

Coupling 3D Eulerian, Heightfield and Particle Methods for Interactive Simulation of Large Scale Liquid Phenomena

Nuttapong Chentanez , Matthias Müller , Tae-Yong Kim

NVIDIA

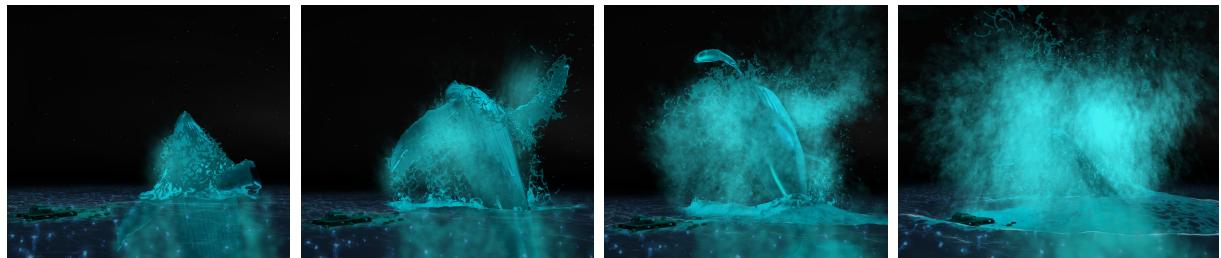


Figure 1: Simulation of a humpback whale breaching using our hybrid method. Near the whale, the water is represented by a combination of particles and a 3D grid. This local domain is surrounded by a height field water simulation based on the Shallow Water Equations (SWE). In addition, a procedural approach is used to add small scale ripples and the ocean in the far distance. Secondary foam and spray particles give the scene the final look. This simulation runs in real time at 30fps on NVIDIA GTX 780Ti. The resolutions of the 3D and SWE grids are 64^3 and 512^2 respectively and there are 112k PBF particles on average.

Abstract

We propose a new method to simulate large scale water phenomena by combining particle, 3D grid and height field methods. In contrast to most hybrid approaches that use particles to simulate foam and spray only, we also represent the bulk of water near the surface with both particles and a grid depending on the regions of interest and switch between those two representations during the course of the simulation. For the coupling we leverage the recent idea of tracking the water surface with a density field in grid based methods. Combining particles and a grid simulation then amounts to adding the density field of the particles and the one stored on the grid. For open scenes, we simulate the water outside of the 3D grid domain by solving the Shallow Water Equations on a height field. We propose new methods to couple these two domains such that waves travel naturally across the border. We demonstrate the effectiveness of our approach in various scenarios including a whale breaching simulation, all running in real-time or at interactive rates.

Categories and Subject Descriptors (according to ACM CCS):
I.3.5 [Computer Graphics]: —Computational Geometry and Object Modeling Physically Based Modeling I.3.7 [Computer Graphics]: —Three-Dimensional Graphics and Realism Animation and Virtual Reality

1. Introduction

Liquid simulation has been successfully used in many feature films. Three widely used approaches are 3D Eulerian simulation, 2D height field simulation and particles based simulation. Each of these approaches is suitable for a certain range of liquid phenomena. To simulate a large scale scene with small scale details, one would normally use a high resolution 3D grid or a large number of particles. Both solutions

are not suitable for real time applications due to their high computational cost.

In this paper we propose a new method that allows switching back and forth between a grid based and a particle based representation depending on where detail is required. Our main idea is to leverage the method of Mullen et al. [MMTD07] to track the free surface with a density field instead of a level set in grid based approaches. The fact that particle methods such as SPH or Position Based Fluids (PBF) [MM13] also use a density field to represent the liquid lets us combine the Eulerian and Lagrangian simulations naturally by adding the two fields. To handle open water scenes we additionally represent the surrounding water far away from the camera with a height field and simulate it by solving the Shallow Water Equations (SWE).

Our combined method leverages the advantages and strengths of each individual representation: the ease of handling water that expands to arbitrary regions and the ability to capture small details of particle based approaches, the regularity of computation and the straightforward way of enforcing incompressibility of grid based methods and the low computational cost of shallow water based solvers. At the same time, our hybrid method avoids the typical drawbacks of those representations, namely the need of a large number of particles to represent the bulk liquid volume, the need for a high resolution grid to capture small details and the inability to represent configurations like overturning waves of height field based solvers.

Our main contributions include

- A fast and GPU-friendly method for coupling a 3D grid and a particle solver, specifically coupling an Eulerian water simulation with Position Based Fluids (PBF) or SPH particles.
- A mechanism to couple the grid representation and the particles with different time step sizes.
- A fast and GPU-friendly method for coupling a 3D grid and an SWE solver with different resolutions and time step sizes.

2. Related Work

There is a large body of work on both, pure Eulerian and pure Lagrangian methods. The most popular Lagrangian method in graphics is SPH [GM77], [DC96], [MCG03] with a variety of extensions such as PCISPH [SP09] and WCSPH [BT07]. Recently a position based fluid [MM13] approach was introduced.

Combining grid based and particle based approaches is possible. Such hybrid approaches leverage the advantages of both methods. This idea has been used intensively in graphics. Here we highlight the papers that are most closely related to our method. Foster and Metaxas [FM96] solved the Navier-Stokes equations on a grid and used marker particles to track the liquid cells. Foster and Fedkiw [FF01] computed

the signed distance field of a liquid on a grid using spheres centered at particles with smoothing to reduce the bumpiness of the surface. Enright et al. [EMF02] proposed the particle-level set (PLS) method which uses particles on both the liquid and the air side of the interface to correct the level set value defined on grid. In these methods, particles are passively advected using the grid velocity field for the purpose of tracking the location of the liquid. This family of methods has been extended to handle multiple fluids [LSSF06] or splashing liquids [KCC*06].

With the Particles in Cells (PIC) [Har63] and the Fluid-Implicit Particles (FLIP) [BR86] methods, the particles are used to carry the velocity field. These methods were introduced to computer graphics by Zhu and Bridson [ZB05]. At each time step, the velocities stored on the particles are rasterized to a grid for pressure projection. The resulting velocities on the grid are then transferred back to particles by directly interpolating the velocity values (PIC) or by interpolating the velocity changes (FLIP) or a linear combination of the two. This idea has been extended to support two phase flow [BB12] and to support adaptively sized particles [HYK08], [ATT12]. The Material Point Method (MPM) [SCS94] is a related approach in which the particles carry additional physical properties such as mass and temperature. MPM was recently used in computer graphics to produce convincing simulations of snow [SSC*13].

