Constraint Bubbles and Affine Regions: Reduced Fluid Models for Efficient Immersed Bubbles and Flexible Spatial Coarsening

RYAN GOLDADE, University of Waterloo, Canada
MRIDUL AANJANEYA, Rutgers University, USA
CHRISTOPHER BATTY, University of Waterloo, Canada

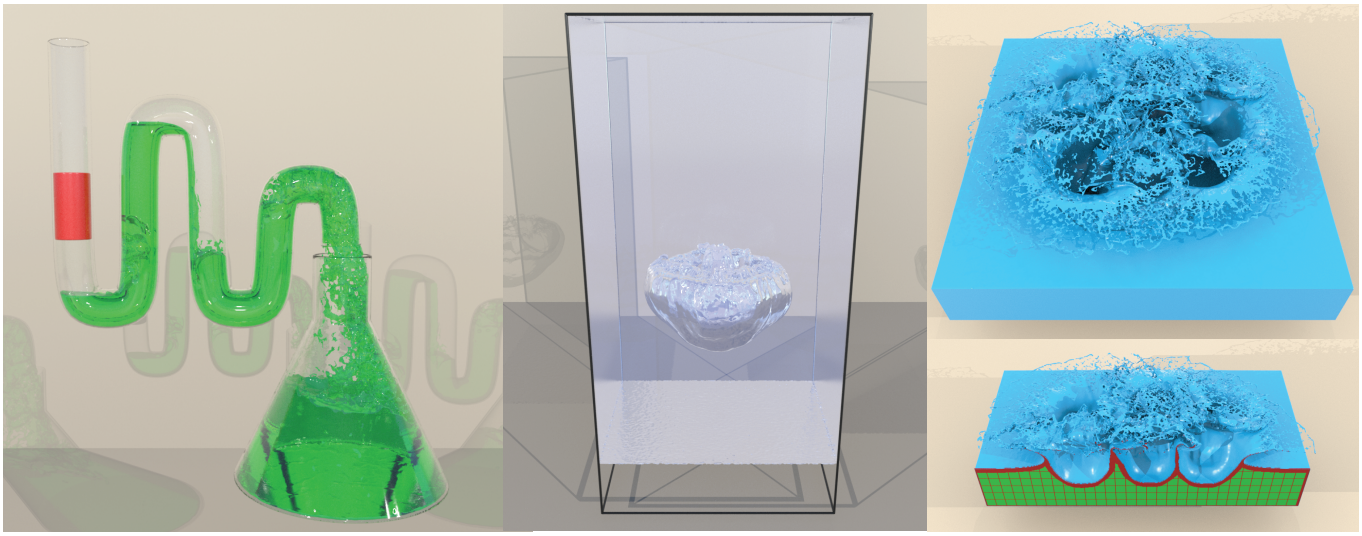


Fig. 1. (Left) Our *constraint bubble* model allows distinct liquid (green) and solid (red) bodies to physically interact across completely unsimulated air gaps. (Middle) Immersed bubbles denser than water correctly sink, despite the interior degrees of freedom being radically reduced with our *affine region* model. (Right) Our affine region model also enables a convenient and flexible approach to liquid adaptivity with irregularly shaped coarse tiles (green).

We propose to enhance the capability of standard free-surface flow simulators with efficient support for immersed bubbles through two new models: constraint-based bubbles and affine fluid regions. Unlike its predecessors, our constraint-based model entirely dispenses with the need for advection or projection inside zero-density bubbles, with extremely modest additional computational overhead that is proportional to the surface area of all bubbles. This surface-only approach is easy to implement, realistically captures many familiar bubble behaviors, and even allows two or more distinct liquid bodies to correctly interact across completely unsimulated air. We augment this model with a per-bubble volume-tracking and correction framework to minimize the cumulative effects of gradual volume drift. To support bubbles with non-zero densities, we propose a novel reduced model for an irregular fluid region with a single pointwise incompressible affine vector field. This model requires only 11 interior velocity degrees of freedom per affine fluid region in 3D, and correctly reproduces buoyant, stationary, and sinking behaviors of a secondary fluid phase with non-zero density immersed in water. Since the pressure projection step in both the above schemes is a slightly modified Poisson-style system, we propose novel Multigrid-based preconditioners for Conjugate Gradients for fast numerical solutions of our new discretizations. Furthermore, we observe that by enforcing an incompressible affine vector field over a coalesced set of grid cells, our reduced model is effectively an irregular coarse super-cell. This offers a convenient and flexible adaptive coarsening strategy that integrates readily with the standard staggered grid approach for fluid simulation, yet supports coarsened regions that are arbitrary voxelized shapes, and provides an analytically divergence-free interior. We demonstrate its effectiveness with a new adaptive liquid simulator whose interior regions are coarsened into a mix of tiles with regular and irregular shapes.

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

Additional Key Words and Phrases: liquid, adaptivity, two-phase, reduced model

ACM Reference Format:
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, Article 43 (July 2020), 15 pages. https://doi.org/10.1145/3386569.3392455

Authors' addresses: Ryan Goldade, University of Waterloo, 200 University Ave W., Waterloo, ON, N2L 3G1, Canada, rgoldade@uwaterloo.ca; Mridul Aanjaneya, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ, 08854, USA, mridul.aanjaneya@rutgers.edu; Christopher Batty, University of Waterloo, 200 University Ave W., Waterloo, ON, N2L 3G1, Canada, christopher.batty@uwaterloo.ca.

0730-0301/2020/7-ART43 $15.00
https://doi.org/10.1145/3386569.3392455

1 INTRODUCTION

As the visual fidelity of commercially available liquid animation software has improved over the last decade or so, there has been increased interest in capturing the delightful and characteristic behaviors of immersed bubbles. Whether considering the glugging of soda poured from a bottle or the swirling air pockets entrained by a splashy flow, we concur with the titular observation of Greenwood and House's early work in this domain: liquid animation is simply *better with bubbles* [Greenwood and House 2004]. Unfortunately, bubbles and related two-phase phenomena have quite often been neglected in practical liquid solvers because the computational cost is significant. One must typically solve for the entire air region's motion, despite its interior velocities being invisible and often largely irrelevant to the surface motion. Furthermore, water and air differ in density by about three orders of magnitude, leading to ill-conditioned linear systems that strain numerical solvers (e.g., [MacLachlan et al. 2008]).

Ideally, one would like to completely avoid simulating the air, or at least drastically reduce the degrees of freedom spent on capturing it. It is standard in computer graphics to ignore the air and assume a *free-surface* boundary condition at the interface [Bridson 2015]. Unfortunately, this treats air as a massless void that collapses when entrained by the liquid, because there is no opposing force preserving its volume. Some lightweight solutions for the air that partially addressed this issue were proposed in prior work, either implicitly through a costly stream function model [Ando et al. 2015], or explicitly with a weakly coupled compressible model that still requires advecting and projecting the air [Aanjaneya et al. 2013; Patkar et al. 2013]. In the present work, we address this challenge with two new models: *constraint bubbles* and *affine regions*.

We first propose a constraint-based model for negligible-density bubbles that considers only the net flux into (and out of) a given bubble. It entirely dispenses with advection and projection inside the bubble, and requires just one Lagrange multiplier per distinct bubble. This surface-only approach entails minimal overhead, it easily integrates with existing free-surface flow solvers, and yet it realistically captures many familiar bubble behaviors. It can even allow for two or more distinct liquid bodies to correctly interact across completely unsimulated air, such as water volumes separated by air in a tube, which is not possible with the schemes noted above [Aanjaneya et al. 2013; Ando et al. 2015]. We augment this model with a per-bubble volume-tracking and correction framework in order to minimize the effects of gradual volume drift.

While nearly zero-density bubbles is a very common situation, non-zero density coefficients also arise in many compelling two-phase scenarios. Therefore, our second contribution is a novel reduced model for an irregularly shaped fluid region over which we assume a single pointwise incompressible affine vector field. This approach requires only 11 interior velocity degrees of freedom per affine fluid region in 3D, and naturally handles the general case of non-zero density coefficients. Applied to simplify the interior of a secondary fluid phase immersed in water, our algorithm allows a sphere of neutrally buoyant immiscible liquid to remain perfectly stationary, while a heavier immiscible liquid sphere will correctly sink. Thus, constraint bubbles and affine regions provide complementary treatments for two-phase scenarios: constraint bubbles offer a simple and efficient mechanism for zero-density bubbles, while affine regions accelerate the traditional ghost-fluid two-phase approach [Hong and Kim 2005; Kang et al. 2000] for non-negligible density settings by radically reducing the number of interior degrees-of-freedom.

Next, we observe that since affine regions enforce a single incompressible vector field over a coalesced set of grid cells, we have in effect constructed an *irregular coarse super-cell* — this idea generalizes beyond bubbles to enable a new adaptivity strategy. Similar in spirit to prior schemes exploiting tall-cell grids [Chentanez and Müller 2011; Irving et al. 2006], stretched grids [Zhu et al. 2013], and octree grids [Losasso et al. 2004], our proposed technique offers a convenient and flexible coarsening strategy that integrates readily with the usual staggered uniform grid; yet unlike its predecessors, it supports coarsened regions that are *arbitrary voxelized shapes* and provides an *analytically divergence-free interior*. Compared to octree, tetrahedra, or nested grid adaptivity schemes our method requires simpler data structures and minimal overhead, and is more flexible than axially stretched or tall-cell approaches. We demonstrate its effectiveness with a new adaptive free-surface liquid solver whose interior regions are coarsened into a mix of tiles with regular and irregular shapes.

The pressure projection step in all of the above schemes can be reduced down to solving a symmetric and positive definite (SPD) linear system, which suggests the potential to develop fast numerical solvers. Therefore, to achieve efficiency in practice, we propose several Multigrid-based preconditioning schemes that are tailored to the particular characteristics of our new discretizations.

In summary, our contributions are the following:

- A constraint-based model for immersed zero-density bubbles, that entirely avoids advection/projection inside the bubble,
- A reduced fluid model based on pointwise divergence-free affine vector fields that supports irregularly-shaped regions,
- A tile-based strategy for flexible and adaptive spatial coarsening that is built on top of the above affine model, with applications to single- and multi-phase flow simulations,
- Efficient Multigrid-based preconditioners for Conjugate Gradients that are tailored to the preceding models, and
- A region-tracking and volume-correction strategy to compensate for drift in multi-component, topology-changing flows.

2 RELATED WORK

We focus our review on the grid-based fluid simulation approaches most relevant to our work; see [Bridson 2015] for an overview. Although outside the scope of our work, smoothed particle hydrodynamics approaches for two-phase flow (e.g., [Müller et al. 2003; Solenthaler and Pajarola 2008]) and adaptivity (e.g., [Adams et al. 2007; Solenthaler and Gross 2011]) have also been widely explored.

Multiphase Flow. Popular immiscible two-phase flow methods in computer graphics primarily derive from the boundary condition-capturing approach of Kang et al. [2000]. This approach simulates both air and fluid, enforcing incompressibility through a pressure projection scheme that treats the discontinuous jump in fluid density

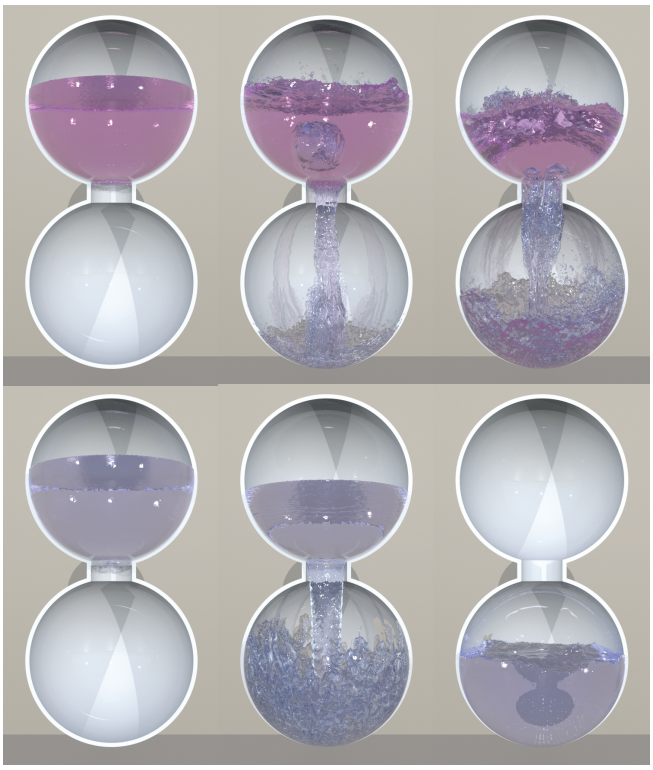


Fig. 2. (Top) Our constraint-bubble approach recovers the familiar *glugging* motion of liquid pouring through the neck of the water cooler, without simulating the air region. (Bottom) A standard free-surface liquid simulation fails to recreate the expected motion.

sharply at the air-water interface using a ghost fluid method. Hong and Kim [2005] first made use of this scheme, with a variety of subsequent enhancements following later [Boyd and Bridson 2012; Kim et al. 2007; Losasso et al. 2006; Mihalef et al. 2006]. The work of Kim et al. [2007] is particularly relevant as it focuses on animating bubbles, but it differs from our work in that their air bubbles are all fully simulated. In contrast to these sharp interface approaches, authors such as Song et al. [2005] and Zheng et al. [2006] have used a continuous variable-density pressure solve to simulate multiphase flow, also referred to as a diffuse interface approach.

