Surface-Only Liquids

Fang Da
Columbia University

David Hahn
IST Austria

Christopher Batty
University of Waterloo

Chris Wojtan
IST Austria

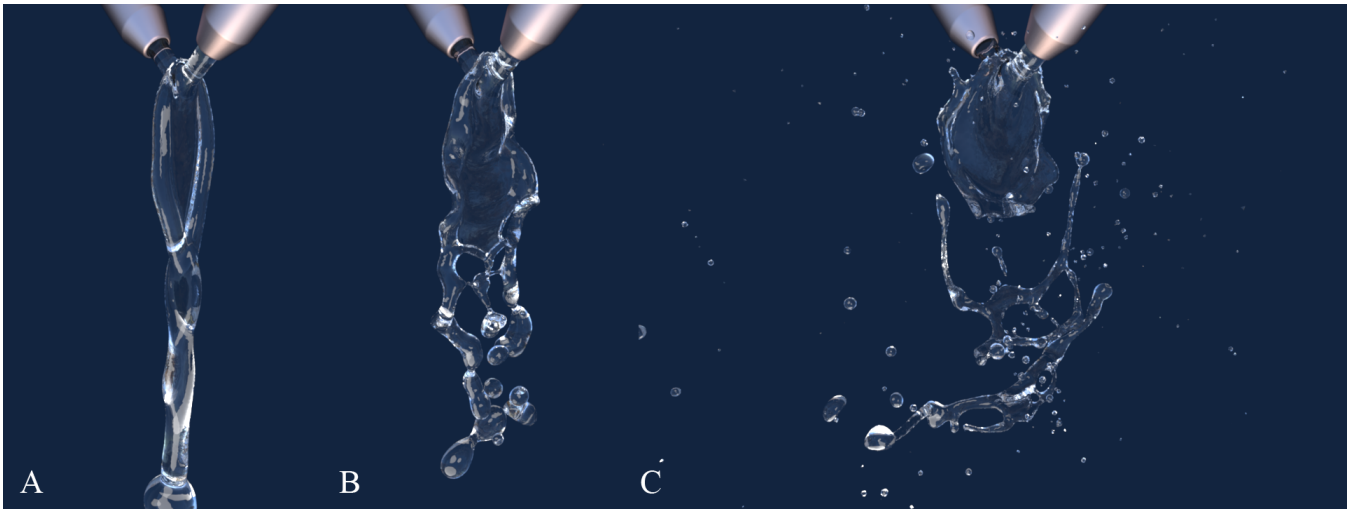Eitan Grinspun
Columbia University

Figure 1: *Reproduction of various patterns resulting from the collision of jets. (A) Fluid chains. (B) Disintegrating sheets. (C) Violent flapping.*

Abstract

We propose a novel surface-only technique for simulating incompressible, inviscid and uniform-density liquids with surface tension in three dimensions. The liquid surface is captured by a triangle mesh on which a Lagrangian velocity field is stored. Because advection of the velocity field may violate the incompressibility condition, we devise an orthogonal projection technique to remove the divergence while requiring the evaluation of only two boundary integrals. The forces of surface tension, gravity, and solid contact are all treated by a boundary element solve, allowing us to perform detailed simulations of a wide range of liquid phenomena, including waterbells, droplet and jet collisions, fluid chains, and crown splashes.

Keywords: liquids, surface tension, boundary element method, Helmholtz decomposition

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

1 Introduction

Many fascinating liquid phenomena, such as the crown splash adored by artists and scientists alike and the fluid chains formed by colliding jets [Bush and Hasha 2004], are driven by subtle balances between

the surface tension force and the inertia of the liquid. The deforming free surface, a defining aspect of liquids, carries the geometric information that drives surface tension effects, which are in turn responsible for the characteristic look and feel of familiar liquids like water. Undoubtedly, capturing the free surface with sufficient detail is crucial to the successful re-creation of these liquid phenomena in computer simulations. From a computer graphics viewpoint, the free surface is also the only visually important element of the final imagery, as it is where the most prominent optical effects such as reflection and refraction happen.

However, traditional fluid simulation techniques, including spatial grids [Foster and Fedkiw 2001], volumetric meshes [Misztal et al. 2010], and particles [Ihmsen et al. 2014b], invest many of their degrees of freedom deep inside the liquid and far from the free surface. Populating the entire *volume* of the liquid is clearly uneconomical, and as we will demonstrate, it is also unnecessary for many phenomena of interest. In a traditional free surface flow solver, volumetric samples serve two roles: (a) to represent an arbitrary velocity field inside the liquid volume, and (b) to enforce incompressibility. Recent numerical experiments suggest that internal vorticity plays only a small role in the perceived dynamics of liquids [Zhang et al. 2015], motivating us to disregard the first role of volumetric samples. But what about incompressibility? Discarding internal samples and working with variables only on the liquid surface, incompressibility can be enforced by projecting the velocity or position of the surface [Zhang et al. 2012], or by working with a surface velocity representation that is, by construction, incompressible [Brochu et al. 2012; Keeler and Bridson 2014]. Motivated by these observations, we propose the **first surface-based treatment of general 3D liquid bodies dominated by surface tension and inertia**.

The first challenge in a surface-only numerical scheme is finding a sufficiently expressive velocity field representation that can be stored on the surface. Capturing an arbitrary three dimensional velocity field with a surface-only representation is an inherently ill-posed problem. Fortunately, many liquid animation scenarios are well approximated by a reduced space of *harmonic velocity fields*

induced by two simplifying assumptions.

First, liquids are typically assumed to be incompressible, which reduces the space of velocity fields to those that are *divergence-free*.

Second, the three primary sources of vorticity in typical flows of interest are baroclinity due to density gradients, surface tension due to curvature gradients, and interaction with solid boundaries due to viscosity. However, for a liquid with uniform internal density, density gradients and surface tension forces occur only at the liquid-air surface. Furthermore, for the idealized case of inviscid flow described by the Euler equations, viscosity is absent and as a result vorticity generated at the solid or air surfaces *does not propagate into the interior* [Stock 2006]. This motivates the assumption that the interior of the liquid volume is irrotational, reducing the space of velocity fields to those that are *curl-free*.

Our work builds on these two assumptions. We develop a surface-based numerical treatment of volumetric liquids, inspired by the representations that become possible when an internal velocity field is assumed to be divergence- and curl-free. For a contractible domain, such as a liquid droplet, the *harmonic vector field* can be expressed as the gradient of a harmonic scalar function. When this integrability condition holds, it becomes possible to express the volumetric velocity field, and in turn to integrate the Euler equations, in terms of only surface position and velocity. Even when we consider scenarios beyond the scope of the integrability condition, the surface-based representation continues to produce visually compelling animations reproducing fascinating liquid behavior (see Figure 1).

This lower dimensional subspace of vector fields has previously been shown to possess a surprising degree of expressive power through successful applications to a variety of phenomena ranging from droplet impact [Davidson 2000] and smoke [Brochu et al. 2012; Weißmann and Pinkall 2010], to soap films [Da et al. 2015] and ocean waves [Xue et al. 2001; Keeler and Bridson 2014]. Unfortunately, existing surface-only techniques are not suited to reliable simulation of many common liquid phenomena, such as splashing and water jets. Vortex methods have difficulty treating baroclinity robustly (the Boussinesq approximation is not applicable to the liquid-air interface), while potential flow approaches suffer from stability issues due to the nonlinearity of the Bernoulli equation. For most liquid phenomena, a full 3D simulation (grid-based, volumetric mesh-based, or particle-based) is currently the only feasible approach.

We propose a novel framework that is exempt from these pitfalls, for robust and flexible simulation of incompressible liquid of uniform density, with surface tension, gravity, and contact with solid objects. By making velocity the simulation state variable, we avoid the destabilizing effect of nonlinear time integration: the nonlinearity in the advective term of the Euler equations is easily handled on a Lagrangian mesh in the form of the total derivative.

Working directly with velocity in a surface-only setting requires a new set of tools. We present the first advection-projection scheme for surface-based liquids, which proceeds in three steps: a Lagrangian mesh advection phase; a surface-only projection phase based on the Helmholtz decomposition that requires only the evaluation of two boundary integrals; and a final boundary element solve to integrate forces such as surface tension, gravity, and solid contact.

The use of an explicit triangle mesh for surface tracking allows for easy and accurate surface tension force computation, which is especially crucial at the three-phase junction where the solid, liquid, and air phases meet. We solve for the pressure and its normal derivative on the surface using a first-order boundary element method (BEM) solver, where the surface tension force and the solid contact force

are conveniently described by Dirichlet and Neumann boundary conditions, respectively. The resulting numerical scheme successfully reproduces a host of surface tension-dominated liquid phenomena (see Figure 2 for an example).
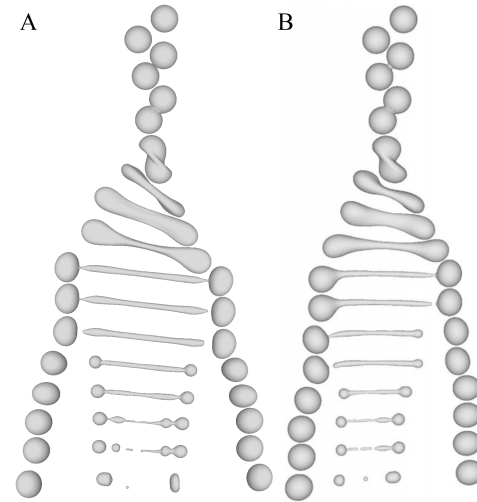

A   B

Figure 2: *Comparison of the evolution of droplets in an off-center coalescence test between our method (A) and an unstructured tetrahedral mesh method (B) [Quan et al. 2009].*

2 Related Work

Free surface flow simulation has been a central topic in the broader literature of computational fluid dynamics since the field's inception. Characterized by a free surface separating a liquid region from an exterior air region assumed to be massless, it presents various challenges in interface tracking, discretization of the governing physical equations, and treatment of liquid-solid-air interaction.