Hong et al. [HYK08] used a grid for the simulation of a liquid and particles to represent bubbles. Raveendran et al. [RWT11] modified the SPH approach and use a grid to globally solve for a divergence free velocity field on a coarse scale. This way, the SPH forces are only needed to counteract local density variations. Zheng et al. [ZZKF13] combine a regular grid inside the liquid volume with a tetrahedral mesh constructed from particles near the surface to solve for incompressibility.

The two methods most closely related to our work are the approaches by Losasso et al. [LTKF08] and the work of Wang et al. [WZKQ13] who combine SPH with the Particle Level Set (PLS) and the Lattice-Boltzmann method [TR04], respectively. In both cases, the particles are rasterized to a grid to solve for incompressibility. Then, similar to PIC and FLIP, either the velocity itself or the change in velocity is transferred back to particles. However, they assume that particles are simulated with the same time step size as the 3D grid. Moreover, the particles creation and deletion are not necessarily mass conserving.

For a large scale liquid simulation, using particles and a regular 3D grid alone is not feasible. A variety of adaptive grids has been proposed to reduce the computational cost such as Octrees [LGF04], tall cell grids [IGLF06], [CM11], chimera grids [EQYF13], rectilinear grids [ZLC*13], and tetrahedral meshes [CFL*07], [BXH10] and [ATW13]. Also, adaptive sampled SPH [APKG07] and two-scale SPH [SG11] have been proposed to reduce the cost of Lagrangian approaches.

An effective way to reduce the simulation cost for large scale liquid simulations further is the reduction of the problem dimension from 3D to 2D using various simplifications [Bri05]. Kass et al. [KM90] introduced the idea of using the wave equation for the simulation of a water surface to graphics. This approach has been extended in various ways such as in [MY97], [Tes99], and [YHK07]. The Shallow Water Equations (SWE) are a more accurate way of reducing the Navier Stokes equations to 2D [LvdP02], [HHL*05], and [WMT07]. In addition to the vertical velocities, they also capture the evolution of a horizontal velocity field which can be used to advect foam and other objects on the fluid surface for instance.

Neither the wave equation nor the SWE are able to capture 3D effects such as breaking waves or splashing. This limitation has been reduced in a number of ways. Mould and Yang [MY97] added bubbles and droplets to the height field representation, Holmberg and Wünsche [HW04] used a weir model to simulate waterfalls and Chentanez and Müller [CM11] coupled SWE with particles to simulate breaking waves, spray and splashes. Thürey et al. [TRS06] coupled a Lattice-Boltzmann grid with SWE to simulate open water phenomena and used particles for spray. Our work differs from all these approaches in that the 3D grid, the particles and the SWE grid are coupled to simulate the un-aerated liquid phase. The aerated part such as spray, foam and bubbles can still optionally be added to further enhance visual realism.

3. Method

The core idea behind our method is to represent the un-aerated liquid phase with both particles and a density field sampled on an Eulerian grid. A given amount of liquid can thereby switch back and forth between these two representations in the course of the simulation. Our 3D grid simulation follows [CM12] whereas the particles can alternatively be simulated using variants of SPH such as [SP09] or PBF [MM13]. For the animation of the liquid surface outside the 3D grid domain we use the shallow water solver (SWE) described in [CMF10].

Throughout the paper, ρ^g denotes the surface density field sampled at the center of the 3D grid cells. As in Mullen et al. [MMTD07], ρ^g is not the physical density of liquid but a normalized density varying between 0 to 1 where the interface is assumed to lie at 0.5 iso-contour. u^g, v^g, w^g are the x-, y- and z-components of the grid velocity field. They are stored on cell faces. The positions, velocities and radii of the particles are denoted by \mathbf{x} , \mathbf{v} and r , respectively. For the SWE simulation we use the symbols h^s, H^s, u^s and w^s for the water depth, the terrain height and the x- and z-component of the velocity field, respectively. For simplicity in exposition, the spacing Δx is assumed to be the same for both the 3D and the SWE grid, an assumption we will remove in Section 4.

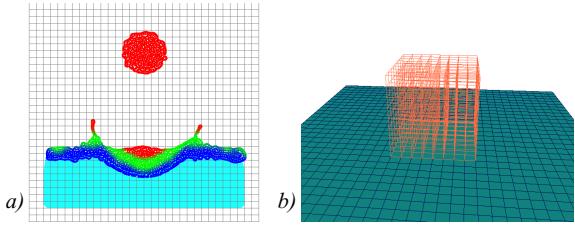


Figure 2: a) Particles and grid representations. Particles are used close to the liquid surface while grid is used further below. The particles are color-coded based on their distance to the water surface defined by the grid. b) The combination of a 3D grid with a surrounding height field representation.

We use t to denote the current time and Δt for the time step size of the 3D grid. The particles are simulated using s_p sub-steps with a time step of $\frac{\Delta t}{s_p}$. Taking smaller time steps for the particles is necessary for both SPH and PBF, in the case of SPH for stability reasons and in the case of PBF to decrease the compressibility of the liquid. SWE is simulated using s_s substeps with a time step of $\frac{\Delta t}{s_s}$.

We represent the part of the liquid near the region of interest by particles, typically near the liquid-air interface as shown in Figure 2a. Wherever we use an SWE simulation outside the 3D grid domain, we assume that the 3D grid lies completely inside the SWE grid (see Figure 2b).

The term $q|_{\mathbf{p}}$ stands for a quantity q defined on a 3D grid and evaluated at position \mathbf{p} via tri-linear interpolation. If q is a quantity defined on the 2D SWE grid then $q|_{\mathbf{p}}$ denotes the bi-linear interpolation of q at position (p_x, p_z) .

Algorithm 1 describes the main simulation loop. The

- 1: Extrapolate 3D grid velocity
- 2: Track surface
- 3: Advect velocity
- 4: Add external forces
- 5: Compute particles influence on 3D grid, §3.1
- 6: Pressure Projection, §3.2
- 7: Compute 3D grid influence on particles and simulate particles, §3.3
- 8: Add and remove particles, §3.4
- 9: Compute 3D grid influence on SWE and simulate SWE, §3.5.
- 10: Compute SWE influence on 3D grid, §3.6

Algorithm 1: Hybrid Simulation Steps

first four steps are common to most 3D fluid simulators and closely follow [CM12]. After adding external forces we couple the velocities of the particles and the grid velocities as described in Section 3.1. Then we solve for pressures that enforce incompressibility on the grid using the multi-grid approach of [CM12] with a few modifications. The cor-

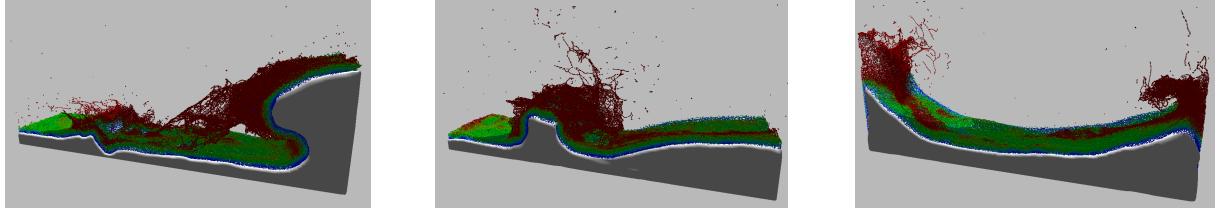


Figure 4: Visualization of the grid and particles representations of the liquid. While most of the liquid's volume is represented by the grid, surface details are represented by particles.

