Animating Fluid Sediment Mixture in Particle-Laden Flows

MING GAO, University of Wisconsin, Madison
ANDRE PRADHANA, University of Pennsylvania
XUCHEN HAN, University of California, Los Angeles
QI GUO, University of California, Los Angeles
GRANT KOT, Phosphorus
EFTYCHIOS SIFAKIS, University of Wisconsin, Madison
CHENFANFU JIANG, University of Pennsylvania

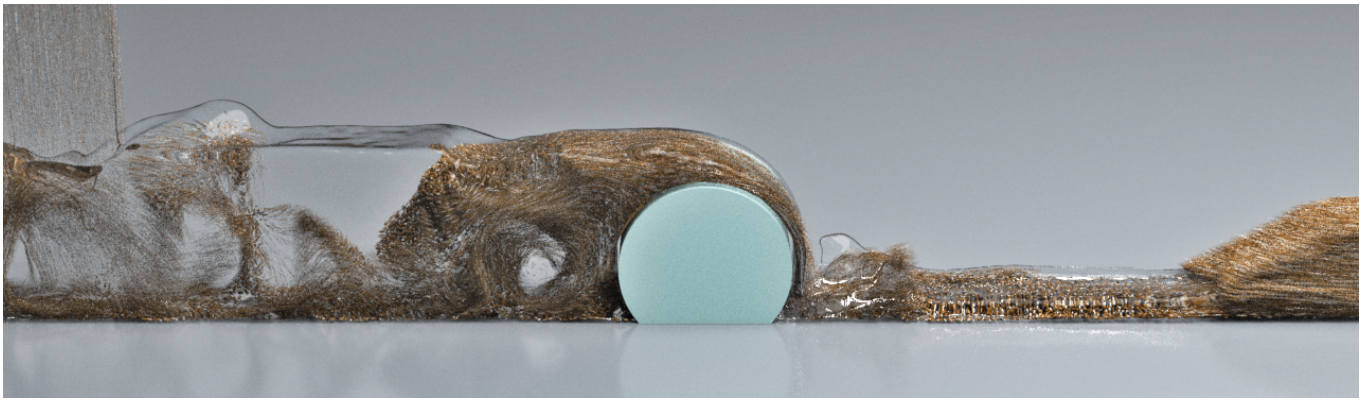

Fig. 1. **Sediment transport:** Our method can animate intricate two-way coupled particle-laden flows such as sediment transport in liquid.

In this paper, we present a mixed explicit and semi-implicit Material Point Method for simulating particle-laden flows. We develop a Multigrid Preconditioned fluid solver for the Locally Averaged Navier Stokes equation. This is discretized purely on a semi-staggered standard MPM grid. Sedimentation is modeled with the Drucker-Prager elastoplasticity flow rule, enhanced by a novel particle density estimation method for converting particles between representations of either continuum or discrete points. Fluid and sediment are two-way coupled through a momentum exchange force that can be easily resolved with two MPM background grids. We present various results to demonstrate the efficacy of our method.

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

Additional Key Words and Phrases: Material Point Method (MPM), particle-fluid interaction, multiphase, sedimentation, sediment transport

ACM Reference format:
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, Article 145 (August 2018), 11 pages.
DOI: 10.1145/3197517.3201309

DOI: 10.1145/3197517.3201309

1 INTRODUCTION

Recently, multi-phase multi-material simulations are increasingly gaining attention from computer graphics researchers. Simulating various phases or materials in a unified framework is particularly favored. Existing work includes coupled Lagrangian particle simulation with Position Based Dynamics (PBD) [Macklin et al. 2014], water-gas mixtures [Nielsen and Østerby 2013] with an Eulerian method, solid-fluid phase-change [Stomakhin et al. 2014] and porous granular media [Pradhana-Tampubolon et al. 2017] with Material Point Method (MPM), as well as interactive solids and fluids based on the mixture model with Smoothed Particle Hydrodynamics (SPH) [Yan et al. 2016].

Most of the existing approaches are based on *continuum* mixture theory [Manninen et al. 1996]. The continuum assumption for each material phase is essential for simulations of macroscopic porous media (e.g., landslides and liquid blending). However, it may fail to capture the correct behavior of particle-laden flows where the solid phase is on a relatively small scale. Note that particle-laden sediment flow is ubiquitous in natural systems. Typical examples include sediment transport, sedimentation, volcano eruption, dune migration by erosion with ripples, and dust storms. The significance of understanding and simulating these phenomena is also recognized in many engineering applications, such as granular material fluidization [van der Hoef et al. 2006] and coastal erosion prediction [Sun and Xiao 2016a].

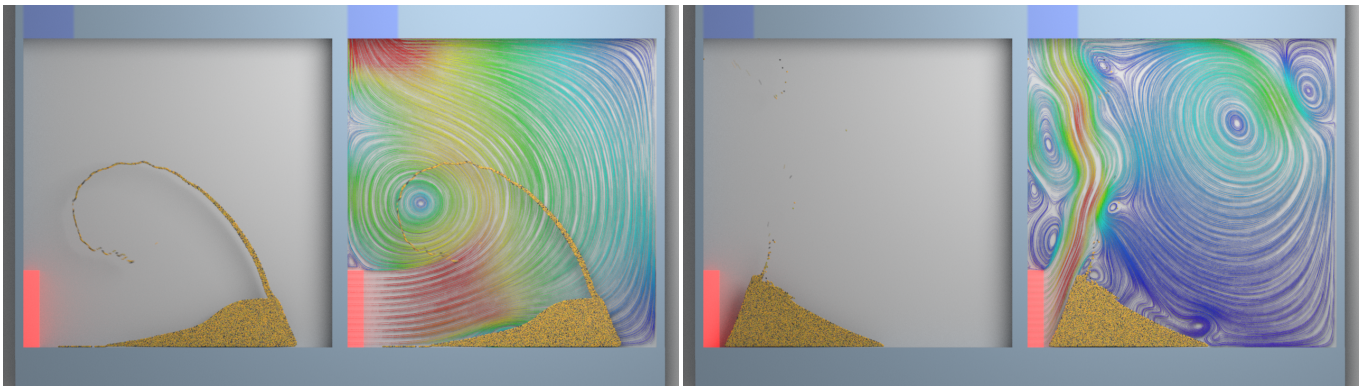

Fig. 2. **Dune migration:** Two-dimensional simulations of wind blowing sand. *Left:* A one-way coupled simulation where sand does not affect wind fails to produce plausible dynamics. *Right:* Two-way coupled simulation from our method captures the characteristic behavior of a sand dune migrating forward.

Our method treats fluid as a continuum and solid sediment as being comprised of small immiscible particles. The majority of engineering literature discretizes these sedimentation problems based on a combination of an Eulerian fluid solver and a Discrete Element Method (DEM) particle solver, which requires velocity interpolation between the two different discretizations to handle the momentum exchange between them. DEM is also too expensive for practical animation in collision-heavy dense flow scenarios. Our framework employs the MPM discretization for both the fluid phase and the solid phase to mitigate these problems.

In standard DEM, each particle is associated with a rigid sphere, and thus trivially represents a discrete spherical body with translational and rotational motions. Even for a collection of many particles, DEM does not make any continuum assumption. In contrast, each MPM particle represents a patch of the continuum and serves as a quadrature point for the spatial discretization of the continuous fields. Furthermore, traditional MPM cannot preserve angular momentum. In this paper, we tackle both difficulties. We use the Affine Particle-In-Cell (APIC) [Jiang et al. 2015] method to grant each particle an affine velocity field which enables the representation of local rigid rotations. Second, we propose a density criterion to allow certain MPM particles to behave as discrete debris separated from the continuum body. As a result, each MPM particle in the sediment can essentially act as a separate spherical body, while the continuum elastoplastic relationship is only activated for clumps of them.

In contrast to the recent work on wet sand by Pradhana et al. [2017], which relied on an intricate modification to the sand elastoplasticity model and is limited in the use of weakly compressible fluids, our work does not require an ad-hoc modification to the sand constitutive model and supports an incompressible fluid solver. Unlike the augmented MPM scheme by Stomakhin et al. [2014], our incompressible fluid solver discretizes velocity degree of freedoms on a semi-staggered grid, thus eliminating the computationally costly MPM discretization on a MAC grid which requires the use of cubic kernels. Xiao and Sun [2011] used constant interpolation to approximate the fluid velocities at solid particle positions, introducing dissipation; instead, our framework circumvents such interpolation.

We summarize our contributions as follows:

- A unified solver for purely incompressible fluids and particle-laden-flows using MPM background grids.
- A sub-stepping scheme with accurate momentum exchange for the mixture of solid particles and fluids, allowing for a larger time-step in the fluid solver.
- A density evaluation strategy for converting particles between continuum clumps and discrete debris.
- A semi-staggered semi-implicit MPM discretization for Locally Averaged Navier-Stokes with a variable coefficient multigrid preconditioner.

We demonstrate the efficacy of our method with various sediment-fluid interaction examples.

