COMMUNICATIONS IN COMPUTATIONAL PHYSICS
Vol. **5**, No. 1, pp. 1-41

Commun. Comput. Phys.
January 2009

REVIEW ARTICLE

Marker-Based, 3-D Adaptive Cartesian Grid Method for Multiphase Flow Around Irregular Geometries

Eray Uzgoren[1,*], Jaeheon Sim[2] and Wei Shyy[2]

[1] *Department of Mechanical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, 24060, USA.*
[2] *Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, 48109, USA.*

Received 12 January 2008; Accepted (in revised version) 27 May 2008

Available online 15 July 2008

Abstract. Computational simulations of multiphase flow are challenging because many practical applications require adequate resolution of not only interfacial physics associated with moving boundaries with possible topological changes, but also around three-dimensional, irregular solid geometries. In this paper, we highlight recent efforts made in simulating multiphase fluid dynamics around complex geometries, based on an Eulerian-Lagrangian framework. The approach uses two independent but related grid layouts to track the interfacial and solid boundary conditions, and is capable of capturing interfacial as well as multiphase dynamics. In particular, the stationary Cartesian grid with time dependent, local adaptive refinement is utilized to handle the computation of the transport equations, while the interface shape and movement are treated by marker-based triangulated surface meshes which freely move and interact with the Cartesian grid. The markers are also used to identify the location of solid boundaries and enforce the no-slip condition there. Issues related to the contact line treatment, topological changes of multiphase fronts during merger or breakup of objects, and necessary data structures and solution techniques are also highlighted. Selected test cases including spacecraft fuel tank flow management and liquid plug flow dynamics are presented.

AMS subject classifications: 76T30

Key words: Multiphase flows, irregular geometry, interface tracking, adaptive Cartesian grid, non-conforming boundary methods, contact line treatment.

*Corresponding author. *Email addresses:* uzgoren@vt.edu (E. Uzgoren), honeypot@umich.edu (J. Sim), weishyy@umich.edu (W. Shyy)

Contents

1 Introduction

Fluid flows involving interactions between liquids and gases can be observed in a wide range of engineering applications. When capillary effects become significant, physical mechanisms and geometric characteristics associated with the interface need to be accurate resolved. In such circumstances, numerical simulations of interfacial flows are required to resolve the location of the interface based on the conditions arising from surface tension, viscous stresses, and pressure distributions, and distinctive material properties of the constituents. As reviewed by multiple authors [1–6], there exist numerous methods for tracking the location and the shape of the interface as well as for applying proper treatments around the interface. The computational techniques for treating moving interfaces are typically categorized into three separate groups:

- Lagrangian methods [7, 8] that modifies the grid to match the interface location,
- Eulerian methods [4, 9–13] that extract the interface location with the help of a scalar function on a stationary grid,
- Eulerian-Lagrangian methods [5, 14–18] that utilize a separate set of grid representing the interface on a stationary grid. Grid that represents the front can move freely based on the solution obtained on the stationary grid.

Once the location is known, various methods are usually employed for establishing the interfacial conditions, which impose the discontinuous pressure and viscous stresses across the interface as a result of the surface tension forces. In the literature [1, 2, 4, 19], these methods can be observed in two separate groups:

- Continuous interface methods [5, 14, 15, 20, 21] (CIM) that solves one set of equations by smearing out the flow properties around the interface,
- Sharp interface methods [12, 22–27] (SIM) that impose the conditions directly on the interface by considering different sets of equations for different phases.

With the fast progress made in recent years, various combinations of the interface tracking methods and interfacial conditions models have been proposed in order to capture

the interfacial dynamics. As highlighted by Uzgoren et al. [1], the difficulties that rise during developing a model for interfacial flows are limited not to the algorithmic challenges, but also to the compromise made between the computational cost and accuracy capturing the interfacial dynamics. The computational cost is mainly determined by solution of the non-linear Navier-Stokes equations on local adaptive meshes, the identification and treatment of marker-based triangularized fluid-fluid interface and solid boundary conditions, and interaction between the Navier-Stokes equations and moving interfaces.

In addition, fluid flow simulations for practical applications often require capabilities of handling complex geometries. Imposing boundary conditions on stationary/moving solid objects face similar challenges with the methods for multiphase flows. The boundary-conforming methods, in which the grid conforms to the boundary, require preprocessing for grid preparation and grid modification and/or re-gridding for moving objects. In addition, methods which modify only the grid layer close to the geometry such as the cut-cell [28] or other similar methods [29], can cause an increase in the condition number of the coefficient matrix of the pressure Poisson equation. Non-boundary-conforming methods are computationally more economic. As highlighted by Gilmanov and Sotiropoulos [30], the challenges for non-boundary tracking methods include (i) the representation of the boundary geometry and its relation with the fixed computational grid, and (ii) imposing the proper boundary conditions. The identification of the boundary's geometry and location can be addressed by the methods described for multiphase front tracking/capturing. Using marker-based multiphase front tracking offers compatibility between the fluid-fluid and the fluid-solid interfaces. A major challenge is how the boundary conditions are incorporated into the overall algorithm without deteriorating the accuracy or significantly adding the computational cost. Even though the continuous interface methods can adequately capture a wide range of problems involving fluid-fluid interfaces, such methods can cause numerical instabilities if the material property jump across the interface is high. Using a sharp interface method [15, 30, 31] for the solid interfaces can accurately capture the boundary conditions but the algorithm can be complicated because of the numerous scenarios which need to be accounted for.

This paper presents an overview of the recent advanced made in adopting and extending the marker-based continuous interface methods capable of handling problems of practical engineering interests. In particular, we discuss a cost-effective approach for a unified three-dimensional multiphase flow solver that is capable of capturing the interfacial flow dynamics,

- around complex geometries,
- with wetting surface as part of a computational boundary or the irregularly shaped solid geometry,
- with multiple multiphase fronts that define existence of multiple constituents in various phases,
- with topological changes, i.e. possibilities of a merger or a break-up.

In order to achieve the goals listed above, the present review focuses on a marker based interface tracking algorithm, which is an Eulerian-Lagrangian method. For interfacial flow conditions at the multiphase fronts, a continuous interface method (CIM) is adopted to allow utilizing a single set of equations. In order to maintain the computational cost at a feasible level, the stationary grid is implemented as a Cartesian grid which employs local adaptation [32] to fully resolve the flow features at locations near the multiphase front. In this combination, the overall method for multiphase fronts is referred to as Immersed Boundary Method [1, 5, 14, 20, 21]. Due to similar reasons for computational efficiency, a non-boundary conforming method, which enforces boundary conditions via sharp interface method (SIM) on the solid interfaces represented by markers, is highlighted.

When moving multiphase front comes in the vicinity of a wetting surface, defined at a computational boundary or at a complex solid interface, there is a possibility of a tri-junction location formation where all phases (solid, liquid, and gas) meet. The corresponding forces at this location is not only determined by the surface tension between the liquid and gas phases, but also interactions due to solid-liquid, solid-gas phase as well as the adhesive forces keeping the contact line in touch with the solid surface. In static equilibrium, the angle between the solid surface and the multiphase front is described by the Young's equation. The major difficulty in numerical simulations involving a contact line is that the contact line is known to slip on the solid surface, which is usually defined by a no-slip condition. One way to incorporate Young's equation into the numerical simulations of droplets/bubbles on solid surface is to impose the static angle determined directly at the tri-junction location to produce the quasi-equilibrium representation of the interface shape [21]. One of the drawbacks of such a treatment is that the modification at the contact line may result in a localized high curvature region on the fluid interface in the vicinity of the contact line. In order to balance pressure, viscous stresses, and surface tension around an interface of high curvatures, numerical instabilities can arise. Instead of imposing the contact angle, another possibility is to let the interface evolve into the equilibrium condition by modeling the force.

In addition to the contact point, representation of the topology of the multiphase front can be handled with the help of connectivity information corresponding to a triangulated surface. This connectivity information allows us to compute the geometric information, such as volume and curvature. Oftentimes, the algorithmic difficulty of finding the intersection between the interfaces brings additional computational cost, especially in three dimensional computations. This computational complexity can be reduced significantly by employing an indicator function, which varies from zero to one smoothly across the interfaces [18, 33], to identify the elements intersecting. Besides such a benefit, another advantage of using the indicator function is to allow a single set equation formulation of the continuous interface methods by treating the fluid properties such as density and viscosity across an interface over a finite thickness. Furthermore, the indicator function is utilized to remove the algorithmic complexity for the problems in which topological changes occur. This can be achieved by means of level-contour construction

algorithm [18,34].

