Advancing Large-Scale Water Simulation in Avatar: Fire and Ash

Sean Flynn
Weta FX
Salt Lake City, USA
sflynn@wetafx.co.nz

Alexey Stomakhin
Weta FX
Mililani, USA
st.alexey@gmail.com

Joel Wretborn
Weta FX
Stockholm, Sweden
joel@wbn.se

Noh-hoon Lee
Weta FX
Seoul, Republic of Korea
nlee@wetafx.co.nz

Job Guidos
Weta FX
Roseburg, USA
physixtential@gmail.com

Nicholas Illingworth
Weta FX
Wellington, New Zealand
nillingworth@wetafx.co.nz

James Robinson
Weta FX
Wellington, New Zealand
jrobinson@wetafx.co.nz

Ryan Bowden
Weta FX
Austin, USA
rbowden@wetafx.co.nz

Figure 1: Giant tulkun breaching and crashing into 400ft sea dragon ships presented some of the largest scale water challenges on *Avatar: Fire and Ash*. Showing the final shot (left) and the simulated particle visualization (right). ©2025 20th Century Studios / Lightstorm Entertainment. All Rights Reserved. ©2026 Wētā FX Ltd.

Abstract

In this talk we discuss the collection of improvements we have made to Pahi [Stomakhin et al. 2023] for the massive-scale water demands of *Avatar: Fire and Ash*.

ACM Reference Format:
Sean Flynn, Alexey Stomakhin, Joel Wretborn, Noh-hoon Lee, Job Guidos, Nicholas Illingworth, James Robinson, and Ryan Bowden. 2026. Advancing Large-Scale Water Simulation in Avatar: Fire and Ash. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks (SIGGRAPH Talks '26), July 19–23, 2026, Los Angeles, CA, USA*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3799818.3812061

1 Introduction

The visual effects requirements of *Avatar: Fire and Ash* pushed the limits of our water toolset beyond that of its predecessor *Avatar: The Way of Water*. A single 400ft sea dragon ship in the previous *Avatar* installment becomes a fleet in *Avatar: Fire and Ash*, led by an even larger 700ft factory ship. Multiple tulkun elders are roughly 4x the size of Payakan—our previous aquatic protagonist. The water in which these giants face off needed to be scaled accordingly (Figure 1), and in order to deliver the total of over 2000 water shots our tools required significant upgrades.

We focused our efforts on massive scalability with increased MPI support and more efficient simulation partitioning. Due to the larger amount of simulation data, our post-processing workflows, such as surfacing, had to be made fully distributed as well.

The setups were automated and streamlined to reduce artists' time spent on preparing and wrangling heavy data. In particular, we replaced cumbersom Fluxed Animated Boundary [Stomakhin and Selle 2017] primitives with lightweight *Simulation Windows*, which also simplified our upresing workflow.

Multiple quality improvements have been made to the core of the solver to reduce artistic iterations. To achieve increased fidelity in air-water interaction we moved to a fully coupled two-phase pressure projection, see Figure 2. We upgraded our water state machine with refined transition heuristics and behaviors of its individual (spray and mist) components. A novel unified model

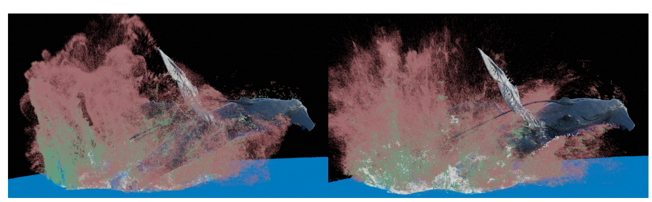

Figure 2: Strong multi-phase coupling between bulk water and air produces more realistic spiky features of massive splashes (right) as opposed to wispy and out-of-scale "hadouken" shapes from a weakly coupled solve (left). ©2026 Wētā FX Ltd.