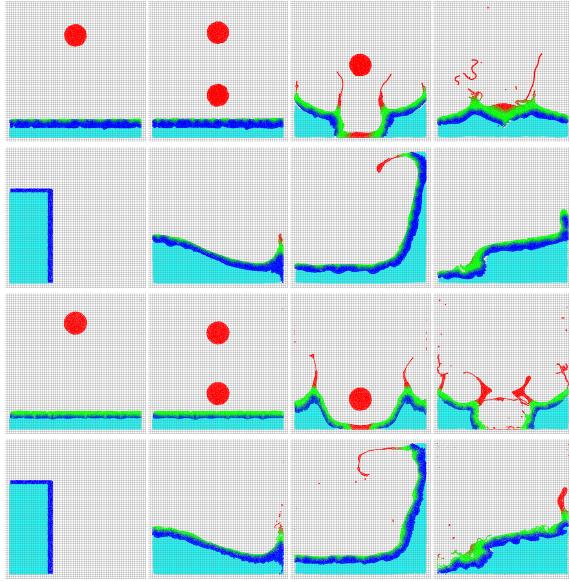


Figure 3: Snapshots from 2D simulations of a liquid ball dropping and a dam break. Our method is independent of the particle simulation method. PBF and SPH are used in the top and the bottom two rows, respectively.

rected grid velocities are then transferred back to the particles. Next, the particles are simulated, potentially with several substeps. Particles are then added and removed depending on several criteria. In case an SWE simulation is used outside the 3D grid domain, three more steps are necessary, namely, transferring the 3D grid velocities to the 2D SWE grid, solve the SWE and transfer the resulting velocities back to the 3D grid. The details for each step are discussed in the following sections.

3.1. Particle Influence on 3D Grid

In this section we describe how we compute an intermediate velocity field at time $t + \Delta t$ that combines particle and grid quantities. Since we are now at stage 5 of Algorithm 1, the grid velocity field is already updated to time $t + \Delta t$ but the particle states still correspond to time t . Therefore we first

predict the velocities and positions of the particles at time $t + \Delta t$ taking the full time step Δt as follows:

$$\bar{\mathbf{v}} = \mathbf{v} + \Delta t \left(\frac{\mathbf{f}_{\text{ext}}}{m} + \mathbf{g} \right) \quad (1)$$

$$\bar{\mathbf{x}} = \mathbf{x} + \Delta t \bar{\mathbf{v}}. \quad (2)$$

Here a full time step Δt is used instead of the particle simulation time step $\frac{\Delta t}{s_p}$ because the uncorrected velocity field at time $t + \Delta t$ is needed for pressure projection.

Next we rasterize the particles mass, m_l , and the predicted the velocities $\bar{\mathbf{v}}_l$ onto grids as

$$\rho_{ijk}^p = \frac{1}{\Delta x^3} \sum_l m_l \phi_{ijk}(\bar{\mathbf{x}}_l) \quad \text{and} \quad (3)$$

$$u_{(i+\frac{1}{2})jk}^p = \frac{\sum_l m_l \bar{\mathbf{v}}_{l,x} \phi_{(i+\frac{1}{2})jk}(\bar{\mathbf{x}}_l)}{\sum_l m_l \phi_{(i+\frac{1}{2})jk}(\bar{\mathbf{x}}_l)}, \quad (4)$$

where $\bar{\mathbf{x}}_l$ is the predicted position and $\phi_{ijk}(\bar{\mathbf{x}})$ is the tri-linear weight at the center of grid cell (i, j, k) with respect to position $\bar{\mathbf{x}}$. The y- and z-components of the velocities, $\mathbf{v}_{l,y}$ and $\mathbf{v}_{l,z}$ are rasterized in the same way. The mass m_l has the unit of liquid density multiplied by volume.

Next we combine the rasterized quantities ρ^p, u^p, v^p and w^p of the particles with quantities ρ^g, u^g, v^g and w^g of the 3D grid simulation as

$$\rho^c = \rho^g + \rho^p \quad (5)$$

$$\tilde{u}_{(i+\frac{1}{2})jk}^c = \frac{\rho_{(i+\frac{1}{2})jk}^g \tilde{u}_{(i+\frac{1}{2})jk}^g + \rho_{(i+\frac{1}{2})jk}^p u_{(i+\frac{1}{2})jk}^p}{\rho_{(i+\frac{1}{2})jk}^c}, \quad (6)$$

where

$$\rho_{(i+\frac{1}{2})jk}^c = \frac{\rho_{(i+1)jk} + \rho_{ijk}}{2}. \quad (7)$$

The y- and z-components of the combined velocity, \tilde{v}^c and \tilde{w}^c , are computed similarly. We use the tilde symbol to denote the intermediate velocity field before projection. Through these steps the particles impact the grid solver because the combined velocity field after projection will be used as the grid velocity field of the next time step.

3.2. Pressure Projection

To make the combined, intermediate velocity field $(\tilde{u}^c, \tilde{v}^c, \tilde{w}^c)$ divergence free, we use a multigrid approach

described [CM12] whose formulation is based on the variational framework of [BBB07]. We define cells with $\rho^c > 0.5$ to be liquid cells. If the grid based simulation is surrounded by an SWE simulation, the two need to be coupled properly. In this case Neumann boundary conditions are enforced at the common boundary.

The pressure projection step yields the divergence free combined velocity field (u^c, v^c, w^c) which we use as the new grid velocity field (u^g, v^g, w^g) at time $t + \Delta t$. In case of a surrounding SWE simulation, additional steps are necessary to ensure that the surfaces properly match up. These will be discussed in Sections 3.5 and 3.6.

3.3. 3D Grid Influence on Particles

It is natural to make the effect of the grid solver on the particles dependent on the distance of the particles to the grid fluid domain. When this distance is small the velocities of the particles should closely match the grid velocity field to ensure spatial continuity while particles far away from the grid should not be affected at all. In between, there should be some influence from the grid, predominantly the effect of the pressure projection step. To avoid temporal discontinuity, the distance measure should be smoothed over time. To avoid spatial discontinuity, these three types of interaction should be blended. All these requirements are achieved with our method described next (see Algorithm 2 for an overview).

