

## Simulating liquid–gas interfaces and moving contact lines with the immersed boundary method

Cite as: Phys. Fluids **34**, 053323 (2022); <https://doi.org/10.1063/5.0086452>

Submitted: 25 January 2022 • Accepted: 26 April 2022 • Published Online: 26 May 2022

Michael Y. Li (李岳),  Daniel Chin (秦楠枫),  Charles Puelz, et al.



[View Online](#)



[Export Citation](#)



[CrossMark](#)

### APL Machine Learning

[LEARN MORE](#)

Open, quality research for the networking communities

MEET OUR NEW EDITOR-IN-CHIEF



# Simulating liquid–gas interfaces and moving contact lines with the immersed boundary method

Cite as: Phys. Fluids 34, 053323 (2022); doi: 10.1063/5.0086452

Submitted: 25 January 2022 · Accepted: 26 April 2022 ·

Published Online: 26 May 2022



View Online



Export Citation



CrossMark

Michael Y. Li (李岳),<sup>1</sup> Daniel Chin (秦楠枫),<sup>2</sup> Charles Puelz,<sup>3</sup> and Pejman Sanaei<sup>4,5,a)</sup>

## AFFILIATIONS

<sup>1</sup>Courant Institute of Mathematical Sciences, New York University, New York, New York 10012-1110, USA

<sup>2</sup>New York University Shanghai, Shanghai, 200120, China

<sup>3</sup>Department of Pediatrics, Section of Cardiology, Texas Children's Hospital and Baylor College of Medicine, Houston, Texas 77030-2372, USA

<sup>4</sup>Department of Mathematics, New York Institute of Technology, New York, New York 10023-7692, USA

<sup>5</sup>Department of Mathematics and Statistics, Georgia State University, Atlanta, Georgia 30303-3965, USA

<sup>a)</sup>Author to whom correspondence should be addressed: psanaei@nyit.edu. URL: <https://sites.google.com/nyit.edu/pejman-sanaei-webpage>

## ABSTRACT

In this work, we use the immersed boundary method with four extensions to simulate a moving liquid–gas interface on a solid surface. We first define a moving contact line model and implements a static-dynamic friction condition at the immersed solid boundary. The dynamic contact angle is endogenous instead of prescribed, and the solid boundary can be non-stationary with respect to time. Second, we simulate both a surface tension force and a Young's force with one general equation that does not involve estimating local curvature. In the third extension, we splice liquid–gas interfaces to handle topological changes, such as the coalescence and separation of liquid droplets or gas bubbles. Finally, we re-sample liquid–gas interface markers to ensure a near-uniform distribution without exerting artificial forces. We demonstrate empirical convergence of our methods on non-trivial examples and apply them to several benchmark cases, including a slipping droplet on a wall and a rising bubble.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0086452>

## I. INTRODUCTION

Physical systems involving the coupling between a fluid and evolving immersed structures are usually impossible to describe analytically. The immersed boundary (IB) method is a numerical approach for solving such problems. Traditionally, this method is used with the assumption that the immersed boundaries are massless.<sup>1</sup> The method describes the fluid in Eulerian coordinates and the immersed boundaries with arrays of linked Lagrangian markers. The fluid advects the markers and the markers exert forces onto the fluid. The massless-boundary assumption is suitable for describing thin elastic membranes common in biological applications, for example, the interaction between blood flow and heart valves.<sup>1</sup> Similarly, the massless-boundary assumption is appropriate for liquid–gas interfaces. Thus, with a surface tension model, the IB method is capable of modeling multi-phase fluid flow.<sup>2–5</sup> This introduces a new challenge, which is the moving contact line (MCL) problem that emerges when a solid

boundary meets with a liquid–gas interface. To avoid the force singularity at the contact line, Lai *et al.*<sup>6</sup> proposed an MCL model that simulates Navier slip with the IB method, but it is limited to fixed solid boundaries.

In this work, we describe a static-dynamic friction condition between a moving liquid–gas interface and an evolving immersed boundary. Two types of immersed boundaries (liquid–gas interface, solid surface) coexist and are governed by the same IB numerical method. The slip condition used in this paper is informed by recent molecular dynamics (MD) simulations.<sup>7</sup> The resulting numerical scheme, unlike related works that either prescribe the contact angle<sup>8,9</sup> or the slip velocity,<sup>6,10</sup> is capable of reaching the dynamic contact angle in an endogenous way, i.e., the dynamic contact angle is the result of droplet size, wall friction, gravity, and other local stresses.

In addition to the MCL model, we propose a surface tension model within the IB method. Using the IB method to simulate surface

tension is often classified as a front-tracking scheme. Most front-tracking methods, including many IB variants, calculate the magnitude of surface tension by estimating the local curvature of the interface using three to six markers.<sup>25</sup> This approach is susceptible to errors and instability issues. Tryggvason *et al.*<sup>11</sup> computed the surface tension using tangent vector subtraction in every Eulerian cell, thus omitting the need to estimate curvature. Popinet and Zaleski<sup>12</sup> used tangent vector subtraction for every link between adjacent Lagrangian markers, exploiting the IB method's Lagrangian representation of the structure. Our method uses tangent vector summation for every Lagrangian marker (instead of every link). The subtle shift in perspective brings a side benefit that the component of the Young's force due to liquid–gas surface tension<sup>13</sup> at the contact point is readily computed by the two markers at both ends of a liquid–gas interface. In addition, our approach uses a stepwise re-sampling procedure to ensure a near-uniform distribution of Lagrangian markers in the liquid–gas interface. Finally, we employ a stepwise interface splicing method to implement interface topological changes, including droplet coalescence and separation.

This study is motivated by a real-life industrial problem, presented by W.L. Gore & Associates in 2018 during the Mathematical Problems in Industry (MPI), which is a workshop that attracts leading applied mathematicians and scientists from universities, industry, and national laboratories.<sup>14</sup> Catalysts are an integral part of many chemical processes. They are usually made of a dense but porous material, such as activated carbon or zeolites, which provide a large surface area. Liquids that are produced as a by-product of a gas reaction at the catalyst site are transported to the surface of the porous material, slowing down transport of the gaseous reactants to the catalyst site. One example of this is in a sulfur dioxide filter, which converts gaseous sulfur dioxide to liquid sulfuric acid.<sup>15,16</sup> Such filters are used in power plants to remove the harmful sulfur dioxide that would otherwise contribute to acid rain. Understanding the dynamics of liquid droplets in the gas channel of a device is critical in order to maintain performance and durability of the catalyst assembly. Among several other tests, the methods presented in this paper are applied to simulate 2D droplets moving on a vertical wall, which corresponds to this application. In this paper, we also empirically study the spatial and temporal convergence of our methods, compare simulation results at equilibrium against an analytical solution, and benchmark our methods with several standard test cases.

The paper is organized as follows: in Sec. II, we describe the IB method, the boundary conditions, the surface tension, and Young's force formulation, the moving contact point friction, the static-dynamic friction boundary condition, the stepwise interface re-sampling procedure, and the interface splicing procedure. An approach for simulating variable fluid density is also discussed. We then introduce the discretization of the governing equations in Sec. III. In Sec. IV, we verify the accuracy of our simulation results. In Sec. V, we apply our methods to several test cases. Finally, we conclude in Sec. VI with a discussion of our model and results, along with some insight into real-world applications.

## II. EQUATIONS OF MOTION

In this section, we describe the IB method that we use in this paper. IB has proven to be an extremely versatile method for fluid–structure interaction problems since its development almost forty

years ago.<sup>1,17–23</sup> The method represents the immersed structure with Lagrangian coordinates and the fluid with Eulerian or “lab frame” coordinates. The interactions between the two coordinate frames are communicated with integral transforms involving Dirac delta function kernels. For the solid surfaces and the variable density considered in this paper, we adopt the penalty immersed boundary (pIB) method.<sup>24,25</sup> For example, the vertical wall introduces a set of tether points that are fixed in space and represent the desired shape and location of the structure. The boundary markers are connected to the tether points via springs, and only the boundary markers interact with the fluid. Stiff springs approximate a rigid boundary. These stiff springs typically result in some numerical stiffness but simplify the implementation. In practice, accurate results can be achieved with an appropriate choice of the spring constant, spatial resolution, time stepping, and other numerical parameters.<sup>23–25</sup> The penalty immersed boundary method has been used to simulate a wide variety of problems that involve immersed structures with mass or variable density fluids. For example, Kim and Peskin used this approach to study oscillations of a cylinder in fluid as well as performed three-dimensional simulations of windsocks and flags.<sup>26</sup> The simulations of the oscillating cylinder were in reasonable agreement with experimental data.<sup>26</sup> For variable density problems, this method has been used to simulate bubble dynamics and Rayleigh–Taylor instabilities.<sup>24</sup> A disadvantage of this approach, as with many types of immersed boundary methods, is the regularization of the interface between low and high density fluid regions. The interface smoothing occurs because of the representation of high density regions using a discrete set of Lagrangian markers and the approximation of the Dirac delta function kernel, to be described below.

The equations of motion for the coupled fluid–structure system are

$$\rho_g \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}_1 + \mathbf{f}_2 + \mathbf{f}_3 + \mathbf{f}_4 + \mathbf{f}_5, \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\mathbf{f}_i(\mathbf{x}, t) = \begin{cases} \int \mathbf{F}_i(s, t) \delta(\mathbf{x} - \mathbf{X}_i(s, t)) ds, & i = 1, 2, \\ \int \int \mathbf{F}_i(r, s, t) \delta(\mathbf{x} - \mathbf{X}_i(r, s, t)) ds dr, & i = 3, \\ \mathbf{F}_i(t) \delta(\mathbf{x} - \mathbf{X}_i(t)), & i = 4, 5, \end{cases} \quad (2)$$

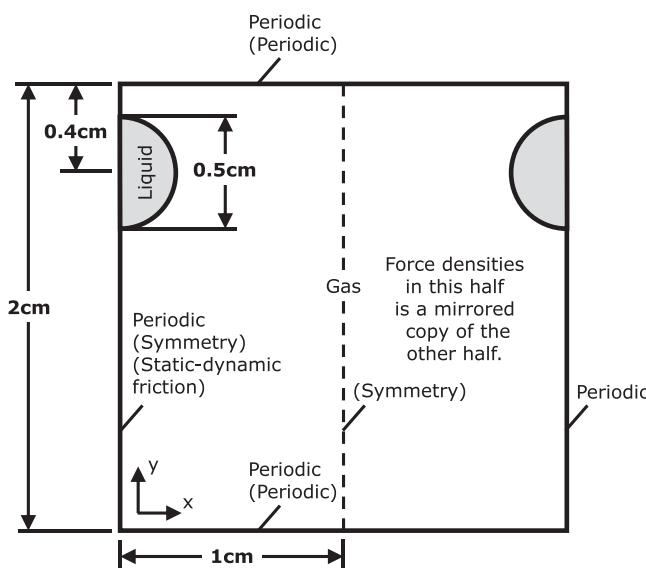
$$\frac{\partial \mathbf{X}_i}{\partial t}(t) = \int \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}_i(t)) d\mathbf{x}, \quad i = 1, 2, 3. \quad (3)$$

Here,  $\rho_g$  is the density of the gas,  $\mu$  is the dynamic viscosity of the gas and the liquid, and  $t$  denotes the time variable. The function  $\mathbf{u}(\mathbf{x}, t)$  is the velocity, and  $\delta$  is the 2D Dirac delta function. The vectors  $\mathbf{x}$  and  $\mathbf{X}_i$  denote the Eulerian fluid coordinates and the Lagrangian structure coordinates. In this paper, we consider three types of immersed structures: a 1D wall boundary ( $\mathbf{X}_1$ ), a 1D fluid–gas interface boundary ( $\mathbf{X}_2$ ), and a 2D variable density area ( $\mathbf{X}_3$ ). The vectors  $\mathbf{X}_4$  and  $\mathbf{X}_5$  are ephemeral markers used to impart the Young's force and the contact point friction force. For the 1D structures,  $s$  is the arc length that specifies a point on the boundary (i.e., a parametrization of the 1D boundary with respect to arc length). For the 2D structures,  $(r, s)$  is a 2D equidistant parametrization of the area. The functions  $\mathbf{f}_i$  and  $\mathbf{F}_i$  are the Eulerian and the Lagrangian force densities corresponding

to the immersed structures. The Lagrangian force densities  $F_1$  and  $F_2$  are 1D force densities,  $F_3$  is a 2D force density, and  $F_4$  and  $F_5$  are 0D force densities. Subsections II A–II G will introduce each type of immersed structure corresponding to  $X_i$  and define its Lagrangian force density  $F_i$ . Equations (1) are the Navier–Stokes equations for incompressible flow of a viscous fluid. In the regime, we study in the paper, there is little compressibility since Mach number is low; therefore, we simplify the simulation by assuming that all fluids are incompressible. Equation (2) calculates the force imparted from the immersed structures onto the fluid by converting Lagrangian force densities to Eulerian force densities using the Dirac delta function kernel. Equation (3) also uses the Dirac delta function to enforce the condition that the velocity of the immersed structure is equal to the fluid velocity, corresponding to a no-slip and no-penetration condition between the fluid and structure.

