

## Height functions for applying contact angles to 3D VOF simulations

S. Afkhami and M. Bussmann<sup>\*,†</sup>

*Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Road, Toronto, Canada M5S 3G8*

### SUMMARY

A rigorous methodology is presented for applying a contact angle as a contact line boundary condition within a 3D volume-of-fluid-based flow algorithm, based on the recently developed height function methodology that yields accurate interface normals and curvatures from volume fractions. We demonstrate that the approach yields accurate estimates of curvature and surface tension at a contact line, values that converge with spatial refinement. We then study the efficacy of this approach via examples of both static and dynamic contact line phenomena. Copyright © 2008 John Wiley & Sons, Ltd.

Received 11 July 2008; Revised 24 October 2008; Accepted 26 October 2008

KEY WORDS: volume-of-fluid; VOF; contact angle; contact line; height function; surface tension

### 1. INTRODUCTION

The volume-of-fluid, or VOF, methodology is a widespread approach for representing fluid interfaces and, when coupled to a flow solver, is also used to compute quantities related to surface tension that enter the flow calculation. Until recently, however, the VOF method was not generally deemed appropriate for the study of phenomena in which surface tension is the driving force (e.g. the wetting and dewetting of solid surfaces, as occurs in droplet and spray technologies as well as in various natural phenomena). However, recent improvements to calculating curvature  $\kappa$  and applying the surface tension force appear to resolve this issue [1–5].

We recently applied these developments to the simulation of 2D flows that include contact lines [6], which appear as a single point in a domain; this work followed up on that of Renardy *et al.* [7], who applied older VOF methods to a similar problem. Here, we extend that model to 3D, where a contact line is the curve at which a fluid–fluid interface and a solid boundary meet. In the

---

\*Correspondence to: M. Bussmann, Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Road, Toronto, Canada M5S 3G8.

†E-mail: bussmann@mie.utoronto.ca

remainder of this Introduction, we briefly summarize the recent developments in VOF methodology that have enabled this work, and then present an overview of other 3D approaches for treating contact lines.

The VOF method is an interface capturing technique that tracks fluid volume, and so interfaces are represented implicitly. This representation makes it difficult to calculate accurate interface normals and curvatures and, hence, VOF simulations of surface tension-driven interfacial phenomena are typically characterized by significant so-called 'spurious currents' [8], flow induced solely by the discretization and by a lack of convergence with mesh refinement. An example of such an older method is the model of Bussmann *et al.* [9, 10] for simulating drop impact phenomena, which was a three-dimensionalization of the RIPPLE code of Kothe and Mjolsness [11], the first implementation of the continuum surface force (CSF) method [12] for treating surface tension as a volumetric rather than as a surface force. The contact line boundary condition was imposed by reorienting normals at cell vertices near the contact line, to reflect a specified contact angle.

Unfortunately, this approach, and the associated methods for evaluating interface normals and curvatures based on calculating the gradient of the volume fraction field, has recently been shown to be relatively inaccurate, and to not converge with mesh refinement. Yet such methods continue to be used (e.g. [13–15]) in 3D VOF codes to model similar phenomena.

Recent work on two fronts appears to improve the VOF/CSF methodology dramatically. First, Renardy and Renardy [4] and Francois *et al.* [3] have clearly shown that surface tension (when applied as a volumetric force acting on fluid near an interface, via the CSF method of Brackbill *et al.* [12]) must be applied in a way that is 'consistent' with the calculation of the pressure gradient field, so that for the simple case of a static drop, when the curvature is exactly prescribed, the calculation yields the exact jump in pressure and induces no spurious currents. Second, methods have been developed recently to accurately calculate interface normals and curvatures from volume fractions: (E)LVIRA [16], PROST [4], and so-called height functions (HF) [1, 3, 5] yield normals and curvatures that converge with mesh refinement. Put together, VOF now appears to be a viable methodology for the accurate representation of surface tension-driven phenomena [2].

Unlike in 2D, there are relatively few examples of 3D interfacial flow codes that incorporate algorithms for treating contact lines; the ones we present here are representative of the variety of approaches that are used. Baer *et al.* [17] and Walkley *et al.* [18] are examples of Lagrangian finite element methods, in which a boundary fitted moving mesh is aligned with fluid interfaces. In both cases, the authors imposed local slip at the contact line to allow mesh nodes to slide along a solid surface. Among Eulerian methods, Kang *et al.* [19] employed a lattice Boltzmann method to study the displacement of a droplet in a duct; in this case fluid wettability was varied via a fluid/solid interactive force, that in turn yielded different contact angles. Ding and Spelt [20] used a diffuse-interface method based on the convective Cahn–Hilliard equation to model the motion of droplets on solid surfaces, and used these results to demonstrate that a geometric-based contact line boundary condition was preferable to one based on specifying surface energies. Liu *et al.* [21] presented a sharp interface approach based on the level set method, in which the level set field in the vicinity of the contact line is redistanced in order to impose a specified contact angle (different for advancing and receding contact lines), and a slip condition is imposed on grid points in the vicinity of the contact line.

Finally, in addition to the previously mentioned 3D codes for modeling drop impact [9, 10, 13–15], two other VOF codes need to be mentioned. First, Veldman *et al.* [22] recently presented a 3D VOF code based on the original scheme of Hirt and Nichols [23] for simulating one-phase free-surface flows. Instead of using the CSF model, surface tension is implemented

as a boundary condition for pressure at the free surface, with curvatures calculated using an HF methodology, albeit one based on a small stencil. That methodology is then extended to treat contact lines, in a manner similar to that presented here, although few details of the implementation are presented, as the emphasis of the paper is on the application of the model to fluid sloshing aboard spacecraft. Second, the commercial software FLOW-3D ([www.flow3d.com](http://www.flow3d.com)), well known as a VOF-based interfacial flow code, includes a model for fluid wetting that is very different from others. Instead of prescribing the orientation of the interface at a contact line, FLOW-3D calculates a surface stress from Young's equation (which describes the force balance at a static contact line), and then applies that stress along the solid surface near the contact line. Consequently, the code predicts a dynamic contact angle; comparison with some experimental data suggests that the method yields qualitatively correct results [24, 25].

With that as background, the focus of this paper is the extension and application of recent developments in VOF/CSF methodology to the treatment of contact lines in a 3D VOF model. The HF method is employed for the accurate computation of normals, curvatures, and surface tension forces, both at fluid–fluid interfaces away from solid boundaries and at contact lines, for a wide range of contact angles ( $30^\circ \leq \theta \leq 150^\circ$ ), albeit not for extreme wetting ( $\theta \rightarrow 0^\circ$ ) and non-wetting ( $\theta \rightarrow 180^\circ$ ) conditions. Contact line slip is achieved implicitly, as the advection scheme that we use to advect volume fractions utilizes face-centered velocities, so that the nearest velocity to the contact line is one half cell width above a solid boundary.

The remainder of this paper is organized as follows: in Section 2, we present a brief overview of the numerical methodology, with emphasis on the 3D techniques for modeling surface tension in general, and contact lines in particular; more detail on the numerical methodology is presented in [6]. In Section 3, we present convergence studies and examples of contact line phenomena, to demonstrate the efficacy of the model. In Section 4 we summarize this work.

## 2. NUMERICAL METHODOLOGY

### 2.1. Mathematical model

The Navier–Stokes equations govern an incompressible two-phase flow:

$$U_t + \nabla \cdot (UU) = \frac{1}{\rho} (-\nabla p + \nabla \cdot (\mu(\nabla U + \nabla U^T))) + F_{st} + F_b \quad (1)$$

where  $U$  is velocity,  $p$  pressure,  $\rho$  density,  $\mu$  viscosity,  $F_{st} = \gamma \kappa \delta \hat{n}$  a volumetric surface tension force ( $\gamma$  is the surface tension coefficient,  $\kappa$  the curvature,  $\delta$  the Dirac delta function, and  $\hat{n}$  the unit normal to the interface), and  $F_b$  represents any other body forces acting on the fluid. Each fluid is considered to be incompressible; hence, the continuity equation:

$$\nabla \cdot U = 0 \quad (2)$$

is valid for the whole domain.