Particle Bubbles. The use of secondary sub-grid scale particles is another natural way to add bubble details to free-surface flows; an early example is the work of Greenwood and House [2004]. More recent instances of this strategy include the work of Hong et al. [2008] and Busaryev et al. [2012]. An interesting recent hybrid is the approach of Patkar et al. [2013], which unifies the treatment of sub-grid and grid-scale compressible bubbles to allow tiny bubbles to oscillate and also coalesce into larger ones. An approach along these lines is likely compatible with our method, but we restrict our attention to grid-resolvable bubbles.

Augmenting Free-Surface Models. Our work is closely related to bubble simulation methods that augment a free-surface flow solver with partially decoupled or fully unsimulated grid-scale bubbles. Aanjaneya et al. [2013] proposed an equation of state approach to simulate two-way coupling of an incompressible liquid to a compressible, fully simulated air phase. They also suggested a simplified variant that assumes constant pressure in the air phase to approximate each bubble's influence with a single pressure degree of freedom and thereby partially decouple the air phase. This approach produces a linear system for liquid incompressibility with a similar structure to our constraint bubbles. However, it involves extra terms to support air compressibility and it assumes that bubbles possess non-negligible air mass that must be tracked. This necessitates a secondary pressure projection within each bubble and conservative advection for the air mass. As such, the method's computational cost scales with the full domain volume, whereas our constraint-bubbles method scales with the liquid volume. Furthermore, despite air density being present in the equations, we show later that the constant pressure model fails unless the bubbles have very low density. Lastly, while accurate bubble oscillations are critical to sound generation (e.g., [Langlois et al. 2016; Zheng and James 2009]), they are irrelevant for a wide range of strictly visual applications, so we prefer a fully incompressible treatment.

Ando et al. [2015] proposed a *stream function* approach for free-surface flows which expresses the pressure projection in terms of a vector potential. Standard vector calculus identities ensure that this representation provides incompressible velocities for the air by construction, even while assuming air density of zero and without simulating air at all. This approach is quite elegant, yet potentially less attractive in practice for a few reasons. First, the stream function approach entails a radically different and relatively complex discretization compared to standard solvers, requiring many standard fluid solver features be re-developed from the ground up. Second and more fundamentally, because the stream function is a three-component vector, the resulting linear systems are vector Poisson problems three times as large as the usual scalar Poisson problem for pressure projection, and are therefore significantly slower to solve. The method we propose instead requires only one extra degree of freedom per bubble and a small additional computational cost over a standard pressure projection. Our constraint-bubble method also supports at-a-distance interactions with solids or between two liquid bodies, mediated only through the unsimulated air. This is not naturally supported by the stream function approach [Ando et al. 2015], since the air region contains no stream function degrees-of-freedom that could enable the disjoint bodies to communicate.

Solid-Fluid Coupling. The manner in which we couple grid-based fluids to our affine fluid model is related to monolithic (or strong) two-way solid-fluid coupling, especially in the rigid-body case. Specifically, as noted by Jiang et al. [2015], an affine vector field has a similar but more general structure compared to a rigid-body vector field. Thus, requiring matching normal fluxes at the boundary between the regular fluid and the reduced affine regions leads to linear systems that have parallels with rigid-body coupling. Klingner et al. [2006] first considered strong coupling of rigid bodies on tetrahedral meshes, though subsequent work on monolithic coupling has largely focused on regular grid approaches. Batty et al. [2007]

considered rigid bodies, Robinson-Mosher et al. [2009; 2008] considered rigid bodies, volumetric elastic objects and thin shells, and Lu et al. [2016] considered reduced deformable solids. Aanjaneya [2018] recently proposed a fast solver for rigid-body coupling making use of Multigrid ideas. Another coupling approach is that of Golas et al. [2012], who used weak coupling between a vortex particle domain and a regular grid-based fluid simulator.

Adaptivity and Irregular Elements. Spatial adaptivity has a long history in liquid simulation. Many different strategies for introducing adaptivity have been proposed in the graphics literature, including octrees [Losasso et al. 2004], tetrahedra [Batty et al. 2010; Klingner et al. 2006], tall cells [Chentanez and Müller 2011; Irving et al. 2006], warped grids [Ibayashi et al. 2018], Chimera grids [English et al. 2013], and Voronoi or Power diagrams [Aanjaneya et al. 2017; Brochu et al. 2010; de Goes et al. 2015], among others. Some advantages of our approach include relative simplicity of implementation, flexibility of element shapes, and exactly divergence-free interior velocities. The use of p-adaptivity (i.e., elements with varying degree polynomials) in computer graphics has been explored in a discontinuous Galerkin (DG) framework [Edwards and Bridson 2014], allowing for highly detailed surfaces to be simulated; while we considered only affine fields, polynomial extensions of our method are also possible. The model of Edwards and Bridson [2014] assumes a standard polynomial basis, but a variety of pointwise divergence-free DG schemes have been developed outside of computer graphics (e.g., [Rhebergen and Wells 2018]). Traditional finite element methods have considered divergence-free elements [Gustafson and Hartman 1983] as well. There has also been recent interest in finite element variants that support general polyhedral elements [Edwards and Bridson 2014; Manzini et al. 2014; Martin et al. 2008]. Our method has the practical advantage of being easy to incorporate into standard fluid animation tools based on regular grid finite volumes.

Fluid Model Reduction and Boundary-Based Approaches. Model reduction has been applied to smoke animation problems [Cui et al. 2018; De Witt et al. 2012; Treuille et al. 2006], by simulating in a reduced, divergence-free basis. The basis is typically global, specific to a particular domain, and precomputed for efficiency. To extend reduced fluid models to large domains, Wicke et al. [2009] considered tiling the space and applying coupling between tiles. Our incompressible affine fluid model could be viewed as a particularly convenient, reduced (low order polynomial) basis, constructed locally on the fly at each time step. Our reduced bubble models also share a philosophical connection with recent boundary-based models for fluids, in that the goal is to minimize the use of interior volumetric degrees of freedom. Keeler and Bridson [2014] applied boundary integral techniques to ocean dynamics, and Da proposed both a vortex-based model for soap bubbles [Da et al. 2015], and a boundary element approach to surface-only liquids [Da et al. 2016].

Volume Correction. Small errors in surface tracking and incompressibility inevitably lead to volume changes in liquid simulation. Kim et al. [2007] first proposed to use divergence sources [Feldman et al. 2003] to recover lost volume. In particle-based models, direct particle position/density corrections have also been incorporated [Kugelstadt et al. 2019; Losasso et al. 2006; Takahashi and Lin 2019]. In the absence of particles, global corrections can lead to the wrong material component being adjusted. In a mesh-based surface tracking context, Thürey et al. [2010] used explicit topological change information to update per-component volume targets. In a volume-of-fluid context, Langlois et al. [2016] tracked bubble identities and volumes using scalar fields in order to use their volumes for sound generation. These authors did not describe how to redistribute volume, especially when multiple kinds of topological changes occur simultaneously among nearby bubbles.

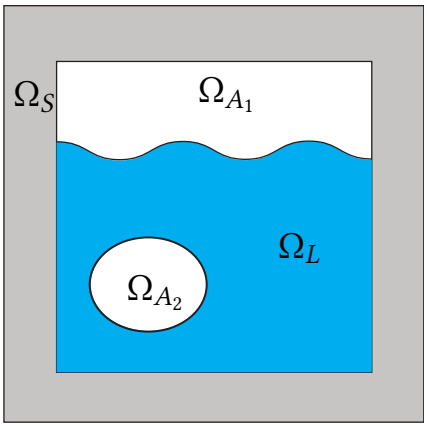


Fig. 3. Simulation domain with air Ω_A, solid Ω_S, and liquid Ω_L regions.

3 BACKGROUND

Our work builds upon standard grid-based liquid animation methods, to which Bridson [2015] provides a thorough introduction. In this context, liquid incompressibility is enforced by the pressure projection step which projects a given velocity field onto the subspace of divergence-free velocity fields. This can be expressed as solving the PDE

$$\begin{aligned} \rho \frac{\partial \boldsymbol{u}}{\partial t} &= -\nabla p, \\ \nabla \cdot \boldsymbol{u} &= 0, \end{aligned} \tag{1}$$

over the liquid domain, subject to $p = 0$ on free surfaces ("air") and the free-slip condition $\boldsymbol{u} \cdot \boldsymbol{n} = \boldsymbol{u}_S \cdot \boldsymbol{n}$ at solid boundaries. In these expressions, $\boldsymbol{u}$ is the liquid velocity, p is the fluid pressure enforcing incompressibility, t is time, ρ is fluid density, $\boldsymbol{u}_S$ is the solid boundary's velocity, and $\boldsymbol{n}$ is the solid boundary's normal. In what follows, unless otherwise stated, we use $\frac{g}{cm^3}$ as the measurement unit for all density values.

4 CONSTRAINT BUBBLES

4.1 Continuous Setting

Our constraint-based bubble model augments the standard pressure projection with support for incompressible air bubbles. As shown in Figure 3, we divide the simulation volume into three material domains, Ω_A, Ω_S, and Ω_L, corresponding to air, solid and liquid regions, respectively. We indicate a particular bubble by integer subscripts, e.g., Ω_{A_1}, Ω_{A_2}, etc. We will refer to any closed continuous

air region as a "bubble". A single liquid region may contain zero or more bubbles within it. A liquid region may also be entirely surrounded by a single "bubble"; that is, we make no distinction between exterior air and submersed air, viewing all as bubbles. Bubble and liquid regions may also be arbitrarily nested.

Our desired behavior is that each bubble preserves its total volume. For the i^{th} bubble, we can express this as a linear velocity constraint,

$$\iint_{\partial\Omega_{A_i}} \boldsymbol{u}_A \cdot \boldsymbol{n}\, dA = 0, \tag{2}$$

which we can separate into liquid and solid parts:

$$\underbrace{\iint_{\Omega_L \cap \partial\Omega_{A_i}} \boldsymbol{u} \cdot \boldsymbol{n}\, dA}_{\mathrm{B}_i(\boldsymbol{u})} + \underbrace{\iint_{\Omega_S \cap \partial\Omega_{A_i}} \boldsymbol{u}_S \cdot \boldsymbol{n}\, dA}_{b_{S_i}} = 0. \tag{3}$$

That is, the integrated flow through the entire boundary of a single continuous bubble region, Ω_{A_i}, must be zero. Enforcement of this constraint involves information about the velocity field everywhere on the bubble's boundary (i.e., either liquid or solid velocities touching the bubble), so we have denoted the contribution of the liquid surface to the i^{th} bubble as the linear operator $\mathrm{B}_i(\boldsymbol{u})$ and the prescribed solid's contribution as b_{S_i}. Crucially however, no information about velocities *interior* to the bubble is required. In some ways this is unsurprising; since we have assigned zero mass to the bubble, its momentum is negligible and therefore a model for the velocity interior to the bubble can be completely avoided.

Collecting all of the bubble constraints into a single vector operator B, our PDE takes the form

$$\begin{aligned} \rho\frac{\partial \boldsymbol{u}}{\partial t} &= -\nabla p - \frac{\partial \mathrm{B}}{\partial \boldsymbol{u}}^{\mathsf{T}} \lambda, \\ \nabla \cdot \boldsymbol{u} &= 0, \\ \mathrm{B}(\boldsymbol{u}) &= -b_S, \end{aligned} \tag{4}$$

where λ is the vector of Lagrange multipliers having one component per bubble.

