

A 2D sharp and conservative VOF method for modeling the contact line dynamics with hysteresis on complex boundary

Chong-Sen Huang^{a,1}, Tian-Yang Han^{b,1}, Jie Zhang^{a,*}, Ming-Jiu Ni^{b,*,**}

^a State Key Laboratory for Strength and Vibration of Mechanical Structures, School of Aerospace, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China

^b School of Engineering Science, University of Chinese Academy of Sciences, Beijing 101408, China

ARTICLE INFO

Keywords:

Volume-of-fluid method
Embedded boundary method
Height function method
Contact line dynamics
Contact angle hysteresis

ABSTRACT

A sharp and conservative numerical method is proposed for studying the 2D contact line dynamics along complex geometrical boundaries, while a hybrid volume-of-fluid and embedded boundary method is designed to model the liquid/gas and fluid/solid interfaces, respectively. Unlike diffusive numerical methods that artificially thicken the interface to enhance stability, our method devises a sharp and precise scheme for discretizing the advection term of the VOF equation, with special attention to arbitrary solid boundaries within the same discretized interfacial cell. This scheme conserves the volume fraction exactly and maintains numerical stability even in small and irregular cells cut by the solid boundary. Another novel aspect of our contribution is the precise imposition of the contact angle condition at the contact line. A special height function method is designed and implemented for cells cut by the solid boundary. Furthermore, the contact angle condition, whether static or dynamic during contact line motion, is extended to more general cases by considering hysteresis phenomena. The code is released on the *Basilisk* website, enabling the first implementation of a sharp geometrical VOF method capable of accurately simulating contact line dynamics on complex solid substrates in 2D.

1. Introduction

Contact line problems are ubiquitous in both nature and industry, representing multiphase scenarios where immiscible fluids interact with solid boundaries. Examples include the condensation and evaporation of droplets on solid substrates, liquid penetration through porous media, and underground oil recovery. This problem is illustrated in Fig. 1, where panel (a) shows a flat solid substrate (“regular boundary”), and panel (b) depicts a more general situation with an irregular solid substrate, which is referred to “complex geometrical boundary” in the present study that it implies an arbitrary solid boundary embedded in the Cartesian grid. The domain consists of three phases, denoted by $\Omega = \Omega_g \cup \Omega_l \cup \Omega_s$, with subscripts g , l , and s representing gas, liquid, and solid, respectively. These immiscible phases are separated by the liquid interface $\Gamma_l = \Omega_g \cap \Omega_l$ and the solid boundary $\Gamma_s = \Omega_s \cap (\Omega_l \cup \Omega_g)$, represented by blue and red lines, respectively, in Fig. 1. The flow behavior of the liquid depends greatly on the wettability of the solid substrate,

* Corresponding author.

** Corresponding author.

E-mail addresses: j.zhang@xjtu.edu.cn (J. Zhang), mjni@ucas.ac.cn (M.-J. Ni).

¹ They contribute equally to this paper.

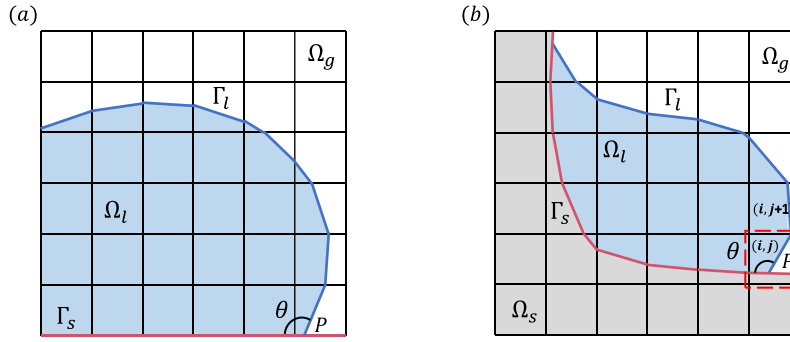


Fig. 1. Schematic illustrating the problem of moving contact lines on (a) a flat solid substrate (regular boundary) and (b) an irregular solid substrate (complex geometrical boundary). The solid domain Ω_s is separated from the fluid domain $\Omega_l \cup \Omega_g$ by the solid boundary Γ_s (red line). The liquid domain Ω_l is separated from the gas domain Ω_g by the liquid/gas interface Γ_l (blue line). The angle between Γ_s and Γ_l in the (i, j) cell is the contact angle θ , and the intersection point is the triple point P . (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

characterized by the contact angle at the triple point P , denoted by $\theta = \Gamma_l \cap \Gamma_s$. Note that θ is not always constant; it likely varies with the local velocity near the triple point and can involve the pinning phenomena known as contact angle hysteresis.

Numerical simulations have become a reliable method for investigating multiphase flows involving contact line dynamics due to their high flexibility and low cost. While numerical methods for modeling contact line motion on flat solid substrates are well-established [1–4], dealing with irregular substrates, including embedded boundaries, remains challenging for numerical simulations. Fig. 1(b) illustrates the two main challenges by focusing on the discretized control volume (i, j) , circled by the dashed box where both Γ_l and Γ_s are present in the mixed cell, which refers to a discretized cell containing an embedded solid boundary. On one hand, accurately tracking Γ_l becomes challenging due to its coexistence with Γ_s in the same mesh. On the other hand, precisely imposing the contact angle condition at the triple point is difficult because Γ_s acts as an irregular solid boundary.

Before discussing the first challenge, it is important to note that for complete discretized cells, such as $(i, j + 1)$, the ordinary volume-of-fluid (VOF) method [5,6] and the level-set (LS) method [7] can be applied directly to track liquid/gas interfaces. These methods enable the development of sharp schemes by calculating volume fluxes using geometrical or distance functions [8–11]. However, things become much more complex in the presence of Γ_s , which renders discretized cells, such as (i, j) , incomplete and only partially covered by liquid/gas. This necessitates careful consideration when transporting Γ_l . In single-phase flows, complex solid geometry can be handled by developing numerical techniques specialized for Cartesian grids. For instance, the immersed boundary method (IBM) [12,13] and the embedded boundary method (EBM) [14,15] convert boundary conditions at embedded boundaries into forces or fluxes added to the momentum equations. Therefore, the combination of these two sharp methods, respectively designed for the liquid/gas free surface and the fluid/solid embedded boundary, is expected to perform well in modeling contact line motion in mixed cells. However, these attempts have not been entirely successful. For instance, hybrid IBM-VOF methods [16,17], EBM-VOF methods [18], and IBM-LS methods [19] have been proposed to extend VOF- and LS-based numerical methods onto complex and irregular solid boundaries. Nevertheless, all of these methods fail to conserve mass during contact line motion, even though VOF-type methods exhibit good conservative performance in simulating multiphase flows on flat substrates. Particularly, Patel et al. [17] highlighted that the VOF method could not conserve mass in the presence of Γ_s , which continuously deteriorated the conservation property in simulations. More recently, Tavares et al. [18] introduced a traditional advection scheme to transport the interface in the presence of Γ_s . This method significantly reduces the complexity of geometric computations within mixed cells and can be naturally extended to 3D. Furthermore, by disregarding the influence of embedded solid boundaries during advection, the approach mitigates the time step constraint imposed by small mixed cells. However, the results remain suboptimal due to the challenges associated with reconstructing the liquid/gas interface within irregular mixed cells. Besides these hybrid sharp-sharp schemes, there are also hybrid sharp-diffusive methods to tackle moving contact line problems with complex boundaries. Liu et al. [20] used the IBM to model the embedded boundary, while a diffuse-interface method was incorporated to simulate free surface flow. Although this hybrid scheme avoided the challenge of handling two sharp interfaces within a single cell, it is still very difficult to accurately conserve the fluid mass. After reviewing the state-of-the-art, we are motivated to design a more accurate and robust sharp-sharp scheme to transport the liquid/gas interface in mixed cells. Moreover, the conservation property of mass should be preserved during advection.

The second challenge involves imposing the contact angle condition at embedded solid boundaries, where the value of θ likely varies with time and crucially influences the computation of the surface tension force, or more precisely, the curvature of the interface near the triple point. Over the past two decades, significant progress has been made in implementing contact angle boundary conditions on regular boundaries using diffusive methods [3] or sharp methods [21,1,22]. Challenges remain in extending these methods to complex geometrical boundaries, while they are addressed in two different directions: utilizing the normal vector determined by the contact angle to compute curvature [23,16], and employing the ghost cell method to extend the computational domain into solid cells [20,18]. Compared to the first kind of methods, the second is considered more accurate, but inevitably more sophisticated to implement, especially for irregular boundaries. In particular, the height function (HF) method is renowned for its accuracy in estimating interface curvature. However, the presence of Γ_s in mixed cells introduces unstructured meshes, leading to the loss of symmetry when imposing the contact angle boundary condition based on the traditional HF method [22]. Recently, O'Brien et al. [16] highlighted the challenge of rigorously defining the height-function in mixed cells due to the loss of geometrical symmetry in the presence of Γ_s .

This reminds us that it is still incomplete and unsuccessful to incorporate the HF technique into those hybrid sharp-sharp approaches. Otherwise, if the contact angle is not constant but determined by possible hysteresis phenomena due to the influence of microscopic roughness and hydrophilic-hydrophobic disparities of real materials, then θ falls between the advancing and receding angles. While the implementation of such contact angle conditions with hysteresis is already quite successful on flat boundaries within the VOF [2,4,24] or LS frameworks [1,25], as far as the authors know, sharp interfacial schemes have rarely been implemented to simulate the contact angle hysteresis on complex boundaries. The main difficulty lies in determining the contact angle within the hysteresis window in mixed cells. Given that the HF method is accurate in estimating curvature, it motivates us to propose a more rigorous method incorporating the HF technique to model contact line hysteresis on irregular boundaries. This method will feature precise calculations of the HF and accurately capture the contact angle in mixed cells.

In summary, this study proposes an accurate and robust numerical method for modeling contact line motion on complex geometries in 2D. The HF-based VOF approach transports the liquid/gas interface and calculates the surface tension force, while the EBM is utilized to approximate irregular boundaries. Under these circumstances, two sharp interfaces, Γ_l and Γ_s , coexist in the mixed cells, posing two main challenges in numerical simulations as discussed above. It should be noted that in such a hybrid framework, the available methods often fail to conserve mass due to the improper treatment of Γ_s during the geometrical advection of Γ_l in mixed cells. Therefore, we propose a conservative scheme to advance Γ_l in the presence of Γ_s . Additionally, a scheme based on volume conservation is implemented to capture the contact angle within the hysteresis window. To address the challenge of symmetry loss, a novel HF method is introduced for contact line cells, precisely imposing the contact angle boundary condition at Γ_s . Consequently, the manuscript is organized as follows: § 2 describes the governing equations and numerical algorithms for regular boundaries. § 3 details the advection scheme in mixed cells, featuring the main techniques developed in this study to address the aforementioned first challenge. § 4 introduces the novel method to define the HF in contact line cells and the method based on volume conservation to obtain the contact angle within the hysteresis window. Consequently, the process of the current method within one time step is summarized. § 5 presents rigorous validations against several benchmarks. More importantly, it shows how these numerical techniques improve computational accuracy and robustness compared to other schemes. Finally, the conclusions are drawn in § 6.

2. Governing equations, boundary conditions, and available VOF method

Fig. 1 illustrates the investigated problems, showcasing two sharp interfaces: the embedded solid surface (Γ_s) and the liquid/gas interface (Γ_l), which delineate the three domains ($\Omega_s \cup \Omega_l \cup \Omega_g$). On the regular and flat substrate, as shown in Fig. 1(a), the solid boundary aligns with the faces of the discretized cells, facilitating the direct imposition of boundary conditions on the face, possibly by placing one or two layers of ghost cells inside the solid substrate. Additionally, the VOF method is employed to track the liquid/gas interface, utilizing the volume fraction field, c , which indicates the proportion of the liquid phase (Ω_l) within the cell. It is important to note that for those interfacial cells containing the contact line, the value of c and the normal vector of the reconstructed interface should adhere to the contact angle (θ) condition at the substrate. On the irregular and complex substrate, as depicted in Fig. 1(b), the embedded boundary divides the discretized cells into solid and fluid parts, resulting in the formation of two sharp interfaces coexisting in mixed cells, e.g., (i, j) in the picture. The stationary solid surface Γ_s is reconstructed using a piecewise linear technique, employing c_s , which represents the proportion of the fluid part ($\Omega_l \cup \Omega_g$) within the cell intersected by Γ_s . Therefore we should note that there are two volume fractions in one discretized volume, namely $c = \Omega_l / (\Omega_l + \Omega_g + \Omega_s)$ and $c_s = (\Omega_l + \Omega_g) / (\Omega_l + \Omega_g + \Omega_s)$. Actually Ω_g is not necessarily a gas phase but could be another immiscible liquid, but we enforce it to be gas throughout the present study for simplicity. In the following context, a mixed cell refers to a discretized cell having an embedded boundary Γ_s , an interfacial cell contains a liquid/gas interface Γ_l , and a contact line cell comprises $\Gamma_s \cup \Gamma_l$ by forming a triple point P (see Fig. 1(b)). A fluid or solid cell means the cell is entirely immersed in the corresponding phase.

Before detailing the numerical methods developed in this study, we will first introduce the governing equations and boundary conditions, and then discuss the difficulties in solving these equations, particularly in imposing boundary conditions on irregular Γ_s .

2.1. Governing equations

For Newtonian, incompressible, viscous, multiphase flows, the momentum and mass conservation equations are expressed as follows:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \nabla \cdot (2\mu \mathbb{D}) + \sigma \kappa \delta_s \mathbf{n}_l + \rho \mathbf{g}, \quad (1)$$

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

where \mathbf{u} (m/s) is the velocity vector, and t (s), ρ (kg/m³), μ (Pa·s), p (Pa) and \mathbf{g} (m/s²) represent time, fluid density, viscosity, pressure and gravity acceleration vector, respectively. \mathbb{D} is the deformation rate tensor defined as $(\partial_i u_j + \partial_j u_i)/2$. The term $\sigma \kappa \delta_s \mathbf{n}_l$ corresponds to the surface tension term, where \mathbf{n}_l is the interface normal vector, κ is the interface curvature, and δ_s is the Dirac distribution function indicating that the force is concentrated on the interface.

The evolution of the liquid/gas interface Γ_l is governed by the following advection equation:

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = 0. \quad (3)$$

We will show later that this advection equation is solved by using a geometrical VOF method, and the interface is also reconstructed via a piecewise linear scheme. We should remind here that although the numerical techniques to solve Eq. (3) are already well-

established in the ordinary VOF framework, nevertheless, these techniques cannot be directly extended to the situations comprising irregular and complex boundaries. Thus proposing a robust and accurate VOF-based scheme to address this problem is one of our main concerns in the present study. Then after updating the volume fraction c , the density ρ and viscosity μ are computed based on c as follows:

$$\rho(c) = c\rho_l + (1 - c)\rho_g, \quad (4)$$

$$\mu(c) = c\mu_l + (1 - c)\mu_g. \quad (5)$$

2.2. Boundary conditions

In addition to the equations governing fluid flow and interface advancement, boundary conditions are one of the major difficulties to fully solve the problem. Specifically, given the focus of the present study on irregular and complex boundaries, the kinematic and dynamic conditions for fluid flow on the embedded surface Γ_s are of paramount importance. Firstly, velocity must adhere to the no-slip boundary conditions on complex boundaries, given that

$$\mathbf{u} = \mathbf{0}, \quad \forall \mathbf{x} \in \Gamma_s. \quad (6)$$

On the other hand, the wettability of the embedded boundary is also reflected by the value of the contact angle at the triple point, yielding

$$\mathbf{n}_s \cdot \mathbf{n}_l = -\cos\theta, \quad \forall \mathbf{x} \in (\Gamma_s \cap \Gamma_l), \quad (7)$$

where \mathbf{n}_s and \mathbf{n}_l are the unit vector normal to Γ_s and Γ_l , which point from 1 to 0 (c_s or c), respectively. Note that this prescribed contact angle is an implicit boundary condition, which actually controls the motion of the liquid interface Γ_l . In other words, the volume fraction c in the vicinity of Γ_s is not solely dependent on the solution of Eq. (3) but also relies on the value of θ , which is not necessarily a constant but probably involves hysteresis characteristics or varies with the local fluid velocity.

Implementing these boundary conditions, i.e., Eqs. (6) and (7), is straightforward in numerical simulations on regular and flat substrates, and we will not elaborate further on this point. However, imposing these boundary conditions on irregular and complex substrates in a Cartesian system becomes non-trivial, as the embedded boundaries may intersect the discretized cells. Thus, specialized interpolation schemes that incorporate the prescribed boundary conditions should be devised near these incomplete cells to maintain the conservative properties of the governing equations. As introduced in § 1, numerous methods have been proposed to accurately impose velocity conditions on Γ_s , such as the IBM-type [12,13] and the EBM-type [14,15] methods, which correspond to the finite difference method and the finite volume method, respectively. However, accurately applying the contact angle condition Eq. (7) on irregular and complex boundaries remains an open question. The coexistence of Γ_s and Γ_l greatly complicates the issue, as maintaining robustness and conservativeness often conflict in numerical simulations [20,17].

In summary, the numerical methods proposed in this study primarily address two aspects: solving the volume of fluid equation Eq. (3) and imposing the contact angle condition Eq. (7) in the presence of irregular and complex boundary Γ_s . Additionally, the open-source software *Basilisk*, developed by S. Popinet [26], handles the remaining governing equations (Eq. (1)) and velocity boundary conditions (Eq. (6)), and has been extensively used in detailed explorations of interfacial flows over the past decade. *Basilisk* employs Cartesian grids with collocated discretization of velocity and scalar fields, solving the 3D Navier-Stokes equations using a second-order approximate projection method [27,28] to ensure divergence-free velocity fields. Interfaces are tracked and geometrically reconstructed using a VOF approach. An accurate well-balanced height function method calculates interface curvature and enforces the contact angle condition, while the continuum surface force (CSF) model computes surface tension [29] as $\mathbf{F}_s = \sigma\kappa\delta_s\mathbf{n}_l = \sigma\kappa\nabla c$. In complex boundary scenarios, *Basilisk* incorporates an EBM method [14] to enforce the no-slip boundary condition by incorporating the flow flux across Γ_s into the discretized momentum equation. This finite volume approach conserves momentum effectively and maintains the sharpness of the embedded boundary. It should be noted that in *Basilisk*, the embedded boundaries, always maintained stationary in numerical simulations, are reconstructed via a piecewise linear technique before starting the simulations, ensuring that the associated geometrical information of Γ_s , e.g., \mathbf{n}_s and c_s , is prior known. Currently in *Basilisk*, there is no fully developed module to address the integration of VOF and EBM to accurately solve the moving contact line problem on complex geometrical boundaries.

Consequently, next we will briefly review the VOF advection scheme and the HF method implemented in the original *Basilisk*, they are for curvature estimation and contact angle condition on a flat substrate. Following this, our methods will be introduced and elaborated to discuss the extension of this VOF-based framework to accommodate 2D irregular and complex boundaries.

