

A volume-of-fluid ghost-cell immersed boundary method for multiphase flows with contact line dynamics

Adam O'Brien, Markus Bussmann*

Department of Mechanical and Industrial Engineering, University of Toronto, Ontario, Toronto M5S 3G8, Canada

ARTICLE INFO

Article history:

Received 16 October 2017

Revised 10 January 2018

Accepted 11 January 2018

Available online 12 January 2018

Keywords:

Immersed boundary method

Volume-of-fluid

Surface tension

Contact lines

ABSTRACT

A method is proposed for modelling contact line dynamics in volume-of-fluid frameworks using a common and robust ghost-cell immersed boundary method in 2D. The method implements a contact line model based on the CELESTE method, used in conjunction with a CICSAM scheme capable of handling multiphase flows in which the phases have large variations in their physical properties. The ability of the model to resolve contact lines on both planar and curved geometries is demonstrated. It is also shown that the model maintains good mass conservation. Dynamic cases studied include a planar water droplet impact onto a cylinder, as well as a planar droplet penetration into a bank of cylinders with varying hydrophobicity.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Many flows, both industrial and natural, involve the wetting/de-wetting of a solid body, in which one fluid is displaced by another. The two fluids often have very different physical properties and are immiscible, resulting in the formation of a nearly discontinuous interface. At the point of contact on the solid body, the interface intersects the body at an angle dependent on the surface energies of the two fluids and the solid, governed by the well-known Young's equation:

$$\theta_s = \arccos\left(\frac{\gamma_{s,l_1} - \gamma_{s,l_2}}{\gamma_{l_1,l_2}}\right) \quad (1)$$

where θ_s is the static contact angle, γ is a surface energy and the subscripts s , l_1 and l_2 denote the solid and fluid phases. Many engineering applications must carefully consider interface dynamics, a few of which include: separation of oil from solid particles [1], ink-jet printing [2], drug delivery through nano- and micro-scale capsules [3], and the cooling of microchannels [4]. In this work, we present a new algorithm for simulating contact angle dynamics on arbitrary bodies using Computational Fluid Dynamics (CFD). In particular, a method is proposed that aims to combine Volume-of-Fluid (VoF) with sharp-interface Immersed Boundary Methods (IBMs). Below a brief summary of IBMs is presented, followed by a discussion of the numerical components required to model multi-

phase flows, as well as some recent works that have attempted to reconcile the two fields.

The Immersed Boundary Method (IBM) [5,6] involves using a fixed Eulerian background mesh on which the fluid equations of motion are solved, with solid bodies represented as Lagrangian entities. Since the mesh does not conform to the shape of the body, forces imparted by the body on the fluid (and vice versa) must be accounted for through a coupling term in the momentum equations. IBMs were pioneered by Peskin for use in modelling cardiac flows [5], and have since been adopted and extended for many different applications, including fluid-structure interaction [7], flow over flexible membranes [8], formation of ice on an airfoil [9], and ship hydrodynamics [10]. The original IB method of Peskin involves introducing a spring-like continuous forcing function into the momentum equations, and thus belongs to the first category of IB methods known as Continuous Forcing (CF) methods. Due to the discrete nature of the background Eulerian mesh, CF methods require the forcing to be spread to several nearby cells through a spreading function. A second category of IB methods known as Discrete Forcing (DF) methods also exists, where the forcing is introduced into the discretized momentum equations. This method produces a sharper interface between the fluid and solid regions, since only cells with immediate proximity to the IB are forced. This second category therefore enjoys a significant advantage in the context of rigid body simulations, and is adopted here. An extensive review of various IBMs in both categories was published by Mittal and Iaccarino [6].

Body-fitted meshes have long been used for the simulation of contact lines on surfaces of various shapes [11–15]. An excellent re-

* Corresponding author.

E-mail address: bussmann@mie.utoronto.ca (M. Bussmann).

view of contact angle methods for body-fitted meshes is presented by Sui et al. [16]. The use of IBMs in lieu of body-fitted meshes is attractive primarily for the flexibility/ease in handling arbitrary geometries and the potential for modelling systems of solid bodies with arbitrary relative motion. While IBMs have been applied previously in the context of multiphase flows, usually for the study of wave-structure interaction [10,17–22], the effect of either surface tension or wetting dynamics is usually negligible and often neglected. For other classes of flow problems, such as wetting/de-wetting of small particles [1] and flow through porous media, surface tension forces may comprise the dominant momentum term. A few methods have been proposed for combining wetting dynamics and IBMs. Yang and Stern [17] combined a sharp-interface IBM with a level-set method, and developed a contact angle model for studying wave-body interactions with surface tension. They noted that the modification of the level-set function led to issues with phase conservation. This was also observed by Liu and Ding [23], who proposed a diffuse-interface method for Moving Contact Line (MCL) problems.

Two primary capabilities are required for the successful simulation of multiphase flows with surface tension: (i) advection of a colour function representing the distribution of phases, and (ii) accurate calculation of surface tension forces from the colour function. Tryggvason [24] et al. proposed the Front-Tracking (FT) method which represents the interface as a Lagrangian entity represented by a set of marker points. The Level-Set (LS) method (e.g. Sussmann et al. [25]) describes the distribution of phases in terms of a signed distance function, with zero representing the interface. This method is simple and has several advantages, including minimizing spurious interface velocities, but suffers from a lack of mass conservation. The Volume-of-Fluid (VoF) method of Hirt and Nichols [26], represents the distribution of phases in each cell as volume fractions. This method conserves mass exactly, but is somewhat more difficult to advect without significant numerical diffusion. Several schemes to advect the VoF field and maintain a sharp interface between phases exist, including geometric schemes such as Simple Linear Interface Calculation SLIC [27] and Piecewise Linear Interface Calculation PLIC [28]. While geometric schemes are accurate, they are difficult to couple with IB methods, particularly when the surface of the IB is curved, since in principle the interface would need to be constructed only in the portion of the cell that is not occupied by the IB. An alternative to geometric interface advection schemes are so-called compressive interface advection schemes, such as the Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM) of Ubbink and Issa [29]. In compressive interface advection schemes, a downwind component is introduced to the advection scheme, and the Normalized Variable Diagram (NVD) is used to determine the weights of the upwind and downwind components such that the Convective Boundedness Criterion (CBC) is satisfied. This approach can be readily implemented in any IBM framework, since no underlying assumptions about the cell shape are required.

The imposition of surface tension forces on the momentum equation requires the calculation of interface normals and mean curvature. While the evaluation of the interface normals is relatively straight-forward so long as the colour function gradient is available, the estimation of the interface mean curvature is much more challenging. This can in principle be done in a relatively straight-forward manner on regular cartesian grids by using the height function method [26] in conjunction with a contact line model such as the one developed by Afkhami et al. [30]. An issue arises, however, in that it may not be practical to sufficiently extend the VoF field into the IB such that well-defined height functions exist. Instead, a method that has proven to work well with irregular stencils should be chosen. An alternative approach is to use a least-squares type approach, such as the CELESTE method

proposed by Denner and van Wachem [31]. This method was originally proposed for irregular unstructured grids, but can be modified to accommodate other cases where a modified stencil is necessary, as will be shown in this paper.

In this paper, a new method is presented for enforcing contact angle dynamics on arbitrarily shaped IBMs on 2D cartesian grids. A Volume-of-Fluid (VoF) method is used to distinguish the two fluid phases, and a sharp-interface IBM based on the ghost-cell method of Majumdar et al. [32] is implemented. In order to incorporate surface tension and wetting dynamics, a modified version of the CELESTE method [31] is implemented. The ability of the algorithm to enforce wetting conditions on both planar and curved geometries is demonstrated.

2. Numerical methodology

2.1. Governing equations

Unsteady, incompressible viscous flows are governed by the Navier–Stokes equations. For variable density, incompressible flows these consist of the momentum equation,

$$\frac{\partial \rho \bar{\mathbf{u}}}{\partial t} + \bar{\nabla} \cdot (\rho \bar{\mathbf{u}} \bar{\mathbf{u}} + p \bar{\mathbf{I}}) = \bar{\nabla} \cdot (\mu [\bar{\nabla} \bar{\mathbf{u}} + (\bar{\nabla} \bar{\mathbf{u}})^T]) + \bar{\mathbf{f}}_b \quad (2)$$

and the divergence free velocity field condition,

$$\bar{\nabla} \cdot (\bar{\mathbf{u}}) = 0 \quad (3)$$

where ρ is the density, $\bar{\mathbf{u}}$ is the velocity, p is the pressure, μ is the viscosity, $\bar{\mathbf{I}}$ is the identity tensor and $\bar{\mathbf{f}}_b$ are all other body forces, including gravity and surface tension. In volume-of-fluid methods [26], a colour function is transported which represents the volume fraction of one phase with respect to the total cell volume. This equation takes the form

$$\frac{\partial \gamma}{\partial t} + \bar{\nabla} \cdot (\bar{\mathbf{u}} \gamma) = 0 \quad (4)$$

In VoF methods, only a single set of equations is solved, with ρ and μ computed using γ as

$$\rho(\gamma) = (1 - \gamma)\rho_1 + \gamma\rho_2 \quad (5)$$

$$\mu(\gamma) = (1 - \gamma)\mu_1 + \gamma\mu_2 \quad (6)$$

Since $\rho = \rho(\gamma)$, Eq. (4) also serves as the continuity or mass conservation equation. Finally, the body forces $\bar{\mathbf{f}}_b$ in Eq. (2) consist of the surface tension and gravitational forces,

$$\bar{\mathbf{f}}_b = \sigma \kappa \delta \bar{\mathbf{n}} + \rho \bar{\mathbf{g}} \quad (7)$$

where σ is the surface tension coefficient, κ is the interface curvature, δ is the Dirac delta function, $\bar{\mathbf{n}}$ is the interface normal vector and $\bar{\mathbf{g}}$ is the gravitational acceleration.

