

Codimensional Surface Tension Flow on Simplicial Complexes

Bo Zhu* Bo Zhu*
Stanford University Stanford University
Ed Quigley* Ed Quigley*
Stanford University Stanford University
Matthew Cong* Matthew Cong*
Stanford University Industrial Light + Magic
Justin Solomon* Justin Solomon*
Stanford University
Ronald Fedkiw* Ronald Fedkiw*
Stanford University Industrial Light + Magic

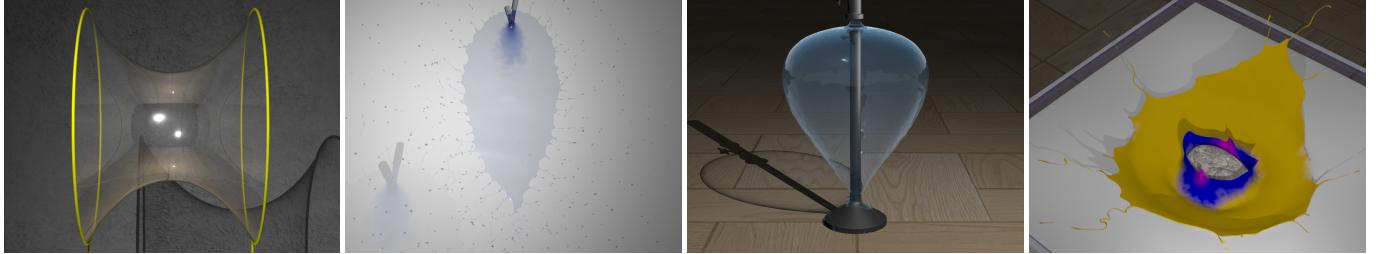


Figure 1: Many fluid phenomena consist of thin films and filaments. (Far Left) Film Catenoid: a membrane suspended between two rings contracts due to surface tension. (Middle Left) Fishbone: two colliding jets form a thin sheet, filaments, and droplets. This phenomena is named for its resemblance to a fish skeleton. (Middle Right) Waterbell: a jet of water striking an impactor results in a closed circular water sheet that resembles a bell. (Far Right) Paint Splash: a splash caused by a rock falling into a tank filled with layers of colored paint.

Abstract

Many visually interesting natural phenomena are characterized by thin liquid sheets, long filaments, and droplets. We present a new Lagrangian-based numerical method to simulate these codimensional surface tension driven phenomena using non-manifold simplicial complexes. Tetrahedra, triangles, segments, and points are used to model the fluid volume, thin films, filaments, and droplets, respectively. We present a new method for enforcing fluid incompressibility on simplicial complexes along with a physically-guided meshing algorithm to provide temporally consistent information for interparticle forces. Our method naturally allows for transitions between codimensions, either from tetrahedra to triangles to segments to points or vice versa, regardless of the simulation resolution. We demonstrate the efficacy of this method by simulating various natural phenomena that are characterized by thin fluid sheets, filaments, and surface tension effects.

CR Categories: I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: incompressible, surface tension, codimension

Links:  DL  PDF

1 Introduction

Interfacial fluid phenomena such as the “fishbone” [Hasha and Bush 2002], “fluid polygon” [Buckingham and Bush 2001], and “water-

bell” [Clanet 2007] exhibit codimensional features including thin films, filaments, and droplets (see Figure 1). It is extremely difficult to simulate these very thin features that are codimension-1 (two-dimensional films), codimension-2 (fluid filaments), and codimension-3 (very small drops) using numerical methods that are intended to represent only codimension-0 volumetric phenomena. Adaptivity can improve these methods, but it is highly inefficient to use vanishingly small tetrahedra when triangles, line segments, and points are more natural representations. Triangles readily model thin films such as in soap bubbles (see e.g. [Saye and Sethian 2013; Kim et al. 2007; Patkar et al. 2013; Durikovic 2001]) or viscous sheets [Batty et al. 2012]. Line segments are a natural way of simulating filaments and viscous threads [Bergou et al. 2010]. Points are a convenient way to express water droplets [Foster and Fedkiw 2001; Müller et al. 2003]. Etc. Motivated by this, we set out to simulate the wide variety of liquid phenomena best represented by simplicial complexes spanning the range from codimension-0 (tetrahedra) to codimension-1 (triangles) to codimension-2 (segments) to codimension-3 (points). Moreover, for many of these phenomena it is important that the liquid retain the ability to change codimension as needed. Our goals are similar to those of [Martin et al. 2010] and [Stam 2009] which aims towards a unified simulation of various codimensional solids, but liquids have the added complexity that changes from one codimension to another are much more rampant.

Our fluid solver utilizes an underlying particle based approach augmented with physically-guided meshing that evolves the simplicial complex dynamically and provides temporally consistent connectivity information for the particles. This connectivity information is quite important for various aspects of our numerical approach: it is used in the codimensional surface tension discretization to model surface tension effects over the simplicial complex. The topology of the simplicial complex is used to detect which particles are on the boundary of a codimension-1 sheet, and to subsequently apply a special rim-based surface tension force to those particles. The mesh is used to detect closed air volumes which apply equation of state-based air pressure forces to the fluid surface. Most importantly, the connectivity determines the nature of the forces emanating from our novel method for enforcing fluid incompressibility over simplicial complexes.

The main contribution of our work is a new computational framework to simulate codimensional surface tension phenomena using

*e-mail: {boolzhu,eqigley,mdcong,justin.solomon,rfedkiw}@stanford.edu

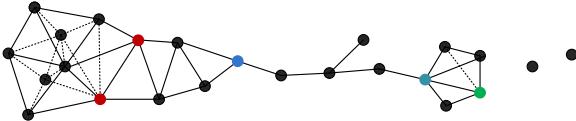


Figure 2: A three-dimensional simplicial complex composed of tetrahedra, triangles, segments, and points. Colored particles are connected to geometry of multiple codimensions.

non-manifold simplicial complexes. The framework uses a Lagrangian representation of fluid features organized into points, segments, triangles, and tetrahedra. It incorporates a novel Poisson solver to enforce incompressibility on simplicial complexes. A physically-guided meshing algorithm is used to naturally handle codimensional transitions and maintain temporally coherent connectivity between particles. Finally, the framework presents an algorithm to construct a skinned mesh of the fluid surface from the simplicial complexes.

2 Related Work

Researchers have developed a variety of data structures and numerical methods with the aim of simulating fluid phenomena with thin features including adaptive grids [Losasso et al. 2004; Chen et al. 2012], unstructured meshes [Misztal et al. 2012; Clausen et al. 2013], front tracking [Thürey et al. 2010], Voronoi diagrams [Brochu et al. 2010; Sin et al. 2009], vortex sheets [Kim et al. 2009; Pfaff et al. 2012], and particle based methods [Ando et al. 2012; Akinci et al. 2013]. Among these, particle based methods are noteworthy for their inherent advantages in conserving mass, tracking fluid volumes, and applying surface tension forces. A disadvantage of these methods is that they do not typically maintain temporally consistent connectivity information and instead compute discretization stencils on-the-fly in order to solve the fluid dynamics equations. See e.g. [Müller et al. 2003; Adams et al. 2007; Premžoe et al. 2003; Becker and Teschner 2007]. There have been some attempts to create a mesh representation for particle based methods in order to solve issues with bumps and noise when rendering [Yu and Turk 2010; Ando et al. 2013]. However, these methods suffer from problems with temporal coherency because topology is recomputed from scratch each time step. Moreover, creating a mesh as a postprocess does not allow for the connectivity information to be used by the simulation to dictate its physical behavior. For example, surface tension forces should not be applied to the interior of a tetrahedral mesh, special rim-based surface tension forces should be applied to particles on the boundary of the triangle mesh, etc. There are previous works such as [Wicke et al. 2010; Clausen et al. 2013; Idelsohn et al. 2004; Misztal et al. 2010] that maintain explicit connectivity information during simulation, but they are typically focused on tetrahedral meshes and not simplicial complexes.