2.3. VOF advection

The update of the fluid volume fraction c in discretized cells is governed by Eq. (3). In *Basilisk*, this equation is solved using a direction-split advection scheme [26] to transport c sequentially in the x - and y -directions. In such a scheme, the compression or expansion of the cell due to the divergent one-dimensional velocity field must be considered. Therefore, Eq. (3) is reformulated as follows:

$$\frac{\partial c}{\partial t} + \nabla \cdot (c\mathbf{u}) - c\nabla \cdot \mathbf{u} = 0, \quad (8)$$

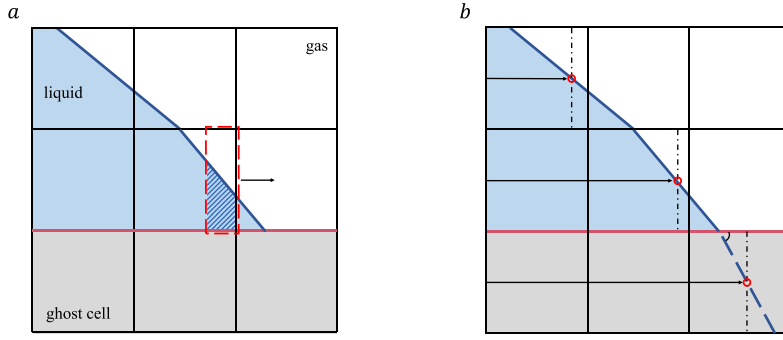


Fig. 2. (a) Schematic for the geometrical advection method on regular and flat substrates. (b) Definition of the height function on regular and flat substrates for the 2D case and calculation of the height function in ghost cells.

and the application of the Gaussian divergence theorem leads the volume integral of Eq. (8) to the following expression:

$$\frac{\partial}{\partial t} \int_V c dV + \oint_{\partial V} (c \mathbf{u} \cdot \mathbf{n}) d\partial V - c \oint_{\partial V} \mathbf{u} \cdot \mathbf{n} d\partial V = 0, \quad (9)$$

where V and ∂V denote the volume and surface integrals of the discretized cell, respectively. In 2D computations, the discretization of Eq. (9) is carried out into two steps,

$$c^* = c^{n-\frac{1}{2}} + \frac{dt}{\Delta} \left((F_{i-\frac{1}{2},j}^{n-\frac{1}{2}} - F_{i+\frac{1}{2},j}^{n-\frac{1}{2}}) - c_c (u_{i-\frac{1}{2},j}^{n-\frac{1}{2}} - u_{i+\frac{1}{2},j}^{n-\frac{1}{2}}) \right), \quad (10)$$

$$c^{n+\frac{1}{2}} = c^* + \frac{dt}{\Delta} \left((F_{i,j-\frac{1}{2}}^* - F_{i,j+\frac{1}{2}}^*) - c_c (u_{i,j-\frac{1}{2}}^{n-\frac{1}{2}} - u_{i,j+\frac{1}{2}}^{n-\frac{1}{2}}) \right), \quad (11)$$

where $(i \pm \frac{1}{2}, j)$ and $(i, j \pm \frac{1}{2})$ are the cell faces normal to the x - and y -directions, respectively and the superscript $*$ represents an intermediate value for volume fractions and fluxes calculated based on c^* . Besides, $(n \pm \frac{1}{2})$ are the corresponding time steps, Δ represents the grid size, dt is the time step, and $F = (c \mathbf{u}_f \cdot \mathbf{n}_f)$ is the flux of volume fraction across the corresponding cell face, with $(\mathbf{u}_f \cdot \mathbf{n}_f)$ denoting the normal component of the face centered velocity. Then the computation of the volume flux through the cell faces, i.e., $F_{i,j \pm \frac{1}{2}}$ and $F_{i \pm \frac{1}{2},j}$, is the key to design numerical schemes to solve Eqs. (10) and (11). *Basilisk* employs a geometrical scheme to estimate F that the sharpness of the interface is thus maintained. Fig. 2(a) sketches the estimation of $F_{i+\frac{1}{2},0}$ across the cell face $(i + \frac{1}{2}, 0)$ in one time step dt . The total volume which will be fluxed to the right-hand neighbor cell $(i + 1, 0)$ is delimited with a dashed line box, which has a width of $un = u_{i+\frac{1}{2},0} \cdot dt$ and length of Δ . Then $F_{i+\frac{1}{2},0}$ can be estimated as

$$F_{i+\frac{1}{2},0} = \frac{\mathcal{A} \cdot u_{i+\frac{1}{2},0}}{\Delta \cdot un}, \quad (12)$$

where \mathcal{A} is marked by the shaded region in the picture, representing the area occupied by liquid ($c = 1$) in the dashed box. The value of \mathcal{A} can be calculated via the geometrical information of the interface, which is already reconstructed via a piecewise linear scheme in cell $(i, 0)$. Then the calculations of $F_{i-\frac{1}{2},0}$ and $F_{i,\pm\frac{1}{2}}$ through other cell faces can be obtained in the same way. c_c is proposed by Weymouth et al. [30], which activates the cell dilation term when the volume fraction is large enough to ensure that the fluid will neither overflow nor over-empty during the two-step update of the volume fraction. So far Eqs. (10) and (11) are solved to update $c_{i,0}$ in the next time step $(n + \frac{1}{2})$.

2.4. Height function method

It is well-established that accurately computing the surface tension force is crucial for simulating interfacial flows. This issue is particularly critical for contact line problems because the capillary effects may become dominant. An accurate surface tension model, as described by $\sigma \kappa \delta_s \mathbf{n}_l$ in Eq. (1), highly relies on the estimation of the curvature κ and the normal vector \mathbf{n}_l of the interface Γ_l . We will introduce how they are calculated in the original *Basilisk*, respectively. The classic height function approach is employed in *Basilisk* to calculate the curvature κ . Fig. 2(b) illustrates the adopted height function (HF) approach, which is evaluated by summing the volume fractions along one principal direction depending on the slope of the interface. In the figure, the height function for cells in row $i = 0$, adjacent to the bottom boundary, is computed along the x -direction. The value of h_0 yields

$$h_0 = \sum c_{i,0} \Delta, \quad (13)$$

where \sum is always performed on a maximum of 9 neighboring cells in that row, namely from $(i-4, 0)$ to $(i+4, 0)$ in Fig. 2(b) (see [26] for details), unless the extremely far cells are void in the computational domain. Then the curvature κ of the local Γ_l in cell $(i, 0)$ can be calculated as [26]:

$$\kappa = \frac{h_{yy}}{(1 + h_y^2)^{3/2}}, \quad (14)$$

where h_y denotes the y -component gradient of the height function, estimated as $h_y = (h_1 - h_{-1})/2\Delta$. In addition, the second-order derivative is $h_{yy} = (h_1 - 2h_0 + h_{-1})/\Delta^2$. Then the computations of h_y and h_{yy} require the unknown value of h_{-1} , which is actually the ghost value inside the solid cells. In *Basilisk*, the estimation of h_{-1} is achieved based on the geometrical constraint implied by the contact angle condition Eq. (15), given that [22]

$$h_{-1} = h_0 + \frac{\Delta}{\tan(\theta)}. \quad (15)$$

Because the regular and flat boundary aligns with the faces of the discretized cells, the height function value h_{-1} can be easily settled in ghost cells according to Eq. (15). This technique actually extrapolates the height function into the solid boundary with second-order accuracy, which ensures a converged curvature estimation near the contact line, as discussed by Afkhami and Bussmann [22].

Further discussions and clarifications are warranted on the estimation of the normal vector \mathbf{n}_l , which needs to be determined in two key areas in numerical computations. One is for evaluating the surface tension force, as denoted by $\sigma\kappa\delta_s\mathbf{n}_l$, and the other is for the implicit treatment of the contact angle condition on the solid substrate, given by $\mathbf{n}_s \cdot \mathbf{n}_l = -\cos\theta$ with $\mathbf{n}_s = (0, -1)$ representing a horizontal and flat boundary. To estimate $\mathbf{F}_s = \sigma\kappa\delta_s\mathbf{n}_l$, the normal vector is computed as $\mathbf{n}_l = \mathbf{n}_{l,c} = \nabla c/|\nabla c|$ using a 3×3 stencil in *Basilisk*. This discretization form maintains consistency with that of the pressure gradient ∇p , enabling the construction of the CSF model and the balanced-force model [31] to reduce spurious flows near the interface. Therefore, even in contact line cells where the normal vector can be obtained via either a geometrical scheme $\mathbf{n}_s \cdot \mathbf{n}_{l,\theta} = -\cos\theta$ or a height function method (see below), $\mathbf{n}_{l,c} = \nabla c/|\nabla c|$ is still applied to calculate the normal vector for the surface tension force, ensuring conformity with the balanced-force model. Additionally, ghost cells immersed in the solid substrate are used to obtain the ghost values of c , allowing for the construction of a 3×3 stencil to estimate ∇c in interfacial cells near the boundary. In contrast, when applying contact angle conditions at the boundary, *Basilisk* uses a height function (HF) method to estimate the normal vector in those contact line cells, formulated as:

$$\mathbf{n}_{l,h} = (1, -h_y), \quad (16)$$

where height function values in ghost cells are required to construct a 3×3 stencil. This HF-based evaluation scheme ($\mathbf{n}_{l,h}$) is considered more accurate than the geometrical scheme ($\mathbf{n}_{l,\theta}$) and the volume fraction scheme ($\mathbf{n}_{l,c}$). After acquiring \mathbf{n}_l and c in interfacial cells, a geometrical scheme is applied to reconstruct the interface. Consequently, the ghost values of the volume fraction and the height function are crucial for estimating the curvature κ and the normal vector $\mathbf{n}_{l,c}$ or $\mathbf{n}_{l,h}$. However, as discussed later in § 4.1, the “ghost cell” technique becomes more challenging in the presence of embedded boundaries, leading to difficulties in accurately imposing the contact angle on arbitrarily shaped substrates.

In summary, these well-established VOF and HF approaches make *Basilisk* a very efficient and accurate solver for the simulation of the contact line motion on regular and flat boundaries. Nevertheless when comprising complex geometrical boundaries, the direct application of the advection scheme (see § 2.3) may lead to significant mass errors within mixed cells, and thus the failure of local volume conservation will degenerate the numerical accuracy and reliability, which is verified by the numerical simulations provided by Patel et al. [17]. Besides, the aforementioned HF technique (see § 2.4) is no longer applicable in mixed cells either. The improperly imposed contact angle condition, via the HF technique, will result in significantly non-physical pinning of contact lines. Consequently, in the following sections, we will design an exact geometrical advection scheme suitable for mixed cells, while a modified HF method is also proposed to precisely impose the contact angle condition there.

3. VOF advection in mixed cells

In this section, we first develop a precise advection algorithm to solve Eq. (3) in mixed cells, which contain both Γ_s and Γ_l . Our primary objective is to conserve the local volume during the advancing of Γ_l .

3.1. Liquid/gas interface reconstruction within mixed cells

In the presence of the embedded boundary Γ_s , the interfacial cells may not be entire but become incomplete, e.g., cell (i, j) in Fig. 1(b), and thus the above mentioned interface reconstruction scheme should be adapted to such “incomplete” or “unstructured” cells first, because the estimation of the volume fraction flux F in Eqs. (10) and (11) highly relies on the geometrical information of the interface in the mixed cell.

Fig. 3 (a) illustrates how to reconstruct the interface (blue line) in the mixed cell, which is cut by the solid embedded boundary Γ_s (red line) that the fluid part is now actually a convex pentagonal shape. In a piecewise linear VOF scheme the interface is represented in the formulation of

$$\mathbf{n}_l \cdot \mathbf{x} = \alpha, \quad (17)$$

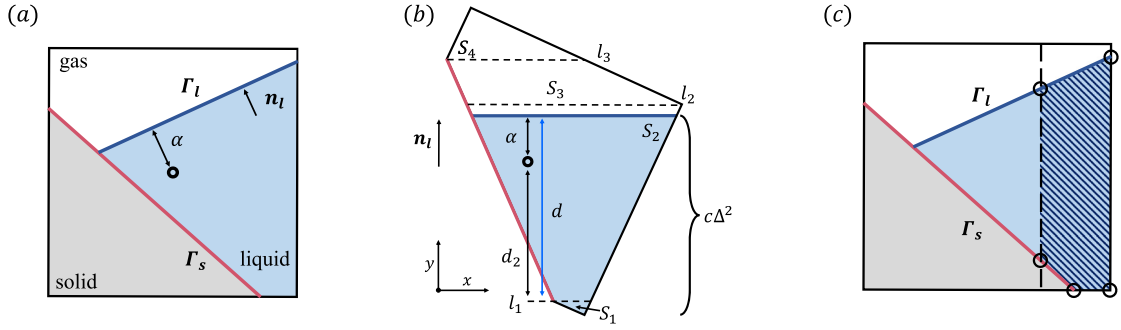


Fig. 3. (a), (b): Sketch of liquid/gas interface reconstruction in the mixed cell. l_i is the horizontal line drawn through each vertex. S_i represents the area of interval divided by l_i and l_{i-1} . d_i is the distance from l_{i-1} to the cell center. When the liquid/gas interface is located in the i -th interval, d denotes the distance from l_{i-1} to the liquid/gas interface, while α is the distance from the liquid/gas interface to the cell center. (c): Calculation of the advection area \mathcal{A} in the mixed cell.

Algorithm 1: Interface reconstruction within the mixed cell.

Input: c, \mathbf{n}_l

Output: α

1. reconstruct the embedded solid boundary Γ_s (the red solid line in Fig. 3(a)) by using a piecewise linear technique prior to starting the simulation;
 2. rotate the cell to make the normal vector \mathbf{n}_l align with the y-axis and the following steps are all shown in Fig. 3(b);
 3. divide the fluid part into smaller intervals by horizontal lines across each vertex, given that l_i ($i = 1, 2, 3, 4$);
 4. calculate the area of each interval as S_i ($i = 1, 2, 3, 4$);
 5. identify the i -th interval where it locates the liquid/gas interface, i.e., $\sum_{i=1}^{i-1} S_i < c\Delta^2 < \sum_{i=1}^i S_i$;
 6. calculate the interface height d in the i -th interval, which contains a Volume-Of-Fluid ($c\Delta^2 - \sum_{i=1}^{i-1} S_i$);
 7. compute α as $\alpha = d - d_2$, where d_2 denotes the distance from the cell centroid to l_1 .
-

where \mathbf{n}_l is the unit normal vector of the interface, and \mathbf{x} is the position vector. Given that \mathbf{n}_l and the volume fraction c are prior known, α can be uniquely determined by ensuring that the Volume-Of-Fluid contained in the cell and lying below the interface is equal to $c\Delta^2$. In our method, α is actually the distance from the cell centroid to the interface, normalized by the cell size Δ . If the cell is a complete square in the absence of Γ_s , then α can be calculated in a simple analytical way [8]. However, in the presence of Γ_s as shown in Fig. 3 (a), the estimation of α is non-trivial because c now denotes the Volume-Of-Fluid contained in this convex pentagonal cell. In order to estimate α in such an irregular cell, we implement a very similar methodology initially proposed by Dyadechko et al. [32] and further developed by others [33–35], while their method is also constructed in the VOF framework but designed for arbitrarily unstructured meshes. We make an extension of that method to the Cartesian cells cut by complex boundaries, referred to as a kind of unstructured-like cells.

The procedure is described in Fig. 3 (b) that we also briefly introduce it in Algorithm 1 while more details could be found in the above mentioned literature. Note that to apply this method to other unstructured mixed cells, e.g., triangles and convex quadrangles, the extension of the procedure written in Algorithm 1 is also straightforward. The first step involves identifying the fluid region within the mixed cell, as illustrated in Fig. 3(a). The subsequent steps are depicted in Fig. 3(b). The second step is to facilitate later the calculation of height. In the third step, the polygonal fluid region is subdivided into smaller intervals, such as triangles and trapezoids. The fourth step involves computing the area of each interval S_i . In the fifth step, the Volume-Of-Fluid value ($c\Delta^2$) is compared with the cumulative area of the intervals $\sum S_i$ to determine the interval where the liquid/gas interface resides. Then a quadratic equation is established between the interface height (d) and the remaining volume ($c\Delta^2 - \sum_{i=1}^{i-1} S_i$) within that interval. In the sixth step, solving this quadratic equation yields the interface height (d) in the identified interval. Finally, in the seventh step, the distance (α) from the liquid/gas interface to the cell center is calculated using d . At the end of this process, the precise liquid/gas interface within the polygonal fluid region is reconstructed, as shown in Fig. 3(a) and (b). In addition, the above procedures can also be extended to the 3D case to reconstruct the interface in a 3D mixed cell. Nevertheless, the 3D scenario poses extra hurdles, notably when calculating sub-interval volumes and developing a cubic equation relating the volume fraction and interface height. Overcoming these difficulties is a crucial focus for our upcoming research.

After obtaining the interface position with Algorithm 1, another problem remaining to be solved is the calculation of the geometrically advected liquid volume \mathcal{A} in a mixed cell, exemplified in Fig. 3(c) with the shaded region. This can be considered the inverse problem of interface reconstruction, where \mathbf{n}_l and α are known, but \mathcal{A} must be determined. To solve this, we find all the vertices (x_n, y_n) of the convex polygon enclosing the liquid volume \mathcal{A} and arrange them counterclockwise based on their angles relative to the center of the polygon, as shown by those open circles portrayed in Fig. 3(c). Then, the advected liquid volume \mathcal{A} can be easily calculated as follows:

$$\mathcal{A} = \frac{1}{2} \sum_n^N x_n(y_{n+1} - y_{n-1}). \quad (18)$$

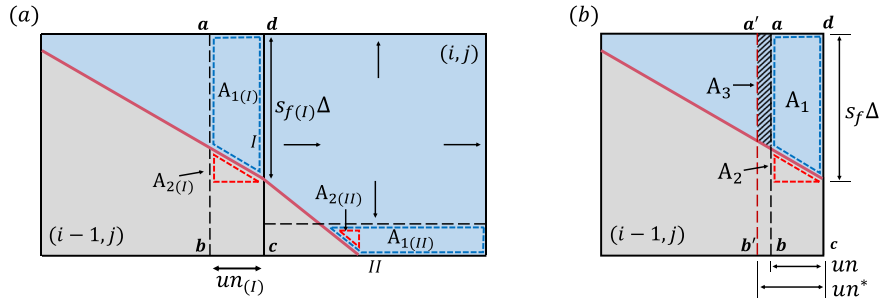


Fig. 4. Sketch of the VOF geometrical advection scheme in mixed cells. (a) s_f denotes the face fraction occupied by the fluid, Δ represents the mesh size, and un refers to the width of the advection interval $abcd$. A_1 and A_2 denote the areas of the trapezoid and triangle, respectively, while I and II correspond to the two cell faces intersected by the embedded solid boundaries. (b) un^* represents the corrected width of the advection interval $a'b'cd$, and A_3 represents the area of the shaded trapezoid.

3.2. Geometrical advection of liquid volume fraction in mixed cells

In this section, we discuss how to construct the advection scheme for solving Eq. (3) in a mixed cell. Based on the discretized forms given in Eqs. (10) and (11), the crucial step lies in how to estimate the volume fraction flux F on incomplete cell faces. For an interfacial cell that does not contain an embedded solid boundary, Eq. (12) can be used to accurately estimate F . However, in a mixed cell, directly applying Eq. (12) with \mathcal{A} estimated by Eq. (18), may not ensure local volume conservation. Next, we provide a detailed explanation of the underlying reasons.

In the geometrical VOF method, the flux of the volume fraction through the cell face of a mixed cell is influenced by three factors: the volume fraction in the mixed cell, the velocity at the cell face and the embedded solid boundary in the mixed cell. The existence of the embedded solid boundary in the mixed cell (the third factor) can be shown to have a significant effect on the flux calculation. To simplify the explanation, we consider a mixed cell, denoted as (i, j) , filled with liquid ($c = c_s$), as illustrated in Fig. 4. This mixed cell is also surrounded by cells filled with liquid. The impact of the volume fraction (the first factor) on the volume fraction flux in the mixed cell can be neglected, since the fluid that is flowed to neighboring cells consists only of liquid. Furthermore, since the neighboring cells of the cell (i, j) are all filled with liquid at $t = n - 1/2$, it can be assumed that the volume fraction of the cell (i, j) remains at $c = c_s$ at $t = n + 1/2$. For the cell face I , we calculate the width of the advection interval $abcd$, $un_{(I)} = u_{f(I)} \cdot dt$, before advecting the volume fraction. We then determine the volume fraction area $\mathcal{A}_{(I)}$ in $abcd$, where $\mathcal{A}_{(I)} = A_{1(I)} = un_{(I)} \cdot s_{f(I)} \Delta - A_{2(I)} = u_{f(I)} s_{f(I)} \cdot (dt \cdot \Delta) - A_{2(I)}$. For cell face II , $\mathcal{A}_{(II)} = A_{1(II)} = u_{f(II)} s_{f(II)} \cdot (dt \cdot \Delta) + A_{2(II)}$. The advection area for the other two cell faces is then calculated, which gives

$$(c^{n+1/2} - c^{n-1/2})\Delta^2 = \nabla \cdot (u_f s_f) \cdot dt \Delta^2 - A_{2(I)} - A_{2(II)}, \quad (19)$$

where $\nabla \cdot (u_f s_f) = 0$, due to the divergence-free velocity. The magnitude of A_2 (containing $A_{2(I)}$ and $A_{2(II)}$) is determined by the shape of the embedded solid boundary within the mixed cell and the velocity on the cell face. In the case illustrated in Fig. 4(a), apparently $c^{n+1/2} < c^{n-1/2}$, the cell (i, j) will become an interfacial cell ($c < c_s$) at the next time step. If we incorporate the symbol for A_2 within the variable itself, we have $(c^{n+1/2} - c^{n-1/2})\Delta^2 = A_{2(I)} + A_{2(II)}$, which implies $c^{n+1/2} = c_s + (A_{2(I)} + A_{2(II)})/\Delta^2$. Regardless of the signs of $A_{2(I)}$ and $A_{2(II)}$, they cannot always cancel each other out. Therefore, at $t = n + 1/2$, the mixed cell will become an interfacial cell ($c^{n+1/2} < c_s$) or an over-filled cell ($c^{n+1/2} > c_s$). This is the primary cause of the failure to conserve local volumes.

Although we have discussed a cell filled with liquid, it is important to note that local volume conservation failure also occurs for cells that contain liquid/gas interfaces due to the presence of A_2 . The underlying reason is the same as that in the process analyzed above, with the difference being in the calculation of the advection area. For such cells, the advection area is given by $\mathcal{A} = u_f s_f \cdot (dt \cdot \Delta) + A_2 - A_{gas}$. We will not describe in detail the interfacial cell here.