### A. Boundary conditions

Our code is adapted from the Matlab implementation of the IB method by Peskin,<sup>27</sup> which uses periodic boundary conditions for the computational domain. In most of our simulations, we fold the domain in half to obtain a symmetry boundary condition on the left and right boundaries. This section describes our implementation of symmetry boundary conditions within Peskin's implementation. In short, symmetry is prescribed in the Eulerian force density field, then the Eulerian fluid treatment solves for both symmetric halves simultaneously on a periodic domain. Specifically, if the computational domain exposed to the fluid solver is a square with side length  $L$ , as shown in Fig. 1, then we select the left half of it to be the problem



**FIG. 1.** The underlying computational domain exposed to the fluid solver is  $2 \times 2 \text{ cm}^2$ . The effective boundary conditions for the  $1 \times 2 \text{ cm}^2$  problem domain are shown via parenthesized texts. The effective symmetry boundary is obtained by mirroring half of the simulation domain. The static-dynamic friction condition is obtained by adding a wall (see Secs. II B and II E) at the left periodic boundary. The droplet is initialized 0.4 cm away from the top boundary, leaving ample space below for slipping.

domain. According to Fig. 1,  $L = 2$  is chosen. During each time step, once the Eulerian force densities (given by (2)) are imparted onto the grid, the simulation mirrors the Eulerian force densities matrix around  $x = L/2$  (i.e., the dashed line in Fig. 1) and adds them back to the original Eulerian force field. In this way, the fluid velocity field is always symmetric around  $x = L/2$ . Analytically, this symmetry treatment prevents structures from crossing the axis of symmetry.

### B. Wall as an immersed boundary

The main case that motivates our study is the simulation of liquid droplets moving on a solid vertical wall. We simulate the wall as an immersed boundary. In this specific case, since the wall is static, it is arguably easier to treat the wall as a boundary condition. However, we want our MCL method to generalize to non-static solid surfaces (see, e.g., Sec. IV C 4), so we treat the wall as an immersed boundary to maximize generality. Each Lagrangian marker of the wall is tethered to its ground-truth location, thus ensuring no-penetration and no-slip. In this case, the Lagrangian force density corresponding to the 1D wall is

$$F_1(s, t) = -k_1 (X_1(s, t) - Z(s)). \quad (4)$$

Equation (4) describes the tether-point construction and the associated spring force on the boundary to model a no-slip and no-penetration wall. The parameter  $k_1$  is the spring constant (taken to be  $5000 \text{ g}/(\text{s}^2 \text{ cm})$  in simulations) associated with the wall, which is a trade-off parameter between accuracy and numerical stability. If the spring constant is too small, the wall markers near the contact points will significantly deform under stress. If it is too large, numerical instabilities will likely arise. Since our domain is 2D, we define each solid-liquid-gas intersection to be a contact point. The vector  $Z$  is the ground-truth location for the markers. At the start of the simulation, we usually initialize  $X_1(s, 0) = Z(s)$  (with Sec. IV B as an exception). The wall behavior will be later modified in the MCL model, as described in Sec. II E.

### C. Surface tension and Young's force

We use the integral formulation as described by Popinet<sup>4</sup> to derive a model for surface tension. In this approach, each interface marker is pulled by its two neighbors at a constant force magnitude. In contrast to the work by Tryggvason *et al.*,<sup>11</sup> our method is entirely in the Lagrangian frame and does not estimate a tangent vector with a four-point polynomial fit. Instead, we use a piecewise linear fit to approximate the unit tangent vector  $\hat{T}$ , which is defined as

$$\hat{T}(s, t) = \frac{\partial X_2(s, t)}{\partial s}. \quad (5)$$

Then, the Lagrangian surface tension force density is

$$F_2(s, t) = \sigma \frac{\partial \hat{T}}{\partial s}, \quad (6)$$

where  $\sigma$  is the surface tension coefficient. Therefore, for a segment  $AB$  of the liquid–gas interface, as shown in Fig. 2, we have

$$\int_{\Omega} F_2 ds = \int_A^B \sigma \frac{\partial \hat{T}}{\partial s} ds = \sigma(\hat{T}_B - \hat{T}_A), \quad (7)$$

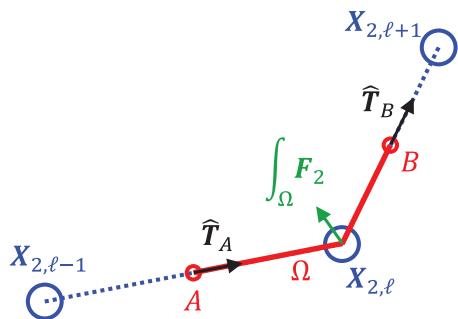


FIG. 2. An interface marker  $X_{2,\ell}$  is responsible for the surface tension force on the red segment  $AB$ .  $\hat{T}_A$  (the unit tangent vector at  $A$ ) is computed by subtracting the marker  $X_{2,\ell}$  from the adjacent marker  $X_{2,\ell-1}$ . The difference between  $\hat{T}_A$  and  $\hat{T}_B$  (the unit tangent vector at  $B$ ), multiplied by the tension coefficient  $\sigma$ , gives the surface tension force on the segment  $AB$ .  $\Omega$  is the set of interface points  $s$  on segment  $AB$  and  $F_2$  is the Lagrangian surface tension force density.

where  $\Omega$  is the set of interface points  $s$  on segment  $AB$ .  $\hat{T}_A$  and  $\hat{T}_B$  are the unit tangent vectors at  $A$  and  $B$ , respectively. With this approach, the sum of surface tension forces in a closed loop is zero.<sup>4</sup> In the discrete implementation, as shown in Fig. 2, we first compute the tangent vectors pointing from one marker to its adjacent markers and then apply the corresponding force of magnitude  $\sigma$  onto the marker.

A consequence of this implementation is that when the gas–liquid interface meets a solid surface, the Young’s force,<sup>13</sup> denoted by  $F_Y$ , arises directly from the tension forces. As shown in Fig. 3, the magnitude of the Young’s force is the tangential component of the sum of three tension forces

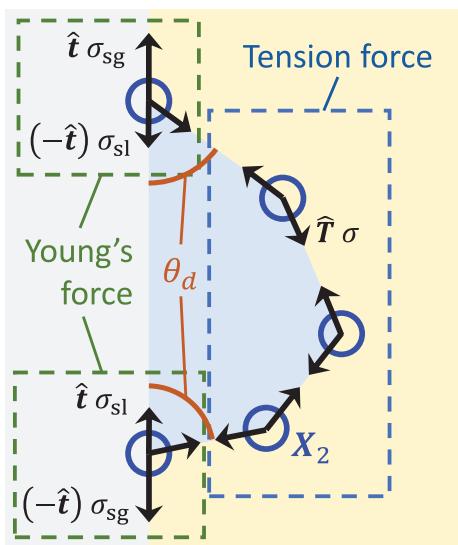


FIG. 3. The blue circles show the interface markers. The surface tension force with magnitude  $\sigma$  is applied in the tangent direction  $\hat{T}$  from every marker to its two adjacent markers. This scheme leaves the two markers at the wall with only one adjacent marker, whose pulling force becomes one of the three parts of the Young’s force. The other two parts of the Young’s force are of magnitude  $\sigma_{sg}$  and  $\sigma_{sl}$ , which are the solid–gas and solid–liquid tension coefficients, respectively. Here,  $\hat{t}$  is the upward unit vector tangent to the wall and  $\theta_d$  is the dynamic contact angle.

where  $F_Y = \hat{t} \cdot (\sigma \hat{T} + \sigma_{sl} \tilde{t} - \sigma_{sg} \tilde{t}) \hat{t}$ ,<sup>8</sup> where  $\hat{T}$  is the unit vector tangent to the liquid–gas interface,  $\hat{t}$  is the upward unit vector tangent to the wall, and  $\tilde{t}$  is the unit vector tangent to the wall and pointing toward the liquid phase.  $\sigma$ ,  $\sigma_{sl}$ , and  $\sigma_{sg}$  are the liquid–gas, solid–liquid, and solid–gas tension coefficients, respectively. The Young’s force is present at any three-phase contact point and is responsible for phenomena like the capillary action. The Young’s force also explains why slipping behaviors are often attributed to the contact angle. The static contact angle  $\theta_s$  is defined as the contact angle that will make  $F_Y = 0$ . Therefore, the static contact angle can be calculated by the following equation:

$$\sigma \cos \theta_s = \sigma_{sg} - \sigma_{sl}. \quad (9)$$

This also gives an alternate equation for the Young’s force that is based on the static contact angle:

$$F_Y = \tilde{t} \sigma (\cos \theta_d - \cos \theta_s). \quad (10)$$

Here,  $\theta_d$  is the dynamic contact angle. As shown in Fig. 3 and also discussed earlier, at a contact point, there is one interface marker that only has one neighbor. Therefore, (6) already supplies the first term of the three-force sum in (8). We compute the other two terms and add them to each contact point, thus completing the implementation of the Young’s force by

$$F_4 = \hat{t} \cdot (\sigma_{sl} \tilde{t} - \sigma_{sg} \tilde{t}) \hat{t} = -\tilde{t} \sigma \cos \theta_s. \quad (11)$$

Note that the points  $X_4$  are the locations where IB imparts  $F_4$ , i.e., the set of contact points.

#### D. Moving contact point friction

A frictional force is applied to each moving contact point. We assume that the friction is proportional to the slip velocity and depends on the dynamic contact angle. In this section, we first assume that the wall is free-slip, contrary to the no-slip setup described in Sec. II B. Toward the end of this section, we will relax this free-slip assumption.

Specifically, for each three-phase contact point, we apply a 0D force density at that point corresponding to a frictional force as

$$F_5 = -(\mathbf{u} \cdot \hat{t}) \eta(\theta_d) \hat{t}, \quad (12)$$

and  $\hat{t}$  is the upward unit vector tangent to the wall (so  $v = \mathbf{u} \cdot \hat{t}$  is the slip velocity) and  $\theta_d$  is the dynamic contact angle. The function  $\eta$  defines a friction coefficient (of units g/s) in terms of the dynamic contact angle  $\theta_d$  and is given by

$$\eta(\theta_d) = \begin{cases} 1.54 & \text{if } \theta_d > 2, \\ -8.48 \theta_d + 18.5 & \text{if } 1.117 < \theta_d \leq 2, \\ -19.1 \theta_d + 30.31 & \text{if } \theta_d \leq 1.117. \end{cases} \quad (13)$$

The parameter values in (13) are based on measured results from the simulation of water–silica contact line movement by Johansson and Hess.<sup>7</sup> Linear interpolation of measured data points is used to obtain  $\eta(\theta_d)$  as given in (13). Note that our MCL method is agnostic to the specific implementation of  $\eta$ ; therefore, future usages are free to modify  $\eta$ .

The linear response of  $F_5$  to the slip velocity [see (12)] is supported by molecular simulations by Johansson and Hess.<sup>7</sup> Such a linear relationship is related to the general Navier boundary condition (GNBC), where the slip velocity is proportional to the sum of the Young's force and the fluid viscous stress:<sup>28</sup>

$$\mathbf{u} \cdot \hat{\mathbf{t}} \propto \hat{\mathbf{t}} \cdot \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \mathbf{n} + \hat{\mathbf{t}} \cdot \mathbf{F}_Y,$$

where  $\mathbf{n}$  is the normal vector to the boundary. Similar to the Young's force, which is also a 0D force density,  $X_5$  is the location where IB imparts  $F_5$ . We simply set  $X_5$  to be the contact point locations. We can apply  $F_5$  to each contact point on an (otherwise) free-slip wall, but that will fail to recreate these two phenomena:

- A sufficiently small droplet should be able to hang statically on the wall without sliding down;
- The non-contact point region of the solid–liquid interface should be no-slip.