Grid and volumetric meshes   The dominant approach to liquid simulation discretizes the space in and around the liquid into a grid, and stores on it the velocity field of the fluid [Foster and Metaxas 1996; Foster and Fedkiw 2001]. The surface evolution is tracked using one of a range of techniques, such as the popular level set method [Osher and Sethian 1988; Enright et al. 2002]. Alternative discretizations include conforming volumetric meshes [Chentanez et al. 2007; Misztal et al. 2010; Clausen et al. 2013], embedded volumetric meshes [Batty et al. 2010], and particle methods [Ihmsen et al. 2014b]. The spatial discretization of the liquid volume is subsequently used to approximate and solve the incompressible Euler or Navier-Stokes equations.

These approaches suffer from memory and computational costs proportional to the volume of the simulation domain due to the discretization of the liquid interior. Furthermore, conforming volumetric mesh methods such as that of Clausen et al. [2013] spend a significant fraction of their computation time maintaining the quality of the mesh interior, while grid-based methods require more complex construction of differential operators to treat surface tension [Fedkiw et al. 1999; Hong and Kim 2005] and triple junctions [Wang et al. 2005] in the absence of a conforming mesh.

Mesh-based surface tracking for Eulerian fluids   In contrast to level set- or particle-based approaches for representing liquid geometry, explicit surface meshes [Wojtan et al. 2011] store geometric information strictly on the surface and are gaining attention due to

their ability to preserve surface details and volume. Their explicit geometry can be beneficial when coupled to an underlying volumetric discretization of the fluid dynamics, such as for the discretization of surface tension. Thürey et al. [2010] achieve a remarkable level of detail by combining a high resolution surface mesh for surface tracking and local wave simulation with a low resolution grid for coarse-scale motion and incompressibility. Brochu et al. [2010] also employ a surface mesh, using it to guide the placement of additional pressure samples in a Voronoi mesh. Bojsen-Hansen and Wojtan [2013] introduce a surface tracking error measure which compares the normal from the mesh to the pressure gradient from the grid, enabling the use of meshes with much higher resolution than the grid. These works and others that utilize surface meshes, such as those by Schroeder et al. [2012] and Pfaff et al. [2012], all rely on a background grid to carry out the pressure solve. While they benefit from a higher resolution or the more straightforward surface description afforded by the explicit surface mesh, the physical degrees of freedom still reside on the grid and a discretization of the volume remains necessary.

Vortex sheets While both grid- and mesh-based volumetric discretizations grow cubically in complexity when refined, surface meshes grow quadratically. Some researchers have therefore abandoned the background grid altogether and sought to formulate the necessary dynamics entirely on the surface. Brochu et al. [2012] advocate a philosophy of "as much as one sees, that much one should compute," achieving linear time complexity for smoke by using vortex sheets represented by an explicit surface mesh. Da et al. [2015] propose a circulation-preserving surface tension model for soap films also based on vortex sheets. The strength of vortex sheets lies in the Biot-Savart integral which recovers the full 3D velocity field from a surface-only representation; however, baroclinity, which is key to modeling the effects of gravity, is difficult to treat near strong density gradients. Brochu et al. [2012] adopt the Boussinesq approximation, applicable only when the density difference across the surface is small (Atwood ratio approaches 0), while Da et al. [2015] do not consider baroclinity. Neither method is suitable for the simulation of sharp liquid-air interfaces in free surface flow where the density ratio is near-infinite (Atwood ratio approaches 1).

Potential flow The simulation of liquid using potential flow formulations has been explored for various phenomena such as droplet impact [Davidson 2000] and ocean waves [Xue et al. 2001; Keeler and Bridson 2014]. By assuming the velocity field is a gradient field and thus irrotational in addition to incompressible, the velocity field can be compactly encoded by the scalar potential function on the surface whose evolution is governed by the Bernoulli equation. Unfortunately, the nonlinear term in the Bernoulli equation [Davidson 2000] destabilizes the time integration. Xue et al. [2001] rely on smoothing to suppress the instability, while Keeler and Bridson [2014] linearize the Bernoulli equation, sacrificing interesting motions such as breaking waves in return for stability. For the phenomena we consider, both a severe lack of stability and a loss of motion modes are undesirable.

Other non-volumetric techniques Several additional techniques have been proposed to simulate liquid phenomena without paying the price of volumetric discretization. Brochu [2006] proposes an incompressibility enforcement technique based on a harmonic partition of the liquid domain. Zhang et al. [2012] efficiently process the motion of a surface mesh using a collection of deformation operators, including a mean curvature flow operator to approximate surface tension. In these works, the masses associated to the surface degrees of freedom only approximately reflect how much of the underlying liquid volume it influences. Batty et al. [2012] and Zhu

Method	Setting	BI	BEM
[Zhang et al. 2012]	Droplets		
[Brochu et al. 2012]	Smoke	✓	
[Keeler and Bridson 2014]	Waves	✓	✓
[Da et al. 2015]	Bubbles	✓	
Current	3D Liquid	✓	✓

Table 1: *A summary of related surface-based fluid methods. BI indicates boundary integrals; BEM indicates the boundary element method.*

et al. [2014] also utilize triangle meshes, but only to capture thin sheet-like structures, not as a representation of the surface of a liquid volume. For bulk bodies of liquid that are not thin in any direction, Zhu et al. [2014] use a volumetric mesh, effectively reverting to a conforming tetrahedral method.

Boundary integrals and the boundary element method The boundary element method (BEM) has proven to be a powerful technique for a variety of phenomena including elasticity [James and Pai 1999] and brittle fracture [Zhu et al. 2015; Hahn and Wojtan 2015], as well as the potential flow schemes above. We refer the interested readers to Sauter and Schwab [2011] for more details on the theory of boundary integrals and the boundary element method.

Despite the success of BEM in elasticity and fracture simulation, applying BEM to the Navier-Stokes equations results in an integral equation containing a domain integral [Ladyzhenskaya 1963], corresponding to the nonlinear convective term. One proposed treatment is the Dual Reciprocity Method (DRM) [Power and Partridge 1994; Florez and Power 2001], which essentially approximates this term using a collection of radial basis functions. However, several issues of DRM have prevented it from gaining widespread popularity, including lack of convergence [Florez and Power 2001] and ill-conditioned systems [Chen et al. 2003].

We summarize the relationships among the surface-based fluid animation methods most relevant to the current work in Table 1. Note that the vortex sheets method is considered a boundary integral method, as is evident from the Biot-Savart integral.

3 Time Integration

Our model of a 3D liquid volume represents the state—position and velocity—only on the liquid boundary surface. We do not explicitly represent state in the interior.

Throughout our method, we adopt the following assumption (which will be referenced as [HVF] in the following text whenever the derivation invokes it):

DEFINITION **3.1** *The* Harmonic Velocity Field *assumption (HVF): A velocity field* $\mathbf{u} : \mathbb{R}^3 \to \mathbb{R}^3$ *fulfills the HVF assumption if:*

- $\nabla \cdot \mathbf{u} = 0$,
- $\nabla \times \mathbf{u} = 0$,

i.e., it corresponds to a flow that is both incompressible and irrotational.

We integrate forward in time by operating directly on the boundary surface, storing and updating the position and velocity of the surface. We employ operator-splitting [Stam 1999] to solve the incompressible Euler equations over the liquid domain Ω:

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\mathbf{F}, \tag{1}$$

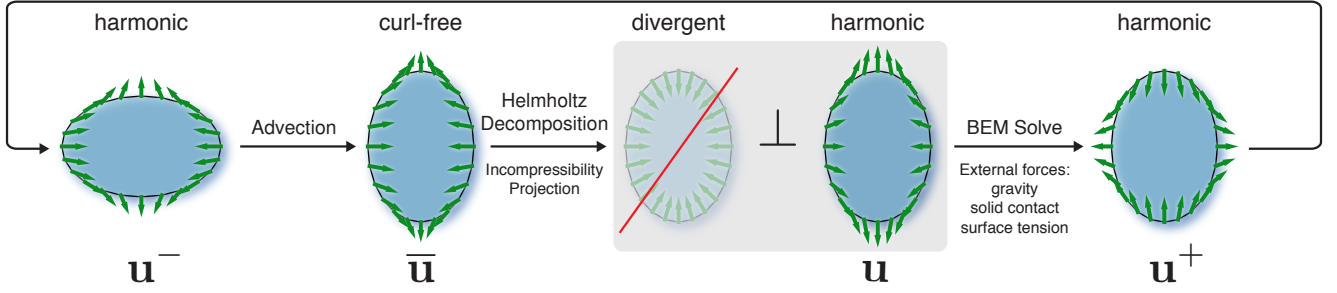$$\nabla \cdot \mathbf{u} = 0. \tag{2}$$

Figure 3: *Overview of a time step. The inset text summarizes the properties on various velocity components.*

We implement this scheme in three steps (Figure 3): advection, projection, and integration of external forces (gravity, surface tension and solid contact). The advection step moves the position of the boundary surface and updates the velocity field absent internal and external forces ($\frac{D\mathbf{u}}{Dt} = 0$). Next, the projection step removes any velocity divergence, $\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho}\nabla p$, introduced by advection. Finally, we integrate the effect of external forces into the velocity field while respecting the solid boundary and incompressibility constraints, $\frac{\partial \mathbf{u}}{\partial t} = \frac{1}{\rho}\mathbf{F}$. We describe each step in detail in the sections below.