2 RELATED WORK

While earlier work in graphics was centered on certain types of physical materials or phenomena, more recent attention has been focused on the simulation of multi-phase multi-material interactions. Teng et al. [2016] simulate two-way coupled deformable solids and incompressible flow. Nielsen and Østerby [2013] used a two-continua mixture model for simulating water droplets in the air, discretized in an Eulerian fashion. Macklin et al. [2014] proposed a unified Lagrangian particle based solid-liquid-gas coupling framework with PBD [Müller et al. 2007]. Müller et al. [2005] were the first to simulate fluid-fluid interaction in SPH. Peer et al. [2015] simulated multiphase viscous SPH fluid with pressure and cohesion forces for their interaction. Ren et al. [2014] simulated both miscible and immiscible fluids with a mixture model. Yang et al. [2015] further took a Helmholtz Free Energy based model to achieve improved performance. These SPH frameworks were extended in Yan et al. [2016] to solids and granular materials, and also by Yang et al. [2017] to a variety of fluid-solid interactions. Daviet simulated granular flow in Newtonian fluid [2016] in two dimensions.

The existance of solid-fluid mixture is also common in porous flow and solid wetting. Lenaerts et al. [2008] simulated fluids in porous deformable media with an Eulerian discretization of Darcy flux. Their approach was extended to mixing fluids and granular materials in [Lenaerts and Dutré 2009]. Rungjiratananon et al. [2008] simulated wet sand with coupled SPH and DEM. They also captured

capillary flow in anisotropic permeable hair in [Rungjiratananon et al. 2012]. Fei et al. [2017] adopted a multi-scale model for wet hair, where they used Affine Particle-In-Cell (APIC) [Jiang et al. 2015] for water and a discrete rod model [Bergou et al. 2008] for the solid. Liquid is also simulated with APIC in our framework, although we will discretize it on a semi-staggered grid.

In computer graphics, Wojtan et al. [2007] simulated erosion, sedimentation, and corrosion with Eulerian fluids and level set solids. Krištof et al. [2009] used an SPH-based fluid scheme to model erosion and avoided the direct simulation of sediments by introducing the boundary particles to represent the sediment exchange with the terrain.

Our method is based on the Material Point Method (MPM) [Sulsky et al. 1995], which has recently been popularized in computer graphics for simulating granular materials [Daviet and Bertails-Descoubes 2016; Klár et al. 2016; Stomakhin et al. 2013], viscoplasticity [Ram et al. 2015; Yue et al. 2015] and cloth [Jiang et al. 2017]. MPM was also used for simulating solid-incompressible fluid coupling in [Stomakhin et al. 2014] and porous sand-water mixture in [Pradhana-Tampubolon et al. 2017]. We compare the feature sets of these methods in Table 1.

There is a large body of related work in Engineering literature on simulating particle-laden flows. One of the standard approaches is to couple mesh-based Eulerian fluids with particle systems described with the discrete element method (DEM). The pioneering work of Xu and Yu [1997] modeled gas-solid flow in a fluidized bed by combining the Discrete Particle Method (DPM) and standard Computational Fluid Dynamics (CFD) solvers. Li et al. [1999] extended it to gas–liquid–solid systems. Xiao and Sun [2011] applied a similar idea to general particle-laden flows and recently released a general-purpose, open-source CFD-DEM coupling solver called SediFoam [2016b]. They treated the fluid-particle interaction term implicitly and this resulted in a highly stable integration scheme. Snider [2001] considered dense particle fluid flows with multiphase Particle-In-Cell (MPIC). Fluid-particle interaction was also achieved with coupled SPH and DEM by Robinson et al. [2011]. More recently Mutabaruka and Kamrin [2017] simulated particle-laden dynamics with lattice Boltzmann fluids and DEM. Narain et al. [2010] and Daviet and Bertails-Descoubes [2016] investigated free-flowing granular media as disperse grains in low density regions. Our strategy for treating debris sediment is similar to the approach by Sachith and Kamrin [2015]. Their approach associates a unilateral pressure with each particle by comparing their density with a threshold and enforcing a *disconnected* stress-free state when the density is low. The main difference is that they evolve material point volume using velocity gradient over time, while we always evaluate the current particle "clumpiness" based on world space locations which avoids accumulated numerical advection error in highly dynamic scenarios.

	AMPM[2014]	MMPM[2017]	Ours
Reducible to incompressible flow (fluid)	✓	✗	✓
Unmodified hyperelasticity (solid)	✓	✗	✓
Discrete debris support (solid)	✗	✗	✓
Saturation dependent cohesion (solid)	✗	✓	✗
Multi-phase miscibility	✗	✓	✓
Mixed integration	✗	✗	✓

Table 1. Comparing Augmented MPM (AMPM) [Stomakhin et al. 2014], Multi-species MPM (MMPM) [Pradhana-Tampubolon et al. 2017] and our method for simulating the mixture of granular solid particles and fluids.

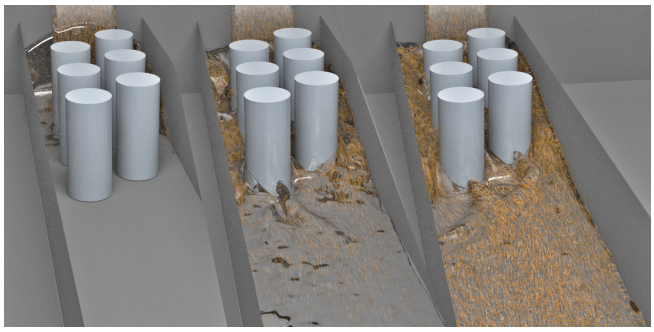

Fig. 3. **Debris flow:** We simulate water-saturated debris flow, which is a common occurrence of particle-laden flows in nature.

3 CONTINUOUS EQUATIONS

Here we describe the governing equations for both fluid and sediment materials in particle-laden flows. We will use superscript f for fluid and s for sediment. Fluid and sediment velocities are denoted with $\mathbf{u}$ and $\mathbf{v}$ respectively.

3.1 Fluid

In particle-laden flows, the fluid has to satisfy the impermeable boundary conditions enforced by discrete particles. However, it is impossible to exactly resolve these boundaries in the presence of a large number of sediment particles, unless an extremely high fluid resolution is used. We instead incorporate the Locally Averaged incompressible Navier-Stokes theory proposed by Anderson and Jackson [1967] to model the fluid motion. As in other previous work on multiphase simulation, each spatial point is simultaneously occupied by different phases or materials. Note that the use of Locally Averaged Navier-Stokes in our work essentially shares the same form for describing the fluid with the mixture model used by Nielsen and Østerby [2013]. Specifically, the mass equation for fluid is given by

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

and the momentum equation is

$$\frac{\partial(\epsilon\rho^f\mathbf{u})}{\partial t} + \nabla\cdot(\epsilon\rho^f\mathbf{u}\otimes\mathbf{u}) = -\epsilon\nabla p + \epsilon\rho^f\mathbf{g} + \mathbf{f}^{fd}, \tag{2}$$

where ϵ, ρ^f, $\mathbf{u}$, p, $\mathbf{g}$, and $\mathbf{f}^{fd}$ are the fluid volume fraction, fluid *intrinsic* density, fluid velocity, pressure, the gravitational constant, and the fluid drag force density respectively. We use the superscript fd, sd and se for fluid drag force, sediment drag force and sediment elastic force respectively. Note that $\mathbf{f}^{fd}$ and $\mathbf{f}^{sd}$ denote the interaction drag *force of the sediment on the fluid* and *force of the fluid on the sediment*. As such, they obey $\mathbf{f}^{fd} = -\mathbf{f}^{sd}$. Combining Eq. (1)

and Eq. (2), we arrive at

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho^f}\nabla p + \mathbf{g} + \frac{1}{\epsilon\rho^f}\mathbf{f}^{fd}. \tag{3}$$

3.2 Sediment

In the scale of particle-laden flows, granular solid materials are usually treated as discrete spherical bodies. This perspective eliminates the need to formulate governing equations for the volume fraction scaled solid material or for the entire mixture as in [Yan et al. 2016]. Normally in a CFD-DEM coupling framework such as [Sun and Xiao 2016b], Newton's second law would be directly enforced on the discrete spherical particles to describe their translational and rotational motions. However, in order to support the massive amount of dense particle suspensions, it is preferable to formulate the sediment governing equations through continuum mechanics, with the extra drag force density term $\mathbf{f}^{sd}$:

$$\frac{\partial(\delta\rho^s)}{\partial t} + \nabla\cdot(\delta\rho^s\mathbf{v}) = 0, \tag{4}$$

$$\frac{\partial(\delta\rho^s\mathbf{v})}{\partial t} + \nabla\cdot(\delta\rho^s\mathbf{v}\otimes\mathbf{v}) = \delta\nabla\cdot\boldsymbol{\sigma}^s + \delta\rho^s\mathbf{g} + \mathbf{f}^{sd}, \tag{5}$$

where δ, ρ^s, $\mathbf{v}^s$, and $\boldsymbol{\sigma}^s$ denote sediment volume fraction, sediment *intrinsic* density, sediment velocity, and the Cauchy stress describing the mechanical responses inside solid clumps. From mixture theory, the volume fraction of fluid and sediment obeys $\epsilon + \delta = 1$. In the case of debris in the air without liquid, we have $\delta = 1$ and $\epsilon = 0$.

