Volume-Preserving LBM-MPM Coupling for Air-Water-Sand Mixtures

XIAOYU XIAO*, Shanghai Jiao Tong University, China
HAOXIANG WANG*, Tsinghua University, China
XIAOKANG YANG, Shanghai Jiao Tong University, China
MATHIEU DESBRUN, Inria – Ecole Polytechnique, France
WEI LI†, Shanghai Jiao Tong University, China

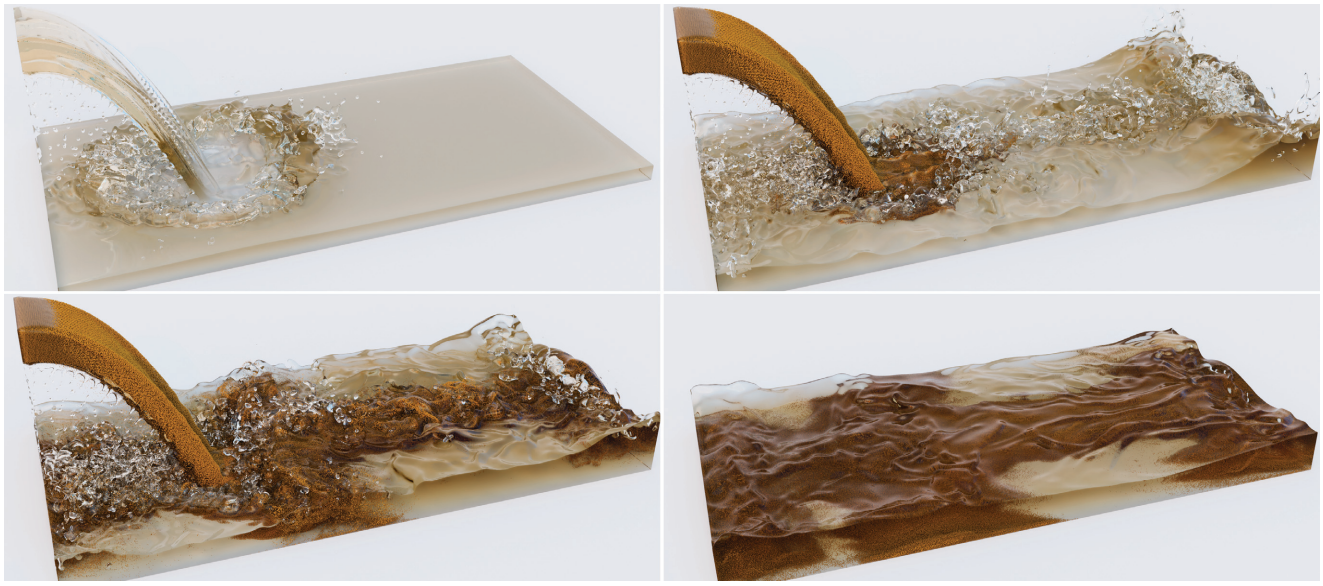

Fig. 1. **Mixture jet.** Our novel simulation technique for fluid flows interacting with granular materials couples a Lattice Boltzmann Method (LBM) two-phase solver with a Material Point Method (MPM) solver for sand to produce visually rich air-water-sand mixing phenomena. Here, a water jet entraining sediments gushes from a rectangular outlet as a yellow-tinted mixture, producing air entrainment, splashing, sand displacement, and abundant bubbles as the ratio of sand to water in the jet increases. When the jet flow ceases, sand has settled in heaps predominantly near the impact area of the jet as expected.

Simulating the dynamic, multiscale interactions between granular materials and multiphase fluids remains a significant computational challenge in computer graphics, as the visual complexity of such mixtures arises from strongly coupled small-scale structures. We present a novel, physically-based simulation framework for sand-water-air mixtures that couples a Lattice Boltzmann Method (LBM) for weakly-compressible two-phase fluids with a Material Point Method (MPM) for granular sand. Our approach is built upon a unified continuum formulation that expresses the governing equations for both fluid phases (air and water) and the granular medium within a consistent framework. To accurately capture the transition of sand from a dry, friction-dominated state to a soaked, sticky medium, we introduce a water retention model that describes how liquid infiltrates and is retained within the granular structure. Furthermore, we enforce volume conservation of the fluids within the mixture, ensuring numerical stability and physical realism. Our robust coupling mechanism enables the simulation of complex phenomena such as sand mobilization, transport, settling, and erosion across a wide range of density ratios. We demonstrate the efficiency of our method through several challenging scenarios, including the breaching of sand-walled basins, sediment-laden flows, and the erosive collapse of sand structures.

CCS Concepts: • **Computing methodologies** → **Physical simulation**.

Additional Key Words and Phrases: Two-phase flow simulation, water-air-sand mixtures, lattice Boltzmann method, phase-field modeling

ACM Reference Format:
Xiaoyu Xiao, Haoxiang Wang, Xiaokang Yang, Mathieu Desbrun, and Wei Li. 2026. Volume-Preserving LBM-MPM Coupling for Air-Water-Sand Mixtures. *ACM Trans. Graph.* 45, 4, Article 77 (July 2026), 17 pages. https://doi.org/10.1145/3811302

1 Introduction

The dynamic interaction between granular materials (such as sand) and fluids (such as water and air) typically gives rise to visually striking and detail-rich phenomena: when these two phases come into contact in natural environments, the resulting mixture displays

*Both authors contributed equally to this research.
†Corresponding author.

Authors' Contact Information: Xiaoyu Xiao (shawnshaoyu99@gmail.com), Xiaokang Yang (xkyang@sjtu.edu.cn), Wei Li (1104720604wei@gmail.com): John Hopcroft Center for Computer Science, School of Computer Science, Shanghai Jiao Tong University, Shanghai, PRC; H. Wang (whx22@mails.tsinghua.edu.cn): Department of Automation, Tsinghua University, Beijing, PRC; M. Desbrun (mathieu.desbrun@inria.fr): Inria Saclay / LIX (École polytechnique/CNRS), Institut Polytechnique de Paris, Palaiseau, France.

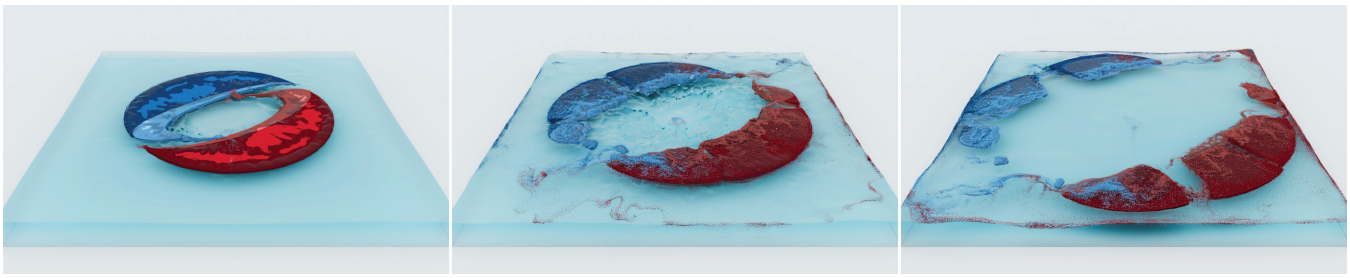

Fig. 2. **Wind-driven water-sand interaction.** A floating sand island (density $\varrho = 600\ \mathrm{kg\,m^{-3}}$) is subjected to a strong vertical airflow, kicking up spray and sand. Our coupled air-water–sand solver faithfully captures the complex coupling effects of this example, with progressive wetting producing clear wet/dry contrast and ultimately breaking the logo apart into discrete fragments that disperse across the container and drift on the water surface.

highly complex motion patterns characterized by small-scale structures and strongly coupled behaviors. The granular phase introduces additional vortical features into the surrounding fluid, while the interstitial fluid significantly alters the mechanical response of the sand, transitioning it from a dry, friction-dominated granular state to a soaked, sticky medium. These transformations can occur rapidly and nonlinearly, making the numerical simulation of sand–water mixtures a significant computational challenge: accurately simulating mixtures requires resolving not only the intrinsic nonlinearity of fluid motion but also the bidirectional coupling between the fluid and the deformable, porous granular material. This coupling spans multiple spatial and temporal scales: grains interact through contact forces, friction, and collisions, while simultaneously exchanging momentum and mass with the surrounding liquid as it fills the small interstices of the granular medium. Capturing these multiscale behaviors places significant demands on numerical methods, which must balance physical fidelity, stability, and computational efficiency. Simulating sand–fluid mixtures is not only important across geophysics, but is also key to producing complex and realistic natural phenomena with high visual impact.

Overview. In this work, we develop a physically-based simulation for two-phase fluid flows interacting with granular sand, which couples a Lattice Boltzmann Method for weakly compressible fluids with the Material Point Method for sand to properly capture coupled transport, phase evolution, and water uptake in sand-like media. Our approach is based on a unified formulation of mixtures that expresses the governing equations for both fluid phases and the granular medium within a consistent continuum framework. From this formulation, we derive a fluid equation coupled to a phase-field representation to evolve the two-phase flow, which we efficiently simulate with a LBM-based kinetic solver. The sand, instead, is modeled using a Material Point Method based formulation. We then express explicit fluid–sand coupling terms that govern momentum exchange, drag, and pressure interactions, enabling the simulation of behaviors such as sand mobilization, transport, settling, and suspension, across a wide range of density ratios. We further introduce a water absorption model for sand, describing how liquid infiltrates and is retained within the granular medium, along with a coupling strategy between phase field and sand that enforces fluid volume preservation to ensure that the air/liquid interface motion and the sand's water content remain consistent at the discrete level. These different contributions conspire to form a robust approach for complex phenomena of water-sand mixtures, as we will demonstrate in a number of simple and complex examples.

2 Background and Related Work

We begin with a review of relevant prior work, covering multiphase flow solvers, sand simulation techniques, and approaches for coupling granular materials with fluids to model their mixtures.

2.1 Multiphase flow solvers

Navier-Stokes based methods. Approaches to flow simulation based on Navier-Stokes (NS) equations can be roughly categorized into particle-based, grid-based, and hybrid methods. These techniques apply macroscopic discretizations of the flow field, but differ in whether they model only the liquid phase or explicitly capture both liquid and surrounding air. Given that we wish to handle mixtures where typically, air, liquid, and sand interact, we focus here on methods designed for multiphase flow simulation.

Multiphase flow solvers are typically based on particle- or grid-based formulations. Particle methods include Smoothed Particle Hydrodynamics (SPH) multiphase schemes [Solenthaler and Pajarola 2008; Ren et al. 2014; Yang et al. 2015; Yan et al. 2016; Yang et al. 2017], which can only handle relatively viscous liquids. Power particles [de Goes et al. 2015; Qu et al. 2022] were introduced to better preserve volume and momentum near interfaces and to improve surface-tension modeling, while their coupling with MPM enabled bubble and foam simulations [Qu et al. 2023] as well, but again only at low Reynolds numbers. The peridynamic framework of Yan and Ren [2023], while robustly handling high density ratio multi-fluids, suffers from the same restriction. The Moving-Least-Squares Reproducing-Kernel (MLSRK) approach of Chen et al. [2020] aims to unify fluids and solids, but requires prohibitively small timesteps in practice. Finally, Li et al. [2024a] recently proposed a Voronoi-based Lagrangian covector approach using path integrals to robustly simulate turbulent free-surface flows without fixed background grids.

Grid-based approaches offer complementary capabilities. Extensions of level-set [Kim 2010] and Volume-of-Fluid (VOF) strategies [Cho and Ko 2013; Langlois et al. 2016] explicitly account for density and pressure jumps, and most solvers [Losasso et al. 2006; Mihalef et al. 2006; Kim et al. 2007; Boyd and Bridson 2012] combine a variable-density projection [Kang et al. 2000] with the ghost-fluid method [Hong and Kim 2005]. A multi-VOF technique

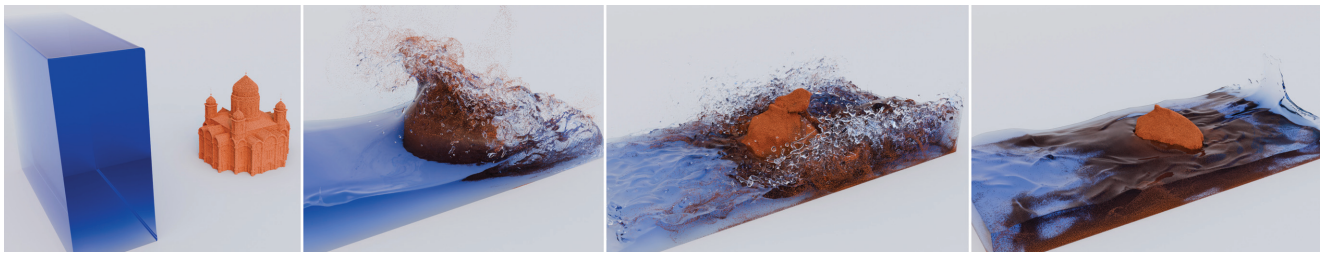

Fig. 3. **Dam break over a sand castle.** As the water reaches the castle, the incoming flow exerts strong shear and impact forces on the sand structure, causing grains to detach and be rapidly transported downstream. Continuous loading and infiltration weaken the structure, leading to turret failure and extensive wetting at first, before the central keep ultimately collapses.

tailored to bubble and foam dynamics [Karnakov et al. 2022] further improved interface handling but at high computational cost. Material-Point-Method (MPM) based models for viscoelastic fluids with phase change [Su et al. 2021; Tu et al. 2024] and Lagrangian mesh-based solvers [Misztal et al. 2013] were also proposed, but they remain limited to low Reynolds numbers. Goldade et al. [2020] presented a constraint-based reduced model that simplifies the internal computation of zero-density bubbles via Lagrange multipliers, achieving efficient and volume-conserving free-surface bubble simulations. Although recent Eulerian–Lagrangian approaches [Deng et al. 2022; Wang et al. 2024b] successfully capture bubble behavior, they do not demonstrate turbulent regimes either. Likewise, hybrid solvers coupling velocity, pressure, and interface position [Ando et al. 2015; Saye 2016, 2017; Sun et al. 2024] have so far shown limited effectiveness in simulating strongly turbulent free-surface scenarios. Recently, Braun et al. [2025] presented an adaptive phase-field–FLIP framework enabling robust and efficient large-scale two-phase fluid simulations, while Chen et al. [2025a] extended flow maps to compressible regimes using Lagrangian path integrals, preserving vortical structures in shallow water simulations or high-Mach shocks.

Kinetic solvers. Originally developed in Computational Fluid Dynamics, the lattice Boltzmann method has recently gained prominence in computer graphics due to its massively parallel nature and support for turbulent flow phenomena. Free-surface LBM (FSLBM) has seen rapid development in bubble simulation, starting from the early contributions of [Körner et al. 2005; Thürey et al. 2007] and evolving through a series of improvements [Anderl et al. 2014; Bogner 2017] that allow bubble breakup and coalescence. Building on this progress, [Wang et al. 2025] introduced an advanced FSLBM formulation that unifies the treatment of bubbles and foam, while greatly improving efficiency and memory use over existing kinetic multiphase solvers. Concurrently, diffuse-interface kinetic multiphase LBM methods [Guo et al. 2017; Li et al. 2021b, 2022; Li and Desbrun 2023; Li et al. 2024b; Ma et al. 2024; Ding and Liu 2025; Xiao et al. 2025] have emerged as a powerful alternative paradigm: by considering a fixed-profile phase field spreading interfacial forces across multiple grid cells, they offer substantially improved numerical stability compared to traditional sharp-interface techniques.