Front-tracking is another related Lagrangian method that explicitly maintains a detailed surface mesh representing the fluid surface. This explicit representation is then advected after enforcing incompressibility on a coarser background discretization such as a Cartesian grid [Thürey et al. 2010], tetrahedral mesh [Batty et al. 2010], or SPH particles [Yu et al. 2012]. Unfortunately features which cannot be resolved by the background discretization are typically deleted or absorbed into the nearby regions resulting in mass and/or feature loss. To avoid this, we store the mass on the particles making it straightforward to conserve mass. In this way, our method has more in common with particle based methods (and finite element methods) than it does with front-tracking methods. However, the use of persistent and physically-guided connectivity information enables our approach to address many more interesting fluid

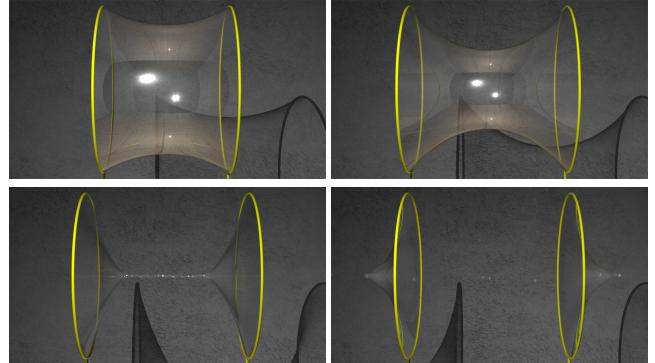


Figure 3: A thin membrane is suspended between two rings. As the rings move farther apart, surface tension causes the membrane to contract. Finally the membrane splits, leaving two separate oscillating surfaces and multiple droplets. Small droplets are modeled using points while larger drops are represented by tetrahedra.

phenomena than a typical particle based method.

3 Simulation Variables

As shown in Figure 2, the fluid is represented by particles connected into a three-dimensional non-manifold simplicial complex containing tetrahedra, triangles, segments, and points. We use tetrahedra to represent fluid volumes, triangles to represent thin films, segments to represent filaments, and points to represent droplets. To differentiate between particles and points, throughout the paper we will use the geometric term “point” to refer to an isolated particle without any incident tetrahedra, triangles, or segments. Each particle has position \vec{x} , velocity \vec{u} , mass m , thickness λ , and mesh connectivities C as attributes. Position and velocity are used to evolve the particles. Mass, thickness, and mesh connectivity are used to determine the local geometry of the fluid. We assume that the fluid has constant density ρ .

Particle connectivity is an important variable in our simulation, because changing the connectivity between particles changes the fluid geometry that is represented by those particles. For example, four particles connected as a tetrahedron represents an incompressible fluid volume whereas the same four particles treated as individual points represents four separate droplets. We define codimension-0 particles to be all particles incident to a tetrahedron, codimension-1 particles to be all particles incident to a triangle, codimension-2 particles to be all particles incident to a segment, and codimension-3 particles to be all points. Thus, a particle incident to a tetrahedron and a triangle would be both a codimension-0 particle and a codimension-1 particle at the same time. Furthermore, we define codimension-0 boundary particles to be all particles on the boundary of the tetrahedron mesh, codimension-1 boundary particles to be all particles on a boundary segment of the triangle mesh that is not a segment on the tetrahedron mesh, and codimension-2 boundary particles to be all particles on the boundary of the segment mesh but not on the tetrahedron or triangle mesh. For $k \in \{0, 1, 2\}$, codimension- k particles that are not codimension- k boundary particles will be denoted as codimension- k interior particles. This implies that a codimension- k interior particle cannot be connected to a simplex of a higher codimension. According to a particle’s connectivity, we distribute its mass m into a codimensional mass $\vec{\mu} = (\mu_0, \mu_1, \mu_2)$ where μ_k is the particle mass attributed to the incident tetrahedra ($k = 0$), triangles ($k = 1$), and segments ($k = 2$) respectively.



Figure 4: Fluid polygons with different numbers of corners. The constant c (which multiplies the in-plane curvature) is replaced with a sawtooth function parameterized by the angular component of the particle's position in cylindrical coordinates relative to the source. The number of sides in the fluid polygon is directly related to the user-specified frequency of the sawtooth function. This produces the fluid polygon behavior depicted in [Buckingham and Bush 2001] and provides a practical way for artists to control this phenomenon.

4 Thin Film and Filament Thickness

The thickness λ of a particle is calculated according to the particle's codimensions, as summarized in Table 1. The expressions for thickness are best understood in terms of the local geometry at a particle. The local geometry for a codimension-3 particle is a sphere. The local geometry for a codimension-2 interior particle is given by multiple cylinders incident to the particle, where the total length l_2 of these cylinders is equal to one half the sum of the corresponding incident segment lengths. A codimension-1 interior particle's geometry is a thin sheet with area A equal to $1/3$ times the sum of the areas of its incident triangles. Codimension-0 particles simply maintain their volumetric geometry and have zero thickness. The geometry for a codimension-2 boundary particle represents a filament endpoint and is defined as a sphere. The geometry for a codimension-1 boundary particle is given by multiple cylinders incident to the particle where each cylinder has an axis along an incident boundary segment of the triangle mesh. The total length l_1 of these incident cylinders is equal to half the sum of the corresponding incident segment lengths.

The local geometry of particles of multiple codimensions can be defined by the combination of geometries from each codimension. Because tetrahedral elements contribute no thickness to their incident particles, the thickness of a particle of codimension-0 and any other codimensions is equal to the thickness contributed by its other codimensions. The geometry of a codimension-1 and 2 particle is given by summing both the cylinders contributed by incident segments and the boundary cylinders contributed by the incident triangle. For this case, $\lambda = 2((\mu_1 + \mu_2)/(\rho\pi(l_1 + l_2)))^{1/2}$. These local geometries may be thought of as an approximation to the Minkowski sum of a sphere and the elements on the simplicial complex, where the sphere's radius varies with particle thickness.

Codimension	0	1	2	3
Interior λ	0	$\frac{\mu_1}{\rho A}$	$2(\frac{\mu_2}{\rho\pi l_2})^{\frac{1}{2}}$	-
Boundary λ	0	$2(\frac{\mu_1}{\rho\pi l_1})^{\frac{1}{2}}$	$2(\frac{3\mu_2}{4\rho\pi})^{\frac{1}{3}}$	$2(\frac{3m}{4\rho\pi})^{\frac{1}{3}}$

Table 1: The thickness λ of particles for different codimensions.

5 Particle Evolution

We solve the incompressible Navier-Stokes equations in Lagrangian form

$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\vec{u} + \vec{f} \quad (1)$$

$$\nabla \cdot \vec{u} = 0 \quad (2)$$

where p is the pressure, ν is kinematic viscosity, and \vec{f} represents other forces such as gravity, surface tension, and adhesion. First, we use forward Euler to advect particles. Next, we address the potential changes in mesh connectivity using the updated particle positions. We apply explicit external forces to the particles to obtain $\vec{u}^* = \vec{u}^n + \Delta t \vec{f}/\rho$. We then apply implicit viscosity to obtain $\vec{u}^{**} = \vec{u}^* + \Delta t (\nu \nabla^2 \vec{u}^{**})$. This equation is solved independently for each Cartesian component of \vec{u} using a first order backward Euler time discretization. Finally, we solve for the pressure

$$\nabla \cdot \frac{1}{\rho} \nabla p = \frac{1}{\Delta t} \nabla \cdot \vec{u}^{**} \quad (3)$$

and apply the pressure forces via

$$\vec{u}^{n+1} = \vec{u}^{**} - \frac{\Delta t}{\rho} \nabla p. \quad (4)$$

6 Surface Tension

The net surface tension force applied to a particle is the sum of the contributions from all incident surface triangles and all incident segments. The triangle-based surface tension force is applied to the surface of the fluid volume and to both sides of the thin films. First, we find the triangles that are either surface faces of the tetrahedron mesh or triangles in the simplicial complex. Then, we duplicate each triangle of the simplicial complex and flip the orientation of the duplicate to account for both sides of the thin film. The surface tension forces applied from these triangles to their particles are calculated using the method of [Zheng et al. 2013]. That is, each triangle t contributes a surface tension force to each of its particles n as $\vec{f}_{t,n} = \sigma \vec{l}_{t,n}/2$ where σ is the surface tension coefficient and $\vec{l}_{t,n}$ is the length-weighted binormal perpendicular to the edge opposite particle n in triangle t in the plane of triangle t . The surface tension forces applied from the segments to their particles are calculated using the method of [Zheng et al. 2013] as well. That is, each segment s contributes a surface tension force to each of its particles n as $\vec{f}_{s,n} = \pi \lambda_n \sigma \vec{d}_{s,n}$ where λ_n is the thickness of particle n and $\vec{d}_{s,n}$ is the unit direction vector pointing from particle n to the opposite endpoint of segment s .