To address this issue, we must re-calculate the real liquid volume \mathcal{A} flowing into the mixed cell (i, j) by considering the computational error in the presence of Γ_s . A modified geometrical scheme should be constructed instead of using Eq. (12), and we still consider the advection through face I in Fig. 4(a) for illustration. Herein, we replot it in Fig. 4(b) and omit the subscript I for simplicity. Clearly, the original scheme, i.e., Eq. (12), calculates the liquid volume \mathcal{A} in the geometrical advection region $abcd$. However, to conserve the local volume, the liquid volume calculated via the reconstructed geometrical scheme must equal the volume $(A_1 + A_2)$ advected through face I . Clearly, A_2 contained in the solid part should be compensated by adding new liquid volume in this reconstructed geometrical scheme. An intuitive approach is to enlarge the geometrical advection region for volume compensation as shown in Fig. 4(b), where a new geometrical advection region $a'b'cd$ is created by extending the width from un to un^* , ensuring the volume of A_3 equals A_2 . The key to preserving volume conservation, i.e., maintaining $A_3 = A_2$, is to estimate the value of un^* accurately. This is achieved geometrically by extending the geometrical VOF reconstruction scheme to this non-interfacial cell to obtain the position of $a'b'$. Specifically, we take $a'b'$ as a virtual liquid/gas interface vertically aligned when considering the advection flux through face I , and its position denoted by un^* herein is what we desire. Thus, the normal vector of this virtual interface is $\mathbf{n}' = (-1, 0)$ with the covering volume fraction of $c = (A_1 + A_3)/\Delta^2 = (un \cdot s_f \Delta)/\Delta^2$. The geometrical information of $a'b'$ is then calculated using Algorithm 1:

$$\alpha = \text{Algorithm 1} \left(c = (un \cdot s_f \Delta) / \Delta^2, \mathbf{n}' = (-1, 0) \right), \quad (20)$$

which again refers to the distance from the cell centroid to $a'b'$, and thus $un^* = (1/2 + \alpha)$ is finally obtained.

Note that although we derive Eq. (20) using the non-interfacial mixed cell in Fig. 4, in the present method, mixed cells containing the liquid/gas interface are handled in the same way. This method recalculates the advected volume \mathcal{A} in Eq. (12) and the flux F in Eqs. (10) and (11). The estimation of F is reformulated as follows:

$$F = c_f \cdot (\mathbf{u}_f \cdot \mathbf{n}_f) = \frac{\mathcal{A}(un^*)}{\Delta \cdot un^*} \cdot u_f^*, \quad (21)$$

where the face velocity $u_f^* = un^* / dt$ is also adjusted using un^* instead of un to match the new advection region, and $\mathcal{A}(un^*)$ represents the geometrically advected liquid volume estimated by Eq. (18) in the adjusted geometrical advection region with a width of un^* . Moreover, the quantity c_c in the dilation term of Eqs. (10) and (11) should also be redefined to account for the presence of the solid phase in a mixed cell, namely,

$$c_c = \begin{cases} 1, & c > c_s/2 \\ 0, & c \leq c_s/2 \end{cases}. \quad (22)$$

Thus, the dilation term is activated when the volume fraction c is larger than $c_s/2$, ensuring that the mixed cell is not over-filled or over-emptied during the direction-split advection.

In summary, combining Eqs. (18), (20) and (21) accurately determines the volume flux F flowing into the mixed cell (i, j) through face $(i - \frac{1}{2}, j)$. Likewise, the geometrical reconstruction of other advection regions corresponding to different cell faces can be evaluated. This method is designed to conserve the fluid volume within the mixed cell during the advection. To verify this conservation property, we revisit the volume fraction of liquid updated in cell (i, j) (see again Fig. 4(a)). Taking the volume fraction advection in the x -direction as an example, substituting Eqs. (21) and (22) into Eq. (10) leads to:

$$c^* - c^{n-\frac{1}{2}} = \frac{dt}{\Delta} \left(\frac{\mathcal{A}_{i-\frac{1}{2},j}}{\Delta \cdot dt} - \frac{\mathcal{A}_{i+\frac{1}{2},j}}{\Delta \cdot dt} - c_c (u_{i-\frac{1}{2},j} s_{i-\frac{1}{2},j} - u_{i+\frac{1}{2},j} s_{i+\frac{1}{2},j}) \right), \quad (23)$$

where, for simplicity, the time level of the quantities on the right-hand side is omitted. Given that the liquid volume in the adjusted geometrical advection region is equal to the actual advected fluid volume, i.e., $\mathcal{A}_{i-\frac{1}{2},j} = u_{i-\frac{1}{2},j} dt \cdot s_{i-\frac{1}{2},j} \Delta$, and assuming $c_c = 1$ for the mixed cell (i, j) filled with liquid, the right-hand side of Eq. (23) reduces to zero. The y -directional advection for the volume fraction in the mixed cell is similar to that in the x -direction. Consequently, the volume fraction in cell (i, j) remains constant in the next time step, thereby ensuring the conservation of local volume.

3.3. Time step restricted by small mixed cells

When solving the advection equations (10) and (11), the time step dt is usually restricted by the Courant-Friedrichs-Lewy (CFL) condition, given that:

$$\frac{dt \cdot u_f s_f}{\Delta} \leq c_s, \quad \forall \text{ cell face}. \quad (24)$$

It is clear that in a mixed cell containing an embedded solid boundary, i.e., $c_s < 1$, the time step dt may be strongly restricted if the fluid volume fraction is much smaller than 1, i.e., $c_s \ll 1$. This restriction is known as the “small-cell” CFL restriction in those cut-cell type methods [36–38]. To address this issue in solving the advection term appearing in the momentum or any other convection-diffusion equation, *Basilisk* redistributes the overflow flux in a small mixed cell into the neighboring cells, thereby relaxing the “small-cell” CFL restriction significantly. However, such a redistribution technique is not applicable for the VOF advection equation due to its reliance on the geometrical information of Γ_l . Hence, a new method must be developed to solve this problem. A straightforward approach is to artificially enforce the fluid volume fraction to be 0 directly if we identify $c_s < 0.01$ in those mixed “small-cells”. In Fig. 5(a), (i, j) represents such “small-cell” having a volume fraction $c_s(i, j) < 0.01$. However, enforcing $c_s(i, j) = 0$ requires the embedded boundaries to extend through cell vertices, labeled by the blue open circle in Fig. 5(a), and this is technically not permitted in the current *Basilisk* framework. Alternatively, we show our solution strategy in Fig. 5(b) which reveals that besides enforcing $c_s(i, j) = 0$, the next step involves setting the fluid volume fraction in the diagonally neighboring cell to be $c_s(i+1, j+1) = 0.99$. Then such modification on $c_s(i+1, j+1)$ actually has impacts on neighboring cells, for instance, \mathbf{n}_s in cells $(i, j+1)$ and $(i+1, j)$ depend on the value of $c_s(i+1, j+1)$ because \mathbf{n}_s in the mixed cell is estimated using the Mixed-Young-Centered (MYC) scheme [9], which highly relies on the volume fraction distribution in a 3×3 cell stencil. Thus after adjusting the value of $c_s(i+1, j+1)$ from 1 to 0.99, the reconstructed embedded boundaries Γ_s reshape to the dashed red line in Fig. 5(b). Consequently, the reconstructed solid boundaries may not be continuous on face $(i + \frac{1}{2}, j)$, as indicated by the intersections N_1 and N_2 from cell $(i, j+1)$ and $(i+1, j+1)$, respectively. To address this, a secondary reconstruction is performed via a simple averaging scheme to find the midpoint M between N_1 and N_2 , ensuring continuity of the embedded boundary represented by the red solid line plot in Fig. 5(c). Note that the adjustment of c_s and the reconstruction of Γ_s are only performed at the first time step of simulation, i.e., $t = 0$, then the embedded solid together with its surrounding meshes will remain unchanged during the simulation. This procedure is summarized briefly in Algorithm 2.

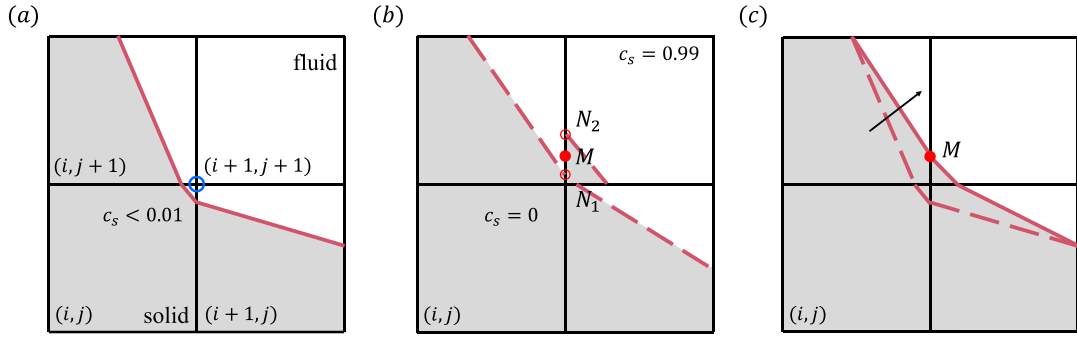


Fig. 5. Sketch of the re-initialization of embedded solid boundaries after clearing small mixed cells. The red solid and dashed lines represent the solid boundaries, while the blue open circle indicates the cell vertex. N_1 and N_2 denote the intersections of the reconstructed solid boundaries with the cell faces after the redistribution of fluid volume c_s . M represents the midpoint between N_1 and N_2 .

Algorithm 2: Clearance of small mixed cells illustrated in Fig. 5.

if $c_s < 0.01$ in cell (i, j) then

1. set $c_s = 0$ in cell (i, j) and $c_s = 0.99$ in cell $(i+1, j+1)$ (Fig. 5(a));
2. reconstruct the embedded solid boundary from the updated c_s (red dashed lines in Fig. 5(b));
3. find the intersections of the embedded boundaries with cell faces (e.g., N_1 and N_2 on face $(i+\frac{1}{2}, j+\frac{1}{2})$);
4. update face fraction s_f based on the midpoints of the above intersections (e.g., M lying halfway between N_1 and N_2);
5. reconstruct the embedded boundaries from the updated s_f and recalculate c_s (Fig. 5(c)).

end

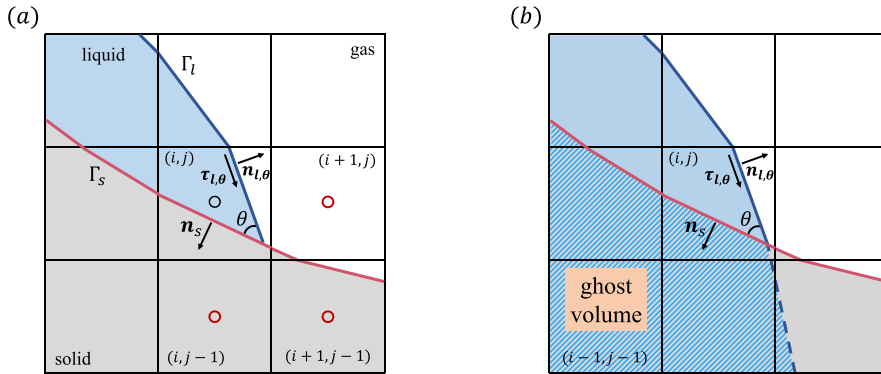


Fig. 6. (a) Identification of contact line cells. $\tau_{l,\theta}$ represents the tangential vector of the liquid/gas interface. (b) Sketch of extending the liquid/gas interface into the solid region. The blue shaded region is the ghost fluid volume, which is denoted by c_e .

3.4. Surface tension computation close to the embedded boundaries

Up to now, the liquid volume fraction c is obtained at time step $(n + \frac{1}{2})$ by solving Eqs. (10) and (11), while its conservative property is preserved using the above-implemented numerical method. Subsequently, Eqs. (4) and (5) facilitate the update of the physical properties necessary for solving the momentum equation (Eq. (1)) as the next step. However, the discretization of the surface tension term $\sigma \kappa \delta_s \mathbf{n}_l$ in Eq. (1) remains problematic, especially since the normal vector in this term is calculated as $\mathbf{n}_l = \mathbf{n}_{l,c} = \nabla c / |\nabla c|$ to match the gradient of the pressure ∇p , as described in § 2.4. Estimating this near the embedded solid boundary fails due to a lack of volume fraction information in solid cells ($c = \text{Null}$ if $c_s = 0$). Consequently, the 3×3 stencil for the gradient calculation of ∇c in the mixed cell cannot be constructed, which is exacerbated in mixed cells containing the contact line (see, for instance, (i, j) in Fig. 1(b)), because $\mathbf{n}_{l,c}$ should also reflect the contact angle θ there. As a solution, a straightforward approach is to extend the liquid/gas interface Γ_l from the contact line cells into the solid cells. The ghost volume of the liquid in the solid cells, denoted by c_e hereafter, can then be estimated geometrically. This requires identifying the contact line cells and designing a numerical scheme for interface extension as a first step.

A contact line cell must contain solid, liquid, and gas phases simultaneously; however, cells containing these three phases (i.e., satisfying $0 < c_s < 1$ and $0 < c < c_s$) are not always contact line cells. For instance, Fig. 6(a) illustrates that cell (i, j) is a contact line cell, but cell $(i-1, j+1)$ is not. Under such circumstances, the liquid volume fractions in neighboring cells are required to identify the true contact line cells. Focusing on cell (i, j) in Fig. 6(a), we keep in mind that the contact angle θ between Γ_l and Γ_s and the

Algorithm 3: Estimation of $\mathbf{n}_{l,\theta}$ in contact line cells.

Input: c, \mathbf{n}_s, θ
Output: $\mathbf{n}_{l,\theta}$
if $(\nabla c \times \mathbf{n}_s) \cdot \mathbf{e}_z > 0$, with $\mathbf{e}_z = (0, 0, 1)$ **then**
 $\mathbf{n}_{l,\theta} = - \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \cdot \mathbf{n}_s$
else
 $\mathbf{n}_{l,\theta} = - \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \cdot \mathbf{n}_s$
end

Algorithm 4: Identification of the contact line cell illustrated in Fig. 6 (a).

Input: $c, c_s, \tau_{l,\theta}$
Output: *true or false*
if $0 < c_s < 1$ & $0 < c < c_s$ **then**
 Find the neighboring cells located in the $\tau_{l,\theta}$ direction, i.e., cells $(i+1, j)$, $(i, j-1)$ and $(i+1, j-1)$
 if $c = 0$ **or** $c = c_s$ for all these neighboring and diagonal cells **then**
 return *true*.
 end
end

normal vector \mathbf{n}_s of the embedded solid are known values. Geometrically, as a first step, rotating \mathbf{n}_s by an angle of $(\pi - \theta)$ in the counter-clockwise direction obtains \mathbf{n}_l . The direction of rotation can be determined by the sign estimation of $\nabla c \times \mathbf{n}_s$. This geometrical rotation scheme is detailed in Algorithm 3, which is inspired by the work in <http://basilisk.fr/sandbox/popinet/contact/>, and we denote this normal direction of Γ_l as $\mathbf{n}_{l,\theta}$. In the second step, we compute the tangential vector $\tau_{l,\theta}$ of Γ_l by rotating $\mathbf{n}_{l,\theta}$ by 90° in the clockwise direction. The obtained vector $\tau_{l,\theta}$ points to two neighboring cells $(i+1, j)$ and $(i, j-1)$, and also identifies one diagonal cell $(i+1, j-1)$ between them in Fig. 6(a). Since none of these three cells contain the liquid/gas interface Γ_l , cell (i, j) can be tagged as a contact line cell. The procedure is summarized in Algorithm 4 and can be applied to other mixed cells for contact line identification.

Once the contact line cells are identified and the associated $\mathbf{n}_{l,\theta}$ is obtained, the next step is to reconstruct the contained liquid/gas interface into a linear segment using Algorithm 1. Subsequently, we extend it into the solid region as a straight line. Fig. 6(b) illustrates such an interface extension based on Fig. 6(a), and we assign the ghost volume for those blue-shadowed cells, i.e., $c = c_e$. In more detail, c_e is evaluated by viewing the original solid part lying below the liquid/gas interface as liquid. For instance, the solid cell $(i-1, j-1)$ initially features $c = \text{Null}$, but it has a ghost value of $c = c_e = 1$ after the interface extension. These ghost settings on c enable us to construct a 3×3 stencil to estimate $\nabla_{(i,j)} c$ surrounding the contact line cell (i, j) . Consequently, the surface tension term $\sigma \kappa \delta_s \mathbf{n}_l$ in Eq. (1) in the vicinity of the embedded solid can also be calculated using $\sigma \kappa \nabla c$, where c now has values even in those solid cells. Please note that this interface extension method, featuring the usage of a constructed c_e to impose the contact angle condition, was first introduced in *Basilisk* by Popinet <http://basilisk.fr/sandbox/popinet/contact/> and by Tavares et al. [18]. In the current work, the calculation of c_e is further refined by incorporating the identification of the contact line cell. However, it is important to mention that we do not directly use c_e to impose the contact angle condition, and we will explain this in detail later.

So far, we have detailed the calculation of the normal vector $\mathbf{n}_{l,c}$ used in the surface tension force in the presence of embedded solid boundaries. The construction of ghost volume fractions enables us to implement a balanced-force model between surface tension and pressure in solving the momentum equations. Additionally, as mentioned in § 2.4, another type of normal vector must be recalculated to reconstruct the liquid/gas interface and reflect the contact angle condition in contact line cells. This normal vector is not $\mathbf{n}_{l,c}$ but rather $\mathbf{n}_{l,h}$, obtained using the height function method implemented in the original *Basilisk*. However, extending the HF method to scenarios involving embedded boundaries is very challenging. Consequently, some studies [16,18] have directly used geometrical rotation or the extended volume fraction method to estimate the normal vector, i.e., $\mathbf{n}_{l,\theta}$ or $\mathbf{n}_{l,c}$, for imposing contact angles on complex geometries. Nonetheless, a non-physical pinning scenario of the contact line has been reported [18] when Γ_l crosses the cell face of interfacial cells. This numerical error accumulates during computations, causing the shape of the contact line to deviate significantly from its physical pattern. The reason is straightforward: the information used to extend the liquid/gas interface only contains $\mathbf{n}_{l,\theta}$ and the volume fraction of the contact line cell itself (or a layer of adjacent mixed cells), whereas \mathbf{n}_l should also rely on the shape of the interface in the surrounding cells. Accurate estimation of \mathbf{n}_l in the contact line cell requires geometrical information from neighboring cells, particularly those away from embedded boundaries. Therefore, a rigorous height function (HF) method, incorporating height values in a wide stencil and aligning with the *Basilisk* framework, is necessary to impose the contact angle condition. Extending the original HF method, specialized for flat solid substrates, to scenarios involving irregular embedded boundaries is our next task.

4. Contact angle condition in contact line cells

In this section, we discuss the application of the contact angle boundary condition in a contact line cell based on the height function method. Fig. 7(a) depicts the contact line scenario without an embedded solid, where formula (13) defines the HF of interfacial cells within the fluid region. A linear extrapolation then assigns ghost values of HF to cells immersed in the solid, and the contact angle

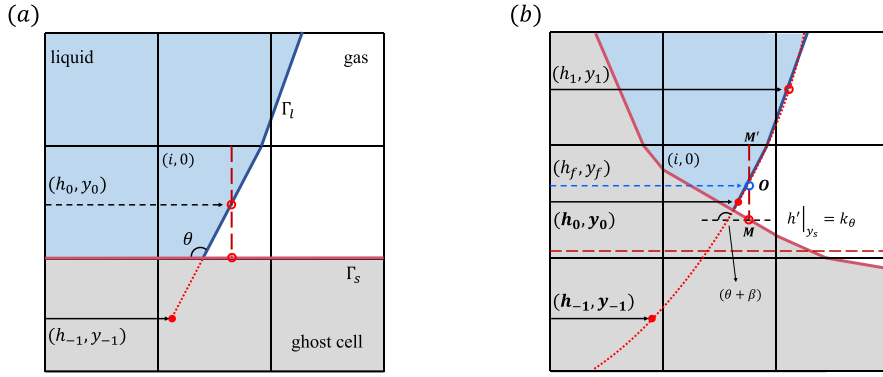


Fig. 7. Schematic diagram for calculating the height function. (a) Definition of the height function near the regular boundary where the contact angle condition is enforced at Γ_s using a linear-fitting method. (b) Height function constructed by a vertical interface in the contact line cell where the contact angle condition is implemented at Γ_s using a parabola-fitting method. MM' is the vertical interface constructed to determine the height function of the fluid region (h_f, y_f) and y_f is the y-coordinate of the midpoint O . M represents the intersection of MM' and the solid boundary, where the contact angle condition is imposed. k_θ represents the slope (contact angle condition) at M . β represents the angle between the solid boundary and the x-axis. The red and blue open circles represent the known points used to fit the parabola, and the solid red circles represent the height values in position to evaluate the normal vector and the curvature.

condition θ is enforced through Eq. (15). However, in the presence of an embedded solid boundary, as shown in Fig. 7(b), while the HF values in non-contact line cells can still be computed directly using the extended liquid volume fraction, the linear extrapolation formula (Eq. (15)) may fail to enforce the contact angle condition. This necessitates a modified HF method.