3.1 Advection

At the start of time integration, we are given the position of the surface, decorated with a harmonic velocity field. In our Lagrangian approach, advection requires only to update the position of the surface; the velocity associated to each surface point is unaltered. Although the initial velocity field was divergence-free [HVF], the change in surface position can introduce velocity divergence.

3.2 Projection

Next, we remove the divergence, projecting back to the space of harmonic velocity fields. We first describe Helmholtz-based projection for a volumetric velocity field, and then present the transformation to the surface-based representation.

Helmholtz decomposition   As depicted in Figure 3, we denote by $\mathbf{u}^-$ the harmonic 3D velocity field before advection, $\overline{\mathbf{u}}$ the (generally divergent) velocity field after passively moving the velocity field with the liquid, and $\mathbf{u}$ the (incompressible) post-advection, post-projection velocity field.

We can think of divergence as inducing a pressure, p, in a projection that brings $\overline{\mathbf{u}}$ back into the linear subspace of incompressible velocity fields to obtain $\mathbf{u}$. In other words, we can find $\mathbf{u}$ by the orthogonality condition $\mathbf{u} \perp (\overline{\mathbf{u}} - \mathbf{u})$ and the incompressibility condition $\nabla \cdot \mathbf{u} = 0$. For continuous vector fields in $\mathbb{R}^3$ over the liquid domain Ω, orthogonality $\mathbf{v} \perp \mathbf{w}$ is defined through the vanishing of the inner product $\langle \mathbf{v}, \mathbf{w} \rangle_\Omega = \int_\Omega \mathbf{v} \cdot \mathbf{w} dV$.

Unlike most previous approaches [Batty et al. 2007; Clausen et al. 2013; Misztal et al. 2010; Ihmsen et al. 2014a], which use a Poisson solve to perform the projection, we adopt a constructive Helmholtz decomposition [Phillips 1933] on $\overline{\mathbf{u}}$,

$$\overline{\mathbf{u}} = -\nabla\Phi + \nabla \times \mathbf{A} \tag{3}$$

$$= \overbrace{\nabla \times \mathbf{A}_\Gamma - \nabla\Phi_\Gamma}^{\text{harmonic}} + \overbrace{\nabla\Phi_\Omega}^{\text{curl-free}} - \overbrace{\underbrace{\nabla \times \mathbf{A}_\Omega}_{\text{(since } \nabla \times \overline{\mathbf{u}} = 0)}}^{\text{div-free}}, \tag{4}$$

with $\underbrace{\phantom{\nabla \times \mathbf{A}_\Gamma - \nabla\Phi_\Gamma}}_{\mathbf{u}}$ and $\underbrace{}_{\overline{\mathbf{u}} - \mathbf{u}}$

Component	Divergence-free	Curl-free	Zero
$\mathbf{u}^-$	✓	✓	
$\overline{\mathbf{u}}$		✓	
$\mathbf{u}$	✓	✓	
$\mathbf{u}^+$	✓	✓	
$\nabla\Phi_\Gamma$	✓	✓	
$\nabla\Phi_\Omega$		✓	
$\nabla\Phi$		✓	
$\nabla \times \mathbf{A}_\Gamma$	✓	✓	
$\nabla \times \mathbf{A}_\Omega$	✓	✓	✓
$\nabla \times \mathbf{A}$	✓	✓	

Table 2: *A summary of various velocity components.*

where the scalar potential is $\Phi(\mathbf{x}) =$

$$\underbrace{\int_\Gamma \mathbf{n}(\mathbf{y}) \cdot \overline{\mathbf{u}}(\mathbf{y}) G(\mathbf{x},\mathbf{y}) dS_\mathbf{y}}_{\Phi_\Gamma} - \underbrace{\int_\Omega \nabla_\mathbf{y} \cdot \overline{\mathbf{u}}(\mathbf{y}) G(\mathbf{x},\mathbf{y}) dV_\mathbf{y}}_{\Phi_\Omega}, \tag{5}$$

and the vector potential is $\mathbf{A}(\mathbf{x}) =$

$$\underbrace{\int_\Gamma \mathbf{n}(\mathbf{y}) \times \overline{\mathbf{u}}(\mathbf{y}) G(\mathbf{x},\mathbf{y}) dS_\mathbf{y}}_{\mathbf{A}_\Gamma} - \underbrace{\int_\Omega \nabla_\mathbf{y} \times \overline{\mathbf{u}}(\mathbf{y}) G(\mathbf{x},\mathbf{y}) dV_\mathbf{y}}_{\mathbf{A}_\Omega = 0}. \tag{6}$$

Here $\Gamma = \partial\Omega$ is the liquid surface, $\mathbf{n}$ is the outward unit surface normal, and $G(\mathbf{x},\mathbf{y}) = -\frac{1}{4\pi\|\mathbf{x}-\mathbf{y}\|}$ is the free space Green's function of Laplace's equation. These formulas can be proven using the property of the Green's function $\nabla^2 G = \delta_D$, where δ_D is the Dirac delta. In the literature, boundary integrals in the form of Φ_Γ or $\mathbf{A}_\Gamma$ are known as *single layer potentials* [Sauter and Schwab 2011].

Equation 4 decomposes $\overline{\mathbf{u}}$ into three *orthogonal* subspaces (see Appendix): (a) harmonic (simultaneously div- and curl-free); (b) divergent but curl-free; (c) rotational but div-free. Since by assumption $\overline{\mathbf{u}}$ is irrotational, the third term $\nabla \times \mathbf{A}_\Omega$ vanishes. Any divergent component of $\overline{\mathbf{u}}$ *must* be in $\nabla\Phi_\Omega$. These properties on the various velocity components are summarized in Table 2. If we equate $\mathbf{u}$ to the harmonic first component, then the divergent second component is $\overline{\mathbf{u}} - \mathbf{u}$. We have identified the solution to the simultaneous conditions $\nabla \cdot \mathbf{u} = 0$, from (a), and $\mathbf{u} \perp (\overline{\mathbf{u}} - \mathbf{u})$, by the orthogonality of the subspaces.

In summary, we can construct the divergence-free velocity, $\mathbf{u}$, using only the boundary integrals Φ_Γ and $\mathbf{A}_\Gamma$. Note that the Green's function, $G(\mathbf{x},\mathbf{y})$, tends to infinity as the evaluation point, $\mathbf{x}$, approaches the source point, $\mathbf{y}$. The resulting singularities in the integrals require special treatment, which we describe in Section 4.

Surface-only representation   Assume that the liquid domain Ω is contractible. By the Poincaré lemma, a curl-free vector field $\mathbf{u}$ over a contractible domain Ω can be always be represented as

the gradient $\mathbf{u} = \nabla\phi$ of a scalar potential field, ϕ. If $\mathbf{u}$ is also divergence-free, $0 = \nabla \cdot \mathbf{u} = \nabla \cdot \nabla\phi$, then ϕ is harmonic, satisfying Laplace's equation $\nabla^2\phi = 0$.

Since our velocity field is harmonic, i.e., curl- and divergence-free [HVF], it can be exactly represented by the gradient of a harmonic potential.

Taking the boundary restriction of the velocity field yields a volume-conserving surface velocity field, whose normal component $\hat{\mathbf{n}} \cdot \mathbf{u}$ equals the normal derivative $(\hat{\mathbf{n}} \cdot \nabla)\phi$ of the harmonic potential by construction. Therefore, we have a specific instance of Laplace's equation, a boundary value problem (BVP), which can be solved to recover ϕ over Ω, using the surface velocity as the Neumann boundary condition. The gradient $\nabla\phi$ of the solution is unique.

If any two harmonic velocity fields have the same normal component in their boundary restriction, then they will both establish the same boundary data for the BVP, and both yield the same gradient $\nabla\phi$. Therefore, two harmonic velocity fields with the same normal component in their boundary restriction must be the same field. In other words, for a contractible domain, a (3D) harmonic velocity field can be fully described by the normal component of its boundary restriction.

On a non-contractible domain, e.g., in 3D, for a genus $k \neq 0$ liquid body, the normal component of the boundary restriction of a harmonic velocity field still forms a compatible Neumann boundary condition, and the gradient of the BVP's solution is still unique. Furthermore, the velocity field obtained from the BVP is still harmonic by construction. However, the Poincaré lemma no longer holds on such a domain: there exists an "extra" *finite-dimensional vector space of harmonic vector fields* that cannot be expressed as gradients of continuous potentials. These correspond to a flow circulating around a finite number of non-contractible loops, e.g., around the handle of the solid donut (see incident figure). The analysis in our approach simply ignores these "extra" modes.
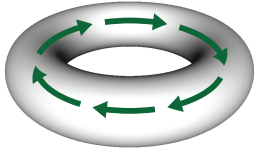
Surface-only projection We are now ready to see that the Helmholtz projection can be directly computed on the surface representation, taking as input and producing as output only surface data. Equation 4 allows us to evaluate the post-projection velocity

$$\mathbf{u} = \nabla \times \mathbf{A}_\Gamma - \nabla\Phi_\Gamma \tag{7}$$

at any location in Ω using only the surface velocity field on Γ, since both Φ_Γ and $\mathbf{A}_\Gamma$ are boundary integrals. The resulting $\mathbf{u}$ is a harmonic velocity field. Therefore, the boundary restriction of (4) allows us to operate directly on surface velocity fields, without representing the interior velocity field. Henceforth, we denote the boundary restriction of $\mathbf{u}$, $\overline{\mathbf{u}}$, etc. by the same symbols.