6.1 Rim-based Surface Tension

We use an additional rim-based surface tension force to calculate the surface tension contribution from the boundary segments of the triangle mesh to codimension-1 boundary particles. This is motivated by [Bush and Hasha 2004] which declares that the curvature force applied on the boundary of a thin sheet is associated with both the out-of-plane curvature (calculated above via triangles) and the in-plane curvature. The in-plane curvature force is associated with the curvature of the boundary centerline and acts as a force

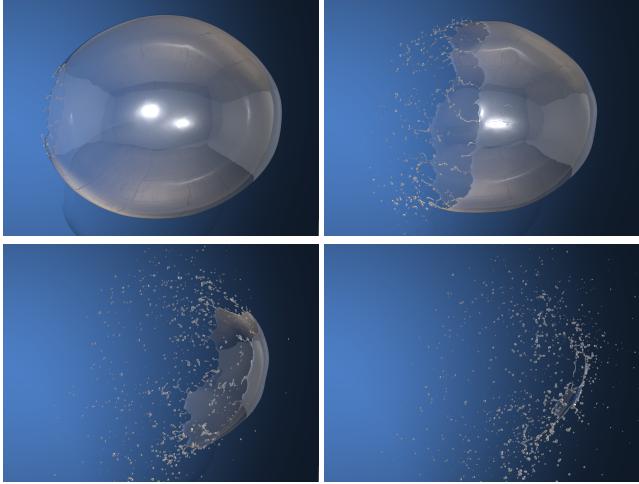


Figure 5: When a projectile breaks a bubble, surface tension causes the rim to retract creating numerous filaments and droplets.

to smooth its curvature. Each boundary segment s applies the in-plane curvature force to its two particles as $\vec{f}_{s,n} = c\hat{\lambda}\sigma\vec{d}_{s,n}$ where c is a constant (which equals $\pi - 1$ in [Bush and Hasha 2004]) and $\hat{\lambda} = (\lambda_i + \lambda_j)/2$ is the diameter of the boundary cylinder. In most of our simulations, the in-plane surface tension force is dominated by the out-of-plane surface tension force but still plays an important role in smoothing the in-plane curvature on the boundary of the triangle mesh. However, there are phenomena where the in-plane surface tension force has a dramatic effect on the physical behavior of the fluid. See Figure 4.

7 Codimensional Transitions

We allow geometry to transition from lower to higher codimensions as needed for both efficiency and modeling as the fluid features thin. When tetrahedra become too thin they degenerate into triangles (see Section 12). Both triangles and segments raise their codimension based on the particle thickness attribute. At each time step, the thickness of a particle is updated based on its mass and connectivity as described in Section 4. We define the thickness of a triangle λ_f to be the average of the thicknesses at its three particles, and the thickness of a segment λ_e to be the average of the thicknesses at its two particles. Whenever λ_f or λ_e are below a predefined threshold, we break the thin film or filament respectively simply by deleting the triangle or segment.

Addressing transitions from higher to lower codimensions is more complex. The philosophy behind our approach is that incompressibility can be described using different models on different length scales. If the thin film or filament thickness is significantly smaller than the length scales along the other directions, we can model incompressibility simply by updating particle thicknesses to keep volume constant. That is, particles separated by large distances are allowed to approach each other without resistance, resulting only in the thickening of the geometric representation. As the particles continue to increase their proximities, eventually the thickness becomes larger than we would like to allow in our higher codimensional representation and as such the codimension should decrease. This is accomplished by activating our novel incompressible flow solver on simplicial complexes for any element whose thickness (λ_f or λ_e) is larger than a predetermined threshold. Our switching on and off of the Poisson equation-based incompressibility solver bears some resemblance to previous works. [Losasso et al. 2008]

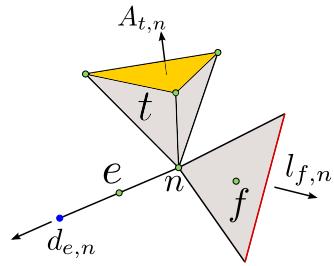


Figure 6: The volume weighted gradient at particle n depends on the incident tetrahedra, triangles, and segments. Pressure degrees of freedom are located at the green circles. See Section 8.1.

turned off the incompressibility to allow spray and foam regions to freely “slip” through the grid, and [Gerszewski and Bargteil 2013] (see also [Narain et al. 2009; Alduán and Otaduy 2011]) turned off the incompressibility in cells where separation was desired. Once incompressibility is enforced, we can implement transitions from higher to lower codimensions using the mesh operations described in Section 12.

8 Simplicial Incompressibility

Pressure degrees of freedom are placed on the particles of tetrahedra following the approaches of [Bonet and Burton 1998; Irving et al. 2007] to avoid locking. In contrast, we place the pressure degrees of freedom on the barycenters of triangles and the centers of segments where we do not expect significant locking because the triangles and segments can bend in and out of plane. This also avoids odd-even decoupling of the segment mesh.

8.1 Volume-Weighted Gradient

We define the total volume-weighted gradient at particle n as

$$W_n \nabla p = \left(V_n \nabla + \lambda_n A_n \nabla + \frac{\pi \lambda_n^2}{4} L_n \nabla \right) p \quad (5)$$

where $V_n \nabla$ represents the volume-weighted gradient contribution from incident tetrahedra, $A_n \nabla$ represents the area-weighted gradient contribution from incident triangles, $L_n \nabla$ represents the length-weighted gradient contribution from incident segments, λ_n is the particle thickness, and $W_n = V_n + \lambda_n A_n + (\pi \lambda_n^2 / 4) L_n$ is the total control volume for particle n obtained by summing of the volume contribution from each codimension (see Section 4).

To define the volume-weighted gradient contribution at particle n due to incident tetrahedra, we first assign a pressure \bar{p}_t to each tetrahedron t by averaging the pressures from its four particles. Then,

$$V_n \nabla p = \sum_{t \in T_n} \bar{p}_t \frac{\vec{A}_{t,n}}{3} \quad (6)$$

where T_n is all incident tetrahedra and $\vec{A}_{t,n}$ is the outward pointing area-weighted normal of the face opposite particle n in tetrahedron t (see [Zheng et al. 2013]). To define the area-weighted gradient contribution at particle n due to incident triangles, we note that the pressure p_f is already defined at the barycenter of each triangle f . So,

$$A_n \nabla p = \sum_{f \in F_n} p_f \frac{\vec{l}_{f,n}}{2} \quad (7)$$

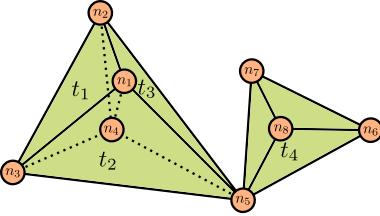


Figure 7: This tetrahedron mesh has 4 tetrahedra and 8 particles resulting in 4 equations and 8 degrees of freedom respectively. Thus, we expect \mathbf{H} to have a null space.

where F_n is all incident triangles. To define the length-weighted gradient contribution at particle n due to incident segments, we note that the pressure p_e is already defined at the center of each segment e . So,

$$L_n \nabla p = \sum_{e \in E_n} p_e \vec{d}_{e,n} \quad (8)$$

where E_n is all incident segments. See Figure 6.

We apply zero Dirichlet boundary conditions at all free surfaces. Neumann boundary conditions are enforced on solid objects and domain boundaries by modifying the volume-weighted gradient

$$W_n \nabla p = (\mathbf{I} - \vec{N} \vec{N}^T) \left(V_n \nabla + \lambda_n A_n \nabla + \frac{\pi \lambda_n^2}{4} L_n \nabla \right) p \quad (9)$$

where \vec{N} is the normal direction of the boundary.

8.2 Poisson Equation

Equations (5) through (8) combined define a matrix \mathbf{G} that maps a vector of pressure values (located at the particles of tetrahedra, barycenters of triangles, and centers of segments) to a vector of pressure gradients located at the particles. Thus, the volume-weighted divergence defined via the transpose operator, i.e. $-\mathbf{G}^T$, maps a vector of \vec{u} values (\mathbf{u}) at the particles to divergence values located at the particles of tetrahedra, barycenters of triangles, and centers of segments. Substituting the volume weighted divergence and gradient into Equation (3), we obtain the discretized volume-weighted Poisson equation written as

$$-\hat{\mathbf{G}}^T \mathbf{M}^{-1} \hat{\mathbf{G}} \mathbf{p} = -\frac{1}{\Delta t} \mathbf{G}^T \mathbf{u} \quad (10)$$

where $\hat{\mathbf{G}}$ is the volume-weighted gradient matrix with the modifications for Neumann boundary conditions as in Equation (9), \mathbf{p} is the vector of pressure values, and \mathbf{M} is a diagonal matrix of $W_n p$ values defined for each component of the gradient at the particles.