We first evaluate the distance field to the fluid domain on the grid as

$$d^g = \begin{cases} 0 & \text{if } \rho^g < 0.5 \\ \text{distance to the iso-surface } \rho^g = 0.5 & \text{otherwise,} \end{cases} \quad (8)$$

using a GPU friendly method of [JW07]. For each particle, we compute a temporally smoothed distance d using the update rule

$$d \leftarrow d + c^{\text{damp}} (d^g|_{\mathbf{x}_i} - d), \quad (9)$$

where c^{damp} is a constant we set to $2/3$ in all examples. Temporal smoothing avoids excessive accelerations of particle due to sudden changes of the distance dependent coupling method.

Next we perform s_p substeps of the particle simulation. At the beginning of each substep, we compute the position and velocity predictions, \mathbf{x}' and \mathbf{v}' , for each particle as

$$\mathbf{v}' = \mathbf{v} + \frac{\Delta t}{s_p} \left(\frac{\mathbf{f}_{\text{ext}}}{m} + \mathbf{g} \right) \quad (10)$$

$$\mathbf{x}' = \mathbf{x} + \frac{\Delta t}{s_p} \mathbf{v}'. \quad (11)$$

Internal pressure forces are not considered for this prediction. Using these predicted quantities we compute PIC and FLIP based velocities for each particle

$$\mathbf{v}^{\text{PIC}} = \mathbf{u}^c|_{\mathbf{x}'} \quad (12)$$

$$\mathbf{v}^{\text{FLIP}} = \mathbf{v}' + \frac{\mathbf{u}^c - \tilde{\mathbf{u}}^c}{s_p}|_{\mathbf{x}'}. \quad (13)$$

The PIC velocity is a direct interpolation of the combined grid velocity field while the FLIP velocity only incorporates the change of the combined grid velocity field due to the pressure projection step. The particles are then coupled with the grid simulation using

$$\mathbf{v}'' = \begin{cases} \mathbf{v}^{\text{PIC}} & \text{if } d \leq d_g \\ \text{lerp}(\mathbf{v}^{\text{PIC}}, \mathbf{v}^{\text{FLIP}}, \frac{d-d_g}{d_{gf}-d_g}) & d_g < d \leq d_{gf} \\ \mathbf{v}^{\text{FLIP}} & \text{if } d_{gf} < d \leq d_f \\ \text{lerp}(\mathbf{v}^{\text{FLIP}}, \mathbf{v}', \frac{d-d_f}{d_p-d_f}) & d_f < d \leq d_p \\ \mathbf{v}' & \text{if } d_p < d, \end{cases} \quad (14)$$

where $d_g < d_{gf} < d_f < d_p$ are the distance thresholds and $. We use $d_g = \Delta x, d_{gf} = 2\Delta x, d_f = 4\Delta x$ and $d_p = 6\Delta x$ in all examples.$

By using these blending operations the velocity of particles close to the liquid domain ($d \leq d_g$) gets overwritten by the interpolated grid velocity while particles further away only incorporate the velocity corrections of the pressure projection step. In the latter case, incompressibility is approximately enforced and the high frequency details are preserved. Note that using ρ^c instead of ρ^g in the pressure projection step makes sure that the particle velocities become divergence free. Particles far away from the liquid domain ($d > d_p$) are not influenced by the grid solver at all.

Next we replace the particle velocities by the blended velocities, $\mathbf{v} \leftarrow \mathbf{v}''$. The predicted positions, \mathbf{x}' , are discarded as they are used only for computing PIC and FLIP velocities. At the end of the substep we run either a PBF or a SPH simulation step of size $\frac{\Delta t}{s_p}$. Note the computations on each particle are independent and hence can be parallelized on GPUs trivially.

```

1: Compute  $d^g$  on grid
2: Update  $d$  of each particle using interpolations of  $d^g$ 
3: for  $s_p$  steps do
4:   Compute particle position and velocity predictions  $\mathbf{x}'$  and  $\mathbf{v}'$  based on external forces only with step size  $\frac{\Delta t}{s_p}$ .
5:   Compute  $\mathbf{v}^{\text{PIC}}$  and  $\mathbf{v}^{\text{FLIP}}$  for each particle using  $\mathbf{x}'$ .
6:   Compute  $\mathbf{v}''$  as a distance dependent blend of  $\mathbf{v}'$ ,  $\mathbf{v}^{\text{PIC}}$  and  $\mathbf{v}^{\text{FLIP}}$ .
7:   Replace  $\mathbf{v}$  with  $\mathbf{v}''$  and discard  $\mathbf{x}'$ .
8:   Perform one SPH or PBF simulation step of size  $\frac{\Delta t}{s_p}$ .
9: end for

```

Algorithm 2: 3D Grid Influence on Particles

3.4. Particle Generation and Deletion

To preserve small scale fluid detail, we need to ensure that the region of interest is covered with particles. On the other hand, to reduce computational cost, particles that have wandered too deep into the grid fluid domain should be deleted.

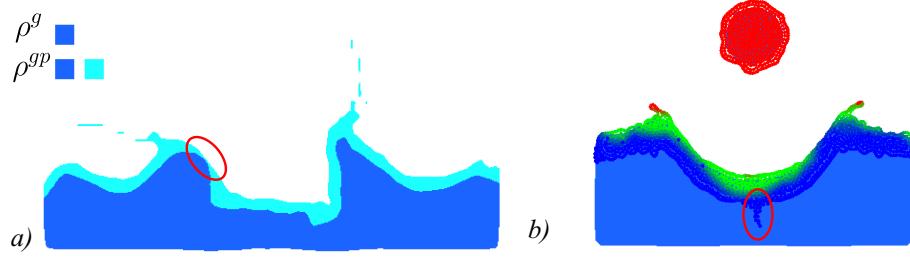


Figure 5: Particles creation and deletion. a) The region where $\rho^g \geq 0.5$ is shown in dark blue. The region where $\rho^c \geq 0.5$ is larger and additionally contains the light blue band. Particles are created in places where the blue band vanishes, i.e. inside the red circle. b) Several particles have traveled deep into the dark blue region and get deleted while their mass is absorbed by the grid.

To determine whether particles need to be created or removed, we first compute the distance field d^c with respect to the combined particle and grid density field as

$$d^c = \begin{cases} 0 & \text{if } \rho^c > 0.5 \\ \text{distance to the iso-surface } \rho^c = 0.5 & \text{otherwise.} \end{cases} \quad (15)$$

Again, we use the method of [JW07]. A grid density $\rho^g > 0$ where d^c is small indicates the absence of particles near the liquid surface (see Figure 5a). Therefore, we generate new particles in all cells with

$$\rho^g > 0 \text{ and } d^c < d^{\text{create}} \quad (16)$$

using Poisson Disk Sampling [Bri07]. Specifically, for each such cell we first randomly choose a position \mathbf{x} inside and create a new particle there if the following two conditions hold:

- $\phi_{ijk}(\mathbf{x}) \frac{m}{(\Delta x)^3} \geq \rho_{ijk}^g$ for all adjacent cells and
- $\rho^c|_{\mathbf{x}} > \rho_{\min}$
- There is no particle within distance r from position \mathbf{x} .