SIGGRAPH Talks '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2541-8/26/07
https://doi.org/10.1145/3799818.3812061

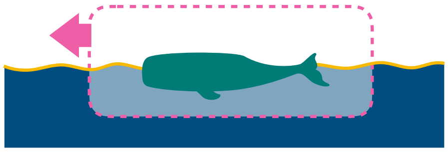

Figure 3: Simulation window setup. ©2026 Wētā FX Ltd.

[Wretborn et al. 2025] was adopted to avoid dealing with different types of bubble representations. Improved temporal integration schemes helped preserve the overall energy of the simulations.

With all of these techniques and also new debugging tools, we were able to deliver the unprecedented scale and complexity of the water shots featured in *Avatar: Fire and Ash*.

2 MPI simulation and processing

To handle the unprecedented scale, we overhauled our MPI-based workflows to ensure both performance and artistic flexibility. While existing space-filling curve partitioners could distribute memory adequately, they often produced partitions with little spatial locality.

We first attempted to address this issue by adding an axis-aligned heuristic to the partitioner (e.g., allowing artists to force partitioning along a specific direction). However, this approach generalized poorly to complex fluid configurations with no clear major axis that could be specified upfront. To eliminate the manual overhead and make the simulation tool more accessible, we developed the Power partitioner [Wretborn et al. 2026], a novel partitioning strategy based on optimal transport. This approach yielded spatially compact, well-conditioned partitions automatically, allowing artists to focus on creative iteration rather than technical wrangling.

The compact partitions generated by our strategy were instrumental in optimizing the performance of our distributed linear solvers, as they led to reduction in size of the halo regions (ghost cells) that must be exchanged via MPI. This reduction in communication latency was particularly vital for the AMG V-cycle, where the coarse-grid operators often suffer from increased communication-to-computation ratios. This optimization ensured that the solver maintained high parallel efficiency and strong scaling characteristics, even as the simulations approached billions of cells.

For post-processing, artists interacted with MPI data through visualizations of the underlying bucket structure (see definition in [Wretborn et al. 2026, §2]), filtering and refining small portions before committing the entire cache to the render wall. We partitioned massive simulation caches into manageable chunks of data (typically using more ranks than during simulation), each surfaced independently into render-ready SDFs. The final render composited these surfaces at runtime.

These strategies allowed us to scale up simulations as needed while maintaining performance sufficient for (mostly) overnight deliveries. However, one persistent challenge remains: the amount of data generated. The sheer volume of simulation data is not only demanding from a infrastructure perspective, but also poses a significant challenge for rendering, since our renderer is not distributed.

3 Simulation windows

The key to creating believable open ocean simulations that blend seamlessly with the surrounding procedural water is precise handling of boundary conditions together with sourcing and sinking (see FAB method of [Stomakhin and Selle 2017]). Building a FAB, however, requires the construction of a complex wave-deformed cradle-like shape around the area of action with precisely aligned kill volume [Stomakhin and Selle 2017, Figure 4].

To streamline the setup process we replaced FABs with a new *Simulation Window* primitive, see Figure 3. While the back-end capabilities remain the same, the users are simply required to provide the shape of the window to simulate inside of, alongside procedural water surface and velocity for emission. This approach also allows for straightforward up-resing by replacing procedural water with the result of a previous simulation.

4 State machine improvements

High-pressure transitions. Most of the bulk-to-spray transitions, resulting from the heuristics in Pahi [Stomakhin et al. 2023, §4.2.2], are due to the low-pressure heuristic [Stomakhin et al. 2023, Eq. 7], and the other heuristics are mainly used to control where these transitions do and do not occur. However, low-pressure transitions, i.e., particles that split due to being nearly in free fall, do not capture the more violent splashes that quickly vaporize large amounts of bulk water into fine spray and mist we observe in reference. Therefore, we introduced a *high-pressure* transition heuristic, which is used in tandem with the low-pressure heuristic to improve effects like a large tulkun smashing into the water surface.