To represent the above phenomena, we apply  $F_5$  to a wall that otherwise follows the static-dynamic friction boundary condition, which we describe in Sec. II E.

### E. Static-dynamic friction boundary condition via length-limited springs

The so-called moving contact line (MCL) problem refers to the apparent contradiction that contact lines can move on a no-slip wall (in the case of our 2D methods, it is rather a “moving contact point” problem; however, we still use the “MCL” abbreviation to align with the literature). The mechanism of an MCL and how to simulate it had been largely a mystery until 1979.<sup>29</sup> Since then, researchers have developed many numerical models of MCLs.<sup>9,30</sup> A recent study<sup>31</sup> showed that hydrogen bonds facilitate the no-slip behavior of water on hydrophilic surfaces. These hydrogen bonds are orders of magnitudes stronger compared to the fluid's internal viscosity, so hydrophilic surfaces usually behave as if they were no-slip. This justifies the traditional no-slip assumption between water and hydrophilic surfaces. An MCL contradicts the no-slip assumption since the Young's force at the contact point

is strong enough to break hydrogen bonds, therefore, allowing slip. In this light, continuum models involving MCLs should have a mechanism to allow for slip. For example, Lai *et al.*<sup>6</sup> use the IB method with a Navier boundary condition at the wall. As opposed to the no-slip condition, the Navier boundary condition prescribes the slip velocity to be proportional to the viscous stress. However, their work has two limitations: (i) their boundary condition is static and cannot be imposed on a solid structure that changes shape; (ii) the entire boundary allows for slip, including segments far away from the MCLs. This section proposes a static-dynamic friction model for the wall, which (i) works with the IB method, (ii) can be prescribed onto an immersed structure that changes shape over time, and (iii) only allows slip near the MCLs.

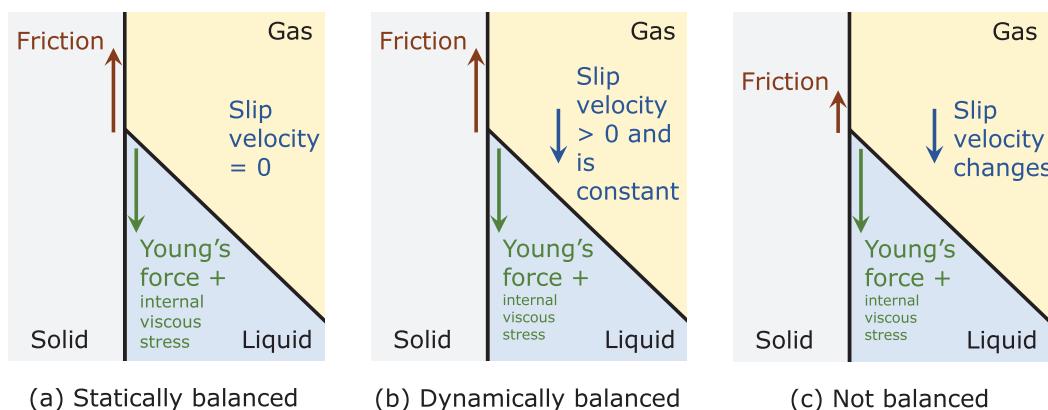
Figure 4 shows that the friction force should sometimes balance with other forces while other times be too weak to balance with other forces. Panels (a) and (b) show the statically and dynamically balanced cases, respectively, where the friction balances with other forces, such as the Young's force and the internal viscous stresses. In addition, the slip velocity is zero in (a) and a constant nonzero value in (b). Panel (c) depicts the situation in which friction is not strong enough to balance with other forces, implying the slip velocity changes with time. We implement this conditional behavior using what we call length-limited springs.

The method still treats the wall as an immersed structure but changes the penalty treatment by overriding the marker advection Eq. (3) for  $i = 1$ . Let  $\tilde{X}_1$  denote the positions that (3) would have advected the wall markers to, i.e.,

$$\tilde{X}_1(t + dt) = X_1(t) + dt \int \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - X_1(t)) d\mathbf{x}, \quad (14)$$

where  $dt$  is a small difference in time. Section III will define  $dt$  more formally in the context of the numerical discretization of these equations.

Then, Table I and Fig. 5 show the boundary conditions that can be achieved by specifying how to update the tangential component of wall marker positions. In the no-slip condition [Fig. 5(a)], we set  $\hat{\mathbf{t}} \cdot \mathbf{X}_1(t + dt) = \hat{\mathbf{t}} \cdot \tilde{X}_1(t + dt)$ . This corresponds to the original form of (3), so the wall markers strictly follow the fluid flow. The friction always statically balances with other forces such as Young's and the



**FIG. 4.** The schematic shows the solid–liquid–gas interfaces. In (a), the contact point is statically balanced. Here, the friction balances with the active forces, and the slip velocity is zero. In (b), the contact point is dynamically balanced and the friction balances with the active forces. Therefore, the slip velocity is positive, but stays constant. In (c), the contact point is not balanced, because the active forces exceed the friction force and as a result, the slip velocity changes.

**TABLE I.** Different rules to advect wall markers yield respective boundary condition and friction model.  $X_1$  is the wall marker, and  $\tilde{X}_1$  denotes the positions that (3) would have advected the wall markers to.  $\hat{t}$  is the upward unit vector tangent to the wall,  $Z$  is the ground-truth location for wall markers, and  $k_1$  is the spring stiffness associated with the penalty method. The length of the springs (when projected onto  $\hat{t}$ ) is limited by an upper bound  $q$ .

$\hat{t} \cdot X_1(t + dt)$	Boundary condition	Friction, i.e., tangential force in spring
$\hat{t} \cdot \tilde{X}_1(t + dt)$	No-slip	Statically balances with other forces
$\hat{t} \cdot Z$	Free-slip	0
$\max(\hat{t} \cdot Z - q, \hat{t} \cdot \tilde{X}_1(t + dt))$	Static-dynamic friction	At most $q k_1$ , otherwise statically balances with other forces

internal viscous stresses. In the free-slip condition [Fig. 5(b)], we set  $\hat{t} \cdot X_1(t + dt) = \hat{t} \cdot Z$ , where  $Z$  is the ground-truth location. The wall markers are not tangentially advected by the fluid flow, and their vertical position remains at the vertical position of their ground-truth location. The springs are at rest (tangentially), and thus, no friction force is applied onto the fluid. In the static-dynamic friction condition [Figs. 5(c) and 5(d)], the length of the springs (when projected onto the wall) is limited by an upper bound  $q$ . In this case, we set  $\hat{t} \cdot X_1(t + dt) = \max(\hat{t} \cdot Z - q, \hat{t} \cdot \tilde{X}_1(t + dt))$ . The friction force is at most  $q k_1$  and otherwise statically balances with other forces. Effectively, if local tangential forces are strong, e.g., containing a strong Young's force from a contact point, then the spring lengths are set to  $q$ , the friction provided is  $q k_1$ , and local tangential slipping occurs. Figure 5(c) shows this slipping case. On the other hand, if local tangential forces are weak, the spring lengths do not exceed  $q$  and static friction is sufficient to produce a no-slip result. Figure 5(d) shows this no-slip case. The slip is represented by

the difference between  $\hat{t} \cdot \tilde{X}_1(t + dt)$  and  $\hat{t} \cdot Z - q$ . Dividing that by  $dt$  gives the slip velocity. The heat dissipation rate is quantified by multiplying the friction force with the slip velocity

$$q k_1 \frac{\hat{t} \cdot \tilde{X}_1(t + dt) - (\hat{t} \cdot Z - q)}{dt}. \quad (15)$$

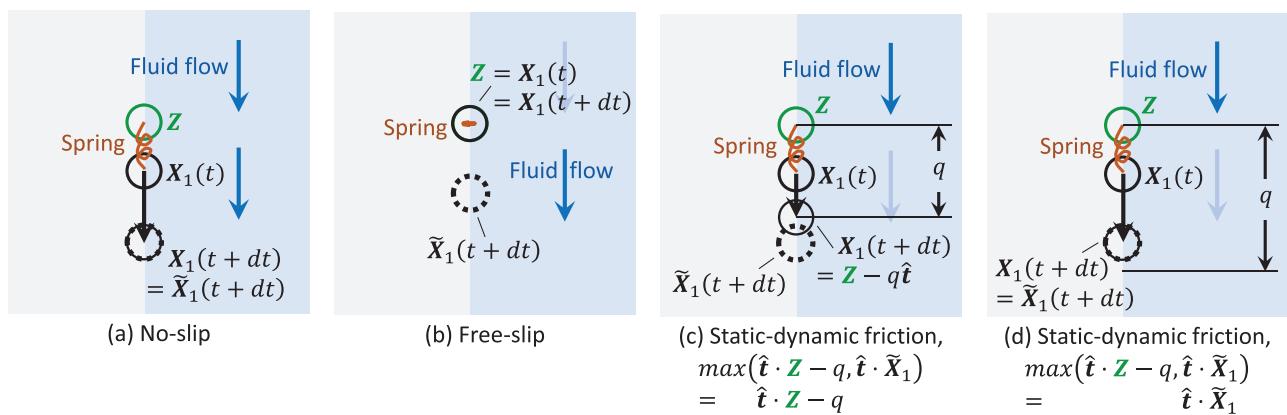
The free-slip and no-slip cases can be obtained from the static-dynamic case for particular choices of  $q$ . When  $q$  goes to infinity, we obtain the no-slip condition. The length limit is effectively removed and  $X_1 = \tilde{X}_1$ . When  $q$  is set to zero, we obtain the free-slip condition. In this light, the static-dynamic friction condition interpolates between the no-slip and the free-slip condition by controlling the value of  $q$ .

In practice, we set  $q = 5\text{Newtons}/k_1$ . This parameter value was chosen in order to obtain realistic results: The contact points can slip while the non-MCL region of the wall is no-slip, and small droplets can hang statically on the wall without sliding. To make  $q$  physically accurate, further molecular simulations or physical experiments are needed.

The static-dynamic friction condition bears similarity to the rigid-body static-dynamic friction model: The friction is a reactive force that responds to the sum of all active forces. If the sum of active forces is lower than a threshold, the rigid body is stationary and the frictional force balances with the other forces. If the sum of active forces exceeds a threshold, the rigid body slides and the dynamic friction equals the maximum static friction.

## F. Step-wise interface re-sampling

The immersed markers, as parametrized by arc length  $s$ , are initialized to be uniform. During a simulation, the markers are advected by the fluid flow, which may not conserve the arc length. The surface tension is always normal to the interface, so the fluid flow easily disrupts the distribution of the interface markers. To equi-distribute the markers on the structure, Lai *et al.*<sup>32</sup> used grid redistribution, while Hou *et al.*<sup>33</sup> and Lai *et al.*<sup>6</sup> applied artificial tangential velocity. Our method is similar to grid redistribution in the sense that we add



**FIG. 5.** The wall marker and its ground-truth location are denoted by  $X_1$  and  $Z$ , respectively. In (a), the wall is no-slip and the wall marker strictly follows the fluid flow. The spring conserves all temporary slip and, thus, forbids consistent slip. In (b), the wall is free slip and the wall marker does not follow the wall-tangential component of the fluid flow. The tangential component of the spring force is always zero; therefore, slip happens without friction. In (c) and (d), we apply the static-dynamic friction condition in which the spring, when projected onto the wall, has a length upper bound  $q$ . (c) Shows the case where  $\tilde{X}_1$  is lower than  $Z - q\hat{t}$ , so the marker is sent to  $Z - q\hat{t}$ . (d) Shows the other case where the spring is within its (tangential) length limit, so the wall marker follows the fluid flow. There is no modifications to the horizontal displacement in order to preserve the no-penetration condition.

markers to wide gaps and remove them from tight spaces at every time step. Specifically, at each time step, the program iterates over all the interface markers. When the distance between any pair of markers exceeds  $\sqrt{2}$  times their initial distance, the program inserts a new marker between them. When the distance between the outer two of any three adjacent markers become smaller than  $\sqrt{2}/2$  times their initial distance, the program removes the inner marker of the three. Note that the value  $\sqrt{2}$  is the geometric mean of 1 and 2. Given the conditions for adding and removing markers, the average distance between adjacent markers will remain in the range  $[\sqrt{2}/2, \sqrt{2}]$  times their initial distance.

