DOI: 10.1111/cgf.70357

EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 2

Adaptive Optical Layers: Efficient Tall Cell Grids for Liquid Simulation

Fumiya Narita[1,2] and Takashi Kanai[1]

[1]The University of Tokyo, Japan [2]GAME FREAK Inc., Japan

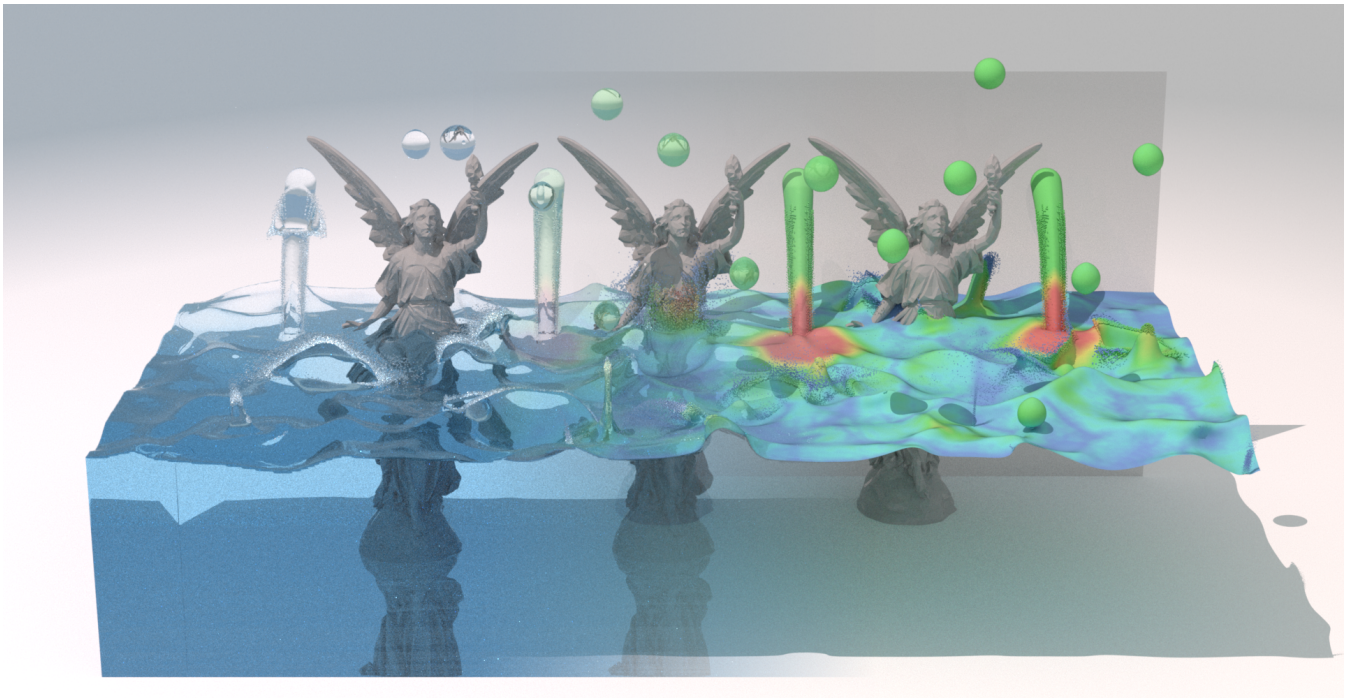


Figure 1: *Lucy in the Rain: Water is poured onto the surface from multiple sources, while many small spherical droplets, resembling rainfall, fall into the liquid. The right shows the liquid surface color-coded by optical layer thickness: blue indicates thin layers, and colors near red represent thicker layers. Thicker layers appear in regions with strong liquid motion, such as where water is poured or droplets strike the surface. The left shows the corresponding transparent rendering.*

Abstract

Tall cell grids have been proposed as an efficient approach to accelerate large-scale liquid simulation. In this framework, regions near the liquid surface are discretized with regular grids, while regions farther away are represented by elongated rectangular cells. The regular grid region close to the surface is referred to as the optical layer. In previous work, the thickness of this optical layer was uniformly fixed across the entire liquid domain. In this paper, we propose a novel tall cell grid structure in which the thickness of the optical layer is dynamically adjusted according to the motion of the liquid. This adaptive strategy reduces the number of grid cells required in the projection step without compromising visual quality, thereby accelerating the overall simulation. Furthermore, we introduce a two-way coupling scheme between rigid bodies and liquids in regions where the optical layer remains thin. Our algorithm is simple and can be easily integrated into existing tall cell grid frameworks.

CCS Concepts

• ***Computing methodologies*** → ***Physical simulation;***

1. Introduction

Eulerian liquid simulation is a central technique in computer graphics. However, as the number of grid cells increases, the computational cost grows rapidly, with pressure projection being the primary bottleneck [LZF10]. To address this issue, various efficient grid structures have been proposed. A common example is the octree grid [LGF04; SABS14; AGL*17; AB20], but other structures such as tetrahedral grids [KFCO06; CFL*07; ATW13] and tall cell grids [IGLF06; CM11; NOKA25] have also been explored.

Tall cell grids combine the traditional two-dimensional height-field approach with three-dimensional Eulerian liquid simulation. Similar to the height-field method, the bulk of the liquid is represented by elongated rectangular cells in which pressure is assumed to vary linearly. To overcome the limitations of height-field approaches—such as their inability to capture effects like entrainment or splashes—the liquid surface is instead resolved using a full three-dimensional Euler equation solver.

In other words, tall cell grids discretize space with regular cubic grids near the liquid surface, while elongated rectangular cells are used farther away from the surface. The region close to the liquid surface is referred to as the *optical layer* (or optical depth in some references). By reducing the number of grid cells along the vertical direction, this approach decreases the total number of pressure unknowns, thereby accelerating the pressure solve. The thickness of the optical layer and the accuracy of the simulation are in a trade-off relationship: increasing the layer thickness yields results closer to those of a uniform grid (Figure 2). In previous work [IGLF06; CM11; NOKA25], the optical layer thickness was set uniformly to one quarter of the liquid height across the entire domain.

In this paper, we extend this idea by proposing a tall cell grid in which the optical layer thickness is varied dynamically according to the motion of the liquid. By assigning thicker optical layers in regions with more dynamic liquid motion, our method achieves faster projection steps without loss of visual quality, resulting in an overall speedup of the simulation. The proposed algorithm is simple and can be easily integrated into existing tall cell grid frameworks. Altogether, our contributions are listed as follows:

- A tall cell grid in which the optical layer thickness is dynamically adapted per region according to liquid motion.
- A two-way coupling scheme for rigid body interactions with tall cell grids in regions of thin optical layers.

Through several test cases, we show that our method achieves faster projection steps while maintaining visual quality compared with prior work [IGLF06]. The proposed approach takes full advantage of the tall cell grid structure.

2. Related Work

Our work builds upon inviscid Eulerian liquid simulation methods. For an in-depth overview of this field, we refer the reader to the textbook by Bridson [Bri15]. Below we summarize the studies most closely related to our research.

2.1. Eulerian Liquid Simulation and FLIP

Eulerian fluid simulation was first introduced to computer graphics by Foster and Metaxas [FM96]. Subsequently, Stam proposed Stable Fluids [Sta99], enabling stable simulations through Semi-Lagrangian advection. Foster and Fedkiw [FF01] then extended this approach to handle liquid simulation, and later, second-order boundary conditions for both free surfaces [ENGF03] and solids [NMG09] were also incorporated. A well-known drawback of Eulerian methods is numerical dissipation. To address this issue, Zhu and Bridson introduced the Fluid Implicit Particle (FLIP) method [ZB05], a hybrid approach combining particles with grids. More recently, FLIP has been extended in several directions: the Narrow Band FLIP (NBFLIP) method [FAW*16] restricts particles to a band near the liquid surface, while the Extended Narrow Band FLIP (EXNBFLIP) [SWT*18] further concentrates particles in surface regions of particular importance. Following Narita et al. [NOKA25], we employ EXNBFLIP as our base simulator, though more general schemes such as FLIP or the Affine Particle-in-Cell (APIC) method [JSS*15] could also be used.