If these conditions hold, we add a particle at position \mathbf{x} and subtract $\phi_{ijk}(\mathbf{x}) \frac{m}{(\Delta x)^3}$ from the density ρ_{ijk}^g of adjacent cells to account for the gain in the combined density. Condition 1 makes sure that the ρ_{ijk}^g do not become negative. Condition 2 ensures we seed particle inside the liquid domain. We use $\rho_{\min} = 0.55$ in all examples. Condition 3 can be tested efficiently by leveraging the neighbor search data structure used for SPH or PBF simulations. The random sampling is repeated between $k = 5$ to 20 times in our examples. We use $d^{\text{create}} = 3\Delta x$ in all examples.

For particle deletion we iterate through all the particles and test whether they have moved too deep into the body of the liquid as shown in Figure 5(b). This is the case if $d^c|_{\mathbf{x}} > d^{\text{delete}}$. Each particle has a timer which is increased whenever this condition is met and is reset when it is not. If the time exceeds t^{delete} , we deposit its mass to adjacent grid cells using tri-linear weights and remove it. For all our examples we use $t^{\text{delete}} = 10\Delta t$ and $d^{\text{delete}} = 6\Delta x$. The use of a timer ensures that the particles which only temporarily move deep

into grid, such as those in a big vortex, do not get deleted immediately and still carry high frequency information around.

In case an SWE simulation is present we also need to check whether a given particle has left the 3D grid domain and entered the fluid volume represented by the SWE height field, i.e. has a y -coordinate smaller than the water height. If this is the case, we deposit the particle's mass in the SWE grid using bi-linear weights. The method is summarized in Algorithm 3. All the steps can be parallelized easily on GPUs with atomic operations.

```

1: Compute distance field  $d^c$ 
2: for each cell with  $\rho^g > 0$  and  $d^c < d^{\text{create}}$  do
3:   for  $k$  iterations do
4:     Generate random position  $\mathbf{x}$  within the cell
5:     if  $\mathbf{x}$  meets creation constraints then
6:       create particle at  $\mathbf{x}$  and decrease  $\rho^g$ 
7:     end if
8:   end for
9: end for
10: for each particle do
11:   if  $d^c|_{\mathbf{x}} > d^{\text{delete}}$  then
12:      $t^{\text{deep}} \leftarrow t^{\text{deep}} + \Delta t$ 
13:     if  $t^{\text{deep}} \geq t^{\text{delete}}$  then
14:       Remove particle and deposit its mass to nearby
         grid cells
15:     end if
16:   else
17:      $t^{\text{deep}} \leftarrow 0$ 
18:   end if
19:   if particle is outside 3D grid and  $\mathbf{x}_y < (h^s + H^s)|_{\mathbf{x}}$ 
      then
20:     Deposit mass to  $h^s$  in nearby cells.
21:   end if
22: end for

```

Algorithm 3: Particles generation and deletion

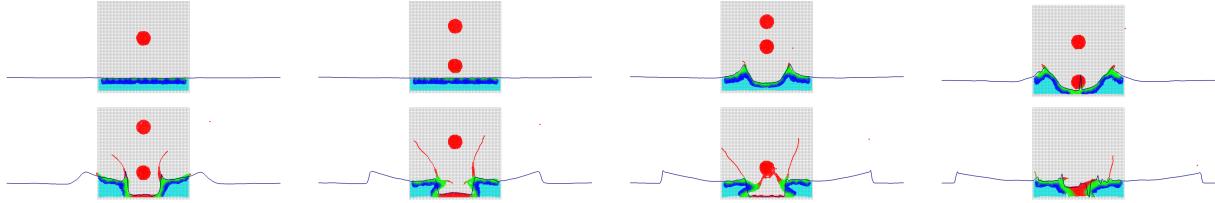


Figure 6: Combined 2D grid, particles and height field simulation. The wave generated by the dropping liquid ball propagates into the height field domain.

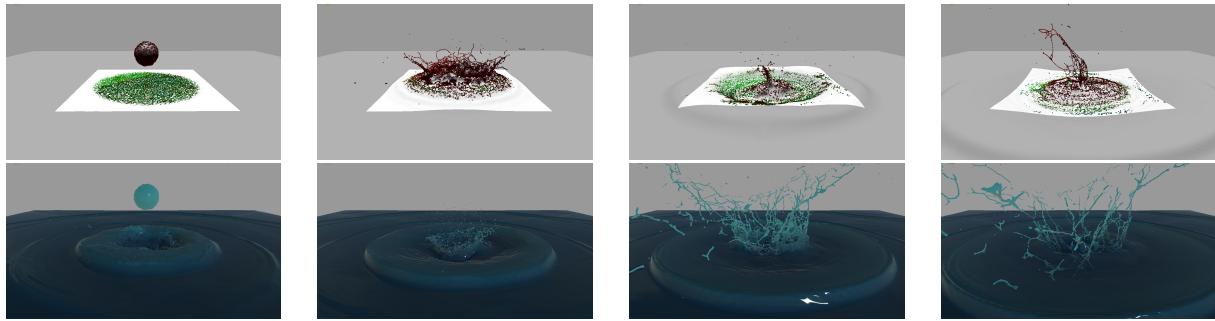


Figure 7: Combined 3D grid, particles and height field simulation. The wave generated in the 3D grid propagates smoothly outward into the height field domain.

3.5. 3D Grid Influence on the SWE grid

When simulating a height field outside of the 3D domain, additional steps are necessary to model the coupling between the two representations. To this end, we first define the region where the interaction takes place. We do this via a static field b_{ik} on a 2D grid with the following values:

$$\begin{aligned} b_{ik} = 0 & \quad \text{grid cells } (i, *, k) \text{ are controlled by the SWE} \\ b_{ik} = 1 & \quad \text{grid cells } (i, *, k) \text{ are controlled by the 3D solver} \\ 0 < b_{ik} < 1 & \quad \text{the simulations are blended.} \end{aligned}$$

In our examples, we set

$$b_{ik} = \text{clamp}\left(\frac{\frac{N}{2} - 1 - \sqrt{(i - \frac{N}{2})^2 + (k - \frac{N}{2})^2}}{N_b}, 0, 1\right), \quad (17)$$

where N is the size of the 3D grid and N_b is the width of the band where b linearly drops from 1 to 0 (we use $N_b = 5$).

