.
- A method to estimate the effective dispersion law for any 3D liquid simulator, which we use to match wave properties between different simulations and eliminate spurious numerical refraction artifacts.

2. State of the Art

2.1. 3D Liquid Simulation

The motion of a liquid surface is typically described by the incompressible Euler Equations:

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\nabla P + \boldsymbol{g} \tag{1}$$

$$\nabla \cdot \boldsymbol{v} = 0 \tag{2}$$

where $\boldsymbol{v}$ is the 3D velocity of the fluid, P is the pressure, and $\boldsymbol{g}$ is the acceleration due to gravity. Researchers in computer graphics solve these equations numerically. Advection is typically handled using a variant of semi-Lagrangian advection [Sta99] or Lagrangian particles [BR86]. The incompressibility constraint in the second equation is enforced by treating pressure as a Lagrange multiplier and solving a Poisson equation [Bri15]. To track the motion of the surface, one can advect an implicit surface using the level set method [OF06], track an explicit surface mesh [WMB11], or reconstruct a surface from FLIP particles [YT13].

Simulating liquids in this manner has immense advantages: it creates highly detailed motions, including non-linear wave propagation and effects like splashes and overturning waves that cannot be captured by a heightfield alone. The main disadvantage of this approach is its computational complexity; detailed visual effects require very fine grids, which require long computation times and large amounts of memory.

To combat this computational complexity, many researchers proposed adaptive simulation methods [MWN*17] which gain speedups and reduce memory by locally reducing simulation detail where it may not be needed. These methods typically use adaptive spatial data structures like octrees [LGF04], warping grids [IWT*18], or meshes [ATW13; BWHT07; WT08; KFCO06] instead of regular grids. Zhu *et al.* [ZLC*13] adaptively stretch out the grid cells away from a central point of interest, effectively extending the size of the computational domain and reducing detail away from the grid center. By adaptively reducing computational degrees of freedom, these methods all make trade-offs between visual detail and computational speed. For surface wave simulation, reducing simulation resolution has the unfortunate side effect of removing detailed water surface ripples with short wavelengths.

2.2. 2D Water Wave Simulation

To reduce computational complexity without removing details at the surface, several researchers investigated the simulation of water waves on top of a flat fluid surface. One approach assumes that the fluid velocity is irrotational, allowing it to be expressed as the gradient of a velocity potential. This assumption significantly simplifies the equation so it can be solved using lower-dimensional degrees of freedom limited to the two-dimensional surface [Tes04; Tes14; CMT*16; JW15; JW17; JSM*18] or even the one-dimensional boundary of the fluid domain [SHW19]. Another approach to the problem assumes that the water is so shallow that the fluid velocity is effectively constant throughout the depth. This assumption gives rise to tall cell methods [IGLF06; CM11] and the shallow water equations (SWE) [LvdP02; WMT07; CM10; SBC*11] which again lead to a computational speed-up due to the reduction in dimensionality from 3D to 2D. The equivalent sources (i.e., "fundamental solutions") approach of Schreck *et al.* [SHW19] also enables infinitely large, open fluid domains. The water wavelength is also independent from simulation degrees of freedom, so the method creates high frequency visual details without increasing the simulation resolution.

Despite their computational efficiency and increased visual detail, these specialized wave simulation approaches have significant limitations. They enforce particular constraints on the velocity of the fluid below the surface (e.g. potential flow or constant with respect to depth), so they are not as expressive as fully three-dimensional simulations. The potential-flow methods follow Airy wave theory (also known as "linear" wave theory) [Air41], so they cannot model non-linear wave propagation. These methods also assume the water surface is a heightfield, so they cannot model breaking waves or flows that make the surface exhibit interesting topology changes.

2.3. Coupling 2D and 3D liquid simulations

Inspired by the strengths of both techniques, several researchers have combined 2D wave simulation methods with 3D fluid solvers. Many methods (ours included) take advantage of the efficiency of 2D solvers to simulate extremely large simulation domains by combining a 3D simulation in the area of interest with a 2D simulation further away. Thuerey *et al.* [TRS06] combines 3D and 2D simulations based on the Lattice Boltzmann method, while Thuerey *et al.* [TMSG07] and Chentanez *et al.* [CM10] use a wave solver on a 2D grid and add 3D details where appropriate. Chentanez *et al.* [CMK15] couples a 3D particle-based simulation with a 2D shallow water simulation for large open scenes, and Huang *et al.* [HQT*21] combines a FLIP-based fluid solver with a surface-only liquid solver using boundary elements.

Other methods leverage the complementary strengths of 2D and 3D methods to add detailed waves directly on top of a 3D fluid surface. Examples in the computer graphics literature solve a wave equation [TWGT10; BLW12; YWTY12; YHW*16], the iWave method [PTM09; KTT13; MBT*15], vortex sheet equations [KSK09; BW13] and simulate wave packets [SSJ*20].

On a related note, some researchers proposed methods for coupling a 3D fluid simulation into an existing fluid environment. Nielsen & Bridson [NB11] propose using an existing fluid simulation as a guide shape for a new 3D fluid simulation; English *et al.* [EQYF13] propose a "chimera grid" strategy for matching fluid simulation variables at domain boundaries; Bojsen-Hansen & Wojtan [BW16] use generalized perfectly matched layers to blend the 3D simulation into the surrounding environment; and Stomakhin *et al.* [SS17] describe how to compute fluid fluxes through the animated boundary of a fluid simulation domain.

Our method couples the 2D wave solver of Schreck *et al.* [SHW19] to the boundary of a 3D fluid simulation. This particular surface wave solver has a number of unique advantages over ones used in the past: First of all, unlike the constant-speed waves produced by SWE and Lattice Boltzmann method wave simulations, [SHW19] is based on Airy wave theory and produces waves with physical behaviors closer to those of the 3D fluid simulation. Secondly, contrary to the methods that use 2D grids or meshes to represent the surrounding water surface waves, our combined method allows fluid domains that are literally infinitely large. Finally, because [SHW19] gives us full control over the dispersion behavior of the waves, so we can actually modify the wave solver's behavior to compensate for numerical errors in the 3D fluid simulation, practically eliminating visual artifacts at the domain boundary.

3. Technical background

3.1. 3D liquid simulation

Our work couples a 3D fluid simulation to a 2D wave solver. We consider 3D fluid simulators based on a regular grid, which represents velocity and pressure fields as functions discretized on the grid, and also tracks the location of the liquid surface. We have implemented our technique into two different methods: the fluid solver implemented in the *Houdini* software by SideFX, which appears to use narrow-band FLIP [FAW*16] for advection and extracts the surface from the FLIP particles [YT13]; and the solver described by Bojsen-Hansen & Wojtan [BW16], which uses semi-Lagrangian advection and represents the liquid surface using the level set method [OF06].