2.2. Spatially Adaptive Methods

Eulerian liquid simulation suffers from rapidly increasing computational cost as the number of grid cells grows. To address this problem, a variety of grid structures have been proposed. Losasso et al. [LGF04] proposed octree-based liquid simulation, later refined by improvements such as better treatments of T-junctions [LFO06] and memory-efficient structures like the sparse paged grid (SPGrid) [SABS14; AGL*17]. More recently, Ando et al. [AB20] proposed a practical method based on a variational pressure solver that is both efficient and easy to implement. Alternatively, tetrahedral-grid-based fluid simulation methods have also been explored. Klingner et al. [KFCO06] proposed a method for smoke simulation, which was later extended to liquids by Chentanez et al. [CFL*07]. Ando et al. [ATW13] proposed a novel discretization combined with a variational pressure solver. In addition, a variety of other grid structures have been proposed; stretched cells [ZLC*13], tall cell grids [IGLF06; CM11; NOKA25], AST grids [XCW*20], and other techniques that couple 2D and 3D representations [CMK15; ZY10]. A more comprehensive discussion of spatially adaptive grid structures is beyond the scope of this paper, and we refer the reader to the survey by Manteaux et al. [MWN*17] for further details.

2.3. Tall Cell Grids

Only a limited number of studies have focused on tall cell grids. The concept was first introduced by Irving et al. [IGLF06], where the optical layer thickness was kept uniform across the liquid region. They also discussed the visual impact of varying the optical layer thickness, showing through simple experiments that setting it to one quarter of the domain height yields results close to those of a full-resolution simulation. Subsequently, Chentanez et al. [CM11] simplified tall cell grids and implemented them on the GPU, enabling stable real-time simulation. Since the coefficient matrix of their pressure solver was not symmetric positive definite, they could not employ the Incomplete Cholesky Conjugate Gradient (ICCG) method and instead relied on a multigrid solver. Their approach

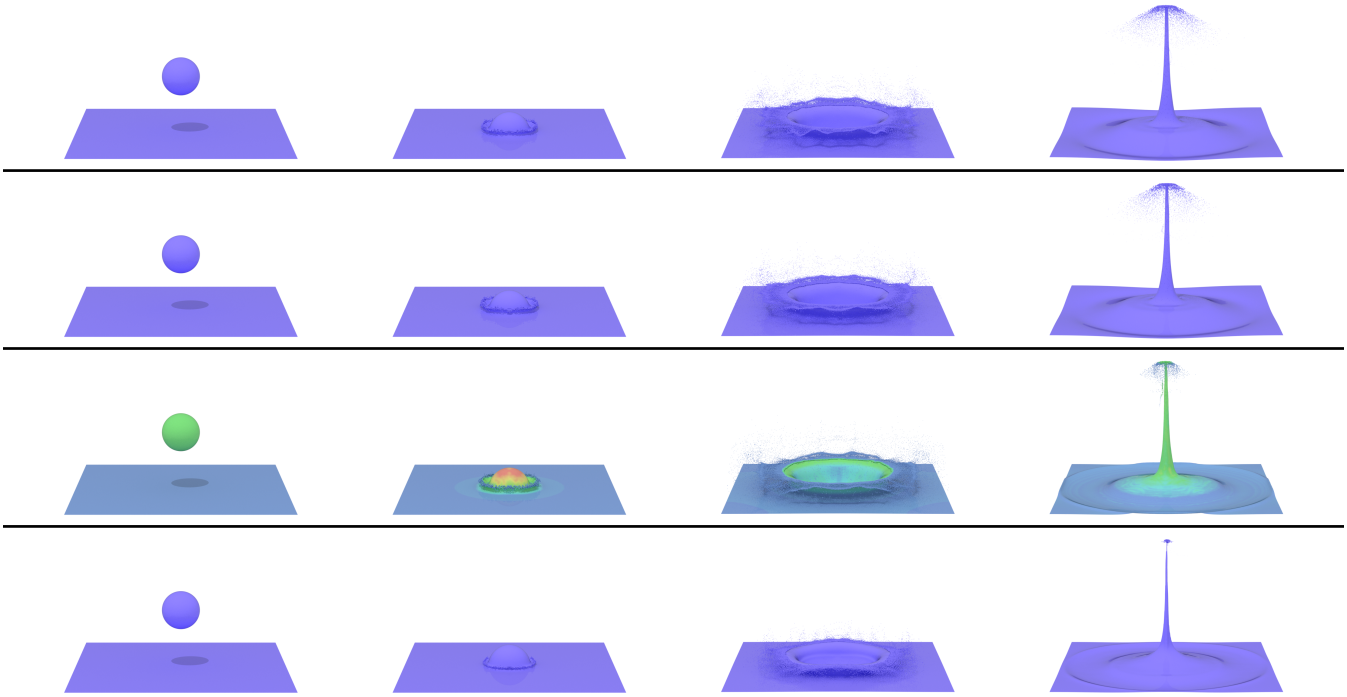


Figure 2: *Water Drop: From top to bottom: results with a uniform grid, a tall cell grid, our method, and a tall cell grid with uniformly reduced optical layer thickness. Our method achieves results comparable to the tall cell grid while using only about one fifth of the grid cells. In contrast, reducing the optical layer uniformly to one fifth fails to capture dynamic liquid behavior, resulting in poor representation of features such as crowns and Worthington columns.*

also assumed a fixed optical layer thickness. More recently, Narita et al. [NOKA25], inspired by Batty and colleagues [BBB07], reformulated tall cell grids using a variational approach. This ensured that the resulting system matrix was symmetric positive definite while simplifying implementation. They further extended tall cell grids with horizontal adaptivity and introduced two-way coupling between rigid bodies and liquids within the optical layer region. Nevertheless, their method also assumed on a fixed optical layer thickness. In contrast, our approach introduces a grid structure in which the optical layer thickness varies spatially according to the liquid surface. Moreover, whereas Narita et al. [NOKA25] considered two-way coupling only within the optical layer, we extend this to handle cases where rigid bodies intersect tall cells as well.

3. Our Method

As in many previous studies, our proposed method assumes a rectangular box domain with a resolution of (N_x, N_y, N_z). In our formulation, (i, j, k) denote the $x-$, $y-$, and $z-$axes respectively, with j representing the vertical direction. Quantities such as $X(i,k)$ are defined per column and therefore independent of j.

3.1. Adaptive Optical Layer

In this section, we describe our approach for constructing the adaptive optical layer as a function of liquid motion. To guarantee stability of the simulation, the optical layer with adaptive thickness is represented as a spatially smooth field through a two-stage procedure. An overview of this procedure is illustrated in Figure 3.

3.1.1. Estimating the Shape of Adaptive Optical Layer

In the first stage, we obtain an initial estimate of the adaptive optical layer by applying a dilation operation with distances that vary spatially from the liquid surface. This stage corresponds to steps (a) through (e) in Figure 3.

First, we evaluate a cost function for each grid cell to characterize liquid motion (Figure 3 (a)). Several formulations of this function can be considered. For example, velocity gradients and surface curvature have often been used to guide adaptive grids [ATW13; AB20]. However, surface curvature is prone to producing extreme values and is therefore not suitable for our purpose. Instead, we design the error measure by drawing inspiration from the height-field origins of tall cell grids, as shown in Figure 4.