4.1. Height-function based contact angle boundary condition

Still taking Fig. 7(b) as an example, the contact line cell $(i, 0)$ contains an embedded boundary, and the problem now is to numerically impose the contact angle θ at the triple point. The HF can still be defined in this row using Eq. (13), which summarizes the liquid volume of the cells corresponding to $j = 0$, however the interval for accumulating liquid volume fraction c is the fluid region in mixed cells rather than the complete cells. The stencil has a maximum of 9 cells, ranging from $(i - 4, 0)$ to $(i + 4, 0)$ as previously stated. The obtained height function has a value of h_f in row $j = 0$, while the interface height is denoted by the vertical dashed line MM' , with M representing the intersection (or triple) point between MM' and Γ_s . It is important to note that h_f is not defined at the center line of the entire cell but at the centroid of the height interface, marked as O in the picture with $y = y_f$. Clearly, the locations of $y = y_0$ and $y = y_f$, as depicted in Fig. 7(b), do not coincide. If we continue to use formula (15) to impose the contact angle condition, the next step is to estimate h_{-1} in the cell row $j = -1$ using a linear extension. In other words, the HF value of h_{-1} is not computed using the liquid volume summation but depends solely on h_f and the contact angle value in cell $(i, 0)$. Additionally, Eq. (15) should be reformulated as $h_{-1} = h_f + |y_f - y_{-1}| / \tan(\theta + \beta)$, where β is the inclination angle of Γ_s with respect to the x-axis, as shown in Fig. 7(b). However, this linear interpolation scheme is no longer second-order accurate in the presence of the embedded solid. This is because $y = y_f$ not aligning with the cell center causes Eq. (15) to deviate from the center-difference scheme. Specifically, the red dashed line representing the center line between $y = y_f$ and $y = y_{-1}$ departs from the horizontal line passing through point M . Consequently, the simple extension of Eq. (15) enforces θ onto the red dashed line rather than on the black one, where it is truly needed.

Then, in order to maintain second-order accuracy in imposing the contact angle condition, a new parabola-fitting approach is proposed for the estimation of h_{-1} , as detailed in Algorithm 5. Specifically, the interface near the contact line cell $(i, 0)$ is approximated using a parabola rather than the aforementioned linear segment:

$$h(y) = a_0 y^2 + a_1 y + a_2, \quad (25)$$

where $h(y)$ is the horizontal height function in Fig. 7(b). To obtain the values of a_0 , a_1 , and a_2 , the information from (h_1, y_1) and (h_f, y_f) is used. Additionally, one extra HF piece of information is required to determine these coefficients. To complete this, the contact angle information implicitly expressed by the slope of the interface at point M , with $y = y_s$, is utilized, given as $h' = dh/dy = k_\theta = 1/\tan(\theta + \beta)$. Subsequently, the following matrix computation is constructed:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_f^2 & y_f & 1 \\ 2y_s & 1 & 0 \\ y_1^2 & y_1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} h_f \\ k_\theta \\ h_1 \end{bmatrix}, \quad (26)$$

by which a_0 , a_1 and a_2 are obtained accordingly. Substituting these coefficients into Eq. (25), h_{-1} is computed directly by enforcing $y = y_{-1}$. Similarly, h_0 is obtained at the center line of complete cells. It is important to note that (h_{-1}, h_0, h_1) are subsequently used to calculate the normal vector and curvature of the liquid/gas interface Γ_l , as formulated by Eqs. (16) and (14), respectively. Finally, the normal vector $\mathbf{n}_{l,h}$ and the corresponding liquid volume fraction c in the contact line cells enable us to reconstruct the interface there

Algorithm 5: A modified HF method in the presence of embedded boundaries described in Fig. 7(b).**Input:** c, c_s **Output:** h_0, h_{-1} if find contact line cell (i, j) by Algorithm 4 then

1. sum c by Eq. (13) starting from a cell of $c = c_s$ to a cell of $c = 0$ to define HF in the $j = 0$ -th cell row;
2. calculate the midpoint (h_f, y_f) of the vertical interface MM' defined by the HF in the $j = 0$ -th row and find the intersection of MM' with the embedded boundary in the contact line cell (M at $y = y_s$);
3. fit a parabola using Eq. (26) with the points (h_f, y_f) , (h_1, y_1) and the slope k_θ at $y = y_s$;
4. calculate h_0 and h_{-1} using Eq. (25).

end

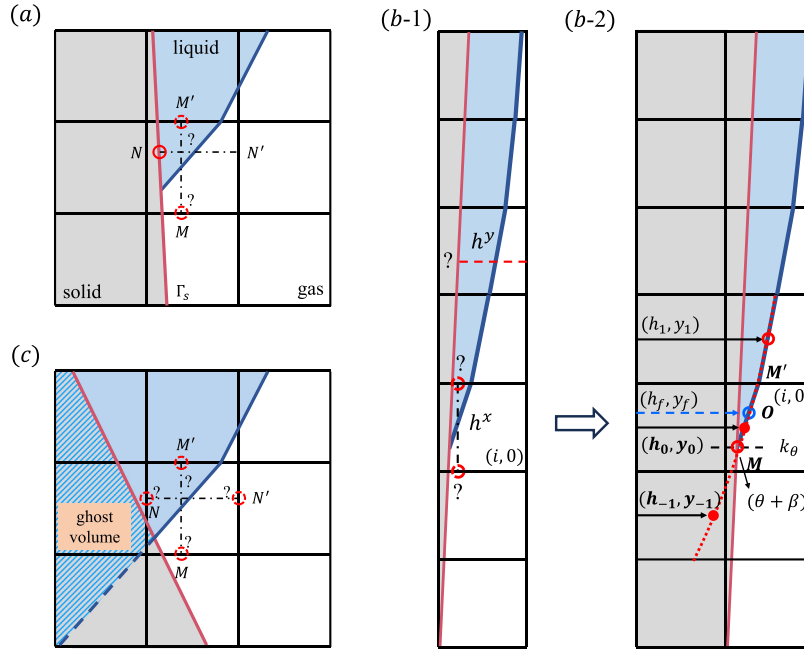


Fig. 8. Sketches of the issue without intersection. The circles of the dashed line indicate that the intersection is not found. The circle of the solid line indicates that the intersection is found. (a) The embedded boundary is almost parallel to the y -axis, causing the height function (HF) in the x -direction to fail. (b-1) On the basis of (a), the y -direction HF definition fails at the same time because the liquid/gas interface exceeds the volume fraction accumulation range from $(i, -4)$ to $(i, 4)$. h^x and h^y represent the HF in the x - and y -directions, respectively. (b-2) Interface height constructed by an interface with $\mathbf{n}_{l,\theta}$, and the procedure for parabola-fitting is the same as that for the vertical interface. (c) Failure in both x - and y -directions, and the HF are determined by the extended c directly.

via Algorithm 1. The contact angle condition is numerically realized in cell $(i, 0)$ in a purely geometrical manner. It is essential to keep in mind that the aforementioned parabola fitting procedure, illustrated in Fig. 7(b), employs horizontal cell rows as an example and is similarly applicable to vertical cell columns with minor adjustments, such as using (h_1, x_1) and (h_f, x_f) instead. Therefore, in each contact line cell, the determination of which direction of HF is preferable for parabola fitting depends on the interface normal, approximated by $\mathbf{n}_{l,c} = \nabla c$ as discussed in § 3.4.

As mentioned earlier, the parabola fitting method requires the height of the interface to intersect with the embedded solid, necessitating the identification of points O and M in Fig. 7(b). However, this condition may not be met when the calculated interface height fails to intersect with Γ_s . We identify three scenarios that correspond to this issue:

1. The embedded boundary is almost parallel to the x - or y -axis. As shown in Fig. 8(a), the criterion $\mathbf{n}_{l,c} = \nabla c$ prefers to select a horizontal HF to evaluate the height interface in the contact line cell, given as MM' . However, this priority choice does not obtain an intersection with Γ_s , thus reinforcing the secondary choice of using the vertical HF to calculate the fluid height NN' and identify an intersection point.
2. Both the embedded boundary and the liquid/gas interface are parallel to the x - or y -axis. This situation is depicted in Fig. 8(b-1), where the embedded boundary resembles that shown in Fig. 8(a). Moreover, the contact angle θ is very small. In such cases, finding a suitable horizontal HF becomes impractical, and constructing a vertical HF also poses challenges. This is because at most a 1×9 stencil can be utilized to determine the HF value for the row (or column) containing the interface. Specifically, in Fig. 8(b-1), the 1×9 stencil fails to locate a starting cell satisfying $c = c_s$ in the i -th column, a prerequisite as per step 1 of Algorithm 5. Hence, both a horizontal and vertical HF in such scenarios fail. Therefore, in this particular situation, the position of point M , where the contact angle is enforced, needs to be redefined. We propose a solution in Fig. 8(b-2), where the interface in

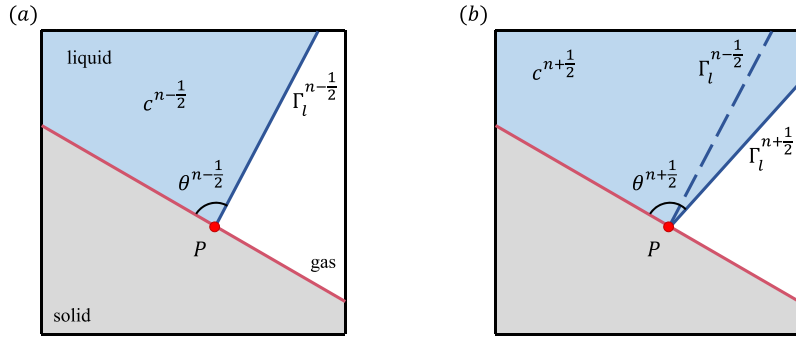


Fig. 9. Sketch of the calculation of $\theta^{n+\frac{1}{2}}$ within the hysteresis window: (a) at the time step of $n - \frac{1}{2}$, (b) at the time step of $n + \frac{1}{2}$, where the dashed and the solid lines represent the liquid/gas interface at $n - \frac{1}{2}$ and $n + \frac{1}{2}$, respectively.

the contact line cell $(i, 0)$ is first reconstructed using the normal vector $\mathbf{n}_{l,\theta}$ (see Algorithm 3) via a geometrical rotation scheme. Subsequently, the reconstructed interface intersects with the embedded boundary at point M . Once again, a parabola fitting scheme defined in Eq. (26) is applied using (h_1, y_1) , (h_0, y_0) , and the interface slope k_θ at point M .

3. The solid fraction in the contact line cell is very small. As illustrated in Fig. 8(c), the solid phase in the contact line cell (i, j) occupies only a tiny fraction at the lower left corner. In this scenario, neither the horizontal fluid height MM' nor the vertical height NN' can intersect with the embedded solid Γ_s . Consequently, we resort to using the ghost fluid fractions c_e , represented by the blue shaded region in the diagram, to define the HF in the cell row (or column) containing the contact line cell (i, j) . Additionally, the HF values in the neighboring row (or column) embedded in the solid phase are updated accordingly. Subsequently, the normal vector (16) and the curvature (14) of the interface can be computed directly using these HF values. It is important to note that this method does not enforce the contact angle at the exact desired position (black dashed line in Fig. 7(b)) but introduces a small deviation (red dashed line in Fig. 7(b)).

The proposed parabola-fitting method allows for precise imposition of the contact angle condition on the embedded solid boundary, whereas the linear extrapolation method [18] may result in an offset from the desired position. The advantage of the parabola-fitting method is demonstrated through a test case discussed in § 5.2.1. Moreover, this method utilizes the same cell stencil used for normal and curvature estimation (see Eqs. (16) and (14)), thereby ensuring straightforward code implementation and parallelization.

4.2. Hysteresis and dynamic contact angle model

Up to now, the numerical method to implement a static contact angle on embedded boundaries has been elucidated. However, a more practical yet challenging issue involving dynamic contact angle models and contact angle hysteresis deserves attention. Briefly, contact angle hysteresis describes the pinned scenario of the contact line if θ falls within the interval $[\theta_r, \theta_a]$, where θ_r and θ_a denote the receding and advancing contact angles, respectively. Addressing this problem, the numerical challenge lies in maintaining the immobility of the contact line within this angle interval. Extending the numerical method to complex boundaries exacerbates this difficulty.

In this study, a numerical model is proposed based on the aforementioned technique specialized for embedded solid boundaries. Particularly, an iterative approach is employed to determine the contact angle when considering hysteresis. Fig. 9(a) illustrates that the contact line is located at point P at time level $n - \frac{1}{2}$. Subsequently, the volume fraction in the contact line cell is updated to $c^{n+\frac{1}{2}}$ after solving the advection equation Eq. (3) in the next time step, as shown in Fig. 9(b). The contact angle hysteresis necessitates computing $\theta^{n+\frac{1}{2}}$ to determine if this angle falls within the interval $[\theta_r, \theta_a]$ or not. Based on this determination, we decide whether point P should remain stationary or move forward. Initially assuming the contact line remains pinned at the same location implies $P^{n+\frac{1}{2}} = P^{n-\frac{1}{2}}$. However, it is crucial to note that the volume fraction must differ between the two time steps, i.e., $c^{n+\frac{1}{2}} \neq c^{n-\frac{1}{2}}$, due to volume fraction transport. Consequently, $\theta^{n+\frac{1}{2}}$ must differ from $\theta^{n-\frac{1}{2}}$. The next step involves geometrically estimating $\theta^{n+\frac{1}{2}}$ using the pinned location P and the constrained volume fraction $c^{n+\frac{1}{2}}$. This strategy employs a bisection iteration scheme detailed in Algorithm 6, which essentially rotates the interface $\Gamma_l^{n-\frac{1}{2}}$ known prior to matching $c^{n+\frac{1}{2}}$ by pinning the triple point at P . The subscript i in Algorithm 6 represents the i -th iteration in the bisection iteration process.

After obtaining $\theta^{n+\frac{1}{2}}$ via Algorithm 6, the hysteresis judgment is further applied. Simply put, the contact line remains pinned in successive time steps if $\theta^{n+\frac{1}{2}} \in [\theta_r, \theta_a]$. Otherwise, it moves along the embedded boundary, requiring further determination of the contact angle $\theta^{n+\frac{1}{2}}$. Under such circumstances, a dynamic model [4] is employed to estimate θ when the contact line moves on the solid boundary. The formulation of this dynamic model is:

Algorithm 6: Calculation of the contact angle within the hysteresis window.**Input:** $\theta_a, \theta_r, c^{n+\frac{1}{2}}, P^{n-\frac{1}{2}}, \theta^{n-\frac{1}{2}}$ **Output:** $\theta^{n+\frac{1}{2}}$

1. Initialization: $\theta_i = \theta^{n-\frac{1}{2}}, \theta_{max} = \theta_a, \theta_{min} = \theta_r$;
2. apply contact angle boundary condition θ_i with Algorithm 5;
3. estimate the interface normal $\mathbf{n}_{i,h}$ in the contact line cell with Eq. (16);
4. reconstruct the interface with $(\mathbf{n}_{i,h}, c^{n+\frac{1}{2}})$ by Algorithm 1 and find the intersection P_i with the solid surface;
5. if P_i locates on the liquid side of $P^{n-\frac{1}{2}}$, set $\theta_{max} = \theta_i$; otherwise, set $\theta_{min} = \theta_i$;
6. if the distance between P_i and $P^{n-\frac{1}{2}}$ is smaller than $10^{-10} \cdot \Delta$, set $\theta^{n+\frac{1}{2}} = \theta_i$;
otherwise, set $\theta_i = (\theta_{max} + \theta_{min})/2$ and then repeat steps 2 to 5.

$$g(\theta_m) = g(\theta_s) + Ca \ln \left(\frac{l_m}{l_s} \right), \quad \text{with} \quad g(\phi) = \int_0^\phi \frac{\phi - \cos \varphi \sin \varphi}{2 \sin \varphi} d\varphi, \quad (27)$$

where θ_s is the static contact angle at the microscopic length scale of $l_s \sim 1$ nm. It is affected by intermolecular forces and is assumed constant in the present simulation. θ_m is the desired dynamic contact angle at the intermediate length scale of $l_m \sim 10 \mu\text{m}$, and it is influenced by θ_s and the contact line speed U_{cl} incorporated in Ca . The function $g(\phi)$ and its inverse, originally expressed by Eq. (27), can be approximated by polynomials $g(\phi) \approx \phi^3/9 - 0.00183985\phi^{4.5} + 1.845823 \times 10^{-6}\phi^{12.258487}$ and $g^{-1}(\phi) \approx (9\phi)^{1/3} + 0.0727387\phi - 0.0515388\phi^2 + 0.00341336\phi^3$. U_{cl} is estimated in the contact line cell by:

$$U_{cl} = \mathbf{u}_i \cdot \boldsymbol{\tau}_s, \quad (28)$$

where $\boldsymbol{\tau}_s$ is the tangential vector along the solid boundary and \mathbf{u}_i is the interface velocity obtained by averaging the velocity in those interfacial cells without solid boundary and neighboring the contact line cell. Moreover, the cell size near the embedded boundary is fixed at $\Delta \sim 10 \mu\text{m}$ in accord with the intermediate scale of l_m . Therefore the computation procedure to obtain dynamic angle is clear, $g(\phi = \theta_s)$ and $Ca \ln(l_m/l_s)$ enables us to obtain $g(\theta_m)$, which can be further combined with $g^{-1}(\phi = \theta_m)$ to calculate θ_m . Once the contact angle is determined using either the contact angle hysteresis model or the dynamic contact angle model, this contact angle condition can be implemented using Algorithm 5.

4.3. Entire computational procedure

So far, we have proposed a geometrical VOF method for simulating contact lines on complex boundaries, which maintains the sharpness of the liquid/gas interface Γ_l and the embedded boundaries Γ_s . In this section, we summarize the entire computational procedure, highlighting the aforementioned numerical techniques. The code for the above numerical techniques is released in <http://Basilisk.fr/sandbox/Chongsen/>. At the initial time step $n = 0$, the embedded solid boundaries are reconstructed using a piecewise linear scheme. Algorithm 2, proposed in § 3.3, is employed to remove small mixed cells, thereby relaxing the CFL condition that limits the time step in solving the advection equation for the volume fraction. Simultaneously, the normal vector \mathbf{n}_s of the embedded boundaries is determined based on the geometrical information in the local mixed cells. Subsequently, the liquid/gas interface is initialized within the computational domain. The numerical procedure for each time step is outlined as follows:

1. Update the liquid volume fraction $c^{n+1/2}$ using a geometrical advection scheme: For non-mixed cells, the liquid volume fraction is updated using Eqs. (10), (11), and (12), which are direction-split schemes. Additionally, a conservative scheme is proposed to estimate the volume fraction flowing into irregular mixed cells. The detailed discretized formulations are provided in Eqs. (10), (11), (18), (20), and (21). Note that the normal vector \mathbf{n}_l of the liquid/gas interface Γ_l is estimated by Eq. (16) based on the height function $h^{n-1/2}$ and Γ_l is reconstructed using Algorithm 1. During this step, the liquid volume fraction $c^{n+1/2}$ is obtained for all discretized cells.
2. Extend $c^{n+1/2}$ to the solid cells: Algorithms 3 and 4 are employed to identify the contact line cells that contain triple points. In the contact line cells, Γ_l is extended into the embedded boundaries using a linear extension scheme (Fig. 6(b)), and thus ghost values of $c^{n+1/2}$ are reassigned as c_e in those mixed and solid cells. Subsequently, the normal vector of the interface in the surface tension term for all discretized cells can be evaluated as $\mathbf{n}_{l,c} = \nabla c / |\nabla c|$ using a 3×3 stencil, even near the mixed cells.
3. Compute the height functions (HFs) $h^{n+1/2}$ in all discretized cells: The HFs of non-contact line cells are computed using Eq. (13) with the extended liquid volume fraction $c^{n+1/2}$ obtained in step 2. Additionally, a parabola fitting scheme (Eqs. (25) and (26)), incorporating the contact angle condition, is proposed to extend the liquid/gas interface Γ_l from the contact line cells into the embedded boundaries. This allows us to obtain ghost height fraction values h_{-1} and h_0 in the solid and contact line cells. These techniques enable the construction of a central difference stencil (Eq. (16)) for estimating the gradient of HF, even in mixed cells. Subsequently, the curvature κ of the local interfacial cells can be evaluated using Eq. (14). It is important to note that in this step, the contact angle may not be static but could exhibit hysteresis. Therefore, a hysteresis and dynamic model is also proposed in § 4.2 to handle the contact angle at the embedded boundary.
4. Estimate the surface tension term and solve the Navier-Stokes equation: After step 2 and 3, in all interfacial cells, including the mixed cells, the normal vector of Γ_l can be evaluated as $\mathbf{n}_{l,c} = \nabla c / |\nabla c|$ using a 3×3 stencil and the curvature of Γ_l can be

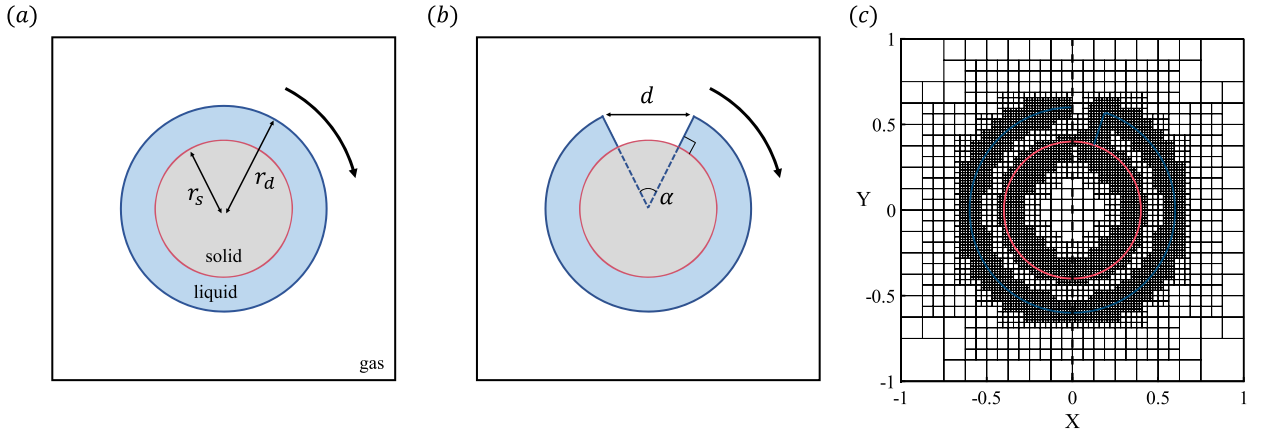


Fig. 10. (a) Sketch of a rotating liquid ring wrapping around a solid cylinder. r_s and r_d denote the radii of the solid cylinder and the liquid ring, respectively. (b) Sketch of a rotating liquid ring with a gap around a solid cylinder. α represents the angle of the gap, and d is the distance between the outermost vertex of the remaining liquid ring. (c) Mesh distribution for both examples.

estimated by a central difference stencil (Eq. (14)) based on the height function method. The local value of the surface tension in the interfacial cells is estimated as $\sigma \kappa \nabla c$. Incorporating the surface tension force, the Navier-Stokes equations are solved using the fractional step method implemented in *Basilisk*, and the numerical solution is updated to the next time step.