2.2 Granular medium simulation

Granular materials such as sand can be modeled by resolving individual grains with the Discrete Element Method (DEM), in which particles are typically represented as soft spheres interacting through penalty-based contact laws based on relative displacements [Cundall and Strack 1979; Kruggel-Emden et al. 2007]. In computer graphics, Bell et al. [2005] employed a Hertzian normal response combined with tangential Coulomb friction, capturing strong granular texture and high-frequency behaviors, at the expense of more involved contact modeling when scaling to large scenes.

To address scalability limitations, several continuum-based approaches have been proposed. Zhu and Bridson [2005], for instance, adapted the PIC/FLIP framework to sand by decomposing particle motion into rigid transport and incompressible shear governed by a yield surface. Building on this idea, Klár et al. [2016] incorporated a Drucker–Prager elastoplastic model into MPM, where yield-surface projection mitigates relative sliding but still falls short of reproducing full grain interlocking within the friction cone. Narain et al. [2010] presented a continuum-based model that enables efficient simulation of granular materials and two-way interaction with solid bodies. A hybrid strategy was later introduced by Yue et al. [2018] using DEM near boundaries and MPM in the interior, with a reconciliation zone to ensure coherent motion and enable interlocking near phase interfaces, thus capturing size-dependent clogging phenomena. Most recently, Chen et al. [2025b] proposed a numerical homogenization method for extracting macroscopic material parameters from grain-level simulations, so as to allow efficient large-scale modeling of granular media within an enhanced Material Point Method.

2.3 Sand–water coupling

Most relevant to our work are the approaches designed to couple granular media with fluids, which we review now based on the type of overall solver they leverage.

Navier-Stokes-based methods. NS–based approaches treat sand and water mixtures by coupling continuum models for each phase. Power Plastics [Qu et al. 2023] represent water as a weakly compressible fluid [Xie et al. 2023] and sand via a Herschel–Bulkley model [Yue et al. 2015], enforcing bidirectional coupling on a shared MPM grid. Because both phases evolve on a single grid with a common velocity field, the mixture responds to a superposition of phase-specific forces rather than a physically consistent mixture law. To address this limitation, Tampubolon et al. [2017] and Gao et al. [2018] propose to solve sand and water on separate grids, exchanging mass and momentum while enforcing mixture-level incompressibility. Recently, Böttcher et al. [2025] used SPH to simulate

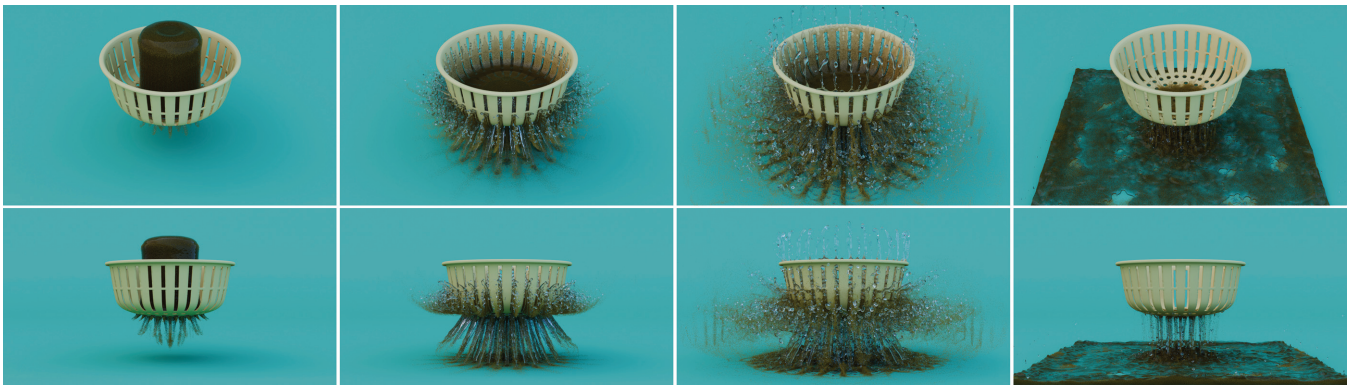

Fig. 4. **Water-sand flow through a colander.** Given the high water content, the sand stays fluidized rather than forming cohesive clumps, so the mixture passes through the colander and exits as multiple yellow-tinted jets of mixture. After impact, water spreads to fill the shallow pool while sand is carried by the water flow and gradually settles to the bed, forming scattered sediment deposits and sand heaps of various dimensions.

the coupling of porous flow with sand particles while Li et al. [2025b] employed a particle-based method to handle poro-elasto-capillary effects between soft porous materials and multiphase fluids.

Discrete Element Method. Sand can also be modeled using DEM and water with SPH: DEM grains absorb nearby SPH particles to track moisture content and experience hydrodynamic forces [Rungjiratananon et al. 2008]. Wang et al. [2021] extended this strategy with seepage, a saturation-dependent capillary law, as well as additional buoyancy, drag, and momentum-exchange terms. However, moisture estimation becomes unreliable inside dense SPH "spherical clumps", and the Rayleigh-limited timestep imposed by DEM (which is typically much smaller than the fluid's CFL limit) results in high computational costs. Yan et al. [2016] bypassed this issue with a unified SPH formulation using a Drucker–Prager model in place of DEM, later extended to support phase transitions [Yang et al. 2017; Jiang et al. 2020]. DEM explicitly resolves individual grains without introducing continuum conservation equations, enabling fine-scale and high-frequency granular dynamics. In CFD–DEM coupling, however, maintaining accurate volume-fraction fields remains important, especially when only a few fluid cells contain large particles. Particle-to-grid transfer is therefore performed through coarse graining or averaging [Xiao and Sun 2011; Zhu and Yu 2002]. DEM has been successfully coupled with conventional CFD solvers to simulate fluid–granular interactions [Wang et al. 2024a; Ranjbari et al. 2025]. Sun and Sakai [2015] presented a 3D framework that couples CFD-based gas–liquid interface capture with DEM particles, and uses signed distance functions and immersed-boundary walls for complex and moving boundaries. More recently, Li et al. [2021a] showed advanced DEM–CFD simulations of a cross-torus paddle mixer. However, DEM resolves particle–particle interactions via explicit contact laws, which makes it prohibitively costly for practical animation in collision-dominated dense flows. As reported in [Tang et al. 2025], the computational burden remains significant even when rotational effects are neglected to reduce costs.

Kinetic solvers. Kinetic-based CFD solvers have also been used to simulate sand–water mixtures. The solid phase is typically described by Lagrangian particles, which are coupled to an LBM fluid through momentum-exchange interactions [Rettinger and Rüde 2017, 2022]; LBM–DEM coupling has been also extensively applied to model fluid–solid interactions, commonly employing immersed-boundary formulations [Feng et al. 2007; Wang et al. 2019] or momentum-exchange techniques [Lominé et al. 2013]. Ding and Xu [2018] proposed a multiphase fluid–solid framework incorporating a free-surface model, while Kano et al. [2020] introduced a three-phase LBM approach for fluid–gas–sand systems. A color-gradient-based coupling has also been demonstrated [Jiang et al. 2022], though its applicability is constrained by the assumption of equal phase densities. Fei et al. [2023] developed a pseudopotential LBM–DEM scheme but reported significant spurious-velocity artifacts. More recently, Li et al. [2025a] presented a phase-field LBM–DEM method, limited to 2D simulations and associated with substantial computational expense. Beyond DEM, coupling LBM with MPM has also gained attention: Liu et al. [2020] introduced an LBM–MPM formulation capable of handling large solid deformation and later extended it to a phase-field LBM variant [Liu et al. 2023]. Nevertheless, these approaches do not model complex sand behaviors, and often assume a static solid phase without accounting for drag forces: they often assume grain sizes spanning multiple cells both in single-phase and two-phase simulations.

2.4 Discussion

This review of past and current methods reveals that there is still no two-phase-flow/sand coupling framework that can claim to deliver both high efficiency and high physical realism. Although recent work [Tang et al. 2025] resolves the longstanding volume-conservation issue of mixture simulation, it remains unable to capture key coupled dynamics such as bubble formation, droplet splashing, wetting capillary coupling, surface-tension dynamics, and their interactions with sand grains; moreover, the implicit density projection [Kugelstadt et al. 2019] imposes an extra computational burden. While a unified framework is appealing, we propose to couple LBM and MPM to maximize both efficiency and fidelity. Indeed, LBM offers exceptional GPU scalability and robustly handles complex fluid interfaces (e.g., wetting, bubble dynamics); conversely, MPM efficiently models the macroscopic elastoplasticity of sand. Absent a pure LBM solution for granular dynamics, developing a scalable LBM-MPM coupling is the most practical path forward.

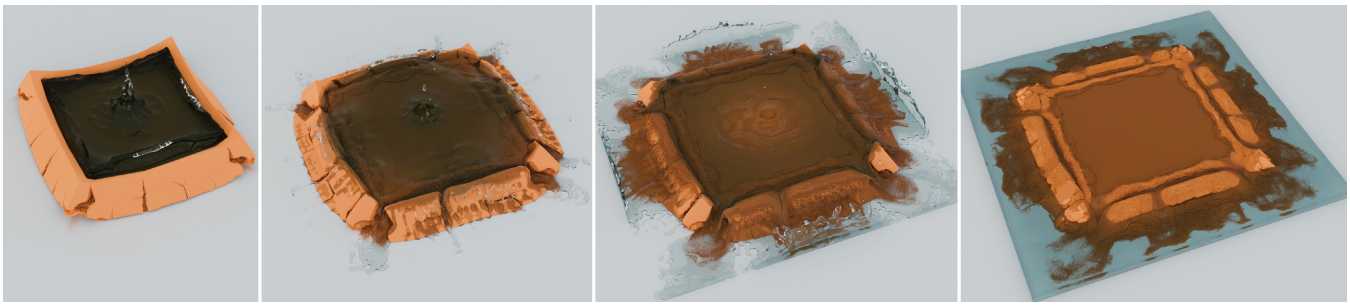

Fig. 5. **Breaching of a sand bailey.** When water is poured inside a bailey made of sand, the induced stress and sediment transport erode the walls, first causing multiple fractures, then ultimately a full collapse, with water eventually spilling out.

Against this backdrop, we further point out that the particle-discretized phase-field formulation of Braun et al. [2025] cannot be directly extended to liquid–sand mixture simulations, primarily because strict volume preservation in a mixture is difficult to guarantee. We propose instead the introduction of a source term in the phase-field evolution equation, which, within our LBM–MPM framework, will enforce volume conservation of the two-phase fluid in a flow-sand mixture while maintaining computational efficiency. We describe our approach to mixture simulation in detail next, before showing a series of comparisons to previous methods, simulations, and a real-life experiment to help evaluate its worth.

3 Modeling a Mixture

We now present the governing equations for both the fluid and sediment phases in our fluid–sand mixture model. The density and velocity of the fluid will be denoted as ρ and $\mathbf{u}$, while the equivalent notions for the sand will be denoted ϱ and $\mathbf{v}$, respectively.

3.1 Fluid formulation in mixture

In the simulation of air-water-sand mixtures, accurately modeling the interaction between the fluid and each individual sand grain poses a significant challenge: resolving these detailed and time-dependent boundary conditions for a large number of sediment particles would demand an impractically fine fluid discretization. To circumvent this computational cost, we adopt a *locally averaged Navier-Stokes framework* [Anderson and Jackson 1967] which treats every point in space as a *coexisting mixture* of fluid phases (liquid and air) and solid (sand). The governing equations for the fluid then reduce to a mixture-type formulation as found in [Manninen et al. 1996; Nielsen and Østerby 2013]. The *mass conservation law* for the fluid phase takes the form given in [Gao et al. 2018] as a function of the fluid fraction ϵ, fluid density ρ, and the fluid velocity $\mathbf{u}$:

$$\frac{\partial(\epsilon\rho)}{\partial t} + \nabla\cdot(\epsilon\rho\mathbf{u}) = 0, \tag{1}$$

When the mixture contains two immiscible fluid phases (air and liquid) and sand as we are focusing on in this paper, the *momentum equation* becomes [Manninen et al. 1996]:

$$\frac{\partial(\epsilon\rho\mathbf{u})}{\partial t} + \nabla\cdot(\epsilon\rho\mathbf{u}\otimes\mathbf{u}) = -\epsilon\nabla p + \nabla\cdot[\epsilon\mu\nabla^{\text{sym}}\mathbf{u}] + \epsilon\mathbf{F}_s + \epsilon\mathbf{F}_b + \mathbf{F}_d, \tag{2}$$

where $\nabla^{\text{sym}} = \nabla + \nabla^T$ represents the symmetrized gradient operator, p and μ are the fluid pressure and dynamic viscosity, while $\mathbf{F}_d$, $\mathbf{F}_s$ and $\mathbf{F}_b$ are the fluid's drag, surface tension, and body forces.

Combining the mass and momentum equations (Eqs. (1) and (2)) yields the *mixture Navier-Stokes equation*, reading:

$$\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\tfrac{1}{\rho}\nabla p + \tfrac{1}{\epsilon\rho}\mathbf{F}_\mu + \tfrac{1}{\rho}\mathbf{F}_s + \tfrac{1}{\rho}\mathbf{F}_b + \tfrac{1}{\epsilon\rho}\mathbf{F}_d, \tag{3}$$

where $\mathbf{F}_\mu = \nabla\cdot[\epsilon\mu(\nabla^{\text{sym}}\mathbf{u})]$ represents the viscous force.

The two-phase fluid's density and viscosity are defined implicitly via a *phase field* $\phi(\mathbf{x},t):\mathbb{R}^3\times\mathbb{R}\to\mathbb{R}$ that represents a fixed-profile indicator function of the light (l, for air) vs. heavy (h, for water) fluid phases, through:

$$\begin{cases}\rho = \rho_l + (\rho_h - \rho_l)\,\phi,\\ \mu = \mu_l + (\mu_h - \mu_l)\,\phi.\end{cases} \tag{4}$$

The evolution of the phase field ϕ is governed by a *conservative phase-field equation* [Li et al. 2022], written as

$$\frac{\partial\phi}{\partial t} + \nabla\cdot(\phi\mathbf{u}) = \nabla\cdot\left[M\big(\nabla\phi - \frac{4}{\xi}\phi(1-\phi)\mathbf{n}\big)\right], \tag{5}$$

where $\mathbf{n} = \nabla\phi/|\nabla\phi|$ is the interface normal, M is the mobility parameter controlling the ability of the interface to split, and ξ is the interface width dictating the profile thickness of the phase field.

3.2 Sand formulation in mixture

In classical approaches coupling fluid and DEM [Tang et al. 2025], the motion of individual sediment grains is typically governed by Newton's laws, which ignore rotational effects. We adopt instead a continuum approach following [Gao et al. 2018] to efficiently simulate densely packed suspensions involving large numbers of sand grains: we treat sand as a *continuous medium* governed by balance laws for both mass and momentum, to which a drag force density term $\mathbb{F}_d$ is added to account for momentum exchange with the surrounding fluid. In this model, the balance of mass for the sediment phase is expressed as:

$$\frac{\partial(\varepsilon\varrho)}{\partial t} + \nabla\cdot(\varepsilon\varrho\mathbf{v}) = 0, \tag{6}$$

where ε is the local sediment volume fraction, ϱ is the sediment density, and $\mathbf{v}$ is the sand velocity. As for the momentum balance equation, it reads:

$$\frac{\partial(\varepsilon\varrho\mathbf{v})}{\partial t} + \nabla\cdot(\varepsilon\varrho\mathbf{v}\otimes\mathbf{v}) = \varepsilon\nabla\cdot\sigma + \varepsilon\varrho g + \mathbb{F}_d + \mathbb{F}_b, \tag{7}$$

where σ is the Cauchy stress tensor for the sand, while $\mathbb{F}_d$, $\mathbb{F}_b$ are the sand's drag and buoyancy forces respectively. From these equations, we derive a numerical integrator to simulate mixtures next.