We highlight the capabilities of the coupled techniques by exploring the following flow problems:

- Attaining equilibrium in three-dimensional capillary tubes to assess the validation of contact line treatment;
- Elongated liquid plug problem as a practical application of the liquid drug delivery into the lungs;
- Dynamics of the draining process of liquid fuel contained in a spacecraft fuel tank subjected to acceleration in axial and radial directions;
- Sloshing motion of the liquid fuel contained in a spacecraft fuel tank subjected to a sudden reduction in acceleration.

2 Numerical methods

The marker based method employs Eulerian and Lagrangian variables in order to perform the interfacial flow computations. Eulerian quantities are solved on the stationary background grid, whereas Lagrangian quantities arise due to the marker points defined on the interface which can move freely. A single fluid formulation for all fluid phases is made possible by smearing the properties across the interface. Incompressible Navier-Stokes equations for mass and momentum conservation are given in Eqs. (2.1) and (2.2) respectively, which accounts for the interfacial dynamics. In Eqs. (2.1) and (2.2), $\boldsymbol{u}$ is the velocity vector, p is the pressure, $\boldsymbol{F_f}$ represents the conditions of interfaces due to surface tension effects of fluid interfaces, and $\boldsymbol{F_s}$, represents the forcing function to establish no-slip condition on solid interfaces.

$$\boldsymbol{\nabla}\cdot\boldsymbol{u}=0, \tag{2.1}$$

$$\frac{\partial\rho\boldsymbol{u}}{\partial t}+\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}\boldsymbol{u})=-\boldsymbol{\nabla}p+\boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}\boldsymbol{u}+\mu\boldsymbol{\nabla}^T\boldsymbol{u})+\boldsymbol{F_s}+\boldsymbol{F_f}+\rho\boldsymbol{g}. \tag{2.2}$$

In general, the pressure and viscous stresses show discontinuities across a fluid interface due to the surface tension force. Eq. (2.3) relates the jump condition of the flow properties in the normal direction, $\boldsymbol{n}$, with the surface tension force, which is directly proportional with the surface tension, σ, and the curvature, κ.

$$(p_2-p_1)-\boldsymbol{n}\cdot(\tau_2-\tau_1)\cdot\boldsymbol{n}=\sigma\kappa. \tag{2.3}$$

Solid interfaces match no-slip wall condition with a prescribed velocity field defining the motion of the solid boundaries via the force field created around the solid phases.

Eqs. (2.1) and (2.2) are solved adopting a projection method using staggered grid finite volume formulation. The pressure and fluid properties are stored at the cell center and the face-normal velocity is stored on Cartesian cell faces. The flow computation follows the following steps:

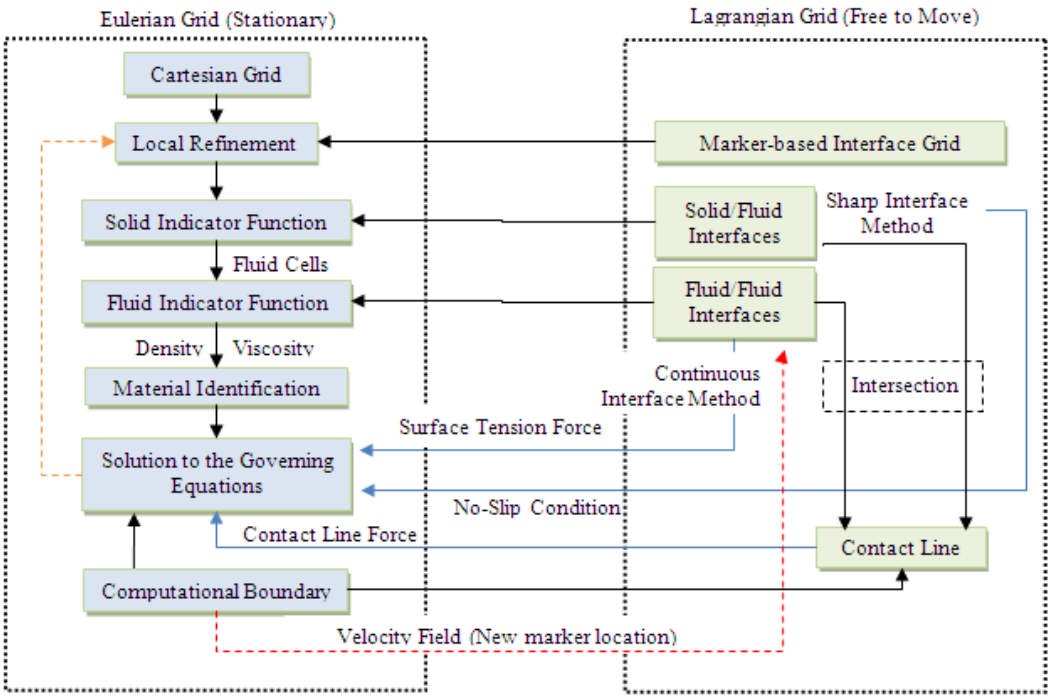


Figure 1: Summary of the numerical method and the interactions between Eulerian and Lagrangian descriptions.

Step 1: Predictor-step

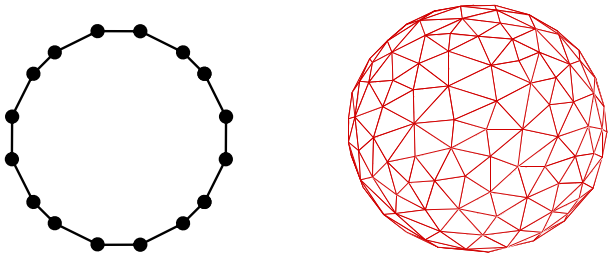
Solve the momentum equation for an intermediate velocity field $\boldsymbol{u}^*$ using Eq. (2.4) where all the known values such as surface tension source, forcing function for solid geometries, gravitation, convection and old time-step viscous term due to Crank-Nicholson method are lumped into $\boldsymbol{S}^n$. The term, $\boldsymbol{a}_v$, corresponds to the other half of Crank-Nicholson method and represents the coefficients of the stiffness matrix ($\boldsymbol{f}^*_{visc} = \boldsymbol{a}_v \boldsymbol{u}^*$). Temporal discretization is facilitated by the 2$^{\text{nd}}$ order Runge-Kutta integration. The pressure term is approximated using the old time pressure field, its gradient is integrated over a control surface, denoted by $\boldsymbol{dA}$. Subsequently, remove the effect of pressure term by shifting the velocity field back to obtain another intermediate velocity field $\boldsymbol{u}^{**}$ using Eq. (2.5).

$$\left[\Delta V \frac{\rho}{\Delta t} - \boldsymbol{a}_v\right] \boldsymbol{u}^* = -\int_{dA} \boldsymbol{\nabla} P^n \cdot \boldsymbol{dA} + \boldsymbol{f}^n_{visc} + \boldsymbol{S}^n, \tag{2.4}$$

$$\boldsymbol{u}^{**} = \boldsymbol{u}^* + \frac{\Delta t \boldsymbol{\nabla} P^n}{\rho^{n+1}}. \tag{2.5}$$

Step 2: Corrector-step

Correct the predicted velocity field ($\boldsymbol{u}^{**}$) using Eq. (2.6). The pressure field for this correction is computed by enforcing the velocity-divergence condition and



(a) (b)

Figure 2: Interface representation by marker points. (a) Line segments in 2D, (b) Triangular elements in 3D.

solving the Poisson equation (Eq. (2.7)). The divergence of the new velocity field $\boldsymbol{u}^{n+1}$ is zero due to incompressible flow.

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^{**} - \frac{\Delta t \nabla P^{n+1}}{\rho^{n+1}}, \tag{2.6}$$

$$\sum_{face} \left(\frac{\nabla P^{n+1}}{\rho^{n+1}} \right) \cdot \boldsymbol{dA} = \frac{1}{\Delta t} \sum_{face} \boldsymbol{u}^{**} \cdot \boldsymbol{dA}. \tag{2.7}$$

The components of the numerical algorithm and their interactions are summarized in Fig. 1.