5. Result

In this section, we present several numerical tests to validate the accuracy and effectiveness of the proposed scheme. The first test, discussed in § 5.1, evaluates the advection algorithm presented in § 3.2 for local volume conservation. Next, in § 5.2, we simulate droplets driven by surface tension on various complex geometrical boundaries to verify the validity of the height function scheme for imposing contact angle conditions, as described in § 4.1. In § 5.3, we verify the numerical modeling of droplet hysteresis and dynamic contact angles by observing the slow motion of droplets under rotating gravity on an inclined plate. The final two numerical examples, presented in § 5.4 and § 5.5, involve more complex scenarios: a droplet impacting a plate and passing through it via a round or sharp orifice, and a droplet penetrating porous media composed of several circular cylinders with varying wettability. In all numerical settings, the embedded boundaries are configured to be completely no-slip.

5.1. VOF advection with a given velocity field

In order to elucidate the significance of using the corrected width un^* instead of un when advancing the liquid volume fraction in the presence of embedded solids, two examples are examined under a prescribed velocity field. Fig. 10(a) depicts the first example, which involves a ring of liquid/gas interface (blue line) rotating around a solid cylinder (red line). Fig. 10(b) illustrates the second example, similar to the Zalesak disk problem [39], involving a ring of liquid/gas interface with a gap. Fig. 10(c) shows the mesh distribution for the two examples, where the smallest mesh spans about 5 layers both inside and outside the interface. The rotational velocity field is defined as:

$$\begin{cases} u_x^f = 2\pi(y - y_c) \\ u_y^f = 2\pi(x_c - x) \end{cases}, \quad (29)$$

where x and y represent the center coordinates of the cell faces of the fluid part, and x_c and y_c are the centers of the cylinder and the ring. Unfortunately, Eq. (29) does not result in a divergence-free velocity field within mixed cells. However, due to the symmetry along the circumference, the sum of the velocity divergence of all mixed cells is nearly zero (approximately 10^{-15} in this example). In the context of the example presented in this section, the cells per initial radius (cpr, hereafter) amount to 38.4.

5.1.1. Rotation of a liquid ring around a cylinder

In the first example, as depicted in Fig. 10(a), a cylinder and a liquid ring are positioned at the center of the numerical domain with radii of $r_s = 0.4$ m and $r_d = 0.6$ m, respectively. The size of the numerical domain is 2 m. The velocity field is initialized and maintained using Eq. (29), so the Navier-Stokes equations are not actually solved, and there are approximately 7000 time steps in one rotation period (1 s). Because the liquid completely covers the solid cylinder at $t = 0$, the mixed cells containing the embedded solid boundary should remain full of liquid ($c = c_s$) during the advection and rotation. To assess the effectiveness of the corrected advection width un^* in maintaining local volume conservation, two error norms are introduced: the root mean square error L_2 and the maximum error L_{inf} , representing the difference between c and c_s in mixed cells.

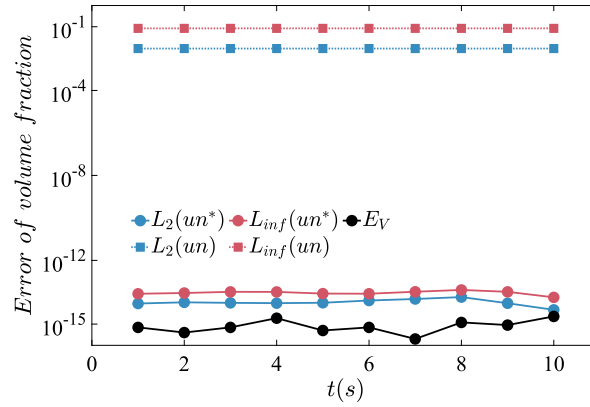


Fig. 11. Error in local and global volume conservation during ring rotation. The blue and red lines correspond to L_2 and L_{inf} norms of the difference between the volume fraction in the mixed cell around the cylinder and the initial value, representing the error of the local volume conservation. Dotted lines denote results directly using un , while solid lines depict results using un^* , as introduced in § 3.2. The black line represents the global volume loss E_V .

$$L_2 = \sqrt{\frac{\sum_i^N (c_{si} - c_i)^2}{N}}, \quad (30)$$

$$L_{inf} = \max_{N,i} (|c_{si} - c_i|), \quad (31)$$

where i identifies the mixed cell in the computational domain, and N is the total number of mixed cells. In addition, there is the relative error $E_V = |V - V_0|/V_0$ of the global volume loss, where V is the summation of the liquid volume in all cells and V_0 is that at the initial time.

The results are presented in Fig. 11. In mixed solid/liquid cells, the situation where the velocity divergence is not zero can be compensated by the central volume fraction c_c (Eq. (22)), leading to an approximation of $\nabla \cdot \mathbf{u} \approx 0$. Using the corrected advection width un^* , the maxima of L_{inf} and L_2 are about 7.2×10^{-14} and 4.6×10^{-14} , respectively, which are about on the order of machine accuracy. In contrast, using the advection width un introduces significant errors, with maximum values of L_{inf} and L_2 approximately 0.084 and 0.0094, respectively. This occurs because the utilization of un leads to overfilling ($c > c_s$) or creating spurious interfaces ($c < c_s$), as illustrated in Fig. 4. Additionally, since $c_c = 1$ in all mixed cells, compensating for the error caused by velocity divergence, the global volume loss E_V can be maintained at the order of $E_V = 10^{-15}$ consistently. Finally, the time consumption for the current advection scheme is also shown in Appendix A. The results indicate that the current scheme increases CPU time by approximately 32% compared to the traditional VOF advection method.

5.1.2. Rotation of a ring with a gap similar to Zalesak disk problem

In this section, we modify the liquid ring by excising a sector-shaped region which is similar to the Zalesak disk problem. This test is to verify whether the advection scheme for volume fraction is still correct in the presence of contact line cells. At the beginning of the simulation, the liquid/gas interface is perpendicular to the cylinder surface at the triple point, with an imposed contact angle of 90° contact angle. Besides, Eq. (29) is still applied to prescribe the velocity field in the domain, and the values of r_s and r_d still remain at $r_s = 0.4$ m and $r_d = 0.6$ m. The notched sector part has a width $d = 0.5$ m, while the angle is $\alpha = 49.2^\circ$. It takes $T = 1$ s for the liquid ring to complete one full rotation around the solid cylinder. We will use T to denote the time in the following discussion. Over one rotation period T , the simulation involves approximately 48,000 time steps.

The results are presented in Fig. 12, demonstrating that the liquid/gas interface contour near the cylinder surface aligns closely with the theoretical shape, even after 10 rotation periods. To quantify the error between the current work and the theoretical shape, we define $E_{shape(1/2)}$ as

$$E_{shape(1/2)} = \frac{\sum_i^{N_{(1/2)}} V_i |c_i(T) - c_i(0)|}{\sum_i^{N_{(1/2)}} V_i c_i(0)}, \quad (32)$$

where $N_{(1)}$ represents the total number of all cells in the simulation domain, $N_{(2)}$ represents the total number of all mixed cells, V_i is the mesh volume, $c_i(T)$ is the volume fraction in the T -th rotation cycle, and $c_i(0)$ is the initial volume fraction. Four grid resolutions are considered—19.2, 38.4, 76.8, and 153.6 cpr. Fig. 13 illustrates the dependence of the liquid ring shape and the shape error on the mesh resolution. As shown in panel (a), increasing the grid resolution significantly reduces the deformation of the liquid ring at $10T$, bringing it closer to the theoretical undeformed shape. Panel (b) demonstrates that the current method achieves first-order convergence in the global shape error $E_{shape(1)} - pre$. This result is consistent with the convergence of errors observed in the standard Zalesak disk problem [40]. Furthermore, given that the proposed advection method is applied exclusively in mixed cells, the shape error of all mixed cells, $E_{shape(2)} - pre$, is also presented in Fig. 13(b). A third-order convergence in the shape error of mixed cells is observed. Additionally, we test the example using the traditional geometric VOF advection method, employing the code available

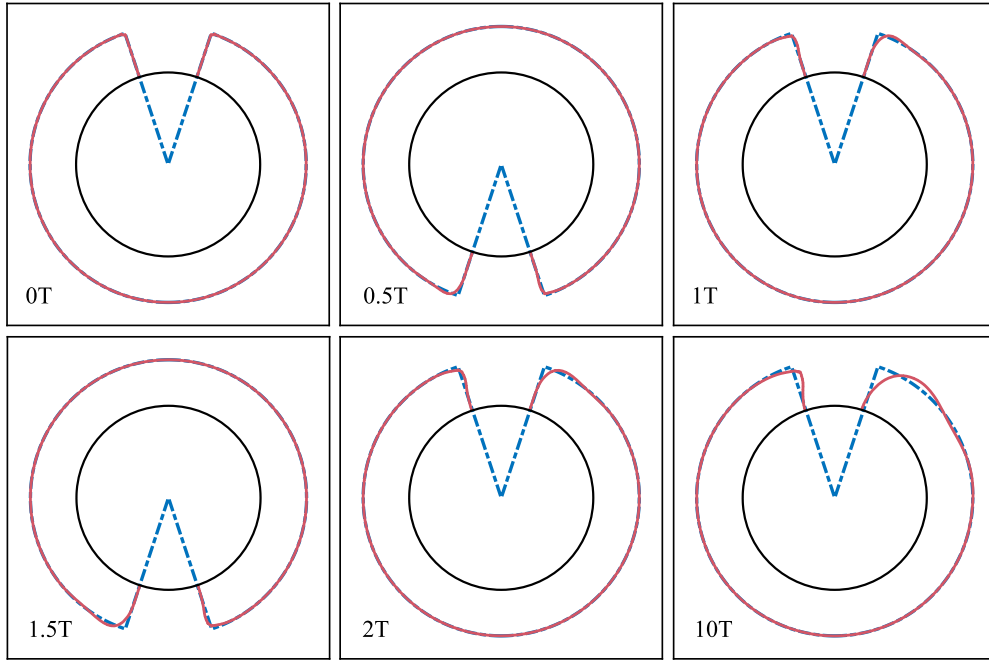


Fig. 12. Outlines after rotation. $cpr = 38.4$. The red lines depict the shapes of the liquid ring with a gap, and the blue lines denote the theoretical shape without deformation.

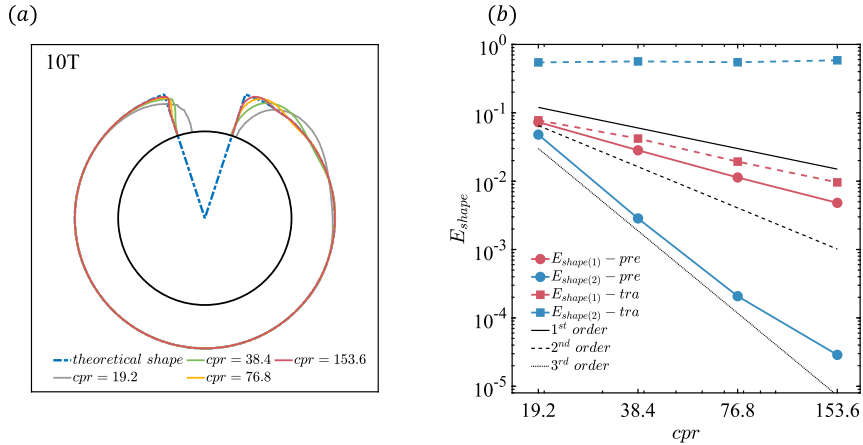


Fig. 13. (a) Outlines after rotation at $t = 10T$ with $cpr = 19.2, 38.4, 76.8$, and 153.6 . (b) Shape error at $t = 10T$. $E_{shape(1)}$ denotes the shape error for all cells in the simulation domain, while $E_{shape(2)}$ represents the shape error for all mixed cells. $-pre$ and $-tra$ indicate the results obtained using the present method and the traditional method, respectively.

at <http://basilisk.fr/sandbox/popinet/contact/> to impose the contact angle condition. The corresponding results are presented in Fig. 13(b), labeled as $-tra$. While the global shape error $E_{shape(1)} - tra$ is comparable to that of the proposed schemes, the shape error of mixed cells $E_{shape(2)} - tra$ is significantly larger and fails to converge with grid refinement. In terms of volume conservation, the volume error remains on the order of $E_V = 10^{-15}$ after 10 rotation periods, confirming the accuracy and conservation properties of the advection scheme when transporting the volume fraction in mixed cells.

5.2. Surface tension-driven droplet spreading

In this section, we present examples of static droplet spreading on both a flat plate and a cylindrical surface in the absence of gravity, with capillary force, strongly relying on the contact angle condition, as the only driver of droplet spreading. In other words, even if we alter the Cartesian grid—either by changing the proportion of the solid region in the mixed cell or by rotating the embedded boundary—the simulation should yield consistent results. The objective is thus to demonstrate the accuracy of applying contact angle

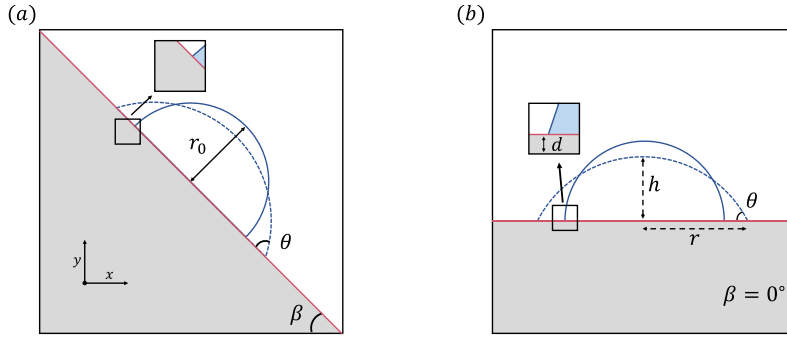


Fig. 14. Schematic representation of the initial (the solid line) and equilibrium (the dashed line) shapes of a droplet on a flat plate inclined at an angle β with the x -axis.

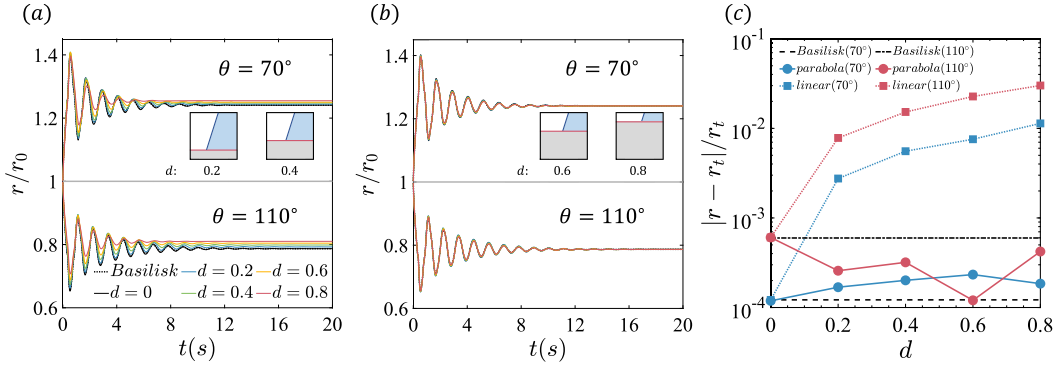


Fig. 15. Dimensionless radius of droplet spreading on a flat plate with $\beta = 0^\circ$: (a) using the linear fitting method, (b) using the parabola fitting method. (c) The relative error between the equilibrium droplet radius and the theoretical value.

conditions in numerical simulations. Unless otherwise stated, the physical properties in this section are specified as $\rho_l = \rho_g = 1 \text{ kg/m}^3$, $\mu_l = \mu_g = 7.5 \times 10^{-4} \text{ kg/(m} \cdot \text{s)}$, and $\sigma = 0.1 \text{ N/m}$, and the grid resolution is 25.6 cpr.

5.2.1. Droplet spreading on a flat plate

As illustrated in Fig. 14(a), an initially semicircular droplet, with a radius of $r_0 = 0.2 \text{ m}$, is placed on a flat plate, and the angle between the plate and the x -axis is denoted as β . By imposing different contact angles θ in simulations, the droplet starts spreading owing to the capillary effect until it reaches equilibrium, determined by the value of θ . For a droplet with a volume V and a contact angle θ , the radius and height at equilibrium are theoretically given by:

$$r_t = \sin \theta \sqrt{\frac{V}{\theta - \cos \theta \sin \theta}}, \quad (33)$$

$$h_t = (1 - \cos \theta) \sqrt{\frac{V}{\theta - \cos \theta \sin \theta}}. \quad (34)$$

It is evident that the theoretical radius and height of the droplet at equilibrium are determined solely by V and θ , and are not influenced by the inclination angle β or the position of the embedded solid boundary within the contact line cell. However, in numerical simulations, improper application of contact angle boundary conditions, such as using the linear fitting method discussed in § 4.1, can significantly affect the droplet's equilibrium radius and height.

To clarify this issue, we first consider a scenario involving a flat plate parallel to the x -axis ($\beta = 0^\circ$), as depicted in Fig. 14(b). The proportion of the solid region in the contact line cell, indicated by the zoomed inset in the figure, is denoted by d . Theoretically, the value of d should not influence the equilibrium radius or height of the droplet. We examine contact angles of $\theta = 70^\circ$ and 110° with five cases: $d = 0, 0.2, 0.4, 0.6$, and 0.8 . At these angles, the height functions used to estimate curvature are fully parallel to the x -direction. Particularly, $d = 0$ implies the embedded solid boundary coincides with the bottom face of the first layer of cells, then the present method, in applying contact angle conditions, aligns with the approach proposed by Afkhami et al. [22] and implemented in *Basilisk*. We mainly compare two methods: the parabola fitting method and the linear fitting method for contact angle imposition, as discussed in § 4.1. We again remind that the linear fitting method uses only two points, (h_f, y_f) and $(h'(y_s) = k_\theta, y_s)$, to fit a straight line for calculating h_0 and h_{-1} , while the parabola approach uses an extra point of (h_1, y_1) (see Fig. 7).