2.2. Discretization of the Navier–Stokes equations

Eqs. (2) and (3) in this work is discretized by a second order fractional-step method, with body forces and pressure discretized using the face-to-cell approach proposed by Francois et al. [33]. Gravity is included by transforming the gravitational body force into an interface force similar to the approach of Mon-tazeri et al. [34], allowing for $\bar{\mathbf{f}}_b$ to be expressed as $\bar{\mathbf{f}}_b = \bar{\mathbf{f}}_b(\gamma)$.

2.2.1. Numerical discretization

In this section, the discretization of the Navier–Stokes equations is briefly presented. A three-step fractional-step method is used. First, a predicted velocity $\bar{\mathbf{u}}^*$ is solved by discretizing Eq. (2), resulting in

$$\begin{aligned} & \frac{\rho_c^{n+1} \bar{\mathbf{u}}_c^* - \rho_c^n \bar{\mathbf{u}}_c^n}{\Delta t} \Omega_c + \sum_f \bar{\mathbf{F}}_{m,f}^n \bar{\mathbf{u}}_f^* \cdot \bar{\mathbf{s}}_f \\ &= \sum_f \mu_f^n \bar{\nabla} \bar{\mathbf{u}}_f^* \cdot \bar{\mathbf{s}}_f - \Omega_c \rho_c^{n+1} \left(\frac{\bar{\nabla} p_f^n - \bar{\mathbf{f}}_{b,f}^n}{\rho_f^n} \right)_{f \rightarrow c} \end{aligned} \quad (8)$$

where Ω_c is the cell volume, $f \rightarrow c$ denotes a face-to-cell interpolation, and the superscript n denotes a previous time level. Face gradients are discretized using the generalized scheme of Mathur and Murthy [35], and body forces on the right-hand side are interpolated to the cell centres using the method proposed by Francois et al. [33]. The momentum flux $\bar{\mathbf{F}}_{m,f}^n$ must be implemented in a manner consistent with the method used to discretize Eq. (4) as noted by Bussmann et al. [36], and will be discussed further in a following section. In order to transform the gravitational source term $\rho \mathbf{g}$ in Eq. (7) into an interface term, we first employ the substitution

$$p = p_s + \rho \mathbf{g} \cdot \bar{\mathbf{r}} \quad (9)$$

where p_s is the classical interpretation of pressure. Evaluating the gradient we get

$$\bar{\nabla} p = \bar{\nabla} p_s + \bar{\nabla} \rho \mathbf{g} \cdot \bar{\mathbf{r}} + \rho \mathbf{g} \quad (10)$$

and therefore

$$\bar{\nabla} p - \bar{\nabla} \rho \mathbf{g} \cdot \bar{\mathbf{r}} = \bar{\nabla} p_s + \rho \mathbf{g} \quad (11)$$

Since $\rho = \rho(\gamma)$, gravity can be expressed in terms of γ and a density jump $\Delta \rho = \rho_2 - \rho_1$:

$$\bar{\nabla} p_s + \rho \mathbf{g} = \bar{\nabla} p - \Delta \rho \mathbf{g} \cdot \bar{\mathbf{r}} \bar{\nabla} \gamma \quad (12)$$

Employing the CSF method of Brackbill et al. [37] for the discretization of the surface tension force, the full body source term can be expressed as a single interface force of the form

$$\bar{\mathbf{f}}_b^n = (\sigma \kappa - \Delta \rho \mathbf{g} \cdot \bar{\mathbf{r}}) \bar{\nabla} \gamma \quad (13)$$

In the second step we remove $\bar{\nabla} p^n$ from $\bar{\mathbf{u}}_c^*$ and compute $\bar{\mathbf{u}}_c^{**}$. Note that the point of retaining this term in Eq. (8) was to ensure valid boundary conditions were used to compute $\bar{\mathbf{u}}^*$ to ensure a correct projection near boundaries, as discussed in detail by Brown et al. [38]. Thus $\bar{\mathbf{u}}_c^{**}$ becomes

$$\bar{\mathbf{u}}_c^{**} = \bar{\mathbf{u}}_c^* + \Delta t \left(\frac{\bar{\nabla} p_f^n}{\rho_f^n} \right)_{f \rightarrow c} \quad (14)$$

$$\bar{\mathbf{u}}_f^{**} = \bar{\mathbf{u}}^* + \Delta t \left(\frac{\bar{\nabla} p_f^n - \bar{\mathbf{f}}_{b,f}^n}{\rho_f^n} \right)_{f \rightarrow c} + \Delta t \frac{\bar{\mathbf{f}}_{b,f}^{n+1}}{\rho_f^{n+1}} \quad (15)$$

where the bar denotes a quantity that is interpolated using a simple arithmetic mean.

The third step corrects $\bar{\mathbf{u}}^{**}$ to be divergence free. A Hodge decomposition of $\bar{\mathbf{u}}^{**}$ leads to a pressure Poisson equation of the form

$$\Delta t \bar{\nabla} \cdot \left(\frac{\bar{\nabla} p^{n+1}}{\rho^{n+1}} \right) = \bar{\nabla} \cdot (\bar{\mathbf{u}}^{**}) \quad (16)$$

which when discretized becomes

$$\Delta t \sum_f \frac{\bar{\nabla} p_f^{n+1}}{\rho_f^{n+1}} \cdot \bar{\mathbf{s}}_f = \sum_f \bar{\mathbf{u}}_f^{**} \cdot \bar{\mathbf{s}}_f \quad (17)$$

Finally, the cell and face velocities are corrected using

$$\bar{\mathbf{u}}_c^{n+1} = \bar{\mathbf{u}}_c^{**} - \Delta t \left(\frac{\bar{\nabla} p_f^{n+1}}{\rho_f^{n+1}} \right)_{f \rightarrow c} + \Delta t \left(\frac{\bar{\mathbf{f}}_{b,f}^{n+1}}{\rho_f^{n+1}} - \frac{\bar{\mathbf{f}}_{b,f}^n}{\rho_f^n} \right)_{f \rightarrow c} \quad (18)$$

$$\bar{\mathbf{u}}_f^{n+1} = \bar{\mathbf{u}}_f^{**} - \Delta t \frac{\bar{\nabla} p_f^{n+1}}{\rho_f^{n+1}} \quad (19)$$

2.3. Discretization of the VoF equation and momentum flux

In the ghost-cell IB method, exact truncated cell geometries are not used for the finite-volume discretization, and thus using a geometrical interface advection scheme such as Piecewise Linear Interface Construction (PLIC) [39–43] is not feasible. Here, the CIC-SAM scheme of Ubbink and Issa [29] is used instead, with careful consideration to ensure the resulting mass fluxes obtained from Eq. (4) are exactly consistent with the momentum flux $\bar{\mathbf{F}}_{m,f}$ used to advect $\bar{\mathbf{u}}^*$, which as noted by Bussmann et al. [36] is critical for flows with significant density ratios. In this section we describe the method used to ensure consistency between the discretizations of Eqs. (2) and (4). First, the value of γ for a donor cell is normalized according to the Normalized Variable Diagram (NVD) of Leonard [44]

$$\tilde{\gamma}_D = \frac{\gamma_D - \gamma_U}{\gamma_A - \gamma_U} \quad (20)$$

where γ_U , γ_D and γ_A are the values of γ at the upwind, donor and acceptor cells respectively. As proposed by Jasak et al. [45], the value of γ_U can be computed generally as

$$\gamma_U = \gamma_A + 2(\bar{\mathbf{x}}_D - \bar{\mathbf{x}}_A) \cdot \bar{\nabla} \gamma_D \quad (21)$$

where $\bar{\mathbf{x}}_D$ and $\bar{\mathbf{x}}_A$ are the position vectors of the donor and acceptor cells respectively. The two schemes used to compute the normalized value of γ at the face are the Hyper-C (HC) scheme:

$$\tilde{\gamma}_f^{HC} = \begin{cases} \min\{1, \frac{\tilde{\gamma}_D}{C_{0D}}\} & \text{when } 0 \leq \tilde{\gamma}_D \leq 1 \\ \tilde{\gamma}_D & \text{otherwise} \end{cases} \quad (22)$$

and the Ultimate Quickest (UQ) scheme [44]:

$$\tilde{\gamma}_f^{UQ} = \begin{cases} \min\left\{\frac{8C_{0D}\tilde{\gamma}_D + (1-C_{0D})(6\tilde{\gamma}_D+3)}{8}, \tilde{\gamma}_f^{HC}\right\} & \text{when } 0 \leq \tilde{\gamma}_D \leq 1 \\ \tilde{\gamma}_D & \text{otherwise} \end{cases} \quad (23)$$

The HC scheme has a tendency to artificially distort the interface as it moves tangentially to cell faces, while the UQ scheme is overly dissipative. In order to remedy this, a blending function is employed to combine both schemes, such that

$$\tilde{\gamma}_f = \psi_f \tilde{\gamma}_f^{HC} + (1 - \psi_f) \tilde{\gamma}_f^{UQ} \quad (24)$$

where ψ_f is a weighting function that depends on the interface orientation. Ubbink and Issa [29] proposed the following form for ψ_f :

$$\psi_f = \min\left\{k_\psi \frac{\cos(2\theta_f) + 1}{2}, 1\right\} \quad (25)$$

where the angle θ_f is defined as

$$\theta_f = \arccos \left\| \bar{\mathbf{n}} \cdot \frac{\bar{\mathbf{r}}_{cn}}{|\bar{\mathbf{r}}_{cn}|} \right\| \quad (26)$$

The constant k_ψ adjusts the relative dominance of Eqs. (22) and (25). Ubbink and Issa [29] proposed a value of $k_\psi = 1$, which is used in this work. Once $\tilde{\gamma}_f$ is determined from Eq. (24), it is used to compute a weighted linear interpolation parameter β from the equation

$$\beta = \frac{\tilde{\gamma}_f - \tilde{\gamma}_D}{1 - \tilde{\gamma}_D} \quad (27)$$

and the colour function value γ_f is interpolated as

$$\gamma_f = (1 - \beta)\gamma_D + \beta\gamma_A \quad (28)$$

which is then used as the face centered value in the discretized form of Eq. (4). Additionally, the term $\bar{\mathbf{F}}_{m,f}$ in Eq. (8) should be computed as

$$\bar{\mathbf{F}}_{m,f} = [(1 - \beta)\gamma_D \Delta \rho + \beta\gamma_A \Delta \rho + \rho_1] \bar{\mathbf{u}}_f^n \quad (29)$$

Note that Eq. (29) effectively means a different value of ρ_f is used for advecting momentum than is used in Eq. (17). Using this interpolation method alongside the density weighted face-to-centre evaluation of the pressure gradient described by Francois et al. [33] has enabled us to simulate flows of density ratios up to 10^{12} without need for the use of harmonic interpolation of densities at faces, or computing the densities based on the convolved γ field as suggested in [46]. Upon computing the flux term, a simple upwind scheme is used to compute \mathbf{u}^* .