2.1 Marker based interface tracking

In the marker based tracking, the interface is represented by marker points, each of which can also keep track of the neighboring markers for maintaining the connectivity information. The corresponding data structure is established through formation of elements; line-segments in two-dimensional computations and triangles in three-dimensional computations, as represented in Figs. 2(a)-(b). Markers store the surrounding elements' indices while elements store the neighboring elements based on the edge that they share. Boundary edges can be identified accordingly when any edge is connected to solely one element. Fig. 3 illustrates the connectivity information for two common scenarios; a regular 3D element, and a 3D boundary element. A regular 3D element, or an inner element, is when it has three neighboring elements while the boundary element has less than three elements, due to one or more edges being on a boundary. The connectivity information for the boundary edge is replaced by the information of the boundary as a negative integer instead of an element index. This negative integer can store relevant information, such as an index of a boundary condition, Absolute value of the boundary index corresponds to the computational boundary for 1 to 6, reserved for the east, west, north, south, front and back faces of the domain boundary, while the larger numbers representing the elements belonging to a possible solid interface.

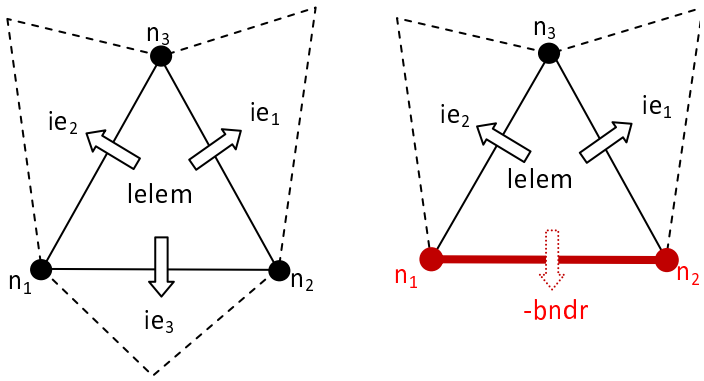


Figure 3: Connectivity information through element edges. (a) Inner element with three neighbors, (b) boundary element with two neighbors.

The marker locations, denoted by $\boldsymbol{X}$ in Lagrangian fashion, for the surface grid are computed using the marker velocities as a function of $\boldsymbol{X}$, similarly denoted by $\boldsymbol{U}$ and shown in Eq. (2.8).

$$\frac{\partial \boldsymbol{X}}{\partial t} = \boldsymbol{U}(\boldsymbol{X}). \tag{2.8}$$

Fluid interfaces use the solution field to compute the marker velocities as shown in Eq. (2.9). In this equation, the discrete Dirac delta function, $\delta(\boldsymbol{x}-\boldsymbol{X})$, is employed for converting the Eulerian velocity field, $\boldsymbol{u}(\boldsymbol{x})$, to Lagrangian form, $\boldsymbol{U}(\boldsymbol{X})$. On the other hand, solid interfaces use the prescribed velocity field to advance the marker points.

$$\boldsymbol{U}^n(\boldsymbol{X}) = \int_v \boldsymbol{u}(\boldsymbol{x})\delta(\boldsymbol{x}-\boldsymbol{X})dV. \tag{2.9}$$

As the marker points advance to a new position with time, it can lead to an unevenly distributed marker points on the interface. Such a representation can cause large errors in computations. In order to prevent such a scenario, the spacing between marker points is rearranged by addition/deletion whenever two markers come too close or too distant from each other. The criteria for the distance between the adjacent markers are estimated based on the requirements posed by the background grid and the function relating Eulerian quantities to Lagrangian. For fluid interfaces, Eq. (2.9) suggests that this function is the approximate form of the Dirac delta function. In order to achieve continuous representation for transferred quantities, each cell should contain at least one and at most two markers within its volume. This constraint can be used to approximate the distance between a marker located at $\boldsymbol{X}$ and an adjacent marker located at $\boldsymbol{X}_{ngbr}$ as a function of Eulerian grid spacing, Δ, given in Eq. (2.10).

$$\frac{\Delta}{3} < \left|\boldsymbol{X}-\boldsymbol{X}_{ngbr}\right| < 2\Delta. \tag{2.10}$$

When the distance between two marker points violate the upper bound of Eq. (2.10), a new marker is placed on the mid-point of the edge between marker $\boldsymbol{X}$ and $\boldsymbol{X}_{ngbr}$. The connectivity information is updated in the vicinity of elements to allow this marker to represent a newly created element. On the other hand, if the distance between the two marker points violates the lower bound of Eq. (2.10), the corresponding edge can be either flipped or collapsed. Furthermore, the interface grid quality can be maintained by enforcing triangular elements to have angles less than 120°. This can be achieved by removing inner nodes that only three elements share. Lastly, the volume lost during edge flipping/collapsing and marker removal can be recovered by modifying the location of the modified marker in its normal direction, as shown by Singh et al. [35, 36]. Readers may refer to Uzgoren et al. [15] for further information on restructuring of triangulated surface representation.

2.2 Indicator function

Cells on the Cartesian grid are represented by a unique material index to identify the constituents separated by interfaces. This brings an algorithmic advantage to identify the interface location as well as to assign proper material properties, i.e. density and viscosity, for flow computations. In order to facilitate a single set of equation formulation of CIM for the whole domain, a smooth variation of discontinuous material properties across interfaces are used. This is achieved with the help of a scalar function, varying from zero to one smoothly. Throughout this document, this function is referred to as the indicator function and denoted by I. Once the *indicator function* is obtained, the fluid properties such as density and viscosity, varying from values between φ_1 and φ_2, are computed using Eq. (2.11).

$$\varphi = \varphi_2 + (\varphi_1 - \varphi_2) I. \tag{2.11}$$

The material indices are assigned on the Eulerian grid using a simple and efficient method based on the painter's algorithm, which is frequently employed in computer-graphics rendering. Unlike the ray-tracing algorithm, the painter's algorithm does not require expensive computation of three-dimensional line-surface intersection. Once the material indices are marked, the sharp representation is corrected during the calculation of the indicator function in the vicinity of the interfacial markers. Indicator function is designed to yield the location of the interface, when its value becomes 0.5 on the cells of the fixed Eulerian grid. Such a feature of indicator function enables handling complex nature of geometric operations in a computationally efficient way. As a result, obtaining indicator function accurately is critical for successful numerically simulating multiphase flow problems. In the literature, two closely related but numerically distinct forms of computations can be found. One of these methods adopts the solution of a Poisson equation using the form in Eq. (2.12), while others utilize a discrete form of the Heaviside step function from the analytical solution of Eq. (2.12) using 1D form of discrete Dirac delta

function given in Eq. (2.13).

$$\nabla^2 I = \nabla \left(\int_A \delta(\boldsymbol{x}-\boldsymbol{X}) d\boldsymbol{A} \right). \tag{2.12}$$

Obtaining a numerical solution to Eq. (2.12), in which the location of the interface is represented by the Lagrangian quantity, $\boldsymbol{X}$, while $\boldsymbol{x}$ corresponds to the Eulerian description of the computational domain. Analytical form of the Dirac delta function is only non-zero at $\boldsymbol{x}=\boldsymbol{X}$, where the interface is located. However, this approach cannot be used along with a discretized set of equations as the discrete points on Eulerian and Lagrangian framework do not necessarily coincide. For this reason, approximations to the Dirac delta function, which introduce a region that represents the interface over a finite thickness, have been studied for their properties [20, 37, 38]. In the present review, the Dirac delta function approximation, that supports the conservation rules dictated by zeroth, first and second moments as described in Peskin [20], is employed as the base discrete form using the one-dimensional representation given in Eq. (2.13).

$$\phi(\boldsymbol{r}) = \begin{cases} 1-\dfrac{11}{16}|\boldsymbol{r}|-|\boldsymbol{r}|^2-\dfrac{1}{6}|\boldsymbol{r}|^3 & 1\le|\boldsymbol{r}|\le 2, \\ 1-\dfrac{1}{2}|\boldsymbol{r}|-|\boldsymbol{r}|^2+\dfrac{1}{2}|\boldsymbol{r}|^3 & 1\le|\boldsymbol{r}|\le 1, \\ 0 & \textit{otherwise}. \end{cases} \tag{2.13}$$

In Eq. (2.13), $|\boldsymbol{r}|$ is the distance between the cell-center to the interface location, and is normalized by the cell spacing, h. Because $\phi(\boldsymbol{r})$ becomes zero when the distance is larger than two cell width, the smearing region becomes limited to two-cell width on each side of the interface.