For a two fluid system, a function  $f$  ( $=0$  in a fluid 1, and 1 in fluid 2) represents the interface; the advection equation for  $f$  is:

$$\partial_t f + U \cdot \nabla f = 0 \quad (3)$$

Density and viscosity may vary from phase to phase, but are assumed constant in a particular phase. These are evaluated via volume-weighted formulae:

$$\rho = \rho_1 + (\rho_2 - \rho_1)f \quad (4)$$

$$\mu = \mu_1 + (\mu_2 - \mu_1)f \quad (5)$$

where the subscripts 1 and 2 refer to the two fluids.

## 2.2. Numerical model

The basis for the numerical model is an early version of Gerris [26, 27], at the time a variable density incompressible Euler solver based on the work of Almgren *et al.* [28], into which we incorporated algorithms for evaluating the surface tension and variable viscosity terms. An adaptively refined approximate projection method based on a variable density fractional-step scheme is utilized to discretize Equations (1)–(5) in space and time. All variables are collocated at cell centers; normal velocities are also calculated at face centers. Velocity, viscosity, and density are calculated at times  $t^n$ , pressure at  $t^{n+1/2}$ .

In the first step of the fractional-step scheme, we calculate an intermediate velocity field  $U^{**}$  as:

$$\frac{U^{**} - U^n}{\Delta t} = -[\nabla \cdot (UU)]^{n+1/2} + \frac{1}{\rho^{n+1}} \nabla \cdot (\mu^n (\nabla U + \nabla U^T)^n) + \frac{1}{\rho^{n+1}} F_b^{n+1} \quad (6)$$

In the second step, we project  $U^{**}$  onto an approximately divergence-free velocity field, and solve the following Poisson problem:

$$L \left( \frac{1}{\rho^{n+1}} p \right) = \nabla \cdot \left( \frac{1}{\Delta t} \tilde{U}^{**} + \frac{1}{\rho^{n+1}} \tilde{F}_{st}^{n+1} \right) \quad (7)$$

for pressure and for an update to the velocity field. Here,  $L$  is an approximate projection operator,  $\tilde{U}^{**}$  is the face-centered velocity approximated from the cell-centered velocity  $U^{**}$ , and  $\tilde{F}_{st}$  is the face-centered surface tension force (to be discussed).  $U^{n+1}$  is then obtained from:

$$U^{n+1} = U^{**} - \frac{\Delta t}{\rho^{n+1}} (\nabla p - F_{st}^{n+1}) \quad (8)$$

Discretization of the viscous term  $\nabla \cdot (\mu^n (\nabla U + \nabla U^T)^n)$  and the nonlinear advection term  $[\nabla \cdot (UU)]^{n+1/2}$  requires quantities available at time  $t^n$ ; a second-order Godunov-type scheme is used to discretize the advection terms. A time- and face-centered velocity field is obtained in the process of evaluating  $[\nabla \cdot (UU)]^{n+1/2}$ . This velocity field is exactly projected onto a divergence-free velocity field, and so we use this face velocity field to calculate a conservative update to  $f$  via Equation (3). Density  $\rho^{n+1}$  and viscosity  $\mu^{n+1}$  are then evaluated from  $f^{n+1}$ .

A balanced-force implementation of the CSF model [12] is used to discretize  $F_{st}$ , in which  $\delta \hat{n}$  is approximated as  $\nabla f$ :

$$F_{st} = \gamma \kappa \nabla f \quad (9)$$

To enforce consistency, the pressure gradient and surface tension forces are considered together (in Equations (7) and (8)) and are discretized in the same way. As we discretize  $\nabla p$  at cell faces (Equation (7)), the surface tension force  $\tilde{F}_{st}$  is evaluated there too. The cell-centered pressure gradient and surface tension forces in Equation (8) are then calculated by simple averaging from face values, again to ensure that the discretization errors associated with  $F_{st}$  and  $\nabla p$  cancel.

The equations are solved on an adaptive Cartesian mesh in which cells that satisfy a given criterion are locally refined or coarsened. For this work, all cells that contained an interface, as well as direct and indirect neighboring cells, were uniformly resolved. This greatly simplified the discretization of discontinuous quantities across the interface. Away from the interface, the mesh was free to adapt.

### 2.3. The VOF method

In the VOF method, the discrete form of the function  $f$  represents the volume fraction of a cell filled with, in this case, fluid 2. Away from the interface,  $f=0$  or 1; ‘interface cells’ are those that are partially full ( $0 < f < 1$ ). Equation (3) is usually solved in two steps: a geometric interface reconstruction, followed by advection of the reconstructed fluid near the interface. In this work, the reconstruction is a ‘piecewise linear interface calculation’ (PLIC), followed by a Lagrangian advection step [29].

In the PLIC technique, given a volume fraction  $f$  and an approximate normal vector  $\mathbf{n}$ , a linear (in 2D) or planar (in 3D) interface is constructed within each interface cell, which corresponds exactly to  $f$  and  $\mathbf{n}$ . Here, we implemented the HF methodology to calculate  $\mathbf{n}$ , and also used it as the basis for the contact line implementation presented here.

### 2.4. The HF method

In what follows, we present the HF methodology for the uniform grid spacing  $\Delta$  that we imposed near interface cells; the extension to a non-uniform grid is relatively straightforward.

The HF methodology [1, 3, 5] is a VOF-based technique for calculating interface normals and curvatures. About each interface cell, fluid ‘heights’ are calculated by summing fluid volume in the direction most normal to the interface (estimated here as  $\nabla f$  using the stencil of [30]). In 2D (a methodology that we require for our 3D contact angle implementation), we construct a  $7 \times 3$  stencil around an interface cell. For example, for the cell  $(i, j)$  illustrated in Figure 1,  $|n_y| > |n_x|$ , and so the heights are evaluated by summing volume fractions vertically:

$$h_i = \sum_{k=j-3}^{k=j+3} f_{i,k} \Delta \quad (10)$$

with similar expressions for  $h_{i-1}$  and  $h_{i+1}$ . The heights  $h$  are then used to compute an interface normal  $\mathbf{n}$  and curvature  $\kappa$ :

$$\mathbf{n} = (h_x, -1) \quad (11)$$

$$\kappa = \frac{h_{xx}}{(1+h_x^2)^{3/2}} \quad (12)$$

where  $h_x$  and  $h_{xx}$  are discretized using second-order central differences.

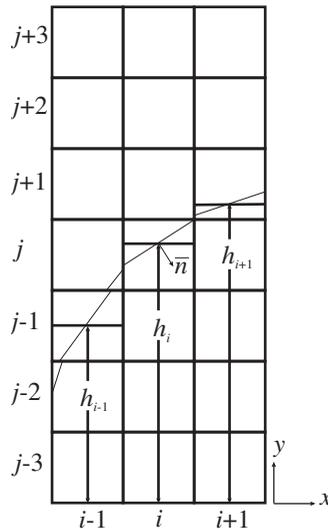


Figure 1. The  $7 \times 3$  stencil used to construct 2D HFs.

In 3D, a  $7 \times 3 \times 3$  stencil is constructed about an interface cell. For example, for an interface oriented most normal to the  $z$  direction,  $\mathbf{n}$  and  $\kappa$  are computed as:

$$\mathbf{n} = (h_x, h_y, -1) \quad (13)$$

$$\kappa = \frac{h_{xx} + h_{yy} + h_{xx}h_y^2 + h_{yy}h_x^2 - 2h_{xy}h_xh_y}{(1 + h_x^2 + h_y^2)^{3/2}} \quad (14)$$

Cummins *et al.* [1] have shown that curvatures calculated in this way are second-order accurate.

As mentioned in Section 2.2, surface tension forces are calculated at cell faces, and so we require face-centered curvatures. When a face neighbors two interface cells, curvature is evaluated as the average of two cell-centered values; when a face neighbors just one interface cell, we simply assign that cell-centered curvature to the face.

### 2.5. The contact angle boundary condition

The value of the contact angle affects the overall flow calculation in two ways: it defines the orientation of the VOF reconstruction in cells that contain the contact line and it influences the calculation of  $F_{st}$  by affecting the curvature calculated in cells at and near the contact line. In what follows, we detail algorithms for identifying such cells, and then for calculating normals and curvatures in these cells. For simplicity of presentation, we arbitrarily define the solid surface as  $x$  (in 2D) and  $x$ - $y$  (in 3D).