As previously mentioned, we treat a clump of sediment particles as an elastoplastic material which obeys Drucker-Prager plasticity. Our density approximation method provides a measure of the spatial distribution of sediment particles. As a sediment particle is separated from its neighbor, it is no longer subject to the elastoplastic constitutive law. We discuss the strategy we adopt for this density evaluation in more detail in Sec. (§4.5.4).

3.3 Mixture incompressibility and interaction term

From the assumption that $\epsilon + \delta = 1$, we can combine Eq. (1) and Eq.(4) to get

$$\nabla\cdot(\epsilon\mathbf{u} + \delta\mathbf{v}) = 0. \tag{6}$$

We would like to note that in this work, the interaction terms between the fluid and the sediment are limited to drag forces ($\mathbf{f}^{sd}$ and $\mathbf{f}^{fd}$), which is a point of departure from [Anderson and Jackson 1967]. The exact model used is described in Sec. (§4.3). This model does not include the effect of buoyancy (necessary for capturing liquefication phenomena) and lift force. A caveat is that to conserve momentum, we only apply the drag force model to the sediment whereas in the fluid step the drag force is computed from the exchanged momentum accumulated during the sediment step.

4 METHOD

We discretize the velocity degrees of freedom and the forces of each constituent with the help of an MPM grid with grid-spacing Δx. We will use subscript i for quantities stored at grid node i. We take a semi-staggered discretization similar to [Zhang et al. 2017] for the fluid momentum equation. Once we commit to simulating sand using MPM, it is natural to simulate fluid on a semi-staggered grid since both fluid and sediment will be discretized on colocated nodes. Another advantage of this is to avoid the need to extrapolate velocity on the fluid surface, which is needed with FLIP. As a result, pressure and velocity divergence are stored at cell centers with subscript c. We use subscript f to denote variables on the faces. Both fluid and sediment materials are tracked and advected with MPM particles, which will be denoted with subscript p. The physical dimension is denoted by $d \in \{2, 3\}$.

4.1 Algorithm overview

We use a semi-implicit discretization for the fluid step and a symplectic Euler time integration for the sediment step. The fluid can thus take a larger time step than the sediment. We use $\Delta t^{f,n}$ to denote a fluid time step from t^n to $t^{n+1} =: t^n + \Delta t^{f,n}$. We divide one sediment timestep into K substeps, i.e. $\sum_{k=1}^{K}\Delta t^{s,n_k} = \Delta t^{f,n}$. Note that both time steps are chosen to satisfy the CFL condition [Stomakhin et al. 2013]. We summarize the essential steps in our method below and provide an illustration in Fig. 5.

First we describe a fluid step from t^n to t^{n+1}. We use APIC [Jiang et al. 2015] with a multi-linear kernel for particle-grid transfers.

(1) **Particles to grid:** Transfer fluid particle mass m_p^f and momentum $(m\mathbf{u})_p^n$ to the grid and divide the mass, giving nodal mass $m_i^{f,n}$ and velocity $\mathbf{u}_i^n$.
(2) **Apply forces:** Add *explicit* drag force and gravity to compute the intermediate velocities $\mathbf{u}_i^* \leftarrow \mathbf{u}_i^n$ on the background grid.
(3) **Pressure projection:** Solve mixture pressure projection with proper boundary conditions to correct the nodal velocities $\hat{\mathbf{u}}_i^{n+1} \leftarrow \mathbf{u}_i^*$ so that Eq. (9) is satisfied.
(4) **Grid to particles:** Particles get velocities $\mathbf{u}_p^{n+1}$ (and APIC coefficients) by interpolating from $\hat{\mathbf{u}}_i^{n+1}$. Advection is done as $\mathbf{x}_p^{f,n+1} = \mathbf{x}_p^{f,n} + \Delta t^{f,n}\mathbf{u}_p^{n+1}$.

Once we are done with the fluid step, we then integrate the sediment governing equations from t^n to t^{n+1} (equivalently $t^{n_0}, \ldots, t^{n_K}$) with symplectic Euler. Multiple sub-steps are usually required since sediment (which is similar to dry sand) has a considerably high Young's modulus which imposes a strong time step restriction. We use APIC with the quadratic B-spline kernel which is optimized, as in [Gao et al. 2017], for particle-grid transfers. We use the notation $N_i(\mathbf{x}_p^{s,n_k})$ as the evaluation of the kernel centered at grid node i at position $\mathbf{x}_p^{s,n_k}$. In each sub-step from t^{n_k} to $t^{n_{k+1}}$, the procedures are

(1) **Particles to grid:** Transfer sediment particle mass m_p^s and momentum $(m\mathbf{v})_p^{n_k}$ to the grid and then average to get nodal mass m_i^{s,n_k} and velocity $\mathbf{v}_i^{n_k}$. We also approximate mass gradient on grid nodes with $(\nabla m^{s,n_k})(\mathbf{x}_i)$. Nodal sediment volume fraction $\delta_i^{n_k}$ is also updated (§4.5.1).
(2) **Apply forces:** We compute the elastic force, fluid drag force, and gravity on the grid and update grid velocities based on these forces, i.e. $\hat{\mathbf{v}}_i^{n_{k+1}} \leftarrow \mathbf{v}_i^{n_k}$. Note that the drag forces conserve total momentum of the fluid-sediment system. We show more details in Sec. (§4.3.2). Next, we apply

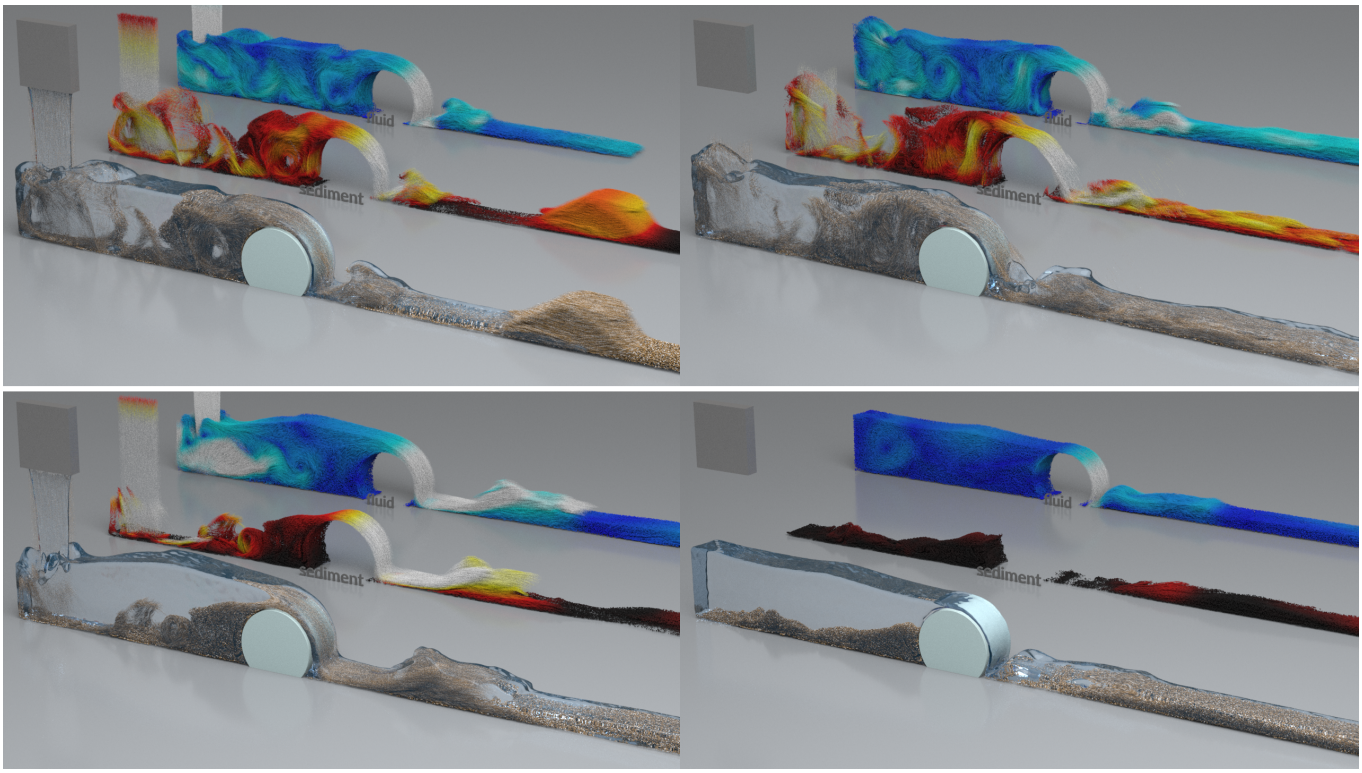

Fig. 4. **Hydraulic jump and sediment transport:** A mixture of water and sediment being poured into a thin channel with a cylindrical obstacle in the middle. The top figure uses an RPIC ratio of 0.2 (and a smaller gravity value), creating a more turbulent flow, while the bottom figure uses an RPIC ratio of 0.4, creating a more laminar flow.

boundary conditions according to [Stomakhin et al. 2013], i.e. $\mathbf{v}_i^{n_{k+1}} \leftarrow \hat{\mathbf{v}}_i^{n_{k+1}}$ (§4.5.2).