One way to extend the one-dimensional representation of the discrete Dirac function to two- and three-dimensions is to use the multiplication rule as presented in Eq. (2.14) [20]. This approach is attractive due to its low computational cost.

$$\delta_h(\boldsymbol{r}) = \frac{1}{h_x h_y h_z} \phi(r_x)\cdot\phi(r_y)\cdot\phi(r_z). \tag{2.14}$$

In Eq. (2.14), the distance vector, $\boldsymbol{r}$, is presented in Cartesian components, indicated by the subscripts. This results in a Dirac delta function, δ_h, which is an approximation based on grid spacing, $\boldsymbol{h}$.

On the other hand, when the information of the minimum distance is readily available as in the case of level-set methods, it is possible to utilize the distance function with the one-dimensional form of discrete Dirac function, ϕ, directly as shown in Eq. (2.15).

$$\delta_h(r_x,r_y,r_z) = \frac{1}{\sqrt{h_x^2+h_y^2+h_z^2}} \phi\left(\sqrt{r_x^2+r_y^2+r_z^2}\right). \tag{2.15}$$

The indicator function can be obtained via solving Eq. (2.12) on the Cartesian grid, where it is defined. The discrete form is presented in Eq. (2.16) for a cell, denoted by the subscript P.

$$\sum_{N=NGBR} \frac{(I_N - I_P)}{\Delta} A_{NP} = \sum_{N=NGBR} (A_{NP} S_{NP}). \tag{2.16}$$

In Eq. (2.16), Δ is the distance between the neighboring cell-center to the current cell, and the right hand side term, S_{NP}, is computed at the cell faces using the Dirac delta function to convert the surface quantity into its volumetric form as given in Eq. (2.17).

$$S_{NP} = \sum_{edges} \delta_h(r_x, r_y, r_z) n_l \cdot A_l. \tag{2.17}$$

In Eq. (2.17), the summation is performed over the interface elements' edges, and its contribution on face between cells N and P is computed based on the non-dimensional distance, (r_x, r_y, r_z). In practice, computing S_{NP} directly at the faces can create oscillations due to a large gradient in the source term. One remedy to this problem is to apply smoothing via computing S_{NP} at the cell centers and computing face values by linear-interpolation.

Considering discrete Delta function's support, the computations are only performed over the two-cell width region on each side of the interface to reduce the computational cost of the Poisson equation. Fig. 4 illustrates this region for the computation. This region is obtained by determining cells around each surface node with two-cell width radius. The boundary conditions away from the interface are set to yield the desired variation, i.e. from zero to one. This approach is computationally effective when these boundary conditions are away from the interface location.

On dealing with contact line problems, in which the interface is on a solid surface, the above approach requires boundary conditions in the vicinity of the interface. Because the variation of the indicator function at this region depends on the normal direction, it is difficult to utilize an appropriate boundary condition. One possible condition is to assume zero variation in the indicator value at the normal direction to the boundary. However, this condition leads to an interface representation that makes 90° to the domain boundary, which can result in a different interface shape on the Cartesian than the actual interface at angles away from 90°. Using linear extrapolation can also cause incorrect interface line ($I = 0.5$) around this region. Fig. 5 illustrates such a scenario.

This issue can be handled using the alternate way of computing the indicator function, which utilizes the shortest distance value between the cell-center to the interface location by the integrating the one-dimensional form of discrete Dirac function given in Eq. (2.18).

Utilization of Eq. (2.17) requires the determination of the shortest distance, $|\boldsymbol{r}|$, which can easily increase the computational time when performed on a large surface grid. In order to avoid such a difficulty, similar analogy to what has been described in Fig. 5 is adopted. Specifically, during the isolation of the computational region indicated in

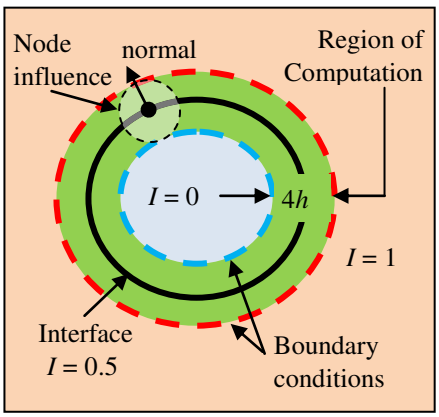


Figure 4: Region of computation and boundary conditions.

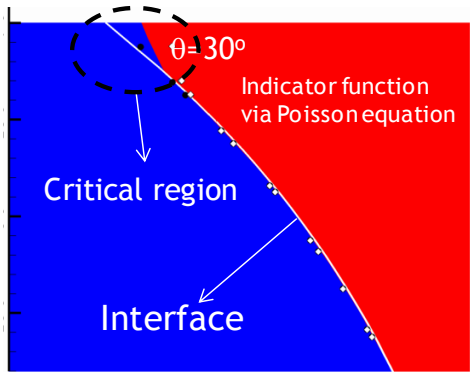


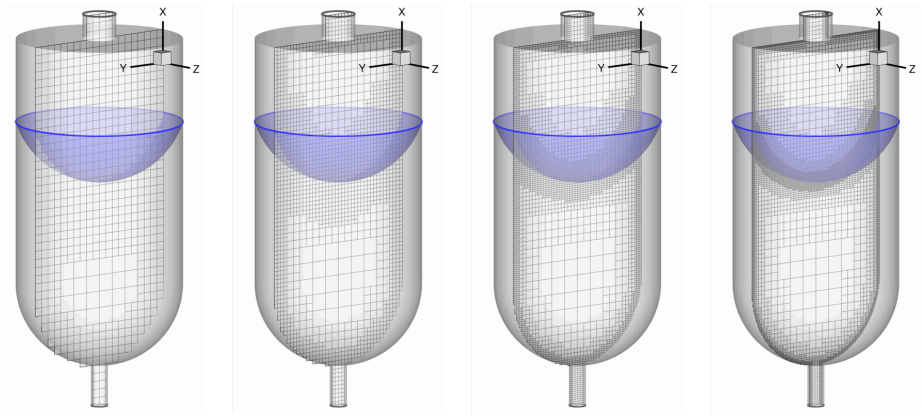
Figure 5: Indicator function constructed solving the Poisson equation misrepresents interface location at angles less than 30°.

Fig. 5, each element computes its distance from its geometric center to the cell-centers which are in their zone of influence and compares it to the distance value contained in that particular cell to find the minimum distance from the cell to the overall interface. In order to overcome the difficulties caused by sharp corners, the distance values are interpolated to face-centers from the cell-centers. Furthermore, this algorithm is coupled with identification of material tags on the Cartesian grid to save computational time.

$$I(r)=\frac{1}{8}\begin{cases} 0 & r<-2, \\ \frac{23}{4}-2\pi+5r+r^2-\left(\frac{r}{2}+\frac{3}{4}\right)\sqrt{-4r^2-12r-7}-\frac{1}{2}\sin^{-1}\left(-\frac{2r+3}{\sqrt{2}}\right) & -2<r\leq -1, \\ \frac{15}{4}-2\pi+3r+r^2+\left(\frac{r}{2}+\frac{1}{4}\right)\sqrt{-4r^2-4r+1}-\frac{1}{2}\sin^{-1}\left(-\frac{2r+1}{\sqrt{2}}\right) & -1<r\leq 0, \\ \frac{17}{4}+2\pi+3r-r^2+\left(\frac{r}{2}-\frac{1}{4}\right)\sqrt{-4r^2+4r+1}-\frac{1}{2}\sin^{-1}\left(-\frac{2r-1}{\sqrt{2}}\right) & 0<r\leq 1, \\ \frac{9}{4}+2\pi+5r-r^2-\left(\frac{r}{2}-\frac{3}{4}\right)\sqrt{-4r^2+12r-7}-\frac{1}{2}\sin^{-1}\left(-\frac{2r-3}{\sqrt{2}}\right) & 1<r\leq 2, \\ 1 & r>2. \end{cases} \tag{2.18}$$

2.3 Adaptive grid

Multiphase flow problems involve multiple length scales. In order to effectively resolve the flow features in such cases, the present study considers adaptive grid refinement employed on Cartesian grids. The approach is based on isotropic refinement which splits the cells into four and eight equal sibling cells in two- and three-dimensions, respectively. The grid is represented using an unstructured data that connects cells through cell faces.