**2.5.1. Identifying ‘contact line’ and ‘adjacent’ cells.** We begin with a brief description of the methodology in 2D, illustrated in Figure 2. Along a solid boundary, we define a ‘contact line cell’ as an interface cell that has at least one neighbor along a solid boundary that is completely empty (for  $\theta \leq 90^\circ$ ) or full (for  $\theta > 90^\circ$ ). We then define ‘adjacent cells’ as interface cells along the solid

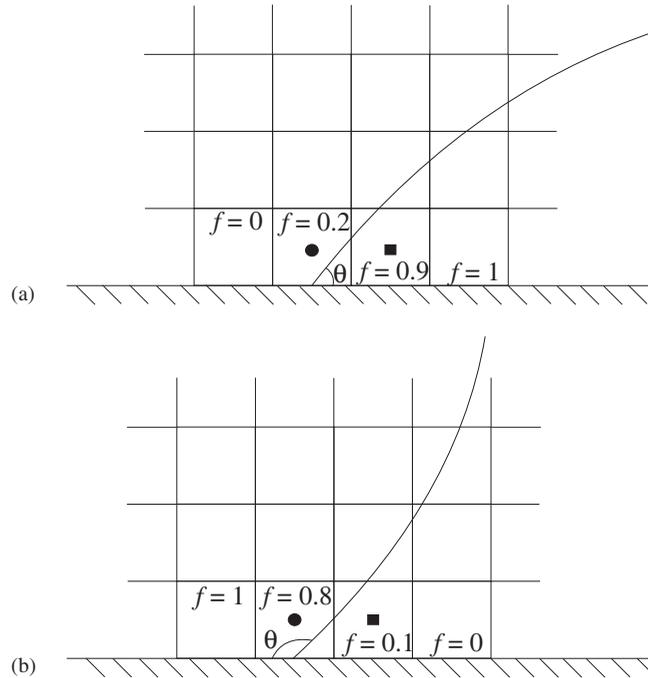


Figure 2. A contact line cell (●) and an adjacent cell (■): (a)  $\theta < 90^\circ$  and (b)  $\theta > 90^\circ$ .

boundary that neighbor contact line cells. Note that this implies that there may be interface cells along a solid boundary that are neither contact line nor adjacent cells, as might occur when a thin film (of thickness less than  $\Delta$ ) wets a solid surface. Such cells are simply treated as any other interface cell away from a contact line.

In 3D, as illustrated in Figure 3 for  $\theta < 90^\circ$ , contact line cells are identified in a similar way, by checking all neighboring cells (including corner neighbors) along a solid boundary. Adjacent cells are then those that neighbor a contact line cell. The pseudocode descriptions of the algorithms for identifying contact line and adjacent cells are in Appendix A.

Having identified contact line and adjacent cells, we now describe methodologies for calculating normals and curvatures in these cells, and we conclude by summarizing these algorithms via pseudocode.

**2.5.2. Normals and curvatures in ‘contact line cells’.** In 2D, a contact line normal  $\mathbf{n}_{cl}$  is uniquely determined by the contact angle. In 3D, however, a family of vectors makes the angle  $\theta$  with the normal to a solid boundary  $\mathbf{n}_s$ , as shown in Figure 4. In order to determine the normal  $\mathbf{n}_{cl}$  to the interface at the contact line, we first use 2D HFs to determine  $\mathbf{n}_{cl-x,y}$ , the projection of  $\mathbf{n}_{cl}$  in the  $x$ - $y$  plane.

Consider cells in the  $x$ - $y$  plane just above a solid boundary, illustrated in Figure 5. For cell  $(i, j, 1)$ , for example,  $\mathbf{n}_{cl-x,y} = (h_y, -1)$ , where in this case, the HF is oriented in the  $x$  direction,

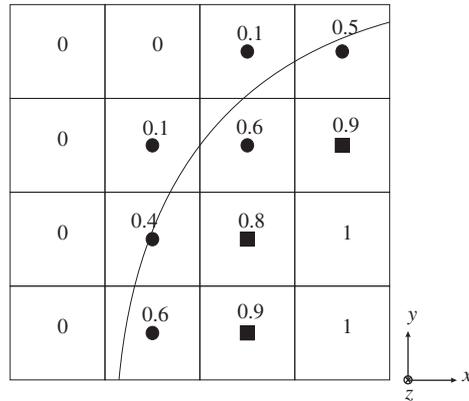


Figure 3. Contact line (●) and adjacent (■) cells along a 3D contact line. The volume fractions correspond to a value of  $\theta < 90^\circ$ .

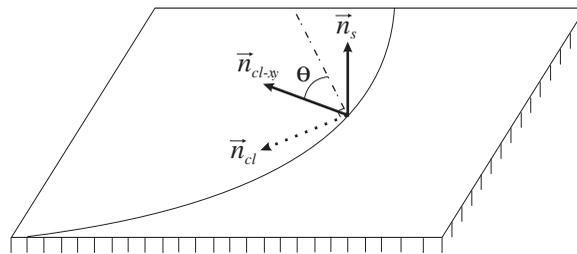


Figure 4.  $\mathbf{n}_{cl-xy}$  is the projection of the contact line normal  $\mathbf{n}_{cl}$  onto the  $x$ - $y$  plane, defined by an outward normal  $\mathbf{n}_s$ .

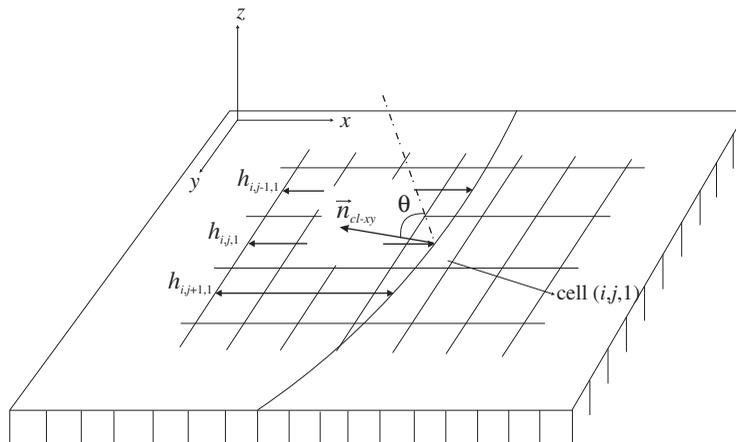


Figure 5. Height functions next to the solid surface are used to calculate  $\mathbf{n}_{cl-xy}$ , the projection of the contact line normal  $\mathbf{n}_{cl}$  onto the  $x$ - $y$  plane.

and  $h_y$  is evaluated as:

$$h_y = \frac{h_{i,j+1,1} - h_{i,j-1,1}}{2\Delta} \tag{15}$$

The  $z$ -component of  $\mathbf{n}_{cl}$  is then defined to reflect the contact angle, which leads to:

$$\mathbf{n}_{cl} = (h_y, -1, |\mathbf{n}_{cl-xy}| / \tan \theta) \tag{16}$$

To compute the curvature in a contact line cell, we require a  $7 \times 3 \times 3$  HF stencil, the orientation of which depends on the contact angle. Strictly speaking, for  $45^\circ < \theta < 135^\circ$ , one would use a ‘horizontal’ (oriented in the  $x$  or  $y$  direction) HF; for  $\theta < 45^\circ$  and  $\theta > 135^\circ$ , which correspond to conditions of near wetting and non-wetting, respectively, one would use a ‘vertical’ (oriented in the  $z$  direction) HF. Vertical HF’s, however, are difficult to construct accurately, and so we limit ourselves to horizontal HF’s along a solid boundary, although we can apply the methodology somewhat beyond  $45^\circ < \theta < 135^\circ$ . However, this methodology is not appropriate for extreme wetting and non-wetting values of  $\theta$ .