At runtime, we determine the heights of the water columns $(i, *, k)$ in the 3D grid that belong to the main body of water. Specifically, we define the discrete height $\tilde{\eta}_{ik}$ of the water column $(i, *, k)$ to be the smallest j for which c^{air} consecutive cells above it are air cells. Grid cells are water (air) cells if $\rho_{ijk}^c \geq 0.5$ ($\rho_{ijk}^c < 0.5$). The rationale behind this definition is that we do not want the bottom of air bubbles to be identified as the liquid surface. In our examples we use $c^{\text{air}} = 3$. Other more sophisticated definitions could be used but we found that this efficient version works well in practice. The continuous height of the water surface, η_{ik}^c , can then be esti-

mated by finding the location where $\rho^c = 0.5$ near $\tilde{\eta}_{ik}$ using linear interpolation.

To couple the 3D grid and the SWE based simulation, we blend η_{ik}^c into h_{ik}^s using

$$h_{ik}^s \leftarrow (1 - b_{ik}) h_{ik}^s + b_{ik} (\eta_{ik}^c - H_{ik}) \quad (18)$$

We found that leaving the velocities u^s and w^s unaffected by 3D grid simulation yields visually plausible coupling results. A justification for this simplification is that h^s is modified a number of cells into the 3D grid every frame so the SWE solver has enough time to make u^s and w^s consistent with h^s before waves reach the interface. After this, we solve the SWE using [CMF10] for s_s substeps with a time step size of $\frac{\Delta t}{s_s}$. We use $s_s = 2$ in all examples with SWE.

3.6. SWE grid influence on the 3D Grid

To couple the 3D grid to the height field we modify the grid densities ρ^g such that the water depth induced by the union of the 3D grid and the particles approaches the depth of the height field. We achieve this by first computing a target density

$$\rho_{ijk}^s = \text{clamp}\left(\frac{1}{2} - \left((j + \frac{1}{2}) - \frac{(h_{ik} + H_{ik})}{\Delta x}\right) c^\rho, 0, 1\right). \quad (19)$$

This density is one below the height field and zero above with a smooth transition. The sharpness of the transition can be controlled by the constant c^ρ for which we use $c^\rho = 1$ in all our examples. Depending on b_{ik} we then pull the grid

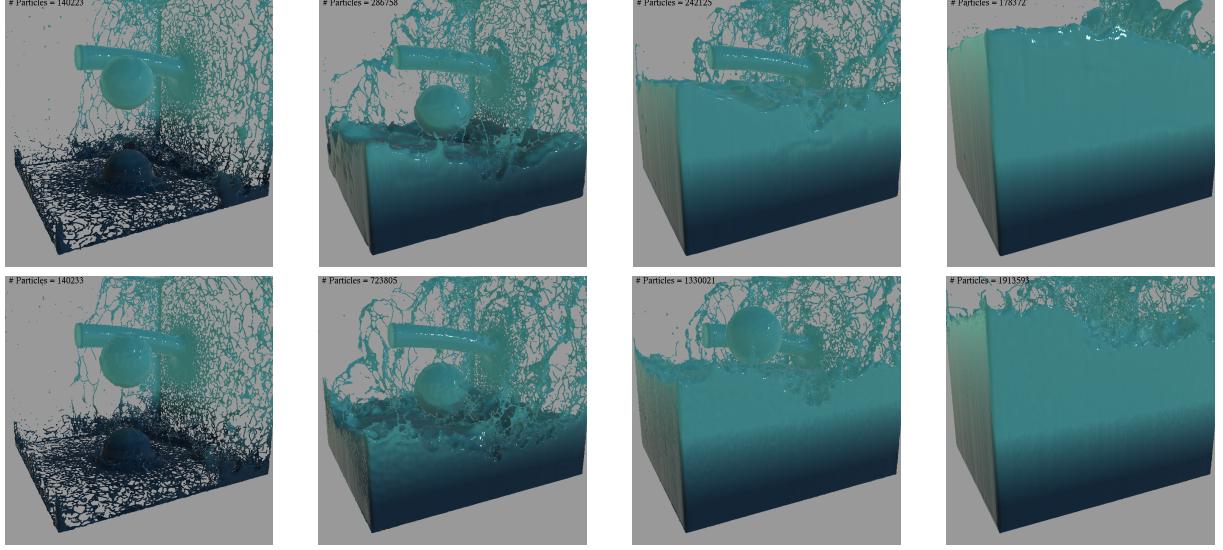


Figure 8: Hybrid approach (top row) and particle only approach (bottom row). In the hybrid case, as the jet fills the tank the number of particles levels off at around 180k while 1.9M particles are needed to represent all of the liquid at the end of the simulation. The total simulation time of hybrid approach is about 4X faster.

density towards this target density via

$$\rho_{ijk}^g \leftarrow \rho_{ijk}^g + (1 - b_{ik})(\rho_{ijk}^s - \rho_{ijk}^c). \quad (20)$$

Note that ρ^c is used on the right hand side instead of ρ^g because we want the update to bring the combined density closer to the target density ρ^s . We also update u^g to be closer to u^s using

$$u_{(i+\frac{1}{2})jk}^g \leftarrow u_{(i+\frac{1}{2})jk}^g + \left(1 - \frac{b_{ik} + b_{(i+1)k}}{2}\right)(u_{(i+\frac{1}{2})k}^s - u_{(i+\frac{1}{2})jk}^g), \quad (21)$$

where w^g is updated similarly.

4. High Resolution SWE grid

Inside the region covered with the 3D grid, small scale features are captured by the particles. This is not the case outside the 3D domain because particle does not exist there. Thus, small surface features have to be captured by the SWE height field itself. So far we have assumed that the resolutions of the 3D and the 2D grids match. We now relax this assumption and allow the 2D grid to have a higher resolution in order to make sure small details propagate outside the 3D domain. Using a higher resolution 2D grid is also a natural choice because the height field simulations are significantly less expensive than the full 3D simulations.