8.3 Null Space

Our solver must be capable of robustly solving a system with a null space. The standard conjugate gradient algorithm may fail to solve Equation (10) because the symmetric negative semidefinite system matrix $-\hat{\mathbf{G}}^T \mathbf{M}^{-1} \hat{\mathbf{G}}$ may have a null space associated with the tetrahedra. As noted in [Zheng et al. 2013], the volume-weighted gradient operator for tetrahedra can be factored into $\mathbf{G}_e \mathbf{H}$ where \mathbf{H} is the matrix that averages the pressures at the particles of the tetrahedra to their barycenters and \mathbf{G}_e is the standard volume-weighted gradient operator for pressure degrees of freedom defined at barycenters of tetrahedra. Thus, we can rewrite the system matrix as $-\mathbf{H}^T \mathbf{G}_e^T \mathbf{M}^{-1} \mathbf{G}_e \mathbf{H}$. Since it is common to avoid discretizing tetrahedra using pressures at tetrahedron barycenters because



Figure 8: Wind blowing on a thin film causes it to stretch into a neck and eventually pinch off to form air bubbles which may collide. The wind flow is simulated on a coarse background grid and is coupled with the bubbles in a force-based way. The bubbles enclose air volumes which are modeled using an equation-of-state based pressure force.

of locking, see e.g. [Irving et al. 2007], one can typically safely assume that $\mathbf{G}_e^T \mathbf{M}^{-1} \mathbf{G}_e$ does not have a null space. In this case, the null space of the system matrix would be equal to the null space of \mathbf{H} (note that while placing pressures at barycenters of tetrahedra causes locking for general tetrahedral meshes, there are some special cases where this is not true such as the BCC lattice, see [Ando et al. 2013]). Hence, we propose an algorithm to find a basis for the null space of \mathbf{H} . This basis is used to project out the null space components of the initial guess and subsequent search directions when using the conjugate gradient method to solve Equation (10).

Consider the non-manifold tetrahedron mesh shown in Figure 7. Here, \mathbf{H} is a 4×8 matrix that averages values from particles to tetrahedra. Suppose $\mathbf{H} \mathbf{z} = \mathbf{0}$ for some $\mathbf{z} \in \mathbb{R}^8$. Then, the equation for tetrahedron t_1 is $z_1 + z_2 + z_3 + z_4 = 0$, and therefore $z_4 = -(z_1 + z_2 + z_3)$ where z_1 , z_2 , and z_3 are independent variables. That is, the null space for this single tetrahedron is spanned by the columns of the following matrix

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}. \quad (11)$$

Incrementally, adding tetrahedron t_2 incorporates an additional degree of freedom for particle n_5 and an additional equation $z_1 + z_3 + z_4 + z_5 = 0$. Substituting in the equation above for z_4 results in $z_5 = z_2$, and thus the null space for these two tetrahedra is spanned by the columns of the following matrix

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -1 & -1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}. \quad (12)$$

Next, consider tetrahedron t_3 which adds no additional particles, and has equation $z_1 + z_2 + z_4 + z_5 = 0$. Substituting in the values above for z_4 and z_5 yields $z_2 = z_3$. Thus, the null space associated with these three tetrahedra is given by the columns of the following matrix

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix}. \quad (13)$$

Finally, consider tetrahedron t_4 which adds three additional degrees of freedom and has equation $z_5 + z_6 + z_7 + z_8 = 0$. This adds two additional independent variables z_6 and z_7 resulting in two additional columns for the matrix given by

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & -2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \\ z_8 \end{pmatrix}. \quad (14)$$

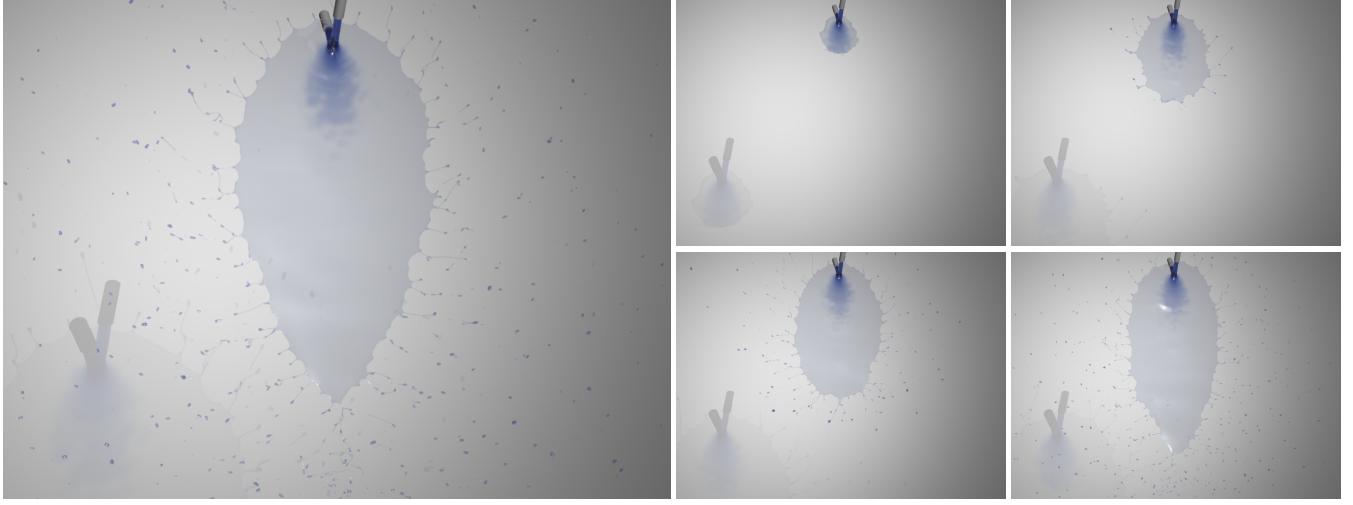


Figure 9: Two colliding liquid jets form a fluid fishbone [Hasha and Bush 2002]. Surface tension causes the rim of the thin sheet to pinch off into many filaments and droplets.

The example above demonstrates how we build up a null space basis for \mathbf{H} by traversing the tetrahedron mesh. First, we initialize every tetrahedron and particle to be unvisited. Starting from any unvisited tetrahedron, we assume the values at three particles to be independent and express the value on the fourth particle in terms of these independent variables (Equation 11). Then, we perform a breadth-first traversal of all tetrahedra that share particles. For each tetrahedron t encountered during the traversal, we check whether all of its particles have been visited. If all of its particles have not been visited, we add an independent variable and a column to the null space basis matrix for all unvisited particles except for one. Then, we calculate the value at the remaining unvisited particle. If all particles of the tetrahedron have been visited, we use the equation for the tetrahedron to express a column of the null space basis in terms of the other columns reducing the dimension of the null space by one. Finally, we mark tetrahedron t and all of its particles as visited. We repeat this traversal starting from any unvisited tetrahedron until all tetrahedra are visited. Then, the remaining columns of the null space basis matrix, if any, span the null space of \mathbf{H} . This procedure can be implemented using integer arithmetic which avoids floating point error during the traversal. When finished, we orthonormalize the null space basis using the modified Gram-Schmidt process [Golub and Loan 1996].

9 Viscosity

Since we use spatially constant viscosity, we can solve separately for each of the three spatial components [Rasmussen et al. 2004]. For the sake of exposition, we outline the discretization for the x -component of the velocity given by u (the others are discretized similarly). Our spatial discretization for the implicit velocity update over the simplicial complex uses variants of the gradient and divergence operators discussed in Section 8. Since the velocity degrees of freedom are colocated with the particles, we define the colocated total volume-weighted gradient at particle n as

$$\hat{W}_n \nabla u = \left(V_n \nabla + \lambda_n \hat{A}_n \nabla + \frac{\pi \lambda_n^2}{4} \hat{L}_n \nabla \right) u \quad (15)$$

where $\hat{A}_n \nabla$ represents the colocated area-weighted gradient contribution from incident triangles, $\hat{L}_n \nabla$ represents the colocated length-weighted gradient contribution from incident segments, and

$\hat{W}_n = V_n + \lambda_n \hat{A}_n + (\pi \lambda_n^2 / 4) \hat{L}_n$ is the total control volume for particle n . We compute \bar{u}_f at triangle barycenters by averaging the u velocities of the triangle's three particles, and then $\hat{A}_n \nabla u$ is defined by Equation (7) with p_f replaced by \bar{u}_f . We similarly define \bar{u}_e averaging from particles to segment centers, and define $\hat{L}_n \nabla u$ via Equation (8) with p_e replaced by \bar{u}_e . The definition of the volume-weighted gradient $\tilde{\mathbf{G}}$ and the volume-weighted divergence $-\tilde{\mathbf{G}}^T$ then follows naturally as outlined in Section 8. Combining this with backward Euler time integration gives

$$(\hat{\mathbf{W}} + \Delta t \tilde{\mathbf{G}}^T \tilde{\mathbf{M}}^{-1} \tilde{\mathbf{G}}) u^{n+1} = \hat{\mathbf{W}} u^n \quad (16)$$

where $\hat{\mathbf{W}}$ is a diagonal matrix of total control volumes \hat{W}_n for each particle defined in Equation (15) and $\tilde{\mathbf{M}}$ is a diagonal matrix of \hat{W}_n / ν values.