Returning briefly to 2D, consider Figure 6, where the fluid heights  $h_{j=1}$  and  $h_{j=2}$  are computed in the horizontal direction. To calculate the curvature in the contact line cell, the value of the fluid height beyond the flow domain  $h_{j=0}$  is required; this is the value that lies in the so-called ‘ghost cells’.  $h_{j=0}$  is extrapolated linearly from  $h_{j=1}$  to reflect the known contact angle:

$$h_{j=0} = h_{j=1} + \Delta / \tan \theta \tag{17}$$

In 3D, the  $7 \times 3 \times 3$  HF stencil is built around the stencil for  $\mathbf{n}_{cl-xy}$ , as illustrated in Figure 5. Six of the nine fluid heights are known; the three others in the ghost cells ( $h_{i,j-1,0}$ ,  $h_{i,j,0}$ , and  $h_{i,j+1,0}$ ) must be calculated. Unlike in 2D, these three heights are functions not only of the contact angle, but also of the orientation of the HF stencil relative to  $\mathbf{n}_{cl-xy}$ , as illustrated in Figure 7, that looks down onto the  $x$ - $y$  plane.

Just like in 2D, Equation (17) defines how  $\theta$  affects the change in fluid height from just above a surface to the ghost cells below, and would define the values of the three ghost cell fluid heights exactly if  $\mathbf{n}_{cl-xy}$  were aligned with the HF stencil. But referring to Figure 7, when the HF stencil makes an angle  $\alpha$  with  $\mathbf{n}_{cl-xy}$ , that angle must also be taken into account, and leads to the following

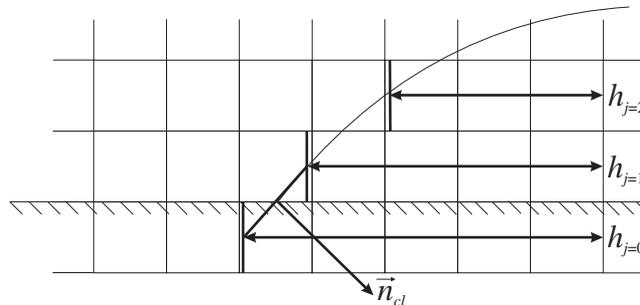


Figure 6. The height  $h_{j=0}$  in the ghost cells is determined so that the interface orientation at the contact line corresponds to  $\mathbf{n}_{cl}$ .

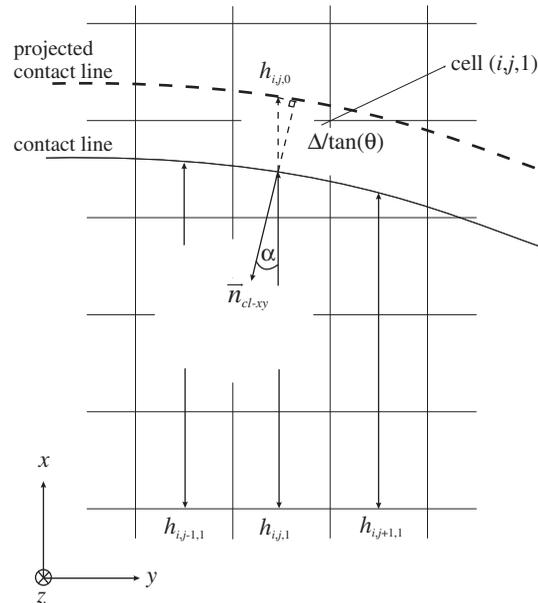


Figure 7. The ghost cell heights  $h_{i,j-1,0}$ ,  $h_{i,j,0}$  and  $h_{i,j+1,0}$  are determined so that the interface orientation at the contact line corresponds to the contact line normal.

expression for the ghost cell fluid height  $h_{i,j,0}$ :

$$h_{i,j,0} = h_{i,j,1} + \Delta / (\tan \theta \cdot \cos \alpha) \quad (18)$$

with corresponding expressions for  $h_{i,j-1,0}$  and  $h_{i,j+1,0}$ .

**2.5.3. Normals and curvatures in 'adjacent cells'.** The methodology for calculating normals and curvatures in adjacent cells (those next to contact line cells) also requires special treatment. Unlike HFs for contact line cells, which we limit to a horizontal orientation (i.e. oriented in the  $x$  or  $y$  directions), we allow HFs for adjacent cells to be oriented in any way, including vertically.

As mentioned in Section 2.4, the HF orientation is chosen by calculating a normal via the stencil of [30]. In order to calculate that normal in adjacent cells, and in order to construct vertical HFs if necessary, volume fractions must be assigned to the neighboring ghost cells, by extending reconstructed interfaces in contact line cells into the underlying ghost cells, as illustrated in 2D in Figure 8(a). Other ghost cells beneath fluid 2 are then assigned as  $f = 1$  (if  $\theta \leq 90^\circ$ ) or ghost cells beneath fluid 1 are assigned as  $f = 0$  (if  $\theta > 90^\circ$ ). Ghost cells on the other side of the contact line do not enter into the calculation of normals and curvatures.

Horizontal HFs for adjacent cells are then constructed as for contact line cells: three fluid heights are calculated in the ghost cells, as described in Section 2.5.2. To construct a vertical HF for an adjacent cell, the extrapolated ghost cell values are incorporated into the calculation, as illustrated in Figure 8(b).

The algorithms for applying a contact angle as a boundary condition are summarized in Appendix A.

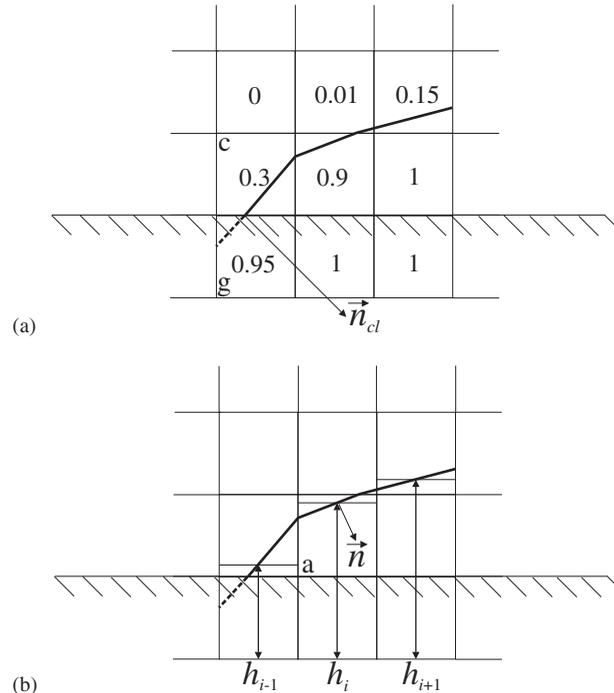


Figure 8. (a) The volume fraction in the ghost cell  $\mathbf{g}$  is determined by extending the reconstructed interface from the contact line cell  $\mathbf{c}$ . (b) The heights  $h_{i-1}$ ,  $h_i$ , and  $h_{i+1}$  are required to determine the normal and the curvature in the adjacent cell  $\mathbf{a}$ .

### 3. RESULTS

We now present various results to illustrate the efficacy of our methodology. We examine errors associated with contact line curvatures, and the amplitude of spurious currents for drops initialized at equilibrium. We present a convergence study comparing the results of contact line-driven computations at different grid resolutions. And we end with a sample result of a more complex contact line-driven flow.

#### 3.1. Convergence study

Consider 3D drops on an  $1 \times 1$  surface, initialized by truncating a sphere of radius  $R=0.25$ . Figure 9 illustrates three such drops, for  $\theta=60, 90$ , and  $120^\circ$ . Corresponding to a test case analyzed by Renardy and Renardy [4], the drops and surrounding fluid have equal viscosity 1 and density 4, and the surface tension  $\gamma=0.357$ . The relevant non-dimensional quantity is the Ohnesorge number  $Oh = \mu / \sqrt{\gamma \rho D} \approx 1.15$ , which is the ratio of viscous to surface tension forces. The initial velocity is zero, and in the absence of gravity, the drops are at equilibrium, and should remain so. The exact solution to the flow equations is a uniform zero velocity field and a pressure jump across the interface that is exactly balanced by the surface tension force  $\gamma \kappa$ .