Let the spacing of the SWE grid be $\frac{\Delta x}{W}$, where $W = 2^e$ and e a non-negative integer. We now use two sets of grids, the low resolution grids h^s, H^s, u^s, w^s as before and the high resolutions grids $\bar{h}^s, \bar{H}^s, \bar{u}^s$ and \bar{w}^s . The SWE are solved on the high resolution grids. For the coupling, the quantities need to

be down-sampled. We use the following simple averaging:

$$h_{ik}^s = \frac{1}{W^2} \sum_{0 \leq a < W} \sum_{0 \leq b < W} \bar{h}_{(Wi+a)(Wk+b)}^s, \quad (22)$$

$$H_{ik}^s = \frac{1}{W^2} \sum_{0 \leq a < W} \sum_{0 \leq b < W} \bar{H}_{(Wi+a)(Wk+b)}^s, \quad (23)$$

$$u_{(i+\frac{1}{2})k}^s = \frac{1}{W} \sum_{0 \leq b < W} \bar{u}_{(Wi+\frac{1}{2})(Wk+b)}^s, \quad (24)$$

$$w_{i(k+\frac{1}{2})}^s = \frac{1}{W} \sum_{0 \leq a < W} \bar{w}_{(Wi+a)(Wk+\frac{1}{2})}^s, \quad (25)$$

which we found to be sufficient in our experiments. These down-sampled quantities are then used for the coupling described in Sections 3.6 and 3.5. After these steps we transfer the results back to the high resolution grids in a FLIP-style fashion. Specifically, we interpolate the changes in height, $h^s - \bar{h}^s$ using bi-linear interpolation, and add them back to \bar{h}^s , where \bar{h}^s are the values before the coupling step.

5. Results

We ran all 2D simulations on the CPU. For the 3D simulations we used a NVIDIA GTX 780Ti GPU and CUDA. The night time whale example uses 2 particle simulation substeps while all other examples use 4 substeps.

To test our method we first performed 2D simulations of two scenarios, falling liquid balls and a dam break scene. The results are shown in Figure 3. Our approach is independent of the choice of the particle simulation method used. The results with PBF are shown in the top two rows, while the results using SPH are shown in the bottom two rows.

Figure 6 shows a 2D simulation that uses an additional surrounding height field. Here, liquid balls drop into a large tank and generate waves and splashes as before but in this case, the waves inside the grid (gray region) propagate outwards into the height field.

We demonstrate the advantage of the hybrid approach over particles only approach in Figure 8. In this case, 3D grid and 180k particles are sufficient to represent the full tank while 1.9M particles are needed in a particle-only approach.

In Figure 4 we visualized the portion of water that is represented by particles in red. These dam break simulations show that our method is able to produce a highly detailed water surface and small scale detail with a rather low number of particles. Figure 7 shows 3D grid and height field coupling in 3D. The drop of the liquid ball creates a detailed splash in the center and waves that travel continuously into the height field domain.

Finally, we used our method to simulate the whale breaching scene shown in Figure 1. The whale generates a big splash when it falls back into the water. To enrich the scene, we added passively advected foam and ballistic spray particles. As expected, the splashed particles generate ripples when they fall back onto the surface of the ocean. This scene runs in real-time at around 30fps.

	GR	PC	SR	GT	PT	ST	CT
HybridFill	64x64x64	235k	-	9.21	58.12	-	7.24
ParFill	-	1.1M	-	-	296.80	-	-
Dambreak	128x64x32	160k	-	9.53	41.40	-	6.61
2Dambreak	128x64x32	183k	-	9.48	49.27	-	7.12
Pool	64x64x64	130k	256x256	9.06	32.62	0.14	5.64
WhaleDay	64x64x64	118k	512x512	9.18	29.80	0.62	5.32
WhaleNight	64x64x64	112k	512x512	9.12	15.24	0.64	5.41

Table 1: 3D grid resolution (GR), average particle count (PC), SWE resolution (SR), simulation times of grid (GT), particles (PT), SWE (ST), and coupling time (CT) of various 3D examples in ms.

6. Conclusion and Discussion

We presented a new method for coupling a grid, particles and shallow water solver to simulate large scale scenes with intermediate and small details at interactive rates. Our method does have some drawbacks. First, the boundary between the SWE and the 3D grid is assumed to not contain solids cells. If this is not the case, some modifications to the method would be required. Second, the SWE and 3D grids simulate inherently different physical phenomena. Therefore, our coupling algorithm does not model a true physical process. Also, the coupling parameters present in our method need to be tuned such that the number of particles remains in a reasonable range while still generating enough surface detail. The sizes and resolutions of the different simulation domains need to be defined as well to get the desired results.

However, we found a set of parameters that produced good results over the variety of our experiments without the need to re-tune from scene to scene.

One direction of future work is to generalize our method to handle non-uniform grids such as octrees, rectilinear grids and tall cell grids and to use adaptive particles fluid simulation techniques such as adaptive SPH [APKG07] and two scale SPH [SG11].

7. Acknowledgments

We would like to thank Miles Macklin for the PBF code and helpful discussion. We thank the members of the NVIDIA PhysX and APEX groups for their valuable inputs and support. We also thank the anonymous reviewers for their helpful comments.

References

- [APKG07] ADAMS B., PAULY M., KEISER R., GUIBAS L. J.: Adaptively sampled particle fluids. *ACM Trans. Graph.* 26 (July 2007), 2, 9.
- [ATT12] ANDO R., THUREY N., TSURUNO R.: Preserving fluid sheets with adaptively sampled anisotropic particles. *IEEE Transactions on Visualization and Computer Graphics* 18, 8 (Aug. 2012), 1202–1214. 2
- [ATW13] ANDO R., THÜREY N., WOJTAN C.: Highly adaptive liquid simulations on tetrahedral meshes. *ACM Trans. Graph.* 32, 4 (July 2013), 103:1–103:10. 2
- [BB12] BOYD L., BRIDSON R.: Multiflip for energetic two-phase fluid simulation. *ACM Trans. Graph.* 31, 2 (Apr. 2012), 16:1–16:12. 2
- [BBB07] BATTY C., BERTAILS F., BRIDSON R.: A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph.* 26, 3 (July 2007). 5
- [BR86] BRACKBILL J. U., REUPPEL H. M.: Flip: a method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *J. Comp. Phys.* 65 (1986), 314–343. 2
- [Bri05] BRIDSON R.: Shallow water discretization, Lecture notes Animation Physics. University of British Columbia, 2005. 3
- [Bri07] BRIDSON R.: Fast poisson disk sampling in arbitrary dimensions. In *ACM SIGGRAPH 2007 sketches* (2007), SIGGRAPH '07, ACM. 6
- [BT07] BECKER M., TESCHNER M.: Weakly compressible sph for free surface flows. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 209–217. 2
- [BXH10] BATTY C., XENOS S., HOUSTON B.: Tetrahedral embedded boundary methods for accurate and flexible adaptive fluids. In *Proc. Eurographics* (2010). 2
- [CFL*07] CHENTANEZ N., FELDMAN B. E., LABELLE F., O'BRIEN J. F., SHEWCHEUK J. R.: Liquid simulation on lattice-based tetrahedral meshes. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 219–228. 2
- [CM11] CHENTANEZ N., MÜLLER M.: Real-time Eulerian water simulation using a restricted tall cell grid. *ACM Trans. Graph.* 30, 4 (July 2011), 82:1–82:10. 2, 3
- [CM12] CHENTANEZ N., MÜLLER M.: Mass-conserving eulerian liquid simulation. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012), pp. 245–254. 3, 5