4.2 Discrete Setting

We begin by directly discretizing the single-phase PDE (1) on a standard MAC grid [Harlow and Welch 1965] in a finite volume fashion, yielding the following indefinite linear system:

$$\begin{pmatrix} \frac{1}{\Delta t}\mathbf{M} & \mathbf{D}^{\mathsf{T}} \\ \mathbf{D} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \frac{1}{\Delta t}\mathbf{M}\mathbf{u}^* \\ \mathbf{0} \end{pmatrix}. \tag{5}$$

Here $\mathbf{p}$ is the vector of (discrete) pressures, and $\mathbf{u}^*$ and $\mathbf{u}$ are the vectors of face-normal velocity components before and after projection, respectively. $\mathbf{M}$ and $\mathbf{D}$ are the usual diagonal fluid mass matrix and discrete divergence operator, which can be straightforwardly adapted to simultaneously incorporate irregular free-surfaces via ghost fluid [Enright et al. 2003] and irregular solid walls via cut-cells [Batty et al. 2007; Ng et al. 2009]. Our examples employ this approach. (Note that diagonal entries of $\mathbf{M}$ are zero for entirely air and solid faces, so the corresponding rows and columns drop out.)

We use a row-vector $\mathbf{B}_i$ to represent the discretization of the i^{th} bubble constraint from (3), which sums the net flow across the bubble's incident liquid faces such that:

$$\mathbf{B}_i\mathbf{u} = \sum_{\text{liquid faces of } \partial\Omega_{A_i}} \mathbf{u} \cdot \mathbf{n}_{\text{face}}\, \mathrm{dA}_{\text{face}}. \tag{6}$$

In this expression, $\mathbf{n}_{\text{face}}$ is the cell face-normal oriented out of the bubble region, and $\mathrm{dA}_{\text{face}}$ is the area of the relevant face. (If a cut-cell methodology is used [Batty et al. 2007; Ng et al. 2009], one should account for only the partial area outside of solids.) Effectively, this constraint measures the aggregate discrete divergence for the entire bubble; its corresponding multiplier λ will act as a collective pseudo-pressure enforcing that its volume is unchanging. Since $\mathbf{B}_i$ only involves liquid velocities touching the bubble, it is relatively sparse.

If the bubble touches any kinematically scripted moving solids, we appropriately modify the right hand side of (6) to add contributions from the surfaces of those solids, i.e.,

$$b_{S_i} = \sum_{\text{solid faces of } \partial\Omega_{A_i}} \mathbf{u}_S \cdot \mathbf{n}_{\text{face}}\, \mathrm{dA}_{\text{face}}. \tag{7}$$

Doing so allows moving solids to affect even liquid surfaces that they are *not in direct physical contact with*, such as when an air bubble in an enclosed tube separates a liquid from a moving piston: the force is communicated through the bubble, as expected (see Figure 4). The same ideas can be extended in a straightforward fashion to model interactions with strongly coupled, dynamic rigid or deformable bodies [Batty et al. 2007; Robinson-Mosher et al. 2009].

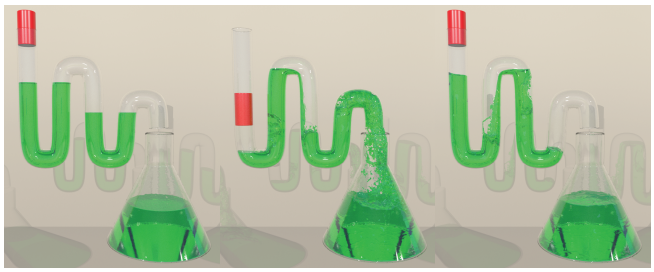


Fig. 4. Our constraint-based approach allows two distinct liquid regions to correctly interact across completely unsimulated air. A moving piston pushes liquid through the tube indirectly through constraint-bubble regions.

Stacking the bubble constraints into a single wide matrix $\mathbf{B}$, and incorporating them into (5) yields a large sparse symmetric indefinite linear system that is the discrete version of (4):

$$\begin{pmatrix} \frac{1}{\Delta t}\mathbf{M} & \mathbf{D}^{\mathsf{T}} & \mathbf{B}^{\mathsf{T}} \\ \mathbf{D} & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \\ \lambda \end{pmatrix} = \begin{pmatrix} \frac{1}{\Delta t}\mathbf{M}\mathbf{u}^* \\ \mathbf{0} \\ -\mathbf{b}_S \end{pmatrix}. \tag{8}$$

This indefinite system includes the pressure, constraint, and velocity degrees-of-freedom and is therefore quite large. However, since $\mathbf{M}$ is diagonal and hence trivially invertible, we can take the Schur complement (i.e., solve the first row for $\mathbf{u}$ and substitute into the latter two rows) to eliminate velocity and arrive at a smaller SPD system in terms of pressure and the bubbles' Lagrange multipliers:

$$\begin{pmatrix} \Delta t\mathbf{D}\mathbf{M}^{-1}\mathbf{D}^{\mathsf{T}} & \Delta t\mathbf{D}\mathbf{M}^{-1}\mathbf{B}^{\mathsf{T}} \\ \Delta t\mathbf{B}\mathbf{M}^{-1}\mathbf{D}^{\mathsf{T}} & \Delta t\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^{\mathsf{T}} \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{D}\mathbf{u}^* \\ \mathbf{B}\mathbf{u}^* + \mathbf{b}_S \end{pmatrix}. \tag{9}$$

After solving this linear system for $\mathbf{p}$ and λ, the final velocity $\mathbf{u}$ can be obtained using the first row of (8). Since $\mathbf{M}$ is diagonal, this

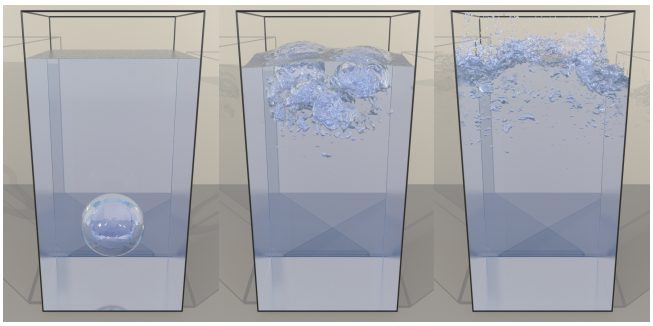
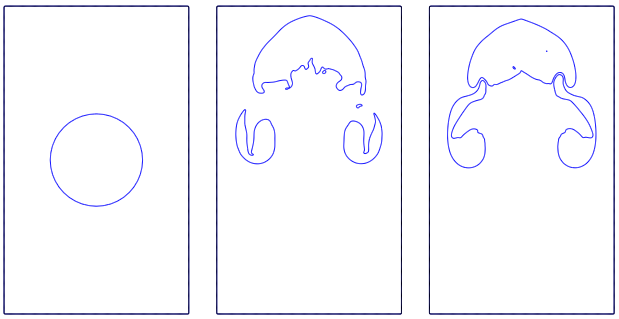


Fig. 5. A zero-density air bubble inside a liquid column rises and breaks apart into many small bubbles. Our per-bubble volume tracking and correction framework allows small bubbles to persist over long periods of time, across complex topology changes, and without explicit air particles.



(a) Initial single bubble (b) Full two-phase (c) Constraint bubble

Fig. 6. Our zero-density constraint-based simulation closely resembles a two-phase simulation with a small air density of $\rho = 1$. The liquid density is $\rho = 1000$ for both simulations.

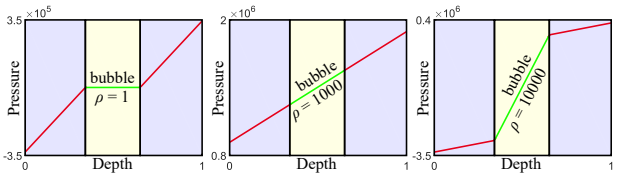


Fig. 7. Ideal pressure profiles for vertical one-dimensional multiphase fluids in hydrostatic balance with each other for three different density ratios. The pale yellow region is the immersed bubble at $\rho = 1$, $\rho = 1000$ and $\rho = 10000$, while the pale magenta region has constant density $\rho = 1000$. A single constant pressure is a good approximation for the bubble region in the low-density (left) case, but would be inaccurate for the other cases.

amounts to a simple matrix-vector multiply. The upper-left block of (9) is the usual Poisson system and the remaining blocks account for interaction with the bubble constraints. Compared to the standard pressure solve, the extra Lagrange multipliers have added one row and one column per bubble.

Our system has a similar structure to the one that arises in the compressible flow method of Aanjaneya et al. [2013], but assumes zero density bubbles, does not require terms related to bubble expansion or compression, and supports scripted moving objects. Most importantly, we do not require a second advection step or pressure solve to determine the (visually imperceptible) interior air motion, which allows us to cheaply simulate very large regions of empty air. For small air densities, our constraint method also closely agrees with a full two-phase flow simulation (see Figure 6).

Identifying the set of individual bubble regions can be done by determining connected components through a simple flooding approach over air cells that share faces. The flooding must be done over the air *volume*, rather than simply over connected *surfaces*, so that any nesting of regions is properly identified and handled. For example, a single bubble might contain a disconnected interior droplet; in this case the bubble should have a single volume constraint accounting for both of these surfaces.

For the purposes of the advection phase, we perform standard extrapolation of the liquid velocity field into the empty air region [Enright et al. 2002].

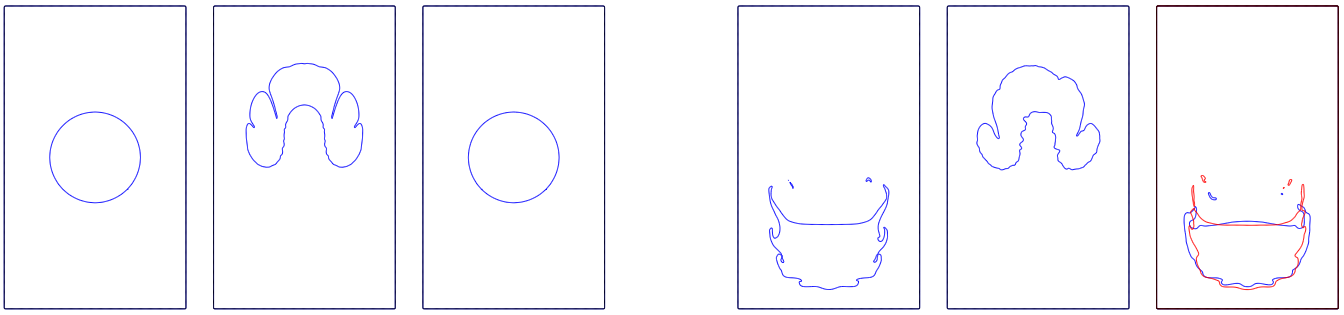
5 AFFINE REGIONS

5.1 Motivation

As our results will demonstrate, the preceding reduced model is ideal for the common case of (approximately) zero-density bubbles, since it requires only one extra degree-of-freedom per bubble and no model whatsoever for the interior air. However, if one is interested in animating two-phase flows with more general density coefficients, so that the immersed phase rises more slowly, remains neutrally buoyant, or even sinks, this choice is insufficient. That is because its Lagrange multipliers are equivalent to one constant pressure value on each bubble's interior, even if one incorporates the bubble's desired density into the equations (as the more complex compressible model of Aanjaneya et al. [2013] attempts to do). Examining the expected pressure profile of a 1D two-phase flow scenario in a vertical column at hydrostatic balance (Figure 7) provides useful intuition. For very low bubble densities, a single constant interior pressure value is a good approximation of the true interior pressure field. However, for moderate to high bubble density coefficients, a constant pressure cannot produce the correct velocity update.

This observation initially led us to consider imposing a *linear* model of interior pressure based on the liquid pressures surrounding the bubble. Since a linear pressure field induces a constant pressure gradient (i.e., velocity update) it can correct the global translation of the bubble, and for example, recover hydrostatic balance for neutrally buoyant bubbles. However, a constant velocity correction is still too limited in the vector fields it can describe. For example, a bubble rising to collide with a boundary cannot spread out in opposing directions. While a higher order pressure model might yield better results, we also found this framework somewhat unwieldy and had difficulty preserving symmetry. Moreover, the fact that the bubble density is no longer negligible in this setting suggests that a model for interior momentum will be necessary.