For each column, we construct a single tall cell spanning from the ground to the liquid surface. (This is distinct from the tall cells later used in the projection step.) For each grid face, we compute the velocity difference before and after conversion into this tall cell. Let the accumulated error at grid position (i, j, k) be denoted as $e(i, j, k)$. $e(i, j, k) = \sum_f \left| v_{f,\text{after}} - v_{f,\text{before}} \right|$, where $v_{f,\text{after}}$ and $v_{f,\text{before}}$ represent the velocities on face f after and before the conversion, respectively. The accumulated value represents how much the local velocity field deviates from the assumptions of the tall cell representation.

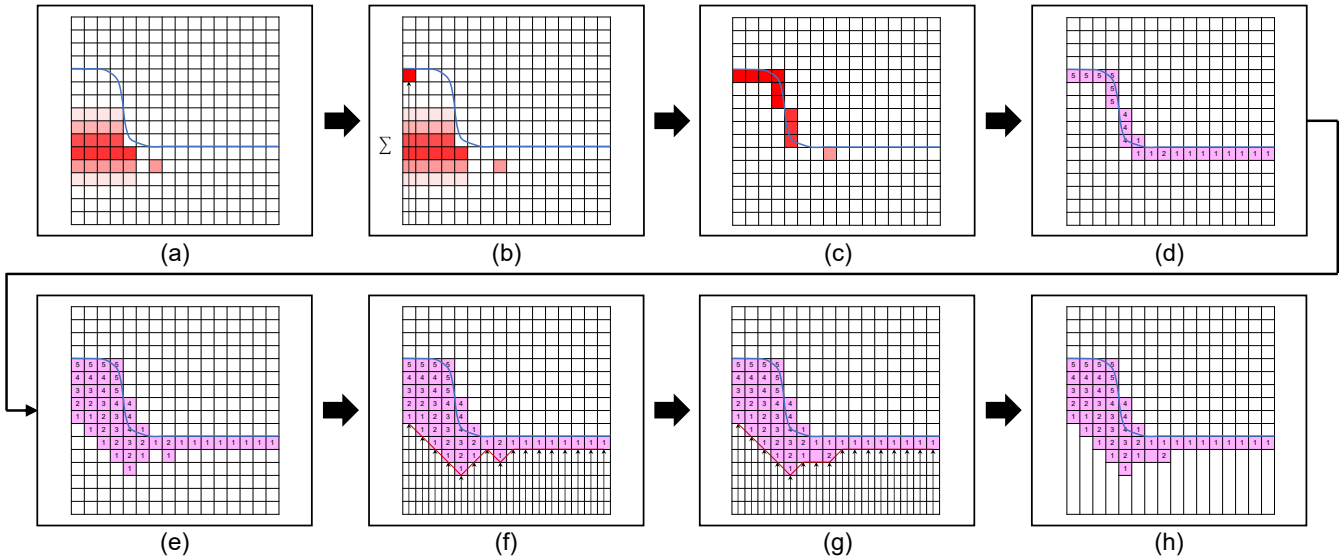


Figure 3: *Overview of our method for constructing the spatially adaptive optical layer. In (a)–(c), red indicates the magnitude of the error, while in (d)–(f), purple marks the grid cells included in the optical layer.*

Next, for each column, we perform a vertical scan to sum the error values and map the accumulated error from the ground to the liquid surface (Figure 3 (b)). Let the accumulated error at column (i,k) be denoted as $E(i,k)$. $E(i,k) = \sum_{j=j_{\text{ground}}}^{j_{\text{surface}}} e(i,j,k)$, where j_{ground} and j_{surface} denote the grid indices of the ground and the fluid surface cell in contact with the ground, respectively. This procedure is carried out for all columns (Figure 3 (c)).

Several choices are possible for defining the dilation distance $d(i,k)$. From our experiments, we adopt the following formulation:

$$d(i,k) = f\big(E(i,k)\big) = \text{clamp}\big(\alpha E(i,k)\Delta x, d_{\min}, d_{\max}\big), \quad (1)$$

where α is a parameter controlling the effect by the dilation distance, Δx is the grid width, and $d_{\min}$ and $d_{\max}$ are the minimum and maximum dilation distances. Empirically, we set $\alpha = 0.5$, $d_{\min} = \max(4, N_y/64)$, and $d_{\max} = N_y/8$, where N_y denotes the number of vertical grid cells (Figure 3 (d)). Note that if $d(i,k) = d_{\max}$ at all liquid surface grid points, the result coincides with the conventional tall cell method.

Finally, we apply dilation from the liquid surface using the spatially varying distances $d(i,k)$, which produces an initial estimate of the adaptive optical layer (Figure 3 (e)). For simplicity, we use the Manhattan distance for the dilation operation.

3.1.2. Smoothing the Optical Layer

If the optical layer from the previous stage is used directly, the tall cell grid height may change abruptly. Such discontinuities can cause numerical instabilities or even lead to simulation failure. To prevent this, the second stage smooths the optical layer so that tall cell height varies gradually. This corresponds to steps (f) through (h) in Figure 3.

First, for each column, we scan vertically to measure the distance $h(i,k)$ (in grid cells) from the ground to the boundary of the optical layer and store these values in a two-dimensional array on the xz-plane (Figure 3 (f)). The values are converted to floating-point type, and a smoothing filter is applied under the constraint that tall cells must not exceed their pre-smoothed height. This step is implemented with a conditional moving filter, after which the smoothed values are then cast back to integers.

For the smoothing step, our method is not tied to a specific filter; any reasonable 2D smoothing operator can be used. In our implementation, we use a constrained 2D moving-average filter (9×9 window, 5 iterations) that ensures the smoothed height does not exceed the original values.

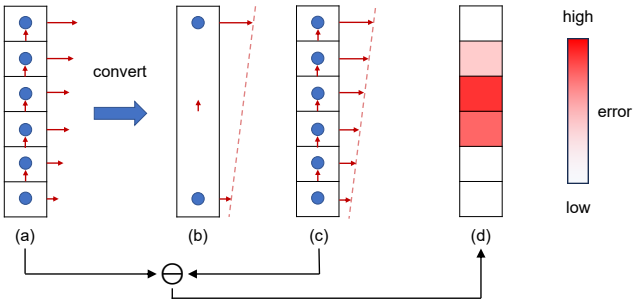


Figure 4: *Computation of the error value. (a) Grid cells from the ground to the liquid surface. (b) These cells are converted into a single tall cell. The horizontal velocity is obtained by least-squares linear fitting, and the vertical velocity is computed as the average [IGLF06]. (c) The tall cell velocity is linearly interpolated back to each grid position. (d) The error value is defined as the sum of the absolute differences between the original grid velocities in (a) and the interpolated velocities in (c).*

Let $h_{\text{before}}(i,k)$ and $h_{\text{after}}(i,k)$ denote the height values of $h(i,k)$ before and after smoothing, respectively. If $h_{\text{before}}(i,k) - h_{\text{after}}(i,k) > 0$, we include the grid cells between these two heights in the optical layer; these correspond to the purple grid cells without a dilation count in Figure 3 (g). This operation effectively smooths the tall cell height. Based on the resulting optical layer, the final tall cell grid is then constructed (Figure 3 (h)).