(a) Level = 1 (b) Level = 2 (c) Level = 3 (d) Level = 4

Figure 6: Snapshots of geometry-based grid adaptation for fluid interfaces.

The details of the algorithm can be found Singh and Shyy [36, 39]. Adaptation is performed based on the interface location and the flow solution quality.

The geometry-based adaptation near interfaces has different characteristics based on what interfaces represent. The fluid interfaces, for which discontinuous flow properties smeared across two layers of cells, perform refinement to achieve full resolution in their vicinity in order to capture interfacial dynamics accurately. This fully resolved region around a fluid interface is further extended to six layers of cells to handle both discrete Dirac function as well as a possibility for large deformation of interface. Fig. 6 illustrates the process of geometry based adaptation starting from a uniform base grid and refining up to four levels. On the other hand, solid interfaces, which do not employ discrete Dirac function, triggers full refinement on vicinity including two-cell layers from the interface.

Cells, those are not already fully refined, are adapted based on solution (of the flow field). The present implementation uses a curl based adaptation criterion [40] that computes a parameter ξ for each cell as shown in Eq. (2.19). The length scale, l, is estimated as the cubic root of cell-volume. The decision to refine or coarsen a cell is made by comparing ξ_{cell} to the standard deviation (Eq. (2.20)) using the criteria in Eqs. (2.21) and (2.22):

$$\xi_{cell} \sim |\nabla \otimes \boldsymbol{u}|^l, \tag{2.19}$$

$$\sigma' = \frac{1}{N_{cell}\sum_i \xi_i^2}, \tag{2.20}$$

$$\xi_{cell} > \sigma' \longrightarrow Refine\ cell, \tag{2.21}$$

$$\xi_{cell} < 0.1\sigma' \longrightarrow coarsen\ the\ cell. \tag{2.22}$$

During the adaptation procedure, the Cartesian cell center values such as pressure, temperature and face normal velocities need to be reconstructed for the newly created cells and faces. Flow variable reconstruction during cell and face coarsening is performed

simply by averaging of the corresponding cell-centered or face-centered values. Because the adaptation algorithm is triggered during the predictor step, just before solving the pressure Poisson equation, the reconstruction algorithm is not required to satisfy the divergence free velocity condition for $\boldsymbol{U}^{**}$.

2.4 Fluid interface treatment

2.4.1 Surface tension

When interface separating fluid phases, the source term arises from the surface tension (σ) and the curvature (κ) as shown in Eq. (2.23).

$$\boldsymbol{F_s} = \int_A \sigma\kappa\delta(\boldsymbol{x}-\boldsymbol{X})\boldsymbol{dA}. \tag{2.23}$$

The surface force is computed using the Lagrangian marker points, $\boldsymbol{X}$, and is translated into an Eulerian quantity, $\boldsymbol{x}$, via the approximate discrete Dirac delta function, $\delta(\boldsymbol{x}-\boldsymbol{X})$. After the governing equations are solved on the Eulerian grid, approximate Dirac delta function is also used for obtaining the marker velocity field to move marker points for obtaining the new geometric surface representation.

The surface tension forces are computed on the interface triangles. The surface tension force on a discretized interface element (curves in 2D and triangles in 3D) can be evaluated in several ways: computation with Eq. (2.24) where unit normal vector and curvature can be computed using curve fitting for two-dimensional interfaces [21,22,41] and surface fitting for three-dimensional interfaces [42]; computation using a line integral form shown in Eq. (2.25) and fitting curves/surfaces to obtain normal and tangent vectors [5,43].

$$\boldsymbol{\delta f} = \int_{\delta A} \sigma\kappa\boldsymbol{dA}, \tag{2.24}$$

$$\boldsymbol{\delta f} = \int_{\delta A} \sigma(\boldsymbol{n}\times\boldsymbol{\nabla})\times\boldsymbol{n}dA = \int_s \sigma(\boldsymbol{t}\otimes\boldsymbol{n})ds. \tag{2.25}$$

There are two important observations to be made here: the net surface tension force on a closed surface should be zero (conservation); curvature computation using interpolation based methods are numerically sensitive and often requires some form of data smoothing [21, 41, 42, 44]. The use of Eq. (2.23) does not enforce conservation whereas the line-integral form, Eq. (2.24), does not require explicit curvature computation and maintains the conservation.

The approach developed by Singh [36] uses the line integral form and computes the local normal and tangent vectors along the triangle edges using the simple approach of Al-Rawahi [43, 44] shown in Eq. (2.26) following Fig. 7. If required, the curvature can be computed using Eq. (2.27). The overall accuracy of this approach to compute surface tension force and its modeling have already been demonstrated for boiling flows [45] and

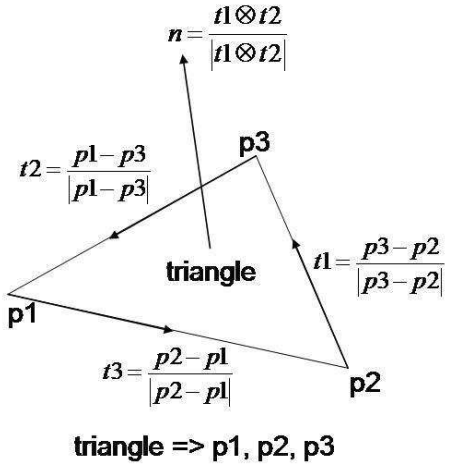


Figure 7: Computation of the unit normal and tangent vectors on interface triangles.

for dendritic solidification [44].

$$\delta f = \sum_{edge=123} \sigma(\boldsymbol{t}\otimes\boldsymbol{n})_{edge}\Delta s, \tag{2.26}$$

$$\kappa = \frac{\delta f \cdot \boldsymbol{n}}{\sigma \Delta A}. \tag{2.27}$$

2.4.2 Merger/break-up and van der Waals forces

When two interfaces come within 'close proximity', they may merge to form a single interface and similarly when sections of the same interface are in 'close proximity' the interface may break in order to form separate interfaces. In the continuum framework, the term 'close proximity' needs to be determined through a model that would facilitate the dynamics of the topological changes with clear criteria.

Many numerical models, such as the CIM approach, base the proximity of two interfaces on the smearing region, described by a scalar, indicator function, which treats the interface in two-cell widths on each side. When the same cell on the computational domain is influenced by two separate sets of interfaces, the model initiates the topological change to yield a merger or a break-up. This procedure is intrinsically utilized in the level set and VOF methods, whereas marker based methods need to update the relevant information, i.e. connectivity, for normal and curvature computations after determining the 'close proximity' via the indicator function. In the literature, these methods are referred to level-contour based reconstruction [18, 45, 46].

One of the issues of the level-contour based reconstruction is that the dynamics of a merger or a break-up becomes dependent of the grid spacing. When the grid is refined for better accuracy, the finite thickness of the interface, defined by the indicator function,

becomes smaller and smaller delaying the time of the topological changes. Furthermore, the long-range molecular forces is usually not considered, even though they can become significant in the determination of the characteristics of the topology changes [47]. These factors lead to considering a model that bases the 'close proximity' on long-range molecular forces due to van der Waals effects. The potential describing the van der Waals potential can be defined based on the minimum distance between the interfaces:

$$\phi(\boldsymbol{r}) = \frac{A'}{|\boldsymbol{r}|^3}. \tag{2.28}$$

In Eq. (2.28), A', is the Hamaker constant describing the strength of the potential based on the materials involved. The force field due to Waals potential can be estimated as follows:

$$\boldsymbol{F}_w = \boldsymbol{\nabla}\phi = \frac{\partial\phi(\boldsymbol{r})}{\partial \boldsymbol{r}}\boldsymbol{r}, \tag{2.29}$$

$$\boldsymbol{F}_w = -\frac{3A'}{|\boldsymbol{r}|^4}(r_x\boldsymbol{i}+r_y\boldsymbol{j}+r_z\boldsymbol{k}). \tag{2.30}$$

The modified incompressible Navier-Stokes equations then incorporate this force as a source term in the momentum equation, similar to the surface tension force treatment.