10 Air Pressure

The pinching and merging of thin films can result in regions of enclosed air, as seen in Figures 8 and 13. These volumes of enclosed air exert an outward pressure force that opposes surface tension and atmospheric pressure, creating visual effects such as bubble oscillations. We model the pressure force due to enclosed air by using the mesh to detect closed air volumes and applying equation of state-based air pressure forces to the fluid surface surrounding such volumes. Once the region becomes enclosed, we calculate its volume V_{air} and constant $k = p_{\text{atm}} V_{\text{air}}$ to be used in Boyle's law $p_{\text{air}} V_{\text{air}} = k$ where V_{air} is the current volume of the mesh. The air pressure force on each triangle bounding the enclosed air region is computed as $(p_{\text{air}} - p_{\text{atm}}) \vec{A}$ where \vec{A} is the outward pointing area-weighted normal of the triangle. The force on each triangle is distributed equally to its particles.

11 Adhesion

Adhesion is an important force for various phenomena such as the rings in Figure 3, the bottom of the waterbell in Figure 13, and the rock that causes the splash in Figure 14. In some scenarios, one can simply apply Neumann boundary conditions to force fluid to remain in contact with the solid, which works well for the rings in

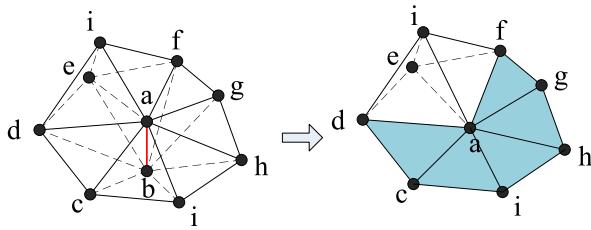


Figure 10: The tetrahedron edge collapse operation can result in a codimension-0 to codimension-1 transition. The collapse of edge ab degenerates 5 of the 7 tetrahedra sharing edge ab into triangles.

Figure 3. However, dynamically changing adhesive forces are more important for the waterbell and rock in Figure 13 and Figure 14 respectively. As in [Guendelman et al. 2003], we use a level set function as auxiliary information for the rigid bodies noting that the level set is defined outside the bodies as well. If a particle’s level set value ϕ is less than a user specified threshold ϕ_a , then we apply a force $F_a = \alpha((\phi - \phi_a)/\phi_a)\vec{N}$ where \vec{N} is the local outward pointing normal computed from the signed distance function, and α is the adhesion coefficient.

12 Meshing

The goals of our meshing algorithm are to maintain a temporally consistent mesh structure and to handle all transitions between codimensions. The key to our meshing approach is to use local mesh improvement operations in all codimensions—this especially helps with time coherency. Previous work in this area for generating high quality meshes of a single codimension includes [Wojtan et al. 2009; Brochu and Bridson 2009; Wicke et al. 2010; Zhang et al. 2012; Clausen et al. 2013]. Our meshing algorithm is based on these approaches with extensions to handle non-manifold simplicial complexes as well as transitions between codimensions. We first perform volumetric meshing operations, followed by thin sheet meshing operations, and finally filament meshing operations. Note that mass and other physical properties of the particles are redistributed based on changes in control volumes induced by meshing operations. Our meshing algorithm does not introduce breaks in thin films or filaments. The breaking of films and filaments is controlled by the triangle thickness λ_f and segment thickness λ_e as discussed in Section 4.

12.1 Volumetric Meshing

We utilize five meshing operations to improve the quality of tetrahedra, which is measured by the volume-length ratio as in [Klingner and Shewchuk 2008]. The *tetrahedron edge-face flip* [Klingner and Shewchuk 2008] supports 2-2, 2-3, 3-2, and 4-4 flips. The *tetrahedron*

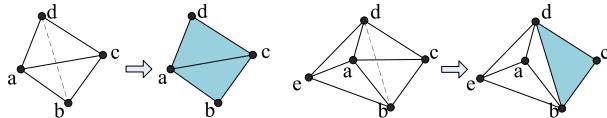


Figure 11: The flat boundary tetrahedron remove operation facilitates a codimension-0 to codimension-1 transition. (Left) None of the faces of tetrahedron abcd are connected to any other tetrahedron, so it is degenerated into triangles abc and acd. (Right) One of the faces of tetrahedron abcd is connected to another tetrahedron (aebd), so the abcd is degenerated into the face bcd.

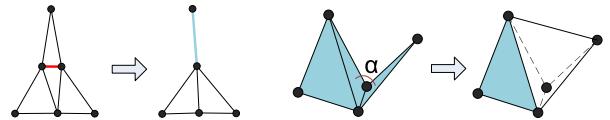


Figure 12: (Left) An edge collapse that results in a codimension-1 to codimension-2 transition. (Right) A triangle crumple merge that results in a codimension-1 to codimension-0 transition. The input triangles can either be triangles in the triangle mesh or boundary faces of the tetrahedron mesh.

dron edge split splits an overly long edge of the tetrahedron mesh by introducing a new particle. We also split an edge if its distance to either of its opposite vertices is smaller than a threshold (i.e. a “spade” tetrahedron). When an edge split is performed, the tetrahedra and triangles incident to that edge are also split. The *tetrahedron face split* splits a face of a tetrahedron into three faces if the tetrahedron is a “cap” tetrahedron where the distance of the face to its opposite particle is smaller than a threshold and the projected point of that particle onto the face plane is inside the face. Similar ideas can be seen in [Clausen et al. 2013] for face contraction. The tetrahedron edge split and tetrahedron face split may generate short edges which are collapsed via the *tetrahedron edge collapse*. This operation merges the particles of the short edge into a single particle. This handles a codimension-0 to codimension-1 transition that allows a fluid volume to become a thin sheet. See Figure 10. We also use *flat boundary tetrahedron remove* to enable transitions from codimension-0 to codimension-1 for sliver tetrahedra as shown in Figure 11 (far left). If none of its faces are connected to other tetrahedra, we simply choose one of the two pairs of in-plane triangles as the codimension-1 representation (Figure 11 left). If one of the faces is connected to another tetrahedron, then the other face in that pair degenerates to a codimension-1 triangle (Figure 11 right). When one pair of faces has two incident tetrahedra and the other pair has none, we can simply delete the other pair of faces.

12.2 Thin Film Meshing

We use four triangle mesh operations for thin film meshing. The *triangle edge flip* [Brochu and Bridson 2009] flips a triangle edge to maintain good triangle aspect ratios. The *triangle edge split* [Brochu and Bridson 2009] splits long triangle edges. The *triangle edge collapse* collapses a short edge by merging its particles. This operation handles the codimension-1 to codimension-2 transition that allows a thin sheet to devolve into a filament (Figure 12 left). The *triangle crumple merge* generates a new tetrahedron between two incident triangles when their dihedral angle is smaller than a threshold, resulting in a codimension-1 to codimension-0 transition where thin films become fluid volumes (Figure 12 right). The collision between elements of different codimensions is handled by deleting the particles and elements of one codimension and merging their mass into the particles of the other codimension. Users can specify the direction of merging between codimensions. E.g., in some examples a droplet represented by tetrahedra may merge into a sheet of triangles via a codimension-0 to codimension-1 merge (see Figure 3), while in other examples a sheet of triangles may merge into a fluid volume via a codimension-1 to codimension-0 merge (see Figure 14).