(3) **Update particles' states:** Particles' velocities and positions are updated by interpolation from the grid (§4.5.3).

(4) **Strain and representation update:** Particle deformation gradients $\mathbf{F}_p^{n_{k+1}}$ are updated using the Drucker-Prager plastic flow return mapping [Klár et al. 2016] with volume correction as proposed by [Pradhana-Tampubolon et al. 2017]. We also evaluate the world space sediment particle density $\rho_p^{s,n_{k+1}}$ to determine whether a particle should be treated as being part of the continuum or as a discrete debris (§4.5.4).

4.2 Volume fraction

Sediment particles are extremely stiff elastically, and conserve volume during the non-associative Drucker-Prager plastic flow [Klár et al. 2016]. In contrast to traditional MPM algorithms, we assume that the volume of each particle almost remains unchanged during the simulation ($V_p^{s,n_k} = V_p^{s,0}$ where $V_p^{s,0}$ is the original volume). Volume fraction of sediment is then given at each node as :

$$\delta_i^{n_k} = \min\left\{\delta_{\max}, \frac{V_i^{s,n_k}}{\Delta x^d}\right\}, \tag{7}$$

where $V_i^{s,n_k} = \sum_p V_p^{s,n_k} N_i(\mathbf{x}_p^{s,n_k})$. The value of $\delta_{\max}$ is chosen for numerical stability ($\delta_{\max} \in [0.5, 0.8]$ in all of our examples). The corresponding nodal fluid volume fraction is then $\epsilon_i^n = 1 - \delta_i^{n_0} \in [1 - \delta_{max}, 1]$. Note that $\epsilon_i^n = 1$ if $m_i^{s,n_0} = 0$. In the absence of sediment particles, our locally averaged fluid discretization (§4.4.2) reduces to the standard incompressible flow discretization from [Zhang et al. 2017].

4.3 Drag force

4.3.1 On sediment. In contrast to CFD-DEM approaches where particle-wise drag forces are evaluated by interpolating fluid velocities to sediment locations, MPM allows us to directly apply drag forces to grid nodes. Corresponding to the drag force density $\mathbf{f}^{sd}$ in the continuum equation, we use a drag force model proposed by Di Felice [1994] in a discrete setting. On grid nodes where fluid and sediment materials are present we compute

$$\mathbf{f}_i^{sd,n_k} = \frac{1}{2} c_i (\epsilon_i^n)^{-\chi} \rho^f A_i^{s,n_k} |\mathbf{u}_i^{n+1} - \mathbf{v}_i^{n_k}| (\mathbf{u}_i^{n+1} - \mathbf{v}_i^{n_k}), \tag{8}$$

with c_i and χ denoting the empirical coefficient and exponent respectively. For high Reynolds number fluids, $\chi = 3.7$ and $c_i = 0.39$. The term A_i^{s,n_k} is the cross-sectional area of an imaginary spherical geometry, given by $A_i^{s,n_k} = 2\left(V_i^{s,n_k}/\pi\right)^{1/2}$ in 2D and $A_i^{s,n_k} = \pi\left(3V_i^{s,n_k}/(4\pi)\right)^{2/3}$ in 3D. This term guarantees nodes

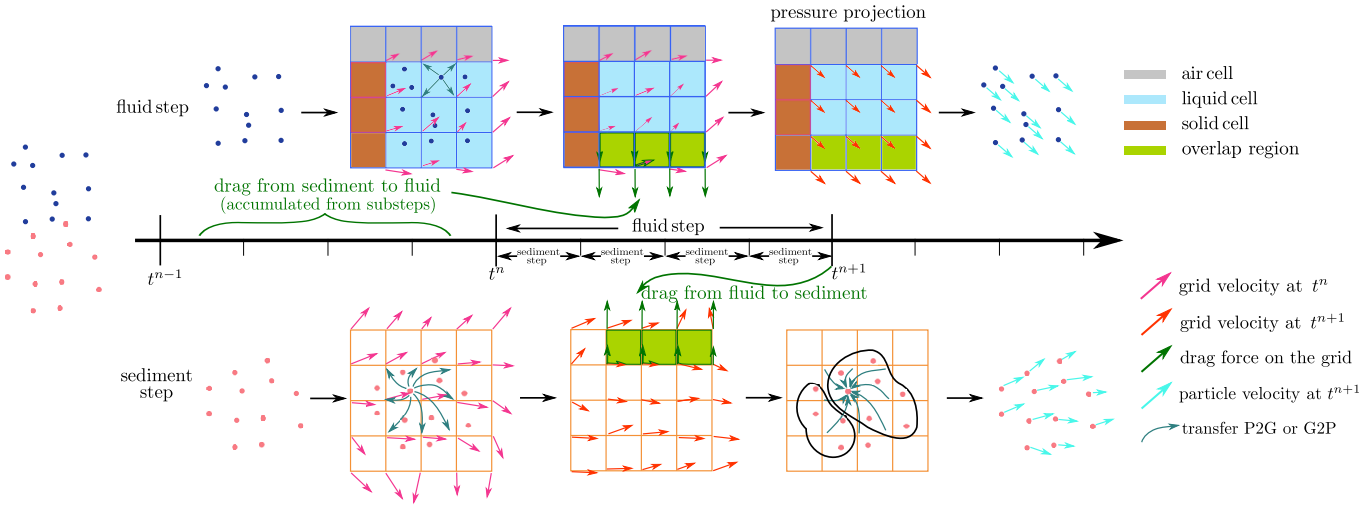

Fig. 5. **Algorithm step:** The top row illustrates the fluid step, where we perform pressure projection to solve for the new velocity on the grid. The result of this computation is then used in the sedimentation step, illustrated in the bottom row. Note that for a single fluid step, there are multiple (explicit) MPM sediment sub-steps. The drag forces from each of the sediment sub-steps are accumulated to compute $\mathbf{f}^{fd}$ for the fluid step.

with tiny mass will receive a tiny force, thus avoiding potential numerical instabilities. In practice we adjust c_i for controlling the overall strength of the drag force. This drag force model is more suitable for particle laden flows and differs from the model used by Pradhana et al. [2017].

4.3.2 On fluid. As we proceed from t^{n_0} to t^{n_K}, the total momentum change on the sediment is $\Delta\mathbf{P}_i^{sd,n} = \sum_{k=1}^{K} \mathbf{f}_i^{sd,n_k} \Delta t^{s,n_k}$. To enforce total momentum conservation in the fluid-sediment system, we set $\mathbf{f}_i^{fd,n} = -\Delta\mathbf{P}_i^{sd,n-1}/\Delta t^{f,n}$ when we apply the drag force to fluid. As a result $\Delta\mathbf{P}_i^{fd,n} = -\Delta\mathbf{P}_i^{sd,n-1}$ is always satisfied, i.e. any momentum change to the sediment caused by the drag force is always negated and applied to the fluid in the upcoming fluid step.

4.4 Fluid

4.4.1 Splitting. We apply the standard splitting treatment of fluid simulation (cf. [Bridson 2008] for details) to Eq. (3) to get separate steps, which includes advection, application of external forces, pressure projection and velocity correction. Here we focus on the grid solve. Particle-grid transfer steps are described in Sec. (§4.1).

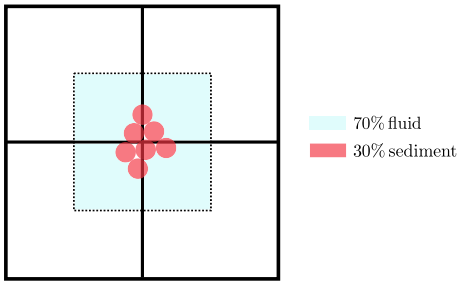

Fig. 6. **Volume fraction:** In this point of view, water cannot enter into sediment. The node at the center represents 70% fluid and 30% sediment.

Advection. In hybrid particle-grid fluid solvers, the nodal velocities are transferred from fluid particles, which have been advected at the end of the previous time step through particle advection. In other words, there is no additional advection required on the grid in this new cycle.

External forces. This step includes the application of both drag forces from sediment particles and gravity to get the intermediate velocities $\mathbf{u}_i^* = \mathbf{u}_i^n + \left(\frac{\mathbf{f}_i^{fd,n}}{m_i^{f,n}} + \mathbf{g}\right)\Delta t$. Note that our method is explicit on the drag force and uses $\mathbf{f}_i^{fd,n}$ directly.

Pressure projection. We adopt a semi-implicit discretization to rewrite Eq. (6) as

$$\nabla \cdot (\epsilon_i^n \mathbf{u}_i^{n+1}) = -\nabla \cdot (\delta_i^{n_0} \mathbf{v}_i^n), \tag{9}$$

where the divergence of volume fraction weighted sediment velocities is treated explicitly. This allows us to combine it with

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{\rho}\nabla p = 0$$

to derive a variable coefficient Poisson's equation for the fluid pressure, namely

$$-\frac{\Delta t}{\rho^f}\nabla \cdot (\epsilon_f^n \nabla p) = -\nabla \cdot (\delta_i^{n_0} \mathbf{v}_i^n) - \nabla \cdot (\epsilon_i^n \mathbf{u}_i^*), \tag{10}$$

where both ϵ_f^n and ∇p are discretized at cell faces. Considering that ϵ_f^n is always positive, Eq. (10) corresponds to a symmetric positive (semi-)definite system.