- [CMF10] CHENTANEZ N., MÜLLER-FISCHER M.: Real-time simulation of large bodies of water with small scale details. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2010). 3, 7
- [DC96] DESBRUN M., CANI M.-P.: Smoothed particles: A new paradigm for animating highly deformable bodies. In *Eurographics Workshop on Computer Animation and Simulation, EGCA '96, August, 1996* (Poitiers, France, 1996), Boulic R., Hégron G., (Eds.), Springer-Verlag, pp. 61–76. Published under the name Marie-Paule Gascuel. 2
- [EMF02] ENRIGHT D., MARSCHNER S., FEDKIW R.: Animation and rendering of complex water surfaces. *ACM Trans. Graph.* 21, 3 (July 2002), 736–744. 2
- [EQYF13] ENGLISH R. E., QIU L., YU Y., FEDKIW R.: Chimera grids for water simulation. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2013), ACM, pp. 85–94. 2
- [FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *Proc. SIGGRAPH* (Aug. 2001), pp. 23–30. 2
- [FM96] FOSTER N., METAXAS D.: Realistic animation of liquids. In *Graphics Interface 1996* (May 1996), pp. 204–212. 2
- [GM77] GINGOLD R. A., MONAGHAN J. J.: Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 181 (1977), 375–389. 2
- [Har63] HARLOW F. H.: The particle-in-cell method for numerical solution of problems in fluid dynamics. *Experimental arithmetic, high-speed computations and mathematics* (1963). 2
- [HHL*05] HAGEN T. R., HJELMERVIK J. M., LIE K.-A., NATVIG J. R., HENRIKSEN M. O.: Visual simulation of shallow-water waves. *Simulation Modelling Practice and Theory* 13, 8 (2005), 716–726. 3
- [HYLK08] HONG J.-M., LEE H.-Y., YOON J.-C., KIM C.-H.: Bubbles alive. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 48:1–48:4. 2
- [HW04] HOLMBERG N., WÜNSCHE B. C.: Efficient modeling and rendering of turbulent water over natural terrain. In *Proc. GRAPHITE* (2004), pp. 15–22. 3
- [IGLF06] IRVING G., GUENDELMAN E., LOSASSO F., FEDKIW R.: Efficient simulation of large bodies of water by coupling two- and three-dimensional techniques. In *Proc. SIGGRAPH* (Aug. 2006), pp. 805–811. 2
- [JW07] JEONG W.-K., WHITAKER R. T.: A fast eikonal equation solver for parallel systems. In *SIAM conference on Computational Science and Engineering* (2007). 5, 6
- [KCC*06] KIM J., CHA D., CHANG B., KOO B., IHM I.: Practical animation of turbulent splashing water. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 335–344. 2
- [KM90] KASS M., MILLER G.: Rapid, stable fluid dynamics for computer graphics. In *Proc. SIGGRAPH* (1990), pp. 49–57. 3
- [LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. In *Proc. SIGGRAPH* (2004), pp. 457–462. 2
- [LSSF06] LOSASSO F., SHINAR T., SELLE A., FEDKIW R.: Multiple interacting liquids. *ACM Trans. Graph.* 25 (July 2006), 812–819. 2
- [LTKF08] LOSASSO F., TALTON J., KWATRA N., FEDKIW R.: Two-way coupled sph and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (2008), 797–804. 2
- [LvdP02] LAYTON A. T., VAN DE PANNE M.: A numerically efficient and stable algorithm for animating water waves. *The Visual Computer* 18, 1 (2002), 41–53. 3
- [MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *the ACM SIGGRAPH 2003 Symposium on Computer Animation* (Aug. 2003), pp. 154–159. 2
- [MM13] MACKLIN M., MÜLLER M.: Position based fluids. *ACM Trans. Graph.* 32, 4 (July 2013), 104:1–104:12. 2, 3
- [MMTD07] MULLEN P., MCKENZIE A., TONG Y., DESBRUN M.: A variational approach to Eulerian geometry processing. In *ACM SIGGRAPH 2007 papers* (2007), SIGGRAPH '07, ACM. 2, 3
- [MY97] MOULD D., YANG Y.-H.: Modeling water for computer graphics. *Computers & Graphics* 21, 6 (1997), 801–814. 3
- [RWT11] RAVEENDRAN K., WOJTAN C., TURK G.: Hybrid smoothed particle hydrodynamics. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2011), ACM, pp. 33–42. 2
- [SCS94] SULSKY D., CHEN Z., SCHREYER H.: A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering* (1994), 118:179–196. 2
- [SG11] SOLENTHALER B., GROSS M.: Two-scale particle simulation. *ACM Trans. Graph.* 30 (Aug. 2011), 81:1–81:8. 2, 9
- [SP09] SOLENTHALER B., PAJAROLA R.: Predictive-corrective incompressible sph. *ACM Trans. Graph.* 28, 3 (July 2009), 40:1–40:6. 2, 3
- [SSC*13] STOMAKHIN A., SCHROEDER C., CHAI L., TERAN J., SELLE A.: A material point method for snow simulation. *ACM Trans. Graph.* 32, 4 (July 2013), 102:1–102:10. 2
- [Tes99] TESSENDORF J.: Simulating ocean water. SIGGRAPH course notes, 1999. 3
- [TR04] THÜREY N., RÜDE U.: Free Surface Lattice-Boltzmann fluid simulations with and without level sets. *Proc. of Vision, Modelling, and Visualization VMV* (2004), 199–207. 2
- [TRS06] THÜREY N., RÜDE U., STAMMINGER M.: Animation of open water phenomena with coupled shallow water and free surface simulations. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 157–164. 3
- [WMT07] WANG H., MILLER G., TURK G.: Solving general shallow wave equations on surfaces. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 229–238. 3
- [WZKQ13] WANG C.-B., ZHANG Q., KONG F.-L., QIN H.: Hybrid particle-grid fluid animation with enhanced details. *The Visual Computer* 29, 9 (2013), 937–947. 2
- [YHK07] YUKSEL C., HOUSE D. H., KEYSER J.: Wave particles. In *Proc. SIGGRAPH* (2007), p. 99. 3
- [ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Trans. Graph.* 24, 3 (July 2005), 965–972. 2
- [ZLC*13] ZHU B., LU W., CONG M., KIM B., FEDKIW R.: A new grid structure for domain extension. *ACM Trans. Graph.* 32, 4 (July 2013), 63:1–63:12. 2
- [ZZKF13] ZHENG W., ZHU B., KIM B., FEDKIW R.: A new incompressibility discretization for a hybrid particle mac grid representation with surface tension. *Submitted* (2013). 2