Whenever a marker is removed, the two adjacent markers are joined together and their coordinates are left untouched. However, a sharpness check compares  $\cos \alpha$  (where  $\alpha$  is the angle formed by the three markers) with a threshold (0.9, in our case) to prevent the program from removing a marker that represents a relatively sharp corner and breaking conservation of area and tension energy. The sharpness check becomes irrelevant as the spatial resolution  $N$  approaches infinity but improves simulation accuracy for finite values of  $N$ .

On the other hand, when a marker is inserted (see Fig. 6), the program calculates two locations: the midpoint (MP) between the two markers and the intersection (IS) of the extrapolated lines from four adjacent markers. A weighted average between MP and IS is used as the location for the newly inserted marker. The averaging is weighted by a re-sampling amendment factor. If there is too much weight on the midpoint, the removal of markers will erroneously decrease tension energy while insertion will not increase tension energy. By having an optimal weight on the extrapolated intersection, we can keep the bias of the energy error to 0 (although the variance will still increase with time). However, finding the optimal amendment factor depends on an initial guess for the distribution of interface curvatures, which is out of the scope of this work. On the other hand, too much weight on the extrapolated intersection leads to alternating jagged edges and eventually to instabilities. In our numerical experiments, we set the re-sampling amendment factor to be 0.5. The amendment factor becomes irrelevant as  $N$  approaches infinity but improves simulation accuracy for finite values of  $N$ .

Our re-sampling routine constantly computes and records energy errors due to the artificial removal and insertion of markers. At the end of a simulation, these errors can be plotted for evaluation. See Sec. V B for the apparent improvement that this procedure brings. This re-sampling procedure, however, is very difficult to extend to 3D with a surface mesh.

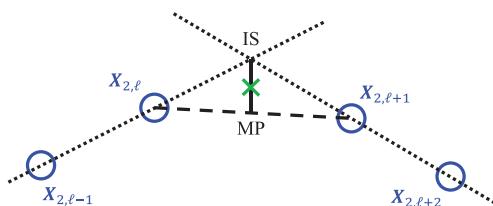


FIG. 6. The blue circles show four adjacent interface markers. When the distance between the inner two is larger than  $\sqrt{2}$  times its initial length, we insert a new marker. IS is the intersection of two lines, one connecting  $X_{2,\ell-1}$  to  $X_{2,\ell}$  and the other connecting  $X_{2,\ell+2}$  to  $X_{2,\ell+1}$ . MP is the midpoint of the line segment between  $X_{2,\ell}$  and  $X_{2,\ell+1}$ . The green cross shows the new marker, which is a linear combination of MP and IS weighted by an amendment factor.

### G. Interface splicing

Figure 7(a) shows six possible scenarios in which the liquid–gas interface changes topology and the chain of interface markers needs to be spliced: two droplets merging, one droplet splitting into two, a droplet attaching to the wall, a droplet detaching from the wall, a droplet splitting on the wall, and two droplets merging on the wall. Those six cases form three pairs of reversible processes as well as three pairs of scenarios whose implementations are exactly the same. The same-implementation cases are locally indistinguishable, since our splicing implementation is agnostic to the phase (liquid or gas) on each side.

To implement splicing, we store the interface markers in circular doubly linked lists. The linking direction preserves polarity information; if one follows the links in the positive direction, the liquid will always be on the right. Figure 7(b) illustrates a splicing event. When the distance between two interfaces is smaller than a threshold, we check whether the two interfaces are approaching using the sign of the dot product of their velocities. If both conditions hold true, the interfaces are spliced together. The method has two different distance thresholds, one for interface-interface events ( $h$ ), and the other for interface-wall

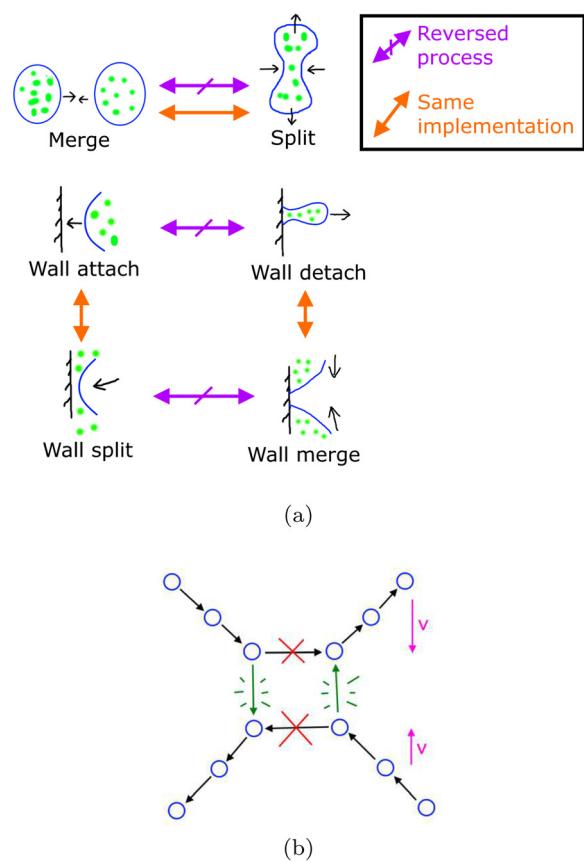


FIG. 7. (a) Among the six interface splicing scenarios, three pairs are reversed in time and three pairs share the same implementation. (b) An important condition for splicing to occur is that the two interfaces are approaching each other, as shown with the violet arrows representing the velocity of each interface. When the distance between four involved markers become small enough, the algorithm splices the interfaces (removing two black links and adding two green links).

events ( $2.3h$ ), where  $h$  is the meshwidth (i.e., diameter of one Eulerian grid cell). These specific parameters result in stable splicing events according to our numerical tests. Finally, a splicing event forces the simulation to skip the splicing subroutine for two subsequent time steps. This rule makes splicing events more atomic so that there will not be multiple splicing events competing to accomplish the same macroscopic effect.

## H. Variable density

Our model uses the technique proposed by Kim and Peskin<sup>24</sup> to describe the different densities for the liquid and gas within and outside of the droplet, respectively. In this approach, a uniform 2D grid of Lagrangian markers is constructed in the liquid phase to represent the density difference. This difference is achieved by allowing the markers to have an effective mass by tethering them to their massive, Newtonian counterparts using the penalty immersed boundary (pIB) method. More formally, the Newtonian particles have velocity  $\mathbf{v}$ ,

$$\mathbf{v}(s, r, t) = \frac{\partial \mathbf{Y}(s, r, t)}{\partial t}, \quad (16)$$

where  $\mathbf{Y}$  is the location of the Newtonian particles. The forces on the Newtonian particles are the tether force and gravity

$$\Delta\rho \frac{\partial \mathbf{v}(s, r, t)}{\partial t} = k_3 (\mathbf{X}_3(s, r, t) - \mathbf{Y}(s, r, t)) - \Delta\rho G \hat{\mathbf{j}}, \quad (17)$$

where  $\Delta\rho = \rho_l - \rho_g$  is the 2D density difference between the liquid and gas,  $G$  is the gravitational constant,  $\hat{\mathbf{j}}$  is the vertical unit vector, and  $k_3$  is the spring constant [taken to be  $1000 \text{ g}/(\text{s}^2 \text{ cm}^2)$  in our simulations] associated with the variable density method. If  $k_3$  is too high, numerical instability occurs and if it is too low, the allowed distance will be violated (explained below). A tether force is applied as a force density onto the fluid,

$$\mathbf{F}_3(s, r, t) = k_3 (\mathbf{Y}(s, r, t) - \mathbf{X}_3(s, r, t)), \quad (18)$$

to couple the particles  $\mathbf{X}_3$  with the Newtonian particles  $\mathbf{Y}$ . Our implementation uses the variable density method,<sup>24</sup> with several modifications. Kim and Peskin<sup>24</sup> used a five-step update method for time stepping, while we use a coarser, midpoint method (see Sec. III). We also set the allowed distance<sup>34</sup> to be one tenth the mesh-width, i.e.,  $h/10$ .

## III. NUMERICAL DISCRETIZATION

For the spatial discretization, we use a global Navier–Stokes solver based on harmonics in the velocity field. This solver uses the Fast Fourier Transform to accelerate the computation, but at the cost of not being able to handle variable density/viscosity out-of-the-box (which is why we need the variable density method (Sec. IIIH) to distinguish the two phases). The time step is denoted by  $dt$  with units of seconds. The spatial resolution parameter  $N$  is the number of Eulerian cells in the problem domain along the  $y$  axis. The length of cell side in the spatial discretization, i.e., the meshwidth, is denoted by  $h$  with units of centimeters. Therefore, we always have  $Nh = L$ , where  $L$  is the height of the square computational domain as shown in Fig. 1.

The time step index is denoted by  $n \in \{0, 1, 2, \dots\}$  and the physical time at time step  $n$  is given by  $t^n = n dt$ . The discrete Eulerian space coordinates that label the cells in the Cartesian mesh are

$(i, j) \in \{0, 1, 2, \dots, N - 1\}^2$ . A physical cell position in the problem domain is given by  $\mathbf{x}_{ij} = (ih, jh)$ . The fluid velocity  $\mathbf{u}(x_{ij}, t^n)$  at position  $x_{ij}$  and time  $t^n$  is approximated by the discrete fluid velocity, denoted by  $\mathbf{u}_{ij}^n$ . The array of discrete velocities at time step  $n$  is denoted by  $\mathbf{u}^n = (\mathbf{u}_{ij}^n)$ . The discrete spatial parametrization variables for the immersed structures are  $\ell \in \{0, 1, 2, \dots, \ell_{\max}\}$  and  $m \in \{0, 1, 2, \dots, m_{\max}\}$ , where  $\ell_{\max}$  and  $m_{\max}$  can change during simulation due to the interface re-sampling. A point within the parametrization of the immersed structure is given by  $(r_\ell, s_m) = (\ell \Delta r, m \Delta s)$ , where  $\Delta r$  and  $\Delta s$  control the discrete spatial resolution of the structures. In our numerical experiments, we initialize  $\ell_{\max}$  and  $m_{\max}$  so that  $\Delta r = \Delta s = h/2$ . The spatial resolution of the structures corresponding to the set of Lagrangian markers has implications on discrete mass conservation and correspondingly the computed pressures. If the resolution of the structures is too coarse, for example, the droplet might “leak” through the interface and begin to nonphysically shrink in volume. The Lagrangian resolution of  $h/2$ , which is explicitly finer than the fluid resolution, is chosen to avoid these numerical artifacts. The discrete 2D structure positions corresponding to the variable density area  $\mathbf{X}_3$  are indexed as  $\mathbf{X}_{3,\ell m}^n \approx \mathbf{X}_3(r_\ell, s_m, t^n)$ , and we collect them in a single array denoted by  $\mathbf{X}_3^n = (\mathbf{X}_{3,\ell m}^n)$ . We use the same notation for the discretizations of the 1D wall boundary  $\mathbf{X}_1$  as well as the 1D fluid–gas interface boundary  $\mathbf{X}_2$ , although in these cases, there is only a single parametrization variable. We represent the array of discrete structure positions at time step  $n$  by  $\mathbf{X}^n = (\mathbf{X}_1^n, \mathbf{X}_2^n, \mathbf{X}_3^n)$ .  $\mathbf{X}_4$  and  $\mathbf{X}_5$  correspond to 0D points, so there is not a need for their discretization.