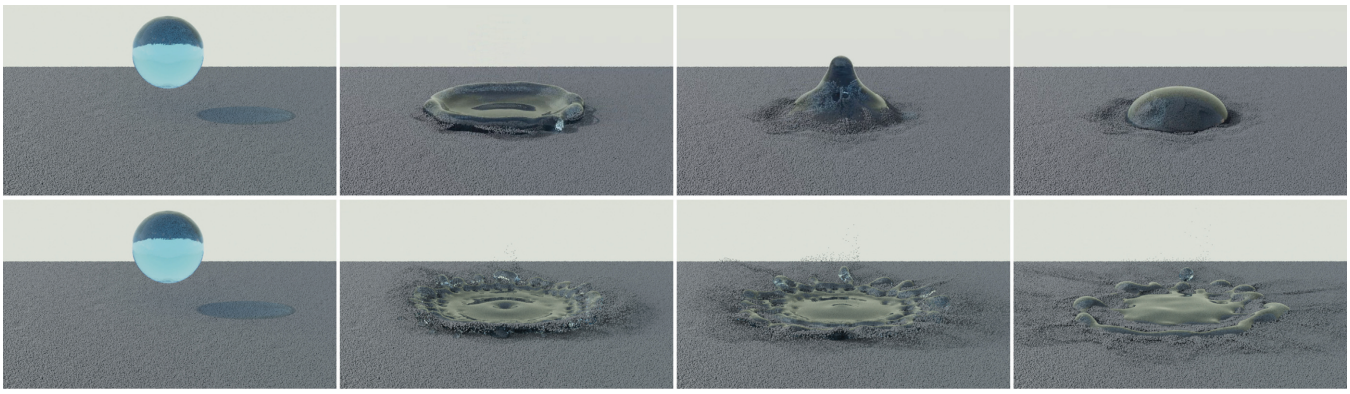

Fig. 6. **Raindrops falling on sand.** Raindrops with two different amounts of surface tension impact a flat sand pile, exhibiting distinct dynamic behaviors. High surface tension (top) quickly contains the droplet which has trapped sand grains, resulting in a local disturbance of the surface. Low surface tension (bottom) allows strong deformation and splashing that ejects sand radially. In both cases, the impact ultimately leaves a crater-like imprint on the sand.

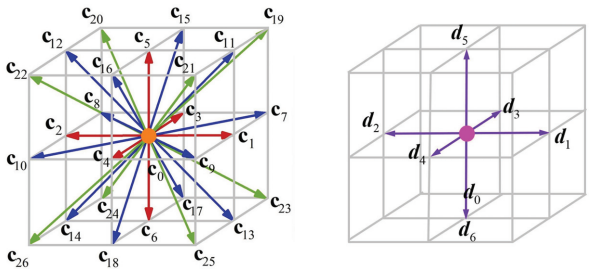

Fig. 7. **Lattice structures**. We use two different lattice structures to simulate mixtures: a D3Q27 structure (a) for flow equations (where higher accuracy is needed), and a D3Q7 structure (b) for the phase-field equation.

4 Numerical Implementation of Mixtures

We now go over how we implement numerically the mixture model we just described, detailing the LBM-based discretization of the two-phase fluid, the MPM-based simulation of sand, and the coupling between them via forces.

4.1 Two phase fluid solver in mixture

The Lattice Boltzmann Method has proven highly effective for single- and two-phase weakly compressible flows, outperforming conventional Navier–Stokes solvers due to its massively-parallel nature and high accuracy for turbulent flows. We extend this framework to fluid–mixture settings to retain LBM's key advantages (algorithmic simplicity, parallel efficiency, and robust interface handling) even in the presence of fluid–sand interactions. This section describes our LBM-based formulation to solve the governing two-phase fluid equations, Eqs. (3) and (5), within our mixture model.

LBM evolution equation. The flow equation is solved by the standard LBM evolution equation on a regular grid using a $D3Q27$ lattice structure (see Fig. 7), i.e.,

$$f_j(\mathbf{x}+\mathbf{c}_j, t+1) = f_j(\mathbf{x}, t) + \Omega_j(\mathbf{x}, t) + \mathbf{F}_j(\mathbf{x}, t), \tag{8}$$

where f_j is the discrete particle distribution function, Ω_j is the discrete collision operator relaxing f_j to its equilibrium state f_j^{eq}, and $\mathbf{F}_j$ is the forcing term; the spatial position of a grid node is denoted $\mathbf{x}$, the local *lattice velocity* directions at $\mathbf{x}$ are written as $\{\mathbf{c}_j\}_j$, while t denotes the discrete time. This equation is integrated via operator splitting, consisting of a collision step followed by a streaming step:

- **collision step:** the distribution function is first updated based on the collision and forcing terms through:

$$f_j^*(\boldsymbol{x}, t) = f_j(\boldsymbol{x}, t) + \Omega_j + F_j \ . \tag{9}$$

- **streaming step:** the values f^* are then streamed to neighboring nodes to provide the next distribution function values via

$$f_j(\boldsymbol{x}, t+1) = f_j^*(\boldsymbol{x} - \mathbf{c}_j, t) \ . \tag{10}$$

Velocity-based formulation. To solve Eq. (3), we employ a *velocity-based* D3Q27 formulation, which offers superior numerical stability compared to traditional momentum-based approaches [Kim and Pitsch 2015; Sitompul and Aoki 2019] due to a reduced impact of density discontinuities which, otherwise, may cause instability [Brackbill et al. 1992]. In practice, this simply means that we use a *density-scaled* distribution function $\mathbf{f}$ compared to usual LBM solvers, such that its zeroth-order velocity-based moment represents the local fluid density ratio (fluctuating near unity due to the weakly-compressible nature of LBM) rather than the absolute fluid density. While this encoding necessitates dealing with pressure terms through forcing, this treatment will be key, later on, to deal with interaction between fluid and sand in a manner that enforces *incompressibility* of the fluid in the mixture.

NOCM-MRT collision model. As the choice of collision evaluation is also crucial for numerical stability and accuracy, we employ the *Non-Orthogonal Central-Moment Multi–Relaxation-Time* (NOCM–MRT) collision model recommended in [Li et al. 2020]: the NOCM-MRT collision operator is computed in moment space, then projected back in distribution space, through:

$$\Omega = -\mathbf{M}^{-1}\mathbf{R}(\mathbf{m} - \mathbf{m}^{\text{eq}}), \tag{11}$$

where $\mathbf{R}$ is a diagonal matrix containing the individual relaxation rates $\{r_j\}_{j=1\ldots26}$. Following two-phase LBM practices, the low and high-order relaxation rates are set to unity, resulting in $\{r_j\}_{j=1\ldots26} = \{1, \ldots, 1, r_4, r_5, 1, \ldots, 1\}$ — see, e.g., [Li and Desbrun 2023]. Here, $\mathbf{m}$ and $\mathbf{m}^{\text{eq}}$ are the moment-space distribution function and its equilibrium

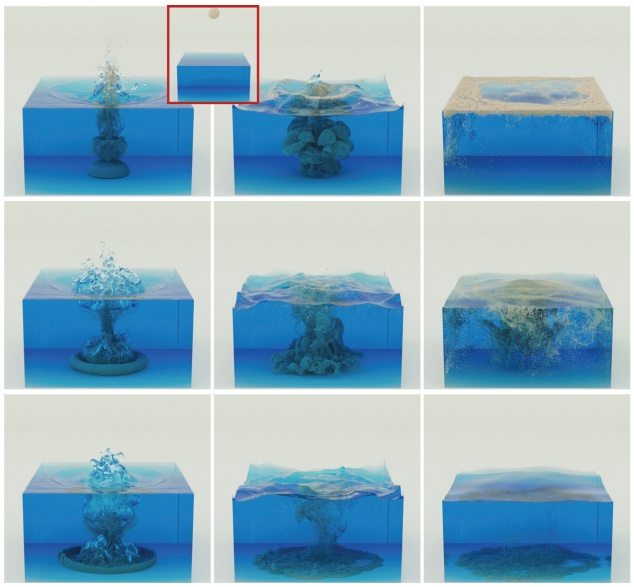

Fig. 8. **Sands of varying densities dropping into water.** Sand balls with varying densities (600, 950 and 1500 kg m^{-3}) are dropped into water (initial configuration shown in the top inset) illustrating density-dependent water–sand interactions. Light sand rises and forms distinct suspended patterns near the surface, whereas heavy sand sinks rapidly and accumulates as a heap at the container's bottom.

state, that is, $\mathbf{m}^{\mathrm{eq}} = \mathbf{M}\mathbf{f}^{\mathrm{eq}}(\boldsymbol{u})$ where density-scaled $\mathbf{f}^{\mathrm{eq}}(\boldsymbol{u})$ is the thermodynamic equilibrium state for a macroscopic velocity $\boldsymbol{u}$, which we evaluate as a sixth-order Hermite expansion of the continuous Maxwell–Boltzmann distribution following [Li et al. 2020] modulo the normalization by the density to adapt their approach to our velocity-based formulation. This expansion is highly sparse when evaluated in non-orthogonal central-moment space: only a small subset of moments remains non-zero, reducing the computational costs. As a result, the discrete velocity-based equilibrium can be written in a compact form, with all entries being zero except for the following non-vanishing terms:

$$m_0^{\mathrm{eq}} = m_9^{\mathrm{eq}} = 1,\ m_{17}^{\mathrm{eq}} = \frac{1}{3},\ m_{18}^{\mathrm{eq}} = \frac{1}{9},\ m_{26}^{\mathrm{eq}} = \frac{1}{27}. \tag{12}$$

This NOCM-MRT approach to evaluating the collision operator enhances numerical stability and improves isotropy by relaxing central moments in a non-orthogonal basis with independently prescribed relaxation rates, offering a robustness which will be proven key in our demanding context involving large density ratios and strong interfacial dynamics.

Macroscopic behavior. After the collision/streaming steps in our velocity-based formulation, one can recover the macroscopic velocity $\mathbf{u}$ through the first velocity-based moment of $\mathbf{f}$, i.e.,

$$\sum_{i=0}^{26} f_j \mathbf{c}_j = \mathbf{u}. \tag{13}$$

However, it was shown [Banari et al. 2014; Li 2015] through Chapman-Enskog analysis that this use of a normalized distribution leads to a *pressure-less* momentum equation, i.e.,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot[\nu\nabla^{\mathrm{sym}}\mathbf{u}], \tag{14}$$

where the kinematic viscosity $\nu = \mu/\rho$ is related to the relaxation rates r_4 and r_5 via $r_4 = r_5 = (\nu/c_s^2 + 1/2)^{-1}$, with c_s^2 being the lattice speed of sound. To recover Eq. (3), we must add all the terms missing from Eq. (14) as proposed in [Sitompul and Aoki 2019], i.e.,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = \nabla\cdot[\nu\nabla^{\mathrm{sym}}\mathbf{u}] + \left[\frac{\nu}{\epsilon\rho}\nabla^{\mathrm{sym}}\mathbf{u}\ \nabla(\epsilon\rho) - \frac{\nabla p}{\rho} + \frac{\mathbf{F}_s}{\rho} + \frac{\mathbf{F}_b}{\rho} + \frac{\mathbf{F}_d}{\epsilon\rho}\right]. \tag{15}$$

These terms in brackets are incorporated as *forcing terms* (i.e., external forces), allowing the intended momentum equation to be recovered. The components of the added forcing term $\mathbf{F}$ to the velocity-based distribution to get the correct momentum equation are then formulated as follows. The viscous force is expressed as:

$$\mathbf{F}_\mu = \frac{\nu}{\epsilon}\nabla^{\mathrm{sym}}\mathbf{u}\ \nabla(\epsilon\rho) = \nu\nabla^{\mathrm{sym}}\mathbf{u}\left[\nabla\rho + \frac{\rho}{\epsilon}\nabla\epsilon\right], \tag{16}$$

while the pressure force, for a pressure expression that we will discuss later, is expressed as

$$\mathbf{F}_p = -\nabla p. \tag{17}$$

All gradients are evaluated via most-rotationally-symmetric finite difference evaluations; for instance, for the fluid volume fraction,

$$\nabla\epsilon = \sum_{i=0}^{26} 3w_j^c\mathbf{c}_j\ \epsilon(\mathbf{x}+\mathbf{c}_j), \tag{18}$$

where w_j^c is the lattice weight for direction $\mathbf{c}_j$ [Fakhari et al. 2017]. The gravity-based body force is simply

$$\mathbf{F}_b = \rho\mathbf{g}, \tag{19}$$

where $\mathbf{g}$ is the gravity acceleration, and the last term is the surface tension force $\mathbf{F}_s$, for which we adopt the formulation from [Fakhari et al. 2017] to write it as a function of the phase field ϕ through:

$$\mathbf{F}_s = \chi_\phi\nabla\phi, \tag{20}$$

where the phase-field gradient is approximated through

$$\nabla\phi = \textstyle\sum_j 3w_j^c\mathbf{c}_j\phi(\mathbf{x}+\mathbf{c}_j). \tag{21}$$

The scalar field χ_ϕ, viewed as a chemical potential, is chosen as in [Jacqmin 1999], i.e.,

$$\chi_\phi = 4\beta\phi(\phi-1)(\phi-\tfrac{1}{2}) - \kappa\nabla^2\phi, \tag{22}$$

where the second-order Laplacian of the phase is approximated as:

$$\nabla^2\phi(\mathbf{x}) = 6\sum_j w_j^c(\phi(\mathbf{x}+\mathbf{c}_j) - \phi(\mathbf{x})). \tag{23}$$

Finally, the total forcing is $\mathbf{F} = \mathbf{F}_\mu + \mathbf{F}_p + \mathbf{F}_b + \mathbf{F}_s + \mathbf{F}_d$.

4.2 Sand solver

To efficiently resolve the granular phase while accurately capturing its continuum behavior, we employ the Material Point Method (MPM) proposed in [Gao et al. 2018]. The sediment is modeled using a collection of Lagrangian particles, each characterized by its mass m_p, position $\mathbf{x}_p$, velocity $\mathbf{v}_p$, and deformation gradient $\mathbf{F}_p$. To reduce dissipation and conserve angular momentum during the necessary particle-to-grid transfer in MPM, we utilize the Affine Particle-In-Cell (APIC) transfer scheme from Jiang et al. [2015].

Particle-to-grid transfer. At each time step t^n, particle mass and momentum are first transferred from the particles to the nodes of a background Eulerian grid. Using quadratic B-spline shape functions $\{N_k\}_k$, the grid mass m_i^s and momentum $(m\mathbf{v})_i$ for each node i are computed from nearby particles (indexed by p below) as:

$$m_i^s = \sum_p m_p N_i(\mathbf{x}_p),\quad (m\mathbf{v})_i = \sum_p m_p N_i(\mathbf{x}_p)[\mathbf{v}_p + \mathbf{B}_p(\mathbf{x}_i - \mathbf{x}_p)], \tag{24}$$

where $\mathbf{B}_p$ is the affine velocity matrix stored on particles. The intermediate grid velocity is then obtained via $v_i^n = (m\mathbf{v})_i/m_i^s$.