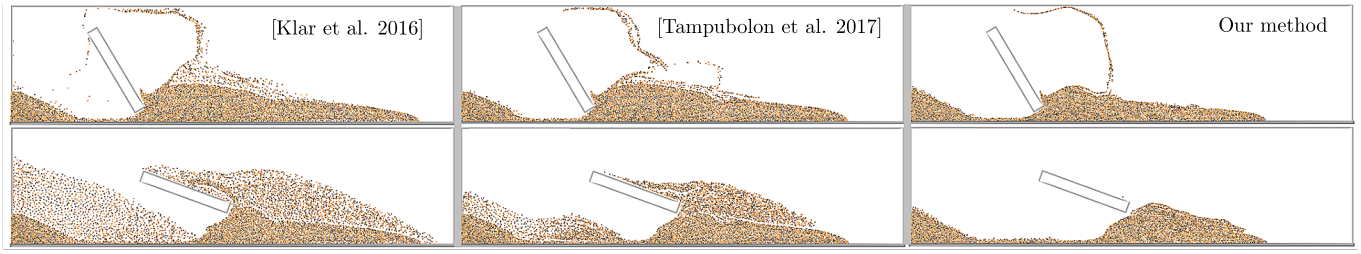

Fig. 7. **Volume gain problem:** Previous MPM granular methods can introduce volume gain for highly dynamic scenarios. [Pradhana-Tampubolon et al. 2017] proposed a method to mitigate this volume gain effect to a certain extent. Our method successfully resolves this problem.

Velocity correction. After the pressure values are solved at cell centers, they can be used to correct fluid velocities as

$$\frac{\hat{\mathbf{u}}_i^{n+1} - \mathbf{u}_i^*}{\Delta t} = -\frac{1}{\rho^f}\nabla p, \tag{11}$$

where $\hat{\mathbf{u}}_i^{n+1}$ is further transferred back to particles using APIC (§4.1).

4.4.2 A Semi-staggered Discretization. As discussed in Sec. (§1), constructing an incompressible solid-fluid coupling solver on a fully staggered MAC grid as done by Stomakhin et al. [2014] would require multiple axis-independent MPM transfers between particles and cell faces with a cubic kernel. To avoid this overhead, we adopt the recently proposed semi-staggered MPM incompressible fluid discretization [Zhang et al. 2017] and adapt it to solve the Locally Averaged Navier-Stokes with a variable coefficient multigrid preconditioner.

We first summarize the MPM discretization for the pressure solve in a pure fluid pipeline as in [Zhang et al. 2017]. A standard pressure solve can be described by

$$-\frac{\Delta t}{\rho}\nabla \cdot (\nabla p) = -\nabla \cdot \mathbf{u}. \tag{12}$$

Pressures p are stored at cell centers, while all velocities are stored at cell nodes and are transferred between MPM particles and the background grid via the standard multi-linear kernel with FLIP [Brackbill and Ruppel 1986]. We use the angular momentum conserving APIC [Jiang et al. 2015] instead for stability and low dissipation. We further adopt the RPIC damping as described in [Jiang et al. 2017] as a controllable artificial viscosity for changing liquid behavior. The left-hand side of Eq. (12) can be discretized with a standard 7-point Laplacian stencil. For the right-hand side, the standard way to compute the divergence of velocity in the semi-staggered discretization is to first compute the partial derivatives on the edges and then average them to the cell center. Zhang et al. [2017] compute this differently by directly applying the divergence operator to the interpolated velocity which is then transferred to the weighting function, ending up with the identical form. It is well known that this semi-staggered discretization causes spurious hourglass velocity modes which cannot be eliminated by the pressure projection. We mitigate this problem by using an hourglass damping as proposed by Zhang et al. [2017].

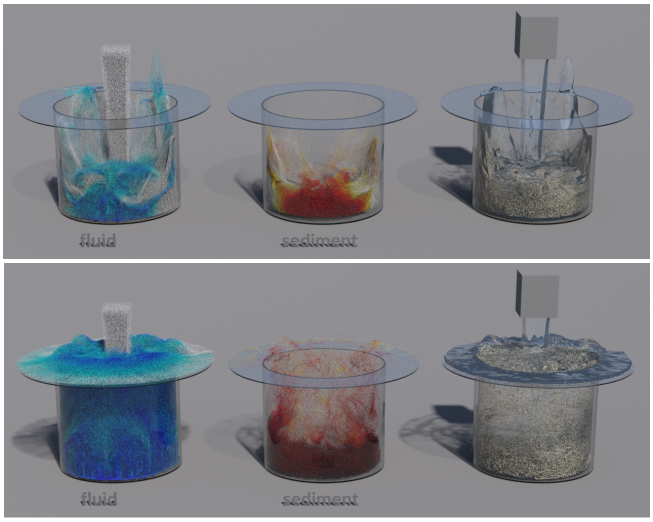

Fig. 8. **Glass flush:** Water is poured onto sediment in a glass. The overflow of water transports the sediment out of the glass.

Ghost pressure. We mark a cell to be a water cell whenever there is a water particle present in the cell, and mark a cell to be Neumann when the cell center is inside an object or a wall. All other cells are air (Dirichlet) cells.

We use a simplified ghost pressure treatment [Gibou et al. 2002] to get a smooth free surface without tracking a level set or doing reconstruction from particles. One popular interpretation of the ghost pressure method is to compute the pressure gradient using a more accurate distance from the cell center to the free surface $\theta\Delta x$ (cf. Fig. 9(a)), instead of the distance between two cell centers Δx. We first identify the water particle closest to the face separating the water cell and the air cell. We then assign a sphere of radius $\frac{\Delta x}{2}$ to the particle to define an axis-aligned free surface and the corresponding θ (illustrated in Fig. 9(b)). It can be shown that when the water particle crosses the cell, this new free surface is continuous and passes through the air cell center; thus this scheme is free from cell-crossing discontinuity artifacts.

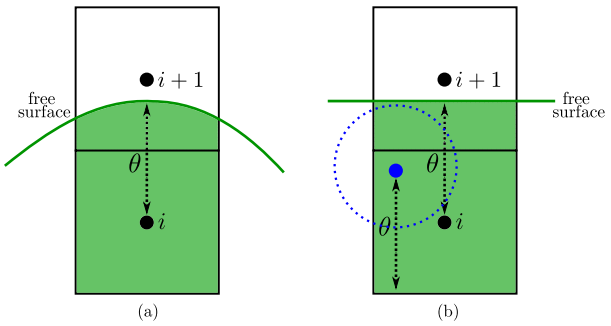

Fig. 9. **Simple ghost pressure computation:** *Left:* With a more accurate free-surface representation, e.g. using level set, one can compute θ. *Right:* We propose a simplified way to compute θ even without the presence of a level set.

Adaptation to the mixture pipeline. For solving the mixture projection Eq. (10), we need to compute the divergence of the volume-fraction-averaged mixture velocity: $\hat{\mathbf{u}} = \delta_i^{n_0}\mathbf{v}_i^n + \epsilon_i^n\mathbf{u}_i^*$. It is also located at cell nodes, so the divergence operator remains the same. Moreover, the 7-point stencil becomes a variable coefficient Laplacian $\nabla\cdot(\epsilon_f^n\nabla p)$. ϵ_f^n is usually stored at the cell faces to be collocated with the standard finite difference pressure gradient. Instead, we choose to interpolate nodal water volume fractions ϵ_i^n to cell centers to store them as ϵ_c^n and compute the face average whenever needed. In this way, the system to be solved remains symmetric. This choice also simplifies the definition of the restriction operator for the volume fraction in the multigrid preconditioner as discussed below.

Notice that the velocity correction Eq. (11) remains the same as in the standard pure fluid pipeline [Zhang et al. 2017]. Pressure gradients are first computed for each face; then the weighted average of these is used to update the nodal velocities. The face weight is set to be one whenever that face belongs to a water cell; otherwise it is set to be zero.

A variable coefficient multigrid preconditioner. A multigrid preconditioner is utilized to improve the convergence rate of the conjugate gradient solver. Since the fluid volume fraction term ϵ_f^n is confined within a narrow range (due to clamping), we are able to slightly modify the technique proposed by McAdams et al. [2010] to achieve a reasonable convergence acceleration. In the multigrid V-cycle, two operators, namely the prolongation and restriction operators, are defined to act on the pressure degrees of freedom. The restriction operator averages the finer data to be more coarse, and the prolongation operator interpolates finer values from the coarser grid. During each iteration, the pressures at each level are updated. In contrast, volume fraction coefficients only need to be restricted from finer levels to coarser levels once because they do not act as real degrees of freedom and can be re-used during the whole V-cycle.

As mentioned, although ϵ_f^n is required to solve the variable coefficient Poisson equation, we choose to store the volume fraction at cell centers and use them to compute ϵ_f^n whenever needed. This choice preserves the symmetry of the system. Furthermore, the restriction of volume fractions can be computed by averaging the interior cells.