3.2. 2D Water wave simulation

We use the method of Schreck *et al.* [SHW19] for 2D water wave simulation. By assuming that water depth is constant and that waves are not sloped too steeply, they represent the water surface motion with Airy theory [Air41]. The water surface can then be decomposed as a set of sinusoidal frequency components of wave number k –related to the wavelength λ by $k=\frac{2\pi}{\lambda}$. Each component is a function of the form $a\cos(kx-\omega t)$, where a is the amplitude and ω is the angular frequency ($\omega = 2\pi f$ where f is the time frequency). The heightfield is a discrete sum of these frequency components:

$$\eta(\boldsymbol{x},t) = \mathrm{Re}(u(\boldsymbol{x},t)) \text{ and } u(\boldsymbol{x},t) = \sum_k p_k(\boldsymbol{x}) e^{-i\omega_k t}, \tag{3}$$

where p_k is the complex spatial field corresponding to wavenumber k. Each p_k is a solution of the Helmholtz equation: $\nabla^2 p + k^2 p = 0$.

Water waves are dispersive, which means that their angular frequency depends on their wavenumber. This dispersion relation makes their phase speed c_p depend on wavelength as well, via the relationship $c_p = \frac{\omega}{k}$. In particular, gravity waves follow the theoretical relation $\omega_k = \sqrt{gk\tanh(kh)}$ with g being the gravity constant and h being water depth. We refer interested readers to [Joh97] for a more complete introduction to the linear theory of water surface waves.

Schreck *et al.* propose to use the Equivalent Sources Method (ESM) to compute the spatial components p_k^{sc} of the field u^{sc} scattered by the obstacle. The idea of ESM is to approximate a field by a set of n fundamental solutions:

$$p_k^{sc}(\boldsymbol{x}) = \sum_{i=1}^{n} a_{k,i}\phi_k(\boldsymbol{x}-\boldsymbol{x}_i) \tag{4}$$

where the $a_{k,i}$ are complex amplitudes and $\phi_k(\boldsymbol{x}-\boldsymbol{x}_i)$ is a fundamental solution of the Helmholtz equation, or *source*, located at $\boldsymbol{x}_i$. More precisely, this source is a point emitting circular waves and can be explicitly computed as $\phi_k(\boldsymbol{x}-\boldsymbol{x}_i) = -\frac{i}{4}H_0^{(2)}(k||\boldsymbol{x}-\boldsymbol{x}_i||)$ where $H_0^{(2)}$ is the 0^{th} order Hankel function of the second kind. The sources are uniformly set on an offset curve inside the obstacle.

To compute how an incoming wave $u^{in}$is scattered by an obstacle, Schreck *et al.* define the boundary condition at the border of the simulation domain:

$$u^{sc}(\boldsymbol{x},t) = -u^{in}(\boldsymbol{x},t) \tag{5}$$

for all times t and for all $\boldsymbol{x}$ on the boundary. The functions $\left\{e^{-i\omega_k t}\right\}_k$ (defined in Equation 3) are orthogonal, so the boundary conditions are enforced separately for each wave number k. Combining Equation 3 and Equation 5 gives the boundary condition for each k:

$$p_k^{sc}(\boldsymbol{x}) = -p_k^{in}(\boldsymbol{x}) \quad \forall x \text{ on the boundary.} \tag{6}$$

By discretizing Equation 6 at a set of m points $\left\{\boldsymbol{y}_j\right\}_{j=1..m}$ and

combining it with Equation 4, a set of m linear equations with the amplitudes $\{a_{k,i}\}_{i=1..n}$ as unknown is obtained for each k:

$$\sum_{i=1}^{n} a_{k,i}\phi_k(\boldsymbol{y}_j - \boldsymbol{x}_i) = p_k^{in}(\boldsymbol{y}_j) \forall \boldsymbol{y}_j. \quad (7)$$

Schreck *et al.* then solve this using a least squares optimization. The height of the surface can finally be computed using Equation 3 as

$$\eta(\boldsymbol{x},t) = \mathrm{Re}\left(\sum_k \sum_i a_{k,i}\phi_k(\boldsymbol{y}_j - \boldsymbol{x}_i)e^{-i\omega_k t} + u^{in}(\boldsymbol{x},t)\right). \quad (8)$$

To animate aperiodic wave motions, Schreck *et al.* replace each amplitude $a_{k,i}$ by a time-dependent function $a_{k,i}(t)$. The latter is a linear interpolation of a set of discrete amplitudes $\left\{a^h_{k,i} = a_{k,i}(t_T)\right\}$, where t_T is the time after T timesteps. In this case, $p_k(\boldsymbol{x},t) = \sum_i a_{k,i}(t - \boldsymbol{x}/c_g)\phi_k(x - x_i)$, where $c_g = \frac{\partial \omega}{\partial k}$ is the wave's group speed. This method animates time-dependent waves at the cost of solving a system similar to Equation 7 at each time step.

Note that, unlike 2D wave simulations that discretize the heightfield onto a grid, this method analytically defines the wave heights everywhere on $\mathcal{R}^2$. This makes it trivial to represent infinitely large bodies of water and to represent high-frequency ripples without using an adaptive grid representation. In a sense, it maximizes the benefits of 2D wave simulations (efficient computation of large heightfield domains) and is a good candidate for coupling to a 3D simulation for locally enhanced details.

4. Objectives and Overview

The goal of this project is to propose a tool to efficiently simulate a detailed scene on an arbitrarily large water surface. For this we want to use a computationally expensive full 3D fluid simulation only where really needed and a cheaper 2D surface wave propagation using sources everywhere else (see Figure 2).

As waves propagate outward from a disturbance, the law of energy conservation causes their amplitudes to decrease, and these small-amplitude waves are well-described by linear wave theory. We assume that the 3D fluid simulation is large enough not only to capture all 3D non-linear effects, but also such that the waves leaving the 3D simulation domain can be approximated using Airy theory –the water depth and wavelengths are significantly larger than the wave height– (see Figure 3).

The surface in these regions can then be approximated using the equivalent sources method.

The 3D fluid simulation represents only part of a larger surface, so we need to prevent any erroneous internal reflections on the border of the simulation. We use either a perfectly matched layer (PML) method or an increasing viscosity to damp to waves before they reach the end of the simulated domain.

We consider a fluid surface represented by domain $\mathcal{D}$ that can be divided into two subdomains $\mathcal{D}_{3d}$ and $\mathcal{D}_{2d}$ (see Figure 2). $\mathcal{D}_{2d}$ corresponds to the regions where the surface of the fluid can be represented by linear (Airy) waves and displayed on a heightfield. $\mathcal{D}_{3d}$ is the domain where more complex behavior is needed (3D water droplets or breaking waves, for example).

In Section 5, we explain how to ensure a seamless transition between $\mathcal{D}_{2d}$ and $\mathcal{D}_{3d}$ for linear waves radiating from $\mathcal{D}_{3d}$. We build upon [SHW19] and use the state given by the 3D liquid simulation as boundary conditions for the 2D solver. We also discuss how to modify the boundary conditions of the 3D fluid simulation to prevent artificial internal reflections, and to send in new waves from the 2D simulation.