Fig. 15(a) shows the results obtained using the linear fitting method, revealing that with larger d , the dimensionless radius r/r_0 increasingly deviates from the dashed line, which represents the result obtained from the original *Basilisk* on the regular boundary

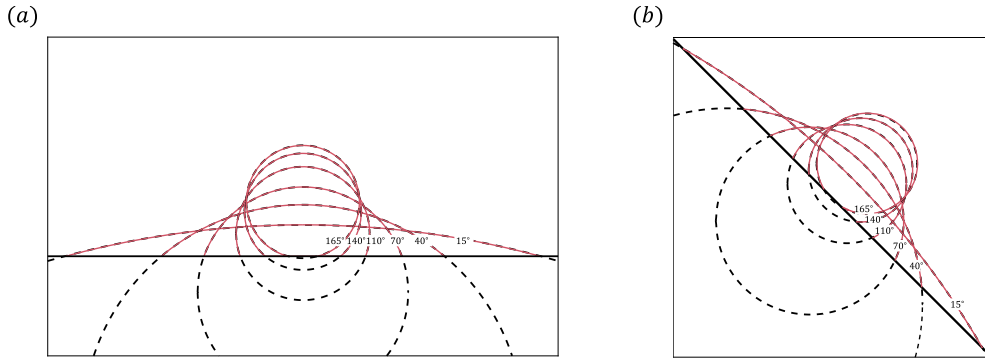


Fig. 16. Shapes of the droplet at equilibrium. $\theta = 15^\circ, 40^\circ, 70^\circ, 110^\circ, 140^\circ, 165^\circ$. (a) $\beta = 0^\circ$ and $d = 0.8$. (b) $\beta = 45^\circ$. --- : Theoretical shapes; — : Simulation shapes.

without solid (referred to as the regular boundary case, hereafter). Such inconsistency strongly indicates that by applying the linear fitting method, the numerical results of spreading behaviors depend on the position of the embedded solid, owing to improperly imposing the contact angle condition. In contrast, Fig. 15(b) shows that the radius obtained using the current parabola fitting method closely matches the regular boundary case regardless of the value of d . Fig. 15(c) gives the error statistics against different values of d by using the two fitting methods. We observe that the error with the parabola fitting method, smaller than 5×10^{-4} , remains consistent with that of the regular boundary case, showing no increasing trend as d rises (as shown by the red and blue solid lines). However, the errors with the linear fitting method are always larger than 10^{-3} except at $d = 0$, indicating a strong d -dependent spreading behavior.

It is now desirable to discuss the origin of the errors introduced by using the linear fitting method. This linear method is valid in regular boundary cases [22] because the fluid/ghost cells are symmetric to the boundary, allowing the contact angle information to be precisely reflected at the cell face, similar to the template of central difference. However, in the presence of complex geometrical boundaries, such symmetry near the contact line cells is disrupted, as discussed in § 4.1. Under such circumstances, extending a straight line into the solid region leads to a deviation between the location where the contact angle condition is imposed and the actual embedded boundary. Conversely, by considering the varying slope of the ghost interface within the embedded solid region—i.e., using the parabola fitting method to enforce the contact angle condition—the application position of the contact angle condition is ensured to be on the embedded solid boundary all the time, regardless of how the solid boundary is embedded within the contact line cell. Thus, the contact angle condition is accurately imposed. In addition, the additional time consumption required by the current parabola fitting method is less than 5% compared to that of the linear extension scheme.

Furthermore, we examine the different ratios of density or viscosity to test this example, as demonstrated in Appendix B. The results obtained with the present method for applying the contact angle condition align closely with those of regular boundary cases.

Subsequent examinations are conducted on droplets with more contact angles, while the flat plate also rotates to form arbitrary inclination angles with the x -axis. Six contact angles are considered: $\theta = 15^\circ, 40^\circ, 70^\circ, 110^\circ, 140^\circ$, and 165° , while the inclinations of the plate are $\beta = 0^\circ$ and $\beta = 45^\circ$. In the case of $\beta = 0^\circ$, the position of the embedded plate is still controlled by d , which is either 0.2 or 0.8 in this series of tests. In principle, the numerical results about the droplet spreading behaviors should only rely on θ but be independent of β or d as we stated previously. It should be noted that for a hydrophobic plate with $\theta = 165^\circ$, we adjust the viscosity to higher values of $\mu_l = \mu_g = 2.5 \times 10^{-2} \text{ kg/(m} \cdot \text{s)}$ to avoid the bounce of the droplet from the plate. In addition, for a horizontal plate with a prescribed contact angle of $\theta = 15^\circ$ and 165° , the curvature is primarily calculated using the height function perpendicular to the x -axis (computed as Fig. 8(b-2)), whereas for other cases the parallel height function (calculated as Fig. 7(b)) is predominantly utilized.

The shapes of the droplet at equilibrium, corresponding to different contact angles and inclinations, are depicted in Fig. 16, which plots the numerical results with solid lines and the theoretical solution with dashed lines. It can be observed that the two sets of results are in excellent agreement, again verifying that the contact angle condition realized using the parabola fitting method is independent of the position and inclination of the embedded boundary. More quantitatively, we further compare the computational errors of spreading radius and height in different scenarios, as shown in Fig. 17(a), where the theoretical solutions are also presented for comparison. Clearly, the numerical results of r_i and h_i almost coincide with each other at the same θ and their values show close agreement with the theoretical solutions, while the method implemented by Tavares et al. [18] gives some inconsistent results at small and large contact angles. Besides, without considering gravity, the balance relationship between the surface tension and the difference between internal and external pressure can be easily obtained, given as $\Delta p_i r_0 / \sigma = \sqrt{(2\theta - \sin 2\theta) / \pi}$. The comparisons of $\Delta p_i r_0 / \sigma$ between the numerical results and this analytical solution are also presented in Fig. 17(b), and still, very good agreement is observed. In addition, Fig. 17(c) compares the volume loss obtained in the present numerical simulations to the regular boundary cases and the work of others [19,17], who either used a Level-Set or a VOF method. We find their values of E_V are on the order of 10^{-2} and 10^{-5} , respectively, significantly higher than our numerical errors falling between 5×10^{-6} and 5×10^{-8} , as depicted by the shadowed region of the picture. Therefore, in comparison to other works, the current scheme shows a better performance in

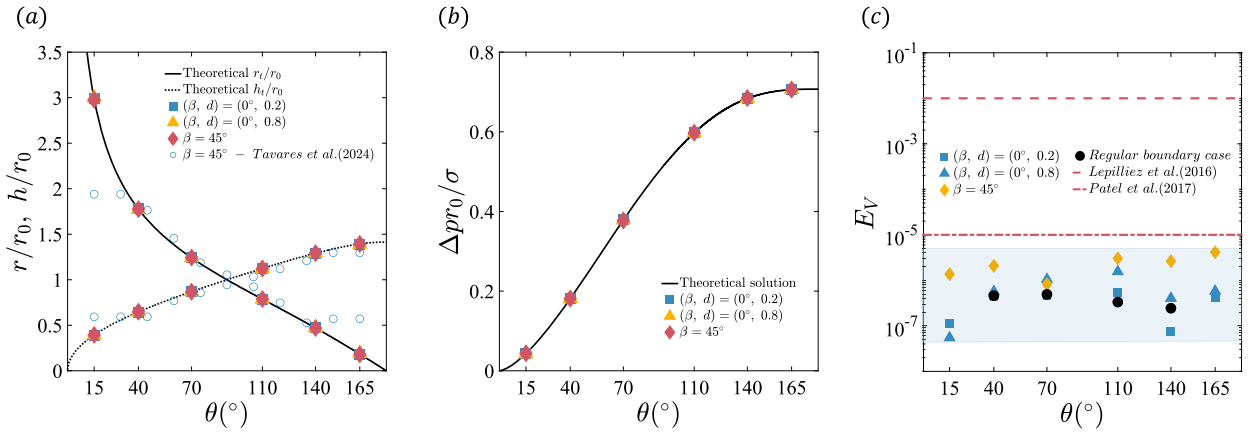


Fig. 17. (a) Dimensionless droplet radius and height of the equilibrium shape, closed points are our numerical results while the open circles are those obtained by Tavares et al. [18]. (b) Dimensionless pressure difference for a droplet on a flat plate at equilibrium. (c) Volume loss for the droplet at equilibrium.

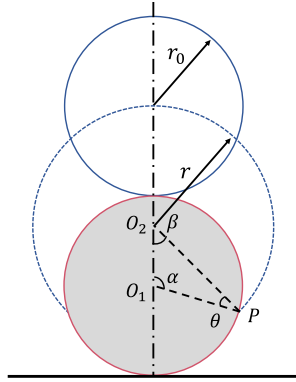


Fig. 18. Schematic representation of droplet spreading on a cylinder driven by surface tension. r_0 and r denote the initial and equilibrium radii of the droplet, respectively. O_1 and O_2 represent the centers of the solid cylinder and the equilibrium circular cap, respectively. P is the triple point, and θ is the specified contact angle.

conserving the volume during droplet spreading, maintaining similar numerical errors as those corresponding to regular boundary cases.

In summary, the results demonstrate that in applying contact angle boundary conditions at complex geometries, the utilization of the parabola fitting scheme instead of the ordinary linear fitting scheme is rather important. This numerical technique ensures consistent numerical results. While more recently, similar numerical studies have been carried out by Tavares et al. [18], they are able to obtain good results for droplet spreading on a horizontal plate; however, their results are less satisfactory when the plate is inclined with an angle of $\beta = 45^\circ$. This deterioration at non-zero β is primarily due to the inaccurate model of imposing the contact angle boundary condition at the embedded solid boundary.

5.2.2. Droplet spreading on a cylinder

Next, we turn to a more complex scenario involving droplet spreading on a circular cylinder without gravity, as illustrated in Fig. 18. Both the droplet and the cylinder have a radius of $r_0 = 1$ mm, and only half of the droplet is simulated due to the symmetry of the configuration. In this scenario, the final spreading radius of the droplet, denoted by r in Fig. 18, is fully controlled by the contact angle. At equilibrium, the contact line point P and the centers O_1 and O_2 of the solid cylinder and the spreading droplet, respectively, form two angles, α and β , as depicted in Fig. 18. The contact angle equals the angle between lines O_1P and O_2P , leading to the following geometrical relations:

$$\alpha + \beta + \theta = \pi, \quad (35)$$

$$r^2(\cos \beta \sin \beta - \beta + \pi) + r_0^2(\cos \alpha \sin \alpha - \alpha - \pi) = 0, \quad (36)$$

$$r \sin \beta = r_0 \sin \alpha. \quad (37)$$

Solving these equations determines the theoretical radius at equilibrium for different contact angles. Five contact angles are considered in this section: $\theta = 30^\circ, 60^\circ, 90^\circ, 120^\circ$, and 150° . In numerical simulations, the equilibrium radius r of the droplet is

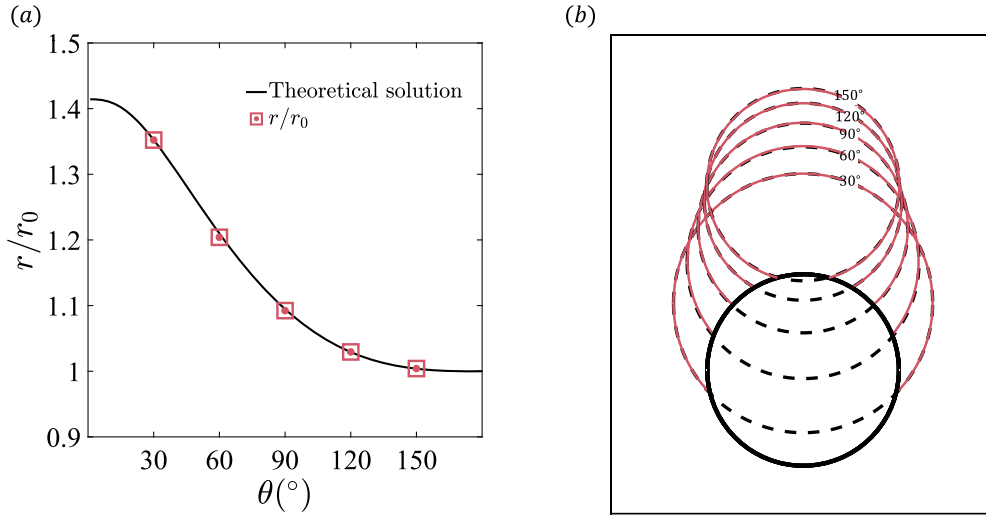


Fig. 19. Droplet spreading on a cylinder driven by surface tension. (a) Dimensionless droplet radius at equilibrium. (b) Shapes of the droplet at equilibrium with $\theta = 30^{\circ}, 60^{\circ}, 90^{\circ}, 120^{\circ}, 150^{\circ}$. --- : Theoretical shapes; — : Simulation shapes.

Table 1

E_V of the droplet spreading on the cylinder.

| $\theta(^{\circ})$ | 30 | 60 | 90 | 120 | 150 |
|--------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| E_V | 1.4×10^{-5} | 5.3×10^{-7} | 2.2×10^{-5} | 9.4×10^{-6} | 1.1×10^{-6} |

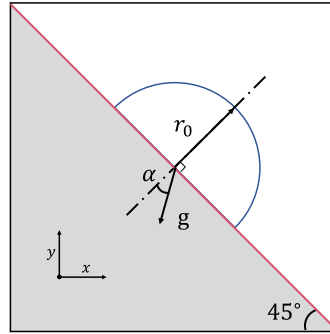


Fig. 20. Sketch of a droplet sliding on an inclined solid boundary under the influence of gravity.

obtained using the least squares method to fit the droplet's configuration. Fig. 19(a) presents the computed droplet radius r/r_0 at equilibrium for different contact angles, alongside the theoretical solution for comparison. The results show excellent agreement, and Fig. 19(b) further reveals that the overall shapes of the droplets match well with the theoretical ones. Additionally, Table 1 shows the volume loss, indicating that the errors never exceed 2.2×10^{-5} . In contrast, recent studies by Liu et al. [20] and Tavares et al. [18] report volume errors of the same problem on the order of 10^{-3} , demonstrating that our numerical results achieve better volume conservation due to the conservative approach to transporting the volume fraction along complex geometries.

5.3. Droplet sliding on an inclined embedded boundary

In this section, two examples are presented to validate the numerical methods for dynamic contact angle model with hysteresis implemented in § 4.2. To facilitate comparison with previous studies, we conducted simulations of the droplet moving on a flat plate inclined at 45° to the x -axis, with gravity included in the numerical simulations.

5.3.1. Contact angle hysteresis

As depicted in Fig. 20, we place an initially hemispherical droplet on a flat plate inclined at 45° to the x -axis. The droplet then slides, deforms, and spreads under the combined effects of surface tension and gravity. The physical properties of the droplet are $\rho_l = 1000 \text{ kg/m}^3$, $\rho_g = 1 \text{ kg/m}^3$, $\mu_l = 0.01 \text{ kg/(m} \cdot \text{s)}$, $\mu_g = 2 \times 10^{-5} \text{ kg/(m} \cdot \text{s)}$. The initial radius is $r_0 = 0.5 \text{ mm}$, the numerical domain is a square of $5r_0$, the time step is $dt = 10^{-6} \text{ s}$, and the grid resolution is 51.2 cpr, meeting the requirement of $\Delta \sim 10 \mu\text{m}$.

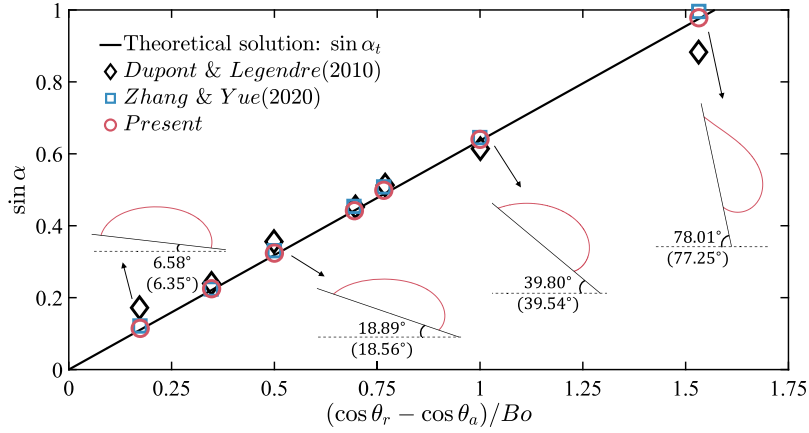


Fig. 21. Critical inclination angles and droplet shapes. Black diamonds and blue squares represent results from [4] and [25] on regular boundaries, respectively. Red circles denote results from the present study. The red outlines depict droplet shapes from the current work. The first angle is the simulated angle, while the second angle in parentheses represents the theoretical solution. In the diagram, the angle between the solid boundary and the horizontal line corresponds to the critical rotation angle α in the simulation.

as mentioned in § 4.2. The gravity vector has an inclination of α with respect to the direction perpendicular to the solid boundary, with a magnitude of $g = 9.81 \text{ m/s}^2$. The Bond number $Bo = \rho_l g r_0^2 / \sigma$ characterizes the ratio of gravitational force to surface tension force, varying between $Bo = 0.5$, 1, and 2 in these test cases. This numerical example is designed to test the hysteresis model of the contact angle. Therefore, we gradually increase α to produce droplet sliding, ensuring the rotation time scale is greater than the inertial capillary time $t_i \sim (\rho_l r_0^3 / \sigma)^{1/2}$ and the viscous capillary time $t_\mu \sim \mu_l r_0 / \sigma$ to maintain the droplet in an equilibrium state during rotation. The prescribed hysteresis angles are consistent with other reference studies [4,25], namely $(\theta_r, \theta_a) = (80^\circ, 100^\circ)$, $(60^\circ, 120^\circ)$, and $(40^\circ, 140^\circ)$. In principle, the droplet should remain pinned on the inclined solid boundary if the contact angle falls within (θ_r, θ_a) during gravity vector rotation. Otherwise, it partially slides along the solid boundary when the lower-right or upper-left contact angle reaches θ_a or θ_r , and then the angle is set to the corresponding θ_a or θ_r . Additionally, if both contact angles reach the corresponding θ_a or θ_r during the rotation, the droplet starts completely sliding along the inclined solid boundary.

By examining the force balance between gravity and surface tension along the tangential direction of the solid surface, the critical inclination angle α_t at which the droplet starts completely sliding can be determined by the following formulations [41,42],

$$\frac{1}{2} \pi r_0^2 \rho_l g \sin \alpha_t = \sigma (\cos \theta_r - \cos \theta_a), \quad (38)$$

which can be further reformulated by incorporating variables into Bo ,

$$\sin \alpha_t = \frac{2}{\pi Bo} (\cos \theta_r - \cos \theta_a). \quad (39)$$

In the nine examples under investigation, we find that the droplets with $(\theta_r, \theta_a) = (60^\circ, 120^\circ)$ and $(40^\circ, 140^\circ)$ with $Bo = 0.5$ are always pinned on the inclined solid boundary even when α increases to 90° . Correspondingly, we only display the results of the other seven examples in Fig. 21, which shows the critical inclination angle α_t obtained in numerical simulations. The values of α_t in different situations agree well with the theoretical solutions, indicated in brackets in the figure. Additionally, the droplet shapes at the moment of complete sliding initiation are also shown for reference.

5.3.2. Dynamic contact angle

In this section, the numerical test is similar to that presented in Fig. 20, except we use a dynamic contact angle model rather than maintaining the contact angles at $\theta_{r,a}$ after the droplet starts sliding. While we retain the same physical properties as in the previous test cases, the droplet radius is now $r_0 = 1 \text{ mm}$, the Bond number is $Bo = 1$, and the receding and advancing contact angles are 60° and 120° , respectively. At the start of the simulation, we maintain the inclination angle at $\alpha = 0^\circ$ until $t = 0.1 \text{ s}$, allowing the droplet to reach equilibrium on a horizontal plate. We then rotate the gravity vector with an angular velocity of $1^\circ / 10^{-3} \text{ s}$ to induce droplet sliding along the inclined solid boundary.

The variation in the contact angles at the upper-left and lower-right contact lines of the droplet is shown in Fig. 22(a). We observe slight non-equilibrium oscillations at $t < 0.04 \text{ s}$ as the droplet adapts to the hysteresis window. During $t < 0.1 \text{ s}$, the angles of the two contact lines match well with those obtained on the regular boundary, as reported in [24]. When gravity vector starts rotating at $t > 0.1 \text{ s}$, bifurcations in the contact angles at the upper-left and lower-right of the droplet are observed. Specifically, the lower-right contact line starts sliding at $t \approx 0.12 \text{ s}$ when the contact angle exceeds $\theta_a = 120^\circ$, and the upper-left contact line starts sliding at $t \approx 0.145 \text{ s}$ when the contact angle drops below $\theta_r = 60^\circ$. The two separate curves indicate the angles of the contact lines vary with time, thus validating the dynamic contact angle model described by Eq. (27) in the numerical simulations. Fig. 22(b) shows the time histories of the contact line positions at the upper-left and lower-right of the droplet. For clarity, we shift the starting position of the upper-left contact line $X_{cl(\text{upper-left})}$ from -1 mm to 1 mm . This explains why $X_{cl(\text{upper-left})}$ overlaps with $X_{cl(\text{lower-right})}$ at

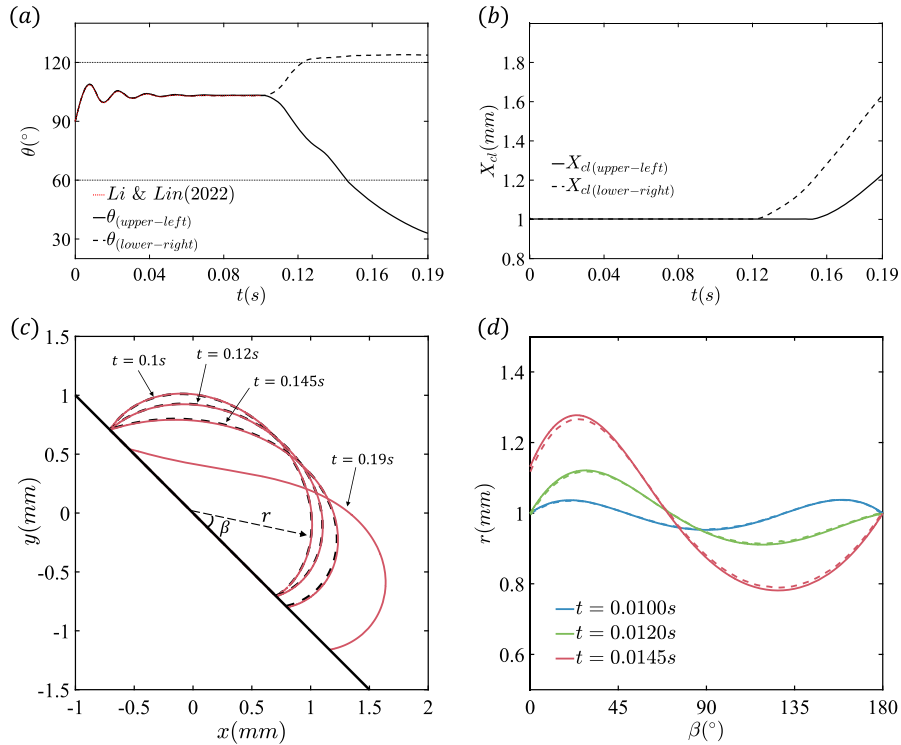


Fig. 22. (a), (b) Contact angles of the upper left and lower right contact lines and positions of the upper left (solid line) and lower right (dashed line) contact lines. The red dashed line represents the result from [24]. The black lines represent the result in the present work. The solid black line indicates the upper left contact line, while the black dashed line represents the lower right contact line. (c), (d) Droplet shapes at various characteristic times and quantitative comparison of droplet shapes. The definitions of r and β in panel (d) are shown in panel (c). — : Results in [24]; — : Results in the present work.