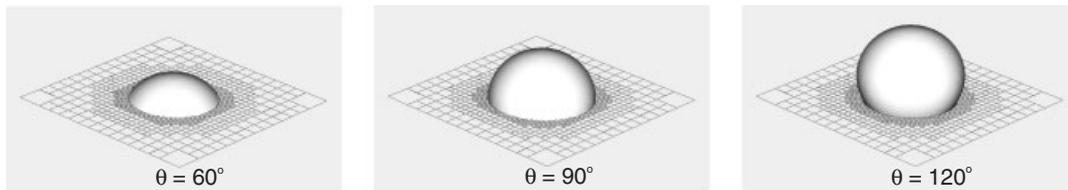


Figure 9. 3D drops at equilibrium,  $\theta=60, 90,$  and  $120^\circ$ .

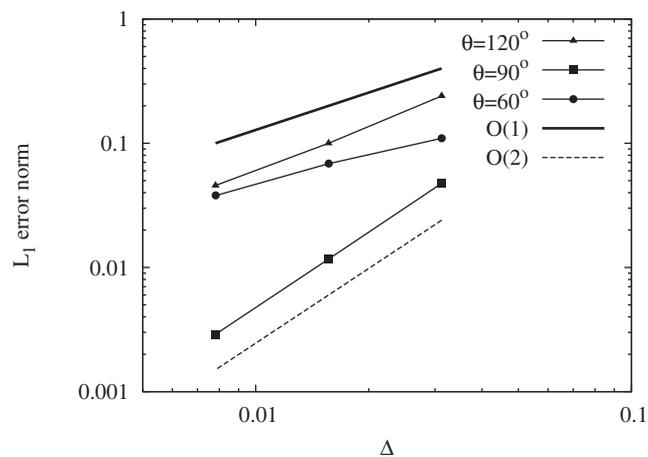


Figure 10.  $L_1$  norm of errors in contact line curvature, for drops at equilibrium ( $\theta=60, 90,$  and  $120^\circ$ ).

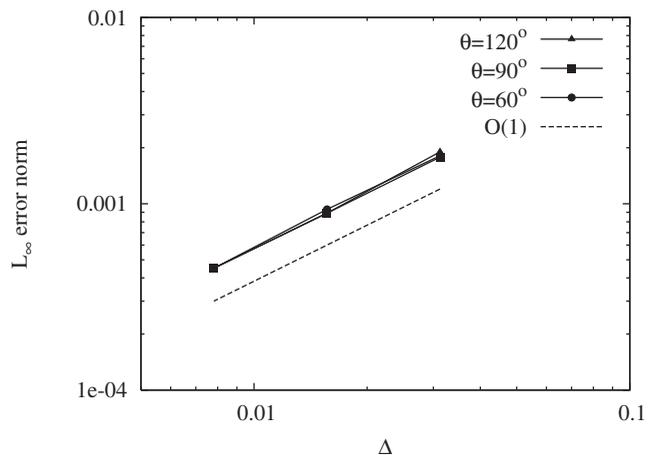


Figure 11.  $L_\infty$  norm of spurious currents versus  $\Delta$ , for drops at equilibrium ( $\theta=60, 90,$  and  $120^\circ$ ):  $\Delta t = 10^{-5}, t = 0.01$ .

Table I.  $L_1$  and  $L_\infty$  norms of errors in volume fractions, after an initially hemispherical drop of radius  $R=0.2$  is drawn to a new equilibrium configuration by changing  $\theta$  to  $60^\circ$ .

$\Delta$	$L_1$	$L_\infty$
$\frac{1}{32}$	$7.13 \times 10^{-4}$	$3.08 \times 10^{-1}$
$\frac{1}{64}$	$1.62 \times 10^{-4}$	$1.72 \times 10^{-1}$
$\frac{1}{128}$	$4.59 \times 10^{-5}$	$1.07 \times 10^{-1}$

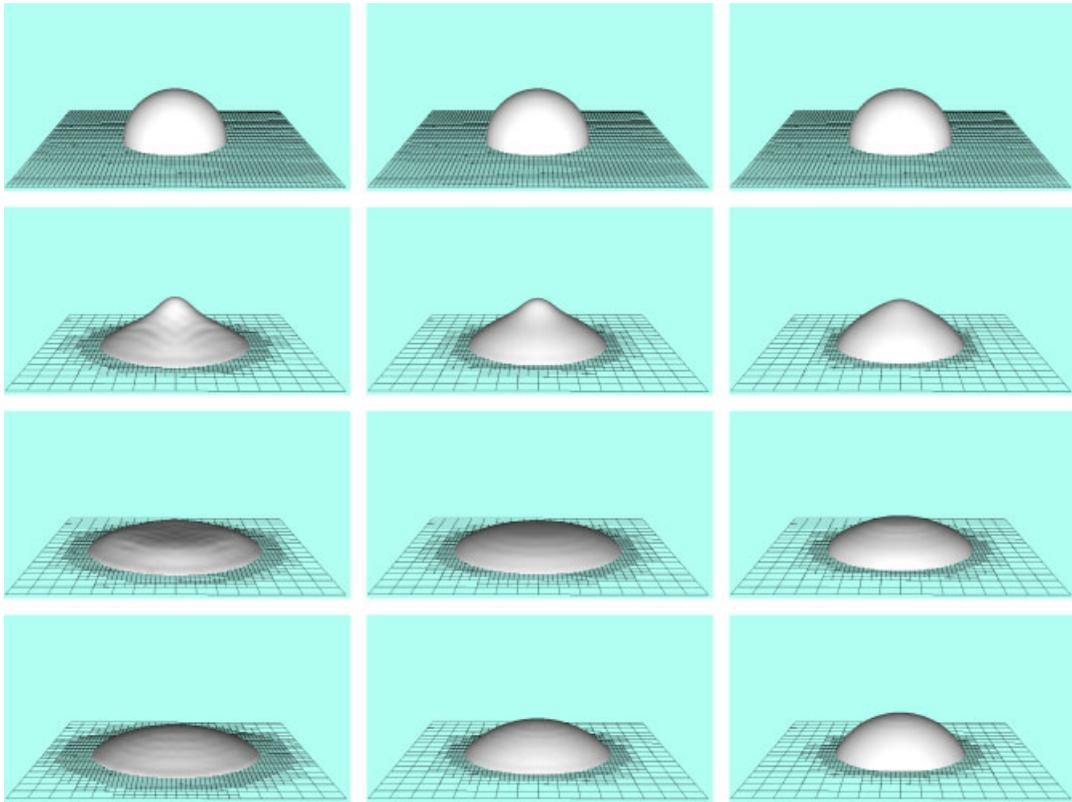


Figure 12. Drops initially at equilibrium ( $\theta=90^\circ$ ) are exposed to a sudden change in  $\theta=30, 45,$  and  $60^\circ$  (left to right) and viewed at  $\tau=0, 5, 10, 50$  (top to bottom).