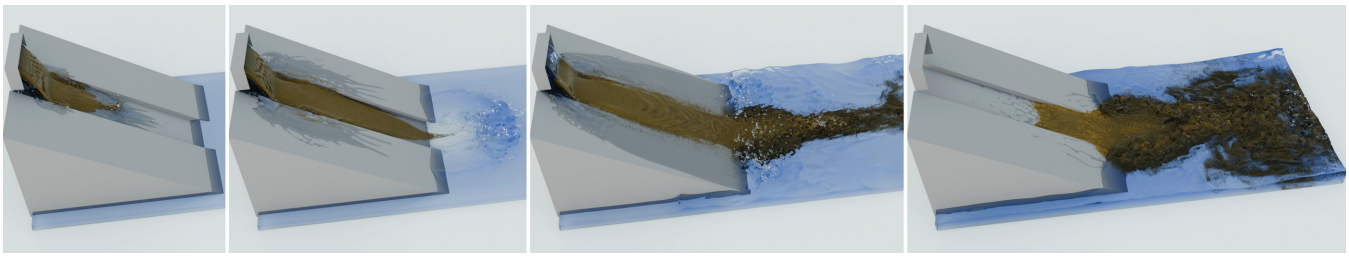

Fig. 9. **Sediment-laden flow and delta formation.** As a mixture flows down a slope, the water phase moves fast while the sand slows down due to gravity and reduced near-bed velocities. Water spreads into a thin wetting film along the incline, forming a leading wetting front around the sand-rich core, and it deposits sediment at the slope exit which builds a fan-shaped delta.

Volume fraction computation. A critical component of our two-phase coupling is the accurate evaluation of the sediment volume fraction ε on the grid, which informs the LBM solver of the part of the domain occupied by the solid phase. Following [Gao et al. 2018], we estimate the nodal volume fraction by locally rasterizing particle volumes V_p (assumed constant) using

$$\varepsilon_i = \min\big(\varepsilon_{\max}, \frac{1}{V_{\text{cell}}} \sum_p V_p N_i(x_p)\big), \tag{25}$$

where V_{cell} is the volume of a grid cell, while $\varepsilon_{\max} = 0.64$ [Baranau and Tallarek 2014] represents the maximum packing density for sand grains. The fluid volume fraction is then defined as $\epsilon_i = 1 - \varepsilon_i$, as the fluid fills up the interstices in between the packed sand grains.

Grid momentum update. The momentum equation (7) is solved on the Eulerian grid. The total force acting on a grid node $\mathbb{F}_i^{\text{tot}}$ is

$$\mathbb{F}_i^{\text{tot}} = \mathbb{F}_{e,i} + m_i^s \mathbf{g} + \mathbb{F}_{d,i} + \mathbb{F}_{b,i}, \tag{26}$$

where the elastic force $\mathbb{F}_{e,i}$ is derived from the divergence of the stress tensor through:

$$\mathbb{F}_{e,i} = -\textstyle\sum_p V_p^0 \sigma_p \nabla N_i(\mathbf{x}_p), \tag{27}$$

where V_p^0 is the initial particle volume and σ_p is the Cauchy stress, We follow [Tampubolon et al. 2017; Su et al. 2023] and derive the Cauchy stress from the hyperelastic potential energy density ψ^s using the elastic deformation gradient. To capture the cohesive effects of the sand-fluid mixture, we also use their modified Drucker-Prager yield criterion that adapts to the saturation state, i.e.,

$$c_F \text{tr}(\sigma_p) + \left\| \sigma_p - \frac{\text{tr}(\sigma_p)}{d} \mathbf{I} \right\|_F \leq c_C(S_{r,p}), \tag{28}$$

where d is the spatial dimension, and $c_C(S_{r,p})$ represents the cohesion strength as a function of the water saturation $S_{r,p}$, which, itself, is approximated by the surrounding grid water volume fraction as $S_{r,p} = \sum_i \epsilon_i \phi_i N_i(\mathbf{x}_p)$. The term c_F is the friction coefficient determined by the internal friction angle, which evolves according to the hardening model of Klár et al. [2016]. We adopt the hardening parameters $h_0 = 35°$, $h_1 = 9°$, $h_2 = 0.2$, and $h_3 = 10°$, while the relationship between c_C and $S_{r,p}$ is modeled using the piecewise linear function used in [Su et al. 2023]

Finally, $\mathbb{F}_{d,i}$ represents the discrete fluid-sand interaction drag and $\mathbb{F}_{b,i}$ the buoyancy force described in Sec. 4.3. Grid velocities are then updated using the same timestep Δt (in physical space) as the LBM integrator via

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n + \tfrac{\Delta t}{m_i^s} \mathbb{F}_i^{\text{tot}}. \tag{29}$$

Constitutive model and advection. We model the sand as a cohesionless, elastoplastic material with a Hencky strain energy density obeying the Drucker-Prager yield criterion [Gao et al. 2018]. The trial deformation gradient is computed as $\mathbf{F}_p^{\text{trial}} = (\mathbf{I} + \Delta t \nabla \mathbf{v}_p)\mathbf{F}_p^n$. If the stress state exceeds the yield surface defined by the friction angle, a plasticity flow return mapping algorithm projects the deformation back to the yield surface. Finally, the updated grid velocities are transferred back to update the particle velocities through

$$\mathbf{v}_p^{n+1} = \sum_i \mathbf{v}_i^{n+1} N_i(x_p) \tag{30}$$

so that the particles can be advected via a simple Euler integration step $\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^{n+1}$, while the affine matrices $\mathbf{B}_p$ are updated to prepare for the next time step.

4.3 Interaction terms

We now go over the interaction between sand and fluid simulation.

Drag force on sand. The dominant momentum exchange between the carrier fluid and the sediment phase is modeled via a *drag force* density, appearing with opposite signs in the fluid and sand momentum equations. The drag force on node i is defined through $\mathbb{F}_d^i = w(\mathbf{u}_i, \mathbf{v}_i)\ (\mathbf{u}_i - \mathbf{v}_i)$, where $\mathbf{u}$ and $\mathbf{v}$ denote the fluid and sand velocities, respectively, and w is the effective drag coefficient used in [Gao et al. 2018], i.e., $w(\mathbf{u}_i, \mathbf{v}_i) = \frac{1}{2} c_i (\epsilon_i^n)^{-\chi} \rho A_i^s \left|\mathbf{u}_i - \mathbf{v}_i\right|$. The experimental coefficients c_i and χ are set to $\chi = 3.7$ and $c_i = 0.39$, while $A_i^s = 2\left(V_i^s/\pi\right)^{1/2}$ in 2D and $A_i^s = \pi\left(3V_i^s/(4\pi)\right)^{2/3}$ in 3D are the cross-sectional areas of a round ball of volume V_i^s. Since drag forces scale as A_i^s, grid nodes containing only a small amount of solid material undergo only a small drag contribution, which helps suppress spurious oscillations and other related numerical instabilities. Note that the coefficient c_i can be used as a tuning parameter for the relative drag magnitude.

Drag force on fluid. In order to guarantee conservation of total momentum in the mixture, we prescribe the drag force on the fluid as $\mathbf{F}_i^d = -\mathbb{F}_d^i / V_{\text{cell}}$ such that the force drag densities are opposite.

Buoyancy. In addition to drag, fluid-sand interaction must also account for buoyancy. It acts on the sediment phase as

$$\mathbb{F}_{b,i} = -V_{\text{cell}} \varepsilon_i \rho_i \mathbf{g}, \tag{31}$$

which accounts for the weight of the displaced fluid, with a negative sign because it counteracts the gravitational pull of the fluid, corresponding at the mixture level to the buoyant force exerted by the fluid on the sediment phase.

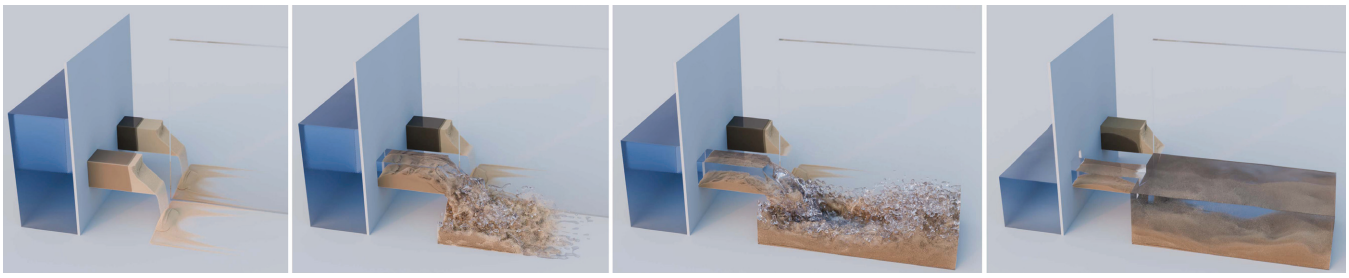

Fig. 10. **Water retention.** To demonstrate the effects of our retention model, we place two sand-filled pipes in the openings of a partition wall. Water retention is enabled for the sand within the background pipe (farthest from the camera), and disabled for the sand in the foreground pipe. With water on one side of the partition, it infiltrates the sand in the foreground pipe significantly faster, eventually eroding the sand into the pool on the opposite side. In contrast, the background sand (with retention enabled) significantly delays fluid ingress, with the bound moisture helping it to withstand the hydraulic pressure.

Pressure definition. The evaluation of pressure in our mixture context is particularly important as the pressure force (Eq. (17)) is a key part of the fluid solver in order to enforce incompressibility. In the context of multiphase LBM simulation (without sand), it is well known that the pressure p is proportional to the fluid density ρ (with a coefficient of proportionality equal to the square of the speed of sound, c_s^2), while the time derivative of the pressure p varies proportionally to the density times the divergence of the macroscopic velocity, i.e., $\frac{\partial p}{\partial t} \propto -\rho \nabla \cdot \mathbf{u}$ — see, for instance, [He et al. 1999] for more details on this weakly compressible nature of LBM simulation. In our mixture context, we need to account for the presence of sand as well. First, because we target *low Mach number* flows, the mass conservation law from Eq. (1) can be simplified to

$$\epsilon \frac{\partial \rho}{\partial t} + \rho \frac{\partial \epsilon}{\partial t} + \rho \, \nabla \cdot (\epsilon \mathbf{u}) \approx 0, \tag{32}$$

as the advective transport of the density involving $\mathbf{u} \cdot \nabla \rho$ can be neglected since the spatial density gradient and the velocity are both small in terms of lattice units due to weak compressibility and low Mach number. Moreover, we know by definition of our mixture that the fluid and sand volume fractions are summing to one everywhere:

$$\epsilon + \varepsilon = 1. \tag{33}$$

Finally, sand simulation is done via MPM, so unlike in the fluid case, the sand's material density ϱ stays constant, simplifying the sediment's mass equation from Eq. (6) to:

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot (\varepsilon \mathbf{v}) = 0, \tag{34}$$

We deduce that the time derivative of ϵ and ε are linked via:

$$\frac{\partial \epsilon}{\partial t} \overset{\text{Eq. (33)}}{=} -\frac{\partial \varepsilon}{\partial t} \overset{\text{Eq. (34)}}{=} \nabla \cdot (\varepsilon \mathbf{v}). \tag{35}$$

Now, Eq. (32) can thus be rewritten:

$$\epsilon \frac{\partial \rho}{\partial t} + \rho \nabla \cdot (\varepsilon \mathbf{v} + \epsilon \mathbf{u}) \approx 0, \tag{36}$$

which now exhibits the divergence of the mixture velocity. We thus simply define the time derivative of pressure to be:

$$\frac{\partial p}{\partial t} + \frac{c_s^2 \rho}{\epsilon} \nabla \cdot (\varepsilon \mathbf{v} + \epsilon \mathbf{u}) = 0, \tag{37}$$

to mimic the same weak compressibility of multiphase LBM flows, but now to drive the mixture towards incompressibility.

Pressure update. We use a simple Euler time integration step to turn Eq. (37) into a pressure update per time step. First, expanding the divergence $\epsilon \mathbf{u} + \varepsilon \mathbf{v}$ yields $\epsilon \nabla \cdot \mathbf{u} + \varepsilon \nabla \cdot \mathbf{v} + (\mathbf{u} - \mathbf{v}) \cdot \nabla \epsilon$, showing that the pressure time derivative responds not only to the volume changes of the individual fluid and sediment phases, but also to the relative slip between them in regions of non-uniform volume fraction. Second, we use the approximation proposed in [Sitompul and Aoki 2019] to evaluate the divergence of the fluid velocity:

$$1 - \sum_j f_j(\mathbf{x}, t) \approx \nabla \cdot \mathbf{u}. \tag{38}$$

We finally update the pressure from time t_n to time t_{n+1} as:

$$p^{n+1} = p^n - c_s^2 \rho^{n+1} \left(1 - \sum_i f_j(\mathbf{x}, t) + \frac{\varepsilon \nabla \cdot \mathbf{v} + (\mathbf{u} - \mathbf{v}) \cdot \nabla \epsilon}{\epsilon}\right), \tag{39}$$

where the terms involving gradients are computed using Eq. (18). Finally, a (local average) filter is applied to the pressure field [Sitompul and Aoki 2019] to enhance stability:

$$\bar{p}(\mathbf{x}) = \sum_j w_j^c p(\mathbf{x} + \mathbf{c}_j). \tag{40}$$

Eq .(18) is then used on this filtered pressure $\bar{p}$ to compute the pressure gradient present in Eq. (17).

4.4 Phase-field solver in mixture

To model a two-phase flow for two immiscible fluids of densities ρ_h and ρ_l and and kinematic viscosities ν_h and ν_l (where the subscripts h and l denote the high- and low-density fluids, respectively), we employ the diffuse-interface phase-field formulation of Li et al. [2022]. In this framework, a time-dependent phase indicator $\phi(x, t) \in [0, 1]$ is used as an order parameter that distinguishes spatial regions occupied by the two fluids through

$$\phi(x, t) = \begin{cases} 1 & \text{inside the high-density fluid,} \\ 0 & \text{inside the low-density fluid.} \end{cases} \tag{41}$$

The conservation equation for the fluid volume fraction states that

$$\partial_t \epsilon + \nabla \cdot (\epsilon \mathbf{u}) = 0. \tag{42}$$

The phase field being purely advected in the macroscopic velocity of the fluid, one obtains:

$$\partial_t \phi + \mathbf{u} \cdot \nabla \phi = 0, \tag{43}$$

which can be re-expressed in a conservative form with an extra source term S_ϕ as:

$$\partial_t \phi + \nabla \cdot (\phi \mathbf{u}) = \overbrace{\phi \nabla \cdot \mathbf{u}}^{:= S_\phi}, \tag{44}$$

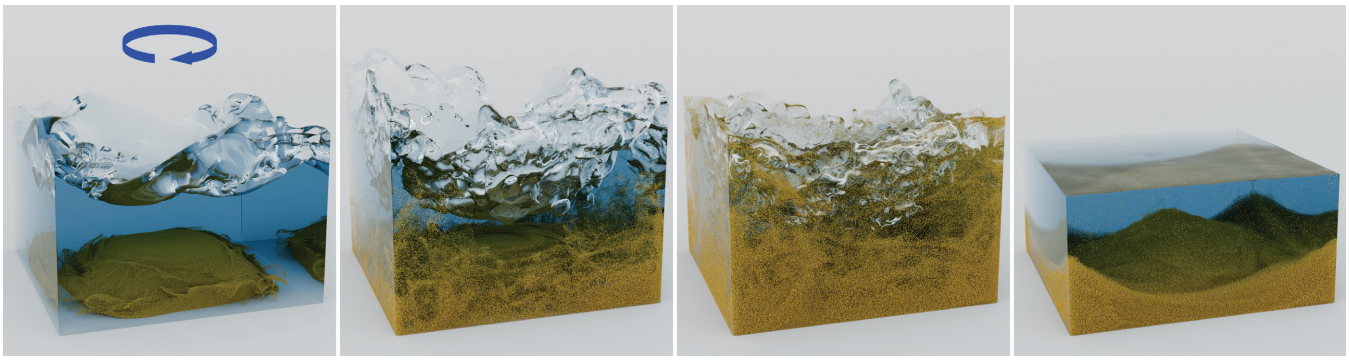

Fig. 11. **Centrifugal stirring of water and sand.** After initiating a swirl in the fluid that produces the expected parabolic free-surface profile, the rotating flow entrains the sand and transports it inward, eventually forming a sand heap at the bottom, while some sand stays stuck in the corner of the container.