2.4. Computing the surface tension force

In the CSF method of Brackbill et al. [37], the interface force is $\mathbf{f}_i = \sigma \kappa \nabla \gamma$, with the curvature κ computed as

$$\kappa_c = \nabla \cdot (\mathbf{n}) \quad (30)$$

where \mathbf{n} is the interface unit normal vector, defined as

$$\mathbf{n} = -\frac{\nabla \tilde{\gamma}}{\|\nabla \tilde{\gamma}\|} \quad (31)$$

where $\tilde{\gamma}$ is a convolved volume fraction field, explained in the following section.

2.4.1. Computation of the interface normals

The unit normal vector to the interface is computed directly from a smoothed γ field. Using the actual γ field leads to large errors due to the discontinuous nature of the interface. Numerous authors [37,47] have noted that these errors can be greatly reduced by computing the gradient based on a convolved γ field instead, denoted as $\tilde{\gamma}$. Thus, for the purpose of evaluating the interface normals, a convolved volume fraction field $\tilde{\gamma}$ is computed as

$$\tilde{\gamma} = \int_{\Omega} K(\mathbf{x}) \gamma d\Omega \quad (32)$$

where the kernel function $K(\mathbf{x})$ is a simplified version of the one used by Peskin [5], which takes the form

$$K(\mathbf{x}') = \begin{cases} \frac{A}{2r_k} (1 + \cos(\pi r/r_k)) & \text{if } r < r_k \\ 0 & \text{otherwise} \end{cases} \quad (33)$$

where r_k is the effective radius of the kernel, A is a constant specified by the unity property and $r = \|\mathbf{x}' - \mathbf{x}\|$. Through experimentation, it was found that a kernel radius of $r_k = 2\Delta l$, where Δl is a characteristic cell length, resulted in the smallest parasitic currents. In order to compute $\nabla \tilde{\gamma}$ in the CELESTE scheme [31], the Taylor series of $\tilde{\gamma}$ is expanded to second-order from neighbouring cells to the cell centre of interest in the following manner:

$$\tilde{\gamma}_c = \tilde{\gamma}_n + \frac{\partial \tilde{\gamma}_n}{\partial x} \Delta x + \frac{\partial \tilde{\gamma}_n}{\partial y} \Delta y + \frac{\partial^2 \tilde{\gamma}_n}{\partial x^2} \frac{\Delta x^2}{2} + \frac{\partial^2 \tilde{\gamma}_n}{\partial y^2} \frac{\Delta y^2}{2} + \frac{\partial^2 \tilde{\gamma}_n}{\partial x \partial y} \Delta x \Delta y + \mathcal{O}(\Delta x^3, \Delta y^3) \quad (34)$$

Eq. (34) can be arranged into a weighted, over-determined linear system of the form $Ax = b$, in terms of the unknown derivatives, yielding

$$\begin{pmatrix} \frac{\Delta x_1}{\Delta s_1^2} & \frac{\Delta y_1}{\Delta s_1^2} & \frac{\Delta x_1^2}{2\Delta s_1^2} & \frac{\Delta y_1^2}{2\Delta s_1^2} & \frac{\Delta x_1 \Delta y_1}{\Delta s_1^2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\Delta x_i}{\Delta s_i^2} & \frac{\Delta y_i}{\Delta s_i^2} & \frac{\Delta x_i^2}{2\Delta s_i^2} & \frac{\Delta y_i^2}{2\Delta s_i^2} & \frac{\Delta x_i \Delta y_i}{\Delta s_i^2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\Delta x_N}{\Delta s_N^2} & \frac{\Delta y_N}{\Delta s_N^2} & \frac{\Delta x_N^2}{2\Delta s_N^2} & \frac{\Delta y_N^2}{2\Delta s_N^2} & \frac{\Delta x_N \Delta y_N}{\Delta s_N^2} \end{pmatrix} \begin{pmatrix} \frac{\partial \gamma}{\partial x} \\ \frac{\partial \gamma}{\partial y} \\ \frac{\partial^2 \gamma}{\partial x^2} \\ \frac{\partial^2 \gamma}{\partial y^2} \\ \frac{\partial^2 \gamma}{\partial x \partial y} \end{pmatrix} = \begin{pmatrix} \frac{\Delta \gamma_{cn_1}}{\Delta s_1^2} \\ \vdots \\ \frac{\Delta \gamma_{cn_i}}{\Delta s_i^2} \\ \vdots \\ \frac{\Delta \gamma_{cn_N}}{\Delta s_N^2} \end{pmatrix} \quad (35)$$

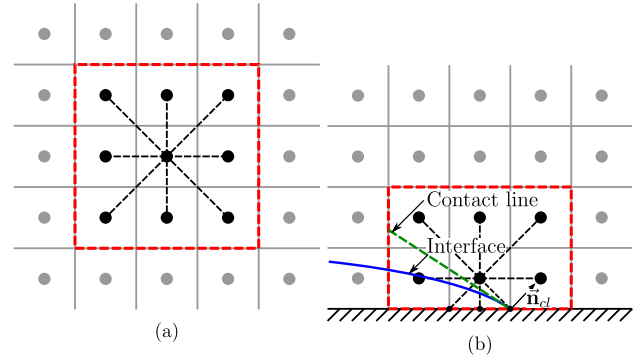


Fig. 1. Least-squares stencil used for both the evaluation of \mathbf{n} and κ , for (a) a regular cell and (b) boundary cell.

Haselbacher and Vasilyev [48] identified two constraints on the use of least-squares methods for reconstructing a gradient from a Taylor series: (1) the stencil should be as small as possible to reduce truncation error, and (2) the stencil should be as symmetric as possible in order to cancel out truncation error terms. Since at least five cells are needed to solve Eq. (35), the cell face and diagonal neighbours are used, resulting in the 3×3 Cartesian stencil shown in Fig. 1.

As per Fig. 1 (b), when adjacent cell neighbours are not available, for example along boundaries, the corresponding face or node is used instead, with the value of $\tilde{\gamma}$ depending on the boundary condition.

2.4.2. Computation of interface curvature

The procedure for computing the curvature follows a similar process to the interface normals. Eq. (30) is expanded as

$$\kappa = \nabla \cdot (\mathbf{n}) = \frac{\partial n_x}{\partial x} + \frac{\partial n_y}{\partial y} \quad (36)$$

A linear system is then solved for each component of \mathbf{n} , denoted by the subscript j , taking the form

$$\begin{pmatrix} \Delta x_1 & \Delta y_1 & \frac{\Delta x_1^2}{2} & \frac{\Delta y_1^2}{2} & \Delta x_1 \Delta y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta x_i & \Delta y_i & \frac{\Delta x_i^2}{2} & \frac{\Delta y_i^2}{2} & \Delta x_i \Delta y_i \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta x_N & \Delta y_N & \frac{\Delta x_N^2}{2} & \frac{\Delta y_N^2}{2} & \Delta x_N \Delta y_N \end{pmatrix} \begin{pmatrix} \frac{\partial n_j}{\partial x} \\ \frac{\partial n_j}{\partial y} \\ \frac{\partial^2 n_j}{\partial x^2} \\ \frac{\partial^2 n_j}{\partial y^2} \\ \frac{\partial^2 n_j}{\partial x \partial y} \end{pmatrix} = \begin{pmatrix} \Delta \mathbf{n}_{j, cn_1} \\ \vdots \\ \Delta \mathbf{n}_{j, cn_i} \\ \vdots \\ \Delta \mathbf{n}_{j, cn_N} \end{pmatrix} \quad (37)$$

where \mathbf{n}_j is a component of the unit normal vector \mathbf{n} . Our numerical experiments have shown consistently that more accurate curvatures are achieved when using an un-weighted least-squares procedure for the computation of interface normal divergence. Upon solving Eq. (37), the appropriate derivatives are substituted into Eq. (36) to obtain the cell centered curvature. This approach differs slightly from the original approach of Denner and van Wachem [31], in that for the evaluation of κ , they considered only the first derivatives of \mathbf{n} were considered. Overall, the use of higher order derivatives was found to produce a smoother curvature field and fewer parasitic currents for the cases studied here.

Finally, unlike the original CELESTE method, no smoothing of the curvature was performed. Instead, the curvature was simply not evaluated in cells where $\|\nabla \tilde{\gamma}_c\| < \epsilon$, where ϵ is a small cut-off value. For interpolating curvatures to faces, $\|\nabla \gamma_f\| < \epsilon$ was also enforced, even if the cell centred gradient was non-zero.