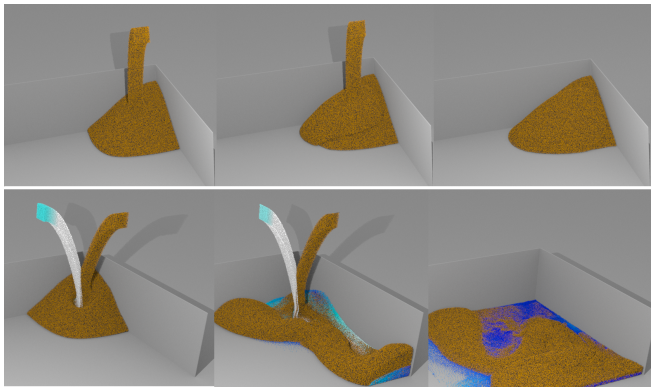

Fig. 10. **Pile:** *Top:* Pure sediment particles form a pile with static friction. *Bottom:* By injecting liquid, the pile collapses and interacts with the fluid.

Our method differs from McAdams et al. [2010] in that we do not average values from all finer cells with zeros substituted into non-interior cells. With this modification, we make sure that when all of the volume fractions are one, the variable coefficient multigrid preconditioner can transform back to the constant coefficient case.

4.5 Sediment

The following sections describe each of the steps involved in advancing the states of the sediment particles from sub-step n_k to n_{k+1}, with $0 \le k < K$.

4.5.1 Particles to Grid. Following [Klár et al. 2016], we transfer mass and momentum from particles to the grid using quadratic basis function according to

$$m_i^{s,n_k} = \sum_p m_p^s N_i(\mathbf{x}_p^{s,n_k}), \tag{13}$$

$$(m\mathbf{v})_i^{n_k} = \sum_p m_p^s N_i(\mathbf{x}_p^{s,n_k})\left(\mathbf{v}_p^{n_k} + \frac{4}{(\Delta x)^2}\mathbf{B}_p^{s,n_k}(\mathbf{x}_i - \mathbf{x}_p^{s,n_k})\right), \tag{14}$$

where the APIC $\mathbf{B}_p^{s,n_k}$ is originally set to zero and is updated at each substep according to Eq.(21). This is an APIC transfer [Jiang et al. 2015], with the factor $4/(\Delta x)^2$ applied to a quadratic kernel. The velocity on the grid is then computed as $\mathbf{v}_i^{n_k} = \frac{(m\mathbf{v})_i^{n_k}}{m_i^{s,n_k}}$.

In addition to this standard MPM transfer, we also compute mass gradients on grid nodes which will be used later in our particle density approximation, namely

$$(\nabla m^{s,n_k})(\mathbf{x}_i) = -\sum_p m_p^s \nabla N_i(\mathbf{x}_p^{s,n_k}). \tag{15}$$

Furthermore, we update the nodal sediment volume fraction according to Eq.(7) during the transfer.

4.5.2 Update grid momentum and boundary conditions. In this step, we update the velocity $\mathbf{v}_i^{n_k} \to \hat{\mathbf{v}}_i^{n_{k+1}}$ by taking into account the drag force in regions where the fluid and sediment materials overlap along with the elastic forces from the sediment constitutive

law. The drag force is given by Eq.(8). The elastic force is computed as in [Klár et al. 2016]:

$$\mathbf{f}_i^{se,n_k} = -\sum_p V_p^0 \left(\frac{\partial \psi}{\partial \mathbf{F}}(\mathbf{F}_p^{n_k})\right)(\mathbf{F}_p^{n_k})^\top \nabla N_i(\mathbf{x}_p^{s,n_k}), \tag{16}$$

with

$$\frac{\partial \psi}{\partial \mathbf{F}}(\mathbf{F}) = \mathbf{U}\left(2\mu\Sigma^{-1}\ln\Sigma + \lambda\,\mathrm{tr}(\ln\Sigma)\Sigma^{-1}\right)\mathbf{V}^\top, \tag{17}$$

where $\mathbf{F} = \mathbf{U}\Sigma\mathbf{V}^\top$ is the SVD of $\mathbf{F}$. The parameters λ and μ are the Lamé coefficients. The velocity update on the sediment grid is given by

$$m_i^{s,n_k}\hat{\mathbf{v}}_i^{n_{k+1}} = m_i^{s,n_k}\mathbf{v}_i^{n_k} + \Delta t\left(\mathbf{f}_i^{sd,n_k} + \mathbf{f}_i^{se,n_k} + m_i^{s,n_k}\mathbf{g}\right). \tag{18}$$

Finally, we apply the boundary conditions following [Stomakhin et al. 2013] to get the final update on the grid $\mathbf{v}_i^{n_{k+1}} \leftarrow \hat{\mathbf{v}}_i^{n_{k+1}}$.

4.5.3 Grid to particles. In this step, we update the velocity, position, and $\mathbf{B}_p^{s,n_{k+1}}$ of each particle following [Jiang et al. 2015]:

$$\mathbf{v}_p^{n_{k+1}} = \sum_i N_i(\mathbf{x}_p^{s,n_k})\mathbf{v}_i^{n_{k+1}}, \tag{19}$$

$$\mathbf{x}_p^{s,n_{k+1}} = \mathbf{x}_p^{s,n_k} + \Delta t^{s,n_k}\mathbf{v}_p^{n_{k+1}}, \tag{20}$$

$$\mathbf{B}_p^{s,n_{k+1}} = \sum_i N_i(\mathbf{x}_p^{s,n_k})\mathbf{v}_i^{n_{k+1}}(\mathbf{x}_i - \mathbf{x}_p^{s,n_k})^\top. \tag{21}$$

4.5.4 Strain update with density-based projection. To update the deformation gradient, we first compute

$$\mathbf{F}_p^{n_{k+1}} = \left(\mathbf{I} + \Delta t\sum_i \mathbf{v}_i^{n_{k+1}}\nabla N_i(\mathbf{x}_p^{s,n_k})^\top\right)\mathbf{F}_p^{n_k}.$$

This deformation is then projected so that it obeys the Drucker-Prager plasticity model as in [Pradhana-Tampubolon et al. 2017]. However, particles that are separated from clumps should not be subject to plasticity and should be treated as discrete debris with zero stress.

We approximate the particle density using a first-order accurate formula. Specifically, we expand $m(\mathbf{x})$ around $\mathbf{x}_i$ as $m(\mathbf{x}) \approx m(\mathbf{x}_i) + (\nabla m)(\mathbf{x}_i)\cdot(\mathbf{x}-\mathbf{x}_i)$. If we reconstruct this mass field at $\mathbf{x}_p$ accordingly, we get $m^i(\mathbf{x}_p) \approx m(\mathbf{x}_i) + (\nabla m)(\mathbf{x}_i)\cdot(\mathbf{x}_p - \mathbf{x}_i)$, where we use the superscript i to denote the use of node i for linearizing the mass field. We estimate particle density using the weighted average of grid masses:

$$\rho_p = \sum_i N_i(\mathbf{x}_p)m^i(\mathbf{x}_p) \approx \sum_i N_i(\mathbf{x}_p)(m_i + (\nabla m)(\mathbf{x}_i)\cdot(\mathbf{x}_p - \mathbf{x}_i)). \tag{22}$$

In the discrete case, this is written as

$$\rho_p^{s,n_{k+1}} = \sum_i N_i(\mathbf{x}_p^{s,n_k})\left(m_i^{s,n_k} + (\nabla m^{s,n_k})(\mathbf{x}_i)^\top(\mathbf{x}_p^{s,n_k} - \mathbf{x}_i)\right),$$

where $\nabla m^{s,n_k}(\mathbf{x}_i)$ is defined in Eq. (15). One can view the gradient of mass term as the first order approximation of nodal mass from the mass of the surrounding particles. We validated the accuracy of this strategy for a fluid simulation based on a weakly compressible Equation-of-State model [Becker and Teschner 2007] and found that an interpolation solely based on mass produces grid artifacts. We would like to note that the gradient term can be seen as a second-order term in the Taylor expansion of the mass expression.

To utilize this density approximation, at time $t = 0$, we compute the initial average density

$$\hat{\rho} := \frac{1}{N_p}\sum_{p=1}^{N_p}\rho_p^{s,0}, \tag{23}$$

where the average is taken over all sediment particles that exist at time 0. If the current density approximation is lower than this initial average density, i.e. $\rho_p^{s,n_{k+1}} < \hat{\rho}$, we then identify the particle as a discrete kinematic body that does not undergo any elastic deformation. This is equivalent to setting

$$\mathbf{F}_p^{n_{k+1}} = \mathbf{I}.$$

Note that due to the usage of APIC, the discrete particle still acts as an independent body with a finite volume and maintains its angular momentum.

5 RESULTS

We summarize the performance of our implementation in Table 2. In our pipeline, the Preconditioned Conjugate Gradient (PCG) projection remains the most expensive component (this step takes about 6.25 seconds in the 3D transport example, compared to 3.3 seconds for all the other operations). The sediment substep count depends on the CFL condition. In the debris flow simulation, we take $60-100$ sediment substeps per fluid step.

Fig. 7 shows the benefit of our debris treatment. As a block stirs a pile of sand, previous elastoplastic MPM algorithms introduce a severe volume gain. Pradhana et al. [2017] proposed a method to mitigate this volume gain effect to a certain extent. However, in a highly dynamic scenario like this, disconnected particles tend to suffer from intense resistance when connecting back to a more massive bulk. Our method offers a way to approximate state change, i.e., when particles act like debris based on the surrounding density. For a more accurate ballistic motion, more careful modeling is necessary [Smith et al. 2012].

