A levelset-based cut-cell method for interfacial flows: part 2-free-surface flows and dynamic contact angle treatment

Alejandro Quirós Rodríguez, Tomas Fullana, Taraneh Sayadi, Vincent Le Chenadec

A levelset based cut-cell method for interfacial flows. Part 2: Free-surface flows and dynamic contact angle treatment

Alejandro Quirós Rodríguez[1*], Tomas Fullana[1,2], Taraneh Sayadi[1,3], Vincent Le Chenadec[4]

[1*]Institut Jean le Rond d'Alembert, Sorbonne Université, CNRS, Paris, F-75005, France.
[2]Laboratory of Fluid Mechanics and Instabilities, EPFL, Laussane, CH-1015, Switzerland.
[3]Mathematical and Numerical Modelling Laboratory, Conservatoire National des Arts et Métiers, Paris, F-75003, France.
[4]MSME, Université Gustave Eiffel, UPEC, CNRS, Marne-la-Vallée, F-77454, France.

*Corresponding author(s). E-mail(s): alejandro.quiros_rodriguez@etu.sorbonne-universite.fr;
Contributing authors: tomas.fullana@epfl.ch; taraneh.sayadi@lecnam.net; vincent.le-chenadec@univ-eiffel.fr;

Abstract

This second segment of a two-part study investigates the numerical modeling of the Navier slip boundary condition at the contact line at the junction of free and solid surfaces, a key element in many natural and technological processes. The first segment introduced a method based on the cut-cell formalism. This second segment demonstrates how this simple formulation can be used to flexibly mix Navier, no-slip and free-surface boundary conditions near sharp interfaces described using a levelset representation. The study emphasizes a unified treatment of boundaries and interfaces that retains the simplicity of the original methodology without requiring further simplifying assumptions. To that end, a two-levelset approach is employed, with one levelset defining the solid wall and the other defining the free-surface. The spreading of a droplet over straight and circular walls for different contact angles is used as validation cases. The expected

first-order accuracy in mass loss/gain is achieved, with the maximum error in mass conservation reaching 10% in the worst-case scenario.

Keywords: Cut-cell method, levelset method, interfacial flows, dynamic contact line

1 Introduction

The contact line that forms at the intersection of a fluid-fluid and a solid boundary is a key element of many natural and several technological processes [1]. The dynamics of the contact line has been a subject of debate for several decades [2–4]. The classical no-slip boundary condition at a moving fluid interface leads to a force singularity at the contact line, as demonstrated by Huh and Scriven [5], implying that contact line motion would be impossible. In the context of continuum mechanics, in sharp-interface models, one common approach is to introduce a small slip velocity at the contact line using the Navier boundary condition. The slip length, typically on the order of nanometers, is often associated with surface roughness or the mean free path of molecules [6]. The Navier slip model combined with an equilibrium contact angle on a mesh-aligned boundary has been extensively studied across various physical setups using different numerical methods. For instance, a rapidly advancing contact line in a curtain coating scenario was simulated in a Volume-of-Fluid (VOF) framework [7], the Phase-Field method and VOF were benchmarked against molecular dynamics in a sheared droplet system [8], different numerical methods with the same contact line model were tested in a capillary rise setup [9], a withdrawing plate was simulated using the VOF method with a mesh-dependent slip length [10], spreading droplets were explored with mesh-dependent slip [11], and the levelset method was employed for a spreading drop case, incorporating contact line hysteresis by varying the contact angle based on the contact line speed [12]. However, when dealing with an arbitrarily shaped solid, the tangential and normal velocity components as well as the imposed equilibrium contact angle depend on the local orientation of the solid boundary. In the recent work of Tavares et al. [13], a coupled VOF/embed numerical method with a height functions methodology for the curvature term is proposed to address this specific challenge. In this contribution, we provide a unified treatment of boundaries and interfaces for cases with contact lines on arbitrary geometry using the Navier boundary condition with a constant equilibrium angle as our choice of contact line model.

In the low Reynolds number regimes considered, the gas phase has a limited impact on the contact line dynamics, and the free-surface assumption is therefore applicable. Alternatively, the one-fluid model [14] can also be used, as recently done in related studies [13, 15–17]. The one-fluid method however models the two immiscible phases as a single fluid with discontinuous material properties, a representation that may not be suitable when discontinuities exist in the velocity fields.

Free-surface flows come with their own challenges. They are subjected to additional kinematic and dynamic boundary conditions at the (free) surface, resulting in

a numerical treatment that is more complex than for stationary surfaces. The kinematic condition states that in the absence of mass accumulation or transfer across the free-surface, the normal component of the velocity field matches the surface velocity. The dynamic boundary condition on the other hand states that, in the absence of mass accumulation or transfer, the force exerted on either of the interface must match up to the capillary force. The kinematic condition can be accommodated by Arbitrary Lagrangian Eulerian or Eulerian scheme. The latter is closely related to traction boundary condition, which have seen recent renewed interest from the literature [18].

The beginnings of numerical modelling of free-surfaces can be traced back to the numerical modelling of incompressible flows, and in particular the original work of Harlow and Welch [19]. To simplify the discretization of the dynamic boundary condition, a typical assumption is to neglect the viscous stress, or to assume that the interface is mesh-aligned [20]. In the low Reynolds number regime considered here, the viscous contribution to the total viscous stress at the interface cannot be neglected. Similarly, the dominant effect of the contact line dynamics dictates that the interface orientation must be taken into account.

One of the major challenges is then to switch from a Navier slip condition applied in the vicinity of the contact line, to a no-slip or traction boundary condition, depending on whether the vicinity of the solid or the free-surface is considered. Another challenge is the representation of the boundaries themselves (or interface, both terms will be used interchangeably in this manuscript to denote either one of the fluid/void and fluid/solid interfaces). Arbitrary Lagrangian Eulerian methods are certainty the most accurate way to accommodate such configurations [21], but they come with overhead induced by unstructured grids. Alternatively, structured grids (Cartesian, in particular), can be used, to keep the simplicity of linear data structures. They do however require an overhaul of the solution algorithm, usually based on the Immersed Boundary Method [22] and its variants [23].

In the first segment of this two-part contribution [24], one such method was proposed based on the cut-cell formalism. The objective of this second segment contribution is to show how this simple formulation can be used to flexibly mix Navier, no-slip and traction boundary conditions in the vicinity of sharp interfaces using a levelset representation [25]. Emphasis is set on a unified treatment of boundaries and interfaces that retains the simplicity of the original methodology without requiring further simplifying assumptions regarding the magnitude of the different terms or the orientation of the boundaries/interfaces. The aim in proposing such a simple set of primitives is to facilitate implementation in existing Cartesian solvers and to pave the way for numerical modelling of complex transfer phenomena in the vicinity of interfaces/boundaries.

Fig. 1 displays an example of the configuration that will be treated in this work. We see a liquid phase with a free-surface Γ_1 in contact with a solid wall at the interface Γ_2. Both interfaces intersect in the contact points denoted by the black dots. In the vicinity of these contact points at the wall, a Navier slip boundary condition will be applied. In the rest of the wall a no-slip condition will be imposed and the corresponding jump conditions will be applied at the free-surface. In the following, Γ_1 and Γ_2 will always refer to the free-surface and the solid wall, respectively.

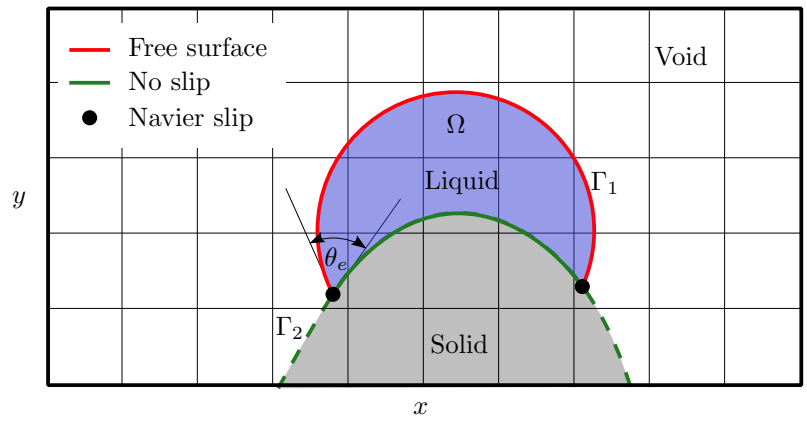


Fig. 1: Schematic representation of a free-surface flow with contact line dynamics on an arbitrarily oriented solid wall.

The manuscript is structured as follows. In Sec. 2, the continuous models are presented together with the boundary conditions. Sec. 3, the extension of the cut-cell methodology to two interfaces is discussed. This includes the definition of additional geometric moments, as well as new cut-cell operators for transport and boundary terms. In Sec. 4, validations and applications are finally presented.

2 Governing equations

2.1 Front capturing by the levelset method

In this work, a levelset function ϕ is defined on the computational domain Ω to map the locus of one of its iso-levels $(\{\boldsymbol{x} \in \Omega | \, \phi(t, \boldsymbol{x}) = \phi_0\})$ to an interface $\Gamma(t)$ that separates two non-overlapping open domains, $\Omega_1(t)$ and $\Omega_2(t)$, each occupied by a different phase. The value $\phi_0 \equiv 0$ is used here, and ϕ is defined as the signed distance to the interface,

$$\phi(t, \boldsymbol{x}) = \begin{cases} -d(\boldsymbol{x}, \Gamma(t)), \, \boldsymbol{x} \in \Omega_1(t), \\ 0, \, \boldsymbol{x} \in \Gamma(t), \\ d(\boldsymbol{x}, \Gamma(t)), \, \boldsymbol{x} \in \Omega_2(t), \end{cases} \tag{1}$$

where $d(\boldsymbol{x}, \Gamma(t))$ denotes the minimal distance between the point $\boldsymbol{x}$ and the interface $\Gamma(t)$,

$$d(\mathbf{x}, \Gamma(t)) = \text{argmin}_{\boldsymbol{y} \in \Gamma(t)} \|\boldsymbol{x} - \boldsymbol{y}\|.$$

Here, $\|\cdot\|$ denotes the Euclidean distance. The levelset function evolution can be shown [26] governed by the advection equation

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0 \tag{2}$$

where $\boldsymbol{u}$ represents a continuous extension of the interface velocity to the domain Ω.