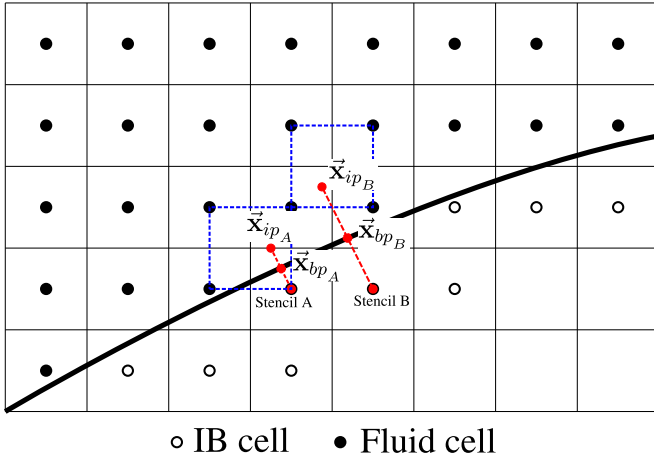


Fig. 2. Interpolation stencil used in the ghost-cell immersed boundary method, showing the image point \vec{x}_{ip} and boundary point \vec{x}_{bp} .

2.5. Ghost-cell immersed boundary method

In recent years, immersed boundaries have received much attention for simulating flows over complex bodies [6]. The “ghost-cell” IB method is a direct forcing method that defines the flow variables within the boundary itself, such that the effect of the boundary is incorporated when flow variables are computed on the cell faces. Our ghost-cell method follows that of Tseng and Ferziger [49] with a slight modification for near-boundary cells. In Tseng and Ferziger’s method [49], a symmetric interpolation stencil is constructed across the boundary as shown in Fig. 2.

The nearest intersection point from the IB cell is found, and an image point in the fluid domain is computed as

$$\vec{x}_{ip} = 2\vec{x}_{bp} - \vec{x}_{gc} \quad (38)$$

The nearest grid node to the image point is then found, and the four cells that connect to this grid node are used for bilinear interpolation. The bilinear polynomial for an arbitrary scalar ϕ is defined as

$$\phi = a_3xy + a_2x + a_1y + a_0 \quad (39)$$

which can be used to compute the value of ϕ within the bounded region. Thus, the four coefficients $a_0 - a_3$ in Eq. (39) must be evaluated. In order to compute these coefficients, a Vandermonde matrix [50] is constructed using the four bounding cell centers shown in Fig. 2

$$\begin{pmatrix} 1 & y_1 & x_1 & x_1y_1 \\ 1 & y_2 & x_2 & x_2y_2 \\ 1 & y_3 & x_3 & x_3y_3 \\ 1 & y_4 & x_4 & x_4y_4 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{pmatrix} \quad (40)$$

In order to implicitly evaluate Eq. (40) when solving Eq. (8), the coefficient matrix is inverted to obtain the unknown coefficients in terms of a linear combination of the values of ϕ at the bounding cell centers. It can easily be shown that an un-modified ghost-cell method can break down if the ghost-cell is close to the IB and is thus part of the image point stencil. In order to avoid this, we use a different approach when the image point stencil includes the ghost-cell. For stencil A in Fig. 2, the boundary point is used instead of the image point, resulting in the following linear equations for Neumann and Dirichlet conditions:

$$\text{Dirichlet: } \sum_s a_i \phi_i = \begin{pmatrix} x_{bp}y_{bp} \\ x_{bp} \\ y_{bp} \\ 1 \end{pmatrix}^T \{V\}^{-1} \{\phi\} \quad (41)$$

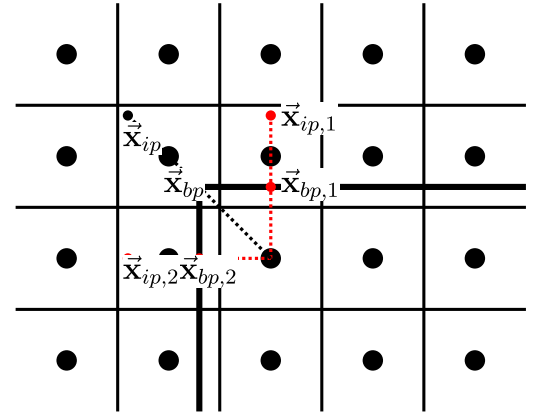


Fig. 3. Degenerate case resulting in an ambiguous choice of interpolation stencil. The red lines show the two possible stencils based on a nearest intersection, while the black stencil is the actual stencil used. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\text{Neumann: } \sum_s a_i \phi_i = \begin{pmatrix} y_{bp}n_x + x_{bp}n_y \\ n_x \\ n_y \\ 0 \end{pmatrix}^T \{V\}^{-1} \{\phi\} \quad (42)$$

where \sum_s denotes a sum over all the stencil cells, $\{V\}$ and $\{\phi\}$ are the Vandermonde and right-hand-side vectors in Eq. (40) respectively, and n_x and n_y denote the x and y components of the unit normal vector of the IB at the boundary point. The above procedure works well when the ghost-cell is close to the IB, but can break down as the IB approaches one of the other image point cells. Therefore, in the case of stencil B, a standard ghost-cell method is used, with the image point value computed as

$$\phi_{ip} = \begin{pmatrix} x_{ip}y_{ip} \\ x_{ip} \\ y_{ip} \\ 1 \end{pmatrix}^T \begin{pmatrix} 1 & y_1 & x_1 & x_1y_1 \\ 1 & y_2 & x_2 & x_2y_2 \\ 1 & y_3 & x_3 & x_3y_3 \\ 1 & y_4 & x_4 & x_4y_4 \end{pmatrix}^{-1} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{pmatrix} \quad (43)$$

The Dirichlet and Neumann conditions can then be evaluated along the symmetric stencil as follows:

$$\text{Dirichlet: } \frac{\phi_{ip} + \phi_{gc}}{2} = \phi_{bp} \quad (44)$$

$$\text{Neumann: } \frac{\phi_{gc} - \phi_{ip}}{\Delta l} = \phi_{bp} \quad (45)$$

where Δl is the scalar length of the interpolation stencil. Finally, if Robin boundary conditions are desired, a linear combination of the of Dirichlet and Neumann conditions proposed above can be used.

2.5.1. Degenerate cases

Degenerate cases can make the construction of an IB stencil ambiguous. For example, an IB cell centre may lie at an equal distance from two edges of a polygonal IB, such as the case depicted in Fig. 3. Cases such as this can lead to significant issues in simulations where mass conservation is important. In order to minimize the impact of such degenerate cases, two possible remedies exist. One approach is to overlay a second cell on top of the IB cell, effectively allowing for two stencils to be constructed, but introducing an extra unknown into the problem. While effective and conceptually simple, its implementation involves a lot of programming overhead since the cell connectivity has been altered. Instead, the approach applied here is when more than one boundary point candidate is found within some tolerance, the nearest IB vertex is used instead.

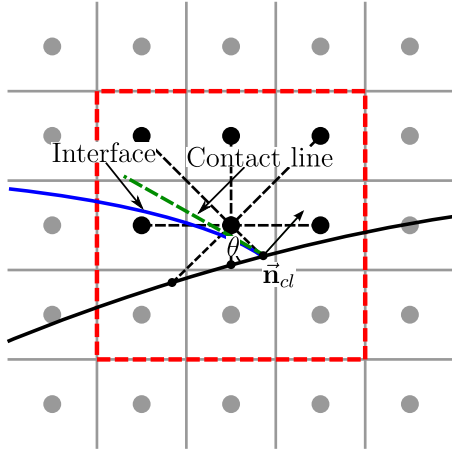


Fig. 4. Least-squares stencil used for the evaluation of κ near the IB surface.

2.6. Implementing contact line dynamics on ghost-cell immersed boundaries

Although IB methods have seen increasing attention over the years due to their versatility, efficiency and increasing accuracy, little work has been done on the development of multiphase models coupled to IB methods. Liu and Ding [23] developed a moving contact line model for the Direct Forcing (DF) IB method of Ren et al. [51], using a diffuse-interface multiphase model. Zhang et al. [19], and Fotis and Yang [7] used an IB model to simulate wave-structure interactions at free surfaces, but did not consider surface tension effects. Niels et al. [52] used IB methods for multiphase flows, but did not consider contact line dynamics. Ghasemi et al. [21] studied interactions between moving solids and two-phase systems using a PLIC-VOF multiphase method, but also did not consider contact line dynamics. Here we propose a simple method for modelling contact angle dynamics based on the CELESTE algorithm of Denner and van Wachem [31], which is effective even for curved surfaces.

2.6.1. Computing the interface curvature near the immersed boundary

In order to compute the interface curvature in cells neighbouring an IB cell, we employ a straight-forward method involving a modified CELESTE stencil. The orientation of the interface normal at the contact line, \mathbf{n}_{cl} , where the stencil intersects the IB is known, since it can be determined using the prescribed contact angle. Thus, the CELESTE stencil in Fig. 1 is modified near the IB as shown in Fig. 4.

At each point where the stencil intersects the IB, the value of \mathbf{n}_{cl} is used instead of a value of \mathbf{n} calculated from Eq. (31) for the calculation in Eq. (37). Therefore, cells which neighbour the IB account for the proximity of the IB when computing κ , creating a wetting force in Eq. (2) when the interface intersects the IB at an angle that deviates from the prescribed contact angle. For the computation of $\nabla \tilde{\gamma}$, an unmodified CELESTE stencil which includes the ghost-cells is used.

3. Results

A number of test cases are presented to demonstrate that the proposed numerical scheme enforces the correct wetting dynamics along the surface of an immersed boundary. The first case simply demonstrates the ability of the proposed algorithm to enforce the contact angle on a curved surface. Then, we investigate the quality of the results for both well-aligned and poorly aligned IBs. We then investigate two applications, a planar droplet impact onto a cylinder and a planar droplet penetration into a porous media.