When a physical wave propagates from one medium to another, it can exhibit reflections and refractions. We observe similar effects when coupling two different simulation wave representations; the subtle differences in discretization affect the way that waves propagate and can lead to similar *artificial* wave reflections and refractions when a wave exits one simulation and enters another. This is a common problem with spatially adaptive discretizations, where high-frequency waves cannot be represented by a coarser discretization and cause an internal reflection [Vic81]. Artificial damping and dispersion from the 3D simulation can significantly change the behavior of water surface waves, especially when compared to the theoretically exact behaviors described by linear wave theory. These differences cause significant artificial refraction artifacts at the boundary between the 3D and 2D simulations, as illustrated in Figure 6. In Section 6, we explain how to estimate the effective wave behaviors from any 3D fluid simulation; we then use these imperfect behaviors in our 2D wave solver to remove artificial refractions.

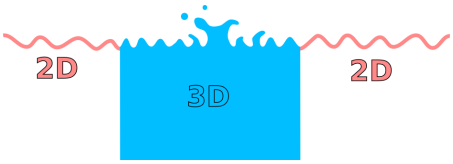


Figure 2: *Side view of our configuration: waves radiating from the 3d simulation are propagated as 2d wave on the surface.*

5. Coupling method

To obtain a seamless transition between the two regions, we must take care to set boundary conditions of each solver so that waves cleanly propagate from one domain to the other. We first describe our novel method for transferring waves from the 3D simulation into the 2D simulation. Afterward, we describe wave-blending techniques for transferring waves back from 2D to 3D.

5.1. Boundary conditions for the 2D simulation

In Section 3, we explained how Schreck *et al.* approximate the waves dispersed by an obstacle. We use a similar approach to approximate the wave field radiating from a 3D domain with a set of 2D sources.

As explained in Section 4, we assume that the 3D surface respects Airy theory near the boundary and can be represented as a heightfield. Let's call u^{3d} the field representing the surface of the

water from the fluid simulation in $\mathcal{D}^{3d}$ and u^{2d} its approximate representation in $\mathcal{D}^{2d}$ using a set of n fundamental sources centered in $\{\boldsymbol{x}_i\}_{i=1\ldots n}$:

$$u^{2d}(\boldsymbol{x},t) = \sum_k \sum_{i=1}^{n} a_{i,k}(t)\Phi(\boldsymbol{x}-\boldsymbol{x}_i)e^{-i\omega_k t} \tag{9}$$

We then use a method similar as the one described in Section 3 to compute the complex amplitudes $a_{i,k}$ such that the transition between $\mathcal{D}^{3d}$ to $\mathcal{D}^{2d}$ is as seamless as possible. For this we simply set the 2D heightfield equal to the 3D surface:

$$u^{2d}(\boldsymbol{x},t) = u^{3d}(\boldsymbol{x},t) \forall \boldsymbol{x} \in \delta\mathcal{D}, \tag{10}$$

where $\delta\mathcal{D}$ is the boundary between the two domains. By resolving the boundary condition separately for each wave number k, we have:

$$p_k^{2d}(\boldsymbol{x},t) = p_k^{3d}(\boldsymbol{x},t) \forall \boldsymbol{x} \in \delta\mathcal{D} \tag{11}$$

with $p_k^{2d}(\boldsymbol{x},t) = \sum_{i=1}^{n} a_{k,i}(t)\phi_k(\boldsymbol{x}-\boldsymbol{x}_i)$. In our examples, we set $\delta\mathcal{D}$ to be an ellipse bounding the 3D simulation, though this choice of domain boundary is up to the user. The sources (red crosses in Figure 3) are placed at an offset distance from $\delta\mathcal{D}$. To satisfy this boundary condition, we discretize $\delta\mathcal{D}$ as a set of m boundary points $\{\boldsymbol{y}_j\}$. As done in [SHW19], we uniformly sample the sources and boundary points with respect to the Nyquist sampling criterion. This gives us a linear system to solve for the amplitudes $a_{k,i}(t)$:

$$\sum_{i=1}^{n} a_{k,i}(t)\phi_k(\boldsymbol{y}_j - \boldsymbol{x}_i) = p_{3d}^k(\boldsymbol{y}_j,t) \forall \boldsymbol{y}_j, j = 1\ldots m \tag{12}$$

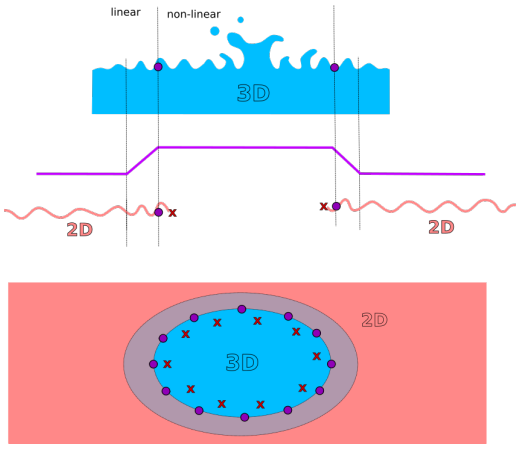


Figure 3: *Top: side view of the 3d fluid simulation and 2d heightfield. Bottom: Top view of the configuration. Boundary points (purple dots) are placed on the border region of the 3d fluid simulation where the waves can be approximated as linear. The sources (red crosses) are placed at an offset distance from them. The surface of the fluid and the 2d heightfield are linearly interpolated (interpolation function in purple) to smooth the transition (between the dotted lines (top) and in the gray region (bottom).*

This system requires a frequency decomposition of the 3D heightfield at each boundary point $\boldsymbol{y}_j$. Setting $H_j(t) = u^{3d}(\boldsymbol{y}_j,t) = \sum_k p_k^{3d}(\boldsymbol{y}_j,t)e^{-i\omega_k t}$, we apply a Fast Fourier Transform (FFT) over a small time window of size $T = w\Delta t$ centered at time τ:

$$FFT([H_j(\tau-\frac{T}{2}),\ldots,H_j(\tau+\frac{T}{2})]) = [p_{f_0}^{3d}(\tau),\ldots,p_{f_w}^{3d}(\tau)], \tag{13}$$

where $[f_0,\ldots,f_w]$ is a range of discrete frequencies separated by $\Delta f = 1/T$. Note that the set of wave numbers used in Equation 9 are the ones corresponding to these frequencies: $[k_0,\ldots,k_w]$ with $f_\xi = 2\pi\omega(k_\xi)$. The precision of the spectral decomposition depends on the window size, so a compromise must be found between precision in frequency and in time. Our implementation uses $w = 256$ for wavelengths ranging between 0.13 and 33 meters. In practice we only keep a few of them that recover the wavelengths created by the simulation. In our examples, we usually use a dozen of wavelengths ranging from 0.6m to 10m.

5.2. Boundary conditions for the 3D simulation

Next we discuss how to inject waves from the 2D region into the 3D fluid simulation. Though similar techniques already exist in the computer animation literature [BW16; SS17], we describe our approach here for completeness. Our results use two different 3D fluid simulation solvers, so this section will discuss how we enforce boundary conditions for both types of simulations.

For the semi-Lagrangian level set simulation, we use the generalized perfectly-matched layers approach of Bojsen-Hansen & Wojtan, treating our 2D wave solver as the background flow. For the narrow band FLIP simulation, we add a boundary layer with gradually increasing viscosity near the boundary, and we blend the positions and velocities of the FLIP particles close to the boundary. Though this gradual blending is not as efficient as the PML-based approach, we found it sufficient to inject the 2D waves and dampen artificial internal reflections.