To numerically solve this equation, an approach initially proposed in Mikula et al. [27] is used that consists in rewriting it as a second-order partial differential equation,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot \left(u_n \phi \frac{\nabla \phi}{\|\nabla \phi\|} \right) - \phi \nabla \cdot \left(u_n \frac{\nabla \phi}{\|\nabla \phi\|} \right) = 0, \tag{3}$$

where u_n denotes the extension of the velocity component normal to the interface. The velocity at the interface needs to be extended to its vicinity in order to accurately advect the levelset. To do this, we keep the normal velocity constant along the lines normal to the Γ interface, as suggested by Peng et al. [28]. This method has already been used in a previous version of the proposed cut-cell method by Fullana et al. [29] in the context of the Stefan problem.

A suitable spatial and temporal discretization of Eq. (3) was proposed by Mikula et al. [27], and shown to be more accurate than classical schemes, allowing larger time-steps to be used, and requiring less frequent reinitialization steps in order maintain the distance function property,

$$\|\nabla \phi\| = 1. \tag{4}$$

Although the need for reinitialization is reduced, it is not completely eliminated. Whenever necessary in this work, it is done by solving the following transient partial differential equation

$$\frac{\partial \phi}{\partial \tau} = S\left(\phi^0\right)\left(1 - \|\nabla \phi\|\right), \tag{5}$$

where τ denotes a pseudo-time, $S(\cdot)$ is the sign function and ϕ^0 is the levelset field prior to the reinitialization step.

Eq. (5) is integrated in time using the Total Variation-Diminishing second-order Runge-Kutta (TVD RK2) scheme developed in Shu and Osher [30], which essentially performs two Forward Euler steps. The spatial derivatives of the levelset are computed using the fifth-order Weighted Essentially Non-Oscillatory (WENO5) scheme derived in [31]. In the interface vicinity, a source term is added following the high-order constrained reinitialization (HCR2) method from [32] to ensure that the reinitialization does not modify the interface position. As mentioned previously, the levelset function is not reinitialized at every iteration. Luddens et al. [33] proposed the following criterion to detect the deformation of the levelset

$$r_g(\phi) = \left(|\nabla \phi| - 1\right)_{L^1} > \delta, \tag{6}$$

where δ is the reinitialization threshold, which has been implemented in the solver to automatize the reinitialization procedure.

2.1.1 Boundary conditions

In the vicinity of contact lines (intersection of the interface with a domain boundary), an inhomogeneous Neumann boundary condition is convenient to impose a target (equilibrium) angle θ_e. Such a boundary condition then takes the form

$$\frac{\partial \phi}{\partial n} = \cos \theta_e, \tag{7}$$

where $\partial \cdot / \partial n$ denotes the component of the scalar gradient normal to the boundary.

Away from the contact line, the definition of θ_e should be extended so as to minimize the deformation of the ϕ iso-levels. For example, in the case of a droplet on a straight surface in the absence of gravity, the angle θ can be extended analytically away from the contact line as follows

$$\theta = \begin{cases} \arctan\left[(1-\phi/d)\tan\theta_e\right] & \text{if } \theta_e < \pi/2, \\ \pi - \arctan\left[(1-\phi/d)\tan\left(\pi-\theta_e\right)\right] & \text{if } \theta_e > \pi/2, \end{cases} \tag{8}$$

where $2d$ denotes the distance between the two intersection points of the $\phi = 0$ iso-level with the solid wall. In two-dimensional configurations, this corresponds to half the distance between the two contact points, as seen in Fig. 2.

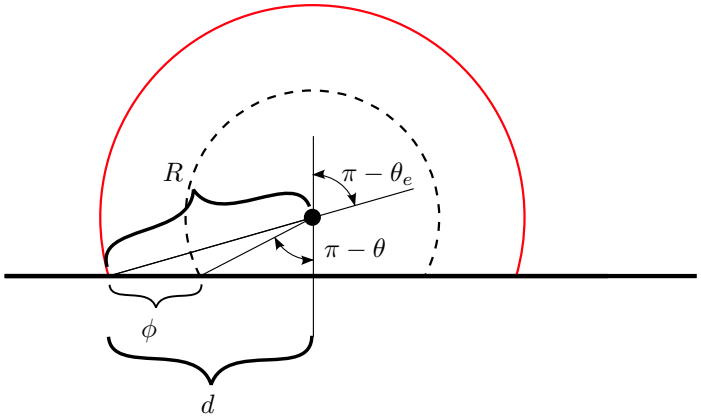


Fig. 2: Geometric definition of the parameter d used in Eq. (8) to define θ, the extension of the interface contact angle θ_e away from the contact line. The solid red line represents the interface Γ, the dashed black line represents the ϕ iso-level and the solid black line represents the solid wall.

2.2 Incompressible Navier-Stokes equations

This section introduces the non-dimensional incompressible Navier-Stokes equations governing the flow under study:

$$\begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} = -\nabla p + \dfrac{1}{\mathrm{Re}}\nabla\cdot(2\boldsymbol{s}) + \dfrac{1}{\mathrm{Fr}^2}\boldsymbol{g}, \\ \nabla\cdot\boldsymbol{u} = 0, \end{cases} \tag{9}$$

where the density has been considered to be $\rho = 1$ and $\boldsymbol{u}$, p, Re, Fr, $\boldsymbol{s}$ and $\boldsymbol{g}$ respectively denote the fluid's velocity, pressure, the Reynolds number, the Froude number, the

body force per unit mass and the strain rate tensor, defined as

$$\boldsymbol{s} \equiv \frac{\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\top}{2}. \tag{10}$$

Following [34], Eq. (9) can be solved using a pressure-projection method. The first step consists in updating a intermediate velocity field $\boldsymbol{u}^\star$, using the momentum equation, where the diffusive transport is solved using a Crank-Nicolson scheme and the convective transport using a second-order Adams-Bashforth integrator,

$$\frac{\boldsymbol{u}^\star - \boldsymbol{u}^n}{\Delta t} + [\boldsymbol{u} \cdot \nabla \boldsymbol{u}]^{n+1/2} = -\nabla p^{n-1/2} + \frac{1}{2\mathrm{Re}} \nabla^2 (\boldsymbol{u}^n + \boldsymbol{u}^\star) + \frac{1}{\mathrm{Fr}^2} \boldsymbol{g}. \tag{11}$$

Here, Δt refers to the time step. In general, the velocity field $\boldsymbol{u}^\star$ is not divergence-free, and the next step enforces this condition by first solving the Poisson equation

$$\Delta t \, \nabla^2 \psi^{n+1} = \nabla \cdot \boldsymbol{u}^\star, \tag{12}$$

for the intermediate pressure field ψ, prior to correcting the intermediate velocity field according to,

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^\star - \Delta t \, \nabla \psi^{n+1}. \tag{13}$$

Finally, the pressure is updated for the velocity prediction at the next time-step,

$$p^{n+1/2} = p^{n-1/2} + \psi^{n+1} - \frac{\Delta t}{2\mathrm{Re}} \nabla^2 \psi^{n+1}. \tag{14}$$

2.2.1 Free-surface boundary conditions

The free-surface boundary condition expresses the conservation of momentum across the interface Γ in the absence of accumulation or transfer of mass at the interface,

$$-p\boldsymbol{n} + \frac{2}{\mathrm{Re}} \boldsymbol{s} \cdot \boldsymbol{n} = -p_0 \boldsymbol{n} + \sigma \kappa \boldsymbol{n}, \tag{15}$$

where σ denotes the surface tension coefficient, κ the sum of the principal curvatures, $\boldsymbol{n}$ the unit vector normal to the interface Γ, and p_0 is the pressure outside the free-surface. This boundary condition neglects the thermal Marangoni effect [35], since the flow under consideration is assumed isothermal. By projecting Eq. (15) in the Cartesian directions $\{x, y\}$ we obtain a Neumann boundary condition for the velocity

$$-p\boldsymbol{e}_\alpha^\top \cdot \boldsymbol{n} + \frac{2}{\mathrm{Re}} \boldsymbol{e}_\alpha^\top \cdot \boldsymbol{s} \cdot \boldsymbol{n} = \sigma \kappa \boldsymbol{e}_\alpha^\top \cdot \boldsymbol{n}, \qquad \forall \alpha \in \{x, y\}, \tag{16}$$

where $\boldsymbol{e}_\alpha$ is a unitary vector in the α-direction.

2.2.2 Navier slip boundary condition

As described in the introduction, the Navier boundary condition is commonly used to model the dynamic contact line in sharp interface methods. At the continuous level, a

no-slip condition at a solid boundary simply anchors a contact line to a fixed location with no possibility to move. Numerically, the imposition of the boundary condition on the levelset for example can result in a contact line motion even when the no-slip condition is applied. However, this is not a satisfying solution, because the predicted dynamics depend both on the interface representation methodology and the grid used for the computation.

To address this issue, we allow the contact line to slip over the substrate with a certain slip length. The Navier slip boundary condition reads

$$\begin{cases} u_t - \lambda \dfrac{\partial u_t}{\partial n} = U_b, \\ u_n = 0 \end{cases} \tag{17}$$

where u_t and u_n denote the velocity components tangential and normal to the solid surface, respectively. The two parameters of the model are the slip length, λ and the slip velocity, U_b.

3 Numerical modeling of arbitrarily-oriented solid and free-surfaces and their intersection

3.1 Dealing with multiple levelset functions

The original methodology presented in the first segment [24] of this two-part contribution introduced simple yet consistent discrete operators for Cartesian grids to solve thermal and fluid flows between phases across a single interface or boundary. If more than one interface were to be considered, or if one of the interfaces is mesh-aligned, the formalism can be used as long as they do not intersect. However when arbitrary interfaces intersect (as is the case for contact line dynamics on non-flat surfaces as represented in Fig. 1), additional geometric moments and discrete operators need to be defined. The following section presents such an extension in the case of two interfaces, one representing a free-surface and the other a solid boundary. Particular attention is dedicated to the implementation of the different boundary conditions required in this particular configuration, and to retain the simplicity of the original methodology to ease implementation in other structured solvers.