Based on Eq. (42), we can express the source term as a function of the volume fraction ϵ instead to enforce the conservation equation for the fluid fraction :

$$S_\phi = \phi \nabla \cdot \mathbf{u} = -\frac{\phi}{\epsilon}\Big(\frac{\partial \epsilon}{\partial t} + \mathbf{u} \cdot \nabla \epsilon\Big) \tag{45}$$

As in [Li et al. 2022], we adopt a phase-field formulation derived from a thermodynamic free-energy functional that combines a Ginzburg–Landau double-well potential for the bulk free-energy density of each phase with a Dirichlet-type gradient term penalizing interfacial variations, expressed in conservative form through an Allen–Cahn equation [Geier et al. 2015], that we simply augment with the source term S_ϕ to account for our mixture setup:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{u}) = \nabla \cdot \Big[M\big(\nabla \phi - \frac{4}{\xi}\phi(1-\phi)\mathbf{n}\big)\Big] - \frac{\phi}{\epsilon}\Big(\frac{\partial \epsilon}{\partial t} + \mathbf{u} \cdot \nabla \epsilon\Big). \tag{46}$$

LBM discretization. For the temporal integration of Eq. (46), we adopt a kinetic formulation to handle the coupling between the phase field and the fluid motion with a *D3Q7* lattice structure (see Fig. 7) as in [Li et al. 2022]. We introduce a distribution function $\mathbf{h}$, from which the phase field is recovered as its zeroth moment:

$$\phi(\mathbf{x}, t) = \sum_{k=0}^{k=6} h_k(\mathbf{x}, t). \tag{47}$$

The phase field dynamics from Eq. (46) is then recast in kinetic form, inducing an evolution of the distribution components h_k through the lattice Boltzmann equations:

$$h_k(\mathbf{x} + \mathbf{d}_k, t+1) - h_k(\mathbf{x}, t) = \Omega_k^h(\mathbf{x}, t) + H_k(\mathbf{x}, t) + S_k(\mathbf{x}, t), \tag{48}$$

where the first force $\mathbf{H}$ represents the conservative part of the dynamics, balancing the diffusive fluxes of the phase equation and relaxing the order parameter toward the prescribed hyperbolic-tangent interface profile $\phi(\mathbf{x}, t) = \frac{1}{2}\left[1 - \tanh\big(2d_\phi(\mathbf{x})/\xi\big)\right]$ towards the equilibrium state, i.e.,

$$\mathbf{H} = \frac{4\phi(1-\phi)}{\xi}\frac{\nabla \phi}{|\nabla \phi|}. \tag{49}$$

This phase field force is projected in distribution space via:

$$H_k(\mathbf{x}, t) = w_k^d \, \boldsymbol{d}_k \cdot \mathbf{H}(\mathbf{x}, t), \tag{50}$$

where $\mathbf{d}_k$ and w_k^d denote the discrete velocities and quadrature weights of the D3Q7 lattice. The second term, S_k, derives from the source term to account for the presence of the sand phase, so that the order parameter is updated consistently with the local solid volume fraction: it ensures that the *total volume of the fluid in the mixture remains conserved*, rather than neglecting the sand presence on the fluid phase field, which would otherwise lead to a significant loss of fluid mass conservation. This term is evaluated as:

$$S_k(\mathbf{x}, t) = w_k^d \, S_\phi(\mathbf{x}, t), \tag{51}$$

where we use the same $\mathbf{d}_k$ and w_k^d of the D3Q7 lattice as above. It is important to note here that the discrete source terms S_k are constructed to strictly satisfy the moment-matching conditions required by the Chapman-Enskog analysis, i.e., $\sum_k S_k = S_\phi$ and $\sum_k S_k \mathbf{d}_k = S_\phi \sum_k w_k^d \mathbf{d}_k = \mathbf{0}$; such that the multiscale analysis of [Xiong and Cao 2025] can be applied to show that Eq. (46) is properly recovered when the LB Eqs. (48) are integrated in time.

Volume preservation in time. Evaluating the source term numerically leads to imperfect volume conservation as *cumulative discretization errors* generate gradual loss or gain of fluid volume over long simulation runs. To mitigate this effect, we record the initial total fluid volume V_0 at the start of the simulation. At each subsequent time step, we calculate the current volume $V(t)$ and apply a *proportional controller* to adjust the source term's net contribution. This ensures the corrected volume V is actively driven back toward V_0 according to the relation $S_\phi^c(\mathbf{x}, t) = \gamma(t)\, S_\phi(\mathbf{x}, t)$, where correction factor $\gamma(t)$ is defined as:

$$\gamma(t) = \text{clamped}\Big(\frac{V_0 - V(t)}{\sum_{\mathbf{x}} S_\phi(\mathbf{x}, t)}, [-\eta, \eta]\Big). \tag{52}$$

The corrected source term $S_\phi^c(\mathbf{x}, t)$ is thus a scaled version of the original source term, with a scaling factor $\gamma(t)$ clamped within a range $[-\eta, \eta]$ for a threshold η which we pick to be $\eta \in [1, 2]$. This mechanism ensures that the correction remains stable while effectively counteracting the numerical volume drift.

4.5 Water retention model

While the governing equations described above effectively capture the macroscopic dynamics of the mixture, they do not explicitly account for the microscopic capillary forces acting within the porous sand structure. In reality, matric suction and surface tension trap a residual amount of fluid between grains during drainage. Relying solely on advection of the phase field would cause the sand to dry out instantaneously (and thus non-physically) as the bulk fluid recedes. To reproduce the realistic behavior of wet sand (where moisture remains effectively bound to the sediment even after the bulk fluid is gone), we adopt a water retention strategy inspired by the Granule-in-Cell (GIC) method [Tang et al. 2025], where free-flowing fluids are

Fig. 12. **Sediment-laden dam break over bunny.** We simulate a dam break involving a water–sand mixture and a rigid bunny obstacle. When the flow strikes the bunny, the water splashes around the geometry while a sand-rich mixture piles up against the obstacle, flowing around it with reduced mobility.

captured by unsaturated sand granules and converted into bound moisture mass. This mechanism prevents the complete separation of the fluid phase from the sand phase and allows for the simulation of varying degrees of saturation, from dry to fully saturated states.

We introduce a time-dependent moisture ratio $r_p(t) \in [0, r_{\max}]$ for each sand particle p, where $r_p(t)V_p$ denotes the volume of water bound to the particle. Rasterizing these bound volumes onto the grid yields the local bound water volume fraction $R_i(t)$ at node i:

$$R_i(t) = \frac{V_p}{V_{\text{cell}}} \sum_p r_p(t) N_i(\mathbf{x}_p). \tag{53}$$

To account for this retention, we define an *effective porosity* $\hat{\epsilon}_i$, which represents the volume fraction of fluid after excluding both the solid sediment and the bound moisture, i.e.,

$$\hat{\epsilon}_i(t) = 1 - \varepsilon_i(t) - R_i(t). \tag{54}$$

We then determine the increment in the local bound water volume fraction at node i, denoted as $\Delta R_i(t)$, through

$$\Delta R_i(t) = \min\big(\hat{\epsilon}_i \phi_i, \min(R_a \Delta t r_{\max} \varepsilon_i, r_{\max}\varepsilon_i - R_i(t))\big), \tag{55}$$

where $r_{\max}\varepsilon_i$ represents the maximum water retention capacity of the sediment, while R_a is a tunable absorption coefficient determining the uptake rate. The inner minimum restricts the absorption to the remaining retention capacity ($r_{\max}\varepsilon_i - R_i(t)$), while the outer minimum caps the intake at the currently available fluid volume ($\hat{\epsilon}_i\phi_i$). Subsequently, the bound water fraction is updated via:

$$R_i(t+1) = R_i(t) + \Delta R_i(t). \tag{56}$$

In order to maintain mass conservation during the transition of water from a free to a bound state, the calculation of the phase field evolution (Eq. (46)) must use a modified phase field source term S_ϕ (see Eq. (45)) using $\hat{\epsilon}$ as the effective volume fraction, that is,

$$S_\phi = -\frac{\phi}{\hat{\epsilon}} \left(\frac{\partial \hat{\epsilon}}{\partial t} + \mathbf{u} \cdot \nabla \hat{\epsilon} \right) - \frac{\Delta R_i}{\Delta t \hat{\epsilon}}. \tag{57}$$

Finally, the particle moisture ratios are updated by transferring the grid-based increments back to the particles:

$$r_p(t+1) = r_p(t) + \textstyle\sum_i \frac{\Delta R_i(t)}{\varepsilon_i} N_i(\mathbf{x}_p). \tag{58}$$

When water retention is active, the water saturation state $S_{r,p}$ — which governs the cohesion strength c_C — is redefined as the sum of the bound water fraction and the local free fluid fraction:

$$S_{r,p} = \textstyle\sum_i (R_i + \hat{\epsilon}_i \phi_i) N_i(\mathbf{x}_p). \tag{59}$$

Furthermore, when water retention is enabled, $\hat{\epsilon}$ replaces ϵ in the calculation of Eqs. (39) and (16), influencing the rate of water infiltration into sand. Figs. 17 and 10 show two examples where water retention meaningfully changes the visual behavior.

5 Results and Discussions

We now present a comprehensive set of evaluations designed to highlight the advantages of our method. These experiments purposely span a wide range of scenarios involving mixtures between two-phase flows and sand, including phase separation, sediment entrainment and deposition, turbulent mixing, splashing, bubbling, wetting and absorption of water by sand, as well as erosion-driven interactions with complex boundary conditions. We include qualitative comparisons against prior approaches and real-world observations. A series of targeted ablation studies are also presented to assess the impact and effectiveness of our specific contributions — most notably, of our proposed mass conservation term.

5.1 Implementation Details

We implemented our mixture solver, whose pseudocode is given in Alg. 1, in C++ and CUDA to best exploit massively parallel architectures. To optimize memory access patterns on the GPU, we employ a structure-of-arrays (SoA) data layout [Chen et al. 2022] for all field variables. Unlike traditional double-buffering LBM schemes, we utilize an inplace streaming algorithm [Li et al. 2022] requiring a single copy of the distribution functions, thus significantly reducing memory use. While most computations are performed in single precision to maximize throughput, we employ double precision for the calculation of the total fluid volume to limit errors accumulating from floating-point truncation (see Sec. 4.4). For the discretization of the governing equations, we utilize different lattice structures tailored to the specific physical fields as mentioned in Sec. 3: the phase field evolution is solved using a D3Q7 lattice, while the hydrodynamic velocity field employs a D3Q27 lattice to ensure better stability and rotational invariance (see Fig. 7). Furthermore, the boundary of the granular phase (sand) is represented using a signed distance field (SDF) computed via `libigl` [Jacobson et al. 2018]. To minimize computational overhead, the fluid solver reuses these pre-computed SDFs to determine connected domains and handle fluid-solid interactions. Finally, the conversion between physical units and LBM units follows the dimensionless procedure described in the supplementary material of [Li et al. 2020]. All our experiments were conducted on a NVIDIA RTX PRO 6000 GPU equipped with 96 GB of memory. We also use Blender [Blender Online Community 2018] to render the results. Water-in-sand effects are visualized with geometry nodes to determine the sand color based on the amount of retained water. Note that in simulations where water retention is disabled, we visually differentiate between dry and wet sand by classifying and rendering particles as "wet" whenever their saturation $S_{r,p}$ exceeds $(1-\varepsilon_{\max})/2$.

ALGORITHM 1: Coupled LBM-MPM Mixture Solver

Initialize: Fluid distributions f_i, h_i, Sand particles $\mathbf{x}_p, \mathbf{v}_p, \mathbf{F}_p$;
for $t = 1$ *to* N **do**
  // — MPM Stage 1: Transfer —
  MPM_ParticlesToGrid() ; ▷ Eq. (24)
  // — MPM Stage 2: Volume Fraction & Grid Update —
  ComputeVolumeFractions() ; ▷ Get ε, ϵ and $\hat{e}$, Eqs. (25), (33), (54)
  // — Coupling: Water Retention —
  UpdateWaterRetentionAndSource() ; ▷ Sec. 4.5
  MPM_UpdateGridMomenta() ; ▷ Add total force to sand, Eqs. (26), (29)
  // — MPM Stage 3: Advection —
  MPM_GridToParticlesAndAdvect() ; ▷ Strain update & Advection, Eq. (30)
  // — LBM Stage 1: Phase & Fluid Macroscopic Variables Updation —
  Update phase field ϕ ; ▷ Eq. (47)
  Update velocity field $\mathbf{u}$; ▷ Eq. (13)
  // — Coupling: Pressure and Drag Force —
  UpdatePressureFieldAndApplyFilter() ; ▷ Eqs. (39), (40)
  ComputeCouplingAndBodyForces() ; ▷ Secs. (4.1), (4.3)
  // — LBM Stage 2: Phase & Fluid Evolution —
  PhaseFieldCollision() ; ▷ Eqs. (48), (45) and (57)
  PhaseFieldStreaming() ; ▷ Eq. (48)
  FluidCollision() ; ▷ Eq. (9), NOCM-MRT
  FluidStreaming() ; ▷ Eq. (10)
end

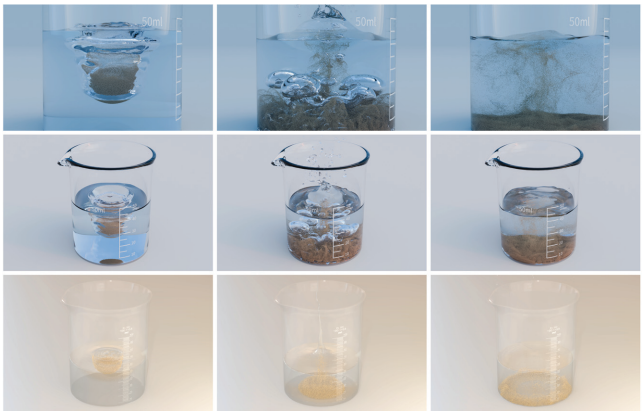

Fig. 13. **Comparison with [Tang et al. 2025]**. For a simple sand drop in a beaker, our approach (top) exhibits the same overall coupling behavior between sand and fluid as in [Tang et al. 2025], but with clearly more visible bubbles and splashes, and at a 12 times lower computational cost.

5.2 Comparisons

We begin the presentation of our results with a few comparisons.

Comparison with an existing method. In Fig. 13, we compare our approach with the state-of-the-art method [Tang et al. 2025] using their example of a sand ball dropped in a beaker, with sand density set to 1500 kg m^{-3}. Both methods show the sand sinking to the bottom of the beaker along with water splashing. Our method produces bubbles and splashing effects due to our less viscous fluid, capturing the settling and dispersal behavior of sand as it falls through water, resulting in more realistic and visually engaging effects. In contrast, [Tang et al. 2025] exhibits noticeably more viscous water behavior, with no observable interaction between sand and bubbles. Moreover, their CPU implementation requires approximately 10.3 mins per frame in this example, whereas our GPU-based method takes only about 0.87 mins per frame, achieving a 12× speedup.