When $\boldsymbol{r}$ becomes smaller, the attraction forces becomes significant leading to a modification of the interface shape. The interface shape can be recovered if curvature effects are more dominant than the attraction forces. If the attraction forces are larger, the interfaces reaches a critical value of $|\boldsymbol{r}|$, for which the surface tension forces can no longer compete with the attraction forces. At this stage, the reduction in the film thickness can eventually lead to a topological change. In order to identify this critical film thickness, the ratio between the non-dimensionalized surface tension and van der Waals attraction forces can be used. The characteristic length is chosen as the film thickness to yield Eq. (2.31).

$$\frac{\rho_1}{\mu_1^2} = \left(\sigma h_0 - \frac{A'}{h_0}\right) \longrightarrow h_0^2 \sim \frac{A'}{\sigma}. \tag{2.31}$$

The distance between individual markers belonging to separate interfaces are tracked to include the influence created by the van der Waals forces. Once a set of markers hold a distance below the critical film thickness value, h_0, a reconstruction scenario is enacted by updating the markers connectivity information. Marker points, those pass beyond critical film thickness, are collected as a subset. When these are removed, some of the remaining elements become unconnected on one of their edges. These open edges are reconnected to the closest markers points from the interacting interface element by updating the connectivity information.

The above described operations can yield edges those violate the bounds described in Eq. (2.10), which can be corrected by restructuring operations that involve marker addition, edge removal/flipping to make the newly formed interface group(s) be compatible with the background fixed (Eulerian) grid.

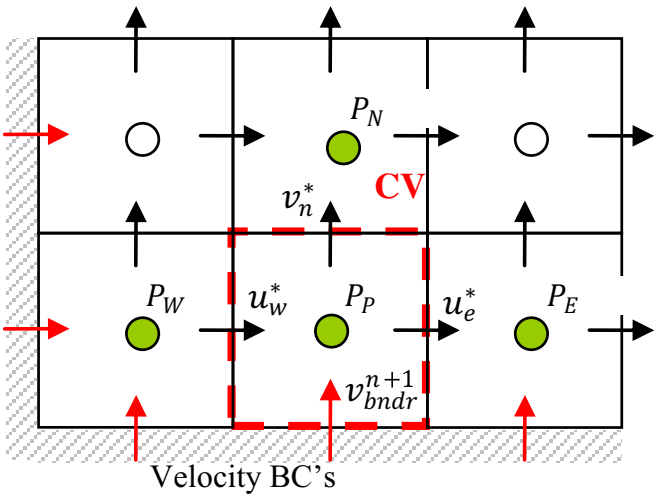


Figure 8: Pressure control volume at the computational boundaries.

2.5 Solid interface treatment

Solid interfaces are modeled using a sharp interface method that imposes the prescribed conditions on an arbitrary interface by reconstructing a force field around a solid phase. Using Eq. (2.2), the source term due to solid interfaces, $\boldsymbol{F_s}$, can be estimated at its non-zero locations with the help of the prescribed velocity at the interface, $\boldsymbol{u}_{int}$, and other forcing terms, i.e. gravity, $\boldsymbol{g}$, and surface tension forces, $\boldsymbol{F_s}$, as presented in Eq. (2.32).

$$\boldsymbol{F_s}=\rho\frac{\partial \boldsymbol{u}_{int}}{\partial t}+\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}\boldsymbol{u})_{int}-\boldsymbol{\nabla}p-\boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}\boldsymbol{u}+\mu\boldsymbol{\nabla}^T\boldsymbol{u})_{int}-\boldsymbol{F}_f-\rho\boldsymbol{g}. \tag{2.32}$$

Following the prior naming by Yang and Balaras [31], these non-zero locations are referred to as forcing faces as the present study considers a staggered variable arrangement, in which the velocity components are defined at the face-centers and the forcing field is formed using the face-centers of the cells surrounding the solid interfaces.

Recalling one of the merits of the staggered grid, there is no need for the artificial pressure boundary conditions in the continuity equation [48, 49]. At locations, where the boundary conditions are defined exactly on the face-centers, i.e. computational boundaries, the procedure of solving the pressure Poisson equation is presented in Eq. (2.33) with the guidance of Fig. 8.

$$\frac{P_E-P_P}{\Delta x}A_e+\frac{P_W-P_P}{\Delta x}A_w+\frac{P_N-P_P}{\Delta y}A_n=u_e^*A_e-u_w^*A_w+v_n^*A_n-v_{bndr}^{n+1}A_s. \tag{2.33}$$

The last term of the right hand side term in Eq. (2.28), is the total mass flux at the boundary face and the corresponding pressure gradient term on the left hand side term is not needed as mass flux is known at the next time step. Similar procedure is adopted

for the forcing faces. Hence the identification procedure considers the pressure Poisson equation on the fluid side and marks the faces as forcing if the material on the other side indicates a solid material flag. This study utilizes negative values for solid phase and positive values for the fluid phases. As a result, any cell that has a negative index value is removed from the solution procedure of the Poisson equation. These faces form the first set of the forcing faces.

Additional faces are also marked as forcing faces due to the nature of the prediction step of the projection method. When we consider the momentum equation, the viscous and advection terms require another set of faces that would yield a correct gradient at the boundary layer. In this study, this set of faces is chosen on the solid side as shown in Fig. 9. It should also be noted that, in some problems dealing with thin or zero-thickness solid interfaces, these faces can also be chosen from the fluid side [30, 31]. However, this approach would make the construction of the interpolation scheme difficult especially at the inner corner locations, where less than sufficient fluid faces exist.

Once the forcing faces are set, the forcing terms on these faces are computed using a linear interpolation scheme between the prescribed velocity field on the interface, and the predicted velocity field at the fluid side. The first point on the interpolation scheme, the closest location on the interface from the forcing face, is found by comparing the distance normalized by the grid spacing for the elements in the vicinity of the forcing face. Once determined, interpolation weights between points i and j, denoted by w_{ij}, can be computed based on inverse distance, Δ_{ij}, using Eq. (2.34) and following Fig. 10. Then the prescribed condition on the interface can be obtained for any function, ϕ, using Eq. (2.35).

$$w_{ij} = \frac{1/\Delta_{ij}}{\sum_{i=1,3}(1/\Delta_{ij})}, \tag{2.34}$$

$$\phi_i = \phi_1 w_{1j} + \phi_2 w_{2j} + \phi_3 w_{3j}. \tag{2.35}$$

The remaining points in the interpolation scheme are the fluid faces, identification of which is one of the most critical parts of the algorithm. The accuracy of the interpolation scheme improves when those fluid faces are selected as close as possible to the forcing face while avoiding an ill-conditioned scheme, which can happen when some of the interpolation points are aligned together. To satisfy the requirements for shortest distance and avoid ill-conditioned scheme between the face-centers included in the scheme, a short list of liquid faces is formed using the neighboring cells. This list is sorted using a *merge-sort* algorithm based on the distance values. The various combinations of faces are checked for their cross-product to verify their alignments starting from the best qualified distance values. This procedure results in an interpolation scheme, which can be geometrically represented in the shape of a triangle in 2D, and a tetrahedron in 3D as illustrated in Fig. 11.

The interpolation procedure is performed assuming a linear variation of any variable,

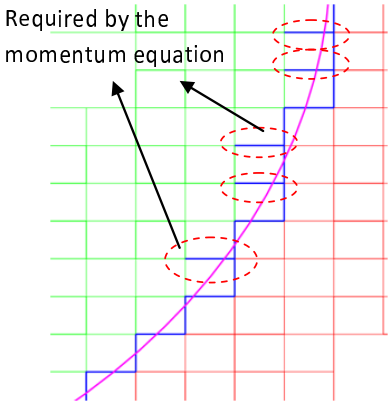


Figure 9: Identification of forcing faces. Red faces and green faces belong to fluid and solid phases, respectively. Blue color indicates the forcing faces based on pressure Poisson and momentum equation of the prediction step.