Therefore, rather than focusing on pressure and ignoring the interior velocity field, we adopt an explicit model for the *velocity* inside a coalesced region by assuming it to be both incompressible and affine. This turns out to have several attractive properties, including ease



(a) Full two-phase ($\rho_B = \rho_L = 10^3$) (b) Constraint bubble ($\rho_B = \rho_L = 10^3$) (c) Affine bubble ($\rho_B = \rho_L = 10^3$) (d) Full two-phase ($\rho_B = 10^4, \rho_L = 10^3$) (e) Constraint bubble ($\rho_B = 10^4, \rho_L = 10^3$) (f) Affine bubble ($\rho_B = 10^4, \rho_L = 10^3$)

Fig. 8. **Model Comparison in 2D**: When the immersed fluid (i.e., bubble) has non-negligible density, constraint bubbles yield incorrect motion compared to full two-phase simulations; however, affine region-based bubbles exhibit correct buoyancy. Initial conditions are a circular bubble; each image shows the same time instant shortly into the simulation. Left trio: A neutrally buoyant bubble should remain stationary. Right trio: A higher density bubble should sink. In (f), the blue simulation used a single interior affine region and the red simulation used interior tiling (§6) for improved fidelity.

of symmetry preservation, an analytically divergence-free interior flow, and a structure very similar to well-known solid-fluid coupling models, while still enabling us to abstract away large collections of adjacent fluid cells.

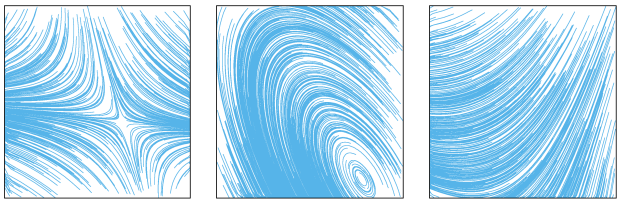


Fig. 9. Representative divergence-free affine velocity fields (plotted as streamlines) generated by our model, highlighting its expressiveness.

5.2 Continuous Setting

Consider a region of fluid Ω_B that is constrained to possess an incompressible affine velocity field. Such a body's velocity $\boldsymbol{u}_B$ at position $\boldsymbol{x}$ can be described by the relation:

$$\boldsymbol{u}_B(\mathbf{x}) = \boldsymbol{u}_{\text{const}} + \mathbf{A}(\boldsymbol{x} - \boldsymbol{x}_{\text{COM}}), \tag{10}$$

where $\mathbf{A} = \nabla \boldsymbol{u}_B$. That is, velocity any point in the affine velocity field is computed from the constant, translational velocity at the center of mass $\boldsymbol{x}_{\text{COM}}$ plus a linear correction dictated by the velocity gradient. (The center of mass is used for convenience, but any fixed reference point suffices.) Some representative examples (computed using (10)) are shown in Figure 9 to highlight the perhaps surprising expressiveness of divergence-free affine vector fields.

To enforce incompressibility, the field must satisfy the usual condition $\nabla \cdot \boldsymbol{u}_B = 0$. Applying this constraint to (10) shows that the velocity gradient must be trace-free, i.e., $\text{Tr}(\mathbf{A}) = 0$. This constraint leads to $\boldsymbol{A}$ having the following reduced form:

$$\mathbf{A}_{2D} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & -a_{11} \end{bmatrix}, \quad \mathbf{A}_{3D} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & -(a_{11} + a_{22}) \end{bmatrix}. \tag{11}$$

Our task now is to develop an understanding of the dynamics of a fluid body with such a velocity field. Notice that if we were to instead require the velocity field to satisfy $\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\mathsf{T}} = \mathbf{0}$, i.e., zero strain rate (see e.g., [Carlson et al. 2004]), we would recover a rigid body motion, where the velocity consists of translational and *rotational* parts. This is a useful analogy for understanding our model, similar to the relationship between the rigid (RPIC) and affine (APIC) particle-in-cell models proposed by Jiang et al. [2015].

For simplicity we consider the 2D case and denote our generalized velocity vector as $\mathbf{v}_B = [u, v, a_{11}, a_{12}, a_{21}]$. We can then define the matrix $\mathbf{C}(\boldsymbol{x})$ that extracts the Euclidean velocity $\boldsymbol{u}_B$ at a point $\boldsymbol{x}$:

$$\boldsymbol{u}_B = \mathbf{C}(\boldsymbol{x})\mathbf{v}_B \tag{12}$$

$$= \begin{bmatrix} 1 & 0 & x - x_{\text{COM}} & y - y_{\text{COM}} & 0 \\ 0 & 1 & -(y - y_{\text{COM}}) & 0 & x - x_{\text{COM}} \end{bmatrix} \mathbf{v}_B. \tag{13}$$

The 3D case follows straightforwardly. Under our rigid body analogy, we will also require a generalized mass matrix for the affine fluid body, which dictates how the body's generalized velocity $\mathbf{v}_B$ changes under applied forces. Since the kinetic energy of the fluid body can be described by the integral

$$\iiint_{\Omega_B} \frac{\rho_B}{2} \|\boldsymbol{u}_B\|^2 \, dV = \iiint_{\Omega_B} \frac{\rho_B}{2} \|\mathbf{C}\mathbf{v}_B\|^2 \, dV \tag{14}$$

$$= \tfrac{1}{2}\mathbf{v}_B^{\mathsf{T}} \left(\iiint_{\Omega_B} \rho_B \mathbf{C}^{\mathsf{T}} C \, dV \right) \mathbf{v}_B, \tag{15}$$

the symmetric positive definite matrix $\mathbf{M}_B = \iiint_{\Omega_B} \rho_B \mathbf{C}^{\mathsf{T}}\mathbf{C} \, dV$ is exactly our desired generalized mass matrix.

Our intention is to immerse this affine fluid body within a regular grid-based fluid solver, and enforce two-way coupling between them; we model our approach loosely on the rigid-body coupling

framework of Batty et al. [2007]. Therefore, our next task is to determine how the affine vector field of this fluid body is affected by the surrounding pressure p of the regular fluid. As in the rigid body case, the net translational component of the pressure force is simply the integral of pressure acting on the body's surface:

$$\boldsymbol{f}_{\text{const}} = \iint_{\partial\Omega_B} p\boldsymbol{n}\, dA, \tag{16}$$

where $\mathbf{n}$ is the affine fluid body's surface normal. We can similarly account for the effect on the velocity gradient components of $\mathbf{v}_B$ using

$$\boldsymbol{F}_{\text{linear}} = \iint_{\partial\Omega_B} p\boldsymbol{n}(\boldsymbol{x} - \boldsymbol{x}_{\text{COM}})^{\mathsf{T}}\, dA, \tag{17}$$

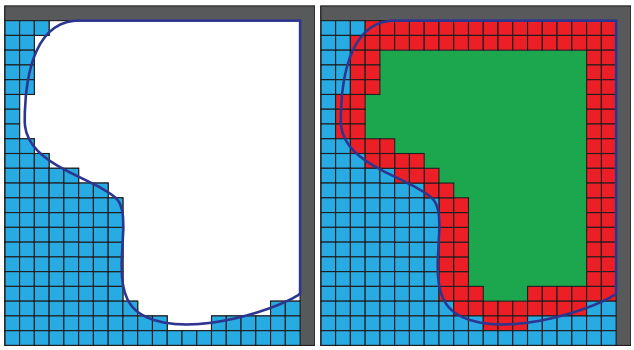
which plays a role analogous to the net torque on a rigid body. (The expressions above can be derived, for example, by differentiating the power that the surrounding fluid pressure applies on the boundary, $\iint_{\partial\Omega_B} \boldsymbol{u}_B \cdot (p\boldsymbol{n})\, dA$, by the affine velocity degrees of freedom.) By flattening F_{linear} into a vector to correspond with $\mathbf{v}_B$ and combining these two contributions, we form a linear operator $\mathbf{J}$ that transforms (regular grid) boundary fluid pressures into generalized forces on the affine fluid body's degrees of freedom. Together with our generalized mass matrix, we have an (Eulerian) update rule for our affine fluid body in terms of our reduced degrees of freedom: $\mathbf{M}_B \frac{\partial \mathbf{v}_B}{\partial t} = \mathbf{J}p$. (We have handled advection separately, hence this expression does not use the material derivative, $\frac{D\mathbf{v}_B}{Dt}$.)

Finally, if $\boldsymbol{u}_F$ denotes the velocity field of the surrounding regular fluid, the combined pressure projection PDE that we seek to solve is

$$\begin{aligned} \rho \frac{\partial \boldsymbol{u}_F}{\partial t} &= -\nabla p \\ \nabla \cdot \boldsymbol{u}_F &= 0 \\ \mathbf{M}_B \frac{\partial \mathbf{v}_B}{\partial t} &= \mathbf{J}p \end{aligned} \tag{18}$$

subject to $\boldsymbol{u}_F = \boldsymbol{u}_B$ on their shared boundary. Incompressibility of Ω_B does not appear explicitly in these expressions because it is enforced implicitly through the reduced velocity model. This PDE will take tentative values for $\boldsymbol{u}_F$ and $\mathbf{v}_B$ after advection and application of forces, and simultaneously ensure incompressibility of both regions while handling the exchange of forces between them. (Our approach can alternatively be derived by considering a variational perspective, as summarized briefly in Appendix A; this also emphasizes that the scheme is intrinsically symmetric.)

The motivating application for our reduced model is simulating entrained bubbles, but bubbles tend to be highly deformable near their interface, especially with low surface tension. As such, the affine model alone is inadequate for this case because it lacks the necessary flexibility to capture the rapidly emerging surface details formed during bubble break-up. However, the affine approximation *is* often effective for the smoother (and invisible) interior air motion away from the interface itself. We therefore adopt a standard ghost-fluid two-phase flow model [Hong and Kim 2005] to handle the interface conditions and a narrow interior band of air, while replacing only the slightly deeper interior air region(s) with our affine model. Figure 10 compares our grid setup for constraint bubbles and affine bubbles.



(a) Constraint bubble (b) Affine bubble

Fig. 10. Constraint bubbles use no interior fluid cells; affine bubbles pad the interior affine region with a thin band of uniform cells to capture detailed surface deformations. Legend: ● = liquid cells, ○= constraint bubble region, ● = exterior bubble cells, ● = interior affine region, ● = solid boundary.

5.3 Discrete Setting

To discretize the coupling problem above, we adopt a standard finite volume approach for the surrounding regular grid fluid. We assume that the boundary between the affine and regular grid regions lies exactly on grid-aligned faces, recalling that it does not correspond to a physical boundary but rather a change of representations. (However, one could adopt a cut-cell formulation [Ng et al. 2009] to generalize to irregular shapes, if this was deemed beneficial.)

Directly discretizing the PDE (18) yields a symmetric indefinite linear system:

$$\begin{bmatrix} \frac{1}{\Delta t}\mathbf{M}_F & \mathbf{D}^{\mathsf{T}} & \mathbf{0} \\ \mathbf{D} & \mathbf{0} & \mathbf{J}^{\mathsf{T}} \\ \mathbf{0} & \mathbf{J} & -\frac{1}{\Delta t}\mathbf{M}_B \end{bmatrix} \begin{bmatrix} \mathbf{u}_F \\ \mathbf{p} \\ \mathbf{v}_B \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t}\mathbf{M}_F \mathbf{u}_F^* \\ \mathbf{0} \\ -\frac{1}{\Delta t}\mathbf{M}_B \mathbf{v}_B^* \end{bmatrix}. \tag{19}$$

The first row is the usual velocity update for the regular fluid region Ω_F, where $\mathbf{M}_F$ is a diagonal mass matrix with entries corresponding to regular fluid faces and $-\mathbf{D}^{\mathsf{T}}$ is the pressure gradient operator across those faces. The second row represents the discrete divergence of regular fluid cells, with contributions from regular fluid faces given by $\mathbf{D}\mathbf{u}_F$ and from coupled affine fluid faces given by $\mathbf{J}^{\mathsf{T}}\mathbf{v}_B$. Finally, the third equation gives the effect of the pressure on the affine region's velocity. The variables $\mathbf{u}_F^*$ and $\mathbf{v}_B^*$ represent the tentative velocities after applying external forces and advection.