In any case, the 3D boundary conditions require us to compute the 3D velocity and displacement fields induced by the 2D waves. For a planar wave $\eta(\boldsymbol{x},t) = a\cos(kx-\omega t)$ (with $\boldsymbol{x} = (x,y,z)$ and z being the vertical direction), Airy wave theory gives us an expression for the velocity and particle displacement below the surface [SS17]:

$$v_x(\boldsymbol{x},t) = \dot{x}(t) = a\omega e^{kz}\cos(kx-\omega t) \tag{14}$$

$$v_z(\boldsymbol{x},t) = \dot{z}(t) = a\omega e^{kz}\sin(kx-\omega t) \tag{15}$$

Similarly, the displacement can then be computed as:

$$\xi_x(\boldsymbol{x},t) = -ae^{kz}\sin(kx-\omega t) \tag{16}$$

$$\xi_z(\boldsymbol{x},t) = ae^{kz}\cos(kx-\omega t) \tag{17}$$

We adapt these formulae for a wave field defined by sources. For a surface defined by a source $\eta(\boldsymbol{x},t) = a\,\phi_k(\boldsymbol{x}-x_0)e^{-i\omega t}$ centered in $\boldsymbol{x}_0$, we obtain the following fields:

$$v_r(\boldsymbol{x},t) = \dot{r}(t) = a\omega e^{kz}\phi_k(r)e^{-i\omega t} \tag{18}$$

$$v_z(\boldsymbol{x},t) = \dot{z}(t) = a\omega e^{kz}\phi_k(r)e^{-i(\omega t+\frac{\pi}{2})} \tag{19}$$

$$\xi_r(\boldsymbol{x},t) = -ae^{kz}\phi_k(r)e^{-i(\omega t-\frac{\pi}{2})} \tag{20}$$

$$\xi_z(\boldsymbol{x},t) = ae^{kz}\phi_k(r)e^{-i\omega t} \tag{21}$$

with $r = |\boldsymbol{x} - \boldsymbol{x}_0|$.

The waves coming toward the 3D domain can be for example coming from reflection on an obstacle (represented by another set of sources as described in [SHW19]), or background planar wind waves. When injecting back these waves –represented by additional background heightfield η_{bg}– into the fluid simulation, we need to account for them by enforcing the boundary condition $\eta_{2d}(\boldsymbol{x},t) + \eta_{bg}(\boldsymbol{x},t) = \eta_{3d}(\boldsymbol{x},t)$. Figure 7 shows how this 2D-to-3D coupling looks for circular waves entering a level-set simulation.

6. Matching wave behaviors

Every numerical method for simulating fluids adds some amount of numerical error to the solution. When coupling two simulations which use two different numerical methods, or even coupling two simulations using the same method but with different resolutions, visible artifacts will appear. These errors usually result in different wave speeds or damping rates, known as *numerical dispersion* and *numerical viscosity*. If the two simulations have different apparent values for the dispersion relation $\omega(k)$, waves will visibly refract at the boundary between the simulations.

In our case, the 2D wave solver has a hard-coded dispersion relation, so it actually exhibits no numerical dispersion errors. On the contrary, in the 3D simulation, the grid discretization significantly affects the behavior of the fluid — the effective dispersion law depends on the parameters of the simulation. Instead of trying to make the 3D simulation more accurate, we propose to intentionally add errors to the hard-coded 2D dispersion law, so that it better matches the 3D simulation. We do this by estimating the effective dispersion relation of the 3D fluid, and then using this estimated law in place of the analytic ω_k in ESM. Rather than analytically trying to derive the dominant error for each specific method, we chose an easy to run method which is based on numerical experiments and agnostic to the kind of 3D solver.

We use the following experiment to estimate a 3D fluid simulation's frequency response $f = 2\pi\omega$ to a given wavelength $\lambda = \frac{2\pi}{k}$: we initialize a standing wave of wavelength λ centered within a a virtual wave tank of size $l = 2\lambda$. We then begin the simulation and measure the height $H(t)$ of the point in the center of the tank over time. Next, we apply an FFT on $H(t)$ and identify the frequency of the spectrum with the highest amplitude to compute f and $\omega = f/2\pi$. Figure 4 illustrates this experimental setup.

By repeating the experiment for different wavelengths we can plot the dispersion law $\omega = \omega_s(k)$ and compare it to the theoretical one $\omega = \sqrt{gk\tanh(kh)}$. Figure 5 plots our findings for both the FLIP simulation and levelset simulators for a fixed simulation resolution and time step $\Delta t = 0.033$. For the levelset simulation we use a cell size of 0.0625m, and for FLIP we use a particle separation of 0.075 with the Houdini parameters particle radius scale equal to 1.2 and a grid scale equal to 2. Figure 5 shows the dispersion laws we obtained for these two simulations compared to the theoretical gravity waves dispersion.

The FLIP simulation substantially increases the wave speed and adds noise. This discrepancy is illustrated in Figure 6 (a) where the difference between the dispersion laws between the 3D FLIP simulation (left) and 2D wave simulation (right) leads to sharp wave refraction at the interface. To compensate for this increased frequency, we found that simply re-scaling the theoretical law by a constant coefficient $d \approx 1.4$ is sufficient to practically eliminate artificial wave refractions, as illustrated in Figure 6 (b).

Although the level-set simulation is close enough to theory for the refraction to be far less noticeable, it still exhibits slightly slower waves than Airy wave theory predicts. This difference in wave speed can create inconsistencies along the border when injecting 2D waves into the 3D simulation, resulting in spurious waves as shown in Figure 7 (left).To correct this, we directly use data from the experiments to associate a more accurate frequency ($\omega = 3.13$) to the injected wavelength ($\lambda = 6m$).

Although our examples apply simple approaches to match the sources dispersion law with the data from the 3D simulation, any data-fitting approach could work; our only requirement is that $\omega_s(k)$ be differentiable, so we can compute the group speed $\frac{\partial\omega}{\partial k}$ for Schreck *et al.* [SHW19]'s wavelet propagation.