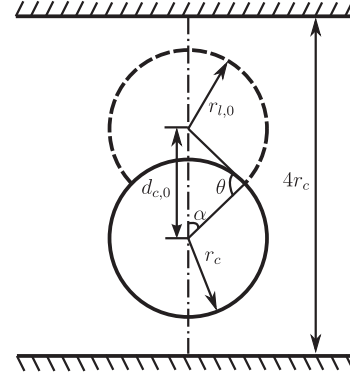


Fig. 5. Initial configuration for a planar droplet on a horizontal cylinder.

For all cases, the time-step is constrained by imposing a maximum allowable CFL number of 0.25, a condition identified by Ubbink and Issa [29] as necessary for maintaining a sharp interface when using a CICSAM advection scheme, as well as by the capillary time-step constraint identified by Brackbill et al. [37] defined as $\Delta t_\sigma \leq \sqrt{\frac{(\rho_1 + \rho_2) \Delta l^3}{4\pi\sigma}}$ where Δl is a characteristic cell length scale defined as $\Delta l = \min \|\vec{r}_{cn}\|$.

3.1. Imposing a static contact angle on an IB cylinder

In this test case, the ability of the proposed method to maintain a contact angle on a cylindrical geometry is demonstrated. The cylinder has a radius $r_c = 0.5$ mm, and the droplet is initialized with a radius of curvature $r_{l,0} = 0.5$ mm, and is initially spaced such that the interface intersects the cylinder at $\theta = 90^\circ$. Fig. 5 depicts the configuration of the computational domain.

In the absence of gravity, the equilibrium configuration can be computed analytically. The area of the planar droplet can be computed as

$$A_0 = \pi r_{l,0}^2 - r_c^2 \arccos\left(\frac{d_{c,0}^2 + r_c^2 - r_{l,0}^2}{2d_{c,0}r_c}\right) - r_{l,0}^2 \arccos\left(\frac{d_{c,0}^2 + r_{l,0}^2 - r_c^2}{2d_{c,0}r_{l,0}}\right) + \sqrt{\frac{(-d_{c,0} + r_c + r_{l,0})(d_{c,0} + r_c - r_{l,0})(d_{c,0} - r_c + r_{l,0})(d_{c,0} + r_c + r_{l,0})}{4}} \quad (46)$$

where $r_{l,0}$ is the initial radius of curvature, r_c is the radius of the cylinder, and $d_{c,0}$ is the initial spacing between the centres of the cylinder and the droplet. The final configuration of the droplet can then be found in terms of the spacing d_c and droplet radius of curvature r_l by finding the roots of the equation

$$f(d_c) = A_0 - (\alpha + \theta)r_l^2 + \alpha r_c^2 - r_l r_c \sin \theta \quad (47)$$

where

$$\alpha(d_c) = \arccos\left(\frac{r_c^2 + d_c^2 - r_l^2}{2r_c d_c}\right) \quad (48)$$

$$r_l(d_c) = r_c \cos \theta + 2\sqrt{r_c^2 \cos^2 \theta - r_c^2 + d_c^2} \quad (49)$$

Five different contact angles were tested to demonstrate the ability of the numerical scheme to handle a wide range of wettability, on an equidistant cartesian mesh with a resolution of 225×300 . This results in 150 cells across the horizontal diameter of the cylinder. The contact angles tested were 30° , 60° , 90° , 120° and 150° . In each case the initial interface is configured such that it intersects the IB surface at 90° , and the simulation is allowed to run until the velocity residuals decay to a reasonably

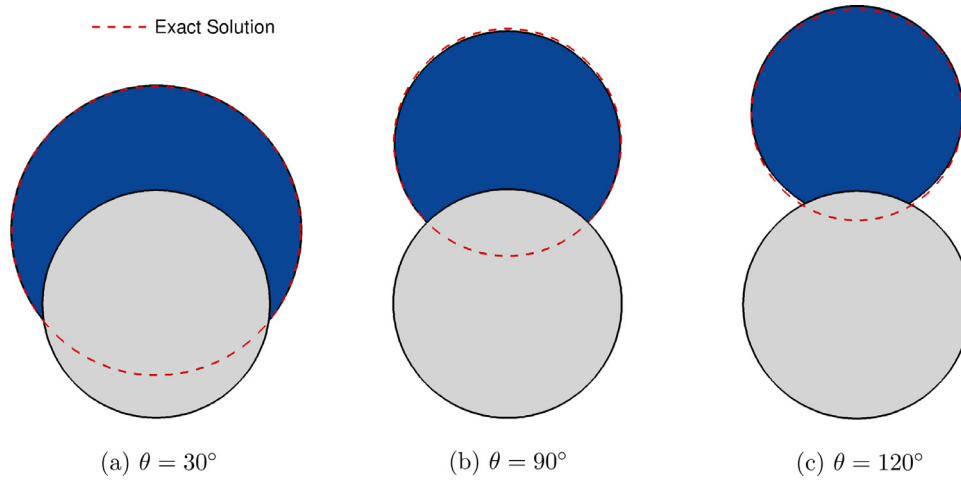


Fig. 6. Static contact angles imposed on a cylindrical IB.

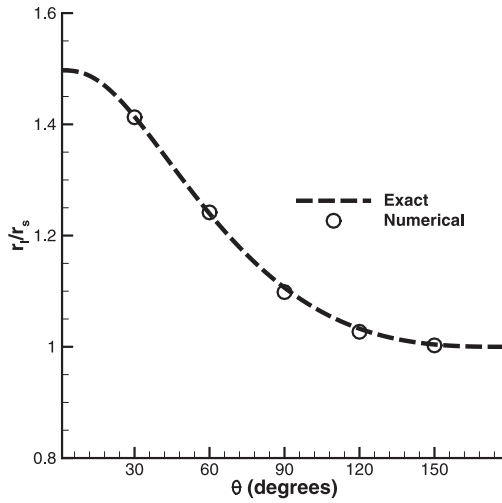


Fig. 7. Numerically computed droplet radii at various contact angles.

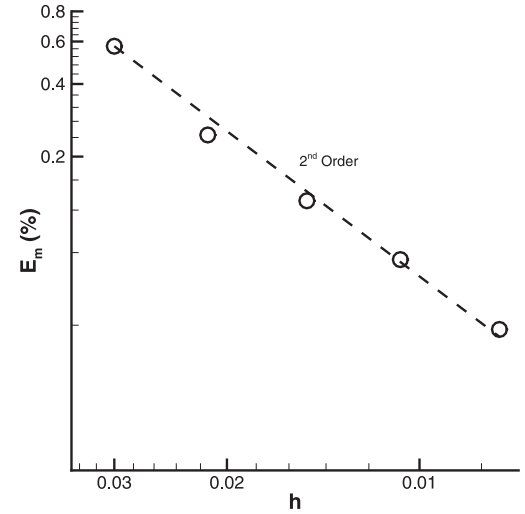
small value. The final contact angle was measured by fitting a circle to the $\gamma = 0.5$ contour, and computing its intersection angle with the cylinder. Fig. 6 depicts the final numerically obtained interface configurations at varying contact angles. (Fig. 7)

The hydrophilic cases produced the largest errors in phase conservation. This is likely a result of the large area projection of the planar droplet onto the IB [53], as well as the higher overall spreading rate. The sharp-interface ghost-cell immersed boundary method has an advantage in that when the IB aligns well with adjacent cell faces, the boundary condition reduces to the face value as if an actual boundary were present. When the mesh is not well-aligned, an error in phase conservation is introduced. Fig. 8 shows the trend of mass error, as a percentage loss/gain in total volume of the immersed phase, as a function of mesh spacing h for the 30° case.

The mass loss itself is due to the well-known issue of transpiration error in IB methods. The mass conservation can be quantified by considering the phase conservation equation for an IB cell at steady state:

$$\frac{d\gamma}{dt} = - \int_{S_b} \mathbf{u}_f \gamma_f \cdot \mathbf{n} dS \quad (50)$$

where S_b denotes the IB surface. When an IB cell is discretized on an equidistant cartesian mesh, and assuming a no-slip boundary condition for velocity, and a Neumann condition for γ , this equa-

Fig. 8. Mass loss error per unit non-dimensional time for the planar droplet situated on a cylinder with $\theta = 30^\circ$. Note that for this case, $t_{\max}^* = 0.2$.

tion becomes

$$\frac{d\gamma}{dt} = - \sum_f \frac{1}{4} (\mathbf{u}_p - \mathbf{u}_{ip}) \cdot \mathbf{s}_f (\gamma_p + \gamma_{ip}) \quad (51)$$

which approaches zero if the solution is at steady state, or in the infinitesimal limit where $\phi_p = \phi_{ip}$. As seen in Fig. 8, the order of convergence with respect to mass error is consistent with the known order of accuracy of the bilinear interpolation used in the ghost-cell method. A detailed summary of all the known mass conservation errors associated with the ghost-cell IB method can be found in Seo and Mittal [54]. Finally, the convergence of the contact angle was tested for five different mesh resolutions. The grid convergence of the $\theta = 30^\circ$ case is depicted in Fig. 9. The method converges rapidly with mesh refinement, with the solution on the three finest meshes being nearly indistinguishable.