Our algorithm can reproduce classical particle-laden phenomena such as sand dune migration (Fig. 2). In this example, a source of wind blows over a pile of sand as it moves from the left to the right, forcing sand to migrate. The left of the figure depicts the result of a simulation that is one-way-coupled (the wind affects the sand), which produces a non-physical behavior. The right figure depicts a two-way-coupled simulation, which creates more physically plausible dynamics on both the fluid and sand materials. The fluid (wind) part of this example is based on the node-based Eulerian fluid solver as proposed by [Guendelman et al. 2005] with vorticity confinement as described in [Fedkiw et al. 2001]. Note that all of the other examples with water as fluids (i.e., based on [Zhang et al. 2017]) do not use vorticity confinement.

Fig. 1 and 4 depict a particle-laden phenomenon that is created when a mixture of sand and water is being poured into a thin channel. The movement of the fluid transports the sediment. The presence of a cylindrical obstacle in the channel creates a hydraulic jump that carries the sand over to the other side. The use of RPIC blending helps to create a more laminar motion of the fluid (Fig. 4 bottom). A

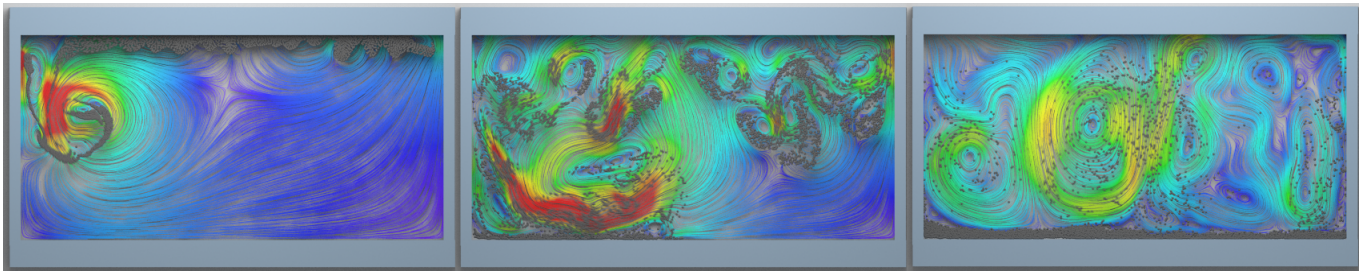

Fig. 11. **Sedimentation 2D:** A box of dirt is dropped into a water container, creating interesting vortices before settling down.

	Min/frame	Initial $\#P^f$	Initial $\#P^s$	Max $\#P^f$	Max $\#P^s$	Δx	c	RPIC	ρ^f	ρ^s	Inflow $\|\|\mathbf{v}^f\|\|$	Inflow $\|\|\mathbf{v}^s\|\|$
Transport (Fig. 4 Top)	4.9	0	3.1×10^5	1.0×10^6	3.9×10^5	1/512	0.9	0.2	1000	17	0.6	0.06
Transport (Fig. 4 Bottom)	4.9	0	3.1×10^5	1.0×10^6	3.9×10^5	1/512	0.9	0.4	1000	17	0.6	0.06
Pile (Fig. 10)	11.1	0	6.5×10^5	3.0×10^5	9.5×10^5	1/256	4	0	100	10	0.4	0.4
Flushing (Fig. 8)	21	0	2.3×10^5	1.9×10^6	2.3×10^5	1/256	2	0	1000	17	0.6	0
Debris (Fig. 3)	2.5	0	0	9.3×10^5	1.0×10^5	1/256	0	0	1000	150	0.15	0.015
Sedimentation (Fig. 12)	6.7	1.5×10^6	0	1.5×10^6	1.0×10^5	1/256	12	0.2	100	20	0	0.01

Table 2. **Simulation performance and parameters.** Superscript f and s represent fluid and sediment quantity respectively. The terms c, RPIC, and ρ are the drag force coefficient, RPIC damping coefficient, and density respectively. Inflow $\|\mathbf{v}\|$ indicates the magnitude of source velocity. The parameter ρ^f choices are adhoc and do not always reflect reality. We denote the number of particles by $\#P$ and grid resolution by Δx. The frame rate of all the examples listed above are 48. The young's modulus and Poisson's ratio of all examples are 1000 and 0.3 respectively. Both Transports (Fig. 4 Top and Bottom) were run with Intel(R) Xeon(R) CPU E3-1270 v3; Pile (Fig. 10) was run with Intel(R) Xeon(R) CPU E3-1241 v3; Flushing (Fig. 8) was run with Intel(R) Xeon(R) CPU E5-1650 v3; Sedimentation (Fig. 12) was run with Intel(R) Core(TM) i7-8700K; Debris (Fig. 3) was run with Intel(R) Core(TM) i7-7700K.

similar phenomenon of debris flow, where a water and sand mixture is poured into a wider channel with more complex collision objects is illustrated in Fig. 3.

Our algorithm can also handle sedimentation phenomena, which is depicted by Fig. 11 and Fig. 12. In these 2D and 3D simulations, a box of sand is being dropped into a water container. We can observe vortices that are created by the interaction between water and sand materials. The sand materials eventually settle down at the bottom of the container.

Fig. 10 depicts a simulation where pure sediment particles form a pile due to static friction (top). When liquid is injected, the pile collapses and interacts with the fluid. Similarly, water is poured on top of a pile of sediment that initially sits at the bottom of a glass container in Fig. 8. The overflow of water washes the sediment out of the glass.

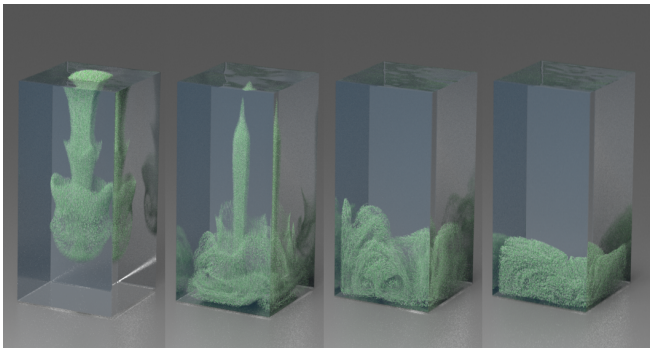

Fig. 12. **Sedimentation 3D:** We pour sediment into a box of water until it settles at the bottom of the container.

6 DISCUSSION

Our model qualitatively captures a number of distinctive dynamic features of particle-laden flows (such as dune migration and sediment transport) and stands as a practical tool in animation and VFX. As suggested in Sec. (§3.3), our model of the interaction term only uses drag forces based on [Di Felice 1994]. We do not make any claims on how broadly this model effectively captures empirical experiments. As such, a more rigorous study of sediment-fluid interaction would require careful modeling of the other nontrivial effects such as the lift force as discussed by Sun and Xiao [2016b].

Another limitation of our method is the clamping of the sediment volume fraction (at locations where sediment and fluid co-exist), which is motivated by numerical stability considerations. As such, our simulations cannot capture the variations of granular flow as studied in [Pailha et al. 2008]. Furthermore, at the interface of dry and immersed sand, we do not apply any special treatment for the sudden jump of sediment volume fraction from $\delta = 0.8$ (the case of liquid and sediment co-existing) to $\delta = 1$ (the case of pure sediment) as no noticeable artifacts were observed.

We presently rely on artificial viscosity from RPIC for adjusting the extent of fluid turbulence. More physical and accurate viscosity models incorporated into the Locally Averaged Navier-Stokes discretization would be appropriate to investigate in future work.

Due to the explicit treatment of the drag force, we occasionally encounter numerical instability when the relative velocity is too high, or the fluid volume fraction is too low (thus resorting to the usage of a clamping threshold). Exploring a fully implicit treatment

of the interaction term would be a potential direction for future study. Furthermore, the design of a preconditioner explicitly tailored to the variable-coefficient Poisson system is an interesting research question.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their valuable suggestions and comments. We thank Joshuah Wolper for narrating the video. This work was supported in part by NSF Grants IIS-1253598, IIS-1755544, CCF-1533885, a gift from Awowd Inc., a gift from SideFX, and NVIDIA GPU Grants.

REFERENCES

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

M. Becker and M. Teschner. 2007. Weakly Compressible SPH for Free Surface Flows. In *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim.* 209–217.

M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun. 2008. Discrete elastic rods. In *ACM Trans Graph*, Vol. 27. ACM, 63.

J. Brackbill and H. Ruppel. 1986. FLIP: A method for adaptively zoned, Particle-In-Cell calculations of fluid flows in two dimensions. *J Comp Phys* 65 (1986), 314–343.

R. Bridson. 2008. *Fluid simulation for computer graphics.* Taylor & Francis.

G. Daviet. 2016. *Modeling and Simulating Complex Materials subject to Frictional Contact.* Ph.D. Dissertation. Universite Grenoble Alpes.

G. Daviet and F. Bertails-Descoubes. 2016. A semi-implicit material point method for the continuum simulation of granular materials. *ACM Trans Graph* 35, 4 (2016), 102:1–102:13.