3.1.3. Handling Regions Above the Liquid Surface

The previous stage determines the adaptive optical layer for liquid regions in contact with the ground. However, liquid may also appear above the main surface due to falling drops or splashes. We handle these regions using the following procedure:

1. For each column, scan vertically to detect the liquid surface in contact with the ground. The dilation distance of this surface is denoted $d(i,k)$.
2. For liquid surfaces within a distance D_{offset} of this surface, we assign the same value $d(i,k)$. As a result, airborne liquid that is about to collide with a ground-contacting liquid surface is assigned the same dilation count as that surface.
3. For surfaces farther than D_{offset}, we assign a constant value d_{air}. As a result, airborne liquid portions separated from the main surface are assigned a constant dilation count.
4. Perform the dilation operation to obtain the final adaptive optical layer.

Empirically, we set $D_{\text{offset}} = \max(4, N_y/32)$ and $d_{\text{air}} = N_y/16$, where N_y is the number of vertical grid cells. Since splashes and liquid jets can deform, we deliberately choose a relatively large value for d_{air}.

3.2. Projection and Two-Way Coupled Rigid Bodies

In the previous section, we constructed the grid structure necessary for pressure projection. Here, we present the pressure projection step along with the two-way coupling scheme.

3.2.1. Preliminaries

Following Narita et al. [NOKA25], we adopt a variational pressure solver:

$$-[\nabla]^T[V][A][F][\nabla]\{p\} = -[\nabla]^T[V][A]\{\boldsymbol{u}^*\}, \tag{2}$$

where $[\nabla]$ and $-[\nabla]^T$ denote the discretized gradient and divergence operators, respectively. $[F]$ and $[A]$ are diagonal matrices representing the fluid area-volume fraction near free surfaces and the flux-area fraction near solids. Finally, $\{p\}$, $\{\boldsymbol{u}^*\}$ and $[V]$ denote the vectorized pressure, the velocity after advection, and a diagonal matrix encoding face–cell volumes, respectively. For details on the construction of these vectors and matrices in the presence of tall cells, we refer the reader to Narita et al. [NOKA25].

To incorporate two-way coupled rigid bodies, (2) can be extended as follows [BBB07]:

$$\begin{aligned}\Delta t\left([\nabla]^T[V][A][F][\nabla] + [J]^T[M_s]^{-1}[J]\right)\{p\} \\ = [\nabla]^T[V][A]\{\boldsymbol{u}^*\} - [J]^T\{\boldsymbol{w}^*\},\end{aligned} \tag{3}$$

where $[J] = [J_{\text{trans}}^T \;\; J_{\text{rot}}^T]^T$, and $\boldsymbol{w}^*$ is a 6×1 vector containing the linear and angular velocity of the rigid body. $[M_s]$ is the 6×6 mass matrix associated with a rigid body in contact with the fluid. $[J_{\text{trans}}]$ and $[J_{\text{rot}}]$ satisfy the following relations:

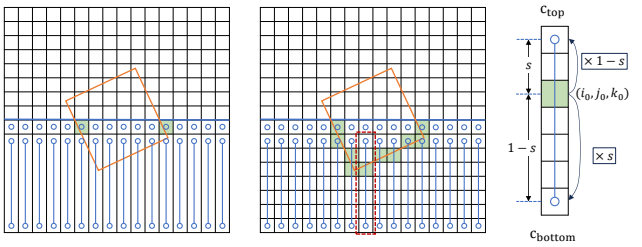


Figure 5: *A two-way coupling scheme for cases where rigid bodies intersect a tall cell. Left: a rigid body intersecting a tall cell. Center: a virtual uniform grid embedded within the tall cell. Right: values computed on the virtual grid cells intersecting the rigid body (shown in green) are distributed to the top and bottom pressure samples of the tall cell.*

$$-[J_{\text{trans}}]\{p\} = \boldsymbol{f}_{\text{trans}} = \iint_s p\boldsymbol{n}dS, \tag{4}$$

$$-[J_{\text{rot}}]\{p\} = \boldsymbol{\tau}_{\text{rot}} = \iint_s (\boldsymbol{x} - \boldsymbol{X}_{\text{com}}) \times p\boldsymbol{n}dS, \tag{5}$$

where $\boldsymbol{f}_{\text{trans}}$ and $\boldsymbol{\tau}_{\text{rot}}$ are the net force and torque acting on the rigid body. Here, $\boldsymbol{X}_{\text{com}}$ denotes the center of mass of the rigid body, and $\boldsymbol{n}$ is the surface normal.

Following Batty et al. [BBB07], we transform (4) and (5) into volume integrals, yielding

$$[J]_1^{i,j,k} = \Delta x^2\left(A'_{i+\frac{1}{2},j,k} - A'_{i-\frac{1}{2},j,k}\right), \tag{6}$$

$$[J]_2^{i,j,k} = \Delta x^2\left(A'_{i,j+\frac{1}{2},k} - A'_{i,j-\frac{1}{2},k}\right), \tag{7}$$

$$[J]_3^{i,j,k} = \Delta x^2\left(A'_{i,j,k+\frac{1}{2}} - A'_{i,j,k-\frac{1}{2}}\right), \tag{8}$$

$$\begin{aligned}[J]_4^{i,j,k} = &-\Delta x^2(z-Z)\left(A'_{i,j+\frac{1}{2},k} - A'_{i,j-\frac{1}{2},k}\right) \\ &+\Delta x^2(y-Y)\left(A'_{i,j,k+\frac{1}{2}} - A'_{i,j,k-\frac{1}{2}}\right),\end{aligned} \tag{9}$$

$$\begin{aligned}[J]_5^{i,j,k} = &-\Delta x^2(x-X)\left(A'_{i,j,k+\frac{1}{2}} - A'_{i,j,k-\frac{1}{2}}\right) \\ &+\Delta x^2(z-Z)\left(A'_{i+\frac{1}{2},j,k} - A'_{i-\frac{1}{2},j,k}\right),\end{aligned} \tag{10}$$

$$\begin{aligned}[J]_6^{i,j,k} = &-\Delta x^2(y-Y)\left(A'_{i+\frac{1}{2},j,k} - A'_{i-\frac{1}{2},j,k}\right) \\ &+\Delta x^2(x-X)\left(A'_{i,j+\frac{1}{2},k} - A'_{i,j-\frac{1}{2},k}\right),\end{aligned} \tag{11}$$

where Δx is grid width, $A' = 1 - A$ and $[X,Y,Z] = \boldsymbol{X}_{\text{com}}$. (x,y,z) is the center of cell (i,j,k). $[J]_n^{i,j,k}$ denotes the entry of $[J]$ at the n-th row and the (i,j,k)-th column corresponding to cell (i,j,k). $A'_{i+\frac{1}{2},j,k}$ is the area fraction subtracted from $1-A$ for the two adjacent cells, namely (i,j,k) and its right neighbor.