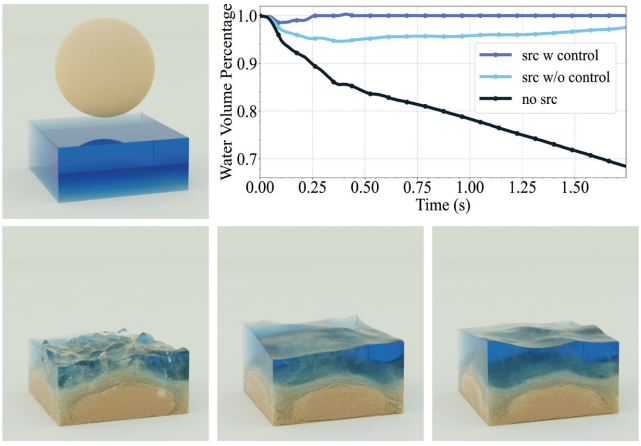

Fig. 14. **Volume preservation in sand drop.** When a big ball of sand drops in a small amount of water, our approach (c) maintains a near-constant fluid volume in time as indicated in the upper right plot. However, if we remove our source term S_ϕ in the phase field equation, the water level does not grow as the sand falls in the container (a), resulting in a drastic loss of volume. If we add back our source term but now remove our volume controller (b), volume variations reach around 5% during the simulation, proving that both the source term and the controller are necessary for proper fluid volume preservation in a mixture simulation.

Ablation test for source term. We keep the sand-drop example to conduct an ablation study on our proposed source-term design (Fig. 14). We first compare simulations with and without the source term from Eq. (45). Without the source term, the water level does not rise to accommodate the presence of sand, which leads to 30% mass loss. In contrast, incorporating the source term correctly elevates the water level to account for the fluid volume displaced by sand. We further compare our results with and without the volume proportional controller to help maintain the initial volume more precisely. As shown in Fig. 14, disabling the controller leads to noticeable volume non-conservation over time (up to 5%). By contrast, with the volume proportional controller, the total fluid volume is well preserved, with variations remaining below 1% over time.

Comparison with real experiment. We further compare our method with a real-world experiment of granular collapse into water, as shown in Fig. 15. Our results are compared against the experimental data reported in [Kane 2025], using the same experimental setup described in [Robbe-Saule et al. 2021] and for a Froude number of 3.6 (representing the square root of the ratio between the initial granular column height and the water height). We reproduce the experiment with our method by setting the sand density to 900 kg m^{-3} as the experiment suggested. Our method captures sand-water interactions well: the sand falls into the water and generates a wave which pushes up the sand and forms bubbles and splashing before settling down, thus matching the overall behavior of the real experiment.

5.3 Simulation results

We now go over a series of examples we ran to test the visual accuracy of our sand-water-air mixture solver. Fluid phase densities for all examples are fixed at $\rho_h = 1000$ kg m^{-3} and $\rho_l = 1.25$ kg m^{-3}.

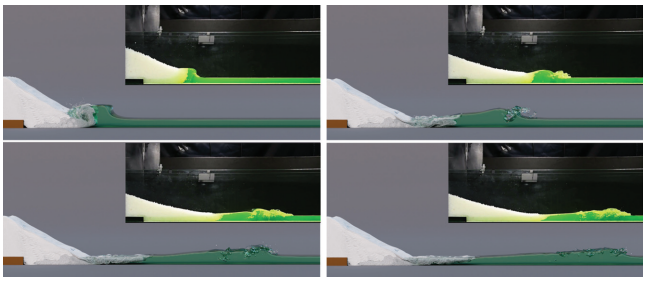

Fig. 15. **Comparison with granular collapse experiments.** A block of sand collapses onto the water and generates a wave, rolling up and carrying the sand forward. Our simulation closely mimics the sand-water dynamics of real-word experiments (showed in top-right snapshots).

5.3.1 *Fluid–sand simulation in simple domains.*

Sands of varying densities dropping into water. In Fig. 8, we drop a small sand ball into water and vary its density (600, 950 and 1500 $\mathrm{kg\,m^{-3}}$) to evaluate how sand density influences the resulting dynamics. When the sand is light, it drops into the water with splashes, forming a mushroom-like shape before finally floating on the water surface as a layer of sand. For medium density sand, the sand drops with stronger splashes, and the sand gradually floats back up the surface of the water due to its density being slightly lower than that of water. For heavy sand, the sand quickly sinks to the bottom of the container to form a sand heap.

Water-sand dam break. We simulate in Fig. 16 a dam-break scenario involving both water and sand. The water rapidly surges forward, impacts the sand pile, and entrains grains into suspension, producing splashes and air-entrained bubbles in the impact area. Since the sand is less dense than water in this setup ($\varrho = 900\ \mathrm{kg\,m^{-3}}$), a part of the sand grains rises and gets trapped by the surface flow, forming coherent, spiral-like sand streaks and vortical patterns near the free surface.

Centrifugal stirring of water and sand. In Fig. 11, we feature an example where a container holds water and a sand pile at the bottom, before water is stirred. At the initial stage, we apply a tangential forcing to the liquid to initiate a swirl. The flow quickly develops into a coherent rotation and, under the combined effects of surface tension and centrifugal acceleration, the free surface rises along the container walls and forms the expected parabolic profile. Meanwhile, the circulating motion entrains and transports sand grains; as the flow evolves, the grains migrate toward the center and gradually accumulate into a pronounced heap at the bottom.

Raindrop on sand. We simulate a raindrop falling on a flat sand pile for two different surface tension values in Fig. 6. Initialized with a downward velocity, the raindrops exhibit markedly different impact dynamics: low surface tension leads to strong droplet deformation, accompanied by pronounced splashing with sand ejected and dispersed radially by the droplet motion, whereas high surface tension stabilizes the droplet after capturing a few sand grains, resulting in a much milder impact creating limited disturbance.

Dam break over a sand castle. Fig. 3 shows a wall of water crashing into a sand castle to demonstrate our method's ability to capture erosion, structural failure, and wetting phenomena. As the water front reaches the castle, the incoming flow exerts strong shear and impact forces on the sand structure, causing surface grains to detach and be rapidly transported downstream. With continued loading, the integrity of the castle progressively degrades, and the turrets collapse under the combined effects of hydraulic pressure and loss of granular support. Following the initial impact and collapse, water infiltrates and wets the remaining body of sand, darkening the structure and further weakening its cohesion, ultimately leading to the collapse of the main keep. The simulation vividly reproduces the erosive action of a dam break flow on a sand castle, including grain detachment, partial structural failure, and large-scale wetting, highlighting the realism and robustness of the proposed approach in modeling fluid-driven erosion processes.

Rectangular dam outlet mixture flow. We simulate a dam release scenario through a rectangular outlet to observe transient discharge behaviors and water–sand interactions. In Fig. 1, at the initial stage, clean water is released from the inlet and forms a high-speed jet. As the jet flows out, sediments are entrained into the flow, and a water-sand mixture is expelled, producing a visually complex, yellow-tinted jet indicative of intense mixing. Upon impingement into the downstream basin, the mixture generates vigorous splashing and bubble formation, accompanied by pronounced sediment settling and complex sand rearrangement across the basin floor.

Breaching of a sand bailey. To evaluate the method's ability to capture free-surface disturbances and erosion-driven structural failure, Fig. 5 shows a scenario where water confined within a sand-walled basin gradually breaches the surrounding sand barriers before breaking them open. As water falls into the basin, it induces localized surface oscillations; these repeated disturbances amplify wave motion within the basin, increasing pressure and shear stresses along the sand walls. Meanwhile, sediment transported by the oscillatory flow settles on the basin floor, forming successive ripple patterns that evolve with the changing flow conditions. Over time, water infiltrates and erodes the sand structure, weakening its mechanical stability. Eventually, fractures emerge in the sand bailey, followed by progressive collapse and outward release of the water.

Wind-driven water-sand mixing. In Fig. 2, we demonstrate wind-driven water–sand interactions using our mixture simulation. A sand island (with density $\varrho = 600\ \mathrm{kg\,m^{-3}}$ so as to float) in the shape of a SIGGRAPH logo is initially placed slightly above the water surface, while a strong air inflow strikes the island from above. After getting in contact with the water, the sand floats but becomes wet. As the wind intensifies, it generates waves and splashes, and simultaneously kicks up the sand in wind-aligned patterns. The airflow gradually lifts and entrains fine sand from the mass and eventually breaks the logo apart into discrete fragments that disperse across the container and drift on the water surface.

Moisture-induced cohesive strengthening. To showcase our ability to couple water retention and cohesion, we simulate the collapse of a sand column under dry and wet conditions (Fig. 17). Upon removing the boundary constraints holding a block of sand together, dry sand lacks interstitial moisture, leading to immediate gravitational collapse. In contrast, if we pour water on it first, the sand

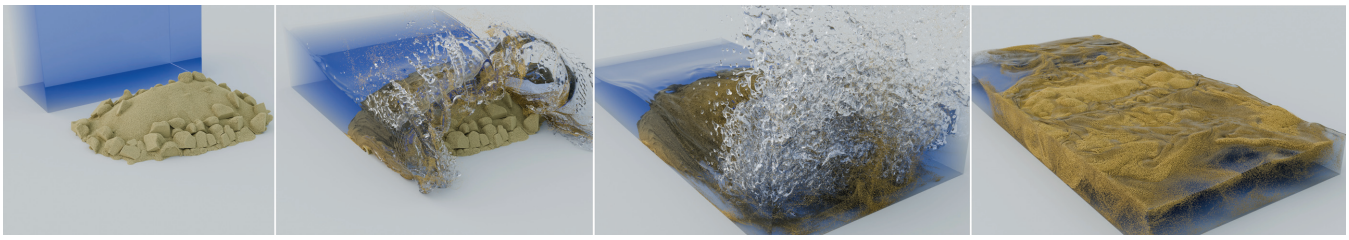

Fig. 16. **Dam break over sand pile.** As a water dam breaks, it impacts a sand pile, entraining grains and generating splashes with air-entrained bubbles. With sand less dense than water in this setup ($\varrho = 900\ \mathrm{kg\,m^{-3}}$), sand grains rise and form vortex-like streaks on the free surface.

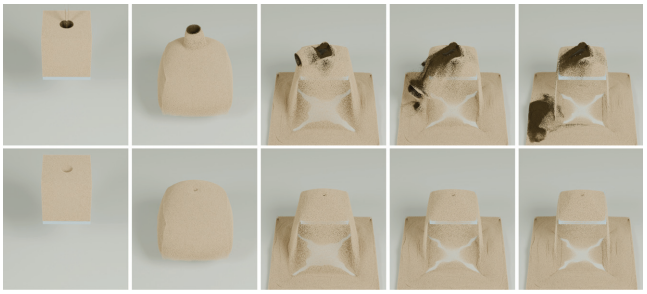

Fig. 17. **Moisture-induced cohesive strengthening** This simulation demonstrates the coupling between our water retention model and the sand cohesive forces. From a sand block in an invisible box-shaped container, we try two scenarios: we either pour water over the sand (top), or not (bottom). When the container is artificially removed, we see that in the case of the wet sand, its cohesive forces have increased due to the local saturation. As a consequence, the block of sand keeps its structural integrity partially by forming a column created by moisture-induced cohesive strengthening, while the dry sand collapses immediately.

captures fluid as bound moisture. This local increase in saturation triggers cohesive forces that significantly enhance the material's shear strength. Consequently, the block of partially wet sand maintains its structural integrity for a while, forming a vertical column which ends up collapsing.

5.3.2 *Fluid–sand simulation in complex domains.*

Sediment-laden flow and delta formation. In Fig. 9, we simulate a water–sand mixture flowing through an outlet and descending along an inclined slope in order to test sediment transport, deposition, and wetting effects. As the mixture accelerates along the downslope, the water phase moves quickly while the sand gradually settles due to reduced near-bed velocities. Along the inclined surface, the water spreads laterally and forms a thin wetting layer, giving rise to a distinct wetting boundary that precedes and surrounds the sand-rich core of the flow. Upon reaching the flatter region at the slope exit, the transported sediment is deposited and progressively forms a fan-shaped accumulation characteristic of a river delta. Continued inflow drives the growth of the delta front and induces evolving flow paths, with water diverting around deposited sand while maintaining visible wetting patterns on the slope.

Sediment-laden bunny dam break. We simulate a dam break scenario in Fig. 12 involving a water–sand mixture hitting a rigid bunny obstacle in order to evaluate the robustness of the proposed method under strongly coupled multiphase dynamics. After release, the mixture undergoes rapid acceleration, during which the heavier sand phase initially lags behind the water and subsequently forms a dense front near the bottom. As the flow evolves, the water phase entrains and lifts sand particles, generating vigorous mixing and complex turbulent structures within the mixture. Upon impact with the bunny, the water splashes upward and around the geometry, while the sand-rich flow exhibits pronounced inertial effects, piling up against the obstacle and flowing around it with reduced mobility. The interaction produces strong splashing, turbulent water-sand mixing, and localized accumulation of sand in concave regions.

Water-sand flow through a colander. To demonstrate the capability of our method to handle complex boundary conditions, we simulate the injection of a water–sand mixture ($\varrho = 1000\ \mathrm{kg\,m^{-3}}$) into a colander to induce multiphase separation, jetting, and deposition behavior in Fig. 4. Owing to the high water content within the sand phase, the mixture does not form cohesive sand clumps; instead, the sand remains sufficiently fluidized to flow together with the liquid and pass through the narrow gaps of the colander. As the mixture exits the perforations, multiple high-speed jets are generated, producing complex, yellow-tinted splashing indicative of strong water–sand mixing. After hitting the ground, the two phases exhibit markedly different behaviors: the water spreads laterally and fills the shallow pool, while the sand moves along with the fluid before settling across the bottom, creating a landscape of distributed sediment and heaps of different scales.

Water retention. To evaluate the physical efficacy of our retention strategy, we conduct a comparative experiment involving two sand-filled pipes subjected to water on one side, as illustrated in Fig. 10. The side water exerts hydraulic pressure on both blocks. The sand within the foreground pipe, where our retention model is *disabled* sees water rapidly infiltrating the porous medium, leading to immediate saturation, loss of stability, and erosion into a slurry. Conversely, the background pipe, whose sand uses our water retention formulation, has water penetrating, which turns into bound moisture in the sand. Crucially, this reduces the effective porosity $\hat{\epsilon}$ used in the pressure force calculations, which physically restricts the flow conductivity and delays the wetting front. Simultaneously, the retained moisture contributes to the local saturation state S_r, preserving the cohesive strength of the sand and allowing the block to withstand the hydraulic load significantly longer.

Seawater inundation of a desert city. A desert city is subjected to an incoming seawater surge, producing intense splashing and spray as the flow impacts the sandy ground and urban geometry in

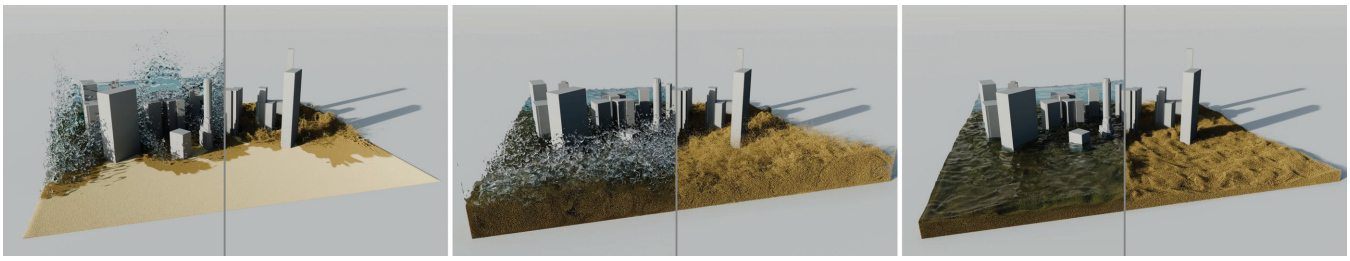

Fig. 18. **Seawater inundation of a desert city.** Seawater inundates a desert city, producing strong splashing as the flow impacts sandy terrain and urban structures. A water–sand mixture is formed through erosion and deposition, ultimately covering the city with water and sand accumulations. Each frame uses a split-screen comparison: the left side renders the full fluid-sand mixture, while the right side only renders sand to better show the final sand distribution.