We incorporate our high-pressure heuristic into the state machine by modifying Equation 7 in [Stomakhin et al. 2023, §4.2.2]

$$n < T_s^n \vee \|\nabla p\| < T_s^p \vee \|\nabla p\| > T_s^h, \tag{1}$$

where the new component compares the pressure gradient magnitude $\|\nabla p\|$ against high-pressure threshold T_s^h. We use a default value of 10^5 Pa/m, which we found worked well in identifying high-impact events causing accelerations greater than 10x gravity. We also filtered these types of transitions to only occur near colliders.

Spray and mist collider interaction. We found that the behavior of spray and mist particles interacting with colliders could lead to undesireable artifacts due to these reduced models not treating colliders properly with, e.g., a pressure projection as in the bulk water. This could manifest in many ways including particles clipping through colliders, pushing against them without spreading out, or falling off in distracting straight lines. We improved this interaction by adding reflection, restitution, and friction controls. The particles velocities can be further modified by the users through noise fields.

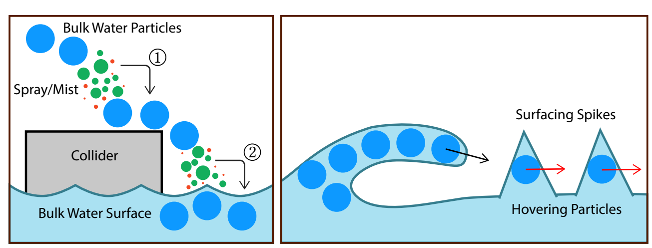

Figure 4: Back-to-bulk transitions (left) occur when hitting colliders ① and entering the water surface ②. The water spikes issue (right) is caused by shallow splashes with particles that hover along the surface (shown with red trajectories). ©2026 Wētā FX Ltd.

Back-to-bulk transitions. When the above method for improving spray and mist collider interaction was not sufficient, we also introduced a new transition path within the state machine for spray and mist particles to go back to bulk near colliders, labeled ① in Figure 4. Previously, the only way for spray and mist particles to become bulk water particles in Pahi was for them to re-enter the bulk water surface, labeled ② in Figure 4. To account for the difference in the size and quantity of spray and mist particles becoming bulk particles, we applied a new stochastic deletion method in both ① and ② to conserve volume and reduce computation.

A drawback to using hybrid Eulerian-Lagrangian fluid simulation methods like we use in our bulk water state is that artefacts can arise due to the discrepancy between a particle and voxel sizes. In particular, we ran into an issue shown in Figure 4 with particles coming off of shallow splashes that would hover and slide across the water surface without fully re-entering because the bulge in the surface they created would be insufficient for the pressure solver to resolve in a timely manner. Our anisotropic surfacer would further exacerbate the issue, as these would often show up as distracting spikes surrounding the splashes. To avoid these issues we would identify these particles and then transition them to the spray state which allowed them to properly fall back into the bulk water.

Spray air drag masking. In Pahi, spray and mist are weakly coupled with the surrounding air through a drag force that acts on particles based on their radius to approximate the real-world effect of smaller droplets falling slower than larger droplets. This works fine for mist where the particles do not bunch up into larger droplets. However, for spray, the SPH-driven droplets are collections of many interacting particles. Drag will act equally on spray particles regardless of whether they actually touch the air. To improve this, we applied a mask on the spray particles that would weight particles on the interior of droplets to be dragged less than particles near the surface. This allowed us to achieve more realistic variation in our spray, with small droplets shedding off of large droplets, as we observe in reference.

Transition history tracking. To help users set the transition heuristic thresholds described here and in [Stomakhin et al. 2023], we maintain a bitmask channel carried across all states which records the cause of the most recent transition as well as a counter for past transitions of every type. This allows the user to diagnose the source of transitions caused by unique or challenging simulation configurations. It also allows us to rapidly tune the transition parameters as we continue to update the state machine.