$t < 0.12$ s, before the lower-right contact line starts sliding. Similarly, $X_{cl(upper-left)}$ begins to move at $t \approx 0.145$ s, consistent with the observations in Fig. 22(a). Moreover, Fig. 22(b) also demonstrates that the proposed method maintains smooth sliding trajectories of the contact lines on complex geometrical boundaries. Finally, Fig. 22(c) presents snapshots of the droplet shapes from pinning to sliding and Fig. 22(d) presents a quantitative comparison of droplet shapes. In this comparison, r denotes the distance from each point on the droplet contour to the origin, while β represents the angle between the line connecting the droplet contour point and the origin and the solid boundary, as illustrated in panel (c). The maximum relative error, $|r - r_{Li}|/r_{Li}$, for the three contours is 0.014, where r_{Li} refers to the results reported in [24].

Additionally, mesh dependence is taken into account, and instances involving either the dynamic or static model are examined when the contact angle exceeds the hysteresis window, as demonstrated in Appendix C. The results indicate that mesh convergence is observed with the dynamic model, while it is not apparent with the static model, aligning with the observations in [43].

5.4. Axisymmetric impact of a droplet on a flat plate with an orifice

Up to now, we have considered droplet motion driven solely by capillary and gravitational effects. In this section, we simulate a droplet falling in an ambient fluid and impacting a plate with an orifice, where inertial and viscous effects become significant. The problem setup is illustrated in Fig. 23. Although it is a 3D problem, we employ a 2D axisymmetric model for the simulation and compare our results with experimental data from [44]. The physical properties are chosen based on experimental conditions: the droplet has a density of $\rho_d = 1130$ kg/m³ and a viscosity of $\mu_d = 0.007$ kg/(m · s), while the ambient fluid has a density of $\rho_f = 960$ kg/m³ and a viscosity of $\mu_f = 0.048$ kg/(m · s). The surface tension coefficient is $\sigma = 0.0295$ N/m, and the gravitational acceleration is $g = 9.8$ m/s². The liquid pool's top is 120 mm from the fixed plate, which has a thickness of $s = 2$ mm. The droplet, with a diameter D , is released 110 mm above the orifice, which has a diameter d with either a round or sharp edge, leading to different scenarios during passage. The plate has no-slip boundary conditions and is placed 12.5 mm from the right boundary of the computational domain, allowing ambient fluid to rise when the droplet interacts with the plate. The mesh size used in computations is $\Delta = 3.91 \times 10^{-2}$ mm, providing a grid resolution of approximately 120 cpr. In experiments, no contact is observed between the droplet and a plate with a round orifice, while contact line development and hysteresis are noted with a sharp orifice [44].

In simulating the impact with a round orifice, the initial droplet diameter is $D = 9.315$ mm, and the Bond number is $Bo = 4.9$. The round orifice is modeled as a semicircle with a diameter of $s = 2$ mm, and the orifice diameter is $d = 6$ mm, corresponding to $d/D = 0.644$. The experimental setup uses a contact angle of $\theta_s = 55^\circ \pm 1.5^\circ$ on the plate, while our numerical simulations enforce a

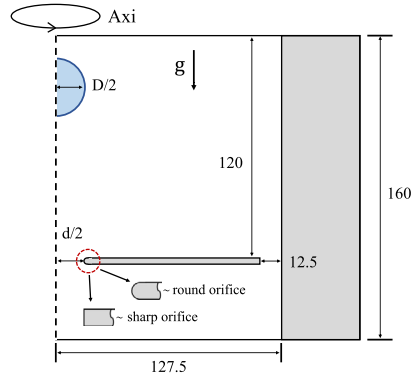


Fig. 23. Schematic representation of droplets impacting a flat plate with a round or sharp orifice (axial symmetry, length unit: mm).

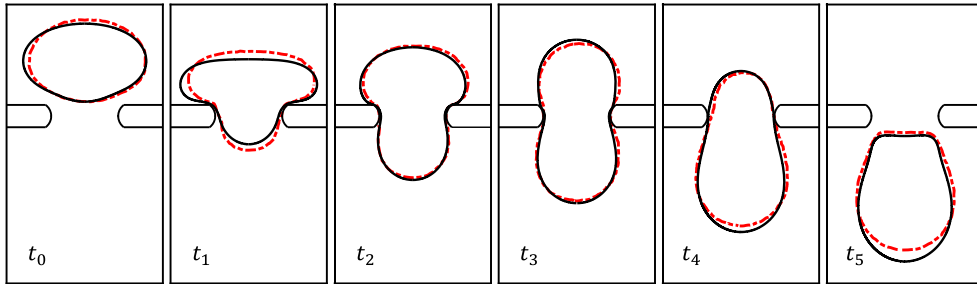


Fig. 24. Shapes of droplets impacting on the plate with a round orifice. $t_n (n = 0, 1, 2, 3, 4, 5) = t/t_g = 0, 0.54, 1.27, 1.81, 2.44$ and 2.96 . $t_g = (\rho_l D / (\rho_l - \rho_g) g)^{1/2}$. - - - - : Experiment in [44]; —: Simulation.

contact angle of $\theta_s = 180^\circ$ to avoid potential contact between the droplet and the plate. The time scale for the problem is defined as $t_g = (\rho_l D / (\rho_l - \rho_g) g)^{1/2}$, based on the gravitational force.

The numerical results and experimental observations are presented in Fig. 24, with dimensionless times $t = t_0$ to $t = t_5$ normalized by t_g . Here, t_0 is the initial time of collision, 10 ms before the droplet reaches the plate's upper surface. The dimensionless time points are $t_0 = 0$, $t_1 = 0.54$, $t_2 = 1.27$, $t_3 = 1.81$, $t_4 = 2.44$, and $t_5 = 2.96$. At all times, the droplet shapes align well with experimental measurements. Notably, due to the enforced contact angle of 180° on the plate surface, the liquid/liquid interface conforms to the arc-shaped outline of the round orifice without penetrating into the solid interior during $t_1 \leq t \leq t_4$.

On the other hand, in case of sharp orifice, numerical settings still align with the experiments: the initial diameter of the droplet is $D = 10.307$ mm, the Bond number is $Bo = 6.0$, and the diameter of the sharp orifice is $d = 6$ mm corresponding to $d/D = 0.58$. Note that the outcomes in this case are seriously influenced by contact angle hysteresis, as reported by experiments. Particularly, the advancing and receding angles of the plate are respectively $\theta_a = 68^\circ$ and $\theta_r = 46^\circ \pm 4^\circ$ in experiments, and accordingly, we also set the hysteresis angles to be $(\theta_r, \theta_a) = (42^\circ, 68^\circ)$ in the simulation that if the contact angles exceed the hysteresis window, they are enforced to the advancing or receding contact angles. In addition, the experiments reported a contact line pinning phenomenon at the sharp edge of the orifice, while reproducing this scenario is rather complicated in simulations. Pathak et al. [45] proposed a method for such motion of the contact line at the sharp edge, and alternatively, the present study artificially enlarges the advancing contact angle to $\theta_a = 150^\circ$ at the sharp edge region, where that the pinned phenomenon is produced numerically.

Additionally, a passage time scale is defined as $t_i = D^3 / (U_i d^2)$, where U_i is the falling velocity of the droplet 10 ms before impacting. The dimensionless time points are $t_1 = 0.49$, $t_2 = 0.73$, $t_3 = 0.90$, and $t_4 = 1.04$. The results are presented in Fig. 25, panel (a) plots the time histories of the position of the leading and the trailing interface lying on the axis, as depicted by points M and N given in panel (b), which shows the snapshots of the droplet shapes during its passage across the orifice. Besides, the experimental results are also attached in both pictures for quantitative comparison, and good agreements are still observed between them.

Moreover, Fig. 25 illustrates the entire process of the droplet passing through the orifice. In the early stage of impact ($t < t_1$), we observe that the position of N initially decreases and then increases. When a droplet with a larger radius passes through an orifice, it is "squeezed" by the orifice, causing the droplet to be elongated in the direction of impact. As the squeezing effect weakens, the trailing interface continues to decrease until $t = t_2$, when the contact line appears on the upper surface. The contact line is pinned at the upper sharp edge due to the larger advancing contact angle. During the period $t_2 < t < t_3$, the contact line on the trailing surface moves completely into the sharp orifice. At $t = t_3$, the trailing surface forms a crescent shape, resembling the experimental observation. Subsequently, the trailing surface in the cavity rebounds above the upper surface, causing the position of N to increase again at $t > t_3$. Finally, at $t = t_4$, the droplet below the lower surface of the plate is nearly disconnected from the droplet in the orifice, with the contact line pinned at the lower sharp edge. These numerical results demonstrate the ability of the proposed algorithm to accurately simulate the evolution of contact lines during dynamic impact processes.

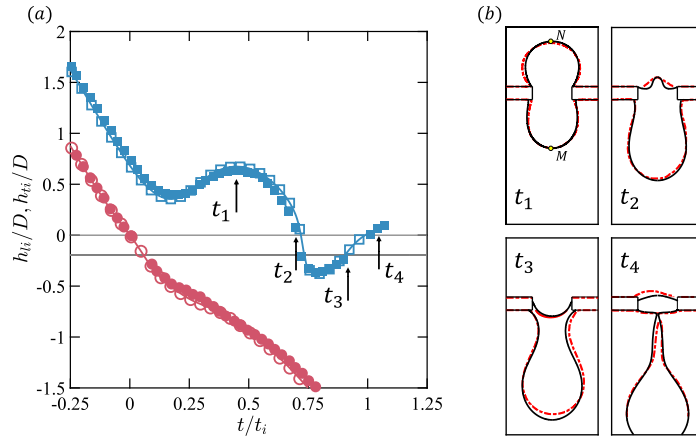


Fig. 25. (a) Locations of the center of leading (h_{li} , red, point M in panel (b)) and trailing (h_{ti} , blue, point N in panel (b)) interfaces during droplet impacting a plate with a sharp orifice. Two gray lines represent the upper and lower surfaces of the flat plate, respectively. \blacksquare/\bullet : Experiment in [44]; \square/\circ : Simulation. (b) Shapes of droplets impacting on the plate with a sharp orifice. $t_n (n = 1, 2, 3, 4) = t/t_i = 0.49, 0.73, 0.90$ and 1.04 . $t_i = D^3/(U_i d^2)$, where U_i is the falling velocity of the droplet 10 ms before impacting. $---$: Experiment in [44]; $---$: Simulation.

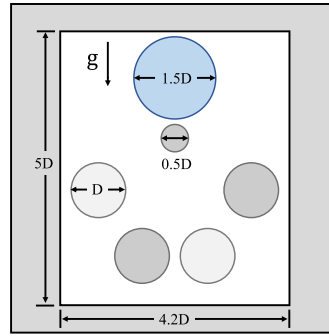


Fig. 26. Penetration of a liquid drop into porous media. The contact angle θ is set to 90° at the rectangular box, 30° at the lighter gray cylinders, and 150° at the darker gray cylinders.

5.5. Penetration of a liquid drop into porous media with large porosity

The next example involves a droplet penetrating into porous media, originally designed by Liu et al. [20]. Fig. 26 illustrates this problem, which features a more complex geometrical structure than those in the previous tests. The computations are performed in a $4.2D \times 5D$ rectangular domain with the lower left corner located at $(0, 0)$. The droplet, with a diameter of $1.5D$, is initially positioned at $(2.1D, 4.15D)$, while four cylinders with a diameter of D are located at $(0.7D, 2.1D)$, $(3.5D, 2.1D)$, $(1.5D, 0.9D)$, and $(2.7D, 0.9D)$, respectively. The contact angles on these cylinders are prescribed as 30° , 150° , 150° , and 30° , respectively. Additionally, a fifth cylinder with a diameter of $0.5D$ is positioned at $(2.1D, 3.05D)$ with a contact angle of 150° , and all boundaries of the rectangular box have a contact angle of 90° . In the figure, lighter and darker gray cylinders correspond to contact angles of 30° and 150° , respectively. The physical properties are set as follows: $\rho_l = 1000 \text{ kg/m}^3$, $\rho_g = 1 \text{ kg/m}^3$, $\mu_l = 0.001 \text{ kg/(m} \cdot \text{s)}$, $\mu_g = 2.5 \times 10^{-5} \text{ kg/(m} \cdot \text{s)}$ and $\sigma = 0.0025 \text{ N/m}$. The Weber number is set to $We = \rho_l u_0^2 1.5D / \sigma = 1.5$, the Reynolds number to $Re = \rho_l u_0 1.5D / \mu_l = 75$, and the Froude number to $Fr = u_0 / \sqrt{g 1.5D} = 1/\sqrt{6}$, where u_0 is the initial falling velocity of the droplet.

The grid resolution in the simulations is set to 64 cpr, and the results are shown in Fig. 27 over time. The figure indicates that at $t = t_1$, the droplet makes contact with the smallest hydrophobic cylinder and is nearly separated into two symmetrical daughter droplets, which then move toward the hydrophilic (left) and hydrophobic (right) cylinders, respectively. At $t = t_2$, when the daughter droplets reach the two cylinders, the contact line on the hydrophilic cylinder spreads rapidly, almost completely wrapping the cylinder, while the droplet on the hydrophobic cylinder continues to move downward. From $t = t_3$, the right-half daughter droplet falls onto the lower right hydrophilic cylinder and begins to spread, eventually wrapping it completely at $t > t_5$. Simultaneously, a splashing droplet occurs on the small cylindrical surface as the two daughter droplets separate, and the left daughter droplet fully wraps the hydrophilic cylinder at $t > t_4$. This entire interaction-spreading-separating process closely resembles that reported by Liu et al. [20], demonstrating the effectiveness of the present numerical method in simulating such penetration problems.

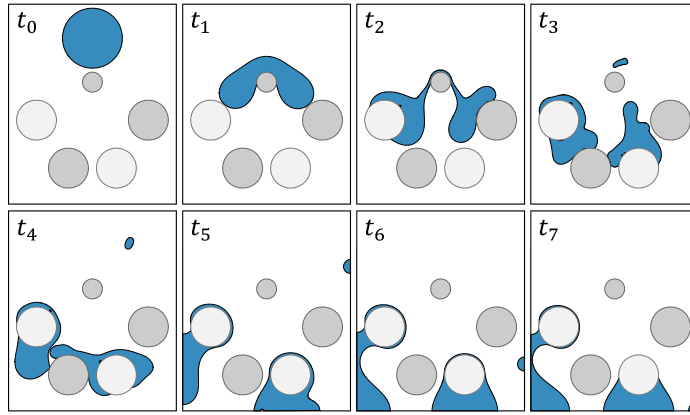


Fig. 27. Snapshots of droplet penetration in porous media at various times $t_0 = 0$ s, $t_1 = 0.013$ s, $t_2 = 0.020$ s, $t_3 = 0.027$ s, $t_4 = 0.035$ s, $t_5 = 0.064$ s, $t_6 = 0.130$ s and $t_7 = 0.360$ s.

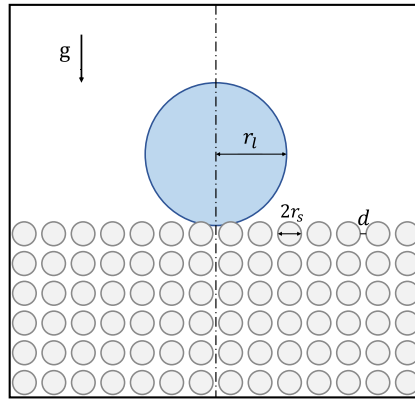


Fig. 28. Penetration of a liquid drop into porous media.

5.6. Penetration of a liquid drop into porous media with small porosity

The final example involves a droplet penetrating a porous medium, as illustrated in Fig. 28. The simulation is carried out in a rectangular domain $17.5r_s \times 33r_s$, with a symmetric boundary condition applied to the left boundary. The droplet, with a diameter of $r_l = 2$ mm, initially intersects partially with the solid cylinders. Each solid cylinder has a radius of $r_s = r_l/6$ and is arranged equidistantly with a spacing of $d = r_s/2$. The contact angle on all cylinders is set to $\theta = 30^\circ$. The physical properties of the fluids are defined as follows: $\rho_l = 1000$ kg/m³, $\rho_g = 1$ kg/m³, $\mu_l = 0.001$ kg/(m · s), $\mu_g = 2 \times 10^{-5}$ kg/(m · s), and $\sigma = 0.072$ N/m. The gravitational acceleration is set to $g = 9.8$ m/s², acting vertically downward. The grid resolution in the simulations is 186 cpr, resulting in approximately 15.5 cells within the gap between adjacent solid cylinders.

The results are presented in Fig. 29, showing the progression from $t_0 = 0$ s to $t_{10} = 0.0686$ s. Panel (a) magnifies the shapes of some liquid/gas interfaces at t_1 and t_8 , while panel (b) displays the interface shapes at other times. As shown in Fig. 29, at $t = t_1$, the liquid/gas interface between cylinder 1 and cylinder 2 forms an inverted U-shape due to the symmetric boundary condition, with the droplet primarily penetrating downward. By t_2 , the droplet contacts cylinder 3, initiating a rapid horizontal movement of the contact line and generating an upward-moving capillary wave. From t_3 to t_4 , the droplet contacts cylinder 4 in the center of the third layer and begins to penetrate vertically into this layer of the porous structure. During this stage, the contact line moves both horizontally and vertically, leading to a multidirectional shape at t_4 . Between t_4 and t_6 , because the liquid does not come into contact with the cylinders of the fourth layer, it spreads mainly horizontally. At t_7 , the droplet starts penetrating into the fourth layer, expanding horizontally and vertically simultaneously. At t_8 , the liquid/gas interface assumes a symmetric shape around cylinder 5 and its adjacent cylinders 6 and 7. By t_9 , the droplet almost fully penetrates the porous structure and eventually stabilizes at t_{10} , without further downward penetration. The permeation shape at t_{10} closely resembles the 3D porous structure permeation simulation by [46]: the liquid within the porous structure assumes an almost rectangular shape, forming an inclined interface between cylinders 8 and 9.

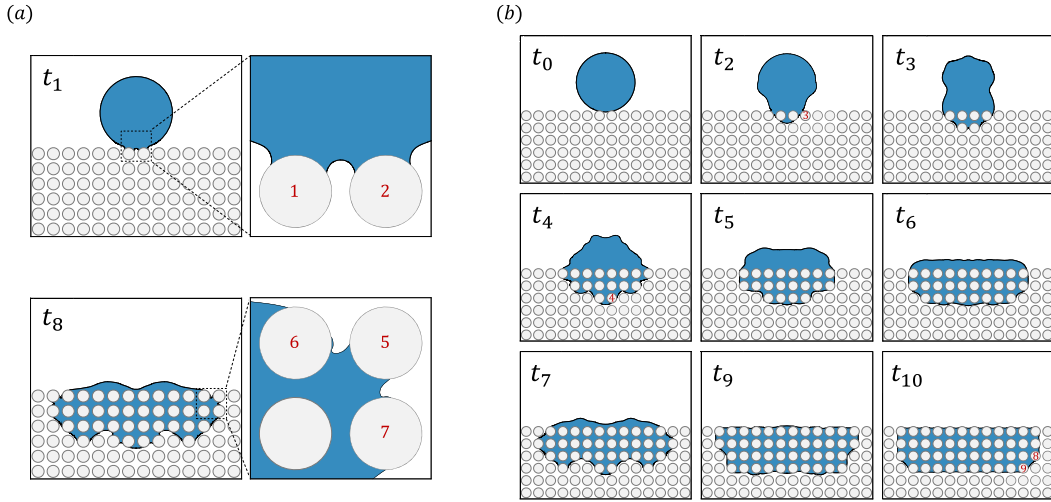


Fig. 29. Snapshots of droplet penetration in porous media at various times $t_0 = 0$ s, $t_1 = 0.0004$ s, $t_2 = 0.0028$ s, $t_3 = 0.0064$ s, $t_4 = 0.0148$ s, $t_5 = 0.0174$ s, $t_6 = 0.0228$ s, $t_7 = 0.0298$ s, $t_8 = 0.0315$ s, $t_9 = 0.0369$ s and $t_{10} = 0.0686$ s.