It remains only to define the discrete mass matrix $\mathbf{M}_B$ and discrete pressure force operator $\mathbf{J}$. The discrete mass matrix is

$$\mathbf{M}_B = \sum_{a,i} \rho \mathbf{C}_a(\mathbf{x}_i)^{\mathsf{T}} \mathbf{C}_a(\mathbf{x}_i)\, \mathrm{dV}, \tag{20}$$

where a iterates over the axis directions (x, y, z), i iterates over all faces in that axis having one or both of its two incident cells inside the affine region, $\mathbf{x}_i$ is the midpoint of the face, and dV is the volume of a cell. The notation $\mathbf{C}_a(\mathbf{x})$ indicates the row of $\mathbf{C}$ corresponding to the axis a, evaluated at $\mathbf{x}$. The discrete $\mathbf{J}$ operator (matrix) is

$$\mathbf{J}\mathbf{p} = \sum_{a,j} \mathbf{C}_a(\mathbf{x}_j)^{\mathsf{T}} p_j \mathbf{n}_j \mathrm{dA}, \tag{21}$$

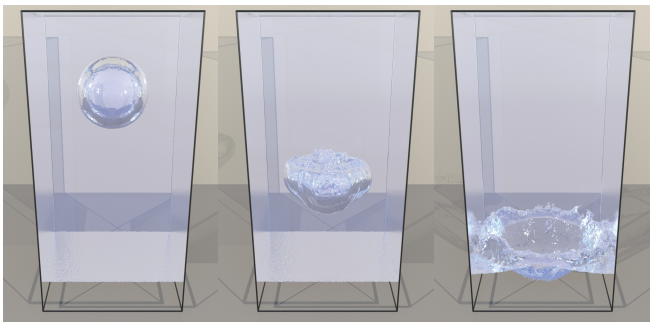


Fig. 11. A bubble with higher density than the surrounding liquid correctly sinks in 3D. Its interior is tiled with multiple affine regions, as in §6.

where a iterates over the 3 axis directions, j iterates over the faces *on the boundary* of the affine region, $\mathbf{x}_j$ is the center of the associated face, p_j is the pressure at the center of the incident regular fluid cell, $\mathbf{n}_j$ is the normal to that face, and dA is the area of a grid face.

To reduce the system size of (19), we can form the Schur complement to eliminate fluid velocity $\mathbf{u}_F$ as we did for constraint bubbles:

$$\begin{bmatrix} -\Delta t \mathbf{D}\mathbf{M}_F^{-1}\mathbf{D}^\mathsf{T} & \mathbf{J}^\mathsf{T} \\ \mathbf{J} & -\frac{1}{\Delta t}\mathbf{M}_B \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{v}_B \end{bmatrix} = \begin{bmatrix} -\mathbf{D}\mathbf{u}^* \\ -\frac{1}{\Delta t}\mathbf{M}_B\mathbf{v}_B^* \end{bmatrix}. \tag{22}$$

This system is still symmetric indefinite, with a form that closely mirrors the rigid-body coupling approaches of Robinson-Mosher et al. [2009, 2008]; as such, it will require an indefinite solver such as MINRES or QMR. However, if desired, a second Schur complement can be applied with respect to the block-diagonal lower-right block, which eliminates $\mathbf{v}_B$ and yields a smaller symmetric positive definite system in terms of fluid pressure alone:

$$-\Delta t(\mathbf{D}\mathbf{M}_F^{-1}\mathbf{D}^\mathsf{T} + \mathbf{J}^\mathsf{T}\mathbf{M}_B^{-1}\mathbf{J})\mathbf{p} = -\mathbf{D}\mathbf{u}^* - \mathbf{J}^\mathsf{T}\mathbf{v}_B^*. \tag{23}$$

This system's structure matches that of Batty et al. [2007] and is amenable to solution with Conjugate Gradients. However, this transformation comes at the cost of introducing the dense block $\mathbf{J}^\mathsf{T}\mathbf{M}_B^{-1}\mathbf{J}$, which mutually couples all pressures incident on the boundary of the affine region. Depending on the numerical strategies chosen to solve this problem, one of these two forms may be preferable. We defer discussion of our numerical solver to §7.2.

At each application of the above projection, we first recover the input $\mathbf{v}_B^*$ for the affine region using a simple least squares fit over the region's fluid grid faces. These are computed from standard advection on the grid (e.g., by semi-Lagrangian, particle-based, etc.) The least squares problem has the form

$$\arg\min_{\mathbf{v}_B^*} \sum_{a,k} \left(u_{a,k} - \mathbf{C}_a(\mathbf{x}_i)\mathbf{v}_B^*\right)^2, \tag{24}$$

where a iterates over the axes, i iterates over all grid faces of the affine region, and $\mathbf{x}_i$ is the face midpoint (velocity sample point).

We conclude this section by observing that, despite our original motivation being bubbles, our divergence-free affine fluid model makes no assumptions that are specific to the two-phase setting. It can therefore also be applied to the interior of a free-surface flow with the same benefit of reducing its active degrees-of-freedom.

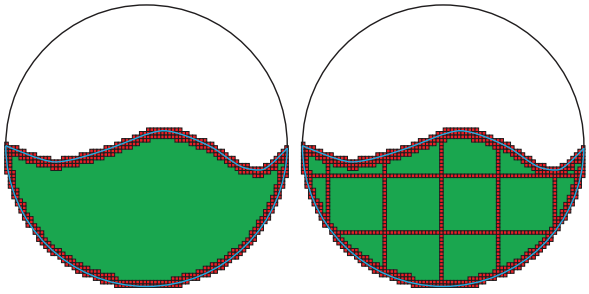


Fig. 12. Left: A free-surface flow with a single interior affine region (green). Right: The same flow subdivided into a mix of regular (square) and irregular affine tiles for greater flexibility.

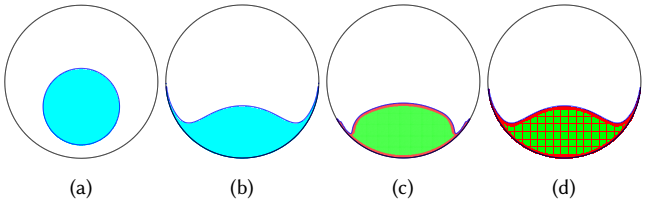


Fig. 13. (a) A 2D ball of free-surface fluid is released under gravity, and falls to collide with the circular domain boundary. (b) Ground truth uniform grid simulation. (c) A simulation with a *single* interior affine region (green) is insufficiently flexible for this scenario. (d) A simulation using *tiled* affine regions provides a much closer match to the ground truth.

6 TILED AFFINE REGIONS FOR ADAPTIVITY

The preceding affine model provides an effective approximation for the interior of fluid bodies with modest sizes, both for single-phase free-surface and multiphase flow scenarios. At the same time, a single affine region is inherently limited in the complexity of velocity fields that it can describe. Therefore, to further extend our model's usefulness to even larger interior regions, we propose a simple tiling strategy that uses multiple medium-sized affine regions, stitched together by thin layers of uniform regular grid cells. As can be seen in Figure 12, this leads to fine uniform cells in a band around interfaces, irregularly shaped affine tiles next to them, and finally large rectangular tiles filling the deeper regions. On the interior, a one-layer band of uniform cells is placed between affine tiles. Our tiling approach drastically reduces the degrees-of-freedom needed to capture a large domain, while providing a good qualitative match to fully uniform grid flows, as in Figures 13 and 14.

There are several additional advantages to this approach. Compared to an octree, tetrahedral mesh, or Chimera grid approach, it is much simpler to integrate into a regular grid simulation pipeline, since it requires no elaborate data structures or complex (re)meshing. It provides more effective and general coarsening than tall cells or stretched grids. Notably, it provides analytically divergence-free fields in the coarsened regions, a feature which may be useful for some applications that no competing adaptivity method for fluid animation provides. The $\mathbf{J}$ and $\mathbf{M}_B$ matrices needed for perfectly rectangular tiles can be precomputed and reused for efficiency.

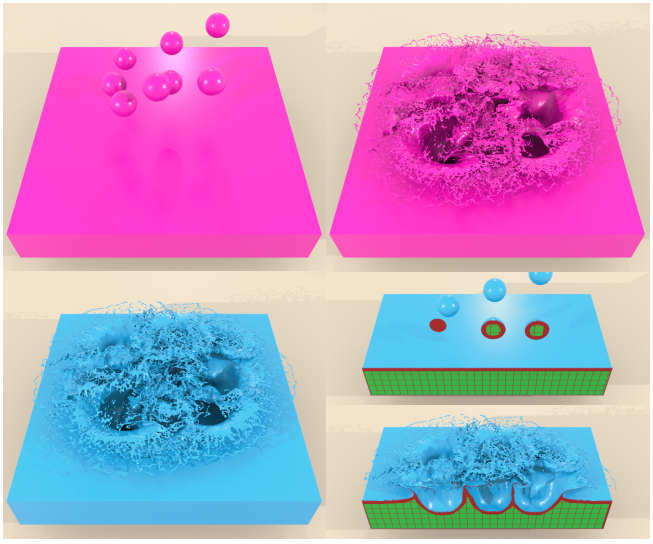
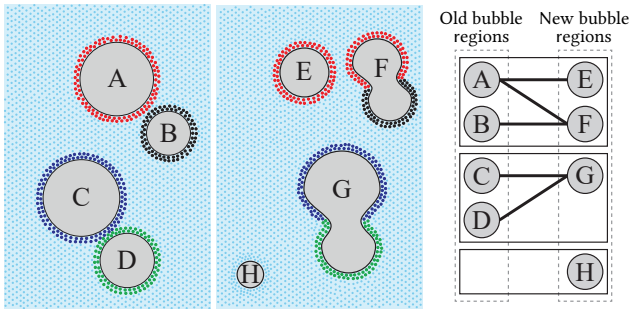


Fig. 14. A high resolution splashing scenario. Top-left: Initial conditions. Top-right: Uniform grid simulation. Bottom-left: A simulation with interior tiled affine regions offers a close qualitative match. Bottom-right: Cutaway view of the interior tiling.



(a) Start of step. (b) After advection. (c) Bubble graph.

Fig. 15. Based on empty regions and their surrounding particle labels (a) before and (b) after advection, we construct a bipartite graph (c) to conservatively redistribute target rest volumes. Two cases of note are: (1) bubble A splits while partially merging with B; (2) bubble H is a spurious void formed by minor particle advection errors, which should be collapsed away.

7 IMPLEMENTATION DETAILS

7.1 Volume Tracking and Correction

To foster adoption in existing industrial pipelines, we have intentionally chosen to extend the standard free-surface FLIP method [Zhu and Bridson 2005]. Whereas previous Eulerian (or hybrid) methods for bubbles employ multi-material level sets or particles to directly track *both* the liquid and bubble regions, we instead extend basic FLIP by simply labeling any simultaneously non-solid and non-liquid region as a bubble. However, it is well-known that accumulated numerical advection errors cause liquid FLIP particles to separate or condense over time [Kugelstadt et al. 2019], inducing erroneous volume change and occasionally creating spurious empty gaps or *voids*. Since we do not explicitly track the air geometry, liquid volume drift destructively modifies the implied bubble volume with it, while artificial voids give birth to false bubbles that begin to rise. In turbulent simulations along solid boundaries, this can even give the appearance of solid surfaces *boiling*. Explicit tracking of the bubble material could prevent voids and somewhat reduce volume change, but would also add nontrivial overhead. To address these issues, we propose to track bubbles implicitly by augmenting each FLIP particle with a new bubble ID attribute.

We build a mapping of bubble identities from one time step to the next using the *old* bubble IDs stored on the FLIP particles from the previous step and the bubble IDs assigned to *new* bubble regions that those particles end up next to in the current step. In Figure 15, particles are initially assigned the ID of their adjacent bubble and, after advection, the particle IDs are used to map the old bubble IDs to the new bubble IDs. This mapping forms a bipartite graph, where nodes represent bubble regions and edges indicate whether bubbles have simply advected or undergone more complex merging and splitting.

Removing Spurious Void Bubbles. Using this mapping, *void* bubbles correspond to new bubble ID nodes with no incoming edges. We seek to collapse them away by applying a negative (rather than zero) divergence [Feldman et al. 2003]. A bubble's volume change relates to its divergence by

$$\int_{\Omega_B} \nabla \cdot \boldsymbol{u} \approx \frac{V_B^{n+1} - V_B^n}{\Delta t}, \tag{25}$$

where a bubble is driven to collapse by setting $V_B^{n+1} = 0$.