Tangential velocity Although (7) already gives us the post-projection velocity, we can simplify it further. For the scenarios we consider, air density is negligible compared to liquid density. Imagine that the pressure gradient ∇p has a tangential component: it would effect an infinite tangential air velocity, in turn instantly restoring equilibrium. This argument leads to the familiar free-surface condition $p = 0$, from which it follows that the pressure gradient in (1) is always normal to the surface Γ. We therefore resolve (7) using only the normal component of $\mathbf{u}$, denoted u^n, while copying the tangential component directly from $\overline{\mathbf{u}}$ using the

tangential projection operator $P = I - \mathbf{n}\mathbf{n}^T$:

$$\mathbf{u} = P\overline{\mathbf{u}} + u^n\mathbf{n}, \tag{8}$$
$$u^n = \mathbf{n} \cdot (\nabla \times \mathbf{A}_\Gamma - \nabla\Phi_\Gamma). \tag{9}$$

Apart from being more efficient, the main advantage of using the Helmholtz decomposition only for the normal component, u^n, is that it simplifies the treatment of the $\nabla \times \mathbf{A}_\Gamma$-term in (17)–(19).

3.3 External force integration

In the final stage of time integration, we consider external forces $\mathbf{F}$. The input to this final stage is the post-projection harmonic velocity field, and the output is the end of time step harmonic velocity field [HVF]. In particular, we emphasize that external forces are added in such a way that the velocity field remains incompressible, in contrast with typical volumetric approaches.

Unlike with volumetric discretizations such as a grid or a tetrahedral mesh, in our surface-only representation the mass associated with each degree of freedom cannot be easily computed; when a surface vertex moves it is not immediately clear how much liquid volume moves with it. We therefore circumvent the computation of the nodal masses altogether.

By harmonicity [HVF], the velocity change induced by external forces can be expressed as the gradient of a harmonic pressure field $p_\mathbf{F}$. Therefore, given appropriate boundary conditions, we can specify $p_\mathbf{F}$ as the solution to Laplace's equation.

We model external forces that act (a) normal to the boundary surface, e.g., surface tension, free-slip solid contact, or (b) uniformly throughout the liquid body, e.g., gravity. Both of these categories allow for straightforward modeling of boundary conditions for Laplace's equation.

The boundary consists of the liquid-air interface Γ_A and the liquid-solid interface Γ_S, satisfying $\Gamma_A \cup \Gamma_S = \Gamma$. The surfaces Γ_A and Γ_S intersect only along a curve T which we refer to as the triple junction. Neglecting air pressure, the surface tension force creates a Dirichlet boundary condition at the liquid-air interface. The solid contact force and gravity body force, on the other hand, create a Neumann boundary condition, i.e., a pressure derivative that prevents interpenetration and separation. Consequently, we can write a well-posed boundary value problem to find the pressure due to external forces p_F:

$$\nabla \cdot \nabla p_\mathbf{F} = 0, \quad \text{subject to} \tag{10}$$
$$p_\mathbf{F} = \sigma H, \quad \text{on } \Gamma_A, \tag{11}$$
$$\frac{\partial p_\mathbf{F}}{\partial \mathbf{n}} = \frac{\rho}{\Delta t}(\mathbf{u} + \mathbf{g}\Delta t - \mathbf{u}_{\text{solid}}) \cdot \mathbf{n}, \quad \text{on } \Gamma_S, \tag{12}$$

where σ is the surface tension coefficient, H is the (signed) mean curvature, and $\mathbf{g}$ is the gravitational acceleration. We explicitly integrate acceleration due to gravity into the vertex velocities; the constraint on the solid contact region ensures that these vertices cannot move into the solid.

A vertex is considered to be in contact with the solid only if at least one of its incident faces is in Γ_S. When a droplet peels off a solid surface, the contact area shrinks as the droplet moves away. Finally, Γ_S consists of only one triangle; once it gets too small, remeshing turns this triangle into a single vertex. This last vertex is then *not* constrained any further and can move away from the solid freely.

We solve this system using the Boundary Element Method [Sauter and Schwab 2011], *without the need to discretize the liquid volume*. The BEM solve yields pressure values and normal derivatives. The

resulting pressure gradient accelerates the input velocity field $\mathbf{u}$ to the end-of-step velocity $\mathbf{u}^+$,

$$\mathbf{u}^+ = \mathbf{u} - \frac{\Delta t}{\rho}\nabla p_{\mathbf{F}}. \tag{13}$$

4   Spatial Discretization

Surface mesh   To track the liquid domain boundary Γ, we employ the mesh-based surface tracking library *Los Topos* [Da et al. 2014], taking advantage of its gentle mesh-merging operation, robust collision resolution, and remeshing capabilities. However, in principle any surface tracking technique that maintains a manifold surface mesh can be used. Γ is represented as a triangle mesh (V, F) consisting of vertices V and triangular faces F. Each connected component of liquid volume is represented by a closed manifold triangle mesh. We use $V(i)$ and $F(i)$ to denote the set of vertices neighboring vertex i and the set of faces incident to vertex i, respectively. The area of face i is denoted a_i, while the *vertex area* is defined as $\hat{a}_i = \frac{1}{3}\sum_{j \in F(i)} a_j$. We store a velocity vector $\mathbf{u}_i$ (the subscript distinguishes the discrete quantity $\mathbf{u}_i$ from its continuous counterpart $\mathbf{u}$) on each vertex $i \in V$. A linear basis function θ_i is used to interpolate the velocity at any point on the surface: $\mathbf{u} = \sum_{i \in V} \theta_i \mathbf{u}_i$.

We implement isotropic spatial adaptivity into the *Los Topos* library by allowing different elements to have different minimum and maximum edge length bounds. Each vertex is endowed with a *target edge length*, from which the its minimum and maximum edge lengths are calculated. The remeshing code makes edge collapsing and splitting decisions based on the smaller of the bounds on the two vertices of the edge. Loosely inspired by Narain et al. [2012], we first determine the vertex target edge lengths individually according to the local curvature (as the largest deviation of the dihedral angles from π on incident edges) and the local velocity variation (as the magnitude of the velocity Laplacian). We then reduce the target edge lengths where necessary, in order to ensure that the target edge lengths on neighboring vertex do not differ by more than a constant factor (we use 1.2 for all tests).

The stored surface velocity is linearly interpolated onto new vertices during edge-split operations, and averaged onto the replacement vertex during edge-collapse operations.

Vertex-neighborhood integration   Using the piecewise-linear triangle element greatly simplifies the geometric evolution of the surface as compared to higher order representations, but makes evaluation of boundary integrals tricky. For example, the normal derivative $\mathbf{n} \cdot \nabla \Phi_\Gamma$ has a strongly singular kernel [Sauter and Schwab 2011] and is not well defined at non-smooth points on the surface (such as a vertex) even in the Cauchy principal value sense. Appropriate measures must be taken to manage the order of singularity.

We discretize (9) by integrating it with the basis function θ_i, in a style similar to the Galerkin BEM:

$$\hat{a}_i u_i^n = \int_\Gamma \theta_i u^n \mathrm{d}S = \int_\Gamma \theta_i \left(\mathbf{n} \cdot \nabla \times \mathbf{A}_\Gamma - \mathbf{n} \cdot \nabla \Phi_\Gamma\right) \mathrm{d}S. \tag{14}$$

The integrals can then be evaluated on each face individually, over which the normal $\mathbf{n}$ is a constant:

$$\int_\Gamma \theta_i \mathbf{n} \cdot \nabla \Phi_\Gamma \mathrm{d}S = \sum_{j \in F(i)} \int_j \theta_i \mathbf{n} \cdot \nabla \Phi_\Gamma \mathrm{d}S, \tag{15}$$

$$\int_\Gamma \theta_i \mathbf{n} \cdot \nabla \times \mathbf{A}_\Gamma \mathrm{d}S = \sum_{j \in F(i)} \int_j \theta_i \mathbf{n} \cdot \nabla \times \mathbf{A}_\Gamma \mathrm{d}S. \tag{16}$$

The right hand sides of (15) and (16) are double integrals, since Φ_Γ and $\mathbf{A}_\Gamma$ are boundary integrals themselves. The inner integrals diverge to infinity near vertex i in a weakly singular fashion, thus we discretize the outer integral using a four-point Gaussian quadrature with Duffy transform [Duffy 1982; Keeler and Bridson 2014]; this analytically removes weak singularities.