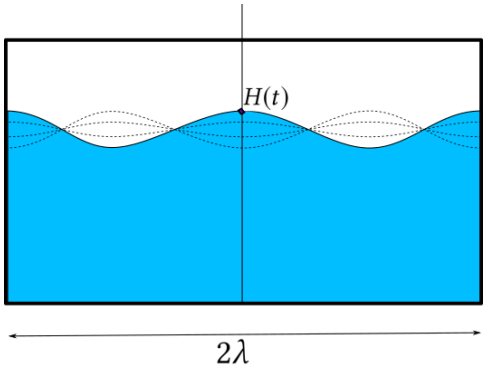


Figure 4: *Set up of our simulation of a standing wave. The tank measures exactly two times the wavelength λ. We record the height in the middle of the tank.*

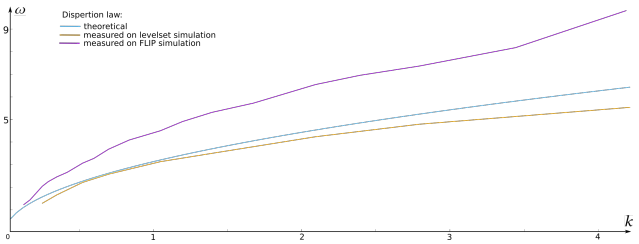
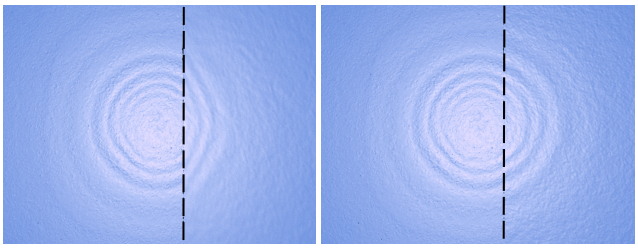


Figure 5: *The recorded dispersion laws for a FLIP simulation and a levelset simulation compared to the theoretical relation of gravity waves.*

7. Post-processing details

Surfaces reconstructed from particles tend to exhibit high-frequency noise, which makes the 3D FLIP simulations look substantially different from the perfectly smooth analytical solutions given by the 2D wave solver. To make the surfaces look similar, we add Perlin noise [Per85] to the 2D wave surfaces:

$$\eta_{\text{noise}}(\boldsymbol{x},t) = \text{Re}(a(\boldsymbol{x})e^{-i\omega t})$$



(a) (b)

Figure 6: *A FLIP simulation on the left side of the dotted line coupled with fundamental sources using the theoretical gravity wave dispersion law (a) and a corrected law to match better the recorded law shown in Figure 5 (b). The left picture shows a spurious refraction effect due to the difference of speed of the waves between the 3d fluid simulation and the 2d wave propagation. This effect is corrected on the right by using the new law.*

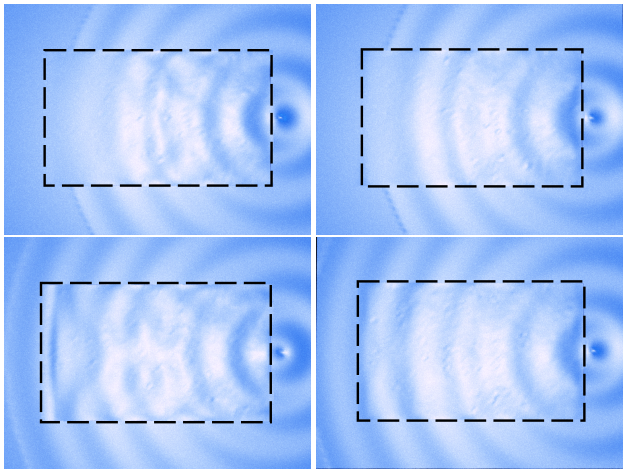


Figure 7: *Injecting the waves from a source inside a levelset simulation using Bojsen-Hansen & Wojtan's [BW16] method. Left: The difference of velocity between the theoretical dispersion law used by the source and that of the simulation causes spurious waves. Right: Correcting the law of the source removes these artifacts.*

where $a(\boldsymbol{x})$ is a complex amplitude whose real and imaginary parts are two noise functions with the same parameters. We also linearly blend the surface functions between the 2D and 3D domains, as illustrated by the purple interpolation curve in Figure 3.

Figure 8 shows an example with and without these post-processing steps.

8. Results

Please see our supplementary video for animations of our results. We first verify our coupling method in the absence of complicating factors by replacing the 3D fluid simulation with a 2D heightfield with a known analytical solution. Figure 9 shows how we generated a few circular ripples within the dashed ellipse, and solved for the fundamental solutions which best matched the behavior at the simulation boundary.

Figure 10 illustrates the consequences of different dispersion laws, comparing the deep water dispersion law $\omega = \sqrt{gk\tanh(kh)}$ to the shallow water one $\omega = k\sqrt{gh}$. The two laws exhibit very different behaviors; the velocity of the waves is similar for all frequencies for shallow water, but they are dispersive in deeper water. Deep water waves also have a phase velocity that differs from their group velocity, causing waves to fade away at the edge of the group (see the video). This is not the case for shallow water for which the group velocity and phase velocity are equal.

Our method is practically agnostic to the type of 3D fluid simulation used. To illustrate this benefit, we show examples using two different kinds of simulation. Figure 11 shows coupling with a semi-Lagrangian levelset method to simulate a few drops falling into water with ripples propagating far away from the original 3D domain. Figure 1 shows a FLIP simulation with a dragon springing out of water, with 2D waves propagating outward into an unbounded fluid domain.

Our method's choice of 2D wave propagation gives an analytical solution defined everywhere in the 2D domain, so the surface can be adaptively computed at render-time to represent very large scenes as well as more detailed close-ups. Figure 12 shows a rotating water wheel with the camera zooming out to view waves propagating far away. Figure 13 shows a zoom on the boundary between the 2D and 3D domain of the example shown in Figure 1, using a finer grid to obtain a smooth and detailed close-up view.

Finally, we compare our coupling method with a full 3d fluid simulation in Figure 14. The visual results are very similar, but our method takes only a fraction of the time to compute: our coupling method takes around 30 seconds per frame (20s for generating the 3D simulation at the center and 11s to compute the 2D wave propagation) and the full simulation takes more than 2 minutes.

Table 1 shows the times for our examples as well as the number of sources, boundary points, and resolution of the display grid. Note that, as described in [SHW19], we are using a different number of sources for each wavelength. For example in Figure 14 (left), the biggest wavelength is 45 meters and the corresponding number of sources is 16. The smallest wavelength is 1.5m with 566 sources.

Discussion about previous work: The recent article from Huang *et al.* [HQT*21] achieves impressive results by coupling a FLIP simulation with a Boundary Element Method (BEM). Our approach uses fundamental solutions instead of BEM to couple the 2D and 3D domain, leading to significantly different practical consequences. Although both methods assume linear waves outside of the 3D domain, our method is strictly limited to height field waves, while the BEM used by Huang *et al.* is not. On the other hand, our method is distinct from meshing-based approaches in that it can handle infinite domains and can simulate arbitrarily high-frequency waves without any spatial adaptivity in the simulation domain. Our work also offers preliminary insights into how to address the coupling errors at simulation boundaries, which may be applicable to methods like Huang *et al.*

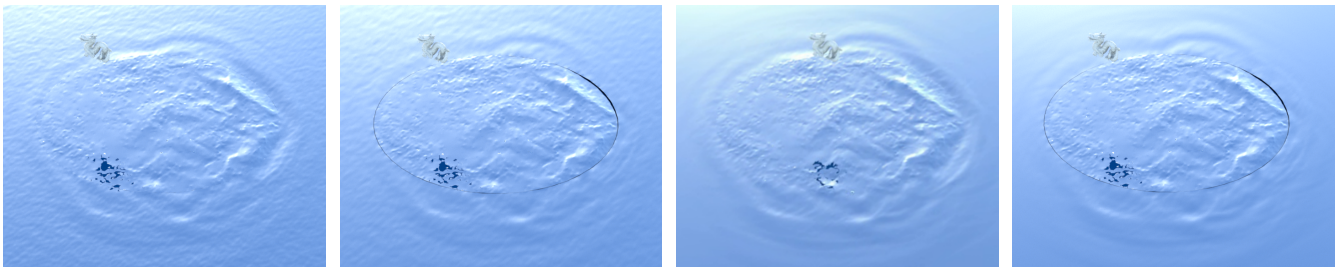


Figure 8: *Our method with both linear blending and noise (left), without the linear blending (center-left), without noise (center-right) and without any post-processing (right).*