5 Energy preservation

Pahi uses APIC [Jiang et al. 2015] and MLPIC (a.k.a. PolyPIC [Fu et al. 2017]) for bulk water simulation to better preserve energy between the particles and grid representations of the water compared to PIC/FLIP. The APIC/MLPIC coefficients are determined by the velocities stored within small neighborhoods of voxels. Issues arise when the cofficients are sampled from velocities that are not guaranteed to be divergence free, like in the regions outside the water surface and inside colliders, which are usually extrapolated from velocities inside the fluid. This is most problematic in particles in regions that should be calm, sitting along a collider or along the surface of resting water. In these regions particles will dance along colliders, shoot up in random directions causing unnatural water spikes, and never fully come to rest as they should. Setting the coefficients to zero (effectively downgrading to PIC) along the surface and colliders does remove the problems, but it also overly damps the motion and can make the water look too viscous. We alleviated these issues by providing a damping control where artists can reduce the magnitude of APIC/MLPIC coefficients only in the direction normal to water surfaces and colliders. This provided a balance between reducing the artefacts and preserving energy.

To further improve the richness of vortex dynamics we implemented SBDF2 time integrator from [Nielsen et al. 2022] and covector fluids solve from [Nabizadeh et al. 2022].

6 Unified bubbles

One key challenge artists identified was the difficulty of authoring both diffuse and hero bubbles within the same scene (see [Stomakhin et al. 2020] for terminology). Traditionally these two regimes are handled with separate solvers, resulting in inconsistent behavior across scales and/or visual artifacts when these two regimes interact. We leveraged the model presented in [Wretborn et al. 2025], which was developed to address this limitation through a multiscale formulation capable of representing both dispersed and stratified bubble flows using a single discretization. Their improved time integrator further improved the stability of drag-based coupling, and we used it for all of our guided secondary fluid simulations (otherwise following the framework in [Wretborn et al. 2022]).

References

Chuyuan Fu, Qi Guo, Theodore Gast, Chenfanfu Jiang, and Joseph Teran. 2017. A polynomial particle-in-cell method. *ACM Trans. Graph.* 36, 6 (Nov. 2017), 1–12.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Trans. Graph.* 34, 4 (July 2015), 1–10.

Mohammad Sina Nabizadeh, Stephanie Wang, Ravi Ramamoorthi, and Albert Chern. 2022. Covector fluids. *ACM Trans. Graph.* 41, 4 (July 2022), 1–16.

Michael B. Nielsen, Morten Bojsen-Hansen, Konstantinos Stamatelos, and Robert Bridson. 2022. Physics-Based Combustion Simulation. *ACM Trans. Graph.* 41, 5, Article 176 (May 2022), 21 pages. doi:10.1145/3526213

Alexey Stomakhin, Steve Lesser, Joel Wretborn, Sean Flynn, Johnathan Nixon, Nicholas Illingworth, Adrien Rollet, Kevin Blom, and Douglas Mchale. 2023. Pahi: A Unified Water Pipeline and Toolset. In *Proceedings of the Digital Production Symposium* (Los Angeles, CA, USA) *(DigiPro '23)*. Association for Computing Machinery, New York, NY, USA, Article 11, 13 pages. doi:10.1145/3603521.3604291

Alexey Stomakhin and Andrew Selle. 2017. Fluxed animated boundary method. *ACM Trans. Graph.* 36, 4 (July 2017), 1–8.

Alexey Stomakhin, Joel Wretborn, Kevin Blom, and Gilles Daviet. 2020. Underwater bubbles and coupling. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Talks.* ACM, New York, NY, USA.

Joel Wretborn, Sean Flynn, and Alexey Stomakhin. 2022. Guided Bubbles and Wet Foam for Realistic Whitewater Simulation. *ACM Trans. Graph.* 41, 4, Article 117 (jul 2022), 16 pages. doi:10.1145/3528223.3530059

Joel Wretborn, Marcus Schoo, Noh-Hoon Lee, Christopher Batty, and Alexey Stomakhin. 2026. A practical partitioner for distributed simulations on sparse dynamic domains using optimal transport. *ACM Trans. Graph.* 3787521 (Jan. 2026).

Joel Wretborn, Alexey Stomakhin, and Christopher Batty. 2025. A unified multi-scale method for simulating immersed bubbles. *Comput. Graph. Forum* (April 2025).