Let n and m denote the numbers of optical-layer liquid cells and tall-cell liquid columns, respectively. The total number of pressure samples is $n+2m$. Thus, $\{p\} \in \mathbb{R}^{(n+2m)\times 1}$ is

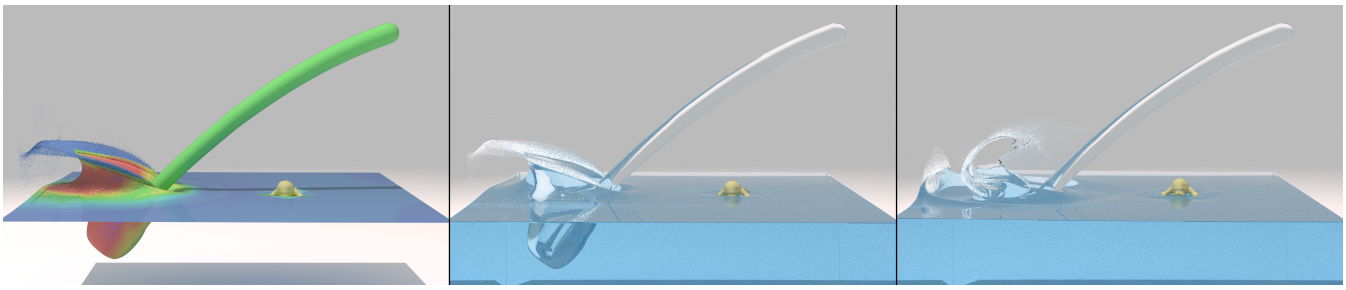


Figure 6: *Pouring Water: Water is poured onto the liquid surface from an elevated outlet. Left: our method with optical layer thickness visualized as a colormap. Center and right: our method with semi-transparent rendering.*

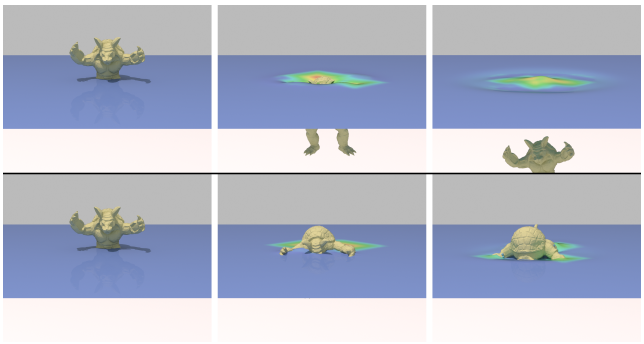


Figure 7: *Results for a case where an object intersects a tall cell. Top: in the conventional method [NOKA25], the object loses buoyancy when it intersects a tall cell. Bottom: with our method, buoyancy is computed correctly, allowing the object to float on the liquid surface.*

$\{p\} = [p_{o_1}, \ldots, p_{o_n}, p_{t_1,\text{top}}, p_{t_1,\text{bottom}}, \ldots, p_{t_m,\text{top}}, p_{t_m,\text{bottom}}]^T$, and $[J] \in \mathbb{R}^{6\times(n+2m)}$ is constructed analogously.

3.2.2. Two-Way Coupled Rigid Bodies with Tall Cells

A key difficulty arises in regions where the optical layer is thin, since rigid bodies may intersect tall cells. In the method of Narita et al. [NOKA25], it is assumed that the optical layer is sufficiently thick, and only cases where rigid bodies intersect the grid cells within the optical layer are considered. In contrast, we extend their method to handle situations where rigid bodies intersect tall cells directly (Figure 7). When rigid bodies intersect tall cells, the problem is more complex than in uniform grids. A tall cell spans multiple cells in the vertical direction, requiring careful tracking of the intersection location. Moreover, each tall cell contains two pressure points, so the correspondence between grid cells and pressure samples is no longer one-to-one.

We consider the case where an object intersects a tall cell (Figure 5). First, we introduce a virtual uniform grid underlying the tall cell. For the virtual uniform grid cell (i_0, j_0, k_0) intersecting the object, we compute $[J]^{i_0,j_0,k_0}$. We then define a parameter s $(0 \le s \le 1)$ that represents the relative position of (i_0, j_0, k_0) within the tall cell. Finally, we distribute the contribution by adding $(1-s)\,[J]^{i_0,j_0,k_0}$ to the column corresponding to c_{top} and $s\,[J]^{i_0,j_0,k_0}$ to the column corresponding to c_{bottom}. The parameter s enables us to capture the exact intersection location within the tall cell. This weighted distribution also naturally reflects the fact that tall cells contain two pressure samples, and thus the mapping between grid cells and pressure unknowns is not strictly one-to-one.

Algorithm 1 Our Simulation Loop

1: Save current grid and variables
2: Constructing the adaptive optical layer {§3.1}
3: Advect velocity and level set
4: Advance rigid body
5: Convert cells into tall cells
6: Solve pressure {§3.2}
7: Map pressure onto cubical cells
8: Update velocity and rigid body

3.3. Simulation Loop

Algorithm 1 outlines the main simulation loop used in our method. Except for the construction of the optical layer, the loop is almost identical to that of the conventional tall cell method [NOKA25]. When direct intersections between objects and tall cells are considered, the projection step requires slight modification. The modified parts are highlighted in red.

4. Results

We evaluate our method on four test cases and compare it with the conventional tall cell grid using an optical layer of uniform thickness. As in Narita et al. [NOKA25], we employ a variational pressure solver and perform the computations with the ICCG method. In all examples, the relative residual is set to 10^{-4}. Following Koike et al. [KMA20], we set the time step using a CFL number of 2. In addition, we adopt the extended narrow-band FLIP (EXNBFLIP) method to improve the fidelity of liquid representation.

The distribution of optical layer thickness in our method is visualized as a colormap on the liquid surface. Regions in blue correspond to thin optical layers, while colors closer to red represent thicker layers. The colormap is defined so that a thickness of 0 maps to blue and a thickness of $N_y/8$ maps to red. Thus, if the entire liquid surface appears red, the result coincides with the conventional

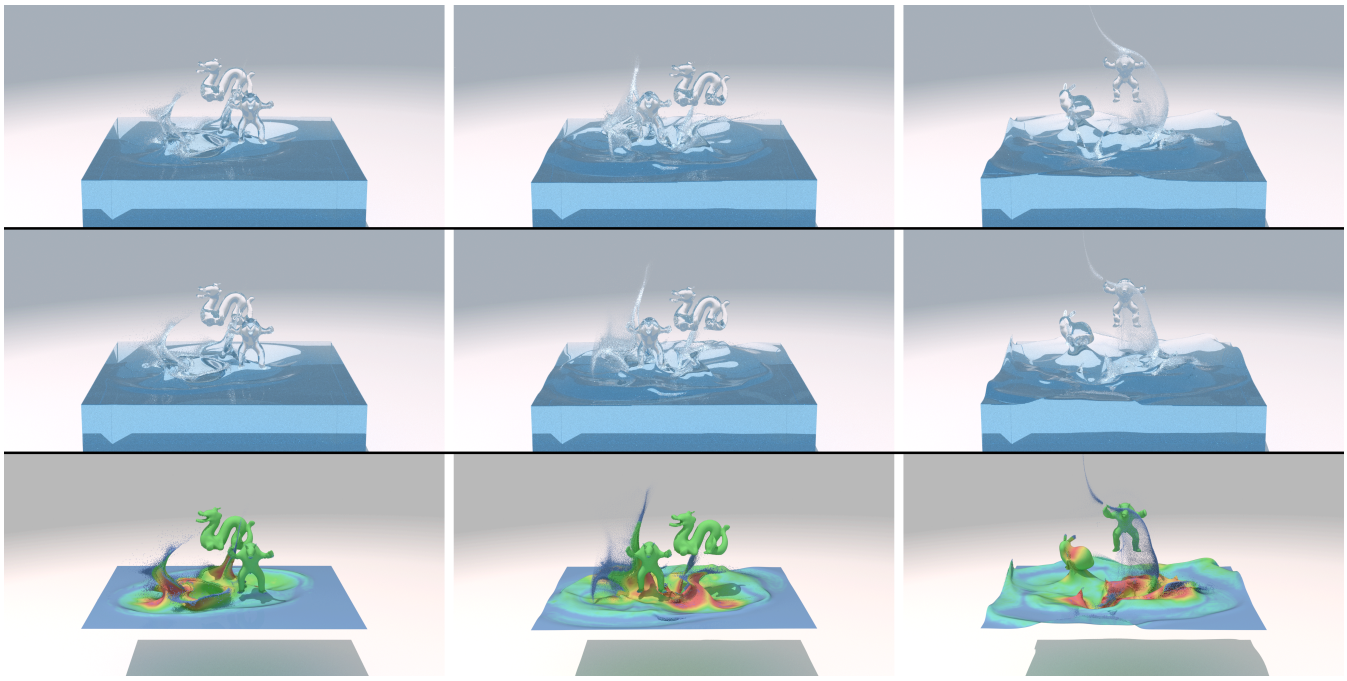


Figure 8: *Animals Drop: Liquids shaped like various characters are continuously dropped onto the water surface. Top: conventional tall cell method. Middle: our method. Bottom: our method with optical layer thickness visualized as a colormap.*

tall cell method using a uniform optical layer thickness. Note that the colormap shows the optical layer thickness before smoothing.