3.2. Imposing a contact angle on a poorly mesh-aligned plane

In this section, initially semi-circular planar droplets are placed on rectangular IBs, which are oriented at either 0° or 45° . The first represents an optimal alignment of the IB with respect to the background mesh, while the latter represents a worst case alignment. No gravity is present, thus only the wetting force drives the

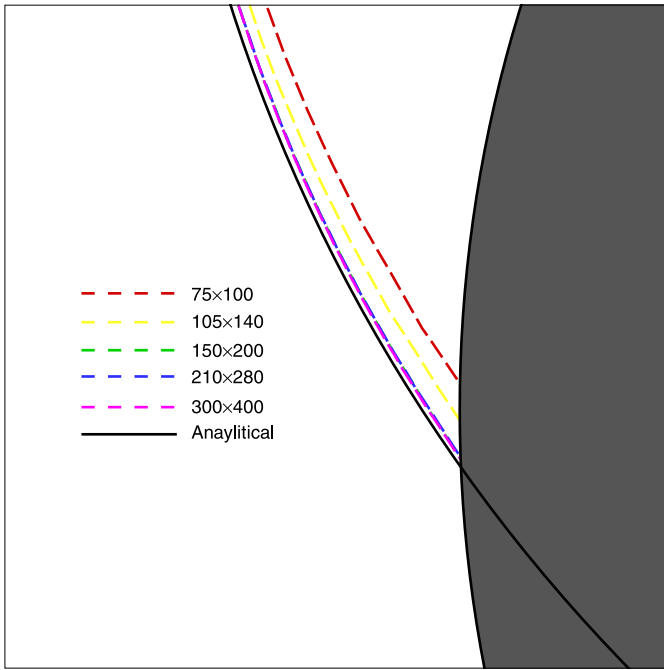


Fig. 9. Convergence of the proposed method with grid refinement, with $\theta = 30^\circ$.

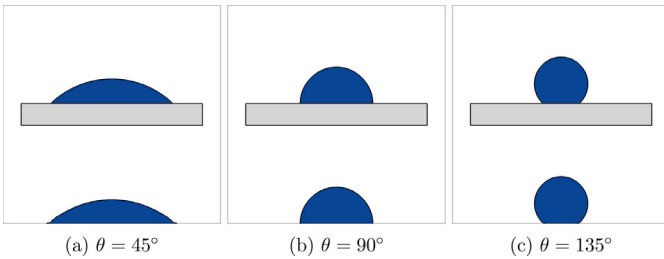


Fig. 10. Planar droplets situated on a flat, well-aligned IB. The bottom droplets are placed on a regular domain boundary for reference.

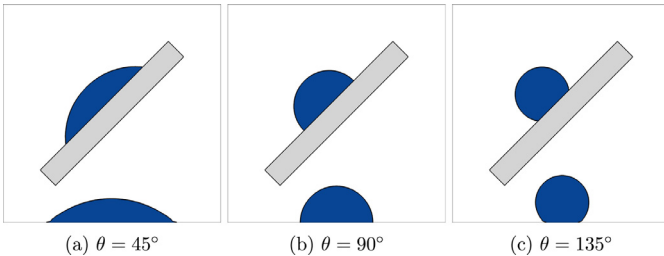


Fig. 11. Planar droplets situated on a tilted, poorly-aligned IB. The bottom droplets are placed on a regular domain boundary for reference.

droplet to spread or contract. The Ohnesorge number for all cases is $Oh = 0.1$. Figs. 10 and 11 depict the obtained solutions for the flat and tilted platforms respectively.

Additionally, it can easily be shown that the internal pressure for a planar droplet in zero gravity as a function of contact angle and area is:

$$p = \sigma \sqrt{\frac{2\theta - \sin(2\theta)}{2A}} \quad (52)$$

Fig. 12 depicts the numerically computed internal pressure for both the flat and tilted planes at various contact angles.

A slight error in the droplet internal pressure is observed between the flat and tilted platforms, particularly in the hydrophilic

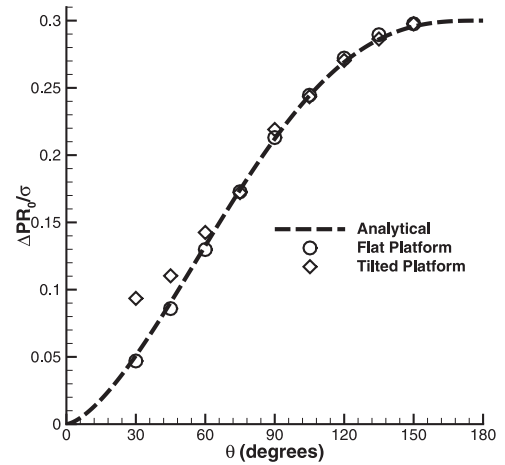


Fig. 12. Internal droplet pressure at varying contact angles.

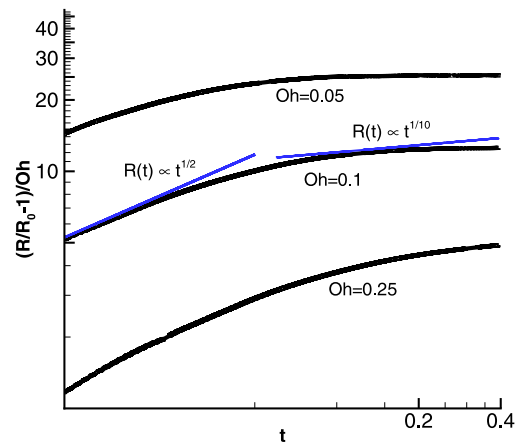


Fig. 13. Non-dimensional droplet radius with respect to time for a planar droplet on the flat platform, with $\theta = 30^\circ$ for various Ohnesorge numbers.

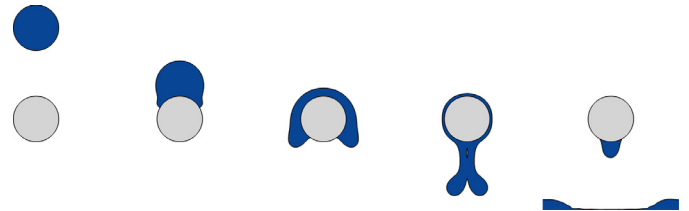


Fig. 14. Planar droplet impact onto a hydrophilic ($\theta = 30^\circ$) cylinder. Frames correspond to $t = 0$ s, $t = 0.05$ s, $t = 0.07$ s, $t = 0.09$ s and $t = 0.2$ s.

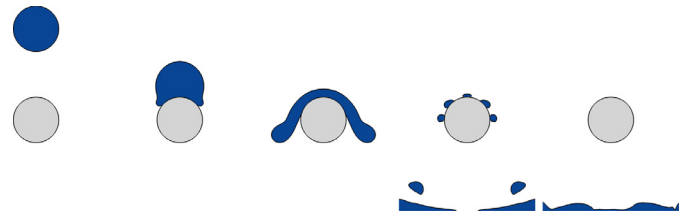


Fig. 15. Planar droplet impact onto a hydrophobic ($\theta = 150^\circ$) cylinder, at $t = 0$ s, $t = 0.05$ s, $t = 0.07$ s, $t = 0.1$ s and $t = 0.2$ s.

cases. The difference in error between the hydrophilic and hydrophobic platforms is due to greater error in extrapolating the γ field to the ghost-cells from the fluid domain at small contact angles, resulting in greater errors in the computation of $\nabla\gamma$ near where the fluid-fluid interface intersects the IB. This issue becomes

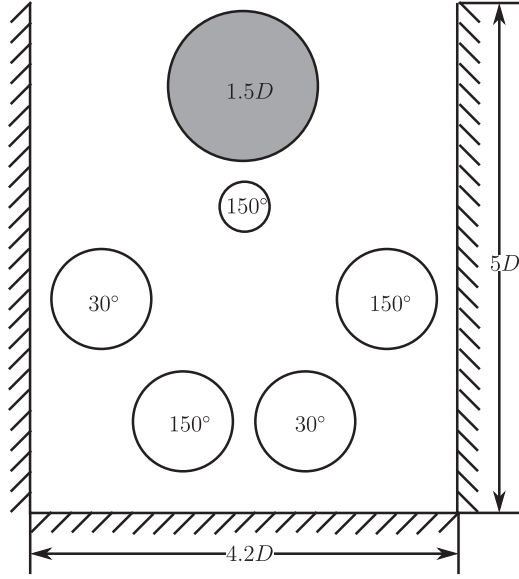


Fig. 16. Droplet penetration into porous media case.

most pronounced when the mesh alignment is poor, resulting in a larger IB stencil length.

Finally, in order to validate the wetting force on the platform, the spreading rate was compared to the well-known Tanner law [55]. For viscous flows, droplet spreading can be characterized into two regimes; an early inertial spreading phase in which the non-dimensional droplet radius changes as $R(t) \propto t^{1/2}$, and a viscous spreading phase in which $R(t) \propto t^{1/10}$.

Fig. 13 shows reasonably good asymptotic agreement between the Tanner law and the numerically computed spreading rate.

3.3. Droplet impact onto a cylinder

In this test case, a planar water droplet in an air environment ($\rho_w/\rho_a \approx 800$, $\mu_w/\mu_a \approx 50$), with a diameter of 1 cm, is placed above a cylinder of the same diameter. The droplet is initially at rest, and is allowed to fall under the influence of gravity until impacting the cylinder. Two different contact angles are compared, 30° and 150°; results are illustrated in Figs. 14 and 15, respectively, demonstrating the effect wettability has on the impact dynamics.