Volume Correction. Equation 25 can also be used to correct volume drift by setting a bubble's target volume to its initial *rest* volume, $V_B^{n+1} = V_B^0$. For old-new bubble pairs having a one-to-one mapping, we simply copy the rest volume from the old bubble to the new. For more complex scenarios where bubbles have possibly split and/or merged, it is not possible to directly assign rest volumes. Instead, we find connected components in the bipartite graph of bubble mappings and redistribute the current rest volumes for all old bubble regions in the set. Updated rest volumes are assigned according to the volume of each new bubble relative to the accumulated volume of each new bubble region in the set:

$$V_{B_i}^0 = \frac{V_{B_i}^N}{\sum_{k \in G_i^N} V_{B_k}^N} \sum_{j \in G_i^0} V_{B_j}^0. \tag{26}$$

where $V_{B_i}^0$ and $V_{B_i}^N$ are the rest and current volumes of the i^{th} new bubble, and G_i^N and G_i^0 are the set of new and old bubbles in the connected component containing the i^{th} bubble, respectively.

This componentwise volume tracking and correction approach is very general in nature, and so can also be applied to the liquid-phase to good effect, or even free-surface FLIP simulations without bubbles.

7.2 Optimized Linear Solver

7.2.1 Faster Matrix-Vector Multiplication. As with rigid-body coupling, the affine contribution $\mathbf{J}^\mathsf{T}\mathbf{M}^{-1}\mathbf{J}$ in the SPD system (23) contains a dense block that mutually couples all cells on an affine region's

boundary, quickly becoming a bottleneck. We adopt the suggestion of Bridson [2015] to exploit this block's low rank [Batty et al. 2007] by storing it in factored form and, when needed for Conjugate Gradients, performing three sparse matrix-vector multiplications in place of a single much denser one.

7.2.2 Multigrid Preconditioning. Both constraint bubbles and affine fluid regions are reduced models that represent immersed bubbles either with one Lagrange multiplier, or a small layer of exterior cells with an interior affine velocity field. This reduction in the total number of degrees of freedom increases simulation performance. However, our experiments using a parallelized diagonal preconditioned Conjugate Gradients (PCG) solver indicate that the major computational bottleneck is ultimately the much larger liquid domain. Recent work by Aanjaneya [2018] proposed an efficient solver for multi-domain systems using a Schur-complement method that applies a direct solver on the reduced model and a geometric Multigrid V-cycle for the liquid domain. Because our affine bubble model includes exterior cells at the bubble's boundary, the cost of a direct solve per solver iteration can be prohibitively expensive. Instead, we propose a simplified preconditioner that couples a lightweight smoothing routine over the bubble's domain and a Multigrid V-cycle over the liquid domain.

The preconditioner computes an approximate solution to $\mathbf{As} = \mathbf{r}$, where $\mathbf{A}$ is the linear system from either (9) for constraint bubbles, or (23) for affine fluid regions. We partition the degrees of freedom into liquid regions Ω_L, and bubble regions Ω_B, which we denote by $\mathbf{s}_L$ and $\mathbf{s}_B$, respectively. We found that extending the bubble domain to overlap with the liquid domain in a three voxel-wide narrow band gave faster convergence and better performance, so the minor additional cost was a worthwhile trade-off.

We first apply several iterations of a damped-Jacobi smoother over the bubble region, denoted by the approximate inverse operator $\mathbf{A}_B^{\ddagger}$ in Algorithm 1. Then, we apply a geometric Multigrid V-cycle to Ω_L, denoted by $\mathbf{A}_L^{\dagger}$, closely following McAdams et al. [2010]. To account for the intermediate values in the bubble domain, we modified the right-hand side vector as $\mathbf{r}_L - \mathbf{A}_{LB}\mathbf{s}_B^{n-1}$, where $\mathbf{A}_{LB}$ is the subsystem of $\mathbf{A}$ that couples the bubble regions with the liquid domain. After applying a V-cycle, the same smoothing routine $\mathbf{A}_B^{\ddagger}$ is applied to each bubble region, this time accounting for intermediate values from the liquid domain, by modifying the right-hand side vector as $\mathbf{r}_B - \mathbf{A}_{BL}\mathbf{s}_L^n$ (where $\mathbf{A}_{BL}$ is the transpose of $\mathbf{A}_{LB}$). Note that each V-cycle $\mathbf{A}_L^{\dagger}$ is sandwiched between two applications of the smoother $\mathbf{A}_B^{\ddagger}$, preserving the symmetry of our preconditioner.

Intuitively, Algorithm 1 is trying to solve a two-way coupled system by performing outer iterations on a partitioned solver until convergence. This is also similar in spirit to the iterative scheme proposed by Aanjaneya [2018] to approximate the solution to the Schur-complement system. Moreover, like Aanjaneya [2018], we observed that increasing the total number of iterations N inside the preconditioner improved solver convergence. However, in our experiments we found that the additional cost of applying subsequent V-cycles outweighed the reduction in the total number of PCG iterations. For this reason, we used $N = 1$ for all our simulations.

Algorithm 1 Coupled Preconditioner $\mathbf{As} = \mathbf{r}$

$\mathbf{s}_B^0 = \mathbf{A}_B^{\ddagger}\mathbf{r}_B$
for $n = 1 \ldots N$ **do**
  $\mathbf{s}_L^n = \mathbf{A}_L^{\dagger}(\mathbf{r}_L - \mathbf{A}_{LB}\mathbf{s}_B^{n-1})$
  $\mathbf{s}_B^n = \mathbf{A}_B^{\ddagger}(\mathbf{r}_B - \mathbf{A}_{BL}\mathbf{s}_L^n)$
end for

8 RESULTS

To demonstrate that our proposed models can be implemented as direct extensions to a standard single-phase FLIP-based liquid simulator, we developed our models as plugins to Houdini's FLIP solver [SideFX 2020]. All our 3D examples were simulated on a 16-core Ryzen 1950x CPU. The linear systems were solved using a custom Conjugate Gradients method using double precision, with a relative tolerance of 10^{-5} for all examples. All our timings are listed in Table 1.

8.1 Example Scenarios

Water Cooler. Figure 2 illustrates the familiar glugging effect of a *water cooler* scenario. The traditional single-phase free-surface approach simply allows the liquid to pour rapidly through as if both chambers were open to the outside air. By enforcing the incompressible bubble constraints, the downward flow of liquid must match the upward flow of air, generating a sequence of rising bubbles that are constantly being created and pinched off.

Because of the large air regions in the simulation, our constraint method greatly reduces the number of degrees of freedom (DOFs) in the pressure projection when compared to a full two-phase simulation. For the very first frame of this example, an equivalent two-phase simulation would contain 8.7M DOFs. For the same substep, our constraint method reduces the total DOFs in the linear system to 2.5M unreduced DOFs and 2 bubble DOFs . Although the bubble region DOF count fluctuates throughout the simulation due to small bubbles being entrained, it is only in the hundreds.

Despite this example containing 3× more air volume than liquid volume, the solver overhead associated with adding bubbles to the linear system, flood-filling, and bubble ID tracking is insignificant. For example, solving the linear system comprises 87% of the total pressure projection time. Flood-filling to build bubble connected components and tracking bubble IDs both comprise less than 1% of the total pressure projection. The lion's share of overhead was dedicated to building the linear system, which would be significantly slower if a full two-phase linear system was constructed with 3.5× more DOFs.

A possible shortcoming of the constraint method is that the row and column corresponding to a given bubble's constraint can be relatively dense depending on the bubble's surface area. This is because each bubble constraint involves all liquid face velocities incident on that bubble; elimination of the velocity variables leads to coupling between the bubble's Lagrange multiplier and the pressures of all its incident cells. This adds overhead compared to a pure free-surface flow solver (in addition to the cost of identifying bubble regions), but we found it to be relatively minor. The (Multigrid preconditioned) free-surface method took a total of 4h07m to simulate

Table 1. **Timing breakdowns:** Timings for all our 3D examples. "Linear Solve Time" refers strictly to the time to solve the pressure projection linear system itself; "Pressure Projection Overhead" involves all the components of our extended constraint bubble / affine region pressure projection process that are not the linear system solver, i.e., flood-filling to find regions, setting up tiling structures, etc.

Simulation	Resolution	Method	Linear Solve Time	Pressure Projection Overhead	Total Simulation Time
Water Cooler	200x450x200	Free Surface (no bubbles)	1h21m	10m	4h07m
		Constraint Method	1h49m	15m	4h42m
Rising Bubbles	200x400x200	Constraint Method	8h44m	40m	12h03m
Bubble Tube	350x390x180	Constraint Method	1h40m	15m	2h42m
Sinking Bubbles	200x400x200	Two-phase	40h38m	1h58m	45h49m
		Affine Method (bubble only)	13h06m	2h13m	18h35
		Affine Method (both fluids)	9h59m	1h47m	15h02
Splash Tank	500x350x500	Uniform Grid	29h04m	51m	33h04m
		Affine Method (16^3 tiles)	6h29m	1h39m	11h54m
		Affine Method (32^3 tiles)	3h46m	1h32m	8h37m

the water cooler vs. our constraint-bubble method at 4h42m, i.e., bubbles were 14% slower. Although the free-surface and constraint bubble water coolers exhibit wildly differing behaviors which affects their efficiency (notably, the former settles down quickly), this data nevertheless suggests that artists can expect to add air bubble effects to scenes for only a small additional cost.

Rising Bubbles. In Figure 5, we simulate a zero-density constraint bubble immersed in water. The incompressibility constraint applied to the bubble prevents the surrounding liquid from collapsing it. As the surrounding liquid flows downwards under gravity, the massless bubble is forced to rise. Subsequently, turbulent velocities cause it to break apart into many small bubbles. Although we use FLIP particles only for the liquid, and the bubbles undergo frequent splitting and merging, our method is able to track these bubble volumes and correct their volume drift.

This example contains a large liquid body consisting of 13.7M liquid DOFs, compared to only a would-be 2.6M bubble DOFs needed for proper a two-phase simulation. Although reducing 2.6M DOFs to only a few constraints DOFs is a significant reduction, it is still relatively small compared to the remaining size of the standard Poisson component of linear system. This example motivates our coupled Multigrid preconditioner (see §7.2.2) in an effort to optimize the residual reduction for the large liquid body. A single substep comparison shows that our coupled Multigrid preconditioner outperforms a traditional diagonal preconditioner for Conjugate Gradients by 6.3×, converging in 15 iterations over 15.9s, compared to 1m38s in 750 iterations using the diagonal preconditioner. The large density ratio between liquid and air regions, 1000:1, for an equivalent two-phase flow solver lead to poorly conditioned linear systems. We observed that over the first five frames, a standard two-phase method required an average of 1752 iterations using the diagonal preconditioner and was 17.6× slower than our Multigrid-preconditioned constraint method.

Bubble Tube. Figure 4 demonstrates how kinematically scripted moving solid boundaries interact with our constraint-based air bubbles. When the red piston pushes down through the glass tube, it creates a net flux at the solid-air boundary (and corresponding right-hand-side terms in the pressure solve) that must be compensated by an opposing flux at the air-liquid boundary. This flux on the liquid surface pushes the liquid volume through the tube, despite it never having come into direct contact with the solid. Similarly, the fully disconnected liquid regions interact through the second air pocket and the resulting chain of interactions, driven by the moving piston, forces liquid out of the spout and into the beaker below. As the piston pulls back to its initial position, it creates a negative flux that draws the liquid back into the tube along with it.

The ambient air region surrounding the glass tube is 21× larger than the volume of liquid in the simulation implying a huge potential speed-up; including this body of air in a full-two phase simulation would clearly be prohibitively expensive. Fortunately in this case, one could choose to treat the ambient region as a Dirichlet condition and remove it from the two-phase pressure system, since it does not affect the tube's inner dynamics (though the interior air gaps would still require full two-phase air DOFs). However, if two liquid bodies were to be hydraulically connected across a potentially huge ambient air region, *only* our approach will suffice. To identify connected air components on which to apply our constraint bubbles, a flood-fill operation across all such *potentially* active cells is required, which could hypothetically be a bottleneck. We found, though, that despite the ambient region being very large in this problem our reasonably optimized/parallel flood-fill comprised only 2% of the total pressure projection time.

We observed that our geometric Multigrid preconditioner was not very effective for this example. This is a well-known limitation of Multigrid schemes like that of McAdams et al. [2010] for simulations with *maze*-like solid boundary conditions. Employing a topology-aware coarsening scheme could alleviate this problem [Dick et al. 2016].

Sinking Bubbles. As demonstrated in Figure 8, our proposed affine velocity method is able to maintain hydrostatic equilibrium for bubbles with density coefficients matching the surrounding liquid, as well as capturing the expected sinking motion of bubbles with even higher density. This same sinking behavior is further illustrated in Figure 11, in which a dense bubble falls through the surrounding liquid into a pool of dense bubble material initially at rest at the bottom of the tank.