3.1.1 Geometric moments

In the methodology presented in Part 1, a single set of volume $\{\mathcal{V}, \mathcal{W}_x, \mathcal{W}_y\}$ and surface $\{\mathcal{A}_x, \mathcal{A}_y, \mathcal{B}_x, \mathcal{B}_y\}$ geometric moments, defined for each set of control volumes, cell- (p) and face-centered (u_x and u_y), were defined in order to assemble the discrete cut-cell operators that were used to discretize the spatial derivatives (pressure gradient, velocity divergence, and convective and viscous transport terms). These geometric moments, extensively defined in Part 1 [24], served as the basis for a finite-volume approximation of conservation laws, and were shown to provide accurate estimates of the fluxes with neighboring fluid elements as well as across a boundary. The underlying rationale was that the surface moments $\mathcal{A}$ implicitly defined a piece-wise linear

representation of the interface. As might be expected, the definition of the moments requires some adapting if two interfaces are to be considered, since the linear approximation of the interface does not hold anymore; when two interfaces intersect within a cell, only part of the total length of the interface belongs to each phase. Therefore, using the surface moments $\mathcal{A}$ as they were defined in Part 1 would result in an incorrect estimate of the length of the interface for each phase.

In this case, a single set of volume moments still suffices, which will be denoted $\{\mathcal{V}, \mathcal{W}_x, \mathcal{W}_y\}$ as in Part 1. Regarding the surface moments, three different sets are required to implement the two-boundaries logic. The first set of surface moments will be denoted with a superscript Ω ($\{\mathcal{A}_x^\Omega, \mathcal{A}_y^\Omega, \mathcal{B}_x^\Omega, \mathcal{B}_y^\Omega\}$) and will be used in the definitions of the discrete operators $\{G, G^\top\}$ that account for the contribution of the bulk fields to the gradient and divergence. The other two sets of surface moments, denoted by the superscripts Γ_1 and Γ_2 (as in $\{\mathcal{A}_x^{\Gamma_i}, \mathcal{A}_y^{\Gamma_i}, \mathcal{B}_x^{\Gamma_i}, \mathcal{B}_y^{\Gamma_i}\}$, $i = 1$ or 2), will be used to define the discrete operators $\{H^{\Gamma_i}, H^{\Gamma_i,\top}\}$ that account for the contribution of the ith-boundary field to the gradient and divergence.

No special treatment is required for cut-cells intersected by a single interface. The interfacial surface moments (denoted by the superscript Γ_i, $i = 1$ or 2) are computed independently for each interface as in the original method and the bulk surface moments (denoted by the superscript Ω) are set to the interfacial surface moments of the interface that is cutting the cell. However, when both interfaces cut the same cell, the interfacial surface moments cannot be set directly to those obtained by treating each interface independently. In these cases, the surface moments obtained for each interface independently are denoted by a subscript 0 and are hereafter referred to as initial surface moments.

In the following, the volume of a cell corresponding to the liquid phase will be referred to as wet volume and the set of cells where this wet volume is non-zero will be referred to as wet cells.

Volume moments $\{\mathcal{V}, \mathcal{W}_x, \mathcal{W}_y\}$ and the surface moments $\{\mathcal{A}_x^\Omega, \mathcal{A}_y^\Omega, \mathcal{B}_x^\Omega, \mathcal{B}_y^\Omega\}$ are always calculated based on the actual wet cell obtained by intersecting the wet volumes corresponding to both interfaces. The wet volumes are calculated using the GEOS library [36], which allows the definition of all necessary geometric quantities. For this purpose, the vertices of the wet cells, together with the points where the linear representation of the interfaces intersects the cell surfaces, are passed to the GEOS library to define the actual wet volume. The calculations of the wet cell centroid and the intersection of the wet cell volume with horizontal and vertical lines to define the $\mathcal{B}^\Omega$ surface moments are also performed using the GEOS library.

The calculation of the missing surface moments in the cut-cells intersected by both interfaces requires further consideration. This is discussed below using the cases shown in Fig. 3, where the red and green shaded areas represent the wet volumes corresponding to domains 1 and 2 respectively. The blue shaded area represents the fluid domain, defined as the intersection of domains 1 and 2 (whose volume corresponds to $\mathcal{V}$). The solid lines are used to represent the parts of the interfaces that also belong to the other domain (*i.e.* $\Gamma_1 \cap \Omega_2$ in red and $\Gamma_2 \cap \Omega_1$ in green), while the dashed lines represent the parts of Γ_i that are outside the fluid domain (i.e. not part of Ω_j $\forall j \neq i$). Two different cases can be considered:

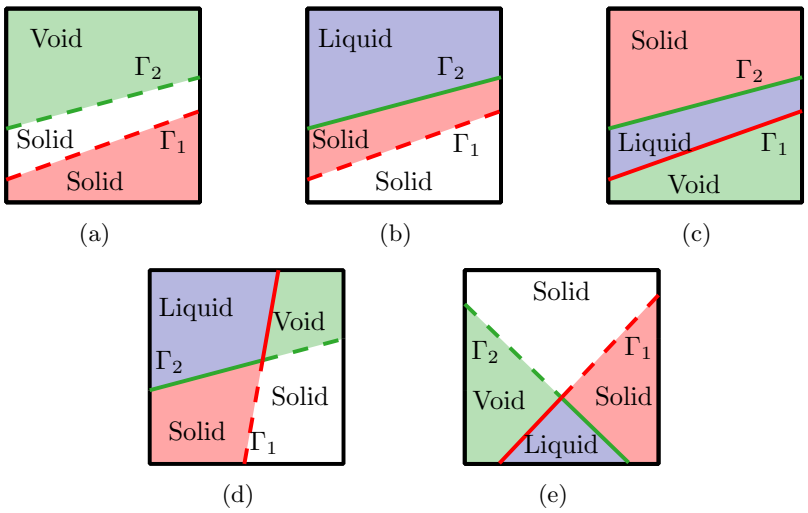


Fig. 3: Configurations relevant to the two-levelset extension. Solid lines denote interfaces adjacent to the liquid phase, where either free-surface (Γ_1) or no-slip (Γ_2) boundary conditions are applied. The intersection of Γ_1 and Γ_2 represents the contact point (cases displayed in Figs. 3d and 3e), where the Navier condition is applied. Dashed lines finally corresponds to fictitious interfaces, that are ignored because they are not in contact with the fluid.

1. **The interfaces do not intersect within the cell**. This leads to three possibilities:
   (a) **The intersection of the domains within the cell is empty**: the cell is not a wet cell and all the moments are set to zero. This corresponds to the case displayed in Fig. 3a.
   (b) **The intersection of the domains within the cell is non-empty and the restriction of one domain to the cell is a subset of the other**: the most restrictive interface keeps its initial values of the surface moments while the ones corresponding to the other interface are all set to zero since there is no contribution from this interface. In the example displayed in Fig. 3b, $\mathcal{A}^{\Gamma_1} = 0$ whereas $\mathcal{A}^{\Gamma_2} = \mathcal{A}_0^{\Gamma_2}$.
   (c) **The intersection of the domains within the cell is non-empty but no domain restricted to the cell is a subset of the other**: both interfaces keep their initial values, so $\mathcal{A}^{\Gamma_1} = \mathcal{A}_0^{\Gamma_1}$, $\mathcal{A}^{\Gamma_2} = \mathcal{A}_0^{\Gamma_2}$ as displayed in Fig. 3c.
2. **The interfaces intersect within the cell**, which leads to two possible cases in each direction for each interface:
   (d) **The section of the interface also belonging to the other domain ($\Gamma_i \cap \Omega_j$, for $j \neq i$) intersects one of the faces in that direction**: the intersected face keeps its initial value and the opposite face is set to the projection of the intersection point between both interfaces. In Fig. 3d, this is the case for the

green interface in the x-direction and for the red interface in the y-direction. In Fig. 3e, this is the case for both interfaces in the y-direction.

(e) **The section of the interface also belonging to the other domain ($\Gamma_i \cap \Omega_j$, for $j \neq i$) does not intersect any face in the direction**: in this case one of the surface moments is set to the projection of the intersection point and the opposite one is set to zero. In Fig. 3d, this is the case for the red interface in the x-direction and for the green interface in the y-direction. In Fig. 3e, this is the case for both interfaces in the x-direction.

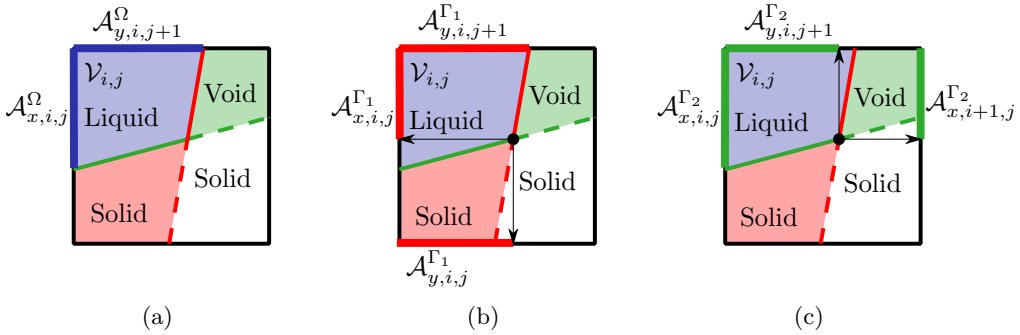


Fig. 4: Correction procedure to the $\mathcal{A}$ moments when the two boundaries intersect as in Fig. 3d (the case displayed in Fig. 3e follows).