We use a midpoint method for evolving the system in time that relies on quantities at intermediate time steps  $\{\frac{1}{2}, \frac{3}{2}, \dots\}$ . This method improves stability and constrains numerical errors. Following the approach from works by Peskin,<sup>27,35</sup> the procedure below describes the time discretization for the equations of motion (1), (2), and (3):

- (1) Given the marker positions  $\mathbf{X}^n$  and the fluid velocity  $\mathbf{u}^n$ , compute the first-order approximation of the marker positions at the midpoint step  $\mathbf{X}^{n+1/2}$  using Eq. (3).
- (2) Given the marker positions at the midpoint step  $\mathbf{X}^{n+1/2}$ , compute the force densities at the midpoint, denoted  $\mathbf{f}^{n+1/2}$ , using Eq. (2).<sup>36</sup>
- (3) Given the fluid velocity field  $\mathbf{u}^n$  and the force densities at the midpoint step  $\mathbf{f}^{n+1/2}$ , compute the fluid velocity field at both the midpoint  $\mathbf{u}^{n+1/2}$  and the next time step  $\mathbf{u}^{n+1}$ , using Eq. (1).
- (4) Given the marker positions  $\mathbf{X}^n$  and the fluid velocity at the midpoint step  $\mathbf{u}^{n+1/2}$ , advect the markers for time  $dt$  to obtain the marker positions at the next time step  $\mathbf{X}^{n+1}$  using Eq. (3).

## IV. BENCHMARK AND CONVERGENCE TESTS

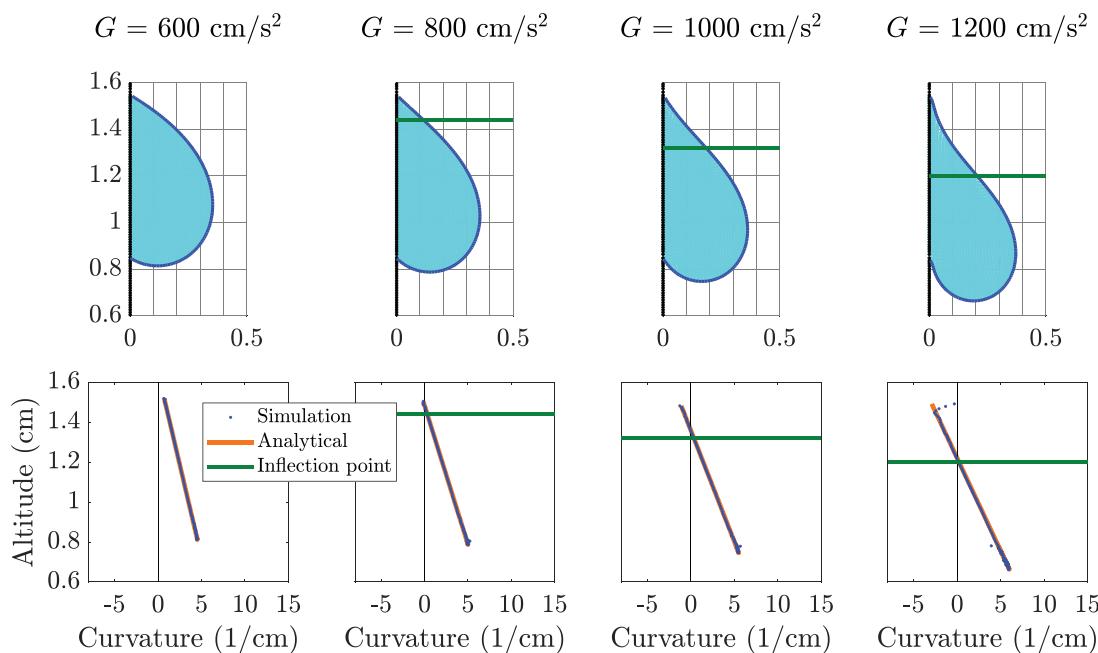
This section focuses on benchmark and convergence tests for our methods. In Sec. IV A, we simulate a hanging droplet on a wall and compare our steady state results with an analytical solution.<sup>14</sup> In Sec. IV B, we compare our simulation results with an established variable density benchmark in which a bubble rises in a liquid column.<sup>37</sup> Finally, in Sec. IV C, we describe results from nontrivial experiments to empirically study the convergence of our methods.

### A. Droplet in hydro-static equilibrium on a wall

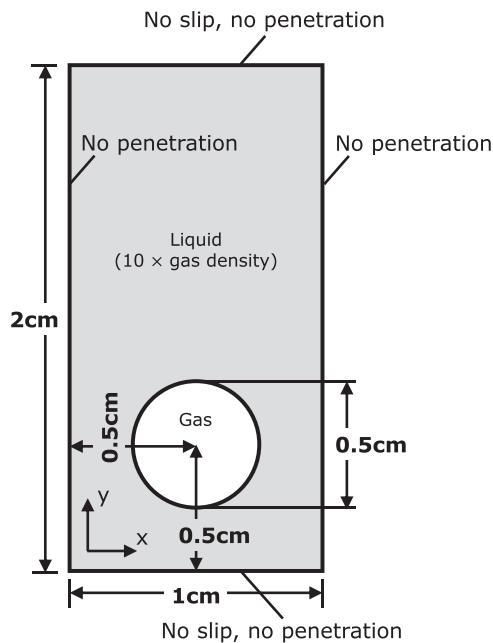
In this test, we compare our simulation results with an analytical solution for the interface shape of a droplet hanging on a vertical

wall.<sup>14</sup> The model describes the hydro-static equilibrium shape of a droplet hanging on a vertical wall as a partial differential equation expressed in Cartesian coordinates. The solution to this problem implies a linear relationship between the local curvature of the interface and the altitude. Details regarding derivation of the analytical solution are given in the work by Cimpeanu *et al.*<sup>14</sup>

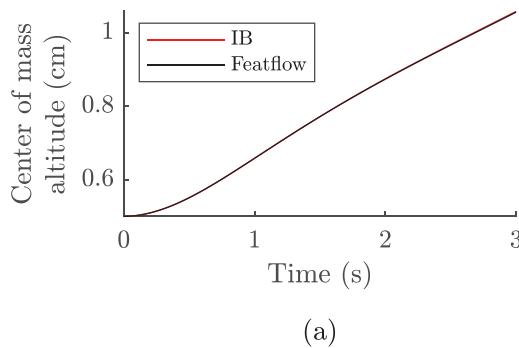
For these simulations, the time step is  $dt = 0.0001$  s, the spatial resolution parameter is  $N = 96$ , the height of the computational domain is  $L = 2$  cm, the surface tension coefficient is  $\sigma = 100$  g cm/s<sup>2</sup>, the gas density is  $\rho_g = 0.1$  g/cm<sup>3</sup>, the liquid density is  $\rho_l = 1$  g/cm<sup>3</sup>, the viscosity is  $\mu = 1$  g/s, and the static contact angle is  $\theta_s = \pi/2$ . We run several simulations and vary the magnitude of gravity coefficient  $G$  to compare curvatures in the equilibrium state with those from the analytical solution given in the work by Cimpeanu *et al.*<sup>14</sup> To estimate local curvature at each marker, we compute the circle that contains the three interface markers and calculate the inverse of the circle's signed radius. The simulations are terminated at  $t = 0.2$  s, at which time all three cases have reached a steady state. Results are shown in Fig. 8. From left to right, the gravitational constant  $G$  (cm<sup>2</sup>/s) is gradually increased to alter the droplet shape. The upper panel depicts the equilibrium droplet shape computed by our simulations. A linear relation is observed between the 2D curvature and the altitude as shown in the lower panel. In addition, there is excellent agreement between our numerical results (blue scattered dots) and the analytical solution (orange line) given in the work by Cimpeanu *et al.*<sup>14</sup> In both panels, the green line represents the location where curvature is zero, i.e., the inflection point in the droplet shape.



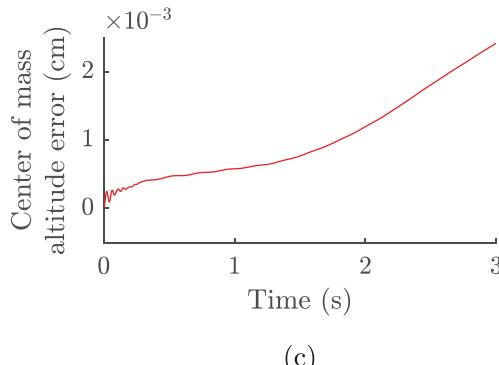
**FIG. 8.** The upper panel shows the equilibrium droplet shape computed from our simulations. The lower panel displays altitude vs curvature for the hanging droplet. The analytical solution obtained by Cimpeanu *et al.*<sup>14</sup> is shown by the orange line, and the blue scattered dots correspond to the curvature calculated from our simulation results. From left to right, the gravitational constant  $G$  (cm<sup>2</sup>/s) is gradually increased to alter the droplet shape. The green line highlights the location where curvature is zero, i.e., the inflection point in the droplet shape. To estimate curvature, we use a fixed-arc length sliding window over the interface markers, and fit a circle through three markers in the window. The three markers consist of both ends of the window and the midpoint in the window.



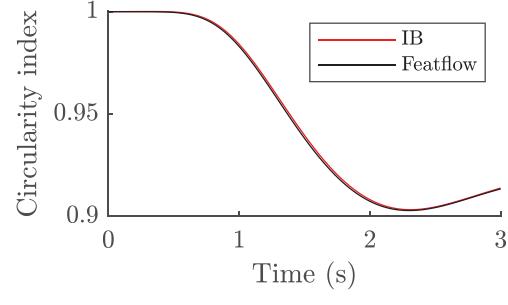
**FIG. 9.** The schematic of the rising bubble benchmark case problem domain. The top and bottom walls are no-slip and no-penetration, while the left and right walls are no-penetration. The gas and the liquid form a 1:10 density ratio.



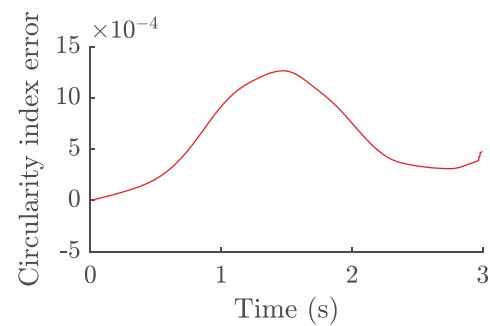
(a)



(c)



(b)



(d)

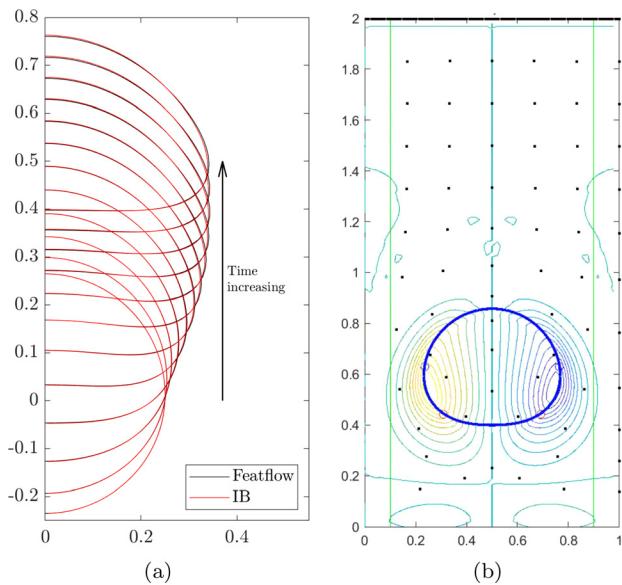
**FIG. 10.** A comparison of our simulation results against those computed from Featflow.<sup>37</sup> The model is of a rising gas bubble in a liquid column. (a) and (b) show the altitude evolution of the center of mass and the circularity index of the bubble, respectively, for our simulation (IB) and Featflow. (c) and (d) show the absolute error of the IB and Featflow results corresponding to (a) and (b), respectively.

Figure 10 depicts the center of mass and the circularity index calculated from our simulation results compared against those from Featflow.<sup>37</sup> To keep track of the center of mass of the bubble, the simulation initializes the problem domain with  $X_6$ , an additional 2D equidistant grid of markers inside the bubble.  $X_6$  is solely for accounting purposes and does not exert any forces. The mean of the current positions of the markers  $X_6$  corresponds to the bubble's center of mass. The circularity index is defined in the work by Turek and Mierka,<sup>37</sup> which is the perimeter of an area-equivalent circle divided by the perimeter of the bubble. Our simulation does not calculate an area-equivalent circle but instead uses the initial circle circumference as a proxy, since in this test, the area is well conserved. The center of mass and circularity index calculated from our simulations show good agreement with the Featflow benchmark results. Figure 11(a) shows excellent agreement in the position and shape of our simulated bubble vs the benchmark results. Figure 11(b) (Multimedia view) provides more details on the rising bubble.