Next, we describe the discretization of the inner integral. For the Φ_Γ term, the integrand in (15) requires evaluating $\mathbf{n} \cdot \nabla \Phi_\Gamma = \frac{\partial}{\partial \mathbf{n}} \Phi_\Gamma = \int \mathbf{n}(\mathbf{y}) \cdot \overline{\mathbf{u}}(\mathbf{y}) \frac{\partial}{\partial \mathbf{n_x}} G(\mathbf{x}, \mathbf{y}) \mathrm{d}S_{\mathbf{y}}$. Although it is divergent on vertices as discussed above, the integral is not singular when evaluated at a quadrature point in the interior of face j, because the kernel $\frac{\partial}{\partial \mathbf{n_x}} G(\mathbf{x}, \mathbf{y})$ evaluates to zero on face j. This means that within a particular face, this face itself does not contribute to the value of the integral. Therefore, this integral can be discretized by Gaussian quadrature directly. In practice we find using only one quadrature point per face sufficient.

The $\mathbf{A}_\Gamma$ term in (16), on the other hand, is vector valued and needs a different treatment. We first transform it via integration by parts:

$$\int_j \theta_i \mathbf{n} \cdot \nabla \times \mathbf{A}_\Gamma \mathrm{d}S \tag{17}$$

$$= \int_j \mathbf{n} \cdot \nabla \times (\theta_i \mathbf{A}_\Gamma) \mathrm{d}S - \int_j \mathbf{n} \cdot (\nabla \theta_i \times \mathbf{A}_\Gamma) \, \mathrm{d}S \tag{18}$$

$$= \oint_{\partial j} \theta_i \mathbf{A}_\Gamma \cdot \mathbf{t} \mathrm{d}s - \int_j \mathbf{n} \cdot (\nabla \theta_i \times \mathbf{A}_\Gamma) \, \mathrm{d}S, \tag{19}$$

where ∂j is the boundary of triangle j, and $\mathbf{t}$ is its unit tangent. Since the integration by parts moved the derivative from $\mathbf{A}_\Gamma$ to the shape function θ_i, the inner integrals now have a weakly singular kernel instead of a strongly singular one. For a given face $j \in F(j)$, the first term consists of three line integrals, one along each edge. The integrals along the two edges incident to vertex i cancel with those from neighboring faces, while the integral along the third edge is zero because θ_i vanishes there (Figure 4). Therefore, the first term can be dropped, resulting in:

Figure 4: *The loop integral involving* $\mathbf{A}_\Gamma$.

$$\int_j \theta_i \mathbf{n} \cdot \nabla \times \mathbf{A}_\Gamma \mathrm{d}S = - \int_j \mathbf{n} \cdot (\nabla \theta_i \times \mathbf{A}_\Gamma) \, \mathrm{d}S. \tag{20}$$

As with the preceding weakly singular integrals, this too can be discretized directly by a four-point Gaussian quadrature with Duffy transform. Consequently, we now have all the required tools to evaluate the normal component of the velocity field in (14).

Eliminating numerical smoothing   The Galerkin-style outer integral (14) is employed to suppress the singularity, but at the same time it introduces a smoothing of the velocity field which can damp the motion significantly at low mesh resolutions. Fortunately, as we show below, the negative effects of this numerical dissipation can be largely counteracted.

For any discrete field ϕ_i on vertices, whose interpolated continuous field is $\phi = \sum_{i \in V} \theta_i \phi_i$, discretizing it back to vertices again via an integral in the form of (14) will result in a discrete field $\tilde{\phi}_i$ that
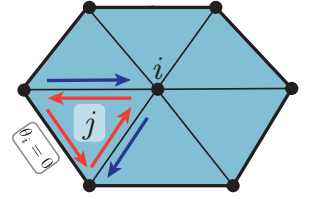
appears smoothed compared to ϕ_i:

$$\tilde{\phi}_i = \frac{1}{\hat{a}_i} \int_\Gamma \theta_i \phi \mathrm{d}S = \frac{1}{\hat{a}_i} \int_\Gamma \theta_i \sum_{j \in V} \theta_j \phi_j \mathrm{d}S \qquad (21)$$

$$= \frac{1}{\hat{a}_i} \phi_i \int_\Gamma \theta_i^2 \mathrm{d}S + \frac{1}{\hat{a}_i} \sum_{i \neq j} \phi_j \int_\Gamma \theta_i \theta_j \mathrm{d}S \qquad (22)$$

$$= \frac{1}{2} \phi_i + \frac{1}{2} \sum_{j \in V(i)} \omega_j \phi_j, \text{ where} \qquad (23)$$

$$\omega_j = \frac{\sum\limits_{k \in F(i) \cap F(j)} a_k}{2 \sum\limits_{k \in F(i)} a_k}. \qquad (24)$$

The weights ω_j can be computed by evaluating the integrals over the linear basis functions θ on triangles containing node i and collecting the resulting terms. It is easy to verify that these weights form a partition of unity: each term in the sum in the nomiator adds the areas of the two triangles containing edge (i, j), the result is scaled by twice the area of the neighborhood of node i, as illustrated in Figure 4. Guided by this analysis, we apply Laplacian smoothing after evaluating the velocity field by (14) and before the pressure solve, with the negative coefficient $-\frac{1}{2}$ and the same weights as above, which effectively sharpens the velocity field just enough to remove the numerical smoothing. In practice we use a slightly smaller coefficient (e.g. -0.4 or -0.45) to retain a small amount of smoothing, so that the explicit time integration is stable.

Boundary Element solve   In order to solve (10) for pressure induced by external forces, we use a direct boundary integral equation formulation [Atkinson 1997], due to the mixed nature of our boundary condition:

$$\Theta(\mathbf{x}) p_{\mathbf{F}}(\mathbf{x}) = \int_\Gamma \left(p_{\mathbf{F}}(\mathbf{y}) \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n_y}} - \frac{\partial p_{\mathbf{F}}(\mathbf{y})}{\partial \mathbf{n_y}} G(\mathbf{x}, \mathbf{y}) \right) \mathrm{d}S_{\mathbf{y}}, \qquad (25)$$

where $\Theta(\mathbf{x})$ is the internal solid angle of the liquid domain Ω at $\mathbf{x} \in \Gamma$, discretized according to Mantic [1993] (below, we will drop the subscript from $p_{\mathbf{F}}$ in derivatives for legibility). We then use a vertex-collocated linear BEM formulation as in Keeler and Bridson [2014].

Special care must be taken at the triple junctions, where the surface Γ is expected to have a sharp crease. Such geometric non-smoothness often plagues collocated BEM solvers, as quantities like $\frac{\partial p}{\partial \mathbf{n}}$ may take on different values on either side of the crease, creating a discontinuity. We use a simplified version of the *double nodes* technique [Hartmann 2012], taking advantage of the fact that the pressure p is always continuous, and that the $\frac{\partial p}{\partial \mathbf{n}}$ discontinuity only occurs at the triple junctions. The local geometry around a triple-junction vertex can be seen as the union of the liquid-air interface (part of Γ_A) and the liquid-solid interface (part of Γ_S), each a smooth patch of surface isomorphic to a half-plane. Therefore, each vertex on a triple junction holds two $\frac{\partial p}{\partial \mathbf{n}}$ values, one for the liquid-air interface, which is an unknown, and the other for the liquid-solid interface, which is given by the Neumann boundary condition in (12). The pressure p, on the other hand, is not duplicated, and its value on the triple junction is given by the Dirichlet boundary condition in (11). This guarantees that there is still one unknown and one collocation equation on each triple junction vertex, as for all other vertices, and thus the resulting linear system is not over- or underdetermined.

Pressure and discrete mean curvature   Equation 11 requires the evaluation of mean curvature H at a vertex. We discretize it as the (signed) scalar mean curvature integral over the vertex local neighborhood (the *discrete mean curvature measure*) [Cohen-Steiner and Morvan 2003]. The integral mean curvature is then divided by the vertex area $\hat{a}_i$ to obtain the pointwise mean curvature H in (11). This discretization scheme has been found to work well for implementing surface tension as a pressure jump condition in the Lagrangian setting [Da et al. 2015].

Again, the triple junctions must be treated differently, as one of the principal curvatures of the liquid surface is essentially infinite, making the mean curvature not well-defined. Examining the force balance at a triple junction point (Figure 5), we find the pressure induced by the pairwise surface tension forces between
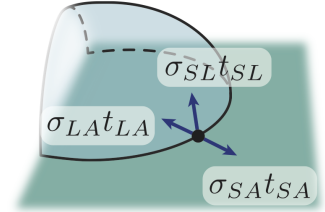
Figure 5: *Surface tension force balance at the triple junction.*

the three phases as an integral over a strip of infinitesimal width covering both sides of the triple junction,

$$\int p_{\text{phases}} \mathrm{d}s = (\sigma_{LA} \mathbf{t}_{LA} + \sigma_{SA} \mathbf{t}_{SA} + \sigma_{SL} \mathbf{t}_{SL}) \cdot \mathbf{t}_{SA}, \qquad (26)$$

where subscripts LA, SA and SL stand for liquid-air, solid-air and solid-liquid respectively, and the $\mathbf{t}$ vectors with various subscripts are the unit tangent vectors in the corresponding interface, pointing away from and orthogonal to the triple junction. The width of this strip over which the surface tension force between three phases is exerted is a numerical parameter, chosen so that the motion of non-equilibrium triple junction configurations can be properly resolved by the explicit time integration, while making the equilibrium configuration sufficiently accurate. We find a value of 0.25 times the average edge length works well.