For these equilibrium configurations, the exact curvature  $\kappa_{\text{exact}}=2/R$ . Figure 10 presents the  $L_1$  norm of errors in curvature calculated only in contact line cells. The curvatures for  $\theta=90^\circ$  converge at a second-order rate and the errors are much smaller than those for the other values of  $\theta$ , because the  $90^\circ$  contact angle is equivalent to a symmetry boundary condition ( $h_{i,j,0}=h_{i,j,1}$  in Equation (18)), so that we would expect the same order of convergence as for curvatures

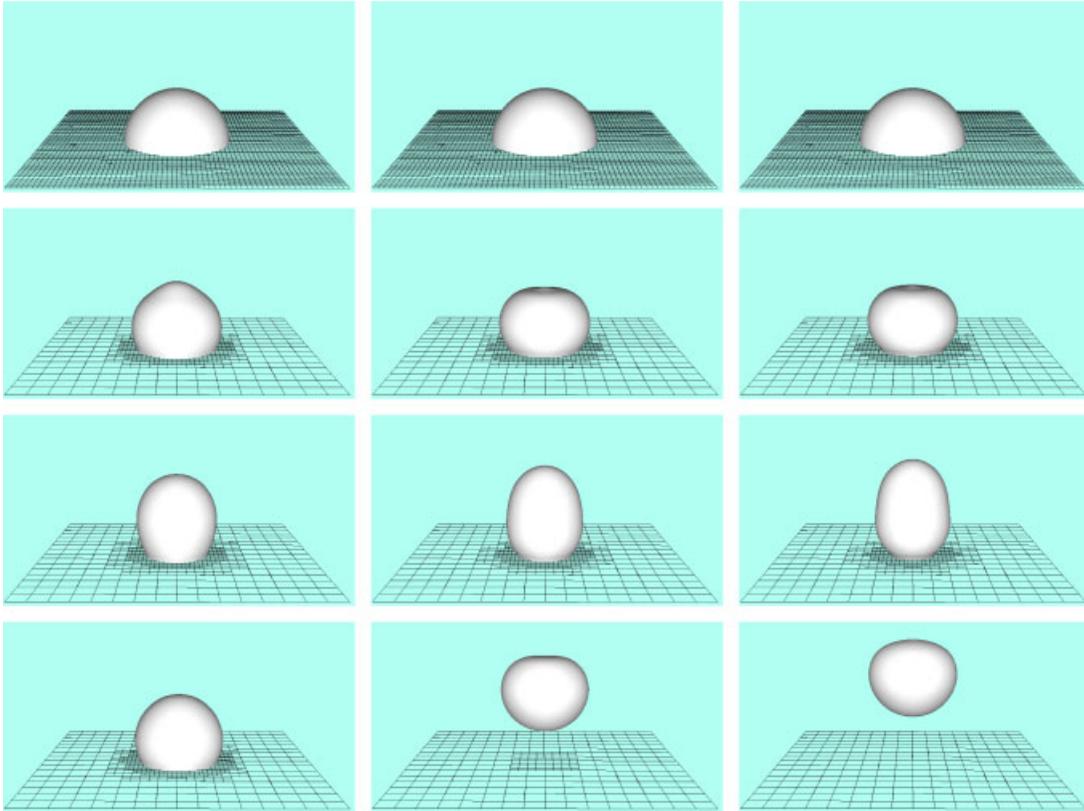


Figure 13. Similar to Figure 12:  $\theta = 120^\circ$  (left) at  $\tau = 0, 5, 10, 50$  (top to bottom) and  $\theta = 135$  and  $150^\circ$  (center and right) at  $\tau = 0, 4, 8, 20$ .

evaluated away from solid/mesh boundaries. On the other hand, for  $\theta = 60$  and  $120^\circ$ , contact line curvatures are first-order accurate, because  $h_{i,j,0}$  is no longer exact, but rather is obtained by linear extrapolation.

Figure 11 presents the  $L_\infty$  norm of spurious currents at  $t = 0.01 (\Delta t = 10^{-5})$ ; these are the velocities induced in the domain by inexact values of curvature (if curvatures were exact, the computed solution would be a zero velocity field). The results show that the spurious currents diminish with mesh refinement when normals and curvatures are calculated via HFs. These results compare well with those of the PROST method of Renardy and Renardy [4]; in their case they studied a 3D static drop in equilibrium, without a contact line, and demonstrated the same behavior of spurious currents with mesh refinement.

### 3.2. Contact line-driven drop spreading

An example of a surface tension-driven flow is a drop on a solid surface initially at equilibrium, when suddenly a different contact angle is imposed; the drop fluid should accelerate toward a steady state configuration defined by the new value of  $\theta$ . Here we present a series of tests for

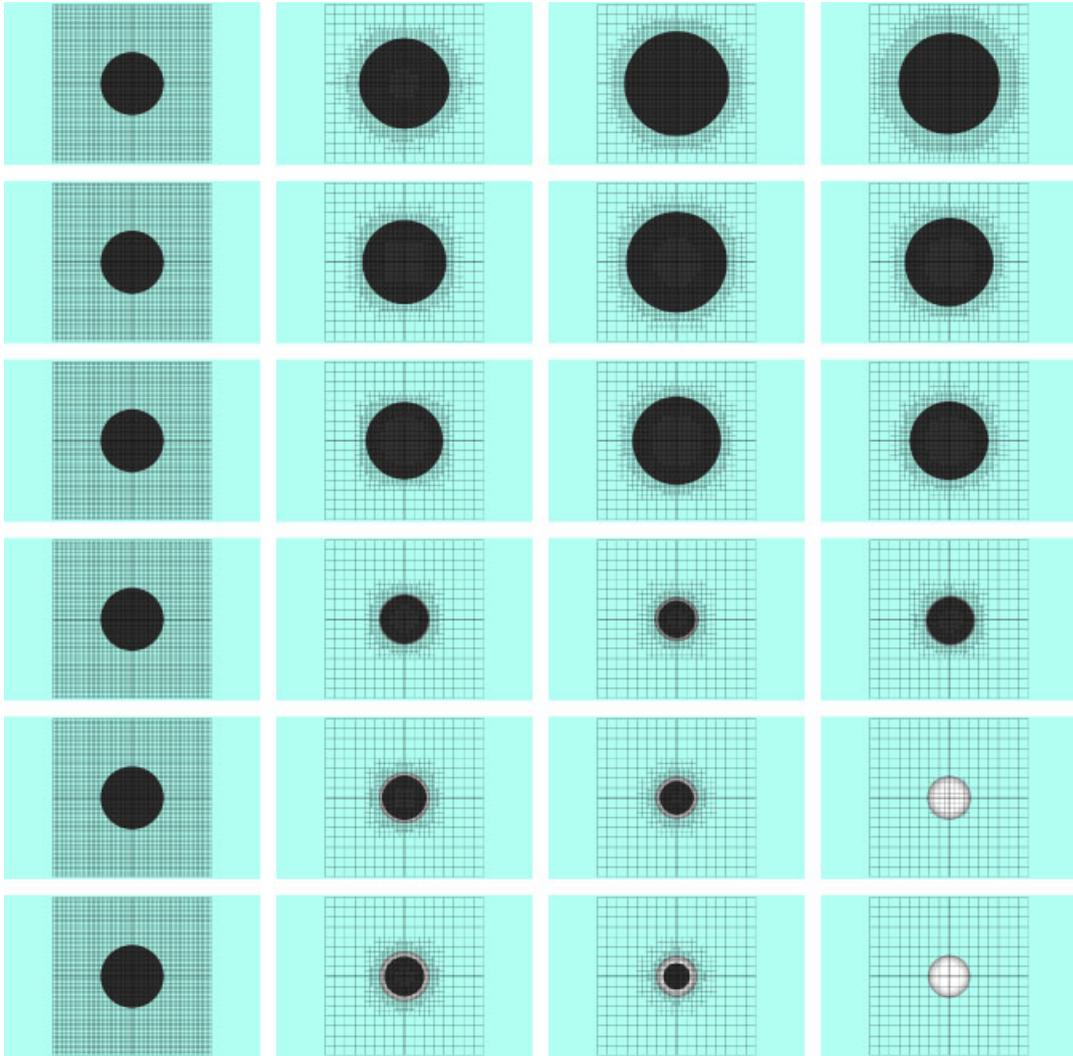


Figure 14. Views from beneath the domain of drops driven by a sudden change in  $\theta=30, 45, 60, 120, 135, 150^\circ$  (top to bottom) and times (left to right) correspond to those of Figures 12 and 13.

$30^\circ \leq \theta \leq 150^\circ$ . An initially hemispherical drop ( $\theta=90^\circ$ ) of radius  $R=0.2$  is placed at the bottom of an  $1 \times 1 \times 1$  domain. The density ratio is  $\rho_\ell/\rho_g=800$ , the viscosity ratio is  $\mu_\ell/\mu_g=100$ , surface tension is  $\gamma=0.1$ , and  $Oh \approx 1.33 \times 10^{-2}$ . A no-slip condition is specified at the bottom of the domain and an open boundary condition on all other walls.