In addition, Table II shows the order of accuracy with regard to  $N$  for this test. The finest simulation ( $N = 162$ ,  $dt = 0.0016$ ) is used as the ground truth, since in this case, we do not have an analytic solution. The fluid velocity ( $\mathbf{u}$ ) error is computed in the  $L^2$  norm. The table shows that the velocity field converges at a second order rate and the center of mass position converges with a rate of approximately 3/2.

### C. Convergence tests

In this section, we describe the results of several tests in which we vary the time step and the spatial resolution to empirically verify



**FIG. 11.** (a) shows a plot of the bubble shape over time for our simulation and the results generated from Featflow.<sup>37</sup> The axes units are in cm. The boundary of the bubble is shown at different points in time, with time increasing from bottom to top. The final bubble position corresponds to  $t = 3$  s. Our method agrees extremely well with the results from Featflow,<sup>37</sup> even with a relatively coarse spatial resolution,  $N = 128$ . (b) provides more details on the rising bubble. Multimedia view: <https://doi.org/10.1063/5.0086452.1>

convergence of our methods. For all tests in this section, the height of the computational domain is  $L = 2$  cm, the surface tension coefficient is  $\sigma = 50$  g cm $^{-2}$ , the gas density is  $\rho_g = 0.1$  g/cm $^3$ , the liquid density is  $\rho_l = 1$  g/cm $^3$ , and the viscosity is  $\mu = 0.01$  g/s. The size of the computational domain is  $2 \times 2$  cm $^2$ , and the size of the problem domain is  $1 \times 2$  cm $^2$ . Therefore,  $x \in [0, L/2]$ ,  $y \in [0, L]$ , where  $L = 2$ . Periodic boundary conditions are used for the top and bottom boundaries, and a symmetry boundary condition is used for the left and right boundaries (see Sec. II A).

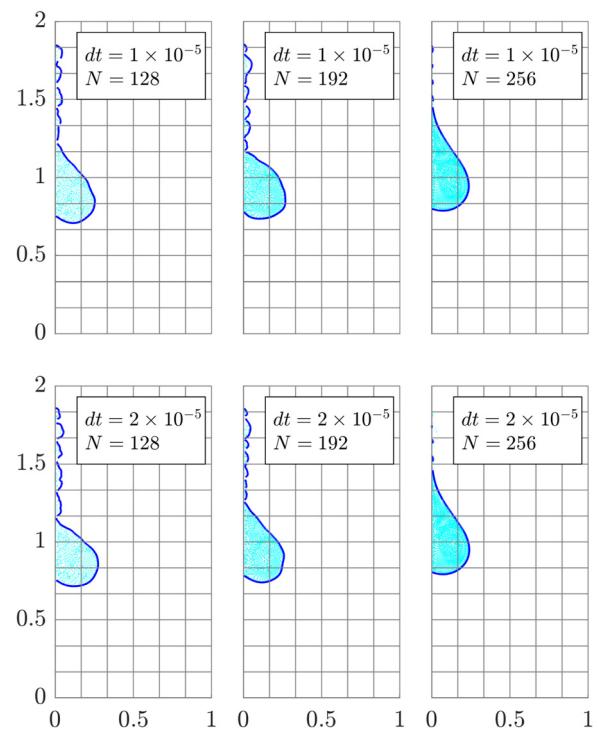
### 1. Test 1: Sliding droplet

For this test, we simulate a droplet sliding down a vertical wall as shown in Fig. 12 (Multimedia view). The Reynolds number is  $Re = 200$ , where the typical length scale is the diameter of the droplet, and the characteristic velocity is selected as the downward velocity of the droplet at its maximum toward the end of the simulation.

There is a vertical wall at  $x = 0$ . The static contact angle is  $\theta_s = 0.3\pi$ . There is a uniform upward flow of gas, which corresponds

**TABLE II.** Order of accuracy for the rising bubble test.

$N$	$dt$	M error	CI error	$u$ error (velocity)	CM order	CI order	$u$ order
		(center of mass)	(circularity index)				
32	0.0016	0.000 042	0.014 118	8.222 747	N/A	N/A	N/A
48	0.0016	0.000 036	0.000 358	4.860 008	0.378 784	9.063 589	1.296 941
72	0.0016	0.000 027	0.001 209	1.904 728	0.642 360	-3.002 786	2.310 188
108	0.0016	0.000 016	0.000 807	0.742756	1.307 378	0.996 498	2.322 585



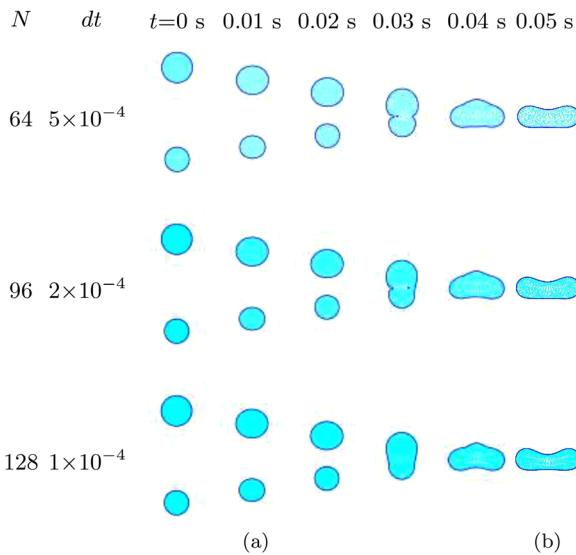
**FIG. 12.** Test 1: this figure shows the shape of the sliding droplet when  $t = 0.09$  s for several different values of  $dt$  and  $N$ . The axes units are in cm. The simulation is terminated after the droplet stops moving. Notice the trail of fluid left by the droplet as it slides down the wall. Multimedia view: <https://doi.org/10.1063/5.0086452.2>

to a prescribed vertical component of the velocity at  $y = 0$ . More precisely, the boundary condition at  $y = 0$  (i.e., top and bottom boundary) for the vertical velocity component is  $30\tanh(4x/L)$  cm/s for  $x \in [0, L/2]$ .

We vary the time step  $dt$  and the spatial resolution  $N$ . The sliding droplet shape at time  $t = 0.09$  s is shown. The results appear to converge if we increase the spatial resolution  $N$  to 256.

### 2. Test 2: Coalescence of two droplets

In test 2, we simulate two droplets coalescing without gravity. Each droplet is given a uniform initial velocity field, one upward and one downward, to simulate their coalescence. Results are shown in Fig. 13(a). The coalescence of two droplets is an extremely challenging test case since the moment of droplet merging amplifies previous numerical perturbations. We increase the spatial and temporal

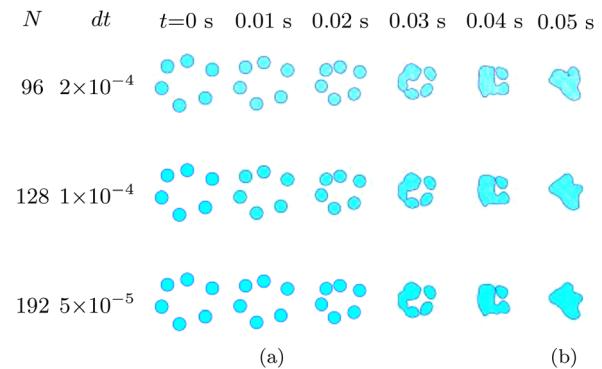


**FIG. 13.** Test 2: (a) shows the coalescence of two droplets. Three levels of spatial and temporal resolutions are shown. Little improvement is gained from  $N = 96$  to  $N = 128$ , rendering  $N = 128$  a high enough spatial resolution, while holding other parameters constant. (b) provides more details on the coalescence of two droplets. Multimedia view: <https://doi.org/10.1063/5.0086452.3>

resolutions (holding other parameters constant), and the simulation results show good spatial and temporal convergence with our surface tension model and interface splicing procedures. Such observations can be more vividly seen in Fig. 13(b) (Multimedia view).

### 3. Test 3: Coalescence of six droplets

Figure 14(a) shows our simulation results for coalescing six droplets for three levels of spatial and temporal resolutions. An artificial radially symmetric, inward pointing body force for gravity

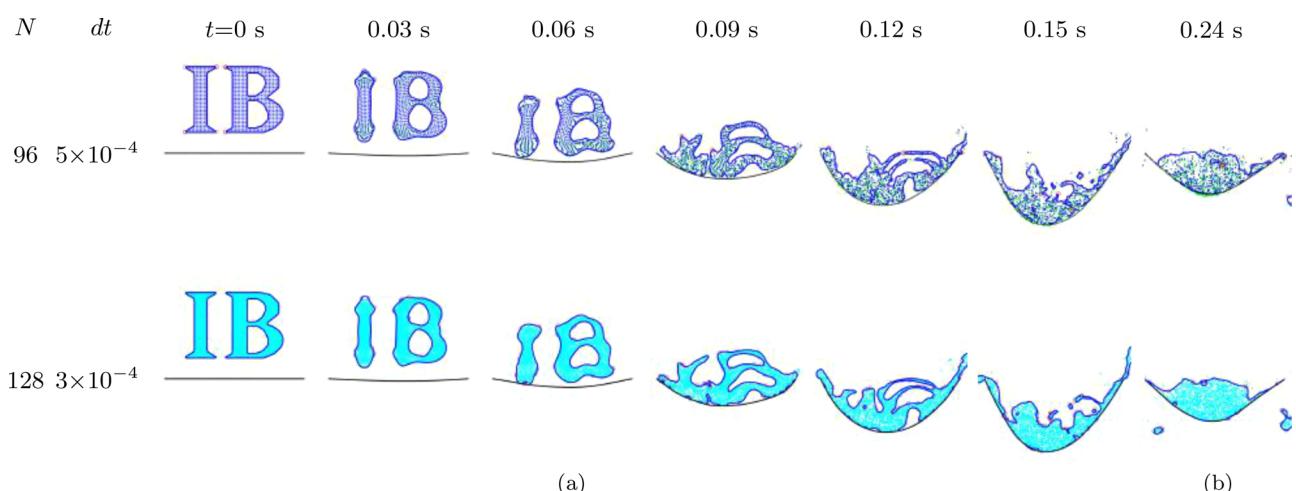


**FIG. 14.** Test 3: (a) six droplets coalesce under three levels of spatial and temporal resolutions. Little improvement is achieved from  $N = 128$  to  $N = 192$ , rendering  $N = 192$  a high enough spatial resolution, while holding other parameters constant. (b) provides more details on coalescence of six droplets. Multimedia view: <https://doi.org/10.1063/5.0086452.4>

( $G = 900 \text{ cm/s}^2$ ) pulls the droplets together. This merging test is difficult since the variance of the system's response to the first coalescence event is amplified by the successive merging events. More details are given in Fig. 14(b) (Multimedia view) for this test.

### 4. Test 4: Letters “IB” fall into an elastic pouch

Here we initialize liquid droplets in the shape of the letters “IB” and let them fall into an elastic pouch. Results are shown in Fig. 15(a). The Reynolds number  $Re = 8000$ , where the typical length scale is 2 cm, the width of the letter “I.” At the beginning of the simulation, before the droplets interact with the pouch, they slightly change shape because of surface tension. As the droplets hit the pouch, the pouch deforms and the letter “B” changes topologically as the gas bubbles exit the droplet. This test demonstrates the capabilities of our method in an all-in-one setting. Note that here the solid membrane is dynamic, rendering static boundary conditions unusable. This shows that our



**FIG. 15.** Test 4: (a) letter-shaped droplets falling into an elastic pouch, simulated with two levels of spatial and temporal resolutions. This test highlights our method's ability to simulate an MCL on a changing-shape solid surface. (b) provides more details on the falling of letter-shaped droplets. Multimedia view: <https://doi.org/10.1063/5.0086452.5>

methods are capable of simulating a moving contact point between a liquid–gas interface and a changing-shape solid surface, simulating fluid dynamics, surface tension, Young's force, moving contact line, and solid elasticity all at the same time. See Fig. 15(b) (Multimedia view) for a video of this test.