6. Conclusions

This paper proposes a 2D sharp interface method for addressing the contact angle condition between a liquid/gas interface and a complex boundary, incorporating considerations for contact angle hysteresis and dynamic contact angle models. The methodology employs a second-order accurate embedded boundary method to handle the embedded boundaries Γ_s in mixed cells, while utilizing the VOF method to track and reconstruct the liquid/gas interfaces Γ_l . This method contributes by focusing on conserving fluid mass when solving the volume fraction equation in mixed cells and by extending the application of the height function method to accurately impose contact angles on embedded boundaries. Finally, the method is implemented in the *Basilisk* code, which solves the Navier-Stokes equations, including the surface tension force.

As our first contribution, we propose a novel geometrical advection method aimed at conserving fluid mass during the update of the liquid/gas interface in mixed cells. The key insight is that in interfacial cells, the presence of embedded boundaries can lead to over- or under-estimation of volume flux through cell faces using ordinary geometrical advection schemes. Therefore, a volume compensation scheme is employed to rectify these discrepancies. Additionally, following each time step update of the liquid volume fraction, we utilize the flood algorithm to aid in reconstructing the piecewise linear liquid/gas interface within mixed cells with irregular geometries. Furthermore, to alleviate time step constraints when solving the volume fraction equation in small mixed cells, a numerical technique is applied to re-initialize the embedded boundaries surrounding these small cells.

As our second contribution, we highlight that the ordinary height function method, traditionally used to impose the contact angle condition on regular solid boundaries, is not directly applicable to complex geometries due to the presence of embedded boundaries. This complicates the determination of ghost height function values in these solid cells. To address this challenge, we propose a parabola fitting scheme in this study to obtain ghost HF values with higher accuracy, facilitating precise imposition of the contact angle on embedded boundaries compared to the original linear fitting scheme. Furthermore, we implement algorithms for contact angle hysteresis based on volume conservation and Cox's dynamic contact angle model. These additions allow for simulation of contact line dynamics on complex geometries using our numerical method.

Then, a series of benchmark tests is conducted in the present study to validate the proposed methods. First, the accuracy and conservation properties of the advection scheme are verified through an advection test under a specified velocity field, similar to the Zalesak disk problem but with an embedded boundary and an associated contact angle. Particularly under a divergence-free velocity field, any loss of liquid mass in the numerical simulation is reduced to the level of machine error. Next, a series of test cases explore droplet spreading on embedded boundaries which are parallel to the x -axis to demonstrate the accuracy of the contact angle condition imposed using the proposed height function method. Specifically, employing the parabola fitting method to determine ghost HF values shows that the equilibrium droplet radius aligns closely with theoretical solutions. In contrast, using the linear fitting method, rather than the proposed parabola fitting method, results in significant errors which are highly dependent on the position of the embedded boundary. Following this, the accuracy of the present method is further confirmed through additional spreading tests on various more complex geometrical boundaries. Results from droplet spreading on inclined plates or cylinders consistently match theoretical predictions. Additionally, a test involving a droplet sliding down a flat plate at 45° to the x -axis verifies the reliability of the contact angle hysteresis algorithm and the application of the dynamic contact angle model. The scheme demonstrates excellent accuracy in capturing intermediate angles within the hysteresis window. Finally, examples of droplets impacting flat plates with round or sharp orifices and penetrating into porous media validate the superiority of the proposed method.

In summary, this sharp VOF-based method addresses the longstanding issues related to the contact line moving on complex geometries and the code is released in <http://Basilisk.fr/sandbox/Chongsen/>. Our future work involves extending this 2D method to

3D situations, which are significantly more challenging due to the more complex interactions between the liquid/gas interface and the embedded boundaries. Detailed numerical techniques to handle the 3D problems will be presented in a separate paper.

CRedit authorship contribution statement

Chong-Sen Huang: Writing – original draft, Visualization, Validation, Software, Methodology, Data curation. **Tian-Yang Han:** Writing – original draft, Visualization, Validation, Methodology. **Jie Zhang:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Conceptualization. **Ming-Jiu Ni:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors gratefully acknowledge the supports from National Key R&D Programmes of China (2023YFA1011000, 2022YFE03130000) and from the NSFC (12222208, 11988102).

Appendix A. Time consumption for the current advection scheme

Compared to the traditional VOF advection scheme, the current VOF advection method ensures local volume conservation within mixed cells but necessitates additional interface reconstruction processes. Assessing the computational cost of the current method is therefore crucial. To this end, we compare the performance of the current scheme with the traditional VOF advection method using the example described in § 5.1.1, where a liquid ring rotates around a cylinder. The volume fraction conservation error of the liquid ring during one rotation cycle was calculated as a function of CPU time, with both methods employing the same time step for the simulation. As in § 5.1.1, three errors are defined as follows:

$$L_2 = \sqrt{\frac{\sum_i^N (c_{si} - c_i)^2}{N}}, \quad (\text{A.1})$$

$$L_{inf} = \max_{N,i} (|c_{si} - c_i|), \quad (\text{A.2})$$

$$E_V = \frac{|V - V_0|}{V_0}. \quad (\text{A.3})$$

The results are presented in Fig. A.30. The traditional VOF advection method completes one rotation in approximately 125 s, with L_{inf} more than 10^{-2} . In contrast, the current method requires approximately 165 s and achieves L_{inf} of about 10^{-13} . Although the current scheme increases CPU time by approximately 32% compared to the traditional VOF advection method, it significantly enhances accuracy, reducing errors by several orders of magnitude. The additional time consumption of the proposed scheme, compared to traditional geometric VOF advection, primarily stems from two factors: (a) the correction of the advection interval width un^* in all mixed cells (see § 3.2) and (b) the computation of geometrically reconstructed fluxes in mixed cells. While we consider this trade-off between computational cost and accuracy acceptable in 2D, extending the method to 3D scenarios would likely incur prohibitive computational expenses, potentially necessitating the development of a simplification algorithm. Addressing this computational challenge remains a key focus of our ongoing research.

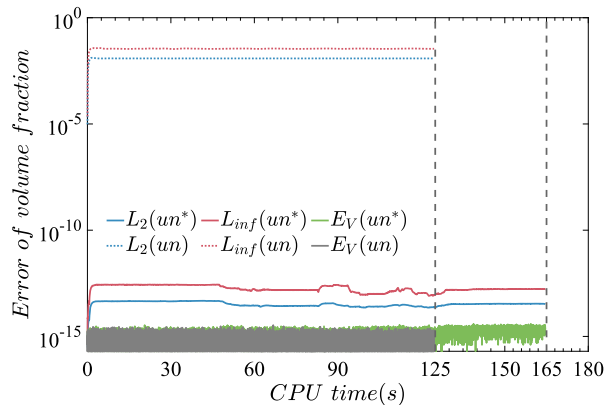


Fig. A.30. Error in local and global volume conservation during one rotation cycle. un represents the traditional VOF advection method; un^* represents the current VOF advection scheme.

Table B.2
Partial properties of the four examples.

| | $\theta(^{\circ})$ | ρ_l/ρ_g | μ_l/μ_g | d |
|-------|--------------------|-----------------|---------------|-----|
| case1 | 70 | 1000 | 40 | 0.2 |
| case2 | 70 | 1000 | 40 | 0.8 |
| case3 | 110 | 100 | 10 | 0.2 |
| case4 | 110 | 100 | 10 | 0.8 |

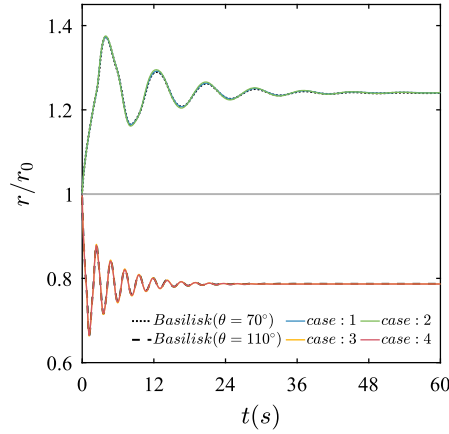


Fig. B.31. Dimensionless radius of droplet spreading on a flat plate of four cases. The dashed line and the dotted line are the regular boundary cases simulated by Basilisk.

Appendix B. Droplet spreading on a flat plate with different density and viscosity ratios

In this section, we provide additional examples of droplet spreading under different density and viscosity ratios. The setup of the examples is identical to that described in § 5.2.1 with $\beta = 0^{\circ}$, except for variations in density and viscosity. The physical parameters for these cases are summarized in Table B.2. Specifically, for Case 1 and Case 2, the liquid density $\rho_l = 100 \text{ kg/m}^3$, the gas density $\rho_g = 0.1 \text{ kg/m}^3$, the liquid viscosity $\mu_l = 0.03 \text{ kg/(m} \cdot \text{s)}$ and the gas viscosity $\mu_g = 7.5 \times 10^{-4} \text{ kg/(m} \cdot \text{s)}$. For Case 3 and Case 4, the liquid density $\rho_l = 10 \text{ kg/m}^3$, the gas density $\rho_g = 0.1 \text{ kg/m}^3$, the liquid viscosity $\mu_l = 7.5 \times 10^{-3} \text{ kg/(m} \cdot \text{s)}$, and the gas viscosity $\mu_g = 7.5 \times 10^{-4} \text{ kg/(m} \cdot \text{s)}$. The dimensional radii r/r_0 for all four cases are shown in Fig. B.31, and the results align well with the case involving a regular boundary.

Appendix C. Mesh-dependent example using a dynamic model or a static model

To assess the grid dependence of the scenario described in § 5.3.2, which features a droplet moving along an inclined embedded boundary, we examine various mesh resolutions while applying either a dynamic model or a static model when the contact angle

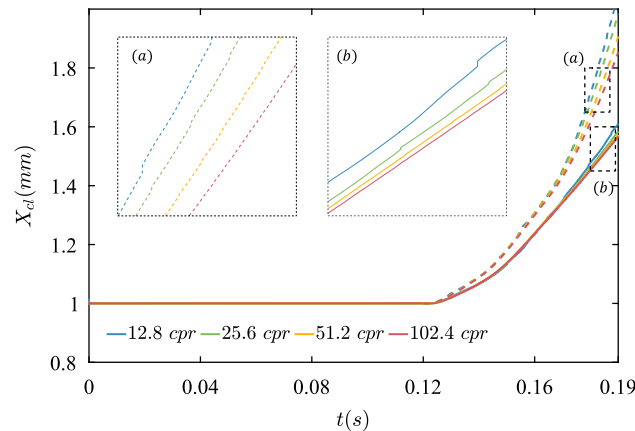


Fig. C.32. Positions of the lower right contact lines at grid resolutions of 12.8, 25.6, 51.2, and 102.4 cpr. The solid lines represent the examples using the dynamic model and the dashed lines represent that using the static model when the contact angle exceeds the hysteresis window.

exceeds the hysteresis window. We explore four grid resolutions (12.8, 25.6, 51.2, and 102.4 cpr) and record the sliding distance of the lower-right contact line.

Fig. C.32 displays the results. The solid lines represent the examples using the dynamic model and the dashed lines represent those using the static model when the contact angle exceeds the hysteresis window. With the application of the dynamic model, the slide distance progressively aligns with that of the highest grid resolution as the mesh becomes finer, demonstrating grid convergence. In contrast, results from examples with the static model show no convergence, which is consistent with the observation in [43].

Data availability

Data will be made available on request.

References

- [1] Peter D.M. Spelt, A level-set approach for simulations of flows with multiple moving contact lines with hysteresis, *J. Comput. Phys.* 207 (2) (2005) 389–404.
- [2] Chen Fang, Carlos Hidrovo, Fumin Wang, Julie Steinbrenner, Eonsoo Lee, Dongrip Kim, Kenneth Goodson, 3-d numerical simulation of contact angle hysteresis for slug flow in microchannel, in: *International Conference on Nanochannels, Microchannels, and Minichannels*, vol. 4272, 2007, pp. 955–963.
- [3] Hang Ding, Peter D.M. Spelt, Wetting condition in diffuse interface simulations of contact line motion, *Phys. Rev. E* 75 (4) (2007) 046708.
- [4] Jean-Baptiste Dupont, Dominique Legendre, Numerical simulation of static and sliding drop with contact angle hysteresis, *J. Comput. Phys.* 229 (7) (2010) 2453–2478.
- [5] Cyril W. Hirt, Billy D. Nichols, Volume of fluid (vof) method for the dynamics of free boundaries, *J. Comput. Phys.* 39 (1) (1981) 201–225.
- [6] Ruben Scardovelli, Stéphane Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* 31 (1) (1999) 567–603.
- [7] Mark Sussman, Peter Smereka, Stanley Osher, A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* 114 (1) (1994) 146–159.
- [8] Ruben Scardovelli, Stéphane Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* (1999).
- [9] E. Aulisa, S. Manservigi, R. Scardovelli, S. Zaleski, Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry, *J. Comput. Phys.* 225 (2) (2007) 2301–2319.
- [10] Stanley Osher, Ronald P. Fedkiw, Level set methods: an overview and some recent results, *J. Comput. Phys.* 169 (2) (2001) 463–502.
- [11] Elin Olsson, Gunilla Kreiss, A conservative level set method for two phase flow, *J. Comput. Phys.* 210 (1) (2005) 225–246.
- [12] Charles S. Peskin, Flow patterns around heart valves: a numerical method, *J. Comput. Phys.* 10 (2) (1972) 252–271.
- [13] Rajat Mittal, Gianluca Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* 37 (2005) 239–261.
- [14] Hans Johansen, Phillip Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, *J. Comput. Phys.* 147 (1) (1998) 60–85.
- [15] Arthur R. Ghigo, Stéphane Popinet, Anthony Wachs, A conservative finite volume cut-cell method on an adaptive Cartesian tree grid for moving rigid bodies in incompressible flows, 2021.
- [16] Adam O'Brien, Markus Bussmann, A volume-of-fluid ghost-cell immersed boundary method for multiphase flows with contact line dynamics, *Comput. Fluids* 165 (2018) 43–53.
- [17] H.V. Patel, S. Das, J.A.M. Kuipers, J.T. Padding, E.A.J.F. Peters, A coupled volume of fluid and immersed boundary method for simulating 3d multiphase flows with contact line dynamics in complex geometries, *Chem. Eng. Sci.* 166 (2017) 28–41.
- [18] Mathilde Tavares, Christophe Josserand, Alexandre Limare, José Ma Lopez-Herrera, Stéphane Popinet, A coupled VOF/embedded boundary method to model two-phase flows on arbitrary solid surfaces, *Comput. Fluids* 278 (2024) 106317.
- [19] Mathieu Lepilliez, Elena Roxana Popescu, Frederic Gibou, Sébastien Tanguy, On two-phase flow solvers in irregular domains with contact line, *J. Comput. Phys.* 321 (2016) 1217–1251.
- [20] Hao-Ran Liu, Hang Ding, A diffuse-interface immersed-boundary method for two-dimensional simulation of flows with moving contact lines on curved substrates, *J. Comput. Phys.* 294 (2015) 484–502.
- [21] Michael Renardy, Yuriko Renardy, Jie Li, Numerical simulation of moving contact line problems using a volume-of-fluid method, *J. Comput. Phys.* 171 (1) (2001) 243–263.
- [22] S. Afkhami, M. Bussmann, Height functions for applying contact angles to 2D vof simulations, *Int. J. Numer. Methods Fluids* 57 (4) (2008) 453–472.
- [23] Savinien Pertant, Manuel Bernard, Giovanni Ghigliotti, Guillaume Balarac, A finite-volume method for simulating contact lines on unstructured meshes in a conservative level-set framework, *J. Comput. Phys.* 444 (2021) 110582.
- [24] Jing Li, Zhiliang Lin, A height function based momentum balance model to simulate contact angle dynamics with hysteresis, *Int. J. Numer. Methods Fluids* 95 (1) (2023) 1–22.
- [25] Jiaqi Zhang, Pengtao Yue, A level-set method for moving contact lines with contact angle hysteresis, *J. Comput. Phys.* 418 (2020) 109636.
- [26] Stéphane Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, *J. Comput. Phys.* 228 (16) (2009) 5838–5866.
- [27] Chorin, Alexandre Joel, Numerical solution of the Navier-Stokes equation, *Math. Comput.* 22 (104) (1968) 745–762.
- [28] David L. Brown, Ricardo Cortez, Michael L. Minion, Accurate projection methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.* (2001).
- [29] Jeremiah U. Brackbill, Douglas B. Kothe, Charles Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* 100 (2) (1992) 335–354.
- [30] Gabriel D. Weymouth, Dick K-P. Yue, Conservative volume-of-fluid method for free-surface simulations on Cartesian-grids, *J. Comput. Phys.* 229 (8) (2010) 2853–2865.
- [31] Marianne M. Francois, Sharen J. Cummins, Edward D. Dendy, Douglas B. Kothe, James M. Sicilian, Matthew W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, *J. Comput. Phys.* 213 (1) (2006) 141–173.
- [32] Vadim Dyadechko, Mikhail Shashkov, Moment-of-fluid interface reconstruction, *Los Alamos Report LA-UR-05-7571*, page 49, 2005.
- [33] Steven Diot, Marianne M. François, Edward D. Dendy, An interface reconstruction method based on analytical formulae for 2D planar and axisymmetric arbitrary convex cells, *J. Comput. Phys.* 275 (2014) 53–64.
- [34] Dezhi Dai, Albert Y. Tong, An analytical interface reconstruction algorithm in the plic-vof method for 2D polygonal unstructured meshes, *Int. J. Numer. Methods Fluids* 88 (6) (2018) 265–276.
- [35] Dezhi Dai, Albert Y. Tong, Analytical interface reconstruction algorithms in the plic-vof method for 3d polyhedral unstructured meshes, *Int. J. Numer. Methods Fluids* 91 (5) (2019) 213–227.
- [36] K.M.T. Kleefman, G. Fekken, A.E.P. Veldman, B. Iwanowski, B. Buchner, A volume-of-fluid based simulation method for wave impact problems, *J. Comput. Phys.* 206 (1) (2005) 363–393.
- [37] Phillip Colella, Daniel T. Graves, Benjamin J. Keen, David Modiano, A Cartesian grid embedded boundary method for hyperbolic conservation laws, *J. Comput. Phys.* 211 (1) (2006) 347–366.

- [38] Lennart Schneiders, Claudia Günther, Matthias Meinke, Wolfgang Schröder, An efficient conservative cut-cell method for rigid bodies interacting with viscous compressible flows, *J. Comput. Phys.* 311 (2016) 62–86.
- [39] Steven T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys.* 31 (3) (1979) 335–362.
- [40] Jieyun Pan, Tian Long, Leonardo Chirco, Ruben Scardovelli, Stéphane Popinet, Stéphane Zaleski, An edge-based interface tracking (EBIT) method for multiphase-flow simulation with surface tension, *J. Comput. Phys.* 508 (2024) 113016.
- [41] Douglas A. Olsen, Powell A. Joyner, Marvin D. Olson, The sliding of liquid drops on solid surfaces, *J. Phys. Chem.* 66 (5) (1962) 883–886.
- [42] Robert Tao-Ping Chow, et al., On the ability of drops or bubbles to stick to non-horizontal surfaces of solids, *J. Fluid Mech.* 137 (1983) 1–29.
- [43] Shahriar Afkhami, Stéphane Zaleski, Markus Bussmann, A mesh-dependent model for applying dynamic contact angles to vof simulations, *J. Comput. Phys.* 228 (15) (2009) 5370–5389.
- [44] Ankur D. Bordoloi, Ellen K. Longmire, Drop motion through a confining orifice, *J. Fluid Mech.* 759 (2014) 520–545.
- [45] Ashish Pathak, Wen Jin, Mehdi Raessi, A three-dimensional numerical scheme for modeling discontinuous pinning at sharp edges using the volume-of-fluid method, *J. Comput. Phys.* 479 (2023) 111986.
- [46] Saurish Das, H.V. Patel, E. Milacic, N.G. Deen, J.A.M. Kuipers, Droplet spreading and capillary imbibition in a porous medium: a coupled ib-vof method based numerical study, *Phys. Fluids* 30 (1) (2018).