To begin, we present the spatial convergence behavior for  $\theta=60^\circ$ . The drop was deemed to reach equilibrium when the magnitude of the maximum velocity reached a steady state; the velocities are then the spurious currents induced by the numerical solution. We evaluated the computed

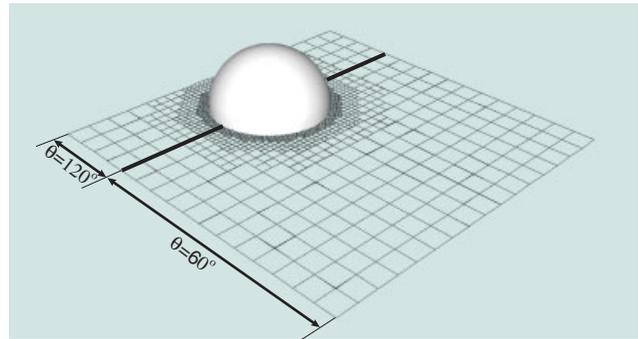


Figure 15. Initial configuration of a drop on a heterogeneous surface:  $\theta = 120^\circ$  imposed beneath the left half of the drop and  $\theta = 60^\circ$  imposed beneath the right half.

equilibrium configurations by comparing the steady state volume fraction field with the expected volume fractions (corresponding to a spherical cap of the same volume, with  $\theta = 60^\circ$ ), computed to machine precision:

$$L_1 = \sum_{i=1}^N |f_{i,\text{exact}} - f_{i,\text{approx}}| \Delta_i^3 \quad (19)$$

$$L_\infty = \max |f_{i,\text{exact}} - f_i| \quad (20)$$

$N$  is the number of cells in the domain and  $\Delta_i$  represents the size of a cell  $i$  in the adaptive mesh. Table I presents  $L_1$  and  $L_\infty$  norms of errors in volume fractions, for  $\Delta = \frac{1}{32}$ ,  $\frac{1}{64}$ , and  $\frac{1}{128}$ . The results show that the mean and maximum errors decrease with mesh refinement at second- and first-order rates, respectively. More important, however, is that the maximum errors are well below one, which implies that the steady state shapes are well within a single mesh cell of the exact solutions; in fact, at  $\Delta = \frac{1}{128}$ , the maximum deviation of the computed steady state profile is approximately  $\Delta/10$  from exact.

We now present results when we suddenly impose  $\theta = 30, 45, 60, 120, 135$ , and  $150^\circ$  on an initially hemispherical drop at  $t=0$ ; these were computed on an adaptive mesh with the maximum grid resolution  $\Delta = \frac{1}{64}$ , and are presented in Figures 12 and 13. For  $\theta \leq 120^\circ$ , the drops reach a steady state configuration, as defined by a maximum velocity that is on the order of the magnitude of the spurious currents. On the other hand, for  $\theta > 120^\circ$ , the drops jump off the surface due to the strong surface tension forces at the contact line, and the absence of gravity.

A difficult test of a 3D code with contact lines is its ability to maintain symmetry during an axisymmetric calculation. An improper implementation of the surface tension force at the contact line, and inaccurate curvatures, can lead to results that appear asymmetric, especially when considering a contact line-driven flow. Figure 14 illustrates the evolution of contact lines for  $\theta = 30, 45, 60, 120, 135$ , and  $150^\circ$ . As shown, the results remain axisymmetric throughout.

### 3.3. An 'electro-wetting' example

Finally, we present the results of a simulation in which the contact line drives asymmetric motion.

Electro-wetting [31] is a technique that utilizes an applied electric field to change the wetting behavior of a fluid on a solid surface. By applying a spatially varying field, the wetting behavior

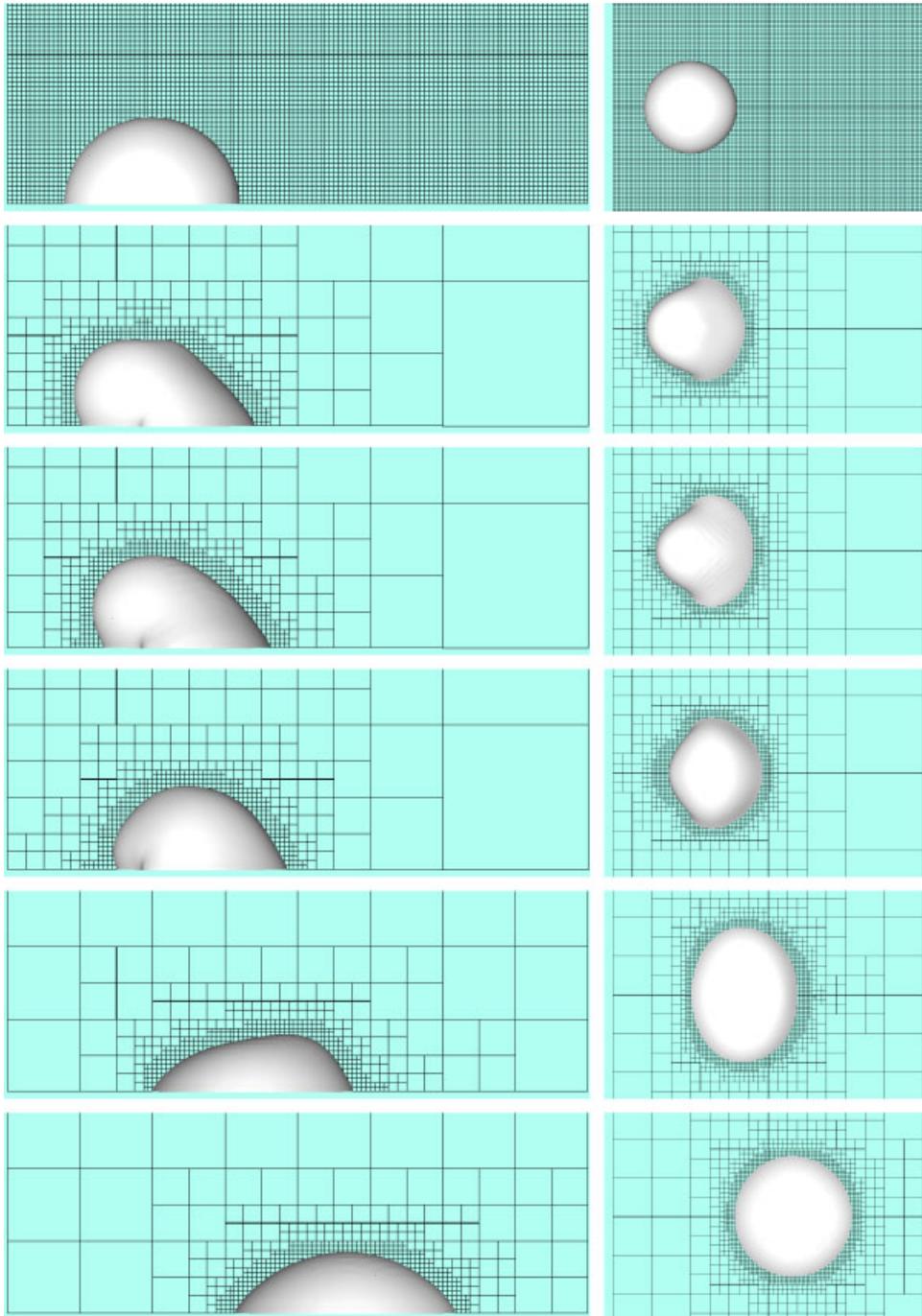


Figure 16. Side (left) and top views of a droplet on a heterogeneous surface, at  $t = 0$ , 2, 5, 7, 12, and 30 ms (top to bottom).

will change spatially, and cause a drop of fluid to move on the surface. There have been recent attempts to simulate such forced movement of a droplet on a surface [17, 18], a problem also of interest in microfluidic processing.

Here, we present the results of a simulation that illustrates such a behavior, by prescribing a spatially varying contact angle. A hemispherical droplet of radius  $R=200\ \mu\text{m}$  is initially placed at the bottom of an  $1 \times 1 \times 1\ \text{mm}$  domain; the fluid properties are those of water and air. The droplet is at rest ( $\theta=90^\circ$ ) when suddenly  $\theta=60$  and  $120^\circ$  are imposed beneath halves of the droplet, as illustrated in Figure 15. The results were computed on an adaptive mesh, with a maximum grid resolution equal to  $\frac{1}{64}$ ; a no-slip boundary condition was specified at the bottom of the domain and an open boundary condition elsewhere. A sequence of results are shown in Figure 16, and illustrate how the droplet is quickly drawn from the non-wetting to the wetting surface, which leads to transient contact line profiles that are far from axisymmetric. Yet by 30 ms, the droplet has reached a steady state shape corresponding to a spherical cap with  $\theta=60^\circ$ .