The cell represented in Fig. 3d is also shown in Fig. 4 to clarify how the intersection point is employed to obtain the surface moments. Fig. 4a displays the bulk surface moments $\mathcal{A}^{\Omega}$, which are non-zero only at the left and top faces since those are the only faces being intersected by the actual wet cell in blue. Fig. 4b shows the interfacial surface moments $\mathcal{A}^{\Gamma_1}$. It can be seen that the intersection point is used to define the geometric moments corresponding to the left and bottom faces and that the top face keeps its initial moment. Finally, Fig. 4c presents the interfacial surface moments $\mathcal{A}^{\Gamma_1}$. In this case, the intersection point is used to define the surface moments corresponding to the top and right faces, whereas the left face keeps its initial moment.

The computation of the remaining surface moments is completed as follows: first the bulk moments $\mathcal{B}^{\Omega}_{\alpha}$ are computed by drawing a segment orthogonal to direction α passing through the intersection point. In order to keep certain properties of the discrete gradient and divergence operators, the $\mathcal{B}^{\Gamma_i}_{\alpha}$ have to be treated carefully. Two different cases can occur:

1. If $\mathcal{B}^{\Omega}_{\alpha}$ is larger than the two neighboring values of $\mathcal{A}^{\Gamma_i}_{\alpha}$ (in direction α), then $\mathcal{B}^{\Gamma_i}$ is set to the largest of the two.
2. If $\mathcal{B}^{\Omega}_{\alpha}$ is smaller than any of the two neighboring values of $\mathcal{A}^{\Gamma_i}_{\alpha}$ (in direction α), then $\mathcal{B}^{\Gamma_i}$ is set to the bulk value: $\mathcal{B}^{\Gamma_i}_{\alpha} = \mathcal{B}^{\Omega}_{\alpha}$.

To summarize, the following sequential steps describe the complete procedure to compute all required geometric moments:

1. Compute initial surface moments $\mathcal{A}_0^{\Gamma_i}$ for each interface i.
2. Compute bulk geometric moments $\{\mathcal{A}^\Omega, \mathcal{B}^\Omega, \mathcal{V}, \mathcal{W}\}$ using the intersection of the wet areas that correspond to each domain.
3. Update surface moments $\mathcal{A}^{\Gamma_i}$ using the initial surface moments and the intersection point.
4. Compute surface moments $\mathcal{B}^{\Gamma_i}$ according to the previous rules.

As done in Part 1 [24], the previously defined vectors containing the geometric moments are used to define diagonal matrices, that will later be employed to define the discrete cut-cell operators. This results in the volume operators,

$$V = \operatorname{diag}(\mathcal{V}) \quad \text{and} \quad W = \begin{bmatrix} \operatorname{diag}(\mathcal{W}_x) & \\ & \operatorname{diag}(\mathcal{W}_y) \end{bmatrix}, \tag{18}$$

surface bulk ($\alpha \in \{x, y\}$)

$$A_\alpha^\Omega = \operatorname{diag}\left(\mathcal{A}_\alpha^\Omega\right) \quad \text{and} \quad B_\alpha^\Omega = \operatorname{diag}\left(\mathcal{B}_\alpha^\Omega\right) \tag{19}$$

and interfacial operators ($\alpha \in \{x, y\}$ and $i \in \{1, 2\}$),

$$A_\alpha^{\Gamma_i} = \operatorname{diag}\left(\mathcal{A}_\alpha^{\Gamma_i}\right) \quad \text{and} \quad B_\alpha^{\Gamma_i} = \operatorname{diag}\left(\mathcal{B}_\alpha^{\Gamma_i}\right). \tag{20}$$

3.1.2 Levelset boundary conditions

To impose a given angle at contact point defined as the intersection of the two interfaces Γ_1 and Γ_2, the angle introduced in the right-hand side of Eq. (7) cannot be directly the contact angle if the solid wall is not aligned with the cell faces. In this case, it is replaced by β, which is the angle between the free-surface Γ_1 and a certain reference direction defined by two cell centers, which depend on the orientation of the solid wall Γ_2. More details will be given in the following. Therefore, the boundary condition now reads

$$\frac{\partial \phi}{\partial n} = \cos\beta, \tag{21}$$

where $\beta = \theta + m(\alpha - \gamma)$. Eq. (21) is discretized as follows

$$\phi_{ij} - \phi_{kl} = h\cos(\theta + m(\alpha - \gamma)), \tag{22}$$

where (i, j) are the indices of the cell where the boundary condition is being applied. This boundary condition is applied to solid cells. By solid cells we refer to cells that only belong to the solid phase and do not have any volume belonging to the void or liquid phases. In the rest of the domain, Eq. (2) is applied.

The additional parameters appearing in Eq. (22) are defined as follows: (k, l) denotes the indices of a reference cell with respect to which the boundary condition is applied. θ is the angle that is to be imposed between both levelsets iso-levels, α denotes the angle of the vector normal to the interface Γ_2 at the cut-cell closest to cell (i, j) in the direction defined by the cell (k, l), and γ is the angle between the x-axis

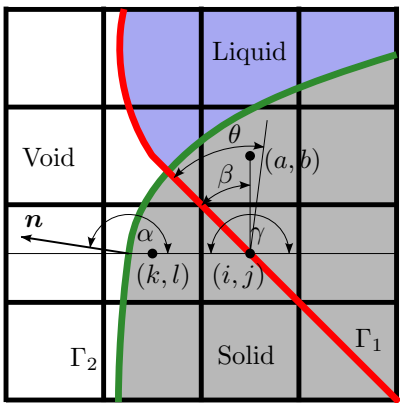


Fig. 5: Illustration of the various parameters defined in the discrete levelset boundary condition (Eq. (22) at a given cell (i,j) in the solid phase).

and the line connecting the centers of the cells (i,j) and (k,l). h denotes the distance between the centers of the cells (i,j) and (k,l) and

$$m = S\left(\phi_{ab} - \phi_{ij}\right), \tag{23}$$

where (a,b) are the indices of the cell in the perpendicular clockwise direction to the line connecting the cells (i,j) and (k,l) and S denotes the sign function. The perpendicular anticlockwise direction can also be used, but in that case the right-hand-side of Eq. (23) has to be multiplied by -1.

The angle β is therefore measured from the line connecting the cell (i,j) with the cell (a,b) and the angle θ is measured from the tangent to the interface Γ_2 at the cell (k,l). Fig. 5 represents an example showing the different angles and cells that are employed to compute the angle β. It can be clearly seen that the difference between the angles γ and α is the same as the difference between the angles θ and β.

3.2 Discrete cut-cell operators and boundary conditions

In this section, the discrete cut-cell gradient and divergence operators used to discretize the Navier-Stokes equations are defined for the case with two interfaces. The discretization of the boundary conditions is then presented. The reader is referred to Part 1 [24] for the definition of the differentiation operators $D_\alpha^\pm$ and the interpolation operators $S_\alpha^\pm$, and to Sec. 2.2 for a description of the pressure-projection algorithm.

3.2.1 Gradient of a cell-centered scalar field

The discrete gradient has three different terms, one that accounts for the contribution of the bulk field and two with the contribution of each interface:

$$\mathrm{grad}\left(p^\omega,\, p^{\gamma_1},\, p^{\gamma_2}\right) = W^\dagger\left(Gp^\omega + H^{\Gamma_1}p^{\gamma_1} + H^{\Gamma_2}p^{\gamma_2}\right), \tag{24}$$

where † denotes the Moore-Penrose pseudoinverse and the linear operators are defined as

$$G = \begin{bmatrix} D_x^- B_x^\Omega \\ D_y^- B_y^\Omega \end{bmatrix} \quad \text{and} \quad H^{\Gamma_i} = \begin{bmatrix} A_x^{\Gamma_i} D_x^- - D_x^- B_x^{\Gamma_i} \\ A_y^{\Gamma_i} D_y^- - D_y^- B_y^{\Gamma_i} \end{bmatrix}. \tag{25}$$

3.2.2 Divergence of a face-centered vector field

Just like the gradient operator, the (volume-integrated) divergence operator is multi-linear and takes one more argument than the number of interfaces:

$$\operatorname{div}(\boldsymbol{q}^\omega, \boldsymbol{q}^{\gamma_1}, \boldsymbol{q}^{\gamma_2}) = -\left(G^\top + H^{\Gamma_1,\top} + H^{\Gamma_2,\top}\right)\boldsymbol{q}^\omega + H^{\Gamma_1,\top}\boldsymbol{q}^{\gamma_1} + H^{\Gamma_2,\top}\boldsymbol{q}^{\gamma_2} \tag{26}$$

where

$$-G^\top - H^{\Gamma_1,\top} - H^{\Gamma_2,\top} = \begin{bmatrix} D_x^+ A_x^\Omega & D_y^+ A_y^\Omega \end{bmatrix} \tag{27}$$

and

$$H^{\Gamma_i,\top} = \begin{bmatrix} B_x^{\Gamma_i} D_x^+ - D_x^+ A_x^{\Gamma_i} & B_y^{\Gamma_i} D_y^+ - D_y^+ A_y^{\Gamma_i} \end{bmatrix}. \tag{28}$$

As proposed in Part 1, whenever the discrete vector fields $\boldsymbol{q}^\omega$, $\boldsymbol{q}^{\gamma_1}$ or $\boldsymbol{q}^{\gamma_2}$ represent the gradient of a scalar field, the closures $\boldsymbol{q}^\omega = \boldsymbol{q}^{\gamma_1}$ and/or $\boldsymbol{q}^\omega = \boldsymbol{q}^{\gamma_2}$ are used (wherever required). The divergence operator then degenerates to the single argument form,

$$\operatorname{div}(\boldsymbol{q}^\omega) = -G^\top \boldsymbol{q}^\omega. \tag{29}$$

3.2.3 Robin boundary condition

The boundary conditions also have to include the contributions from other interfaces. For a general Robin boundary condition applied on the boundary Γ_i of the form:

$$ap + b\partial_n p = g \quad \text{on} \quad \Gamma_i, \tag{30}$$

the discretized version reads

$$I_a^{\Gamma_i} I^{\Gamma_i} p^{\gamma_i} + I_b^{\Gamma_i} H^{\Gamma_i,\top} W^\dagger \left[Gp^\omega + H^{\Gamma_i} p^{\gamma_i} + H^{\Gamma_j} p^{\gamma_j}\right] = I^{\Gamma_i} g^{\gamma_i}, \quad \forall j \neq i, \tag{31}$$

where the diagonal matrices $I_a^{\Gamma_i} = \operatorname{diag}(a^{\gamma_i})$ and $I_b^{\Gamma_i} = \operatorname{diag}(b^{\gamma_i})$ are coefficients that can be used to impose either Dirichlet, Neumann or Robin boundary conditions, the diagonal matrix $I^{\Gamma_i} = \operatorname{diag}\left(\left|H^{\Gamma_i,\top}\mathbf{1}\right|\right)$ measures the length boundary within a given cell.