Table 1 reports the computation times of the major operations per time step. Our method accelerates the pressure solve by about 2.5× to 3× while maintaining visual quality, leading to an overall speedup of approximately 1.5× to 2× for the entire simulation. Table 2 provides further details of the projection step, showing that our method substantially reduces the number of grid cells by adaptively varying the optical layer thickness.

Figure 11 shows the temporal evolution of the cell counts for each scene, comparing the proposed and conventional methods. The cell count is largely dominated by the optical layer, and the proposed method effectively reduces the number of optical layer cells. Figure 12 shows the temporal evolution of the kinetic energy for each scene. Despite reducing the number of optical layer cells, the proposed method exhibits a similar kinetic energy evolution to the conventional method.

4.1. Animals Drop

Figure 8 presents a scenario in which liquids shaped like various characters are continuously dropped onto a water surface. The bottom row visualizes the distribution of optical layer thickness in our method. We observe that thicker optical layers are assigned in regions where the liquid impacts the surface or exhibits large motion. Even in cases with complex surface geometry and splashing, our method produces results nearly identical to those of the conventional approach. This simulation was performed on a Ryzen 9 5950X processor.

4.2. Breaking Dam

Figure 9 shows a dam-break simulation with several static pillars. As illustrated in the bottom row, regions near sharp edges of the initial shape and areas with significant surface motion are assigned thicker optical layers. Our method successfully reproduces results comparable to the conventional approach even when the liquid undergoes large-scale motion. This simulation was performed on a Ryzen 9 9950X processor.

4.3. Pouring Water

Figure 6 shows a case where water is discharged from an elevated outlet. Upon impact with the liquid surface, the discharged water generates large splashes. An armadillo model floats on the liquid surface through two-way coupling. Here, the densities are set to $\rho_{\text{water}} = 1.0\,[\text{kg}\,\text{m}^{-3}]$ for water and $\rho_{\text{object}} = 0.3\,[\text{kg}\,\text{m}^{-3}]$ for the object. These results demonstrate that our method can treat liquid regions above the surface not only in simple ballistic motion but also in more complex scenarios. Moreover, the simulation remains stable even when the surface experiences relatively strong impacts. This simulation was performed on a Ryzen 9 7950X processor.

4.4. Lucy in the Rain

Figure 1 presents a complex scene combining several elements. Water is poured onto the surface from multiple sources, while many small spherical droplets, mimicking rainfall, fall into the liquid. As shown on the right, thicker optical layers are assigned in regions where the liquid is poured or where droplets strike the surface. Our method remains effective even when the liquid surface

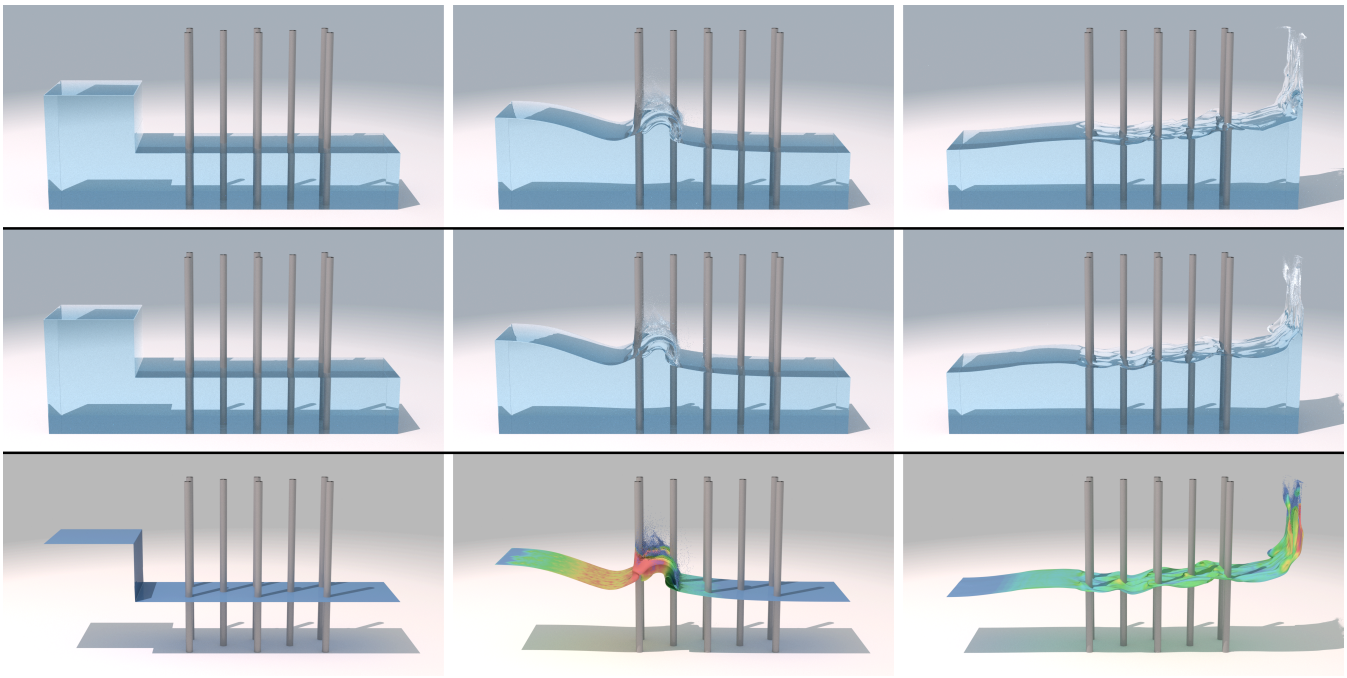


Figure 9: *Breaking Dam: Dam-break simulation with several static pillars. Top: conventional tall cell method. Middle: our method. Bottom: our method with optical layer thickness visualized as a colormap.*

exhibits small-scale ripples. Such scenarios are generally difficult to reproduce with adaptive grids that refine solely based on surface information [AB20]. This simulation was performed on a Ryzen 9 7950X processor.

5. Discussion and Limitations

Numerical Damping. Our method produces results of comparable quality to those of the conventional tall cell approach in many scenarios. However, certain settings reveal noticeable damping of kinetic energy, a phenomenon commonly observed in previous spatially adaptive methods. For example, even in a simple water-drop scene (Figure 2), the kinetic energy gradually decreases over time. A promising direction for future work is to develop improved strategies for constructing the adaptive optical layer based on liquid motion. It may also be beneficial to adaptively reinject energy to compensate for this damping effect [LAF11].

Performance. Although our method significantly accelerates the projection step compared to the conventional tall cell approach, it does not substantially reduce the overall simulation time. This is because, unlike adaptive grid methods [NOKA25], our approach has little effect on the computational cost of other operations, such as advection, outside the projection step. Since our method does not conflict with adaptive grids, it can be naturally combined with such frameworks. We therefore leave the design of a hybrid approach that integrates our method with adaptive grids as a direction for future work.