12.3 Filament Meshing

We use two segment mesh operations for filament meshing. The *segment edge split* splits a long segment into two segments by adding a particle. The *segment edge collapse* merges the particles of a short segment into a single particle. When this segment is iso-

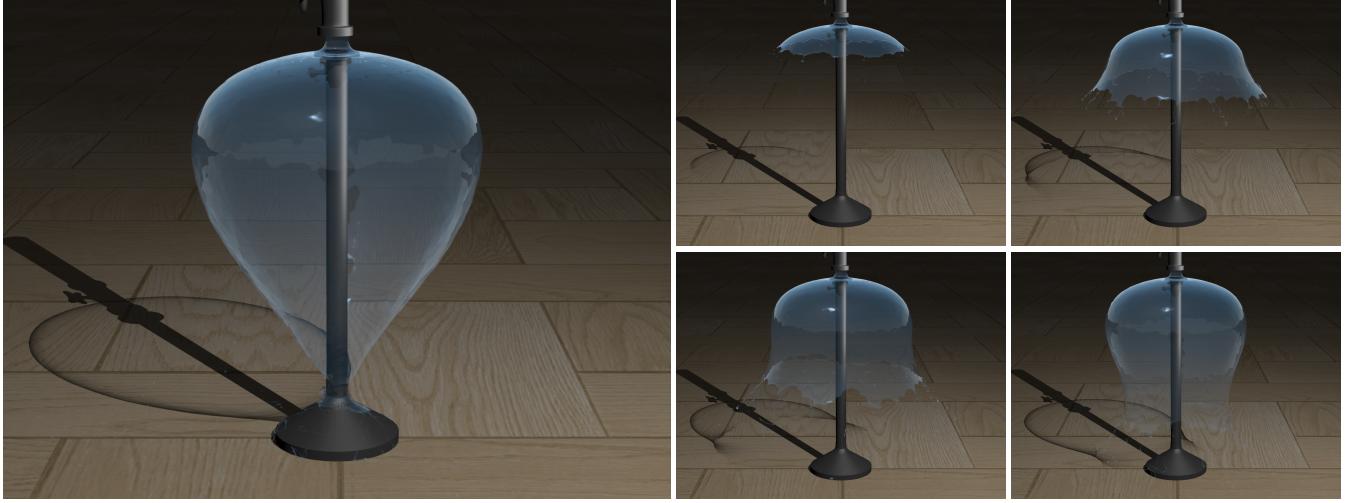


Figure 13: A vertical jet of water strikes a circular horizontal impactor which results in a circular water sheet referred to as a waterbell [Clanet 2007]. The circular sheet of fluid is represented using triangles while the pinch-offs are represented using segments. The circular sheet adheres to the pillar via adhesion and encloses an air volume which is modeled using an equation-of-state.

lated, it becomes a single point, constituting a codimension-2 to codimension-3 transition. Note that we do not allow segments to directly transform into triangles although one could. Instead, we prefer to transform segments directly into tetrahedra by applying a Delaunay remeshing algorithm [Zheng et al. 2013] to codimension-2 particles that are close together.

12.4 Adaptivity

We add adaptivity into our meshing algorithm in order to put more degrees of freedom near the boundaries of thin films and on filaments for greater visual detail. We place more degrees of freedom on filaments by setting the minimum and maximum allowed edge lengths of segments to be smaller than those for triangles and tetrahedra. For thin films, we place successively smaller triangles near the film’s rim. This is accomplished by setting the minimum and maximum allowable edge lengths for the edge collapse and edge split operations based on the graph distances of particles to the rim. At each timestep, before meshing operations are applied, an integer value is assigned to each particle to represent its graph distance to the rim. The minimum and maximum allowed edge lengths increase as a function of the graph distances of an edge’s incident particles up to a threshold, after which they are held constant.

13 Skinned Mesh

Just as an articulated skeleton drives a surface mesh via skinning, our simulated degrees of freedom are a proxy stand-in for the physical mesh that one would observe. Thus, in a post-processing step we use the degrees of freedom from the simulation to construct a representation of the fluid surface. For clarity, we use “nodes” to refer to vertices of the skinned mesh to distinguish them from the particles of the simulation mesh.

The skinned mesh is generated from elements of the simulation mesh according to their codimensions. Points are converted into tessellated spheres. Segments that are not connected to any triangles or tetrahedra are converted into tessellated surfaces. Each particle on a segment is expanded into a ring of nodes, where the radius of the ring is derived by smoothing the thickness values (derived in Section 4) of the particle and its neighbors. The axis about which

the ring is created is calculated using the directions along segments incident to the particle. If a particle is on the codimension-2 boundary, the surface of the skinned mesh at that point is sealed with a tessellated hemisphere, producing rounded-off filament ends. Each triangle is expanded into two new triangular faces, and we align the orientation of neighboring triangles such that these expanded triangles can be labeled as part of either the *upper* or *lower* layer. These two layers are then stitched together along the codimension-1 boundary. This process is described in Algorithm 1. For tetrahedra, we simply add the boundary faces of the tetrahedron mesh to the skinned mesh.

Algorithm 1 Codimension-1 Skinning Process

- 1: // Expand triangles into two layers.
 - 2: Align triangle normals.
 - 3: Add upper node and lower node to skinned mesh for each particle on codimension-1 mesh.
 - 4: Mesh upper and lower layers of triangles such that their normals are opposites.
 - 5: **for all** upper and lower nodes **do**
 - 6: Average incident triangle normals to find the node normal.
 - 7: Move node a distance of $\lambda/2$ in its normal direction.
 - 8: **end for**
 - 9: // Stitch layers together at the fluid rim.
 - 10: Add semicircle of nodes for particles on codimension-1 boundary.
 - 11: Mesh nodes of neighboring semicircles together to seal rim.
 - 12: **if** gap is detected in rim due to tetrahedron connected to triangle boundary **then**
 - 13: Mesh semicircle of nodes from codimension-1 particle to node from codimension-0 particle.
 - 14: **end if**
-

Special consideration must be given to skinning the simulation mesh at particles of multiple codimensions. On the interface between tetrahedron and triangle elements, we test the topology of the simulation mesh to determine whether tetrahedron faces should be connected to the upper or lower layer of the expanded triangles. If a tetrahedron shares an edge with a triangle, we test its topology by considering the winding order of its particles. If the particles of the common edge are in the opposite order on the triangle than they are

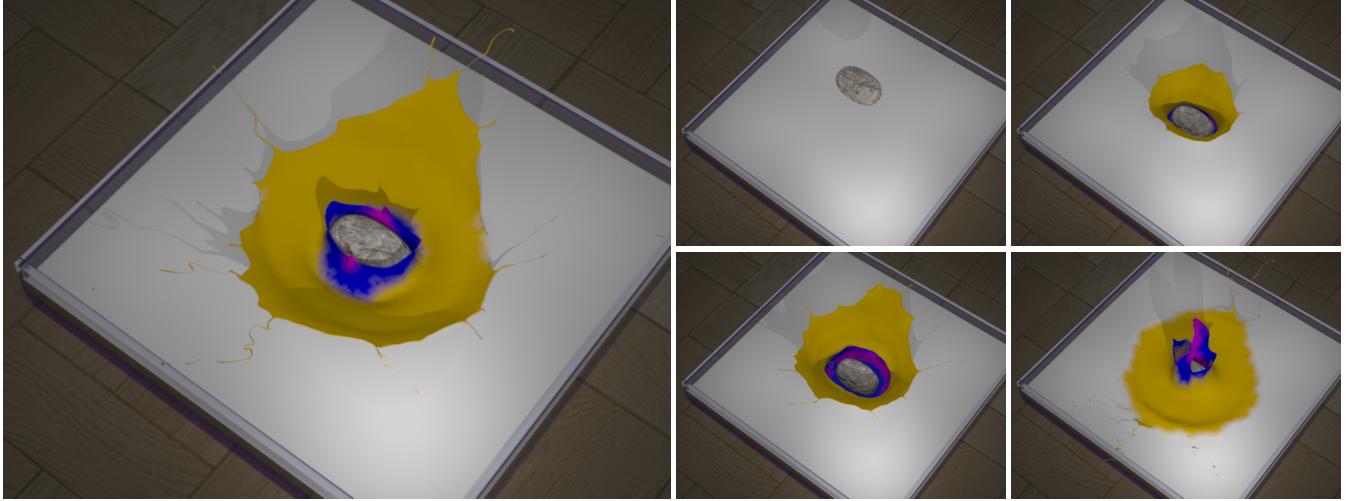


Figure 14: To demonstrate that our method is effective for simulations with a large volumetric component as well, we drop a rock into a tank filled with layers of colored paint. While the fluid in the tank remains composed predominantly of tetrahedra, the resulting splash tapers into a thin sheet and pinches off into filaments along its rim. The thin sheet and filaments later re-merge with the fluid volume.

on the tetrahedron face, then the tetrahedron face is connected to the upper layer; otherwise it is connected to the lower layer. This test is not sufficient, however, to evaluate a tetrahedron face that has only a single particle in common with the triangle mesh. Thus, we use a breadth-first search as a heuristic for assigning upper/lower layers. Beginning with the set of tetrahedron faces whose upper/lower layer is known (because they share an edge with a triangle), we assign the same upper/lower layer to unvisited neighboring tetrahedron faces.