## V. APPLICATIONS

In this section, we describe several applications of our methods. In Sec. V A, we consider the effect of droplet size on its dynamics as it slides down a wall. The importance of our interface re-sampling procedure is explored in Sec. V B. Finally, in Sec. V C, we showcase some additional tests in which interfaces are spliced to represent topological changes. All tests in Sec. V share the setup for the computational and

problem domains found at the beginning of Sec. IV C 1, as well as the following parameters. The surface tension coefficient is  $\sigma = 50 \text{ g cm/s}^2$ , and the gravitational acceleration is  $G = 980 \text{ cm/s}^2$ . The gas density is  $\rho_g = 0.1 \text{ g/cm}^3$ , and the liquid density is  $\rho_l = 1 \text{ g/cm}^3$ . The viscosity for both the liquid and gas phases is  $\mu = 0.01 \text{ g/s}$ . The static contact angle is  $\theta_s = 0.3 \pi$ .

### A. Droplets sliding down a wall

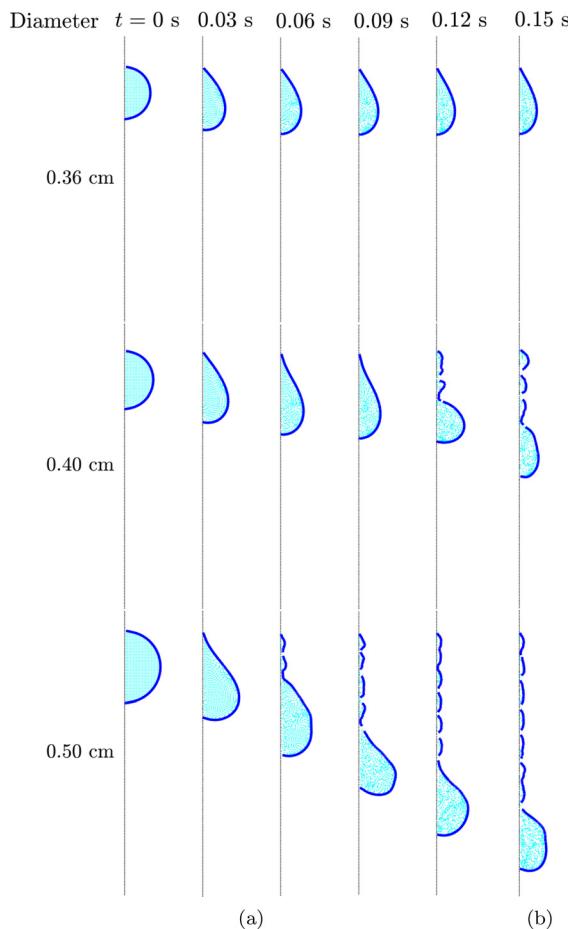
In this section, we consider droplets sliding down a wall, where gravity competes against an upward air flow.

#### 1. Effect of droplet size

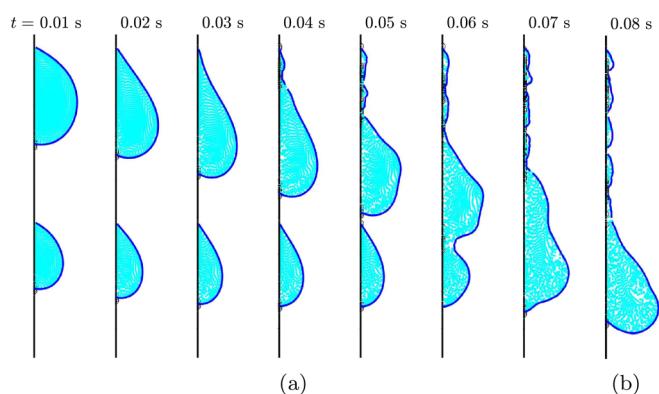
In Fig. 16(a), we analyze the dynamics of various sized droplets. The smallest droplet corresponding to the top panel of Fig. 16(a) does not slide; rather, it obeys the “no-slip” rule. This is consistent with reality and is implemented via the static-dynamic friction condition. Our method tracks the wall markers that are vertically displaced by the MCL model at every time step. We observe that vertical displacements only occur near the contact point (usually within 6–8 cells). This means that although our MCL model applies to the entire wall, the majority of the wall that interacts with the droplet behaves as a no-slip boundary, and only the contact region slips. This is consistent with previous literature, which suggests that the fluid's internal viscous stress is not enough to incur apparent slip in subsonic flows with hydrophilic surfaces.<sup>38</sup> The Young's force is the dominating tangential force near the contact point. Such observations can be more vividly seen in Fig. 16(b) (Multimedia view).

#### 2. Large droplet merges with a small droplet

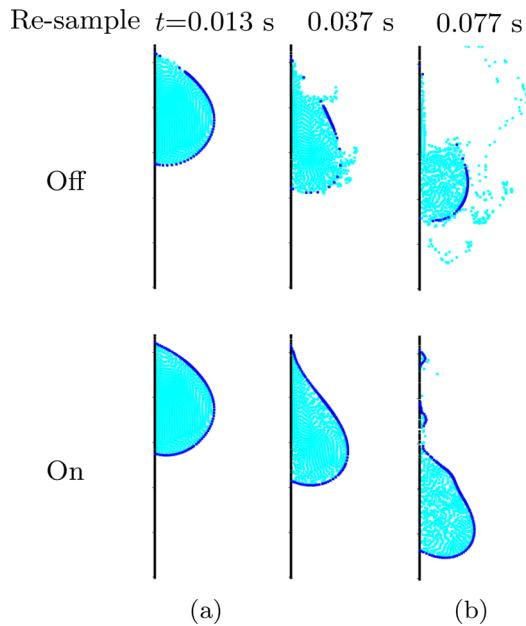
In Fig. 17(a), we initialize two droplets of different sizes on the wall. The larger droplet slides down faster than the smaller one. Therefore, it catches the smaller droplet and coalesces with it. This test case shows the system's response to two different droplet sizes in one simulation and, at the same time, demonstrates the capabilities of our interface splicing capabilities at the wall. More details are given in Fig. 17(b) (Multimedia view) for this experiment.



**FIG. 16.** (a) shows results from several simulations with various droplet sizes. We initialize the droplets as semi circles with diameters corresponding to 0.36, 0.4, and 0.5 cm. The bigger droplets are heavier, so they slide down the wall more quickly. The smallest droplet is stationary. For diameter 0.36 cm, the receding dynamic contact angle is  $29^\circ$  and the advancing dynamic contact angle is  $93^\circ$ . For diameter 0.4 cm, the receding dynamic contact angle is  $16^\circ$  and the advancing dynamic contact angle is  $75^\circ$ . For diameter 0.5 cm, the receding dynamic contact angle is  $13^\circ$  and the advancing dynamic contact angle is  $68^\circ$ . These values are taken before any splitting of droplets. (b) provides more details on sliding droplets with different sizes. Multimedia view: <https://doi.org/10.1063/5.0086452.6>



**FIG. 17.** (a) shows results from a simulation in which a big droplet catches a smaller droplet and coalesces with it. Time increases from the left to right. This test demonstrates the effect of droplet size in a single simulation. (b) Provides more details on the coalescence of two sliding droplets. Multimedia view: <https://doi.org/10.1063/5.0086452.7>



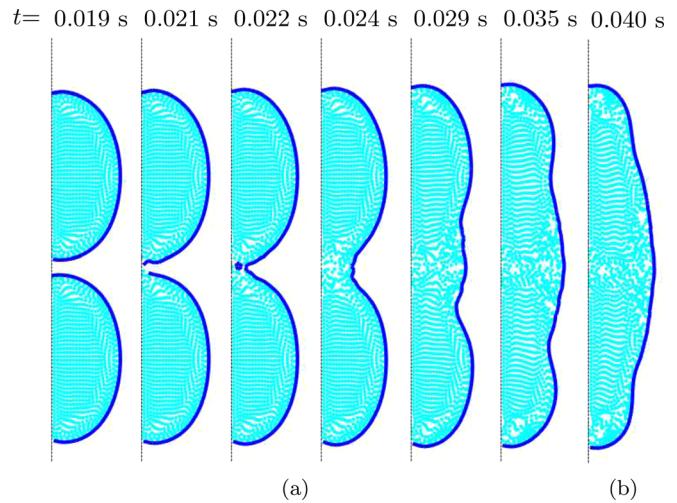
**FIG. 18.** (a) shows results from two simulations, one which uses the interface re-sampling procedure (bottom), and one which does not (top). Without re-sampling, the fluid flow advects the interface markers and the interface breaks into pieces, allowing for penetration through the interface. This shows the importance of our interface re-sampling procedure. (b) provides more details on the interface re-sampling of a sliding droplet. Multimedia view: <https://doi.org/10.1063/5.0086452.8>

### B. Performance of the re-sampling procedure

Figure 18(a) shows a comparison study on the effect of our step-wise re-sampling technique. In the upper row, we do not re-sample, resulting in badly distributed Lagrangian markers and, eventually, topological errors. In the lower row, the re-sampling procedure is used and resolves this issue. See Fig. 18(b) (Multimedia view) for a side-by-side comparison video.

### C. Demos of interface splicing

Figure 19(a) shows results from a simulation in which a single droplet separates into two smaller ones. After the splicing moment, surface tension quickly brings the sharp edge into the body of the

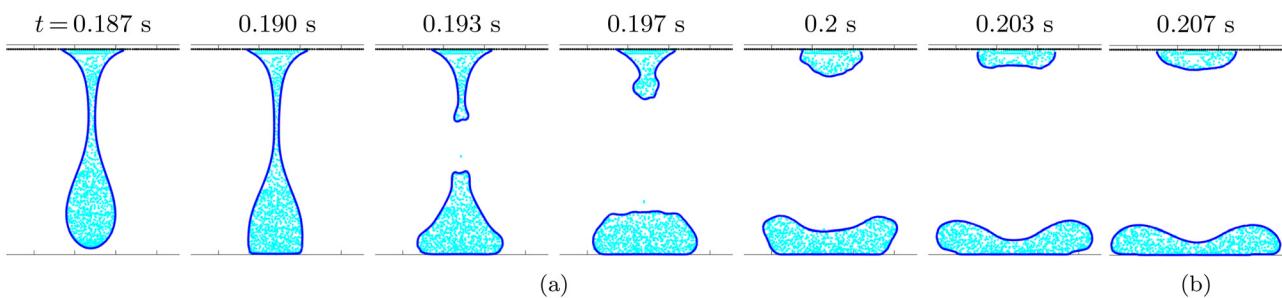


**FIG. 20.** (a) shows the results from a simulation in which two droplets coalesce on the wall. This showcases the capacity of our interface splicing procedure to handle droplet coalescence at the wall due to two approaching contact points. (b) provides more details on the splicing of a droplet. Multimedia view: <https://doi.org/10.1063/5.0086452.10>

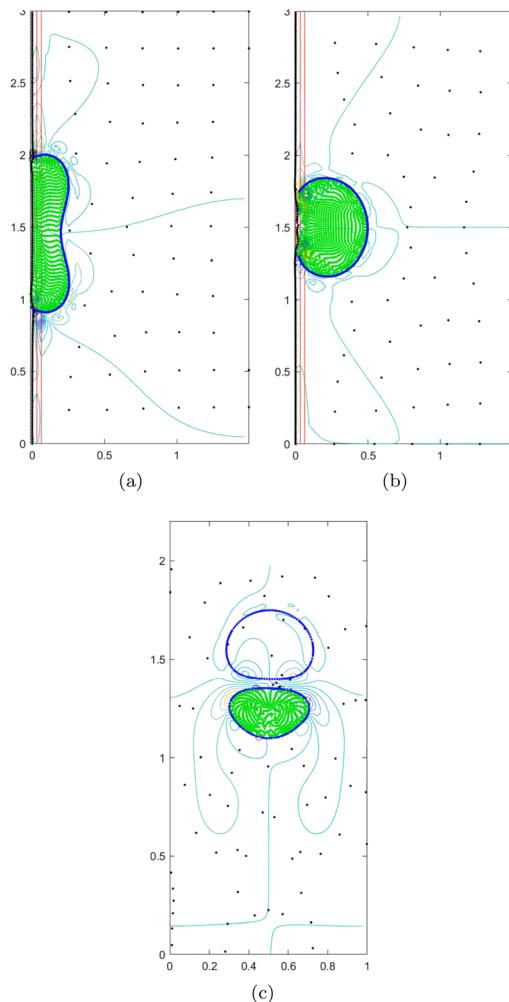
droplet. See Fig. 19(b) (Multimedia view) for a video of this experiment. In Fig. 20(a), two droplets coalesce on the wall. This is one of the six basic cases of interface splicing introduced earlier in Sec. II G. See Fig. 20(b) (Multimedia view) for a video of this experiment. Additional simulations can be found in Figs. 21(a)–21(c) (Multimedia views). In Fig. 21(a), a droplet on the wall separates into two droplets as a result of a collision with the wall. A droplet attaches to the wall in Fig. 21(b). Finally, in Fig. 21(c), an air bubble collides with a droplet, showing the effect of variable density.