Fig. 18. The water rapidly entrains and transports sand, generating a water-sand mixture that flows around buildings, accompanied by strong turbulent motion and localized erosion. As the flow evolves, sediment deposition and redistribution progressively reshape the terrain, and the city is transformed into a water-covered environment characterized by pools and newly formed sand accumulations, highlighting the rich coupled dynamics captured by our solver.

6 Conclusion

In this paper, we develop a coupled two-phase LBM-MPM framework for water-sand mixtures. Building on a unified mixture formulation that expresses both the fluid phase and granular medium within a consistent continuum setting, we derive an LBM-based kinetic two-phase flow solver and model sand using MPM. To obtain explicit fluid-sand momentum-exchange terms, we incorporate a drag force and derive a new pressure update that enforces incompressibility within the mixture, enabling a wide range of coupled behaviors. We further present a simple yet effective mass-conservation treatment for mixture dynamics by adding a source term to the phase-field equation, which may facilitate more complex coupling scenarios in future works — e.g., two-phase flow interacting with rods or cloth. Finally, we introduce a water absorption and retention strategy for sand to model capillary effects. Our method is highly amenable to GPU parallelization, achieving at least a 12× speedup over previous free-surface sand–water coupling approaches such as [Tang et al. 2025], while producing visually richer air-water-sand mixing phenomena.

Limitations and future work. Despite the wide range of results demonstrated in this paper, our method still has several limitations. Our current implementation has a substantial memory footprint, which limits the scale and resolution of practical simulations. In addition, the MPM-based sand solver typically requires a relatively small time step, which often becomes the computational bottleneck of the coupled system. A promising direction is to decouple the fluid and sand solvers and advance them with different time steps (e.g., using subcycling or multi-rate integration), or even explore coupling between LBM and implicit MPM, which could significantly improve overall efficiency. We point out that our current approach is unable to resolve interfacial details at high Reynolds numbers, reducing overall predictability at the macroscopic level. Future extensions may thus focus on integrating multi-resolution grids in the LBM solver and adaptive sampling within MPM to overcome this limitation. Similarly, our continuum-based sand model is inadequate at very small scales: the current formulation and our typical grid resolution cannot accurately capture grain-level dynamics. Finally, our diffuse-interface two-phase flow model does not prevent bubble disappearance; in our experiments, pressure filtering further exacerbates this issue at low resolutions. Furthermore, our global volume controller may negatively impact accuracy for turbulent flows, so we recommend disabling it for relative errors below 4%. In future work, coupling a sharp-interface free-surface formulation (e.g., HOME-FREE [Wang et al. 2025]) with an explicit bubble model and the sand solver could provide a principled remedy, enabling more robust bubble preservation while maintaining stable water–sand interactions. In addition, HOME-FREE is expected to reduce memory consumption by avoiding the need to explicitly simulate the surrounding air phase and its moment representation. Extending the framework to robustly handle water–sand mixtures interacting with dynamic rigid bodies and deformable objects (e.g., cloth or soft solids) is another promising direction for future work.

Acknowledgments

The castle mesh in Fig. 3 is the Christ Church 3D Model from the RigModels.com. The bunny mesh in Fig. 12 is from the Stanford Computer Graphics Laboratory. The beaker mesh in Fig. 13 is from the cgtrader. The city mesh in Fig. 18 is the New York Lower Manhattan from the Sketchfab. This work was supported by the Shanghai Pujiang Program (25PJD058) and SJTU's startup funds. MD acknowledges support of the MediTwin consortium (funded by the French government as part of France 2030), an Inria Chair, and of Synopsis. XY thanks the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and the Fundamental Research Funds for Central Universities.

References

Daniela Anderl, Simon Bogner, Cornelia Rauh, Ulrich Rüde, and Antonio Delgado. 2014. Free surface lattice Boltzmann with enhanced bubble model. *Computers & Mathematics with Applications* 67, 2 (2014), 331–339.

T Bo Anderson and ROY Jackson. 1967. Fluid mechanical description of fluidized beds. Equations of motion. *Ind. & Eng. Chem. Fund.* 6, 4 (1967), 527–539.

Ryoichi Ando, Nils Thuerey, and Chris Wojtan. 2015. A Stream Function Solver for Liquid Simulations. *ACM Trans. Graph.* 34, 4, Article 53 (2015).

Amir Banari, Christian Janßen, Stephan T. Grilli, and Manfred Krafczyk. 2014. Efficient GPGPU implementation of a lattice Boltzmann model for multiphase flows with high density ratios. *Computers and Fluids* 93 (2014), 1–17.

Vasili Baranau and Ulrich Tallarek. 2014. Random-close packing limits for monodisperse and polydisperse hard spheres. *Soft matter* 10, 21 (2014), 3826–3841.

Nathan Bell, Yizhou Yu, and Peter J Mucha. 2005. Particle-based simulation of granular materials. In *Symposium on Computer Animation*. 77–86.

Blender Online Community. 2018. *Blender.* Blender Fndn. http://www.blender.org

Simon Bogner. 2017. *Direct numerical simulation of liquid-gas-solid flows based on the lattice Boltzmann method.* Ph. D. Dissertation. Erlangen-Nürnberg University.

Table 1. **Parameters and timings.** We provide physical parameters and grid/particle configurations used in the paper. Note that the viscosity and the absorption coefficient R_a are given in dimensionless space, time steps in seconds, and timings represent the averaged time per frame in seconds.

Simulation Case	*Grid Resolution*	*Particle Number*	*Spacing* Δx (m)	*Water Retention/*R_a	*Controller Strength* η	*Kinematic Viscosity* ν_h	ν_l	*Sand Density* ϱ (kg m^{-3})	*Time Step* Δt (s)	*Time (s/frame)*
Fig. 1	$800 \times 320 \times 400$	5, 838k	0.0075	no	2	5×10^{-4}	1×10^{-1}	2000	1.0×10^{-4}	252.0
Fig. 2	$600 \times 180 \times 600$	6, 174k	0.001	no	2	5×10^{-4}	2×10^{-3}	600	5.8×10^{-5}	71.6
Fig. 3	$800 \times 400 \times 400$	11, 579k	0.0010	no	2	1×10^{-4}	1×10^{-2}	1400	2.9×10^{-5}	147.7
Fig. 4	$600 \times 500 \times 600$	1, 715k	0.00083	yes/0.003	2	1×10^{-4}	1×10^{-2}	1000	3.0×10^{-5}	288.3
Fig. 5	$600 \times 200 \times 600$	31, 879k	0.0025	no	1	1×10^{-4}	1×10^{-2}	1200	5.6×10^{-5}	82.6
Fig. 6	$300 \times 240 \times 300$	28, 800k	0.0001	no	1	1×10^{-4}	1×10^{-2}	1200	7.5×10^{-6}	4.9
Fig. 8	$300 \times 400 \times 300$	524k	0.0050	no	2	1×10^{-4}	1×10^{-2}	600/950/1500	1.6×10^{-5}	13.2
Fig. 9	$800 \times 400 \times 400$	52, 895k	0.005625	no	2	1×10^{-4}	1×10^{-2}	1200	1.0×10^{-4}	136.6
Fig. 10	$720 \times 400 \times 400$	10, 486k	0.0025	yes/0.001	2	1×10^{-4}	1×10^{-2}	1200	8.0×10^{-5}	226.6
Fig. 11	$300 \times 450 \times 300$	12, 960k	0.00050	no	1	1×10^{-4}	1×10^{-2}	1200	2.1×10^{-5}	93.3
Fig. 12	$800 \times 320 \times 400$	5, 289k	0.01	no	2	1×10^{-4}	1×10^{-1}	1200	6.0×10^{-5}	118.3
Fig. 13	$320 \times 500 \times 320$	524k	0.00050	no	2	5×10^{-4}	1×10^{-2}	1500	1.2×10^{-5}	52.3
Fig. 15	$1000 \times 400 \times 80$	9, 918k	0.002	no	2	1×10^{-4}	1×10^{-2}	900	3.4×10^{-5}	35.3
Fig. 14	$200 \times 400 \times 200$	14, 137k	0.0005	no	2	1×10^{-4}	1×10^{-2}	1200	1.5×10^{-5}	26.5
Fig. 16	$800 \times 400 \times 400$	27, 648k	0.01	no	1	1×10^{-4}	1×10^{-2}	900	6.5×10^{-5}	312.3
Fig. 17	$200 \times 300 \times 200$	4, 042k	0.002	yes/0.001	2	1×10^{-2}	1×10^{-2}	1200	1.2×10^{-4}	29.4
Fig. 18	$1000 \times 320 \times 800$	19, 971k	0.50	yes/0.003	2	3×10^{-4}	1×10^{-2}	1300	1.2×10^{-4}	286.8

Timna Böttcher, Lukas Westhofen, Stefan Rhys Jeske, and Jan Bender. 2025. Implicit Incompressible Porous Flow using SPH. *ACM Trans. Graph.* 44, 6 (2025), 1–13.

Landon Boyd and Robert Bridson. 2012. MultiFLIP for Energetic Two-Phase Fluid Simulation. *ACM Trans. Graph.* 31, 2, Article 16 (2012).

Jeremiah U. Brackbill, Douglas B. Kothe, and Charles Zemach. 1992. A continuum method for modeling surface tension. *J. Comp. Phys.* 100, 2 (1992), 335–354.

Bernhard Braun, Jan Bender, and Nils Thuerey. 2025. Adaptive Phase-Field-FLIP for Very Large Scale Two-Phase Fluid Simulation. *ACM Trans. Graph.* 44, 4 (2025), 1–23.

Duowen Chen, Zhiqi Li, Taiyuan Zhang, Jinjin He, Junwei Zhou, Bart G van Bloemen Waanders, and Bo Zhu. 2025a. Fluid simulation on compressible flow maps. *ACM Trans. Graph.* 44, 4 (2025), 1–17.

Xiao-Song Chen, Chen-Feng Li, Geng-Chen Cao, Yun-Tao Jiang, and Shi-Min Hu. 2020. A moving least square reproducing kernel particle method for unified multiphase continuum simulation. *ACM Trans. Graph.* 39, 6 (2020), 1–15.

Yixin Chen, Wei Li, Rui Fan, and Xiaopei Liu. 2022. GPU Optimization for High-Quality Kinetic Fluid Simulation. *IEEE Trans. Vis. Comp. Graph.* 28, 9 (2022), 3235–3251.

Yi-Lu Chen, Mickaël Ly, and Chris Wojtan. 2025b. Numerical Homogenization of Sand from Grain-level Simulations. *ACM Trans. Graph.* 44, 6, Article 220 (2025).

Junghyun Cho and Hyeong-Seok Ko. 2013. Geometry-Aware Volume-of-Fluid Method. *Comput. Graph. Forum* 32, 2 (2013), 379–388.

Peter Cundall and Otto Strack. 1979. The development of constitutive laws for soil using the distinct element method. *Num. Meth. Geomech.* (1979), 289–317.

Fernando de Goes, Corentin Wallez, Jin Huang, Dmitry Pavlov, and Mathieu Desbrun. 2015. Power Particles: An Incompressible Fluid Solver Based on Power Diagrams. *ACM Trans. Graph.* 34, 4, Article 50 (2015).

Yitong Deng, Mengdi Wang, Xiangxin Kong, Shiying Xiong, Zangyueyang Xian, and Bo Zhu. 2022. A moving eulerian-lagrangian particle method for thin film and foam simulation. *ACM Trans. Graph.* 41, 4 (2022).

Lin Ding and Xiaopei Liu. 2025. A Hybrid LBM-FVM Solver for Simulating Multiphase Multi-Component Fluids. *IEEE Trans. Vis. Comp. Graph.* (2025).

Wen-Tao Ding and Wen-Jie Xu. 2018. Study on the multiphase fluid-solid interaction in granular materials based on an LBM-DEM coupled method. *Powder Technology* 335 (2018), 301–314.

Abbas Fakhari, Diogo Bolster, and Li-Shi Luo. 2017. A weighted multiple-relaxation-time lattice Boltzmann method for multiphase flows and its application to partial coalescence cascades. *J. Comp. Phys.* 341 (2017), 22–43.

Linlin Fei, Feifei Qin, Geng Wang, Jingwei Huang, Binghai Wen, Jianlin Zhao, Kai H Luo, Dominique Derome, and Jan Carmeliet. 2023. Coupled lattice Boltzmann method–discrete element method model for gas–liquid–solid interaction problems. *J. Fluid Mech.* 975 (2023), A20.

YT Feng, K Han, and DRJ Owen. 2007. Coupled lattice Boltzmann method and discrete element modelling of particle transport in turbulent fluid flows: Computational issues. *Int. J. Numer. Methods Eng.* 72, 9 (2007), 1111–1134.

Ming Gao, Andre Pradhana, Xuchen Han, Qi Guo, Grant Kot, Eftychios Sifakis, and Chenfanfu Jiang. 2018. Animating fluid sediment mixture in particle-laden flows. *ACM Trans. Graph.* 37, 4 (2018), 1–11.

Martin Geier, Martin Schönherr, Andrea Pasquali, and Manfred Krafczyk. 2015. The Cumulant Lattice Boltzmann Equation in Three Dimensions: theory and validation. *Comput. Math. Appl.* 70, 4 (2015), 507–547.

Ryan Goldade, Mridul Aanjaneya, and Christopher Batty. 2020. Constraint bubbles and affine regions: reduced fluid models for efficient immersed bubbles and flexible spatial coarsening. *ACM Trans. Graph.* 39, 4 (2020), 43–1.

Yulong Guo, Xiaopei Liu, and Xuemiao Xu. 2017. A Unified Detail-Preserving Liquid Simulation by Two-Phase Lattice Boltzmann Modeling. *IEEE Trans. Vis. Comp. Graph.* 23, 5 (2017), 1479–1491.

Xiaoyi He, Shiyi Chen, and Raoyang Zhang. 1999. A Lattice Boltzmann Scheme for Incompressible Multiphase Flow and Its Application in Simulation of Rayleigh–Taylor Instability. *J. Comp. Phys.* 152, 2 (1999), 642–663.

Jeong-Mo Hong and Chang-Hun Kim. 2005. Discontinuous Fluids. *ACM Trans. Graph.* 24, 3 (2005), 915–920.

Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. https://libigl.github.io/.

David Jacqmin. 1999. Calculation of Two-Phase Navier-Stokes Flows Using Phase-Field Modeling. *J. Comp. Phys.* 155, 1 (1999), 96–127.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The Affine Particle-in-Cell Method. *ACM Trans. Graph.* 34, 4, Article 51 (2015).

Fei Jiang, Haihu Liu, Xian Chen, and Takeshi Tsuji. 2022. A coupled LBM-DEM method for simulating the multiphase fluid-solid interaction problem. *J. Comp. Phys.* 454 (2022), 110963.

Yuntao Jiang, Chenfeng Li, Shujie Deng, and Shi-Min Hu. 2020. A Divergence-free Mixture Model for Multiphase Fluids. In *Comp. Graph. Forum*, Vol. 39. 69–77.

Alexandre Kane. 2025. Granular collapse experiments in water. https://www.youtube.com/watch?v=vcDFZk3vvqw YouTube video.

Myungjoo Kang, Ronald P Fedkiw, and Xu-Dong Liu. 2000. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing* 15, 3 (2000), 323–360.