Velocity update   However, the BEM solve for $p_{\mathbf{F}}$ yields pressure values and normal derivatives, not the gradients required by the update (13). For smooth regions of the surface, we interpolate the vertex pressure linearly per face to obtain a piecewise-constant tangential pressure gradient, and average them from faces to vertices. This tangential gradient is then combined with the normal pressure derivative $\frac{\partial p}{\partial \mathbf{n}}$ from the BEM solve to produce the full pressure gradient. On triple junctions, across which the surface normal and tangent vectors are different, we solve the following overdetermined system via its normal equations for each triple junction vertex i, essentially least-square fitting the various directional derivatives known to us:

$$\begin{pmatrix} \mathbf{n}_S^T \\ \mathbf{n}_A^T \\ \mathbf{t}_{SL}^T \\ \mathbf{t}_{LA}^T \\ \mathbf{t}_{TP}^T \end{pmatrix} \nabla p = \begin{pmatrix} \frac{\partial p}{\partial \mathbf{n}_S} \\ \frac{\partial p}{\partial \mathbf{n}_A} \\ \frac{\partial p}{\partial \mathbf{t}_{SL}} \\ \frac{\partial p}{\partial \mathbf{t}_{LA}} \\ \frac{\partial p}{\partial \mathbf{t}_{TP}} \end{pmatrix}, \qquad (27)$$

where $\mathbf{t}_{TP}$ is the unit tangent of the triple junction curve, and the partial derivatives on the right hand side are obtained either from the BEM solve (for normal derivatives) or from tangentially differentiating the pressure field (for tangential derivatives) as before. The first four equations of this overdetermined system establish the pressure gradient in the plane orthogonal to the triple junction. Although redundant, only together can they span this plane reliably regardless of
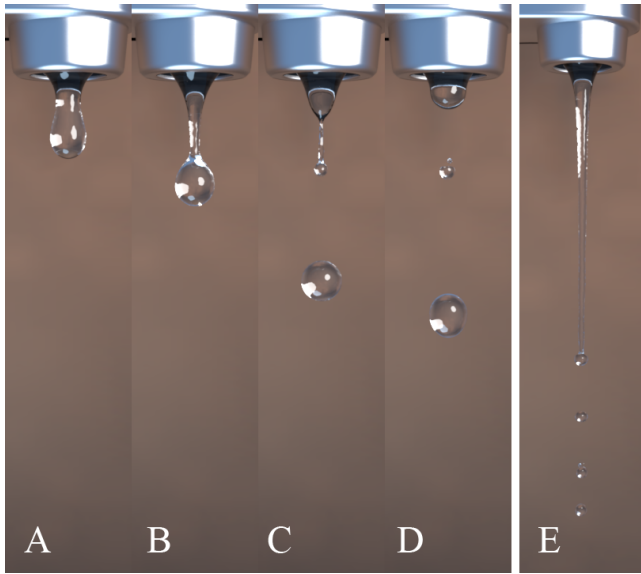
Figure 6: *Dripping water. (A) – (D) A droplet pinches off near the tap at a low flow rate. (E) An elongated stream is created before droplet formation at a high flow rate.*

of the contact angle at the triple junction; removing any of them makes the system prone to ill-conditioning.

5 Results

This section presents the results of some simulation experiments using the proposed method. The purpose of these experiments is to explore what phenomena the method can produce, as well as to investigate how restrictive the HVF assumption is and how expressive the harmonic velocity fields can be.

Dripping The familiar phenomenon of water dripping from a sink tap is driven by surface tension in a process known as the Rayleigh-Plateau instability (Figure 6). Our method successfully captures the entire process from the thinning of the neck to the pinch-off of the droplet.

Water jets collision Bush and Hasha [2004] systematically studied the flow patterns resulting from laminar jets colliding at an angle, identifying several regimes in a phase diagram parameterized by the jet flow rate, surface tension coefficient, nozzle radius, etc. In Figure 1, the most representative patterns are reproduced. Lower flow rates and higher surface tension coefficients cause the liquid jets shooting away from the original collision to bend back and meet again for a second collision (and subsequently, a third, and so on). Each collision flips the orientation of the liquid sheet plane by $90°$, forming a so-called "fluid chain" structure (Figure 1 A). At higher flow rates, the liquid sheet oscillates in the normal direction until it disintegrates (Figure 1 B) or even ruptures violently (Figure 1 C), as a result of the liquid sheet flapping instability (highlighted in Figure 7).

Splash on a hydrophobic surface The behavior of a liquid droplet on a solid surface is greatly affected by the relative strength of the surface tension at the liquid-solid interface and at the air-solid interface. Figure 8 shows the splash created by the impact of a water droplet on a hydrophobic solid surface.
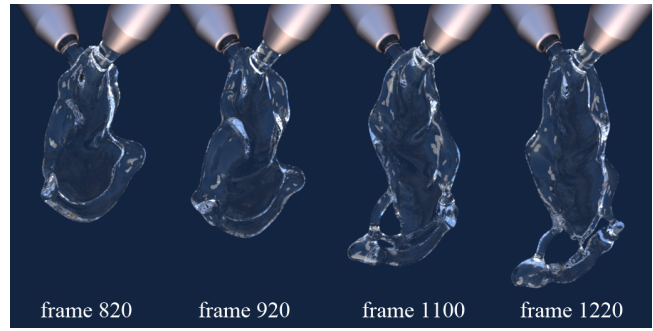

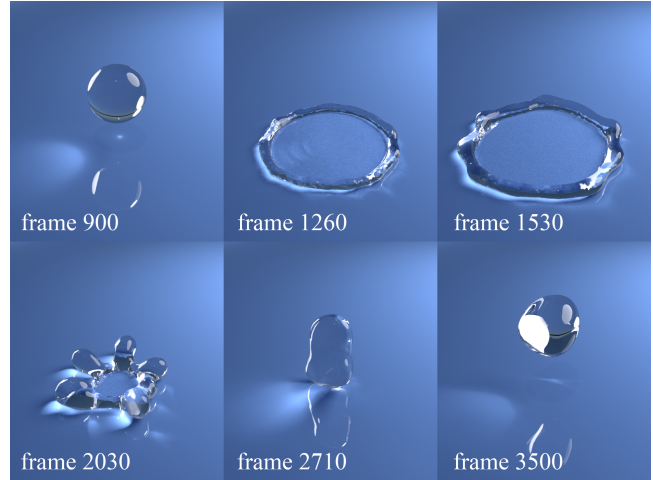**Figure 7:** *Sheet flapping instability following a jet collision.*

Figure 8: *A water droplet impacting, splashing on, and rebounding off a hydrophobic surface.*

Crown splash When a droplet impacts on a water surface, the resulting splash takes a distinctive "crown" shape as a result of the Rayleigh-Plateau instability (Figure 9). The top of the crown usually atomizes into many small droplets shooting off in different directions. Level set based surface tracking methods are known to lose these small droplets quickly. Our method was able to resolve and maintain these droplets, recreating the splashes at a previously unseen level of detail.

Droplet collision Figure 10 shows the splash patterns resulting from the head-on collision of two droplets in zero gravity. The extent of the splash and the atomization of the sheet is dictated by the relative strength of surface tension and liquid inertia, which is described by the dimensionless Weber number $We = \rho v^2 l/\sigma$. Our results qualitatively agree with the experimental observations reported by Pan et al. [2009], successfully reproducing the "fingering and separation" regime and the "breakup" regime (Figure 11) in the correct parameter ranges.

In applications like ink-jet printing, off-center collisions are of more practical interest since they are statistically more likely. Quan et al. [2009] studied the intermediate configurations during the coalescence and subsequent separation of two droplets using an unstructured tetrahedral mesh solver. Figure 2 shows a side-by-side comparison of our simulation and the results reported by Quan et al. [2009] in Figure 16.

Test	Mean verts	Mean faces	Max verts	Max faces	Mean frame time (s)	Max frame time (s)
Dripping (fixed flow rate)	2143	4275	2706	5396	3.311	4.675
Dripping (varying flow rate)	3600	7183	5069	10130	4.296	9.235
Droplet impact (no gravity)	2384	4765	8829	17654	7.482	98.96
Droplet impact (with gravity)	1898	3791	8931	17852	6.465	108.99
Jet collision - fluid chains	7883	15763	11920	23836	64.73	176.8
Jet collision - disintegrating sheets	18195	36356	26733	53422	374.4	649.7
Jet collision - violent flapping	19992	39455	34445	68806	593.1	757.6
Crown splash (low surface tension, σ)	6547	13022	17102	33890	25.83	97.91
Crown splash (high σ)	5582	11131	15755	31358	14.56	73.62
Droplet collision (low σ)	7240	14331	18860	37718	47.08	530.2
Droplet collision (medium σ)	3729	7428	11952	23908	23.56	189.0
Droplet collision (high σ)	1862	3719	8157	16314	13.58	91.00

Table 3: *Simulation run time and mesh complexity for various examples.*

Figure 9: *Crown splash.*

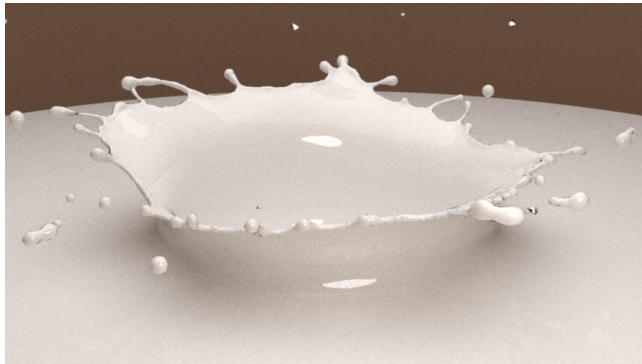**Figure 10:** *Collision of two droplets, with various surface tension coefficients.*

Performance Table 3 summarizes the average and peak mesh complexity over these simulations, and the corresponding run time for one time step (all timing collected on a Core i7 3.4GHz CPU running a single thread). The most expensive step is typically the projection step due to the required double integral, followed by the BEM solve. Besides the potential for fast summation techniques to speed up these two steps (see discussion in Section 6), the projection step is also easily parallelizable, as is the BEM solve if an iterative solver is used. Some of the tests have greatly varying mesh sizes; for example, the droplet collision tests become highly complex when the splash is extensive, but the complexity falls off rapidly as soon as individual droplets are formed.