### VI. CONCLUSIONS

In this paper, we use the immersed boundary method to simulate droplets and their interaction with a wall. The static-dynamic friction is modeled with an immersed boundary. The surface tension and the Young's force were readily integrated with the IB method by computing local unit vectors tangent to the interface. We showed that interface marker re-sampling is necessary to obtain accurate results. Our interface splicing method handles droplet coalescence, separation and other topological changes with stable splicing events. We also



**FIG. 19.** (a) shows the results from a simulation in which a single droplet separates into two smaller ones. It showcases the ability of our interface splicing method to handle droplet separation. (b) provides more details on the splicing of a droplet. Multimedia view: <https://doi.org/10.1063/5.0086452.9>



**FIG. 21.** (a) A droplet on the wall separates into two droplets, (b) a droplet attaches to the wall and (c) an air bubble collides with a droplet. Multimedia views: <https://doi.org/10.1063/5.0086452.11>; <https://doi.org/10.1063/5.0086452.12>; <https://doi.org/10.1063/5.0086452.13>

demonstrated empirical convergence of our methods on non-trivial examples, and we applied them to several benchmark cases, including a slipping droplet on a wall and a rising bubble.

The success of our methods demonstrates the flexibility and extensibility of the IB method. We show that complex dynamics, like a moving contact point on an evolving solid surface, can be successfully modeled with the IB method. In the formulation of the static-dynamic friction condition, we choose to implement contact point slip via length-limiting the tether springs so that IB imparts the correct friction forces onto the fluid. Conceivably there are other ways to impart this frictional force, which we did not investigate in this study. For example, one could formulate the friction as an additional external force that is imparted directly onto the fluid without intermediate markers, or one could prescribe a boundary condition for the fluid velocity according to viscous stress.

Our work is limited in several ways. The methods presented here assume that the fluids are incompressible. This is a result of the

immersed boundary approach used here in which the entire domain is modeled as an incompressible fluid. We expect the interface splicing method to be fragile in the compressible-flow setting because inertia and compressibility will likely make splicing events much less atomic. In addition, although our methods can handle variable density, they do not handle variable viscosity. Finally, the methods are limited to 2D. In 3D, the interface surface becomes a surface mesh (most commonly, a triangle mesh). Care needs to be taken in computing surface tension in this case. Furthermore, the stepwise interface re-sampling procedure, in 3D, needs to also repair local topology of the mesh while preserving accuracy and numerical stability. Therefore, extending our methods to 3D will be a nontrivial task.

An important application of our work is a real-life industrial problem corresponding to the removal of gaseous reactants from power plant exhaust by converting them into liquid at the catalyst site. Future work will be geared toward understanding the dynamics of liquid droplets for this application, in order to maintain performance and durability of the catalyst assembly. More specifically, these types of models will provide details for classifying droplet movement and coalescence in the gas channel based on the filter surface and gas flow properties, which will be used to guide operating conditions for sulfur dioxide removal devices. Our goal is to develop such a mathematical model, which will help us to understand how different surface properties and operating conditions affect the dynamics of the droplet or the formation of a liquid film.

## ACKNOWLEDGMENTS

C.P. and P.S. gratefully acknowledge support from the National Science Foundation (NSF) under Grants No. RTG/DMS-1646339. P.S. is also supported by an Institutional Support of Research and Creativity (ISRC) grant provided by New York Institute of Technology and financial support from NSF under Grants No. DMS-2108161. We thank Otto Mierka and Stefan Turek for providing the 10:10 viscosity benchmark data for rising bubble test case 1. Several very helpful conversations with Guanhua Sun are gratefully acknowledged.

## AUTHOR DECLARATIONS

### Conflict of Interest

The authors have no conflicts to disclose.

### Author Contributions

M.Y.L. and D.C. contributed equally to this work.

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request. All simulations in this study can be reproduced with the source code openly available at repository <https://github.com/daniel-chin/droplet>. The repository also contains the scripts to plot all the figures.

## REFERENCES

- 1C. S. Peskin, “Flow patterns around heart valves: A numerical method,” *J. Comput. Phys.* **10**, 252–271 (1972).
- 2R. J. LeVeque and Z. Li, “Immersed interface methods for Stokes flow with elastic boundaries or surface tension,” *SIAM J. Sci. Comput.* **18**, 709–735 (1997).

<sup>3</sup>D. Lörstad, M. Francois, W. Shyy, and L. Fuchs, "Assessment of volume of fluid and immersed boundary methods for droplet computations," *Int. J. Numer. Methods Fluids* **46**, 109–125 (2004).

<sup>4</sup>S. Popinet, "Numerical models of surface tension," *Annu. Rev. Fluid Mech.* **50**, 49–75 (2018).

<sup>5</sup>W.-X. Huang and W. Guo, "An improved penalty immersed boundary method for multiphase flow simulation," *Int. J. Numer. Methods Fluids* **88**, 447–462 (2018).

<sup>6</sup>M.-C. Lai, Y.-H. Tseng, H. Huang *et al.*, "Numerical simulation of moving contact lines with surfactant by immersed boundary method," *Commun. Comput. Phys.* **8**, 735 (2010).

<sup>7</sup>P. Johansson and B. Hess, "Molecular origin of contact line friction in dynamic wetting," *Phys. Rev. Fluids* **3**, 074201 (2018).

<sup>8</sup>M. Muradoglu and S. Tasoglu, "A front-tracking method for computational modeling of impact and spreading of viscous droplets on solid walls," *Comput. Fluids* **39**, 615–625 (2010).

<sup>9</sup>H.-R. Liu and H. Ding, "A diffuse-interface immersed-boundary method for two-dimensional simulation of flows with moving contact lines on curved substrates," *J. Comput. Phys.* **294**, 484–502 (2015).

<sup>10</sup>S. Manservisi and R. Scardovelli, "A variational approach to the contact angle dynamics of spreading droplets," *Comput. Fluids* **38**, 406–424 (2009).

<sup>11</sup>G. Tryggvason, B. Bunner, A. Esmaeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan, "A front-tracking method for the computations of multiphase flow," *J. Comput. Phys.* **169**, 708–759 (2001).

<sup>12</sup>S. Popinet and S. Zaleski, "A front-tracking algorithm for accurate representation of surface tension," *Int. J. Numer. Methods Fluids* **30**, 775–793 (1999).

<sup>13</sup>T. Quian, X.-P. Wang, P. Sheng *et al.*, "Generalized navier boundary condition for the moving contact line," *Commun. Math. Sci.* **1**, 333–341 (2003).

<sup>14</sup>R. Cimpeanu, Y. Daneshbod, Q. Li, P. Sanaei, P. Dubovski, A. Nadim, and M. Chugunova, "Motion of liquid droplets in gas channels in a  $\text{SO}_2$  module," (2021).

<sup>15</sup>K. B. Kiradjev, V. Nikolakis, I. M. Griffiths, U. Beuscher, V. Venkateshwaran, and C. J. Beward, "A simple model for the hygroscopy of sulfuric acid," *Ind. Eng. Chem. Res.* **59**, 4802–4808 (2020).

<sup>16</sup>K. B. Kiradjev, C. J. Beward, I. Griffiths, and D. W. Schwendeman, "A homogenized model for a reactive filter," *SIAM J. Appl. Math.* **81**, 591–619 (2021).

<sup>17</sup>D. McQueen and C. Peskin, "Shared-memory parallel vector implementation of the immersed boundary method for the computation of blood flow in the beating mammalian heart," *J. Supercomputing* **11**, 213–236 (1997).

<sup>18</sup>K. M. Arthurs, L. C. Moore, C. S. Peskin, E. B. Pitman, and H. Layton, "Modeling arteriolar flow and mass transport using the immersed boundary method," *J. Comput. Phys.* **147**, 402–440 (1998).

<sup>19</sup>M.-C. Lai and C. S. Peskin, "An immersed boundary method with formal second-order accuracy and reduced numerical viscosity," *J. Comput. Phys.* **160**, 705–719 (2000).

<sup>20</sup>B. E. Griffith, X. Luo, D. M. McQueen, and C. S. Peskin, "Simulating the fluid dynamics of natural and prosthetic heart valves using the immersed boundary method," *Int. J. Appl. Mech.* **1**, 137–177 (2009).

<sup>21</sup>F. Balboa, J. Bell, R. Delgado-Buscallioni, A. Donev, T. Fai, B. Griffith, and C. Peskin, "Staggered schemes for incompressible fluctuating hydrodynamics," (2011).

<sup>22</sup>D. Devendran and C. S. Peskin, "An immersed boundary energy-based method for incompressible viscoelasticity," *J. Comput. Phys.* **231**, 4613–4642 (2012).

<sup>23</sup>P. Sanaei, G. Sun, H. Li, C. S. Peskin, and L. Ristroph, "Flight stability of wedges," *J. Fluids Struct.* **101**, 103218 (2021).

<sup>24</sup>Y. Kim and C. S. Peskin, "Numerical study of incompressible fluid dynamics with nonuniform density by the immersed boundary method," *Phys. Fluids* **20**, 062101 (2008).

<sup>25</sup>Y. Kim and C. S. Peskin, "A penalty immersed boundary method for a rigid body in fluid," *Phys. Fluids* **28**, 033603 (2016).

<sup>26</sup>Y. Kim and C. S. Peskin, "Penalty immersed boundary method for an elastic boundary with mass," *Phys. Fluids* **19**, 053103 (2007).

<sup>27</sup>C. Peskin, see <https://github.com/ModelingSimulation/IB-MATLAB> "Ib-matlab" (2019).

<sup>28</sup>J.-F. Gerbeau and T. Lelievre, "Generalized navier boundary condition and geometric conservation law for surface tension," *Comput. Methods Appl. Mech. Eng.* **198**, 644–656 (2009).

<sup>29</sup>E. Dussan, "On the spreading of liquids on solid surfaces: Static and dynamic contact lines," *Annu. Rev. Fluid Mech.* **11**, 371–400 (1979).

<sup>30</sup>Y. Sui, H. Ding, and P. D. Spelt, "Numerical simulations of flows with moving contact lines," *Annu. Rev. Fluid Mech.* **46**, 97–119 (2014).

<sup>31</sup>P. Johansson, A. Carlson, and B. Hess, "Water–substrate physico-chemistry in wetting dynamics," *J. Fluid Mech.* **781**, 695–711 (2015).

<sup>32</sup>M.-C. Lai, Y.-H. Tseng, and H. Huang, "An immersed boundary method for interfacial flows with insoluble surfactant," *J. Comput. Phys.* **227**, 7279–7293 (2008).

<sup>33</sup>T. Y. Hou, J. S. Lowengrub, and M. J. Shelley, "Removing the stiffness from interfacial flows with surface tension," *J. Comput. Phys.* **114**, 312–338 (1994).

<sup>34</sup>The allowed distance is a parameter in the method by Kim and Peskin.<sup>24</sup> It sets an upper bound for the distance between a boundary marker and its Newtonian counterpart. Our program monitors the maximum distance between a boundary marker and its Newtonian counterpart. Once it exceeds the allowed distance, the program raises a warning for the user to increase the pIB spring stiffness and rerun the simulation.

<sup>35</sup>C. S. Peskin, "The immersed boundary method," *Acta Numer.* **11**, 479–517 (2002).

<sup>36</sup>Converting the Lagrangian force density  $F$  to the Eulerian force density  $f$  uses the discrete 2D Dirac delta function.<sup>35</sup>

<sup>37</sup>S. Turek and O. Mierka, "Numerical simulation and benchmarking of drops and bubbles," in *Handbook of Numerical Analysis* (Elsevier, 2021), Vol. 22, pp. 419–465.

<sup>38</sup>T. Qian, X.-P. Wang, and P. Sheng, "Power-law slip profile of the moving contact line in two-phase immiscible flows," *Phys. Rev. Lett.* **93**, 094501 (2004).