	# wavelengths	# sources	# bound points	Surface grid resolution	time / frame (2D)	time / frame (3D)
Figure 1	17	7274	2000	450x600	141s	222s
Figure 6	7	142	600	300x150	0.6s	3.7s
Figure 9	7	140	300	200x200	0.4s	-
Figure 11	6	368	900	300x400	2.4s	1.2s
Figure 12	18	3839	1200	150x200	6.8s	26s
Figure 14 (left)	10	2352	2000	300x300	11s	20s
Figure 14 (right)	-	-	-	-	-	138s

Table 1: *Number of wavelengths, number of sources and boundary points, resolution of the 2D grid for our examples, as well as the time per frame for computing the 2D waves and the time per frame for the original 3D fluid simulation.*

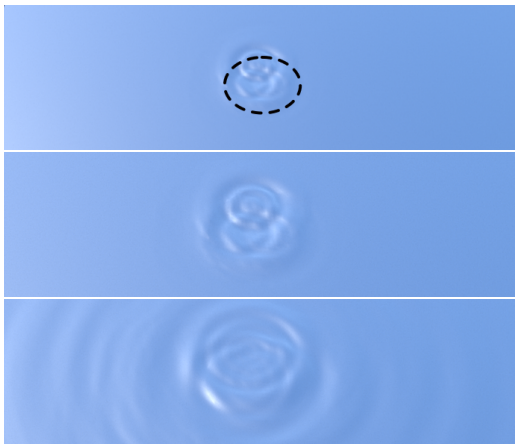


Figure 9: *A controlled experiment to verify our approach: Our coupling method produces 2D waves that radiate away from another 2D domain within the dashed ellipse.*

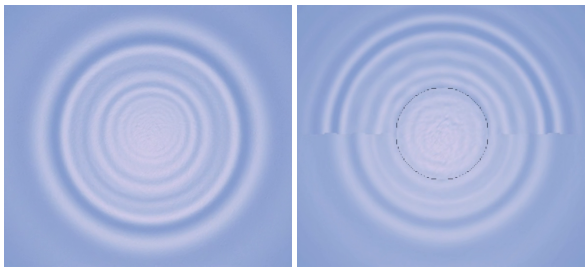


Figure 10: *Left: a full levelset simulation of one drop in the water in a deep pool. Right: a levelset simulation (in the center ring) coupled with 2D wave propagation using shallow water dispersion (top) and deep water dispersion (bottom). The dispersion law used by the 2D simulator has a significant affect on the resulting wave pattern and coupling, and should be chosen to match the 3D simulation.*

9. Discussion and Conclusion

We first note that our method only approximates the outgoing waves from a 3D simulation, so, despite our efforts to reduce visual artifacts, there will still be errors when the two simulations do not exactly match up on the boundary. One potential error source is the finite time source on our FFT used to estimate the 2D wave amplitudes; the choice of time window trades off between accuracy in spatial and temporal frequencies (also known as the Fourier uncertainty principle). The 2D wave solver also requires a choice of how densely to discretize the boundary and the space of wavenumbers; insufficient discretization here can also cause mismatches in the outgoing waves.

We chose to match the wave speeds across simulation boundaries by approximating the effective dispersion relation of the 3D solver. However, damping can also differ across simulations. Our initial investigations show that this is not nearly as strong of an effect as the mismatched wave speeds, but it is possible to develop a method for matching the effective damping rate in the future.

Finally, we note that our implementation does not take full advantage of the equivalent sources method. We used a parallel im-

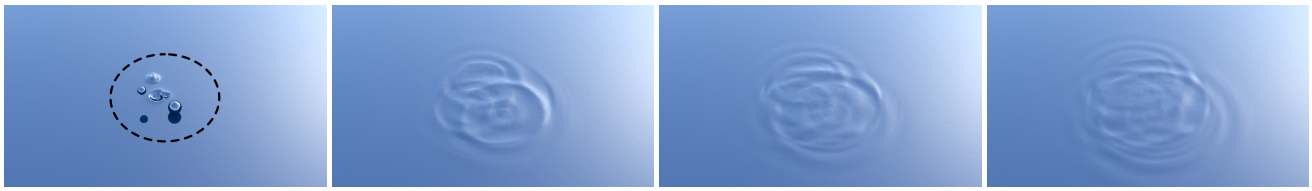


Figure 11: *Drops falling into water. Our coupling method used with a levelset simulation (inside the dashed ellipse).*

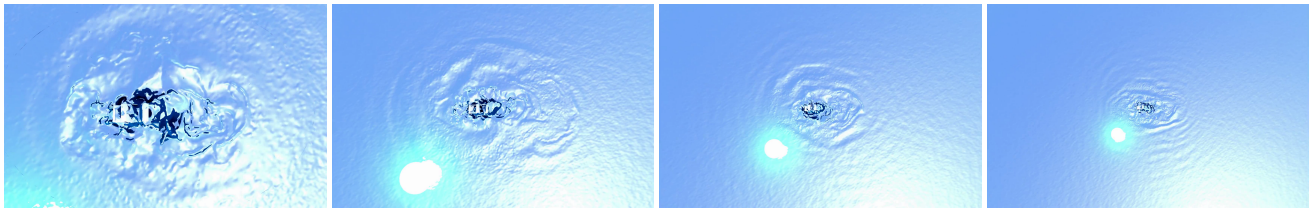


Figure 12: *Zooming out from a water wheel while still displaying waves radiating from the FLIP simulation in the center.*

plementation on CPU but due to constraints in implementing the method within Houdini, we did not parallelize the 2D wave solver on the GPU. We believe the performance of our method would be much faster with a more optimal implementation.

To conclude, we presented a method that adds radiating 2D waves to a localized 3D fluid simulation by coupling it with the equivalent sources method. The method allows efficient simulation of large open domains with arbitrarily high spatial frequencies, and it produces more physically plausible wave behaviors than other 2D solvers which assume constant wave speeds. Finally, we introduced a novel method for discovering and matching the effective dispersion behaviors of an arbitrary 3D fluid simulation, leading to a practical elimination of refraction artifacts at the boundary between simulations.

Acknowledgments: We wish to thank the anonymous reviewers and the members of the Visual Computing Group at IST Austria and MFX Team at INRIA for their valuable feedback. This research was supported by the Scientific Service Units (SSU) of IST Austria through resources provided by Scientific Computing.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No. 638176.

References

[Air41] AIRY, GEORGE BIDDELL. "Tides and waves". (1841) 2, 3.

[ATW13] ANDO, RYOICHI, THÜREY, NILS, and WOJTAN, CHRIS. "Highly adaptive liquid simulations on tetrahedral meshes". *ACM Transactions on Graphics (TOG)* 32.4 (2013), 1–10 2.