#### 4. SUMMARY

Building on recent improvements to the volume-of-fluid (VOF) method, we have presented a height function-based methodology for the accurate representation of surface tension forces at contact lines. Although VOF methods have been used before to model phenomena that include contact lines, the implementation details have not previously been presented. Here we presented a detailed and rigorous implementation, that includes algorithms for identifying so-called ‘contact line’ and ‘adjacent’ cells, as well as for calculating normals and curvatures in these cells. The model was shown to accurately predict steady state configurations defined by the imposed contact angles, from initial conditions far from equilibrium.

#### APPENDIX A

---

##### Algorithm 1: Identify a contact line cell

---

```

for all cell along the solid boundary do
  tag contact line cells:
  if ( $0 < f(\textit{cell}) < 1$ ) then
    for all neighboring cellN of cell do
      if ( $\theta \leq 90$  and  $f(\textit{cell}_N) == 0$ ) then
        cell is contact line
        break
      end if
      if ( $\theta > 90$  and  $f(\textit{cell}_N) == 1$ ) then
        cell is contact line
        break
      end if
    end for
  end if
end for

```

---

---

**Algorithm 2** : Identify an adjacent cell

---

```

for all  $cell$  along the solid boundary that are not contact line do
  tag adjacent cells:
  if  $(0 < f(cell) < 1)$  then
    for all neighboring  $cell_N$  of  $cell$  do
      if  $((cell_N)$  is contact line) then
         $cell$  is adjacent
        break
      end if
    end for
  end if
end for

```

---



---

**Algorithm 3** : Calculate ghost cell volume fractions

---

```

for all ghost  $cell_g$  do
  compute  $f$ :
  if  $(cell_g)$  is beneath a contact line  $cell$  then
     $f(cell_g) = \text{reconstruct\_interface}(cell)$ 
  else if  $(cell_g)$  is beneath fluid 2 and  $\theta \leq 90$  then
     $f(cell_g) = 1$ 
  else if  $(cell_g)$  is beneath fluid 1 and  $\theta > 90$  then
     $f(cell_g) = 0$ 
  else
    /*  $cell_g$  is on the other side of the contact line */
  end if
end for

```

---



---

**Algorithm 4** : Calculating normal  $\mathbf{n}$  and curvature  $\kappa$  in contact line and adjacent cells

---

```

for all interface  $cell$  do
  if  $(cell)$  is contact line then
    calculate  $n_x, n_y$  via 2D HF, and determine  $n_z$  to reflect  $\theta$ 
    extrapolate HFs horizontally
    calculate  $\kappa$ 
  else if  $(cell)$  is adjacent then
    if (horizontal HF stencil) then
      extrapolate HFs
      calculate  $\mathbf{n}$  and  $\kappa$ 
    else
      /* vertical HF stencil */
      construct vertical HFs
      calculate  $\mathbf{n}$  and  $\kappa$  using HF
    end if
  else
    /*  $cell$  is neither contact line nor adjacent */
  end if
end for

```

---

## REFERENCES

1. Cummins SJ, Francois MM, Kothe DB. Estimating curvature from volume fractions. *Computers and Structures* 2005; **83**:425–434.
2. Gerlach D, Tomar G, Biswas G, Durst F. Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows. *International Journal of Heat and Mass Transfer* 2006; **49**:740–754.
3. 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. *Journal of Computational Physics* 2006; **213**:141–173.
4. Renardy Y, Renardy M. A parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of Computational Physics* 2002; **183**:400–421.
5. Sussman M. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics* 2003; **187**:110–136.
6. Afkhani S, Bussmann M. Height functions for applying contact angles to 2D VOF simulations. *International Journal for Numerical Methods in Fluids* 2008; **57**:453–472.
7. Renardy M, Renardy Y, Li J. Numerical simulation of moving contact line problems using a volume-of-fluid method. *Journal of Computational Physics* 2001; **171**:243–263.
8. Scardovelli R, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics* 1999; **31**:567–603.
9. Bussmann M, Mostaghimi J, Chandra S. On a three-dimensional volume tracking model of droplet impact. *Physics of Fluids* 1999; **11**:1406–1417.
10. Bussmann M, Chandra S, Mostaghimi J. Modeling the splash of a droplet impacting a solid surface. *Physics of Fluids* 2000; **12**:3121–3132.
11. Kothe DB, Mjolsness RC. RIPPLE: a new model for incompressible flows with surface tension. *AIAA Journal* 1992; **30**:2694–2700.
12. Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. *Journal of Computational Physics* 1992; **100**:335–354.
13. Nikolopoulos N, Theodorakakos A, Bergeles G. Three-dimensional numerical investigation of a droplet impinging normally onto a wall film. *Journal of Computational Physics* 2007; **225**:322–341.
14. Fujimoto H, Shiotani Y, Tong AY, Hama T, Takud H. Three-dimensional numerical analysis of the deformation behavior of droplets impinging onto a solid substrate. *International Journal of Multiphase Flow* 2007; **33**:317–332.
15. Lunkad SF, Buwa VV, Nigam KDP. Numerical simulations of drop impact and spreading on horizontal and inclined surfaces. *Chemical Engineering Science* 2007; **62**:7214–7224.
16. Pilliod Jr JE, Puckett EG. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics* 2004; **199**:465–502.
17. Baer TA, Cairncross RA, Schunk PR, Rao RR, Sackinger PA. A finite element method for free surface flows of incompressible fluids in three dimensions. Part II. Dynamic wetting lines. *International Journal for Numerical Methods in Fluids* 2000; **33**:405–427.
18. Walkley MA, Gaskell PH, Jimack PK, Kelmanson MA, Summers JL. Finite element simulation of three-dimensional free-surface flow problems with dynamic contact lines. *International Journal for Numerical Methods in Fluids* 2005; **47**:1353–1359.
19. Kang Q, Zhang D, Chen S. Displacement of a three-dimensional immiscible droplet in a duct. *Journal of Fluid Mechanics* 2005; **545**:41–66.
20. Ding H, Spelt PDM. Wetting condition in diffuse interface simulations of contact line motion. *Physical Review E* 2007; **75**:046708.
21. Liu H, Krishnan S, Marella S, Udaykumar HS. Sharp interface Cartesian grid method II: a technique for simulating droplet interactions with surfaces of arbitrary shape. *Journal of Computational Physics* 2005; **210**:32–54.
22. Veldman AEP, Gerrits J, Luppens R, Helder JA, Vreeburg JPB. The numerical simulation of liquid sloshing on board spacecraft. *Journal of Computational Physics* 2007; **224**:82–99.
23. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
24. Hirt CW, Richardson JE, Chen KS. Simulation of transient and three-dimensional coating flows using a volume-of-fluid technique. *Fiftieth Annual Conference of the Society for Imaging Science and Technology*, Cambridge, MA, 18–23 May, 1997.
25. Hirt CW, Brethour JM. Moving contact lines on rough surfaces. *Fourth European Coating Symposium*, Brussels, Belgium, 1–4 October 2001.

26. Popinet S. The Gerris Flow Solver. <http://gfs.sourceforge.net>.
27. Popinet S. Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *Journal of Computational Physics* 2003; **190**:572–600.
28. Almgren AS, Bell JB, Collela P, Howell LH, Welcome ML. A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations. *Journal of Computational Physics* 1998; **142**:1–46.
29. Gueyffier D, Nadim A, Li J, Scardovelli R, Zaleski S. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics* 1998; **152**:423–456.
30. Youngs DL. Time-dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluid Dynamics*, Morton K, Baines M (eds). Academic Press: New York, 1982; 273–285.
31. Mugele F, Baret J-C. Electrowetting: from basics to applications. *Journal of Physics: Condensed Matter* 2005; **17**:705–774.