Similar to the *rising bubbles* example, the liquid volume is significantly larger than the bubble volume and simply reducing the DOF count in the bubble regions would offer only a small relative performance increase. We employ a Multigrid preconditioner similar to the constraint method, but with a small layer of exterior bubble cells included in the bubble smoothing routine. In a single time step comparison, a diagonal preconditioned Conjugate Gradients solve converges in 1274 iterations and takes 3m01s, compared to 50 iterations in 1m08s for our coupled Multigrid preconditioner, giving a performance improvement of 2.7×. We believe the lower performance improvement compared to the constraint-based examples is in part due to the use of a pointwise smoother in the exterior bubble region. Although a Multigrid V-cycle is incredibly effective at reducing the residual in the linear system for the liquid domain, a pointwise smoother over the exterior domain is not. We believe a *box* smoother, that would perform local direct solves over a group of DOFs, might offer improved performance [Aanjaneya et al. 2019].

We also experimented with applying the tiled affine method to *both* materials. Similar to the bubble-only affine method, we keep a layer of exterior cells at the two-phase and solid boundaries. We found that using the same layer thickness as the bubble-only method resulted in subtle grid artifacts as the bubble sank. Doubling the thickness from three voxels to six voxels on each side of the two-phase boundary was enough to remove these artifacts. This approach offered a moderate performance improvement of 1.2× over the bubble-only affine coupled Multigrid method.

Both affine methods significantly outperform the two-phase equivelant by 3.1× for the affine method on the bubble only, and 4.1× for the affine method on both fluids. However, the performance increase is not as dramatic as the *rising bubbles* example because the density ratio, 10:1, of the two fluids is not as extreme.

Splash Tank. Figure 14 demonstrates that our affine model generalizes from the two-phase flows to the more common single-phase free-surface case, and can be used to create highly detailed results. In this example, both the spheres and liquid pool maintain a 9-voxel thick band at the liquid-air boundary, a 2-voxel thick band at the liquid-solid boundary, and their interiors are filled with tiled affine regions that are (at most) 16^3 voxels in size with a single layer of voxels between each tile. The affine regions lead to a reduction from 21.5M DOFs in the regular grid setting to 6.4M unreduced cells and 5.2K affine interior regions. The linear solve time of our affine free-surface model outperforms the regular grid method by 3.7× using a diagonal preconditioned Conjugate Gradients solver for both. We further compared methods at 2× resolution, doubling the affine region size and boundary thickness, and observed a 6.1× performance improvement for a single timestep. This suggests that our proposed method offers much better scaling than the regular grid for high resolutions. We emphasize that existing methods for improving the performance of a regular grid method, such as Multigrid or octree adaptivity, often involve complex data structures and implementations. By comparison, our proposed affine method requires minimal additional implementation.

We also investigated the effect of reducing the affine layer at the free surface from a 9-voxel thick band to only 3 voxels thick. Despite a moderate performance improvement of 1.2×, the thin band resulted in grid artifacts (see Figure 16). However, maintaining a 9-voxel thick band and instead doubling the tile size to 32^3 offers a 5.6× performance improvement over the uniform grid and 1.5× over the 16^3 tile-sized affine method with only slightly damped motion and no surface artifacts.

9 DISCUSSION AND CONCLUSIONS

We have proposed two new reduced fluid models that integrate with and extend the widely adopted staggered grid paradigm, and we have fruitfully applied these models in a range of compelling two-phase and free-surface flow scenarios. Below we discuss some of their limitations and potential directions for exciting future investigation.

Our constraint bubbles assume incompressibility for simplicity. However, if air compressibility is desired, our bubble-tracking could straightforwardly be extended with a per-bubble mass variable to inform a density-based equation of state model. Such terms are used by Aanjaneya et al. [2013], but our approach would again circumvent explicit interior projection or (mass-conserving) advection. Relatedly, it would be interesting to introduce divergence-control for affine regions, which could be useful for explosion effects [Feldman et al. 2003] and volume correction, as well as the extension to equation of state models for compressible bubbles. This could be incorporated through a modified constraint of $\nabla \cdot \boldsymbol{u}_B = \mathrm{Tr}(\mathbf{A}) = const$, leading to additional terms in the right hand side vector.

We only exploited affine regions during pressure projection (because it is often the dominating cost in practice) and used standard grid-based advection in their interiors. This might become a bottleneck for sufficiently large regions. However, it may be possible to radically reduce this cost, perhaps through a *reduced affine advection* model, in the vein of prior model reduced advection treatments [Cui et al. 2018; Wicke et al. 2009], or by carefully recovering (approximate) affine velocities from surrounding grid velocities, in the spirit of our constraint bubbles. Similarly, we did not consider viscosity, but anticipate that our approach will generalize to this setting in roughly the same manner as rigid-viscous coupling [Takahashi and Lin 2019]. Surface tension effects could be added with a standard ghost-fluid approach [Enright et al. 2003; Hong and Kim 2005].

Affine regions, while powerful, cannot always be applied in the same situations as constraint bubbles. For example, a single constraint bubble suffices for each air gap in the winding tube geometry, but multiple affine regions would be needed for comparable non-zero density flows because their explicit interior velocity field cannot represent several twists. Unsurprisingly, we also observed that a too-thin layer of regular cells between the free surface and the affine tiles can cause grid-dependent motion artifacts (see inset figure), similar to prior adaptive

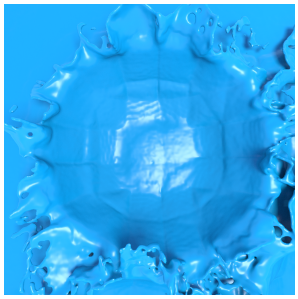


Fig. 16. Grid artifacts appear when the surface layer is too thin.

schemes (e.g., Irving et al. [2006] discuss the notion of problem-dependent "optical depth").

Uniform affine tiling is perhaps the simplest adaptivity strategy that one might consider; because of the geometric flexibility inherent in our affine regions, alternative grading/shaping/sizing patterns could yield even further speed-ups. For example, quickly ramping tile sizes away from the surface, as in octrees, is an obvious next step. We assumed voxelized affine regions for simplicity, but as with rigid bodies, irregular regions could be supported via cut-cells [Ng et al. 2009]. We also assumed that affine tiles were separated by a layer of regular cells, for simplicity and to avoid velocity discontinuities between tiles — it would be interesting to explore direct tile-tile coupling.

Lastly, we would like to explore whether further upgrading the interior model of our affine regions would pay dividends in practice. Adopting higher order polynomial vector fields, as in PolyPIC [Fu et al. 2017], would improve the velocity field's flexibility at the cost of (rapidly) increasing degree-of-freedom counts per region. Coupling instead to a boundary element-type (e.g., *harmonic*) velocity field [Da et al. 2016] is also an exciting avenue, though such models have many velocity degrees of freedom that cover the entire surface, rather than being coefficients of a low order polynomial.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (RGPIN-04360-2014), the Rutgers University start-up grant, and the Ralph E. Powe Junior Faculty Enhancement Award. We would like to thank Cristin Barghiel and SideFX for their generous software donation and Ryoichi Ando for his insightful discussion on comparing our constraint method with stream functions. We would also like to thank the anonymous reviewers for their valuable feedback.

REFERENCES

Mridul Aanjaneya. 2018. An Efficient Solver for Two-way Coupling Rigid Bodies with Incompressible Flow. *Computer Graphics Forum* 37, 8 (2018), 59–68. https://doi.org/10.1111/cgf.13512

Mridul Aanjaneya, Ming Gao, Haixiang Liu, Christopher Batty, and Eftychios Sifakis. 2017. Power diagrams and sparse paged grids for high resolution adaptive liquids. *ACM Transactions on Graphics* 36, 4, Article 140 (2017). https://doi.org/10.1145/3072959.3073625

Mridul Aanjaneya, Chengguizi Han, Ryan Goldade, and Christopher Batty. 2019. An Efficient Geometric Multigrid Solver for Viscous Liquids. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 2, 2, Article 14 (2019). https://doi.org/10.1145/3340255

Mridul Aanjaneya, Saket Patkar, and Ronald Fedkiw. 2013. A monolithic mass tracking formulation for bubbles in incompressible flow. *J. Comp. Phys.* 247 (2013), 17–61. https://doi.org/10.1016/j.jcp.2013.03.048

Bart Adams, Mark Pauly, Richard Keiser, and Leonidas J. Guibas. 2007. Adaptively Sampled Particle Fluids. *ACM Transactions on Graphics* 26, 3, Article 48 (2007). https://doi.org/10.1145/1276377.1276437

Ryoichi Ando, Nils Thuerey, and Chris Wojtan. 2015. A Stream Function Solver for Liquid Simulations. *ACM Transactions on Graphics* 34, 4, Article 53 (2015). https://doi.org/10.1145/2766935

Christopher Batty, Florence Bertails, and Robert Bridson. 2007. A Fast Variational Framework for Accurate Solid-fluid Coupling. *ACM Transactions on Graphics* 26, 3, Article 100 (2007). https://doi.org/10.1145/1276377.1276502

Christopher Batty, Stefan Xenos, and Ben Houston. 2010. Tetrahedral embedded boundary methods for accurate and flexible adaptive fluids. *Computer Graphics Forum* 29, 2 (2010), 695–704. https://doi.org/10.1111/j.1467-8659.2009.01639.x

Landon Boyd and Robert Bridson. 2012. MultiFLIP for energetic two-phase fluid simulation. *ACM Transactions on Graphics* 31, 2, Article 16 (2012). https://doi.org/10.1145/2159516.2159522

Robert Bridson. 2015. *Fluid simulation for computer graphics.* CRC Press.

Tyson Brochu, Christopher Batty, and Robert Bridson. 2010. Matching fluid simulation elements to surface geometry and topology. *ACM Transactions on Graphics* 29, 4, Article 47 (2010). https://doi.org/10.1145/1778765.1778784

Oleksiy Busaryev, Tamal K. Dey, Huamin Wang, and Zhong Ren. 2012. Animating bubble interactions in a liquid foam. *ACM Transactions on Graphics* 31, 4, Article 63 (2012). https://doi.org/10.1145/2185520.2185559

Mark Carlson, Peter J. Mucha, Greg Turk, and Greg Turk. 2004. Rigid Fluid: Animating the Interplay Between Rigid Bodies and Fluid. *ACM Transactions on Graphics* 23, 3 (2004), 377–384. https://doi.org/10.1145/1015706.1015733

Nuttapong Chentanez and Matthias Müller. 2011. Real-time Eulerian Water Simulation Using a Restricted Tall Cell Grid. *ACM Transactions on Graphics*, Article 82 (2011). https://doi.org/10.1145/1964921.1964977

Qiaodong Cui, Pradeep Sen, and Theodore Kim. 2018. Scalable Laplacian Eigenfluids. *ACM Transactions on Graphics* 37, 4, Article 87 (2018). https://doi.org/10.1145/3197517.3201352

Fang Da, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2015. Double bubbles sans toil and trouble: Discrete circulation-preserving vortex sheets for soap films and foams. *ACM Transactions on Graphics* 34, 4, Article 149 (2015). https://doi.org/10.1145/2767003

Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-only liquids. *ACM Transactions on Graphics* 35, 4, Article 78 (2016). https://doi.org/10.1145/2897824.2925899

Fernando de Goes, Corentin Wallez, Jin Huang, Dmitry Pavlov, and Mathieu Desbrun. 2015. Power Particles: An Incompressible Fluid Solver Based on Power Diagrams. *ACM Transactions on Graphics* 34, 4, Article 50 (2015). https://doi.org/10.1145/2766901

Tyler De Witt, Christian Lessig, and Eugene Fiume. 2012. Fluid simulation using Laplacian eigenfunctions. *ACM Transactions on Graphics* 31, 1, Article 10 (2012). https://doi.org/10.1145/2077341.2077351

C. Dick, M. Rogowsky, and R. Westermann. 2016. Solving the Fluid Pressure Poisson Equation Using Multigrid—Evaluation and Improvements. *IEEE Transactions on Visualization and Computer Graphics* 22, 11 (2016), 2480–2492. https://doi.org/10.1109/TVCG.2015.2511734

Essex Edwards and Robert Bridson. 2014. Detailed water with coarse grids: combining surface meshes and adaptive discontinuous Galerkin. *ACM Transactions on Graphics* 33, 4, Article 136 (2014).