Particles of multiple codimensions are also found where a segment is connected to the boundary of the triangle mesh or the tetrahedron mesh. Because we want to visualize the transitions from thin sheets to filaments in various fluid phenomena, it is especially important to maintain these connectivities in the skinned mesh. In the case of a segment connected to the triangle boundary, we stitch a tessellated representation of the segment to the expanded triangle boundary. First, we expand the particle at the junction into two semicircles of nodes. Each semicircle is composed of n nodes, where the first and last nodes are shared by both semicircles. Each semicircle is meshed with the nodes corresponding to one of the particle's two neighbors on the codimension-1 boundary to seal the fluid rim. We then create a tessellated surface comprised of two rings of nodes around the protruding segment. The two semicircles together constitute a simple cycle of $2n - 2$ nodes. This cycle is taken to be the first ring of the surface. The second ring is created by expanding the other particle on the segment into a ring of $2n - 2$ nodes. Triangles are added between these two rings to seal the surface. In the case of a segment connected to a tetrahedron, we opt to take a simple approach. The particle of the segment that is not incident to the tetrahedron is expanded into a ring, but the particle incident to the tetrahedron is not expanded. Triangles are added between the resulting nodes to form a cone.

After transforming the simulation degrees of freedom into the skinned mesh representation, we take steps to improve the quality of the mesh. First, we correct non-manifold edges. These edges are created as a by-product of the naturally non-manifold simplicial complex; i.e., a section of a sheet represented by tetrahedra may have a tetrahedral edge or particle that is on the surface of both sides of the sheet. In these cases, the skinning process duplicates the non-manifold edge or particle and correspondingly adjusts the surface faces of the tetrahedra so that the two sides of the skinned

mesh are not attached. Finally, we apply smoothing operations to the skinned mesh to improve its appearance. For most cases we find that Laplacian smoothing [Sorkine et al. 2004] produces reasonable results. Where it is important to smooth fluid features without losing volume (e.g., for the water droplets in Figure 3), we selectively apply Loop subdivision [Loop 1987] to the isolated parts of the simulation mesh that contain relatively few particles.

14 Examples

In the film catenoid example (Figure 3), a membrane suspended between two rings is stretched until surface tension forces cause the membrane to pinch off and separate into two membranes, producing many small droplets in the process. In the bubble burst example (Figure 5), a projectile breaks a bubble. In the bubble blow example (Figure 8), wind blowing on a membrane causes it to expand and eventually pinch off due to surface tension.

In Figure 9, two colliding liquid jets form a fluid fishbone composed of a thin sheet with filaments and droplets around its rim. A waterbell, shown in Figure 13, is formed when a volumetric jet of fluid strikes a pillar and spreads out in the shape of an umbrella before the combination of surface tension forces and gravity pinches the circular thin fluid sheet into the shape of a bell. The fluid polygon shown in Figure 4 also begins with a volumetric jet of liquid striking a horizontal impactor and spreading out. In Figure 14, we drop a rock into a paint-filled tank to form a multicolored paint splash [Oefner 2013].

The runtimes of these examples vary based on the physical processes being simulated and the resolution of the mesh. The solver and mesher exhibit reasonable runtimes, and an optimized implementation could yield even better performance results. Each of the simulations runs on a desktop machine using between one and twelve cores. The least expensive example is the fluid polygon with about 18,000 particles, which runs at an average of 6 seconds per frame. The per-frame runtime averages of the fishbone, film catenoid, and bubble burst examples are 22 seconds, 11 seconds, and 23 seconds, respectively. With a smaller timestep, the bubble blow example is slower and runs at an average of 90 seconds per frame. The most expensive examples are the waterbell with about 165,000 particles, which runs on a single core at 10 to 30 minutes

per frame depending on frame complexity, and the paint splash with about 140,000 particles, which runs at about 30 minutes per frame.

15 Discussions and Conclusions

We presented a novel method for simulating codimensional surface tension driven phenomena on simplicial complexes. Our method models fluid features (volumes, thin films, filaments, and droplets) in different codimensions and solves incompressible flow with surface tension in those codimensions in a unified way. This enables the simulation of many sophisticated interfacial phenomena.

One interesting direction for future work is to improve the surface tension model on thin films and filaments. It is difficult for our current method to accurately model the highly regular structures of the fluid polygon and fishbone seen in real experiments [Buckingham and Bush 2001; Hasha and Bush 2002]. Although similar results can be achieved by applying artistic controls, such controls may introduce artifacts (e.g., a nonphysical acceleration of the filaments in Figure 4). Another limitation of our method is its assumption that fluid is a continuum: it does not account for atomistic effects in vanishingly thin fluids, which can be important in phenomena like breaking fluid sheets. Reduced thin film [Ribe 2002] and filament [Ribe 2004] models constitute another research direction and the way to mix those models in different codimensions would be interesting to explore further. The primary distinction of our method is its ability to solve the full Navier-Stokes equations with surface tension on different codimensions in a unified way.

There are many other interesting avenues for future work as most of the real-world phenomena that we considered have received little to no attention from the simulation community due to their inherent complexities. For example, we have only barely touched upon the plethora of potential meshing operations not only for simulations but also for reconstructing a skinned mesh from the simulation degrees of freedom. Parallelization of numerical methods on simplicial complexes also poses various challenges similar to but more severe than those for methods that rely heavily on meshing in a single codimension. Surveying the literature, it appears that some of these complex phenomena are only approximately understood especially when compared to the more thorough understanding that researchers have of volumetric fluid effects. It might prove fruitful to collaborate with those researchers as the ability to simulate these phenomena numerically enables a wide variety of parameter studies including those that may be difficult, impossible, or even nonphysical in order to develop a better understanding of these various phenomena.

Acknowledgements

Research supported in part by ONR N00014-13-1-0346, ONR N00014-09-1-0101, ONR N-00014-11-1-0027, ONR N00014-11-1-0707, ARL AHPCRC W911NF-07-0027, and the Intel Science and Technology Center for Visual Computing. Computing resources were provided in part by ONR N00014-05-1-0479. M.C. was supported in part by an NDSEG.