It should be noted that the last term on the left-hand-side of Eq. (31) represents the contribution from the second boundary Γ_j to the gradient, itself projected onto the direction normal to Γ_i.

3.2.4 Free-surface boundary condition

Let Γ_1 be the interface on which the free-surface boundary condition is applied. Eq. (16) is then discretized as follows,

$$-H^{\Gamma_1} p^{\gamma_1} + 2\mu \begin{bmatrix} H_x^{\Gamma_1,\top} S_x \\ H_y^{\Gamma_1,\top} S_y \end{bmatrix} = -H^{\Gamma_1} \sigma \kappa^{\gamma_1}, \tag{32}$$

where S_x is defined following

$$S_x = \begin{bmatrix} S_{xx} \\ S_{xy} \end{bmatrix}, \quad \begin{aligned} S_{xk} =& \frac{1}{2} W_{xk}^{\dagger} \left(G_{xk} u_x^{\omega} + H_{xk}^{\Gamma_1} u_x^{\gamma_1} + H_{xk}^{\Gamma_2} u_x^{\gamma_2} \right) + \\ & \frac{1}{2} W_{kx}^{\dagger} \left(G_{kx} u_k^{\omega} + H_{kx}^{\Gamma_1} u_k^{\gamma_1} + H_{kx}^{\Gamma_2} u_k^{\gamma_2} \right), \quad \forall k \in \{x, y\} \end{aligned} \tag{33}$$

and S_y following

$$S_y = \begin{bmatrix} S_{yx} \\ S_{yy} \end{bmatrix}, \quad \begin{aligned} S_{yk} =& \frac{1}{2} W_{yk}^{\dagger} \left(G_{yk} u_y^{\omega} + H_{yk}^{\Gamma_1} u_y^{\gamma_1} + H_{yk}^{\Gamma_2} u_y^{\gamma_2} \right) + \\ & \frac{1}{2} W_{ky}^{\dagger} \left(G_{ky} u_k^{\omega} + H_{ky}^{\Gamma_1} u_k^{\gamma_1} + H_{ky}^{\Gamma_2} u_k^{\gamma_2} \right), \quad \forall k \in \{x, y\}. \end{aligned} \tag{34}$$

In the above equations, Γ_2 is the interface, where either a no-slip or Navier slip boundary condition is imposed, as discussed in the next section. The first subscript in the cut-cell operators G, W and H highlights the fact that they operate on the staggered grids, whereas the second subscript refers to which component of the operator is being used.

3.2.5 Navier slip boundary conditions

If the Navier slip boundary condition is to be applied to a straight surface, Eq. (31) can be used to discretize Eq. (17). However, if the boundary condition is applied to an arbitrarily oriented boundary surface, some modifications are required which are presented here.

For an arbitrary orientation, the tangential component of the velocity u_t depends on both velocity components u_x and u_y. As presented in Part 1, the proposed methodology (and operators previously defined) leverages on bulk and boundary values, the later being defined implicitly as roots of the discretized boundary conditions. This results in a system of algebraic equation that consists of the discretized bulk (momentum) and boundary conditions, that is solved to determine both bulk and boundary fields.

In the methodology presented in Part 1 [24], both boundary velocity components (u_x^{γ} and u_y^{γ}) were solved. However, for the Navier slip condition, a different approach is used: in the absence of blowing or suction, the velocity component normal to the boundary is identically zero. As far as the velocity component tangential to the boundary is concerned, it is implicitly defined by a discretized form of Eq. (17) (see below). For simplicity, the boundary field u_t^{γ} is collocated with the cells where p is defined. However, in order to assemble the viscous and convective transport terms and the velocity boundary conditions, this information (together with the vanishing normal velocity) must be interpolated to the mesh faces before being included in the viscous transport term.

To illustrate this, let us consider the case where the Navier slip condition (Eq. (17)) is applied on Γ_2, and a different boundary condition (*e.g.* a free-surface or Dirichlet

boundary condition) on Γ_1. The discrete version of Eq. (17) then reads

$$
\begin{aligned}
I^{\Gamma_2} u_t^{\gamma_2} - \lambda \big[& H^{\Gamma_2,\top} W^\dagger H^{\Gamma_2} u_t^{\gamma_2} + \\
& H^{\Gamma_2,\top} W^\dagger H^{\Gamma_1} \left(\operatorname{diag}(\sin\alpha)\, S_x^- u_x^{\gamma_1} + \operatorname{diag}(-\cos\alpha)\, S_y^- u_y^{\gamma_1} \right) + \\
& H^{\Gamma_2,\top} W^\dagger G \left(\operatorname{diag}(\sin\alpha)\, S_x^- u_x^{\omega} + \operatorname{diag}(-\cos\alpha)\, S_y^- u_y^{\omega} \right) \big] = I^{\Gamma_2} U_b. \quad (35)
\end{aligned}
$$

The projection and interpolation of the tangential velocity onto the mesh faces can be performed following

$$
\begin{aligned}
u_x^{\gamma_i} &= \operatorname{diag}(\sin\alpha)\, S_x^+ u_t^{\gamma_i}, \\
u_y^{\gamma_i} &= \operatorname{diag}(-\cos\alpha)\, S_y^+ u_t^{\gamma_i},
\end{aligned} \quad (36)
$$

which is used in the viscous and convective transport terms and in boundary conditions where the Cartesian components of the velocity appear, as mentioned earlier.

Regarding the velocity divergence which appears in the right-hand-side of the Poisson equation for the pressure, there is no contribution from the Navier slip boundary since there is no velocity in the normal direction.

4 Validation and applications

In this section, various aspects of our implementation are validated and analyzed, starting with the implementation of the free-surface boundary condition by analyzing the evolution of a dripping liquid film. Once validated, the Navier-slip and levelset boundary conditions are added and validated using the case of a sessile drop on a straight surface. Finally, the case of a drop over a circular solid is considered, a more challenging configuration that fully leverages the proposed methodology to mix boundary conditions on arbitrary surfaces.

4.1 Dripping of a liquid film

The dripping instability of a liquid film underneath a flat plate subject to the gravity force has been extensively analyzed in the literature [37–41]. In this work, this case is used to validate the implementation of the free-surface boundary condition.

The initial shape of the drop is given by the following expression:

$$
y = -h_0 \left(1 + A \cos\left(\frac{2\pi x}{L} \right) \right), \quad (37)
$$

where, h_0 is the average film thickness, A is the initial amplitude of the drop, and L denotes the length of the domain in the horizontal, x, direction. The non-dimensional parameters that govern the flow are the Reynolds and Bond numbers, defined as

$$
\mathrm{Re} = \frac{h_0^3 g}{3\nu^2}, \quad \mathrm{Bo} = \frac{\rho h_0 g}{\sigma}, \quad (38)
$$

where, g is the gravity force, ν is the kinematic viscosity and ρ the density. These non-dimensional numbers can in turn be used to define the Kapitza and Ohnesorge numbers

$$\mathrm{Ka} = (3\,\mathrm{Re})^{2/3} \,/\, \mathrm{Bo}, \quad \mathrm{Oh} = \mathrm{Ka}^{-3/4}\,\mathrm{Bo}^{-1/4}\,. \tag{39}$$

Following this formulation, the capillary length $l_c = \sqrt{\sigma/\rho g}$ can then be related to the film thickness, using the Bond number as, $h_0/l_c = \sqrt{\mathrm{Bo}}$.

Following [41], a simulation is performed in a domain of length $L = 10l_c$ in both directions. The Ohnesorge number is $\mathrm{Oh} = 0.05$. The density and the surface tension coefficient are set to $\rho = \sigma = 1$, whereas the gravity is set to $g = 10$. Finally, the kinematic viscosity is $\nu = 0.056$. Under these conditions, the capillary length is $l_c = 1/\sqrt{10}$, yielding a computational domain of size $(x, y) = \left[-\sqrt{10}/2, \sqrt{10}/2\right] \times \left[-\sqrt{10}, 0\right]$, which is discretized using 128 grid points in each directions. The levelset is reinitialized every three time-steps, performing two iterations of Eq. (5) every time. Fig. 6 compares qualitatively the solution obtained using the cut-cell method with the numerical results obtained by [41], showing a good agreement between both.

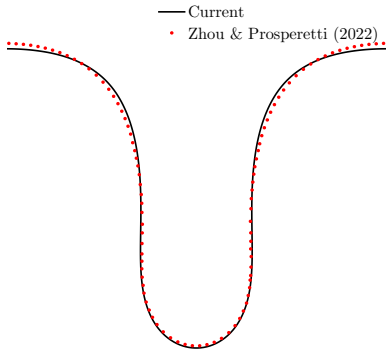


Fig. 6: Shape of the drop when the inflection points appear compared the one in [41].