6 Discussion

Our method represents the first surface-only numerical scheme practical for simulation of liquid phenomena involving large, splashy motions, and therefore opens up a number of new questions and future opportunities.

Our formulation—both the projection step and the boundary element solve—assumes that the liquid domain is irrotational on the interior. This excludes certain motions like swirling and vortices. However, for the kinds of external forces and boundary contact conditions we consider, vorticity cannot originate spontaneously in the interior of an initially-irrotational fluid of uniform density, therefore we have no risk of creating erroneous behavior, if the initial state is irrotational. On the other hand, a no-slip boundary contact condition, such as that needed for modeling the *rolling* of a droplet down an inclined plane, *can* induce rotational motion, and is therefore outside the scope of our current work. It would be interesting to superpose additional rotational degrees of freedom, such as vortex particles [Selle et al. 2005; Park and Kim 2005], vortex rings [Angelidis and Neyret 2005;
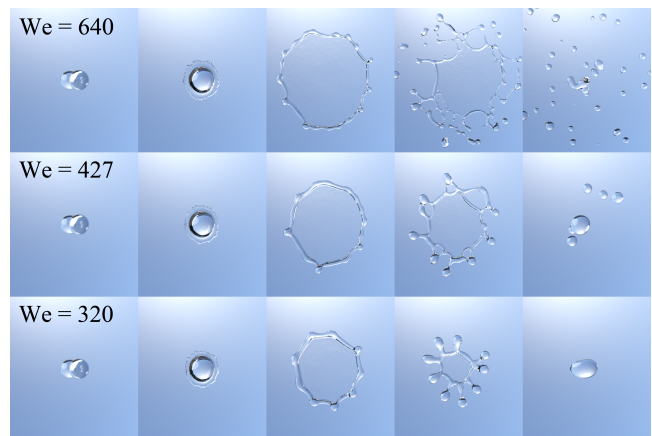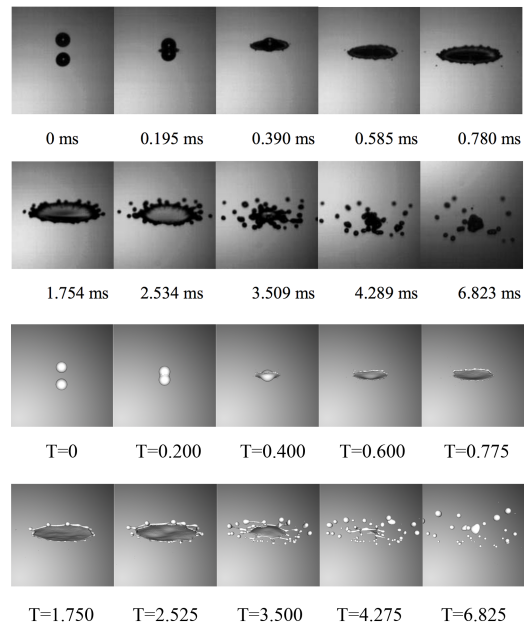

Figure 11: *Droplet collision simulated by our technique (bottom) compared to physical experiment photos (top) by Pan et al. [2009] in Figure 3 (c).*

Weißmann and Pinkall 2010], and vortex sheets [Pfaff et al. 2012; Brochu et al. 2012].

Our formulation also ignores the finite number of harmonic vector fields that cannot be represented as the gradient of harmonic potentials. It would be interesting to study in detail whether accurately tracking the modes can reproduce a broader range of liquid phenomena. As a preliminary exploration, we initialize a solid-donut shaped liquid body with a rotational velocity (see Figure 12, and the accompanying video in the additional supplemental materials). Such a velocity cannot be represented by the gradient of a harmonic potential; note, however, that it can be represented by our surface-based vector field. We observe that the donut spins freely before the Rayleigh-Plateau instability sets in and atomizes the domain. If we consider a stronger surface tension coefficient, the donut collapses into a droplet, which also continues to spin. As stated previously, for the forces that we consider, Euler's equations specify that the modes in the kernel (such as the rotational mode here) should be conserved over time; hence at least in this scenario, qualitatively the correct behavior is observed.
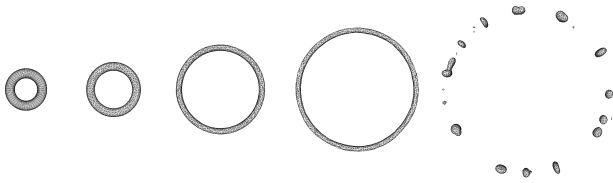

Figure 12: *Donut-shaped liquid body with a rotational velocity.*

Although not required by our solver, it may be desirable to evaluate the velocity at certain locations in the interior of the liquid domain in some situations, such as for one-way solid coupling (an example being entrained bubbles). This information is not directly available on our surface-only representation, but can be obtained by solving for a potential flow using the surface normal velocity as the Neumann boundary condition, as described in Section 3.2 (the "Surface-only representation" paragraph).

Our surface-based approach has the potential to scale favorably with scene resolution compared to volumetric-based approaches. Although we currently implement the boundary integrals via direct summation, there are a number of fast summation techniques available (such as the Fast Multipole Method [Greengard and Rokhlin 1987] and Particle-Particle Particle-Mesh method [Zhang and Bridson 2014], to name just two) to accelerate this process to practically linear complexity. Similarly, the boundary element solve can be performed by an iterative solver with the matrix-vector multiplication accelerated by the same technique. The resulting numerical method would scale linearly with the number of elements in the surface mesh, which in turn scales quadratically with the inverse of the spatial resolution.

In our numerical experiments with asymmetrically colliding jets, we also attempted to reproduce the parameter regime corresponding to the "fishbone" observed by Bush and Hasha [2004]. However our simulation slowed down to ten minutes per frame as we approached the formation of the fishbone, due to the number of triangle mesh elements and the quadratic scaling of the boundary integral quadrature. This suggests that the fishbone is an ideal testbed for these fast summation methods. A 3D

grid, on the other hand, would scale cubically, due to its volumetric nature. As the space of potential geometric features also scales quadratically with the inverse resolution, our method, combined with fast summation techniques, would achieve an asymptotically optimal complexity.

Our Helmholtz decomposition-based incompressibility projection requires evaluation of two boundary integrals, and does not require a linear solve. It could be adapted for other applications, such as volume-conserving elastic deformation to remove divergence, or in other surface-only models like that of Zhang et al. [2012] as an alternative to local volume correction.

Acknowledgments

References

ABRAHAM, R., MARSDEN, J. E., AND RATIU, T. 1988. *Manifolds, Tensor Analysis, and Applications*. Springer.

ANGELIDIS, A., AND NEYRET, F. 2005. Simulation of smoke based on vortex filament primitives. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 87 – 96.

ATKINSON, K. E. 1997. The numerical solution of boundary integral equations. In *The State of the Art in Numerical Analysis, pp. 223–259.*

BATTY, C., BERTAILS, F., AND BRIDSON, R. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. 26*, 3, 100.

BATTY, C., XENOS, S., AND HOUSTON, B. 2010. Tetrahedral embedded boundary methods for accurate and flexible adaptive fluids. In *Computer Graphics Forum*, vol. 29, Wiley Online Library, 695–704.

BATTY, C., URIBE, A., AUDOLY, B., AND GRINSPUN, E. 2012. Discrete viscous sheets. *ACM Trans. Graph. 31*, 4, 113.

BOJSEN-HANSEN, M., AND WOJTAN, C. 2013. Liquid surface tracking with error compensation. *ACM Trans. Graph. 32*, 4, 68.

BROCHU, T., BATTY, C., AND BRIDSON, R. 2010. Matching fluid simulation elements to surface geometry and topology. *ACM Trans. Graph. 29*, 4, 47.

BROCHU, T., KEELER, T., AND BRIDSON, R. 2012. Linear-time smoke animation with vortex sheet meshes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 87–95.

BROCHU, T. 2006. *Fluid animation with explicit surface meshes and boundary-only dynamics*. Master's thesis, University of British Columbia.

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, C., GOLBERG, M., AND SCHABACK, R. 2003. Recent developments in the dual reciprocity method using compactly supported radial basis functions. *Transformation of Domain Effects to the Boundary (YF Rashed and CA Brebbia, eds), WITPress, Southampton, Boston*, 138–225.

CHENTANEZ, N., FELDMAN, B. E., LABELLE, F., O'BRIEN, J. F., AND SHEWCHUK, J. R. 2007. Liquid simulation on lattice-based tetrahedral meshes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer animation*, 219–228.