Although it is not possible to create a planar droplet impact in reality, similar experiments have been carried out for droplets onto spherical objects [56]. Droplet impacts are typically characterized by two different phases: a fast, inertia driven phase fol-

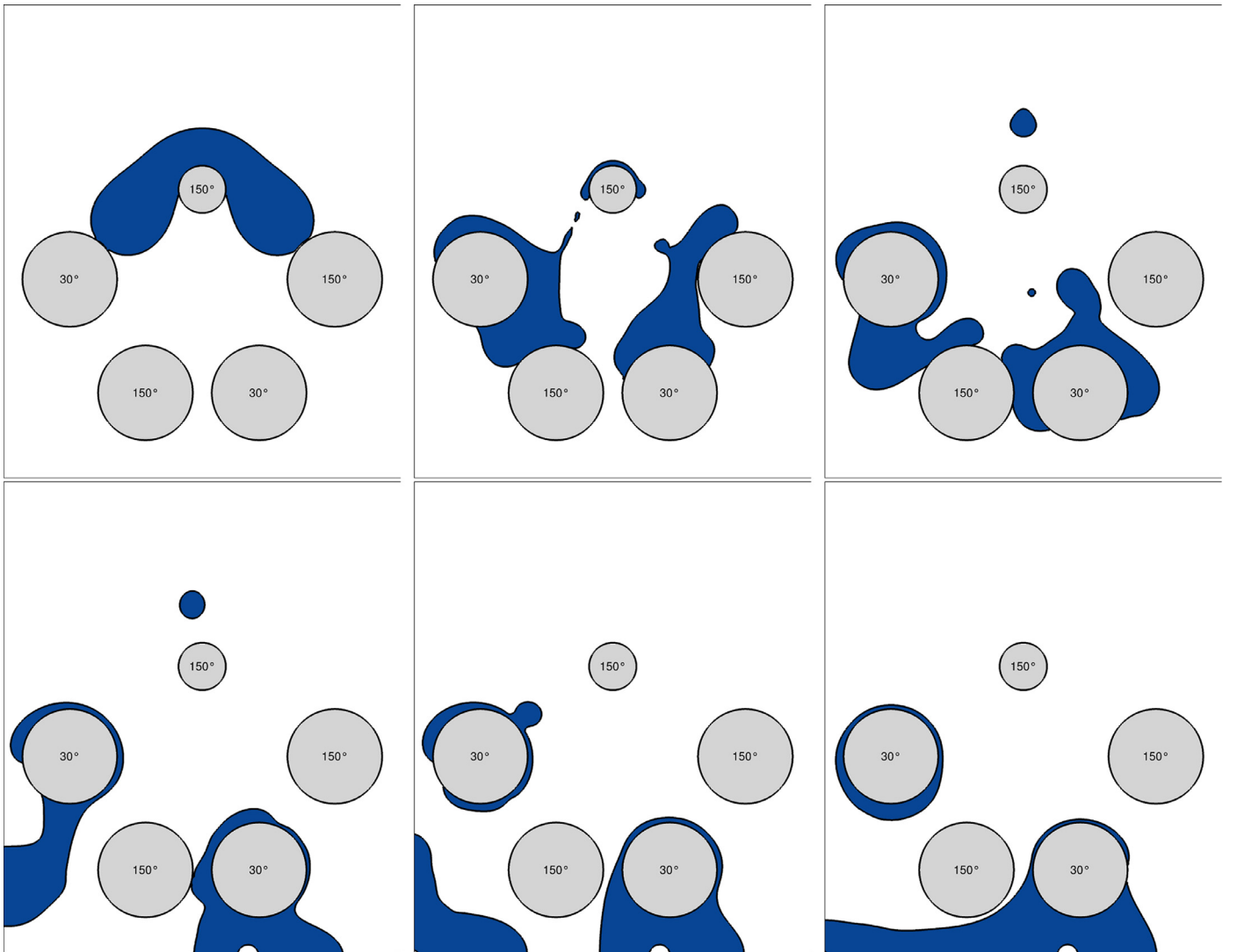


Fig. 17. Penetration of a planar droplet into multiple cylinders of varying wettability; snapshots at $t^* = 5$, $t^* = 8$, $t^* = 10$, $t^* = 15$, $t^* = 20$ and $t^* = 30$.

Table 1
Non-dimensional parameters for droplet penetration into a porous media.

$\lambda_\rho = \frac{\rho_1}{\rho_2}$	$\lambda_\mu = \frac{\mu_1}{\mu_2}$	$Re = \frac{\rho_2 u_0 D}{\mu_2}$	$We = \frac{\rho_2 u_0^2 D}{\sigma}$	$Fr = \frac{u_0}{\sqrt{gD}}$
0.001	0.025	62	1	0.44

lowed by a slow phase driven by capillary forces [57]. Comparison of the initial impact phase in Figs. 14 and 15 at $t = 0.05$ reveals an initial spreading that is similar in both cases, as inertial effects overwhelm the capillary forces. As the droplet spreading slows and wetting forces become more dominant, the two cases diverge. The hydrophilic cylinder is covered completely in a thin liquid film, while the liquid on the hydrophobic cylinder beads up before separating completely from the cylinder.

3.4. Droplet penetration into a porous media

As a final illustration of the efficacy of the method presented here, the 2D planar droplet penetration into a porous media simulation studied by Liu and Ding [23] is presented. A schematic of the setup for the problem is shown in Fig. 16. For this case, dimensionless parameters are used. The droplet has a diameter of 1.5D; cylinders of diameter D are arranged at (1.5, 0.9), (2.7, 0.9), (0.7, 2.1), (3.5, 2.1), and one cylinder of diameter 0.5D is placed at (2.1, 3.05). The domain is discretized on a uniform mesh of dimensions 320×400 . The droplet starts from rest, falls under the influence of gravity only. The dimensionless parameters are defined in Table 1. The characteristic velocity u_0 is defined to be the droplet speed just before impacting the first cylinder.

As can be seen in Fig. 17, the droplet is split symmetrically by the first hydrophobic cylinder. The symmetry is broken when the interfaces on either side make contact with the top left hydrophilic cylinder and the top right hydrophobic cylinder. Eventually the two halves separate completely, and a small satellite droplet is formed on the top hydrophobic cylinder, which is repulsed. Finally, the two halves come to rest, and only the hydrophilic cylinders remain “wet”.

4. Conclusion

A novel algorithm for imposing contact angles on ghost-cell immersed boundaries was presented. The algorithm combines sharp-interface, ghost-cell immersed boundaries [32] and the CELESTE method for evaluating curvatures [31]. The ability of the algorithm to handle surfaces of differing wettability was demonstrated. Thus, rather than imposing an angle at which the interface intersects a boundary, the curvature itself is computed by incorporating the orientation of the contact line, resulting in a momentum source that accounts for varying wettability.

Future work will focus on extending the proposed methods to moving boundaries in order to study particle interactions at fluid-fluid interfaces, as well as an extension to 3D. In principle the method presented should extend to 3D in a relatively straightforward manner. One additional challenge is that in 3D, there is not a single unique tangent vector along the IB surface, thus determining the contact line normal along the IB surface requires information from the most recent iteration of the VoF field. This interface vector must then be projected onto the IB surface first in order to determine the contact line normal. Additionally, in 3D a hexagonal cell has 26 neighbouring cells on its faces, edges and vertices, as opposed to only 8 neighbours in 2D. Thus, extending to 3D will require some numerical experimentation in order to determine which Taylor series terms should be retained in the least-squares procedure.

References

- [1] Mehrabian S, Bussmann M, Acosta E. Breakup of high solid volume fraction oilparticle cluster in simple shear flow. *Colloids Surf A* 2015;483:25–35. doi:10.1016/j.colsurfa.2015.06.054.
- [2] Soltman D, Smith B, Morris S, Subramanian V. Inkjet printing of precisely defined features using contact-angle hysteresis. *J Colloid Interface Sci* 2013;400:135–9. doi:10.1016/j.jcis.2013.03.006.
- [3] Hazra M, Dasgupta Mandal D, Mandal T, Bhuniya S, Ghosh M. Designing polymeric microparticle drug delivery system for hydrophobic drug quercetin. *Saudi Pharm J* 2015;23(4):429–36. doi:10.1016/j.jsps.2015.01.007.
- [4] Karayiannis T, Mahmoud M. Flow boiling in microchannels: fundamentals and applications. *Appl Therm Eng* 2017;115:1372–97. doi:10.1016/j.applthermaleng.2016.08.063.
- [5] Peskin CS. Numerical analysis of blood flow in the heart. *J Comput Phys* 1977;25(3):220–52. doi:10.1016/0021-9991(77)90100-0.
- [6] Mittal R, Iaccarino G. Immersed boundary methods. *Annu Rev Fluid Mech* 2005;37(1):239–61. doi:10.1146/annurev.fluid.37.061903.175743.
- [7] Sotiropoulos F, Yang X. Immersed boundary methods for simulating fluid-structure interaction. *Prog Aerosp Sci* 2014;65:1–21. doi:10.1016/j.paerosci.2013.09.003.
- [8] Sigenza J, Mendez S, Ambard D, Dubois F, Jourdan F, Mozul R, et al. Validation of an immersed thick boundary method for simulating fluid-structure interactions of deformable membranes. *J Comput Phys* 2016;322:723–46. doi:10.1016/j.jcp.2016.06.041.
- [9] Sarofeen C.M, Noack R.W, Kreeger R.E. A non-cut cell immersed boundary method for use in icing simulations. 50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition 2012.
- [10] Huang J, Carica PM, Stern F. Semi-coupled air/water immersed boundary approach for curvilinear dynamic overset grids with application to ship hydrodynamics. *Int J Numer Methods Fluids* 2008;58(6):591–624. doi:10.1002/ldf.1758.
- [11] Afkhami S, Zaleski S, Bussmann M. A mesh-dependent model for applying dynamic contact angles to VOF simulations. *J Comput Phys* 2009;228(15):5370–89. doi:10.1016/j.jcp.2009.04.027.
- [12] Fan ES-C, Bussmann M. Piecewise linear volume tracking in spherical coordinates. *Appl Math Model* 2013;37(5):3077–92. doi:10.1016/j.apm.2012.07.006.
- [13] Guillaumet R, Vincent S, Caltagirone J-P. An original algorithm for VOF based method to handle wetting effect in multiphase flow simulation. *Mech Res Commun* 2015;63:26–32. doi:10.1016/j.mechrescom.2014.11.002.
- [14] Mahady K, Afkhami S, Kondic L. A volume of fluid method for simulating fluid/fluid interfaces in contact with solid boundaries. *J Comput Phys* 2015. doi:10.1016/j.jcp.2015.03.051.
- [15] Malgarinos I, Nikolopoulos N, Marengo M, Antonini C, Gavaises M. VOF simulations of the contact angle dynamics during the drop spreading: standard models and a new wetting force model. *Adv Colloid Interface Sci* 2014;212:1–20. doi:10.1016/j.cis.2014.07.004.
- [16] Sui Y, Ding H, Speltz PD. Numerical simulations of flows with moving contact lines. *Annu Rev Fluid Mech* 2014;46(1):97–119. doi:10.1146/annurev-fluid-010313-141338.
- [17] Yang J, Stern F. Sharp interface immersed-boundary/level-set method for wave-body interactions. *J Comput Phys* 2009;228(17):6590–616. doi:10.1016/j.jcp.2009.05.047.
- [18] Zhang C, Zhang W, Lin N, Tang Y, Zhao C, Gu J, et al. A two-phase flow model coupling with volume of fluid and immersed boundary methods for free surface and moving structure problems. *Ocean Eng* 2013;74:107–24. doi:10.1016/j.oceaneng.2013.09.010.
- [19] Zhang C, Lin N, Tang Y, Zhao C. A sharp interface immersed boundary/VOF model coupled with wave generating and absorbing options for wave-structure interaction. *Comput Fluids* 2014;89:214–31. doi:10.1016/j.compfluid.2013.11.004.
- [20] Calderer A, Kang S, Sotiropoulos F. Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures. *J Comput Phys* 2014;277:201–27. doi:10.1016/j.jcp.2014.08.010.
- [21] Ghasemi A, Pathak A, Raessi M. Computational simulation of the interactions between moving rigid bodies and incompressible two-fluid flows. *Comput Fluids* 2014;94:1–13. doi:10.1016/j.compfluid.2014.01.027.
- [22] Yang J, Stern F. A non-iterative direct forcing immersed boundary method for strongly-coupled fluid-solid interactions. *J Comput Phys* 2015;295:779–804. doi:10.1016/j.jcp.2015.04.040.
- [23] Liu HR, Ding H. A diffuse-interface immersed-boundary method for two-dimensional simulation of flows with moving contact lines on curved substrates. *J Comput Phys* 2015;294:484–502. doi:10.1016/j.jcp.2015.03.059.
- [24] Tryggvason G, Bunner B, Esmaeeli A, Juric D, Al-Rawahi N, Tauber W, et al. A front-tracking method for the computations of multiphase flow. *J Comput Phys* 2001;169:708–59.
- [25] Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *J Comput Phys* 1994;114(1):146–59. doi:10.1006/jcph.1994.1155.
- [26] Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *J Comput Phys* 1981;39:201–25.
- [27] van de Vooren AJ, Zandbergen PJ, editors. SLIC (simple line interface calculation). Berlin, Heidelberg: Springer Berlin Heidelberg; 1976.
- [28] Youngs DL. Time-dependent multi-material flow with large fluid distortion. *Numer Methods Fluid Dyn* 1982;24(2):273–85.
- [29] Ubbink O, Issa R. A method for capturing sharp fluid interfaces on arbitrary meshes. *J Comput Phys* 1999;153(1):26–50. doi:10.1006/jcph.1999.6276.