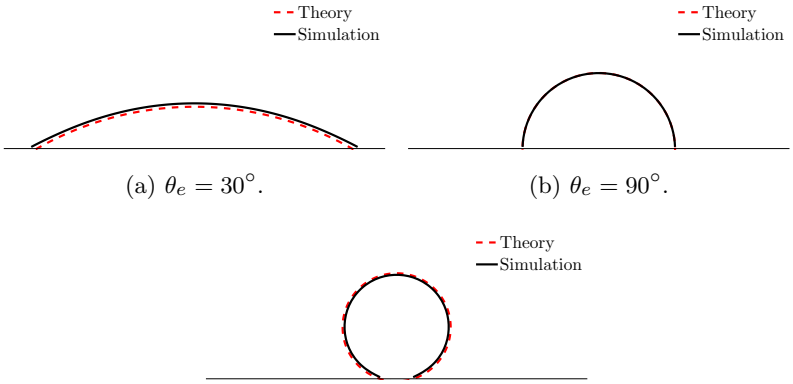
4.2 Sessile drop on a straight surface

This case, allows the validation of our implementation of the free-surface, the Navier slip and the levelset boundary conditions. For this purpose, a circular drop is placed on a straight solid wall with an initial contact angle $\theta_e = 90°$, in the absence of a buoyancy force. The simulations are performed for contact angles between 30° and 165°. Under these conditions, the final shape of the drop is a circle and its radius can be computed as

$$R_f(V_0, \theta_e) = \sqrt{\frac{V_0}{\theta_e - \sin\theta_e \cos\theta_e}}, \tag{40}$$

where, $V_0 = \pi R_0^2/2$ is the initial volume of the drop, and the initial radius is set to $R_0 = 0.5$. A Navier slip boundary condition is imposed at the wall with a slip length

of $\lambda = 0.01$. The size of the computational domain is $(x, y) = [-1.5, 1.5] \times [-1, 1]$ and each unit length is discretized using 32 points in each direction. The Reynolds number is set to $\mathrm{Re} = 1$. The simulations are carried out until the shape of the drop reaches a steady state. A reinitialization of the levelset field is performed every 10 time-steps using 2 pseudo-temporal iterations of the reinitialization equation (5), only if the deformation of the levelset exceeds a threshold $\delta = 0.01$. The stop condition of the simulations is based on the difference of the droplet radius between two consecutive iterations. If the difference is smaller than a certain percentage of the current radius, then the simulation is stopped.



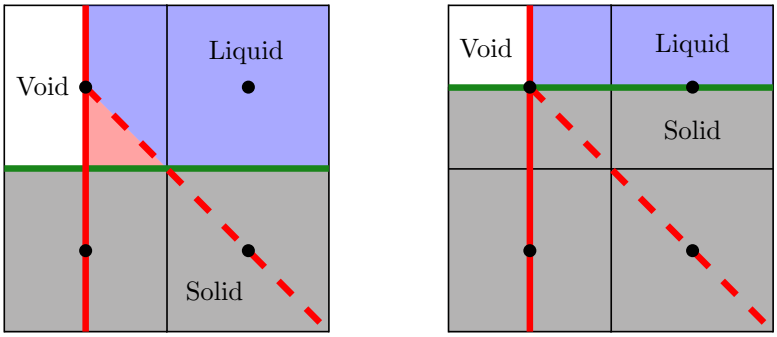
(a) $\theta_e = 30^\circ$. (b) $\theta_e = 90^\circ$.

(c) $\theta_e = 165^\circ$.

Fig. 7: Final drop shapes compared to theoretical solution.

Fig. 7 shows the cases with the flattest and steepest contact angles, along with the equilibrium case. Fig. 7a shows that a small volume is gained for the flattest angle, $\theta_e = 30^\circ$. In contrast, as shown in Fig. 7c, a small volume is lost during the simulation until reaching steady state. There are several sources of mass gain/loss that can be attributed to the use of the levelset method. The first is due to the fact that the levelset does not represent a conserved quantity and therefore its advection results in volumes that are not conserved. Since a piecewise linear representation of the interface is used, some mass is lost at each cell. This quantity is bounded but changes as the levelset is advected. In addition to the advection, the reinitialization procedure can also generate spurious mass loss or gain and interface deformation [42]. Being aware of such restriction, special attention has been given to the reinitialization method used in this work, specially chosen to minimize such mass loss/gain. Different methods were tested [32, 43, 44] and as stated in Sec. 2.1, the HCR2 of Hartmann et al. [32] was selected. The mass loss/gain due to reinitialization can also be controlled by reducing the frequency of reinitialization. The smaller the frequency, the smaller the mass loss/gain. The final source is due to the applications of the boundary condition on the levelset in the solid phase. As shown in Fig. 8, where the red solid line is the initial position of the free-surface and the red dashed line is its final position,

applying the boundary conditions on the levelset results in a change of the liquid volume (represented by the red shaded area), which is maximum if the solid wall is placed at the cell faces (Fig. 8a) and zero if the solid wall is placed at the cell centers (Fig. 8b). This adds a small volume at each iteration for $\theta_e > 90°$ and removes it for $\theta_e < 90°$ before reaching equilibrium. The greater the difference between θ_e and 90°, the greater the mass loss/gain. It is expected that the loss/gain decreases linearly with the cell size for all the contact angles [13].



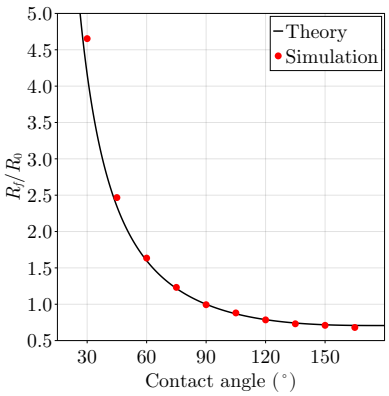
(a) Solid wall at the cell face.

(b) Solid wall at the cell center.

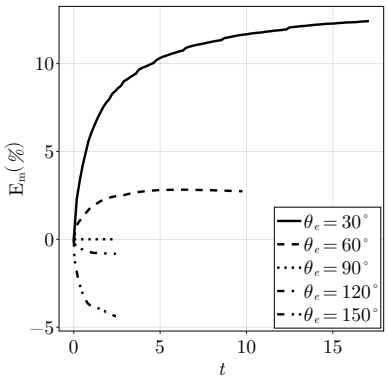
Fig. 8: Example of mass loss while imposing $\theta_e = 135°$ to the levelset on an interface with a previous contact angle of $\theta_e = 90°$. The solid and dashed red lines represent the free-surface interface before and after applying the boundary conditions, respectively. The green solid line represents the solid wall. The blue shaded area is the liquid phase and the red shaded area is the lost volume due to the application of the boundary condition on the solid cells.

In order to more systematically analyze the impact of mass gain/loss on the results, a more detailed analysis of the results is presented in Fig. 9. Fig. 9a compares the theoretical ratio of final and initial radius with the one obtained numerically, showing a good agreement between both for all the cases. The solid wall has been placed as in Fig. 8a, which results in a maximal mass loss/gain. In order to quantify it over the range of simulations, the evolution of mass absorption is plotted in Fig. 9b for different contact angles. This figure shows that for angles $\theta_e < 90°$, the numerical radius is larger than the theoretical one, whereas the opposite happens when $\theta_e > 90°$. This result suggests that the main source of mass loss/gain is due to the imposition of the boundary conditions on the levelset. As stated before, this can be reduced to zero if the solid wall had been placed at the cell centers. However, in a more general case with an arbitrary orientation of the solid wall, this cannot be controlled and therefore we present here the most adverse case. This figure also shows that different contact angles need different times to converge to the equilibrium position, which is linked to the different distances that the interface has to travel to reach the final position. The longer the distance, the larger the time it takes the droplet to reach its equilibrium position. This also explains why the error is not symmetric with respect

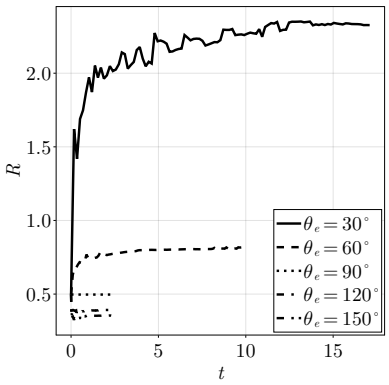
to the $\theta_e = 90°$ case. The distances travelled by the cases where $\theta_e > 90°$ are smaller than their symmetric counterparts, yielding smaller mass loss in the former than mass gain in the latter. In Fig. 9c, the evolution of the radius of the drop is presented. It can be seen that all the cases start with $R_0 = 0.5$ and converge towards the respective equilibrium positions. Finally, Fig. 9d presents the expected first-order convergence rate of the mass absorption for the case with a contact angle of $\theta_e = 165°$, where the number of cells per length unit have been set to 16, 32 and 64.



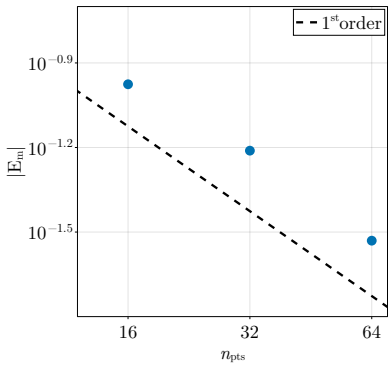
(a) Ratio of initial and final radius as function of the contact angle.



(b) Mass absorption for $\theta_e = \{30°, 60°, 90°, 120°, 150°\}$.



(c) Evolution of the radius of the drop for $\theta_e = \{30°, 60°, 90°, 120°, 150°\}$.



(d) Mass absorption with respect to number of grid points per unit length for $\theta_e = 165°$.

Fig. 9: Results for sessile drop on a straight surface.

4.3 Drop over a circular solid

The spreading of a droplet over a cylindrical solid surface is investigated for contact angles varying between 30° and 135°. This case has recently been used for validation by [13, 15–17], where a volume-of-fluid method was used to track the interface. This case provides an implementation on non-straight surfaces, but still allows access to the analytical solution, which can be used to validate the numerical models. In the absence of gravity, the analytical solution is simply given by a circular shape, as in the previous case. Fig. 10 shows a schematic representation of the underlying configuration.

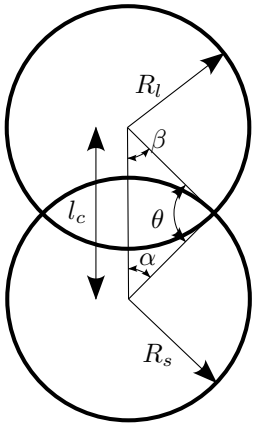


Fig. 10: Representation of the drop over the cylinder for a contact angle of θ.