R Di Felice. 1994. The voidage function for fluid-particle interaction systems. *Int J Mult Flow* 20, 1 (1994), 153–159.

S. Dunatunga and K. Kamrin. 2015. Continuum modelling and simulation of granular flows through their many phases. *J Fluid Mech* 779 (2015), 483–513.

R. Fedkiw, J. Stam, and H.W. Jensen. 2001. Visual simulation of smoke. In *Proc of the 28th Ann Conf on Comp Graph and Int Techs.* ACM, 15–22.

Y.R. Fei, H.T. Maia, C. Batty, C. Zheng, and E. Grinspun. 2017. A multi-scale model for simulating liquid-hair interactions. *ACM Trans Graph* 36, 4 (2017), 56.

M. Gao, A.P. Tampubolon, C. Jiang, and E. Sifakis. 2017. An Adaptive Generalized Interpolation Material Point Method for Simulating Elastoplastic Materials. *ACM Trans Graph* 36, 6 (2017).

F. Gibou, R.P. Fedkiw, L.T. Cheng, and M. Kang. 2002. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *J Comp Phys* 176, 1 (2002), 205–227.

E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw. 2005. Coupling Water and Smoke to Thin Deformable and Rigid Shells. *ACM Trans Graph* 24, 3 (July 2005), 973–981.

C. Jiang, T. Gast, and J. Teran. 2017. Anisotropic elastoplasticity for cloth, knit and hair frictional contact. *ACM Trans Graph* 36, 4 (2017).

C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin. 2015. The affine particle-in-cell method. *ACM Trans Graph* 34, 4 (2015), 51:1–51:10.

G. Klár, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, and J. Teran. 2016. Drucker-prager elastoplasticity for sand animation. *ACM Trans Graph* 35, 4 (2016), 103:1–103:12.

P. Krištof, B. Beneš, J. Křivánek, and O. Šťava. 2009. Hydraulic erosion using smoothed particle hydrodynamics. In *Comp Graph Forum*, Vol. 28. Wiley Online Library, 219–228.

T. Lenaerts, B. Adams, and P. Dutré. 2008. Porous flow in particle-based fluid simulations. *ACM Trans Graph* 27, 3 (2008), 49.

T. Lenaerts and P. Dutré. 2009. Mixing fluids and granular materials. *Comp Graph Forum* 28, 2 (2009), 213–218.

Y. Li, J. Zhang, and L.S. Fan. 1999. Numerical simulation of gas–liquid–solid fluidization systems using a combined CFD-VOF-DPM method: bubble wake behavior. *Chem Eng Sci* 54, 21 (1999), 5101–5107.

M. Macklin, M. Muller, N. Chentanez, and T. Kim. 2014. nified particle physics for real-time applications. *ACM Trans Graph* 33, 4 (2014), 153:1–153:12.

M. Manninen, V. Taivassalo, S. Kallio, and others. 1996. On the mixture model for multiphase flow. (1996).

A. McAdams, E. Sifakis, and J. Teran. 2010. A parallel multigrid Poisson solver for fluids simulation on large grids. In *Proc of the 2010 ACM SIGGRAPH/Eurograph Symp Comp Anim.* Eurographics Association, 65–74.

M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff. 2007. Position Based Dynamics. *J Vis Comm Imag Repre* 18, 2 (2007), 109–118.

M. Müller, B. Solenthaler, R. Keiser, and M. Gross. 2005. Particle-based fluid-fluid interaction. In *Proc of the 2005 ACM SIGGRAPH/Eurograp Symp Comp Anim.* ACM, 237–244.

P. Mutabaruka and K. Kamrin. 2017. A simulation technique for slurries interacting with moving parts and deformable solids with applications. *arXiv preprint arXiv:1703.05158* (2017).

R. Narain, A. Golas, and M. Lin. 2010. Free-flowing granular materials with two-way solid coupling. *ACM Trans Graph* 29, 6 (2010), 173:1–173:10.

M.B. Nielsen and O. Østerby. 2013. A two-continua approach to Eulerian simulation of water spray. *ACM Trans Graph* 32, 4 (2013), 67.

M. Pailha, M. Nicolas, and O. Pouliquen. 2008. Initiation of underwater granular avalanches: Influence of the initial volume fraction. *Phys of Fluids* 20, 11 (2008), 111701.

A. Peer, M. Ihmsen, J. Cornelis, and M. Teschner. 2015. An implicit viscosity formulation for sph fluids. *ACM Trans Graph* 34, 4 (2015), 114.

A. Pradhana-Tampubolon, T. Gast, G. Klár, C. Fu, J. Teran, C. Jiang, and K. Museth. 2017. Multi-species simulation of porous sand and water mixtures. *ACM Trans Graph* 36, 4 (2017).

D. Ram, T. Gast, C. Jiang, C. Schroeder, A. Stomakhin, J. Teran, and P. Kavehpour. 2015. A material point method for viscoelastic fluids, foams and sponges. In *Proc ACM SIGGRAPH/Eurograph Symp Comp Anim.* 157–163.

B. Ren, C. Li, X. Yan, M.C. Lin, J. Bonet, and S.M. Hu. 2014. Multiple-fluid SPH simulation using a mixture model. *ACM Trans Graph* 33, 5 (2014), 171.

M. Robinson and M. Ramaioli. 2011. Mesoscale fluid-particle interaction using two-way coupled SPH and the Discrete Element Method. In *Proc of the 6th SPHERIC Work.* 72–78.

W. Rungjiratananon, Y. Kanamori, and T. Nishita. 2012. Wetting effects in hair simulation. In *Comp Graph Forum*, Vol. 31. Wiley Online Library, 1993–2002.

W. Rungjiratananon, Z. Szego, Y. Kanamori, and T. Nishita. 2008. Real-time Animation of Sand-Water Interaction. In *Comp Graph Forum*, Vol. 27. Wiley Online Library, 1887–1893.

B. Smith, D. M. Kaufman, E. Vouga, R. Tamstorf, and E. Grinspun. 2012. Reflections on Simultaneous Impact. *ACM Trans Graph* 31, 4, Article 106 (July 2012), 12 pages.

D. M. Snider. 2001. An incompressible three-dimensional multiphase particle-in-cell model for dense particle flows. *J Comp Phys* 170, 2 (2001), 523–549.

A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle. 2013. A material point method for snow simulation. *ACM Trans Graph* 32, 4 (2013), 102:1–102:10.

A. Stomakhin, C. Schroeder, C. Jiang, L. Chai, J. Teran, and A. Selle. 2014. Augmented MPM for phase-change and varied materials. *ACM Trans Graph* 33, 4 (2014), 138:1–138:11.

D. Sulsky, S. Zhou, and H. Schreyer. 1995. Application of a particle-in-cell method to solid mechanics. *Comp Phys Comm* 87, 1 (1995), 236–252.

R. Sun and H. Xiao. 2016a. CFD–DEM simulations of current-induced dune formation and morphological evolution. *Adv Water Res* 92 (2016), 228–239.

R. Sun and H. Xiao. 2016b. SediFoam: A general-purpose, open-source CFD–DEM solver for particle-laden flow with emphasis on sediment transport. *Comp & Geo* 89 (2016), 207–219.

Y. Teng, D. Levin, and T. Kim. 2016. Eulerian Solid-fluid Coupling. *ACM Trans Graph* 35, 6 (2016), 200:1–200:8.

M.A. van der Hoef, M. Ye, M. van Sint Annaland, A.T. Andrews, S. Sundaresan, and J.A.M. Kuipers. 2006. Multiscale modeling of gas-fluidized beds. *Adv Chem Eng* 31 (2006), 65–149.

C. Wojtan, M. Carlson, P.J. Mucha, and G. Turk. 2007. Animating Corrosion and Erosion. In *Eurograph Work Nat Phen.* 15–22.

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

B.H. Xu and A.B. Yu. 1997. Numerical simulation of the gas-solid flow in a fluidized bed by combining discrete particle method with computational fluid dynamics. *Chem Eng Sci* 52, 16 (1997), 2785 – 2809.

X. Yan, Y. Jiang, C. Li, R. Martin, and S. Hu. 2016. Multiphase SPH simulation for interactive fluids and solids. *ACM Trans Graph* 35, 4 (2016), 79.

T. Yang, J. Chang, M.C. Lin, R. Martin, J. Zhang, and S.M. Hu. 2017. A Unified Particle System Framework for Multi-Phase, Multi-Material Visual Simulations. *ACM Trans Graph* 36, 6 (2017).

T. Yang, J. Chang, B. Ren, M.C. Lin, J.J. Zhang, and S.M. Hu. 2015. Fast multiple-fluid simulation using Helmholtz free energy. *ACM Trans Graph* 34, 6 (2015), 201.

Y. Yue, B. Smith, C. Batty, C. Zheng, and E. Grinspun. 2015. Continuum foam: a material point method for shear-dependent flows. *ACM Trans Graph* 34, 5 (2015), 160:1–160:20.

F. Zhang, X. Zhang, K.Y. Sze, Y. Lian, and Y. Liu. 2017. Incompressible material point method for free surface flow. *J Comp Phys* 330 (2017), 92–110.