Heuristic Parameters. As noted, the method involves several heuristic parameters; however, the proposed approach is highly insensitive to these values and remains stable across a wide range of settings. Importantly, the same parameter set was used for all scenes in the paper, and no scene-specific tuning was required. In our existing experiments, we observed that varying these parameters within reasonable ranges does not materially affect the simulation behavior, further confirming the robustness of the method. In practical use, the main adjustable parameter is α: a larger α makes the result closer to that of the full tall-cell computation, while a smaller alpha prioritizes faster performance.

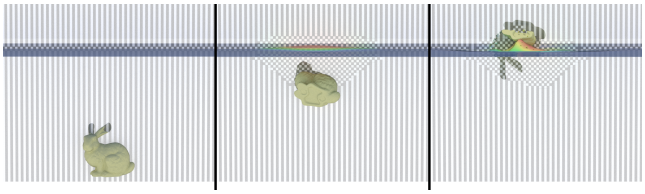


Figure 10: *A test case in which an object is fully inside the tall-cell region.*

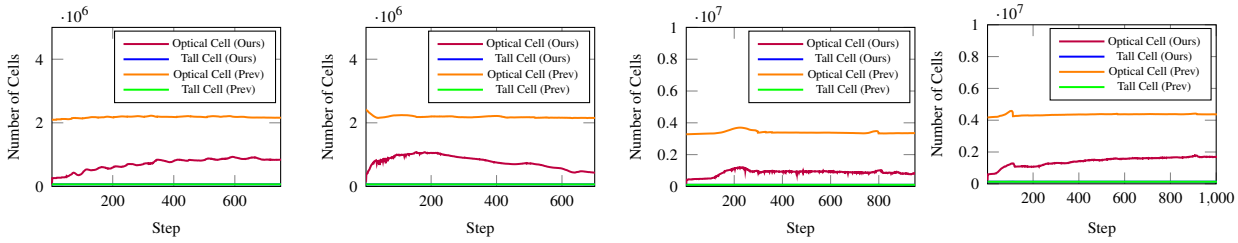
Two-Way Coupled Rigid Bodies. We acknowledge that tall-cell methods in general are not designed to accurately capture fully immersed rigid bodies whose dominant interactions occur far below the liquid surface. Therefore, we will explicitly state this as a scope limitation, while clarifying that the method remains robust and does not exhibit numerical failure. In our experiments, such situations only lead to slightly reduced accuracy in buoyancy or rising speed, but the motion remains stable and visually plausible (see Figure 10).

Table 1: *Timing breakdown of the four test cases. Numbers indicate the average time per step (in seconds) for each computational stage.*

Scene	Resol.	Method	Optical layer construction	Advection	Tall cells construction	Projection	Extrapolation	EXNBFLIP operation	Total
Fig. 8	256^3	Tall	-	2.123	1.049	15.576	0.623	2.48	22.439
		Ours	1.077	2.004	1.246	5.112	0.542	2.166	12.656
Fig. 9	$512 \times 256 \times 128$	Tall	-	1.396	0.583	12.672	0.373	1.261	16.652
		Ours	0.548	1.441	0.775	4.402	0.390	1.300	9.241
Fig. 6	$320 \times 256 \times 320$	Tall	-	2.738	1.179	20.985	0.757	1.825	30.082
		Ours	1.140	2.745	1.402	7.725	0.699	1.835	18.111
Fig. 1	512×256^2	Tall	-	3.464	1.519	24.309	1.003	3.070	34.545
		Ours	1.622	3.592	1.937	9.248	1.067	3.263	21.988

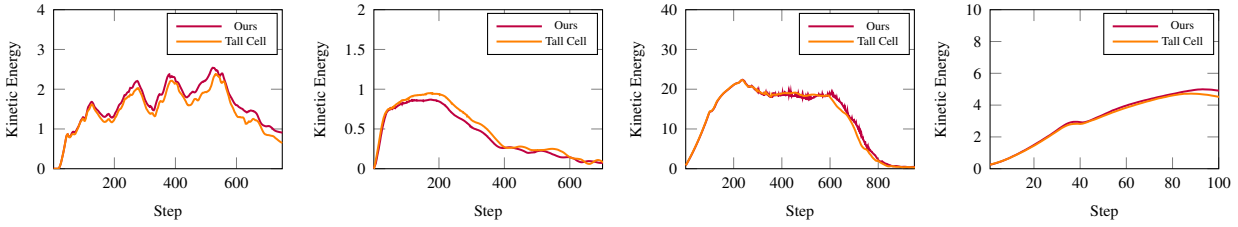
Table 2: *Breakdown of projection steps. All numbers are reported as averages per step. "Assemble matrix" includes all processes involved in matrix assembly, such as gradient evaluation, divergence computation, and multiplications (including operations for two-way coupling with rigid bodies, if applicable). "Residual" denotes the absolute residuals in PCG, measured using the $\|L\|_\infty$ norm.*

Scene	Resol.	Method	Total cells	Assemble matrix (seconds)	Solve Poisson's equation (seconds)	Residual	Iterations
Fig. 8	256^3	Tall	2306K	2.232	12.047	1.3×10^{-5}	264
		Ours	841K	0.594	3.392	1.5×10^{-5}	225
Fig. 9	$512 \times 256 \times 128$	Tall	2306K	0.988	10.942	1.0×10^{-5}	343
		Ours	884K	0.370	3.340	1.1×10^{-5}	227
Fig. 6	$320 \times 256 \times 320$	Tall	3590K	2.899	14.351	2.7×10^{-4}	200
		Ours	1057K	0.668	5.130	2.7×10^{-4}	199
Fig. 1	512×256^2	Tall	5739K	2.598	19.803	1.0×10^{-4}	232
		Ours	1941K	0.983	6.471	1.0×10^{-4}	213



(a) *Animals drop (Fig. 8).* **(b)** *Breaking dam (Fig. 9).* **(c)** *Pouring water (Fig. 6).* **(d)** *Lucy in the rain (Fig. 1).*

Figure 11: *Temporal evolution of cell counts for each scene, comparing the proposed and conventional methods. The blue and green curves overlap, as the number of tall cells is almost identical in both methods.*



(a) *Animals drop (Fig. 8).* **(b)** *Breaking dam (Fig. 9).* **(c)** *Pouring water (Fig. 6).* **(d)** *Lucy in the rain (Fig. 1).*

Figure 12: *Temporal evolution of the kinetic energy for each scene, comparing the proposed and conventional methods.*

6. Conclusions

We have presented a method that dynamically adjusts the optical layer thickness at each location on the liquid surface according to the liquid motion. This approach accelerates the projection step while preserving visual quality and reduces the overall simulation time. The method is simple to implement and can be readily integrated into existing tall cell grid frameworks. It also fully exploits the advantages of the tall cell grid structure. We expect this work to stimulate further research and development on tall cell grid methods.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments and constructive suggestions. This work is supported in part by JSPS KAKENHI (25K15401).

References

[AB20] ANDO, RYOICHI and BATTY, CHRISTOPHER. "A Practical Octree Liquid Simulator with Adaptive Surface Resolution". *ACM Trans. Graph.* 39.4 (July 2020). DOI: 10.1145/3386569.3392460 2, 3, 8.

[AGL*17] AANJANEYA, MRIDUL, GAO, MING, LIU, HAIXIANG, et al. "Power Diagrams and Sparse Paged Grids for High Resolution Adaptive Liquids". *ACM Trans. Graph.* 36.4 (July 2017). DOI: 10.1145/3072959.3073625 2.

[ATW13] ANDO, RYOICHI, THÜREY, NILS, and WOJTAN, CHRIS. "Highly Adaptive Liquid Simulations on Tetrahedral Meshes". *ACM Trans. Graph.* 32.4 (July 2013). DOI: 10.1145/2461912.2461982 2, 3.