The radius of the solid cylinder and the initial radius of the drop are $R_s = R_{l,0} = 0.5$. Initially, the contact angle is set to 90°. The size of the computational domain is $(x, y) = [-1, 1] \times [-1.125, 1.125]$ and again each unit length is discretized using 32 points in each direction. The Reynolds number is again set to $\mathrm{Re} = 1$. The reinitialization of the levelset is performed every 3 temporal iterations, with 2 pseudo-temporal iterations of Eq. (5) performed every time. The same stop condition as in Sec. 4.2 is employed. As in the previous case, 32 grid points are used per length unit in each direction. In this configuration, the initial distance between both centers is given by

$$l_{c,0} = 2R_s \cos 45°. \tag{41}$$

The area of the liquid drop can be calculated analytically by subtracting from the total area of the liquid cylinder the area overlapped by the solid cylinder, which is given by the circular segments of angles 2α and 2β (see Fig.10), as

$$A\left(R_s, R_l, l_c\right) = \pi R_l^2 - R_s^2 \left[\alpha - \frac{1}{2}\sin\left(2\alpha\right)\right] - R_l^2 \left[\beta - \frac{1}{2}\sin\left(2\beta\right)\right], \tag{42}$$

where α and β can be computed using the cosine rule

$$\begin{aligned} \alpha\left(R_s, R_l, l_c\right) &= \arccos\left(\frac{l_c^2 + R_s^2 - R_l^2}{2 l_c R_s}\right), \\ \beta\left(R_s, R_l, l_c\right) &= \arccos\left(\frac{l_c^2 + R_l^2 - R_s^2}{2 l_c R_l}\right). \end{aligned} \tag{43}$$

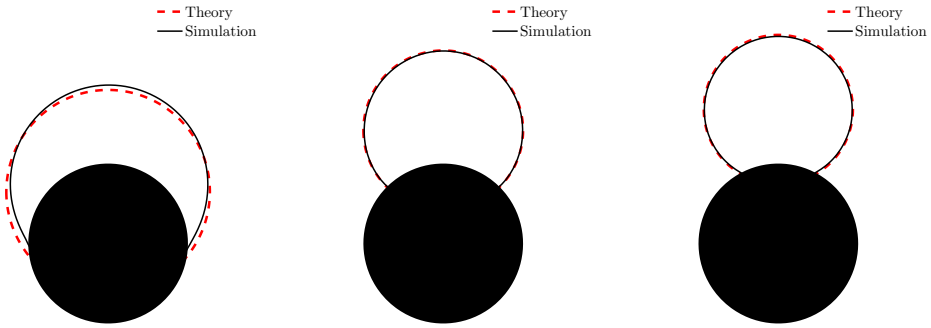
The final radius of the drop can be computed by matching the areas of the initial and final liquid drop, or equivalently by finding the roots of the following function

$$f\left(l_{c,f}\right) = A\left(R_s, R_{l,0}, l_{c,0}\right) - A\left(R_s, R_{l,f}, l_{c,f}\right), \tag{44}$$

The final radius of the liquid drop is therefore given by the following expression

$$R_{l,f} = R_s \cos\theta_e + \sqrt{R_s^2 \cos{\theta_e}^2 - R_s + l_{c,f}^2}, \tag{45}$$

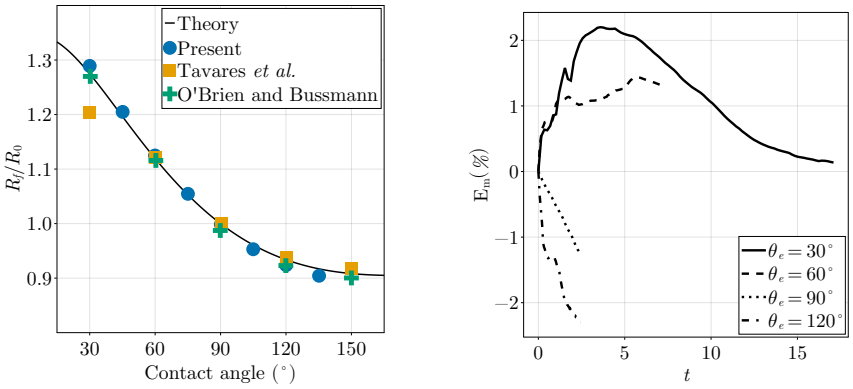
which is obtained by applying the cosine rule to the contact angle.



(a) $\theta_e = 30°$. (b) $\theta_e = 90°$. (c) $\theta_e = 120°$.

Fig. 11: Final drop shapes compared to the theoretical solution.

Fig. 11 shows the final drops for the cases $\theta_e = \{30°, 90°, 120°\}$ and compares them with the theoretical solution. The comparison shows good qualitative agreement. A quantitative comparison of the final radius with the theoretical one is shown in Fig. 12a for different angles. The results from Tavares et al. [13] and O'Brien and Bussmann [16] are also shown. It can be seen that overall, all the methods behave well for intermediate contact angles and as we move away from the 90° case the results tend to be slightly worse. As in the cases presented in Sec. 4.2, for contact angles $\theta_e < 90°$, an increase in the mass of the drop is obtained due to the imposition of the boundary conditions on the levelset, resulting in a final radius slightly larger than the theoretical one. The opposite is observed for contact angles $\theta_e > 90°$. This mass loss/gain explains the slight



(a) Ratio of initial and final radius as function of the contact angle.

(b) Mass absorption for $\theta_e = \{30°, 60°, 90°, 120°\}$.

Fig. 12: Results for sessile drop on a cylindrical surface.

deviation from the theoretical results at both ends of the angle range considered. This is confirmed by Fig. 12b, which shows the mass absorption for different contact angles. We observe that the effect of imposing the levelset boundary conditions is weaker if compared to the previous case, since the mass loss/gain is smaller. The reason is that in this case the interface travels a shorter distance for the same imposed contact angle before reaching the equilibrium. Moreover, all the cases tend to lose mass towards the end of the simulation. This trend can be explained by the frequent reinitialization of the levelset. The frequency of the reinitialization had to be increased with respect to the case presented in Sec. 4.2 for stability reasons.

5 Conclusions and future work

The cut-cell framework presented in the first segment of this two-part contribution was successfully extended to simulate contact line dynamics in two-dimensional free-surface flows. In particular, no assumptions were made on either the interface orientation or the magnitude of the terms at play. The simplicity of the original method is retained and constitutes an important step towards its extension to three-dimensional flows, which is currently under progress. The extension of the "two interfaces" logic is however more challenging than the simpler case of a single boundary/interface exemplified in Part 1.

The results on the spreading droplet are on par with those obtained by approaches recently developed in other groups using the one-fluid formulation instead of the cut-cell method to impose the free-surface boundary conditions. The mass conservation error due to the imposition of the levelset boundary conditions is first-order, as expected, reaching a maximum of 10% mass loss/gain for the worst-case scenario. When applied to non-straight surfaces, further reinitialization of the levelset (when

compared to the straight surface case) is required for robustness. Nevertheless, problems associated with the use of the levelset can be circumvented by substituting it by other methods such us Volume-of-Fluid or front-tracking, since the proposed cut-cell methodology can be coupled with any interface representation method.

Furthermore, our numerical implementation of the Navier boundary condition allows for the incorporation of more advanced models. For instance, dynamic contact angles can be modeled using the Cox-Voinov relation [45, 46], where the contact angle is dependent on the local capillary number. Additionally, the Generalized Navier Boundary Condition [47, 48] could be employed, in which a mobility relation links the contact line speed to local flow field properties via the uncompensated Young's stress.

References

[1] Blake, T.D., Shikhmurzaev, Y.D.: Dynamic Wetting by Liquids of Different Viscosity. Journal of Colloid and Interface Science **253**(1), 196–202 (2002) https://doi.org/10.1006/jcis.2002.8513

[2] Hocking, L.M.: Rival contact-angle models and the spreading of drops. Journal of Fluid Mechanics **239**, 671–681 (1992) https://doi.org/10.1017/S0022112092004579

[3] Eggers, J., Evans, R.: Comment on "dynamic wetting by liquids of different viscosity," by t.d. blake and y.d. shikhmurzaev. Journal of Colloid and Interface Science **280**(2), 537–538 (2004) https://doi.org/10.1016/j.jcis.2004.07.001

[4] Shikhmurzaev, Y.D.: Singularities at the moving contact line. mathematical, physical and computational aspects. Physica D: Nonlinear Phenomena **217**(2), 121–133 (2006) https://doi.org/10.1016/j.physd.2006.03.003

[5] Huh, C., Scriven, L.E.: Hydrodynamic model of steady movement of a solid/liquid/fluid contact line. Journal of Colloid and Interface Science **35**(1), 85–101 (1971) https://doi.org/10.1016/0021-9797(71)90188-3

[6] Lācis, U., Johansson, P., Fullana, T., Hess, B., Amberg, G., Bagheri, S., Zaleski, S.: Steady moving contact line of water over a no-slip substrate: Challenges in benchmarking phase-field and volume-of-fluid methods against molecular dynamics simulations. The European Physical Journal Special Topics **229**(10), 1897–1921 (2020) https://doi.org/10.1140/epjst/e2020-900280-9

[7] Fullana, Tomas, Zaleski, Stéphane, Popinet, Stéphane: Dynamic wetting failure in curtain coating by the Volume-of-Fluid method - Volume-of-Fluid simulations on quadtree meshes. Eur. Phys. J. Special Topics **229**(10), 1923–1934 (2020) https://doi.org/10.1140/epjst/e2020-000004-0

[8] Lacis, U., Johansson, P., Fullana, T., Hess, B., Amberg, G., Bagheri, S., Zaleski, S.: Steady moving contact line of water over a no-slip substrate challenges in benchmarking phase-field and volume-of-fluid methods against molecular

dynamics simulations. EUROPEAN PHYSICAL JOURNAL-SPECIAL TOPICS **229**(10, SI), 1897–1921 (2020) https://doi.org/10.1140/epjst/e2020-900280-9