- [30] Afkhami S. Applying Dynamic Contact Angles To A Three-Dimensional VOF Model. University of Toronto; 2007. Ph.D. thesis.
- [31] Denner F, van Wachem BGM. Fully-coupled balanced-force VOF framework for arbitrary meshes with least-Squares curvature evaluation from volume fractions. *Numer Heat Transfer, Part B* 2014;65(3):218–55. doi:[10.1080/10407790.2013.849996](https://doi.org/10.1080/10407790.2013.849996).
- [32] Majumdar S, Iaccarino G, Durbin P. RANS solvers with adaptive structured boundary non-conforming grids. *Ann Res Briefs* 2001:353–66.
- [33] Francois MM, Cummins SJ, Dendy ED, Kothe DB, Sicilian JM, Williams MW. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *J Comput Phys* 2006;213(1):141–73. doi:[10.1016/j.jcp.2005.08.004](https://doi.org/10.1016/j.jcp.2005.08.004).
- [34] Montazeri H, Bussmann M, Mostaghimi J. Accurate implementation of forcing terms for two-phase flows into SIMPLE algorithm. *Int J Multiphase Flow* 2012;45:40–52. doi:[10.1016/j.ijmultiphaseflow.2012.05.003](https://doi.org/10.1016/j.ijmultiphaseflow.2012.05.003).
- [35] Mathur SR, Murthy JY. A pressure-based method for unstructured meshes. *Numer Heat Transfer, Part B* 1997;31(2):195–215. doi:[10.1080/10407799708915105](https://doi.org/10.1080/10407799708915105).
- [36] Bussmann M, Kothe D, Sicilian J. Modeling high density ratio incompressible flows. In: *Proceedings of ASME fluids engineering division summer meeting*; 2002.
- [37] Brackbill J, Kothe D, Zemach C. A continuum method for modeling surface tension. *J Comput Phys* 1992;100(2):335–54. doi:[10.1016/0021-9991\(92\)90240-Y](https://doi.org/10.1016/0021-9991(92)90240-Y).
- [38] Brown DL, Cortez R, Minion ML. Accurate projection methods for the incompressible Navier–Stokes equations. *J Comput Phys* 2001;168:464–99.
- [39] Rider WJ, Kothe DB. Reconstructing volume tracking. *J Comput Phys* 1998;141(2):112–52. doi:[10.1006/jcph.1998.5906](https://doi.org/10.1006/jcph.1998.5906).
- [40] Scardovelli R, Zaleski S. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *J Comput Phys* 2000;164(1):228–37. doi:[10.1006/jcph.2000.6567](https://doi.org/10.1006/jcph.2000.6567).
- [41] Ubbink O. Numerical prediction of two fluid systems with sharp interfaces. Imperial College of Science, Technology & Medicine; 1997. Ph.D. thesis.
- [42] López J, Hernández J, Gómez P, Faura F. An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows. *J Comput Phys* 2005;208(1):51–74. doi:[10.1016/j.jcp.2005.01.031](https://doi.org/10.1016/j.jcp.2005.01.031).
- [43] López J, Hernández J, Gómez P, Faura F. A new volume conservation enforcement method for PLIC reconstruction in general convex grids. *J Comput Phys* 2016;316:338–59. doi:[10.1016/j.jcp.2016.04.018](https://doi.org/10.1016/j.jcp.2016.04.018).
- [44] Leonard B. The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection. *Comput Methods Appl Mech Eng* 1991;88(1):17–74. doi:[10.1016/0045-7825\(91\)90232-U](https://doi.org/10.1016/0045-7825(91)90232-U).
- [45] Jasak H, Weller HG, Gosman AD. High resolution NVD differencing scheme for arbitrarily unstructured meshes. *Int J Numer Methods Fluids* 1999;449:431–49.
- [46] Denner F, van Wachem BG. On the convolution of fluid properties and surface force for interface capturing methods. *Int J Multiphase Flow* 2013;54:61–4. doi:[10.1016/j.ijmultiphaseflow.2013.03.004](https://doi.org/10.1016/j.ijmultiphaseflow.2013.03.004).
- [47] Cummins SJ, Francois MM, Kothe DB. Estimating curvature from volume fractions. *Comput Struct* 2005;83(6–7):425–34. doi:[10.1016/j.compstruc.2004.08.017](https://doi.org/10.1016/j.compstruc.2004.08.017).
- [48] Haselbacher A, Vasilyev OV. Commutative discrete filtering on unstructured grids based on least-squares techniques. *J Comput Phys* 2003;187:197–211. doi:[10.1016/S0021-9991\(03\)00095-0](https://doi.org/10.1016/S0021-9991(03)00095-0).
- [49] Tseng Y-H, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. *J Comput Phys* 2003;192(2):593–623. doi:[10.1016/j.jcp.2003.07.024](https://doi.org/10.1016/j.jcp.2003.07.024).
- [50] Trefethen LN, Bau D. *Numerical linear algebra*. SIAM; 1997.
- [51] Ren W, Shu C, Wu J, Yang W. Boundary condition-enforced immersed boundary method for thermal flow problems with Dirichlet temperature condition and its applications. *Comput Fluids* 2012;57:40–51. doi:[10.1016/j.compfluid.2011.12.006](https://doi.org/10.1016/j.compfluid.2011.12.006).
- [52] Deen NG, Annaland MvS, Kuipers J. Direct numerical simulation of complex multi-fluid flows using a combined front tracking and immersed boundary method. *Chem Eng Sci* 2009;64(9):2186–201. doi:[10.1016/j.ces.2009.01.029](https://doi.org/10.1016/j.ces.2009.01.029).
- [53] Liu C, Hu C. An efficient immersed boundary treatment for complex moving object. *J Comput Phys* 2014;274:654–80. doi:[10.1016/j.jcp.2014.06.042](https://doi.org/10.1016/j.jcp.2014.06.042).
- [54] Seo JH, Mittal R. A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. *J Comput Phys* 2011;230(19):7347–63. doi:[10.1016/j.jcp.2011.06.003](https://doi.org/10.1016/j.jcp.2011.06.003).
- [55] Tanner LH. The spreading of silicone oil drops on horizontal surfaces. *J Phys D Appl Phys* 1979;12(9):1473–84. doi:[10.1088/0022-3727/12/9/009](https://doi.org/10.1088/0022-3727/12/9/009).
- [56] Bakshi S, Roisman IV, Tropea C. Investigations on the impact of a drop onto a small spherical target. *Phys Fluids* 2007;19(3). doi:[10.1063/1.2716065](https://doi.org/10.1063/1.2716065).
- [57] Grishaev V, Iorio CS, Dubois F, Amirfazli A. Complex drop impact morphology. *Langmuir* 2015;31(36):9833–44. doi:[10.1021/acs.langmuir.5b02162](https://doi.org/10.1021/acs.langmuir.5b02162).