References

- ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. J. 2007. Adaptively sampled particle fluids. *ACM Trans. Graph. (SIGGRAPH Proc.)* 26, 3.
- AKINCI, N., AKINCI, G., AND TESCHNER, M. 2013. Versatile surface tension and adhesion for SPH fluids. *ACM Trans. Graph.* 32, 6 (Nov.), 182:1–182:8.
- ALDUÁN, I., AND OTADUY, M. A. 2011. SPH granular flow with friction and cohesion. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, SCA '11, 25–32.
- ANDO, R., THUREY, N., AND TSURUNO, R. 2012. Preserving fluid sheets with adaptively sampled anisotropic particles. *Visualization and Computer Graphics, IEEE Transactions on* 18, 8, 1202–1214.
- ANDO, R., THÜREY, N., AND WOJTAN, C. 2013. Highly adaptive liquid simulations on tetrahedral meshes. *ACM Trans. Graph. (Proc. SIGGRAPH 2013)* 32, 4, 103:1–103:10.
- BATTY, C., XENOS, S., AND HOUSTON, B. 2010. Tetrahedral embedded boundary methods for accurate and flexible adaptive fluids. In *Proceedings of Eurographics*, vol. 29, 695–704.
- BATTY, C., URIBE, A., AUDOLY, B., AND GRINSPUN, E. 2012. Discrete viscous sheets. *ACM Trans. Graph. (TOG)* 31, 4, 113.
- BECKER, M., AND TESCHNER, M. 2007. Weakly compressible SPH for free surface flows. In *Proc. of the 2007 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 209–217.
- BERGOU, M., AUDOLY, B., VOUGA, E., WARDETZKY, M., AND GRINSPUN, E. 2010. Discrete viscous threads. *ACM Transactions on Graphics (TOG)* 29, 4, 116.
- BONET, J., AND BURTON, A. 1998. A simple average nodal pressure tetrahedral element for incompressible and nearly incompressible dynamic explicit applications. *Comm. Num. Meth. Eng.* 14, 437–449.
- BROCHU, T., AND BRIDSON, R. 2009. Robust topological operations for dynamic explicit surfaces. *SIAM Journal on Scientific Computing* 31, 4, 2472–2493.
- BROCHU, T., BATTY, C., AND BRIDSON, R. 2010. Matching fluid simulation elements to surface geometry and topology. *ACM Trans. Graph. (SIGGRAPH Proc.)*, 47:1–47:9.
- BUCKINGHAM, R., AND BUSH, J. W. M. 2001. Fluid polygons. *Physics of Fluids (1994-present)* 13, 9, S10–S10.
- BUSH, J. W., AND HASHA, A. E. 2004. On the collision of laminar jets: fluid chains and fishbones. *Journal of fluid mechanics* 511, 285–310.
- CHEN, X., MA, D., AND YANG, V. 2012. Dynamics and stability of impinging jets. *ILASS Americas, 24th Annual Conference on Liquid Atomization and Spray Systems*.
- CLANET, C. 2007. Waterbells and liquid sheets. *Annu. Rev. Fluid Mech.* 39, 469–496.
- CLAUSEN, P., WICKE, M., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2013. Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Trans. Graph.* 32, 2 (Apr.), 17:1–17:15.
- DURIKOVIC, R. 2001. Animation of soap bubble dynamics, cluster formation and collision. *Comput. Graph. Forum* 20, 3, 67–76.
- FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In *Proc. of ACM SIGGRAPH 2001*, 23–30.
- GERSZEWSKI, D., AND BARGTEIL, A. W. 2013. Physics-based animation of large-scale splashing liquids. *ACM Trans. Graph.* 32, 6 (Nov.), 185:1–185:6.
- GOLUB, G., AND LOAN, C. 1996. *Matrix Computations*. The John Hopkins University Press, Baltimore.

- GUENDELMA, E., BRIDSON, R., AND FEDKIW, R. 2003. Non-convex rigid bodies with stacking. *ACM Trans. Graph. (SIGGRAPH Proc.)* 22, 3, 871–878.
- HASHA, A. E., AND BUSH, J. W. M. 2002. Fluid fishbones. *Physics of Fluids (1994-present)* 14, 9, S8–S8.
- IDELOHN, S., ONATE, E., AND DEL PIN, F. 2004. The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International Journal For Numerical Methods in Engineering* 61, 964–989.
- IRVING, G., SCHROEDER, C., AND FEDKIW, R. 2007. Volume conserving finite element simulations of deformable models. *ACM Trans. Graph. (SIGGRAPH Proc.)* 26, 3, 13.1–13.6.
- KIM, B., LIU, Y., LLAMAS, I., JIAO, X., AND ROSSIGNAC, J. 2007. Simulation of bubbles in foam with the volume control method. In *Proc. of ACM SIGGRAPH 2007*, 98:1–98:10.
- KIM, D., SONG, O.-Y., AND KO, H.-S. 2009. Stretching and wiggling liquids. *ACM Trans. Graph. (SIGGRAPH Asia Proc.)* 28, 5, 120:1–120:7.
- KLINGNER, B. M., AND SHEWCHUK, J. R. 2008. Aggressive tetrahedral mesh improvement. In *Proceedings of the 16th International Meshing Roundtable*, Springer, 3–23.
- LOOP, C. 1987. *Smooth subdivision surfaces based on triangles*. Master's thesis, The University of Utah.
- LOSASSO, F., GIBOU, F., AND FEDKIW, R. 2004. Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH Proc.)* 23, 457–462.
- LOSASSO, F., TALTON, J., KWATRA, N., AND FEDKIW, R. 2008. Two-way coupled SPH and particle level set fluid simulation. *IEEE TVCG* 14, 4, 797–804.
- MARTIN, S., KAUFMANN, P., BOTSCHE, M., GRINSPUN, E., AND GROSS, M. 2010. Unified simulation of elastic rods, shells, and solids. *ACM Trans. Graph.* 29, 4, 39:1–39:10.
- MISZTAL, M. K., BRIDSON, R., ERLEBEN, K., BÆRENTZEN, J. A., AND ANTON, F. 2010. Optimization-based fluid simulation on unstructured meshes. In *Proc. of the 7th Workshop on Virtual Reality Interactions and Physical Simulations (VRPhys2010)*, 11–20.
- MISZTAL, M. K., ERLEBEN, K., BARGTEIL, A., FURSUND, J., CHRISTENSEN, B., BÆRENTZEN, J. A., AND BRIDSON, R. 2012. Multiphase flow of immiscible fluids on unstructured moving meshes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, 97–106.
- MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 154–159.
- NARAIN, R., GOLAS, A., CURTIS, S., AND LIN, M. C. 2009. Aggregate dynamics for dense crowd simulation. In *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 122.
- OEFNER, F. 2013. Orchid. <http://fabianoefner.com/?portfolio=orchid>.
- PATKAR, S., AANJANEYA, M., KARPMAN, D., AND FEDKIW, R. 2013. A hybrid Lagrangian-Eulerian formation for bubble generation and dynamics. In *Proceedings of 2013 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 105–114.
- PFAFF, T., THUREY, N., AND GROSS, M. 2012. Lagrangian vortex sheets for animating fluids. *ACM Trans. Graph.* 31, 4, 112:1–112:8.
- PREMŽOE, S., TASDIZEN, T., BIGLER, J., LEFOHN, A., AND WHITAKER, R. T. 2003. Particle-based simulation of fluids. In *Computer Graphics Forum*, vol. 22, Wiley Online Library, 401–410.
- RASMUSSEN, N., ENRIGHT, D., NGUYEN, D., MARINO, S., SUMNER, N., GEIGER, W., HOON, S., AND FEDKIW, R. 2004. Directable photorealistic liquids. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 193–202.
- RIBE, N. 2002. A general theory for the dynamics of thin viscous sheets. *Journal of Fluid Mechanics* 457, 255–283.
- RIBE, N. M. 2004. Coiling of viscous jets. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 460, 2051, 3223–3239.
- SAYE, R. I., AND SETHIAN, J. A. 2013. Multiscale modeling of membrane rearrangement, drainage, and rupture in evolving foams. *Science* 340, 6133, 720–724.
- SIN, F., BARGTEIL, A. W., AND HODGINS, J. K. 2009. A point-based method for animating incompressible flow. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, ACM, New York, NY, USA, SCA '09, 247–255.
- SORKINE, O., COHEN-OR, D., LIPMAN, Y., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, ACM, 175–184.
- STAM, J. 2009. Nucleus: Towards a unified dynamics solver for computer graphics. In *Computer-Aided Design and Computer Graphics, 2009. CAD/Graphics' 09. 11th IEEE International Conference on*, IEEE, 1–11.
- THÜREY, N., WOJTAN, C., GROSS, M., AND TURK, G. 2010. A multiscale approach to mesh-based surface tension flows. *ACM Trans. Graph. (TOG)* 29, 4, 1–10.
- WICKE, M., RITCHIE, D., KLINGNER, B., BURKE, S., SHEWCHUK, J., AND O'BRIEN. 2010. Dynamic local remeshing for elastoplastic simulation. In *Proc. of ACM SIGGRAPH 2010*, 49:1–49:11.
- WOJTAN, C., THÜREY, N., GROSS, M., AND TURK, G. 2009. Deforming meshes that split and merge. In *ACM Trans. Graph. (TOG)*, vol. 28, ACM, 76.
- YU, J., AND TURK, G. 2010. Reconstructing surfaces of particle-based fluids using anisotropic kernels. In *Proc. of the 2010 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 217–225.
- YU, J., WOJTAN, C., TURK, G., AND YAP, C. 2012. Explicit mesh surfaces for particle based fluids. *Comp. Graph. Forum* 31, 2pt4 (May), 815–824.
- ZHANG, Y., WANG, H., WANG, S., TONG, Y., AND ZHOU, K. 2012. A deformable surface model for real-time water drop animation. *Visualization and Computer Graphics, IEEE Transactions on* 18, 8, 1281–1289.
- ZHENG, W., ZHU, B., KIM, B., AND FEDKIW, R. 2013. A new incompressibility discretization for a hybrid particle MAC grid representation with surface tension. (*submitted*).