R. Elliot English, Linhai Qiu, Yue Yu, and Ronald Fedkiw. 2013. Chimera Grids for Water Simulation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '13)*. Association for Computing Machinery, 85–94. https://doi.org/10.1145/2485895.2485897

Douglas Enright, Stephen Marschner, and Ronald Fedkiw. 2002. Animation and Rendering of Complex Water Surfaces. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '02)*. Association for Computing Machinery, 736–744. https://doi.org/10.1145/566570.566645

Doug Enright, Duc Nguyen, Frederic Gibou, and Ron Fedkiw. 2003. Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In *ASME/JSME 2003 4th Joint Fluids Summer Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 337–342. https://doi.org/10.1115/FEDSM2003-45144

Bryan E. Feldman, James F. O'Brien, and Okan Arikan. 2003. Animating Suspended Particle Explosions. *ACM Transactions on Graphics* 22, 3 (2003), 708–715. https://doi.org/10.1145/882262.882336

Chuyuan Fu, Qi Guo, Theodore Gast, Chenfanfu Jiang, and Joseph Teran. 2017. A polynomial particle-in-cell method. *ACM Transactions on Graphics* 36, 6, Article 222 (2017). https://doi.org/10.1145/3130800.3130878

Abhinav Golas, Rahul Narain, Jason Sewall, Pavel Krajcevski, Pradeep Dubey, and Ming Lin. 2012. Large-scale fluid simulation using velocity-vorticity domain decomposition. *ACM Transactions on Graphics* 31, 6, Article 148 (2012). https://doi.org/10.1145/2366145.2366167

S. T. Greenwood and D. H. House. 2004. Better with Bubbles: Enhancing the Visual Realism of Simulated Fluid. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '04)*. Eurographics Association, 287–296. https://doi.org/10.1145/1028523.1028562

Karl Gustafson and Robert Hartman. 1983. Divergence-free bases for finite element schemes in hydrodynamics. *SIAM J. Numer. Anal.* 20, 4 (1983), 697–721.

Francis H. Harlow and J. Eddie Welch. 1965. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids* 8, 12 (1965), 2182–2189. https://doi.org/10.1063/1.1761178

Jeong-Mo Hong and Chang-Hun Kim. 2005. Discontinuous fluids. *ACM Transactions on Graphics* 24, 3 (2005), 915–920. https://doi.org/10.1145/1073204.1073283

Jeong-Mo Hong, Ho-Young Lee, Jong-Chul Yoon, and Chang-Hun Kim. 2008. Bubbles alive. *ACM Transactions on Graphics* 27, 3, Article 48 (2008). https://doi.org/10.1145/1360612.1360647

Hikaru Ibayashi, Chris Wojtan, Nils Thuerey, Takeo Igarashi, and Ryoichi Ando. 2018. Simulating Liquids on Dynamically Warping Grids. *IEEE Transactions on Visualization and Computer Graphics* in press (2018). https://doi.org/10.1109/TVCG.2018.

2883628
Geoffrey Irving, Eran Guendelman, Frank Losasso, and Ronald Fedkiw. 2006. Efficient Simulation of Large Bodies of Water by Coupling Two and Three Dimensional Techniques. *ACM Transactions on Graphics* (2006), 805–811. https://doi.org/10.1145/1179352.1141959
Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The affine particle-in-cell method. *ACM Transactions on Graphics* 34, 4, Article 51 (2015). https://doi.org/10.1145/2766996
Myungjoo Kang, Ron Fedkiw, and Xu-Dong Liu. 2000. A boundary condition capturing method for multiphase incompressible flow. *SIAM J. Sci. Comput.* 15, 3 (2000), 323–360. https://doi.org/10.1023/A:1011178417620
Todd Keeler and Robert Bridson. 2014. Ocean waves animation using boundary integral equations and explicit mesh tracking. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '14)*. Eurographics Association, 11–19. https://doi.org/10.5555/2849517.2849520
Byungmoon Kim, Yingjie Liu, Ignacio Llamas, Xiangmin Jiao, and Jarek Rossignac. 2007. Simulation of bubbles in foam with the volume control method. *ACM Transactions on Graphics* 26, 3, Article 98 (2007). https://doi.org/10.1145/1275808.1276500
Bryan M. Klingner, Bryan E. Feldman, Nuttapong Chentanez, and James F. O'Brien. 2006. Fluid animation with dynamic meshes. *ACM Transactions on Graphics* 25, 3 (2006), 820–825. https://doi.org/10.1145/1179352.1141961
Tassilo Kugelstadt, Andreas Longva, Nils Thurey, and Jan Bender. 2019. Implicit Density Projection for Volume Conserving Liquids. *IEEE Transactions on Visualization and Computer Graphics* (2019). https://doi.org/10.1109/TVCG.2019.2947437
Timothy R. Langlois, Changxi Zheng, and Doug L. James. 2016. Toward Animating Water with Complex Acoustic Bubbles. *ACM Transactions on Graphics* 35, 4, Article 95 (2016). https://doi.org/10.1145/2897824.2925904
Frank Losasso, Frédéric Gibou, and Ron Fedkiw. 2004. Simulating Water and Smoke with an Octree Data Structure. *ACM Transactions on Graphics* 23, 3 (2004), 457–462. https://doi.org/10.1145/1015706.1015745
Frank Losasso, Tamar Shinar, Andrew Selle, and Ronald Fedkiw. 2006. Multiple interacting liquids. *ACM Transactions on Graphics* 25, 3 (2006), 812–819. https://doi.org/10.1145/1179352.1141960
Wenlong Lu, Ning Jin, and Ronald Fedkiw. 2016. Two-way coupling of fluids to reduced deformable bodies. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '16)*. Eurographics Association, 67–76. https://doi.org/10.5555/2982818.2982829
Scott P MacLachlan, Jok Man Tang, and Cornelis Vuik. 2008. Fast and robust solvers for pressure-correction in bubbly flow problems. *J. Comput. Phys.* 227, 23 (2008), 9742–9761. https://doi.org/10.1016/j.jcp.2008.07.022
Gianmarco Manzini, Alessandro Russo, and N Sukumar. 2014. New perspectives on polygonal and polyhedral finite element methods. *Mathematical Models and Methods in Applied Sciences* 24, 08 (2014), 1665–1699. https://doi.org/10.1142/S0218202514400065
Sebastian Martin, Peter Kaufmann, Mario Botsch, Martin Wicke, and Markus Gross. 2008. Polyhedral finite elements using harmonic basis functions. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1521–1529. https://doi.org/10.1111/j.1467-8659.2008.01293.x
A. McAdams, E. Sifakis, and J. Teran. 2010. A Parallel Multigrid Poisson Solver for Fluids Simulation on Large Grids. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '10)*. Eurographics Association, 65–74. https://doi.org/10.5555/1921427.1921438
Viorel Mihalef, B. Unlusu, Dimitris Metaxas, Mark Sussman, and M. Y. Hussaini. 2006. Physics based boiling simulation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '06)*. 317–324. https://doi.org/10.1145/3005358.3005385
Matthias Müller, David Charypar, and Markus Gross. 2003. Particle-based fluid simulation for interactive applications. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '03)*. 154–159. https://doi.org/10.5555/846276.846298
Yen Ting Ng, Chohong Min, and Frédéric Gibou. 2009. An efficient fluid–solid coupling algorithm for single-phase flows. *J. Comput. Phys.* 228, 23 (2009), 8807–8829. https://doi.org/10.1016/j.jcp.2009.08.032
Saket Patkar, Mridul Aanjaneya, Dmitriy Karpman, and Ronald Fedkiw. 2013. A Hybrid Lagrangian-Eulerian Formulation for Bubble Generation and Dynamics. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '13)*. ACM, 105–114. https://doi.org/10.1145/2485895.2485912
Sander Rhebergen and Garth N Wells. 2018. A hybridizable discontinuous Galerkin method for the Navier–Stokes equations with pointwise divergence-free velocity field. *Journal of Scientific Computing* 76, 3 (2018), 1484–1501. https://doi.org/10.1007/s10915-018-0671-4
Avi Robinson-Mosher, R. Elliot English, and Ronald Fedkiw. 2009. Accurate tangential velocities for solid fluid coupling. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '09)*. 227–236. https://doi.org/10.1145/1599470.1599500
Avi Robinson-Mosher, Tamar Shinar, Jon Gretarsson, Jonathan Su, and Ronald Fedkiw. 2008. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Transactions on Graphics* 27, 3, Article 46 (2008). https://doi.org/10.1145/1360612.1360645
SideFX. 2020. Houdini.
Barbara Solenthaler and Markus Gross. 2011. Two-scale particle simulation. *ACM Transactions on Graphics* 30, 4, Article 81 (2011). https://doi.org/10.1145/2010324.1964976
Barbara Solenthaler and Renato Pajarola. 2008. Density contrast SPH interfaces. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '08)*. 211–218. https://doi.org/10.5555/1632592.1632623
Oh-young Song, Hyuncheol Shin, and Hyeong-Seok Ko. 2005. Stable but non-dissipative water. *ACM Transactions on Graphics* 24, 1 (2005), 81–97. https://doi.org/10.1145/1037957.1037962
Tetsuya Takahashi and Ming C Lin. 2019. A Geometrically Consistent Viscous Fluid Solver with Two-Way Fluid-Solid Coupling. *Computer Graphics Forum* 38, 2 (2019), 49–58. https://doi.org/10.1111/cgf.13618
Nils Thürey, Chris Wojtan, Markus Gross, and Greg Turk. 2010. A multiscale approach to mesh-based surface tension flows. *ACM Transactions on Graphics* 29, 4, Article 48 (2010). https://doi.org/10.1145/1833349.1778785
Adrien Treuille, Andrew Lewis, and Zoran Popović. 2006. Model reduction for real-time fluids. *ACM Transactions on Graphics* 25, 3 (2006), 826–834. https://doi.org/10.1145/1179352.1141962
Martin Wicke, Matt Stanton, and Adrien Treuille. 2009. Modular bases for fluid dynamics. *ACM Transactions on Graphics* 28, 3, Article 39 (2009). https://doi.org/10.1145/1531326.1531345
Changxi Zheng and Doug L. James. 2009. Harmonic fluids. *ACM Transactions on Graphics* 28, 3, Article 37 (2009). https://doi.org/10.1145/1531326.1531343
Wen Zheng, Jun-Hai Yong, and Jean-Claude Paul. 2006. Simulation of bubbles. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '06)*. Eurographics Association, 325–333. https://doi.org/10.5555/1218064.1218109
Bo Zhu, Wenlong Lu, Matthew Cong, Byungmoon Kim, and Ronald Fedkiw. 2013. A New Grid Structure for Domain Extension. *ACM Transactions on Graphics* 32, 4, Article 63 (2013). https://doi.org/10.1145/2461912.2461999
Yongning Zhu and Robert Bridson. 2005. Animating Sand as a Fluid. *ACM Trans. Graph.* 24, 3 (July 2005), 965–972. https://doi.org/10.1145/1073204.1073298

A VARIATIONAL INTERPRETATION OF AFFINE COUPLING

It is straightforward to see that our model will be symmetric by considering a variational interpretation:

$$\underset{\boldsymbol{u}_F, \boldsymbol{u}_{\text{const}}, \mathbf{A}}{\arg\min} \quad \iiint_{\Omega_F} \frac{\rho}{2} \|\boldsymbol{u}_F - \boldsymbol{u}_F^*\|^2 + \iiint_{\Omega_B} \frac{\rho}{2} \|\boldsymbol{u}_B - \boldsymbol{u}_B^*\|^2 \tag{27}$$

$$\text{subject to} \quad \nabla \cdot \mathbf{u}_F = 0 \text{ on } \Omega_F \tag{28}$$

$$\boldsymbol{u}_B = \boldsymbol{u}_{\text{const}} + \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}_{\text{COM}}) \text{ on } \Omega_B \tag{29}$$

$$\nabla \cdot \boldsymbol{u}_B = \text{Tr}(\boldsymbol{A}) = 0 \text{ on } \Omega_B \tag{30}$$

$$\boldsymbol{u}_F = \boldsymbol{u}_B \text{ on } \partial\Omega_B \cap \partial\Omega_F \tag{31}$$

In essence, we seek the nearest incompressible field over both fluid regions as measured under the kinetic energy norm, subject to incompressibility, an affine velocity on the reduced region Ω_B, and matching velocities at their shared interface. This is a quadratic problem with linear equality constraints.