[BBB07] BATTY, CHRISTOPHER, BERTAILS, FLORENCE, and BRIDSON, ROBERT. "A Fast Variational Framework for Accurate Solid-Fluid Coupling". *ACM Trans. Graph.* 26.3 (July 2007), 100–es. DOI: 10.1145/1276377.1276502 3, 5.

[Bri15] BRIDSON, ROBERT. *Fluid Simulation for Computer Graphics*. 2nd ed. A. K. Peters/CRC Press, 2015 2.

[CFL*07] CHENTANEZ, NUTTAPONG, FELDMAN, BRYAN E., LABELLE, FRANÇOIS, et al. "Liquid Simulation on Lattice-Based Tetrahedral Meshes". *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Goslar, DEU: Eurographics Association, 2007, 219–228 2.

[CM11] CHENTANEZ, NUTTAPONG and MÜLLER, MATTHIAS. "Real-Time Eulerian Water Simulation Using a Restricted Tall Cell Grid". *ACM Trans. Graph.* 30.4 (July 2011). DOI: 10.1145/2010324.1964977 2.

[CMK15] CHENTANEZ, NUTTAPONG, MÜLLER, MATTHIAS, and KIM, TAE-YONG. "Coupling 3D eulerian, heightfield and particle methods for interactive simulation of large scale liquid phenomena". *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '14. Copenhagen, Denmark: Eurographics Association, 2015, 1–10 2.

[ENGF03] ENRIGHT, DOUG, NGUYEN, DUC, GIBOU, FREDERIC, and FEDKIW, RON. "Using the Particle Level Set Method and a Second Order Accurate Pressure Boundary Condition for Free Surface Flows". *Proc. Fluids Engineering Division Summer Meeting*. ASME, 2003, 337–342. DOI: 10.1115/FEDSM2003-45144 2.

[FAW*16] FERSTL, FLORIAN, ANDO, RYOICHI, WOJTAN, CHRIS, et al. "Narrow Band FLIP for Liquid Simulations". *Computer Graphics Forum* 35.2 (2016), 225–232. DOI: 10.1111/cgf.12825 2.

[FF01] FOSTER, NICK and FEDKIW, RONALD. "Practical Animation of Liquids". *Proc. SIGGRAPH '01*. New York, NY, USA: ACM, 2001, 23–30. DOI: 10.1145/383259.383261 2.

[FM96] FOSTER, NICK and METAXAS, DIMITRI. "Realistic Animation of Liquids". *Proc. Graphics Interface '96*. CAN: Canadian Information Processing Society, 1996, 204–212 2.

[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 Trans. Graph.* 25.3 (July 2006), 805–811. DOI: 10.1145/1141911.1141959 2, 4.

[JSS*15] JIANG, CHENFANFU, SCHROEDER, CRAIG, SELLE, ANDREW, et al. "The Affine Particle-in-Cell Method". *ACM Trans. Graph.* 34.4 (July 2015). DOI: 10.1145/2766996 2.

[KFCO06] KLINGNER, BRYAN M., FELDMAN, BRYAN E., CHENTANEZ, NUTTAPONG, and O'BRIEN, JAMES F. "Fluid Animation with Dynamic Meshes". *ACM Trans. Graph.* 25.3 (July 2006), 820–825. DOI: 10.1145/1141911.1141961 2.

[KMA20] KOIKE, TATSUYA, MORISHIMA, SHIGEO, and ANDO, RYOICHI. "Asynchronous Eulerian Liquid Simulation". *Computer Graphics Forum* 39.2 (2020), 1–8. DOI: https://doi.org/10.1111/cgf.13907 6.

[LAF11] LENTINE, MICHAEL, AANJANEYA, MRIDUL, and FEDKIW, RONALD. "Mass and Momentum Conservation for Fluid Simulation". *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. New York, NY, USA: ACM, 2011, 91–100. DOI: 10.1145/2019406.2019419 8.

[LFO06] LOSASSO, FRANK, FEDKIW, RONALD, and OSHER, STANLEY. "Spatially Adaptive Techniques for Level Set Methods and Incompressible Flow". *Computers & Fluids* 35.10 (2006), 995–1010. DOI: https://doi.org/10.1016/j.compfluid.2005.01.006 2.

[LGF04] LOSASSO, FRANK, GIBOU, FRÉDÉRIC, and FEDKIW, RON. "Simulating Water and Smoke with an Octree Data Structure". *ACM Trans. Graph.* 23.3 (Aug. 2004), 457–462. DOI: 10.1145/1015706.1015745 2.

[LZF10] LENTINE, MICHAEL, ZHENG, WEN, and FEDKIW, RONALD. "A novel algorithm for incompressible flow using only a coarse grid projection". *ACM Trans. Graph.* 29.4 (July 2010). DOI: 10.1145/1778765.1778851 2.

[MWN*17] MANTEAUX, P.-L., WOJTAN, C., NARAIN, R., et al. "Adaptive Physically Based Models in Computer Graphics". *Computer Graphics Forum* 36.6 (2017), 312–337. DOI: https://doi.org/10.1111/cgf.12941 2.

[NMG09] NG, YEN TING, MIN, CHOHONG, and GIBOU, FRÉDÉRIC. "An Efficient Fluid-Solid Coupling Algorithm for Single-Phase Flows". *J. Comput. Phys.* 228.23 (Dec. 2009), 8807–8829. DOI: 10.1016/j.jcp.2009.08.032 2.

[NOKA25] NARITA, FUMIYA, OCHIAI, NIMIKO, KANAI, TAKASHI, and ANDO, RYOICHI. "Quadtree Tall Cells for Eulerian Liquid Simulation". *SIGGRAPH Conference Papers '25*. SIGGRAPH '25. New York, NY, USA: ACM, Aug. 2025. DOI: 10.1145/3721238.3730652 2, 3, 5, 6, 8.

[SABS14] SETALURI, RAJSEKHAR, AANJANEYA, MRIDUL, BAUER, SEAN, and SIFAKIS, EFTYCHIOS. "SPGrid: A Sparse Paged Grid Structure Applied to Adaptive Smoke Simulation". *ACM Trans. Graph.* 33.6 (Nov. 2014). DOI: 10.1145/2661229.2661269 2.

[Sta99] STAM, JOS. "Stable Fluids". *Proc. SIGGRAPH '99*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, 121–128. DOI: 10.1145/311535.311548 2.

[SWT*18] SATO, TAKAHIRO, WOJTAN, CHRIS, THUEREY, NILS, et al. "Extended Narrow Band FLIP for Liquid Simulations". *Computer Graphics Forum* 37.2 (2018), 169–177. DOI: 10.1111/cgf.13351 2.

[XCW*20] XIAO, YUWEI, CHAN, SZEYU, WANG, SIQI, et al. "An adaptive staggered-tilted grid for incompressible flow simulation". *ACM Trans. Graph.* 39.6 (Nov. 2020). DOI: 10.1145/3414685.3417837 2.

[ZB05] ZHU, YONGNING and BRIDSON, ROBERT. "Animating Sand As a Fluid". *ACM Trans. Graph.* 24.3 (July 2005), 965–972. DOI: 10.1145/1073204.1073298 2.

[ZLC*13] ZHU, BO, LU, WENLONG, CONG, MATTHEW, et al. "A New Grid Structure for Domain Extension". *ACM Trans. Graph.* 32.4 (July 2013). DOI: 10.1145/2461912.2461999 2.

[ZY10] ZHU, BO and YANG, XUBO. "Animating Sand as a Surface Flow". *Eurographics 2010 - Short Papers*. Ed. by LENSCH, H. P. A. and SEIPEL, S. The Eurographics Association, 2010. DOI: 10.2312/egsh.20101035 2.