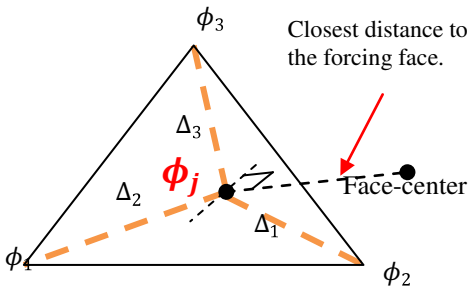
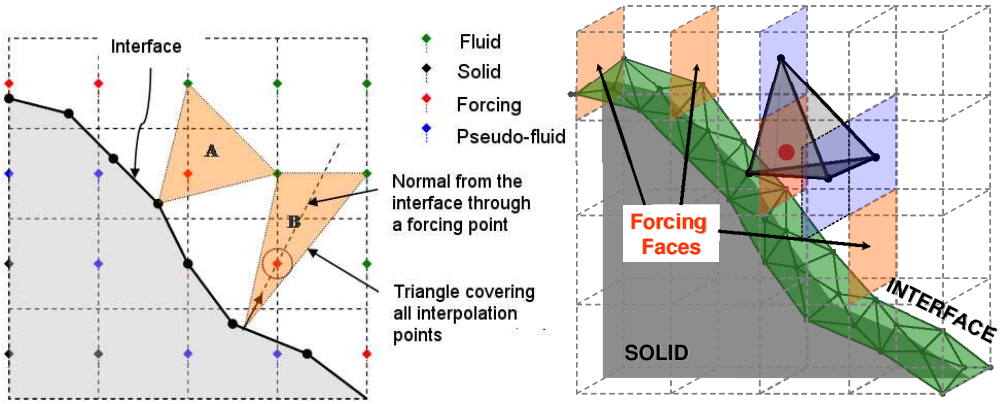


Figure 10: The closest interface element to a forcing face.



(a) (b)

Figure 11: Definition of faces around the solid interface for u-velocity (a) in 2D, (b) in 3D.

ϕ. Eqs. (2.36) and (2.37) are the formulation of the procedure in 2D.

$$\phi = b_1 + b_2 x + b_3 y, \tag{2.36}$$

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}. \tag{2.37}$$

In Eqs. (2.31) and (2.32), x_i and y_i represents the corners of the triangle presented in Fig. 11. For stationary objects, the coefficients can be obtained once and then be used for reconstructing the velocity field at each time step. On the other hand, the system has to

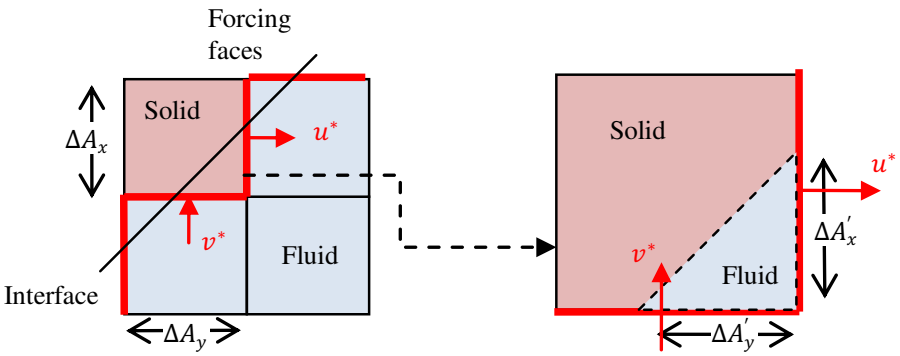


Figure 12: Region of correction to ensure divergence free velocity field.

be solved at every time step for moving boundaries. 3D computations are achieved in a similar manner by adding an additional point to obtain the coefficient of the z-coordinate, b_4.

The above interpolation scheme is utilized in the estimation of the forcing term, $\boldsymbol{F_s}$, in Eq. (2.2). Because this term is computed at the prediction step of the projection method, it is not computed explicitly which is the case for surface tension forces, $\boldsymbol{F_f}$, for the fluid interfaces. Instead, it is reflected in the predicted velocity field using the interpolation scheme discussed in the prior sections. These forcing velocity terms are utilized in the prediction step to determine the flux computation in the advection-diffusion equation as well as the mass flux values in the continuity equation as described in Eq. (2.33).

The corrected velocity field is required to satisfy the divergence free condition both locally and globally. In the staggered grid configuration, having the sum of local control volumes equal to the global control volume, this condition is automatically satisfied when the correct mass flux information is used at the boundary cells that utilize the concept presented in Eq. (2.33). As one may expect, the constructed forcing velocity field does not impose the conditions required by the continuity equation and hence is not divergence free.

Fig. 12 is an example to illustrate the correction algorithm. The forcing faces that divides solid and fluid creates staircase-like boundary faces for the pressure Poisson equation. Let u^* and v^* be the forcing velocities defined at the faces between fluid and solid phases. The correct flux to be included in the x-face needs to account for the area that is cut by the interface, $\Delta A_x'$ instead of the full area of the face, ΔA_x. It is a similar case for the y-face, that is $\Delta A_y'$ instead of ΔA_y. Hence, assuming the velocity field, u^* and v^*, is divergence free, Eq. (2.38) should be satisfied, resulting an error, e, when the full area for each face is utilized as given in Eq. (2.39).

$$u^*\Delta A_x' - v^*\Delta A_y' = 0, \tag{2.38}$$

$$u^*\Delta A_x - v^*\Delta A_y = e. \tag{2.39}$$

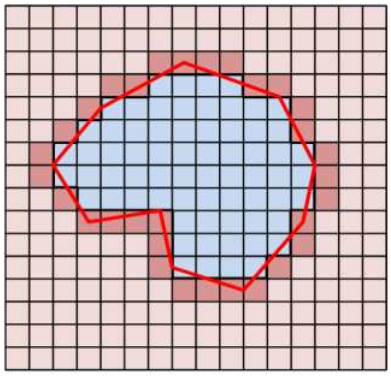


Figure 13: Global conservation with local correction.

The determination of correct face areas requires finding intersection points between the interface grid and the Cartesian grid, which can be computationally challenging and expensive. Instead, the requirement can be obtained by introducing Eq. (2.40), a correction term for the forcing face velocities to adjust the fluxes accordingly.

$$e=\epsilon(\Delta A_x+\Delta A'_y), \tag{2.40}$$

$$(u^*-\epsilon)\Delta A_x-(v^*-\varepsilon)\Delta A_y=0. \tag{2.41}$$

When this correction is applied cell-by-cell, the global conservation is automatically satisfied as a result of their individual sum at the corresponding control volumes. This is illustrated in Fig. 13. The condition that needs to be satisfied is the divergence free mass flux at the region enclosed by the irregular interface line, marked with red color. The condition is enforced locally at the black line on the regular Cartesian boundary faces, where the material tag changes sign. In Fig. 13, this corresponds to a layer of cells, colored dark. This also satisfies the global conservation requirement on the other side of the bold Cartesian boundary cells.

Starting from a divergence free velocity field at a given time step, n, the forcing function is incorporated in the predicted velocity field, u^*, on the forcing faces using Eq. (2.36) leaving the fluid faces at the u^n values. This corresponds to the first step in Table 1, which summarizes the algorithm advancing from time step $[n]$ to time step $[n+1]$. The second step involves the correction step on the first set of forcing faces, which is required only by the Poisson equation. The idea behind the correction is to enforce the global correction based on the way that the cells are cut, so that the continuity equation becomes well-posed. The rest of the terms in the momentum equation are accounted for on solving the advection-diffusion equation to predict the velocity field as appears in third item in Algorithm 2.1. The fluxes for convection and diffusion equation are computed using the conditions posed by all of the forcing faces and the predicted velocities are updated only at the fluid faces while faces on the solid side are ignored. Similarly, pressure field, defined at the cell centers, are solved at only on the fluid side using the fluxes computed by

Algorithm 2.1: Overall algorithm for solid boundary treatment

1. Apply forcing terms: $\boldsymbol{u}_i^* = \begin{cases} \boldsymbol{u}_i^n & i \longrightarrow Fluid\ faces \\ \boldsymbol{b}_{i1}+\boldsymbol{b}_{i2}x_i+\boldsymbol{b}_{i3}y_i+\boldsymbol{b}_{i4}z_i & i \longrightarrow Forcing\ faces \end{cases}$
2. Apply local correction for global conservation:
   $\boldsymbol{u}_j^{n+1} = \boldsymbol{u}_j^* + \boldsymbol{\epsilon}_j \quad j$: *Forcing faces required only by Poisson equation*
3. Solve for advection diffusion equation:
   $$\left(\Delta V \frac{\rho^{n+1}}{\Delta t} - \boldsymbol{a}_v\right)\boldsymbol{u}^{**} = -\int_{dA} \boldsymbol{\nabla} p^n \cdot \boldsymbol{n} dA + \boldsymbol{f}_{visc}^{**} - \boldsymbol{f}_{conv}^{**} + \boldsymbol{f}_f^{n+1} + \boldsymbol{g}$$
   $$\boldsymbol{u}_i^{***} = \boldsymbol{u}_i^{**} + \frac{\Delta t (\boldsymbol{\nabla} p^n)_i}{\rho^{n+1}} \qquad i: only\ fluid\ faces$$
4. Solve for continuity equation (pressure Poisson equation):
   $$\sum_i \left(\frac{(\boldsymbol{\nabla} p^{n+1})_i}{\rho^{n+1}}\right)\cdot \boldsymbol{n} dA = \frac{1}{\Delta t}\sum_i \boldsymbol{u}_i^{***}\cdot \boldsymbol{n} dA + \frac{1}{\Delta t}\sum_j \boldsymbol{u}_j^{n+1}\cdot \boldsymbol{n} dA \quad \begin{matrix} i: only\ fluid\ faces \\ j: only\ forcing\ faces \end{matrix}$$
5. Correct velocity field: $\boldsymbol{u}_i^{n+1} = \boldsymbol{u}_i^{***} - \frac{\Delta t (\boldsymbol{\nabla} p^{n+1})_i}{\rho^{n+1}} \quad i: only\ fluid\ faces$

predicted velocity field for fluid faces and the fluxes at the corrected forcing faces. Hence, the correction is only applied on the fluid faces to satisfy the divergence free condition for the next time step.