[BLW12] BOJSEN-HANSEN, MORTEN, LI, HAO, and WOJTAN, CHRIS. "Tracking surfaces with evolving topology." *ACM Trans. Graph.* 31.4 (2012), 53–1 3.

[BR86] BRACKBILL, JEREMIAH U and RUPPEL, HANS M. "FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions". *Journal of Computational physics* 65.2 (1986), 314–343 2.

[Bri15] BRIDSON, ROBERT. *Fluid simulation for computer graphics*. CRC press, 2015 2.

[BW13] BOJSEN-HANSEN, MORTEN and WOJTAN, CHRIS. "Liquid surface tracking with error compensation". *ACM Transactions on Graphics (TOG)* 32.4 (2013), 68 3.

[BW16] BOJSEN-HANSEN, MORTEN and WOJTAN, CHRIS. "Generalized Non-Reflecting Boundaries for Fluid Re-Simulation". *ACM Transactions on Graphics (SIGGRAPH 2016)* 35.4 (2016) 3, 5, 7.

[BWHT07] BARGTEIL, ADAM W, WOJTAN, CHRIS, HODGINS, JESSICA K, and TURK, GREG. "A finite element method for animating large viscoplastic flow". *ACM transactions on graphics (TOG)* 26.3 (2007), 16–es 2.

[CM10] CHENTANEZ, NUTTAPONG and MÜLLER, MATTHIAS. "Real-time Simulation of Large Bodies of Water with Small Scale Details." *Symposium on Computer Animation*. 2010, 197–206 2, 3.

[CM11] CHENTANEZ, NUTTAPONG and MÜLLER, MATTHIAS. "Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid". *ACM SIGGRAPH 2011 Papers*. New York, NY, USA: Association for Computing Machinery, 2011. ISBN: 9781450309431 2.

[CMK15] CHENTANEZ, N., MÜLLER, M., and KIM, T. "Coupling 3D Eulerian, Heightfield and Particle Methods for Interactive Simulation of Large Scale Liquid Phenomena". *IEEE Transactions on Visualization and Computer Graphics* 21.10 (2015) 3.

[CMT*16] CANABAL, JOSÉ A., MIRAUT, DAVID, THÜEREY, NILS, et al. "Dispersion Kernels for Water Wave Simulation". *ACM Trans. Graph.* 35.6 (Nov. 2016), 202:1–202:10. ISSN: 0730-0301 2.

[EQYF13] ENGLISH, R ELLIOT, QIU, LINHAI, YU, YUE, and FEDKIW, RONALD. "Chimera grids for water simulation". *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2013, 85–94 3.

[FAW*16] FERSTL, FLORIAN, ANDO, RYOICHI, WOJTAN, CHRIS, et al. "Narrow Band FLIP for Liquid Simulations". *Computer Graphics Forum* 35.2 (2016), 225–232 3.

[HQT*21] HUANG, LIBO, QU, ZIYIN, TAN, XUN, et al. *Ships, Splashes, and Waves on a Vast Ocean*. 2021. arXiv: 2108.05481 [cs.GR] 3, 7.

[IGLF06] IRVING, GEOFFREY, GUENDELMAN, ERAN, LOSASSO, FRANK, and FEDKIW, RONALD. "Efficient Simulation of Large Bodies of Water by Coupling Two and Three Dimensional Techniques". *ACM SIGGRAPH 2006 Papers*. SIGGRAPH '06. Boston, Massachusetts: Association for Computing Machinery, 2006. ISBN: 1595933646 2.

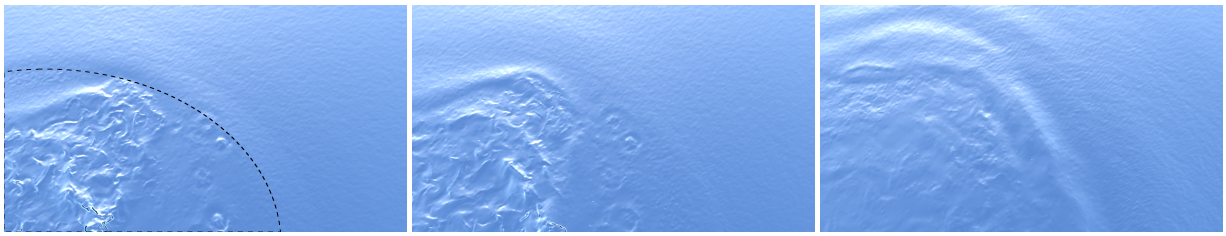


Figure 13: *A close-up view of the example in Figure 1, using the same wave sources but a finer resolution grid for the displayed surface.*

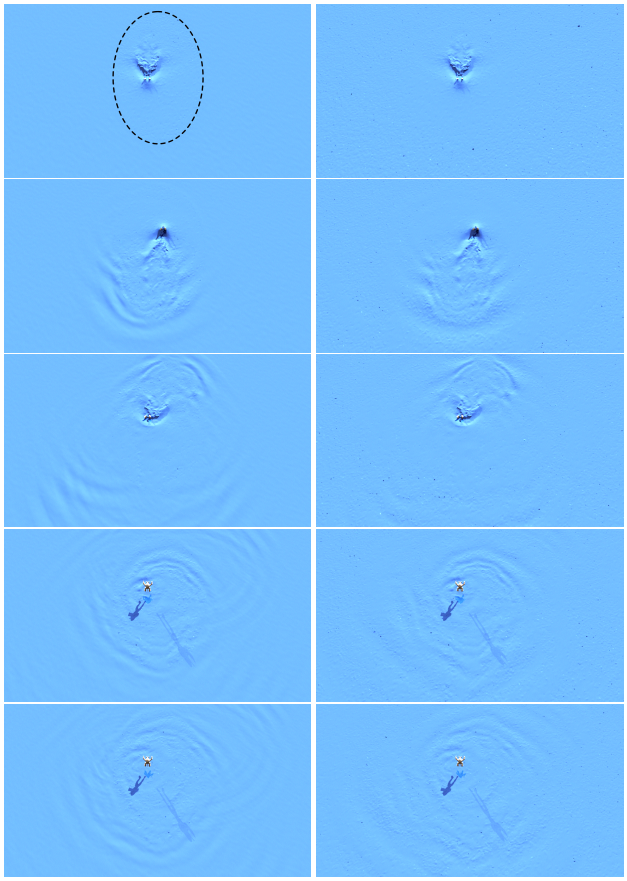


Figure 14: *An armadillo being cruelly dipped into water. (left) Our coupling method with a FLIP simulation at the center (dashed ellipse) and 2d waves everywhere else. (right) Large FLIP simulation.*

[IWT*18] IBAYASHI, HIKARU, WOJTAN, CHRIS, THÜEREY, NILS, et al. "Simulating liquids on dynamically warping grids". *IEEE transactions on visualization and computer graphics* 26.6 (2018), 2288–2302 2.

[Joh97] JOHNSON, R. S. *A Modern Introduction to the Mathematical Theory of Water Waves*. Cambridge University Press, 1997. ISBN: 052159832X 3.