[9] Gründing, D., Smuda, M., Antritter, T., Fricke, M., Rettenmaier, D., Kummer, F., Stephan, P., Marschall, H., Bothe, D.: A comparative study of transient capillary rise using direct numerical simulations. Applied Mathematical Modelling **86**, 142–165 (2020) https://doi.org/10.1016/j.apm.2020.04.020 . Accessed 2024-08-19

[10] Afkhami, S., Zaleski, S., Bussmann, M.: A mesh-dependent model for applying dynamic contact angles to VOF simulations. Journal of Computational Physics **228**(15), 5370–5389 (2009) https://doi.org/10.1016/j.jcp.2009.04.027

[11] Legendre, D., Maglio, M.: Comparison between numerical models for the simulation of moving contact lines. Computers & Fluids **113**, 2–13 (2015) https://doi.org/10.1016/j.compfluid.2014.09.018 . Accessed 2022-01-14

[12] Spelt, P.D.M.: A level-set approach for simulations of flows with multiple moving contact lines with hysteresis. Journal of Computational Physics **207**(2), 389–404 (2005) https://doi.org/10.1016/j.jcp.2005.01.016 . Accessed 2024-02-09

[13] Tavares, M., Josserand, C., Limare, A., Lopez-Herrera, J.M., Popinet, S.: A coupled VOF/embedded boundary method to model two-phase flows on arbitrary solid surfaces. Computers & Fluids **278**, 106317 (2024) https://doi.org/10.1016/j.compfluid.2024.106317

[14] Kataoka, I.: Local instant formulation of two-phase flow. International Journal of Multiphase Flow **12**(5), 745–758 (1986) https://doi.org/10.1016/0301-9322(86)90049-2

[15] Patel, H.V., Das, S., Kuipers, J.A.M., Padding, J.T., Peters, E.A.J.F.: A coupled Volume of Fluid and Immersed Boundary Method for simulating 3D multiphase flows with contact line dynamics in complex geometries. Chemical Engineering Science **166**, 28–41 (2017) https://doi.org/10.1016/j.ces.2017.03.012

[16] O'Brien, A., Bussmann, M.: A volume-of-fluid ghost-cell immersed boundary method for multiphase flows with contact line dynamics. Computers & Fluids **165**, 43–53 (2018) https://doi.org/10.1016/j.compfluid.2018.01.006

[17] Asghar, M.H., Fricke, M., Bothe, D., Maric, T.: Numerical wetting benchmarks – advancing the plicRDF-isoAdvector unstructured Volume-of-Fluid (VOF) method. arXiv. arXiv:2302.02629 [physics] (2023). http://arxiv.org/abs/2302.02629

[18] Bozonnet, C., Desjardins, O., Balarac, G.: Traction open boundary condition for incompressible, turbulent, single- or multi-phase flows, and surface wave simulations. Journal of Computational Physics **443**, 110528 (2021) https://doi.org/10.1016/j.jcp.2021.110528

[19] Harlow, F.H., Welch, J.E.: Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. Physics of Fluids **8**(12), 2182 (1965) https://doi.org/10.1063/1.1761178

[20] Veldman, A.E.P., Gerrits, J., Luppes, R., Helder, J.A., Vreeburg, J.P.B.: The numerical simulation of liquid sloshing on board spacecraft. Journal of Computational Physics **224**(1), 82–99 (2007) https://doi.org/10.1016/j.jcp.2006.12.020

[21] Donea, J., Huerta, A., Rodrıguez-Ferran, A.: Chapter 14 Arbitrary Lagrangian–Eulerian Methods. In: The Encyclopedia of Computational Mechanics vol. 1, pp. 413–437. John Wiley & Sons, Ltd, ??? (2004). https://doi.org/10.1002/0470091355.ecm009

[22] Peskin, C.S.: Flow Patterns Around Heart Valves: A Numerical Method. Journal of Computational Physics **10**, 252–271 (1972) https://doi.org/10.1016/0021-9991(72)90065-4

[23] Mittal, R., Iaccarino, G.: Immersed boundary methods. Annual Review of Fluid Mechanics **37**, 239–261 (2005) https://doi.org/10.1146/annurev.fluid.37.061903.175743

[24] Quirós Rodríguez, A., Fullana, T., Le Chenadec, V., Sayadi, T.: A level-set based cut cell method for interfacial flows. Part 1: Navier-Stokes equations and Rayleigh-Bénard instability with melting boundary. Acta Mechanica (2024)

[25] Sethian, J.A.: Level Set Methods and Fast Marching. Cambridge Monographs on Applied and Computational Mathematics, vol. 1. Cambridge University Press, University of California, Berkeley (1999)

[26] Tryggvason, G., Scardovelli, R., Zaleski, S.: Direct Numerical Simulations of Gas-liquid Multiphase Flows. Cambridge University Press, Cambridge ; New York (2011). https://doi.org/10.1017/CBO9780511975264

[27] Mikula, K., Ohlberger, M., Urbán, J.: Inflow-implicit/outflow-explicit finite volume methods for solving advection equations. Applied Numerical Mathematics **85**, 16–37 (2014) https://doi.org/10.1016/j.apnum.2014.06.002

[28] Peng, D., Merriman, B., Osher, S., Zhao, H., Kang, M.: A PDE-Based Fast Local Level Set Method. Journal of Computational Physics **155**(2), 410–438 (1999) https://doi.org/10.1006/jcph.1999.6345

[29] Fullana, T., Le Chenadec, V., Sayadi, T.: Adjoint-based optimization of two-dimensional Stefan problems. Journal of Computational Physics **475**, 111875 (2023) https://doi.org/10.1016/j.jcp.2022.111875

[30] Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory

shock-capturing schemes. Journal of Computational Physics **77**(2), 439–471 (1988) https://doi.org/10.1016/0021-9991(88)90177-5

[31] Jiang, G.-S., Peng, D.: Weighted ENO Schemes for Hamilton-Jacobi Equations. SIAM Journal on Scientific Computing **21**(6), 2126–2143 (2000) https://doi.org/10.1137/S106482759732455X

[32] Hartmann, D., Meinke, M., Schröder, W.: The constrained reinitialization equation for level set methods. Journal of Computational Physics **229**(5), 1514–1535 (2010) https://doi.org/10.1016/j.jcp.2009.10.042

[33] Luddens, F., Bergmann, M., Weynans, L.: Enablers for high-order level set methods in fluid mechanics. International Journal for Numerical Methods in Fluids **79**(12), 654–675 (2015) https://doi.org/10.1002/fld.4070

[34] Brown, D.L., Cortez, R., Minion, M.L.: Accurate Projection Methods for the Incompressible Navier–Stokes Equations. Journal of Computational Physics **168**(2), 464–499 (2001) https://doi.org/10.1006/jcph.2001.6715

[35] Bothe, D.: Sharp-interface continuum thermodynamics of multicomponent fluid systems with interfacial mass. International Journal of Engineering Science **179**, 103731 (2022) https://doi.org/10.1016/j.ijengsci.2022.103731

[36] Contributors, G.: GEOS coordinate transformation software library. Open Source Geospatial Foundation (2021). https://libgeos.org/

[37] Yih, C.-S.: Stability of Liquid Flow down an Inclined Plane. Physics of Fluids **6**(3), 321 (1963) https://doi.org/10.1063/1.1706737

[38] Kofman, N., Rohlfs, W., Gallaire, F., Scheid, B., Ruyer-Quil, C.: Prediction of two-dimensional dripping onset of a liquid film under an inclined plane. International Journal of Multiphase Flow **104**, 286–293 (2018) https://doi.org/10.1016/j.ijmultiphaseflow.2018.02.007

[39] Lerisson, G., Ledda, P.G., Balestra, G., Gallaire, F.: Instability of a thin viscous film flowing under an inclined substrate: steady patterns. Journal of Fluid Mechanics **898**, 6 (2020) https://doi.org/10.1017/jfm.2020.396

[40] Ledda, P.G., Lerisson, G., Balestra, G., Gallaire, F.: Instability of a thin viscous film flowing under an inclined substrate: the emergence and stability of rivulets. Journal of Fluid Mechanics **926**, 2 (2021) https://doi.org/10.1017/jfm.2021.699

[41] Zhou, G., Prosperetti, A.: Dripping instability of a two-dimensional liquid film under an inclined plate. Journal of Fluid Mechanics **932**, 49 (2022) https://doi.org/10.1017/jfm.2021.1032

[42] Solomenko, Z., Spelt, P.D.M., Ó Náraigh, L., Alix, P.: Mass conservation and

reduction of parasitic interfacial waves in level-set methods for the numerical simulation of two-phase flows: A comparative study. International Journal of Multiphase Flow **95**, 235–256 (2017) https://doi.org/10.1016/j.ijmultiphaseflow.2017.06.004

[43] Min, C., Gibou, F.: A second order accurate level set method on non-graded adaptive cartesian grids. Journal of Computational Physics **225**(1), 300–321 (2007) https://doi.org/10.1016/j.jcp.2006.11.034

[44] Russo, G., Smereka, P.: A Remark on Computing Distance Functions. Journal of Computational Physics **163**(1), 51–67 (2000) https://doi.org/10.1006/jcph.2000.6553

[45] Cox, R.G.: The dynamics of the spreading of liquids on a solid surface. part 1. viscous flow. Journal of Fluid Mechanics **168**, 169–194 (1986) https://doi.org/10.1017/S0022112086000332

[46] Voinov, O.V.: Hydrodynamics of wetting. Fluid Dynamics **11**(5), 714–721 (1976) https://doi.org/10.1007/BF01012963

[47] Qian, T., Wang, X.-P., Sheng, P.: A variational approach to moving contact line hydrodynamics. Journal of Fluid Mechanics **564**, 333–360 (2006) https://doi.org/10.1017/S0022112006001935

[48] Fullana, T., Kulkarni, Y., Fricke, M., Popinet, S., Afkhami, S., Bothe, D., Zaleski, S.: A consistent treatment of dynamic contact angles in the sharp-interface framework with the generalized Navier boundary condition. Zenodo (2024). https://doi.org/10.5281/zenodo.12749951 . https://zenodo.org/records/12749951 Accessed 2024-08-21