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, 17.

COHEN-STEINER, D., AND MORVAN, J.-M. 2003. Restricted Delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual Symposium on Computational Geometry*, ACM, 312–321.

DA, F., BATTY, C., AND GRINSPUN, E. 2014. Multimaterial mesh-based surface tracking. *ACM Trans. Graph. 33*, 4, 112:1–112:11.

DA, F., BATTY, C., WOJTAN, C., AND GRINSPUN, E. 2015. Double bubbles sans toil and trouble: discrete circulation-preserving vortex sheets for soap films and foams. *ACM Trans. Graph. 34*, 4, 149.

DAVIDSON, M. R. 2000. Boundary integral prediction of the spreading of an inviscid drop impacting on a solid surface. *Chemical Engineering Science 55*, 6, 1159–1170.

DUFFY, M. G. 1982. Quadrature over a pyramid or cube of integrands with a singularity at a vertex. *SIAM journal on Numerical Analysis 19*, 6, 1260–1262.

ENRIGHT, D., FEDKIW, R., FERZIGER, J., AND MITCHELL, I. 2002. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics 183*, 1, 83–116.

FEDKIW, R. P., ASLAM, T., MERRIMAN, B., AND OSHER, S. 1999. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics 152*, 2, 457–492.

FLOREZ, W., AND POWER, H. 2001. Comparison between continuous and discontinuous boundary elements in the multidomain dual reciprocity method for the solution of the two-dimensional Navier–Stokes equations. *Engineering analysis with boundary elements 25*, 1, 57–69.

FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In *Proceedings of SIGGRAPH 01, Annual Conference Series*, 23–30.

FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. *Graphical models and image processing 58*, 5, 471–483.

GREENGARD, L., AND ROKHLIN, V. 1987. A fast algorithm for particle simulations. *Journal of Computational Physics 73*, 2, 325–348.

HAHN, D., AND WOJTAN, C. 2015. High-resolution brittle fracture simulation with boundary elements. *ACM Trans. Graph. 34*, 4, 151.

HARTMANN, F. 2012. *Introduction to boundary elements: theory and applications*. Springer Science & Business Media.

HONG, J.-M., AND KIM, C.-H. 2005. Discontinuous fluids. In *ACM Trans. Graph.*, vol. 24, ACM, 915–920.

IHMSEN, M., CORNELIS, J., SOLENTHALER, B., HORVATH, C., AND TESCHNER, M. 2014. Implicit incompressible SPH. *Visualization and Computer Graphics, IEEE Transactions on 20*, 3, 426–435.

IHMSEN, M., ORTHMANN, J., SOLENTHALER, B., KOLB, A., AND TESCHNER, M. 2014. SPH Fluids in Computer Graphics. *Eurographics 2014 - State of the Art Reports*.

JAMES, D. L., AND PAI, D. K. 1999. ArtDefo: accurate real time deformable objects. In *Proceedings of SIGGRAPH 99, Annual Conference Series*, 65–72.

KEELER, T., AND BRIDSON, R. 2014. Ocean waves animation using boundary integral equations and explicit mesh tracking. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 11–19.

LADYZHENSKAYA, O. A. 1963. *The mathematical theory of viscous incompressible flow*. Gordon and Breach New York.

MANTIČ, V. 1993. A new formula for the C-matrix in the Somigliana identity. *Journal of Elasticity 33*, 3, 191–201.

MISZTAL, M. K., BRIDSON, R., ERLEBEN, K., BÆRENTZEN, J. A., AND ANTON, F. 2010. Optimization-based fluid simulation on unstructured meshes. In *VRIPHYS*, 11–20.

NARAIN, R., SAMII, A., AND O'BRIEN, J. F. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph. 31*, 6, 152.

OSHER, S., AND SETHIAN, J. A. 1988. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics 79*, 1, 12–49.

PAN, K.-L., CHOU, P.-C., AND TSENG, Y.-J. 2009. Binary droplet collision at high Weber number. *Phys. Rev. E 80* (Sep), 036301.

PARK, S. I., AND KIM, M. J. 2005. Vortex fluid for gaseous phenomena. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 261–270.

PFAFF, T., THUEREY, N., AND GROSS, M. 2012. Lagrangian vortex sheets for animating fluids. *ACM Trans. Graph. 31*, 4, 112.

PHILLIPS, H. B. 1933. *Vector analysis*. Wiley New York.

POWER, H., AND PARTRIDGE, P. W. 1994. The use of Stokes' fundamental solution for the boundary only element formulation of the three-dimensional Navier-Stokes equations for moderate Reynolds numbers. *International journal for numerical methods in engineering 37*, 11, 1825–1840.

QUAN, S., LOU, J., AND SCHMIDT, D. P. 2009. Modeling merging and breakup in the moving mesh interface tracking method for multiphase flow simulations. *Journal of Computational Physics 228*, 7, 2660–2675.

SAUTER, S. A., AND SCHWAB, C. 2011. *Boundary element methods*. Springer.

SCHROEDER, C., ZHENG, W., AND FEDKIW, R. 2012. Semi-implicit surface tension formulation with a Lagrangian surface mesh on an Eulerian simulation grid. *Journal of Computational Physics 231*, 4, 2092–2115.

SELLE, A., RASMUSSEN, N., AND FEDKIW, R. 2005. A vortex particle method for smoke, water and explosions. In *ACM Trans. Graph.*, vol. 24, ACM, 910–914.

STAM, J. 1999. Stable fluids. In *Proceedings of SIGGRAPH 99, Annual Conference Series*, ACM Press/Addison-Wesley Publishing Co., 121–128.

STOCK, M. J. 2006. *A regularized inviscid vortex sheet method for three dimensional flows with density interfaces*. PhD thesis, Worcester Polytechnic Institute.

THÜREY, N., WOJTAN, C., GROSS, M., AND TURK, G. 2010. A multiscale approach to mesh-based surface tension flows. In *ACM Trans. Graph.*, vol. 29, ACM, 48.

WANG, H., MUCHA, P. J., AND TURK, G. 2005. Water drops on surfaces. In *ACM Trans. Graph.*, vol. 24, ACM, 921–929.

WEISSMANN, S., AND PINKALL, U. 2010. Filament-based smoke with vortex shedding and variational reconnection. *ACM Trans. Graph. 29*, 4, 115.

WOJTAN, C., MÜLLER-FISCHER, M., AND BROCHU, T. 2011. Liquid simulation with mesh-based surface tracking. In *ACM SIGGRAPH 2011 Courses*, ACM, 8.

XUE, M., XÜ, H., LIU, Y., AND YUE, D. K. 2001. Computations of fully nonlinear three-dimensional wave–wave and wave–body interactions. Part 1. Dynamics of steep three-dimensional waves. *Journal of Fluid Mechanics 438*, 11–39.

ZHANG, X., AND BRIDSON, R. 2014. A PPPM fast summation method for fluids and beyond. *ACM Trans. Graph. 33*, 6, 206.

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.

ZHANG, X., BRIDSON, R., AND GREIF, C. 2015. Restoring the missing vorticity in advection-projection fluid solvers. *ACM Trans. Graph. 34*, 4.

ZHU, B., QUIGLEY, E., CONG, M., SOLOMON, J., AND FEDKIW, R. 2014. Codimensional surface tension flow on simplicial complexes. *ACM Trans. Graph. 33*, 4, 111.

ZHU, Y., BRIDSON, R., AND GREIF, C. 2015. Simulating rigid body fracture with surface meshes. *ACM Trans. Graph. 34*, 4, 150.

Appendix: Orthogonality of Helmholtz Decomposition

We dissect (3) from the perspective of the Hodge decomposition [Abraham et al. 1988], which decomposes a differential k-form ω into the sum of an exact component $d\alpha$, a co-exact component $\delta\beta$, and a harmonic (closed and co-closed) component γ, while guaranteeing the components are mutually orthogonal provided certain boundary conditions are met. Applying it to our scenario, we decompose the 1-form $\overline{\mathbf{u}}^\flat$, in which case d and δ correspond to gradient and curl respectively. Based on the observation on the divergence and curl of each term above, we find that $(\nabla\Phi_\Omega)^\flat = d\Phi_\Omega$ is the exact component, $(-\nabla\times\mathbf{A}_\Omega)^\flat = -\delta(*\mathbf{A}_\Omega^\flat)$ is the co-exact component, and the two boundary integral terms $(-\nabla\Phi_\Gamma+\nabla\times\mathbf{A}_\Gamma)^\flat$ together become the harmonic component. For this scenario, Abraham et al. [1988] prove in Theorem 8.5.5 that the decomposition is indeed orthogonal, and in particular, the exact component is orthogonal to
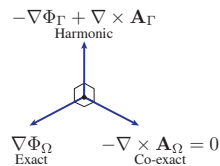
the harmonic component (summarized in the incident figure). Converting back to vector fields, this means that the velocity constructed from the two boundary integral terms,

$$\mathbf{u} = -\nabla\Phi_\Gamma + \nabla\times\mathbf{A}_\Gamma, \tag{28}$$

satisfies

$$\nabla\cdot\mathbf{u} = 0, \tag{29}$$
$$(\mathbf{u}-\overline{\mathbf{u}}) \perp \mathbf{u}, \tag{30}$$

and we have found the desired post-projection velocity $\mathbf{u}$.