[JSM*18] JESCHKE, STEFAN, SKŘIVAN, TOMÁŠ, MÜLLER-FISCHER, MATTHIAS, et al. "Water surface wavelets". *ACM Transactions on Graphics (TOG)* 37.4 (2018), 94 2.

[JW15] JESCHKE, STEFAN and WOJTAN, CHRIS. "Water Wave Animation via Wavefront Parameter Interpolation". *ACM Transactions on Graphics* 34.3 (2015), 1–14 2.

[JW17] JESCHKE, STEFAN and WOJTAN, CHRIS. "Water Wave Packets". *ACM Transactions on Graphics (SIGGRAPH 2017)* 36.4 (2017) 2.

[KFCO06] KLINGNER, BRYAN M, FELDMAN, BRYAN E, CHENTANEZ, NUTTAPONG, and O'BRIEN, JAMES F. "Fluid animation with dynamic meshes". *ACM SIGGRAPH 2006 Papers*. 2006, 820–825 2.

[KSK09] KIM, DOYUB, SONG, OH-YOUNG, and KO, HYEONG-SEOK. "Stretching and wiggling liquids". *ACM SIGGRAPH Asia 2009 papers*. 2009, 1–7 3.

[KTT13] KIM, THEODORE, TESSENDORF, JERRY, and THÜREY, NILS. "Closest Point Turbulence for Liquid Surfaces". *ACM Trans. Graph.* 32.2 (Apr. 2013). ISSN: 0730-0301 3.

[LGF04] LOSASSO, FRANK, GIBOU, FRÉDÉRIC, and FEDKIW, RON. "Simulating water and smoke with an octree data structure". *ACM SIGGRAPH 2004 Papers*. 2004, 457–462 2.

[LvdP02] LAYTON, ANITA T and van de PANNE, MICHIEL. "A numerically efficient and stable algorithm for animating water waves". *The Visual Computer* 18.1 (2002) 2.

[MBT*15] MERCIER, OLIVIER, BEAUCHEMIN, CYNTHIA, THÜEREY, NILS, et al. "Surface Turbulence for Particle-Based Liquid Simulations". *ACM Trans. Graph.* 34.6 (Oct. 2015). ISSN: 0730-0301 3.

[MWN*17] MANTEAUX, P-L, WOJTAN, CHRISTOPHER, NARAIN, RAHUL, et al. "Adaptive physically based models in computer graphics". *Computer Graphics Forum*. Vol. 36. 6. Wiley Online Library. 2017, 312–337 2.

[NB11] NIELSEN, MICHAEL B and BRIDSON, ROBERT. "Guide shapes for high resolution naturalistic liquid simulation". *ACM SIGGRAPH 2011 papers*. 2011, 1–8 3.

[OF06] OSHER, STANLEY and FEDKIW, RONALD. *Level set methods and dynamic implicit surfaces*. Vol. 153. Springer Science & Business Media, 2006 2, 3.

[Per85] PERLIN, KEN. "An image synthesizer". *ACM Siggraph Computer Graphics* 19.3 (1985) 6.

[PTM09] PATEL, SANJIT, TESSENDORF, JERRY, and MOLEMAKER, JEROEN. "Monocoupled 3D and 2D river simulations". *Proc. ACM/Eurographics Symp. Comp. Anim., Posters Session*. 2009 3.

[SBC*11] SOLENTHALER, BARBARA, BUCHER, PETER, CHENTANEZ, NUTTAPONG, et al. "SPH based shallow water simulation". (2011) 2.

[SHW19] SCHRECK, CAMILLE, HAFNER, CHRISTIAN, and WOJTAN, CHRIS. "Fundamental solutions for water wave animation". *ACM Transactions on Graphics (TOG)* 38.4 (2019) 2–7.

[SS17] STOMAKHIN, ALEXEY and SELLE, ANDREW. "Fluxed Animated Boundary Method". *ACM Trans. Graph.* 36.4 (July 2017). ISSN: 0730-0301 3, 5.

[SSJ*20] SKRIVAN, TOMAS, SODERSTROM, ANDREAS, JOHANSSON, JOHN, et al. "Wave curves: Simulating lagrangian water waves on dynamically deforming surfaces". *ACM Transactions on Graphics (TOG)* 39.4 (2020) 3.

[Sta99] STAM, JOS. "Stable fluids". *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 1999, 121–128 2.

[Tes04] TESSENDORF, JERRY. "Interactive water surfaces". *Game Programming Gems* 4 (2004) 2.

[Tes14] TESSENDORF, JERRY. "eWave: Using an Exponential Solver on the iWave Problem". *Technical Note* (2014) 2.

[TMSG07] THÜREY, N., MÜLLER-FISCHER, M., SCHIRM, S., and GROSS, M. "Real-time Breaking Waves for Shallow Water Simulations". *15th Pacific Conference on Computer Graphics and Applications (PG'07)*. 2007 3.

[TRS06] THÜREY, NILS, RÜDE, ULRICH, and STAMMINGER, MARC. "Animation of open water phenomena with coupled shallow water and free surface simulations". *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 2006 3.

[TWGT10] THÜREY, NILS, WOJTAN, CHRIS, GROSS, MARKUS, and TURK, GREG. "A Multiscale Approach to Mesh-Based Surface Tension Flows". *ACM Trans. Graph.* 29.4 (July 2010). ISSN: 0730-0301 3.

[Vic81] VICHNEVETSKY, ROBERT. "Propagation through numerical mesh refinement for hyperbolic equations". (1981) 4.

[WMB11] WOJTAN, CHRIS, MÜLLER-FISCHER, MATTHIAS, and BROCHU, TYSON. "Liquid simulation with mesh-based surface tracking". *ACM SIGGRAPH 2011 Courses*. 2011, 1–84 2.

[WMT07] WANG, HUAMIN, MILLER, GAVIN, and TURK, GREG. "Solving general shallow wave equations on surfaces". *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. 2007, 229–238 2.

[WT08] WOJTAN, CHRIS and TURK, GREG. "Fast viscoelastic behavior with thin features". *ACM SIGGRAPH 2008 papers*. 2008, 1–8 2.

[YHW*16] YANG, SHENG, HE, XIAOWEI, WANG, HUAMIN, et al. "Enriching SPH Simulation by Approximate Capillary Waves". *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '16. Zurich, Switzerland: Eurographics Association, 2016. ISBN: 9783905674613 3.

[YT13] YU, JIHUN and TURK, GREG. "Reconstructing surfaces of particle-based fluids using anisotropic kernels". *ACM Transactions on Graphics (TOG)* 32.1 (2013), 1–12 2, 3.

[YWTY12] YU, JIHUN, WOJTAN, CHRIS, TURK, GREG, and YAP, CHEE. "Explicit mesh surfaces for particle based fluids". *Computer Graphics Forum*. Vol. 31. 2pt4. Wiley Online Library. 2012 3.

[ZLC*13] ZHU, BO, LU, WENLONG, CONG, MATTHEW, et al. "A new grid structure for domain extension". *ACM Transactions on Graphics (TOG)* 32.4 (2013), 1–12 2.