Yuki Kano, Toru Sato, and Hiroyuki Oyama. 2020. Numerical study on the formations of gas channels and subsequent bubbles in unconsolidated sandy seabed sediment using a coupled LBM-DEM method. *J. Nat. Gas Eng.* 74 (2020), 103101.

Petr Karnakov, Sergey Litvinov, and Petros Koumoutsakos. 2022. Computing foaming flows across scales: From breaking waves to microfluidics. *Science Advances* 8, 5 (2022), eabm0590.

Byungmoon Kim. 2010. Multi-phase fluid simulations using regional level sets. *ACM Trans. Graph.* 29, 6 (2010), 1–8.

Byungmoon Kim, Yingjie Liu, Ignacio Llamas, Xiangmin Jiao, and Jarek Rossignac. 2007. Simulation of bubbles in foam with the volume control method. *ACM Trans. Graph.* 26, 3, Article 98 (2007).

Seung Hyun Kim and Heinz Pitsch. 2015. On the lattice Boltzmann method for multiphase flows with large density ratios. *J. Comp. Phys.* 303 (2015), 19–27.

Gergely Klár, Theodore Gast, Andre Pradhana, Chuyuan Fu, Craig Schroeder, Chenfanfu Jiang, and Joseph Teran. 2016. Drucker-prager elastoplasticity for sand animation. *ACM Trans. Graph.* 35, 4 (2016), 1–12.

Carolin Körner, Michael Thies, Torsten Hofmann, Nils Thürey, and Ulrich Rüde. 2005. Lattice Boltzmann model for free surface flow for modeling foaming. *Journal of Statistical Physics* 121 (2005), 179–196.

Harald Kruggel-Emden, Erdem Simsek, Stefan Rickelt, Siegmar Wirtz, and Viktor Scherer. 2007. Review and extension of normal force models for the discrete element method. *Powder Technology* 171, 3 (2007), 157–173.

Tassilo Kugelstadt, Andreas Longva, Nils Thuerey, and Jan Bender. 2019. Implicit density projection for volume conserving liquids. *IEEE Trans. Vis. Comp. Graph.* 27, 4 (2019), 2385–2395.

Timothy R Langlois, Changxi Zheng, and Doug L James. 2016. Toward animating water with complex acoustic bubbles. *ACM Trans. Graph.* 35, 4 (2016).

Jun Li. 2015. Chapman-Enskog expansion in the lattice Boltzmann method. *arXiv preprint arXiv:1512.02599* (2015).

Ruixin Li, Linlin Fei, Kai H Luo, Hong Liang, Xitong Zhang, Dominique Derome, and Jan Carmeliet. 2025a. Coupled phase-field lattice Boltzmann method and discrete element method for gas–liquid–solid multiphase flows. *Physics of Fluids* 37, 5 (2025).

Ruolan Li, Yanrui Xu, Yalan Zhang, Jiri Kosinka, Alexandru C Telea, Jian Chang, Jian Jun Zhang, Xiaojuan Ban, and Xiaokun Wang. 2025b. Multiphase Particle-Based Simulation of Poro-Elasto-Capillary Effects. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*. 1–11.

Shuo Li, Shintaro Kajiwara, and Mikio Sakai. 2021a. Numerical investigation on the mixing mechanism in a cross-torus paddle mixer using the DEM-CFD method. *Powder Technology* 377 (2021), 89–102.

Wei Li, Yixin Chen, Mathieu Desbrun, Changxi Zheng, and Xiaopei Liu. 2020. Fast and scalable turbulent flow simulation with two-way coupling. *ACM Trans. Graph.* 39, 4, Article 47 (2020).

Wei Li and Mathieu Desbrun. 2023. Fluid-Solid Coupling in Kinetic Two-Phase Flow Simulation. *ACM Trans. Graph.* 42, 4, Article 123 (2023).

Wei Li, Daoming Liu, Mathieu Desbrun, Jin Huang, and Xiaopei Liu. 2021b. Kinetic-Based Multiphase Flow Simulation. *IEEE Trans. Vis. Comp. Graph.* 27, 7 (2021), 3318–3334.

Wei Li, Yihui Ma, Xiaopei Liu, and Mathieu Desbrun. 2022. Efficient Kinetic Simulation of Two-Phase Flows. *ACM Trans. Graph.* 41, 4, Article 114 (2022).

Wei Li, Kui Wu, and Mathieu Desbrun. 2024b. Kinetic Simulation of Turbulent Multifluid Flows. *ACM Trans. Graph.* 43, 4, Article 55 (2024).

Zhiqi Li, Barnabás Börcsök, Duowen Chen, Yutong Sun, Bo Zhu, and Greg Turk. 2024a. Lagrangian Covector Fluid with Free Surface. In *ACM SIGGRAPH 2024 Conference Papers*. Article 43.

Yu Liu, Hongfei Ye, Hongwu Zhang, and Yonggang Zheng. 2020. Coupling lattice Boltzmann and material point method for fluid-solid interaction problems involving massive deformation. *Int. J. Numer. Methods Eng.* 121, 24 (2020), 5546–5567.

Yu Liu, Hongfei Ye, Hongwu Zhang, and Yonggang Zheng. 2023. Coupling Phase-Field LB–MP Method for Multiphase Fluid–Deformable Solid Interaction Problems Involving Large Density and Viscosity Contrasts. *Int. J. Appl. Mech.* 15, 06 (2023), 2350050.

Franck Lominé, Luc Scholtes, Luc Sibille, and Philippe Poullain. 2013. Modeling of fluid–solid interaction in granular media with coupled lattice Boltzmann/discrete element methods: application to piping erosion. *Int. J. Numer. Anal. Methods Geomech.* 37, 6 (2013), 577–596.

Frank Losasso, Tamar Shinar, Andrew Selle, and Ronald Fedkiw. 2006. Multiple interacting liquids. *ACM Trans. Graph.* 25, 3 (2006), 812–819.

Yihui Ma, Xiaoyu Xiao, Wei Li, Mathieu Desbrun, and Xiaopei Liu. 2024. Hybrid LBM-FVM solver for two-phase flow simulation. *J. Comput. Phys.* 506 (2024), 112920.

Mikko Manninen, Veikko Taivassalo, and Sirpa Kallio. 1996. *On the mixture model for multiphase flow.* Technical Report 288. VTT Technical Research Centre of Finland.

V. Mihalef, B. Unlusu, D. Metaxas, M. Sussman, and M. Y. Hussaini. 2006. Physics based boiling simulation. In *Symposium on Computer Animation*. 317–324.

Marek Krzysztof Misztal, Kenny Erleben, Adam Bargteil, Jens Fursund, Brian Bunch Christensen, Jakob Andreas Bærentzen, and Robert Bridson. 2013. Multiphase flow of immiscible fluids on unstructured moving meshes. *IEEE Trans. Vis. & Comp. Graph.* 20, 1 (2013), 4–16.

Rahul Narain, Abhinav Golas, and Ming C. Lin. 2010. Free-flowing granular materials with two-way solid coupling. In *ACM SIGGRAPH Asia*. Article 173.

Michael B Nielsen and Ole Østerby. 2013. A two-continua approach to Eulerian simulation of water spray. *ACM Trans. Graph.* 32, 4 (2013), 1–10.

Ziyin Qu, Minchen Li, Fernando De Goes, and Chenfanfu Jiang. 2022. The power particle-in-cell method. *ACM Trans. Graph.* 41, 4 (2022), 1–13.

Ziyin Qu, Minchen Li, Yin Yang, Chenfanfu Jiang, and Fernando De Goes. 2023. Power Plastics: A Hybrid Lagrangian/Eulerian Solver for Mesoscale Inelastic Flows. *ACM Trans. Graph.* 42, 6 (2023).

Pouya Ranjbari, Mohammadreza Ebrahimi, Farhad Ein-Mozaffari, Simant Upreti, and Ali Lohi. 2025. A critical review of the coupled CFD-DEM method for the simulation of two-phase liquid-solid systems. *Powder Technology* (2025), 120677.

Bo Ren, Chenfeng Li, Xiao Yan, Ming C Lin, Javier Bonet, and Shi-Min Hu. 2014. Multiple-fluid SPH simulation using a mixture model. *ACM Trans. Graph.* 33, 5 (2014), 1–11.

Christoph Rettinger and Ulrich Rüde. 2017. A comparative study of fluid-particle coupling methods for fully resolved lattice Boltzmann simulations. *Computers & Fluids* 154 (2017), 74–89.

Christoph Rettinger and Ulrich Rüde. 2022. An efficient four-way coupled lattice Boltzmann–discrete element method for fully resolved simulations of particle-laden flows. *J. Comp. Phys.* 453 (2022), 110942.

Manon Robbe-Saule, Cyprien Morize, Robin Henaff, Yann Bertho, Alban Sauret, and Philippe Gondret. 2021. Experimental investigation of tsunami waves generated by granular collapse into water. *J. Fluid Mech.* 907 (2021), A11.

Witawat Rungjiratananon, Zoltan Szego, Yoshihiro Kanamori, and Tomoyuki Nishita. 2008. Real-time animation of sand-water interaction. In *Comp. Graph. Forum*, Vol. 27. Wiley Online Library, 1887–1893.

Robert Saye. 2016. Interfacial gauge methods for incompressible fluid dynamics. *Science Advances* 2, 6 (2016), e1501869.

Robert Saye. 2017. Implicit mesh discontinuous Galerkin methods and interfacial gauge methods for high-order accurate interface dynamics, with applications to surface tension dynamics, rigid body fluid–structure interaction, and free surface flow: Part I. *J. Comp. Phys.* 344 (2017), 647–682.

Yos Panagaman Sitompul and Takayuki Aoki. 2019. A filtered cumulant lattice Boltzmann method for violent two-phase flows. *J. Comp. Phys.* 390 (2019), 93–120.

B. Solenthaler and R. Pajarola. 2008. Density contrast SPH interfaces. In *Symposium on Computer Animation*. 211–218.

Haozhe Su, Tao Xue, Chengguizi Han, Chenfanfu Jiang, and Mridul Aanjaneya. 2021. A unified second-order accurate in time MPM formulation for simulating viscoelastic liquids with phase change. *ACM Trans. Graph.* 40, 4 (2021), 1–18.

Haozhe Su, Siyu Zhang, Zherong Pan, Mridul Aanjaneya, Xifeng Gao, and Kui Wu. 2023. Real-time Height-field Simulation of Sand and Water Mixtures. In *SIGGRAPH Asia 2023 Conference Papers*. 1–10.

Xiaosong Sun and Mikio Sakai. 2015. Three-dimensional simulation of gas–solid–liquid flows using the DEM–VOF method. *Chemical Engineering Science* 134 (2015), 531–548.

Yuchen Sun, Linglai Chen, Weiyuan Zeng, Tao Du, Shiying Xiong, and Bo Zhu. 2024. An Impulse Ghost Fluid Method for Simulating Two-Phase Flows. *ACM Trans. Graph.* 43, 6 (2024).

Andre Pradhana Tampubolon, Theodore Gast, Gergely Klár, Chuyuan Fu, Joseph Teran, Chenfanfu Jiang, and Ken Museth. 2017. Multi-species simulation of porous sand and water mixtures. *ACM Trans. Graph.* 36, 4 (2017), 1–11.

Yizao Tang, Yuechen Zhu, Xingyu Ni, and Baoquan Chen. 2025. The Granule-In-Cell Method for Simulating Sand–Water Mixtures. *ACM Trans. Graph.* 44, 6, Article 266 (2025).

N. Thürey, F. Sadlo, S. Schirm, M. Müller-Fischer, and M. Gross. 2007. Real-time simulations of bubbles and foam within a shallow water framework. In *Symposium on Computer Animation*. 191–198.

Zaili Tu, Chen Li, Zipeng Zhao, Long Liu, Chenhui Wang, Changbo Wang, and Hong Qin. 2024. A Unified MPM Framework Supporting Phase-field Models and Elastic-viscoplastic Phase Transition. *ACM Trans. Graph.* 43, 2 (2024).

Hui Wang, Zhi Wang, Shulin Hong, Xubo Yang, and Bo Zhu. 2024b. A Moving Least-Squares/Level-Set Particle Method for Bubble and Foam Simulation. *IEEE Trans. Vis. Comp. Graph.* (2024), 1–15.

Haoxiang Wang, Kui Wu, Hui Qiao, Mathieu Desbrun, and Wei Li. 2025. Kinetic Free-Surface Flows and Foams with Sharp Interfaces. *ACM Trans. Graph.* 44, 6, Article 267 (2025).

Min Wang, YT Feng, DRJ Owen, and TM Qu. 2019. A novel algorithm of immersed moving boundary scheme for fluid–particle interactions in DEM–LBM. *Computer Methods in Applied Mechanics and Engineering* 346 (2019), 109–125.

Tuo Wang, Shihang Chen, Mengli Li, and Mengke An. 2024a. A resolved CFD-DEM investigation into sand production under water flooding in unconsolidated reservoir. *Powder Technology* 442 (2024), 119859.

Xu Wang, Makoto Fujisawa, and Masahiko Mikawa. 2021. Visual simulation of soil-structure destruction with seepage flows. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 4, 3 (2021), 1–18.

Heng Xiao and Jin Sun. 2011. Algorithms in a robust hybrid CFD-DEM solver for particle-laden flows. *Comm. Comp. Phys.* 9, 2 (2011), 297–323.

Xiaoyu Xiao, Ding Lin, Yiheng Wu, Kai Bai, and Xiaopei Liu. 2025. Simulating Two-phase Fluid-rigid Interactions with an Overset-Grid Kinetic Solver. *IEEE Trans. Vis. Comp. Graph.* 31, 10 (2025), 8397–8412.

Tianyi Xie, Minchen Li, Yin Yang, and Chenfanfu Jiang. 2023. A contact proxy splitting method for Lagrangian solid-fluid coupling. *ACM Trans. Graph.* 42, 4 (2023), 1–14.

Zhewen Xiong and Haishan Cao. 2025. Phase-field-based lattice Boltzmann modeling of thermal phase change initiated from a single phase. *Physics of Fluids* 37, 3 (2025).

Han Yan and Bo Ren. 2023. High Density Ratio Multi-Fluid Simulation with Peridynamics. *ACM Trans. Graph.* 42, 6 (2023), 1–14.

Xiao Yan, Yun-Tao Jiang, Chen-Feng Li, Ralph R Martin, and Shi-Min Hu. 2016. Multiphase SPH simulation for interactive fluids and solids. *ACM Trans. Graph.* 35, 4 (2016), 1–11.

Tao Yang, Jian Chang, Ming C Lin, Ralph R Martin, Jian J Zhang, and Shi-Min Hu. 2017. A unified particle system framework for multi-phase, multi-material visual simulations. *ACM Trans. Graph.* 36, 6 (2017), 1–13.

Tao Yang, Jian Chang, Bo Ren, Ming C Lin, Jian Jun Zhang, and Shi-Min Hu. 2015. Fast multiple-fluid simulation using Helmholtz free energy. *ACM Trans. Graph.* 34, 6 (2015), 1–11.

Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, and Eitan Grinspun. 2015. Continuum foam: A material point method for shear-dependent flows. *ACM Trans. Graph.* 34, 5 (2015), 1–20.

Yonghao Yue, Breannan Smith, Peter Yichen Chen, Maytee Chantharayukhonthorn, Ken Kamrin, and Eitan Grinspun. 2018. Hybrid grains: Adaptive coupling of discrete and continuum simulations of granular media. *ACM Trans. Graph.* 37, 6 (2018), 1–19.

HP Zhu and AB Yu. 2002. Averaging method of granular materials. *Physical Review E* 66, 2 (2002), 021302.

Yongning Zhu and Robert Bridson. 2005. Animating sand as a fluid. *ACM Trans. Graph.* 24, 3 (2005), 965–972.