2.6 Contact line treatment

When we consider a fluid-fluid interface intersecting a solid surface, the treatment of the tri-junction locations needs to account for the presence and interactions of all three phases, fluid-fluid-solid, which can be challenging. One of the mostly discussed issues for modeling the tri-junction location, or the contact line, with Navier-Stokes equations is that the imposed no-slip condition for velocity leads to a non-integrable singularity in stress. Among the various models, ones which produce slip condition [50–52] is adopted in this study.

The angle at the contact line, shown in Fig. 14, can be used for representing by the balance of forces resulting from different intermolecular forces between solid, liquid and gas phases. The tangential component of the resulting force, F_R, is shown in Eq. (2.42). In static equilibrium, Eq. (2.42) leads to the well-known Young's equation as given in Eq. (2.43).

$$F_R = \sigma_{sg} - \sigma_{sl} - \sigma\cos\theta, \tag{2.42}$$

$$\sigma_{sg} = \sigma_{sl} + \sigma\cos\theta_0. \tag{2.43}$$

In Eq. (2.42), σ_{sg} is the surface force due to the interaction of solid and gas, σ_{sl} is the surface force due to the interaction of solid and liquid and m is the surface tension defined for liquid and gas.

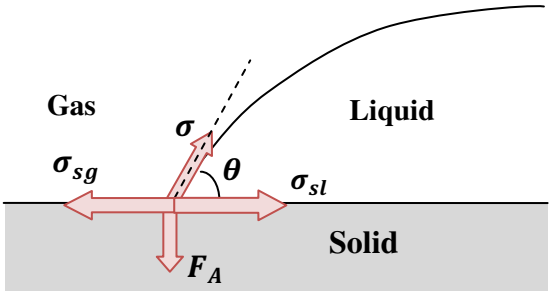


Figure 14: Forces at tri-junction (contact line). F_A represents the adhesive forces.

In this work, we adopt an approach based on a simplified version of Huang et al. [52] without considering the effects of the slip velocity on the contact angle. In the present work, flow dynamics moves the contact angle asymptotically towards a prescribed static contact angle. The force at the contact line is obtained by plugging Eq. (2.43) into Eq. (2.42), to estimate the force to recover the static contact angle.

$$F_R = \sigma(\cos\theta_0 - \cos\theta). \tag{2.44}$$

During the computation of the source term due to surface tension, the contact line region contributes to the recovery force in the tangential direction to the solid surface, $\boldsymbol{t}_s$, described by Eq. (2.44), instead of the curvature effects given in Eq. (2.25). The contact angle is enforced on open edges where the element connects to either a computational boundary or a solid interface by modifying Eq. (2.25) into Eq. (2.45). In Eq. (2.45), the angle, θ, defined by the solid surface and the line formed by the mid-point of the edge and the node across. The difference in cosine values between the computed angle, θ, and the prescribed static angle, θ_0, is translated into a force acting on the contact line. For a static problem, i.e. capillary tube simulation, this force alone derives the contact line into a given static contact angle from any given initial condition.

$$\delta \boldsymbol{f} = \sum_{\substack{i=closed \\ edges}} \sigma(\boldsymbol{t}_f \otimes \boldsymbol{n})_i \Delta s + \sum_{\substack{i=closed \\ edges}} \sigma(\cos\theta_0 - \cos\theta)_i \boldsymbol{t}_s \Delta s. \tag{2.45}$$

The slip velocity condition on interface markers at the contact line is imposed during the transfer of Eulerian velocity field into the Lagrangian marker points. Instead of imposing the full Dirac function region in Eq. (2.9), we consider a one-sided averaging via Dirac function. This treatment is represented by Eq. (2.46).

$$\boldsymbol{U}(\boldsymbol{X}) = \frac{\sum_x \boldsymbol{u}(\boldsymbol{x})\delta(\boldsymbol{X}-\boldsymbol{x})}{\sum_x \delta(\boldsymbol{X}-\boldsymbol{x})}. \tag{2.46}$$

In Eq. (2.46), the summation of delta function in denominator will yield unity (as a property of delta function) further away from the contact line whereas it is less than unity for markers in the proximity of a solid surface. As a result, casting the velocity field on the contact line yields a slip condition on the solid surface.

2.7 Intersecting solid and fluid interfaces

The contact lines can occur not only on the boundaries of the fixed Cartesian grid but also at the solid interfaces that have irregular geometries. Computing the contact line force requires identification of the angle between the fluid and solid interfaces. In order to estimate the contact angle, it is crucial to know which elements of fluid and solid interfaces are in contact. This requirement is imposed by identifying the intersection marker and relating the corresponding fluid and solid markers/elements with each other.

This identification process starts during the computation of the indicator function. The material properties assigned in a particular order, starting with solid interfaces followed by fluid interfaces. This allows us to leave an identification number for solid marker on the Cartesian grid cells. As the indicator function computation involves the shortest distance, each cell around a solid interface keeps the index of the closest solid marker point. This information is then used among the fluid interfaces to determine whether they are in the vicinity of a solid boundary.

When there is wetting on a solid interface, the fluid interfacial structure includes open elements, as described earlier in Section 2.2. The open elements lack of connectivity information on one of their edges to another element. The markers on this edge is extended or trimmed to fit on the closest solid interface. During this process, the orientation, i.e. normal direction, of the element is maintained.

The identification of open elements in a fluid interface is achieved by utilizing the indicator function which is available on the Cartesian grid. The information of solid indicator function, I_s, is transferred onto fluid interface markers, $\boldsymbol{X}_f$, via Eq. (2.47) from the Cartesian grid cells, $\boldsymbol{x}$.

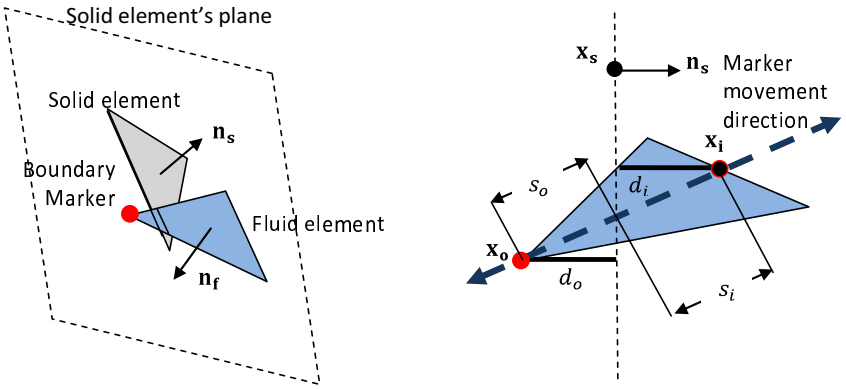
$$I_s(\boldsymbol{X}_f) = \int_v I_s(\boldsymbol{x})\delta(\boldsymbol{x}-\boldsymbol{X}_f)dv. \tag{2.47}$$

When the value of I_s at any marker is less than half, which corresponds to the location of the solid interface, it is marked to be trimmed. In addition, elements having markers all to be trimmed are removed from the fluid interface list. For elements having some markers lying inside and outside the solid interfaces are marked as open elements, for which the connectivity information is filled with the corresponding solid marker point, determined by checking index value of the closest solid marker from the underlying Cartesian grid cell.

Once the intersecting solid and fluid elements are known, markers on the open edge are moved on to the solid interface using the distance information as shown in Fig. 15.

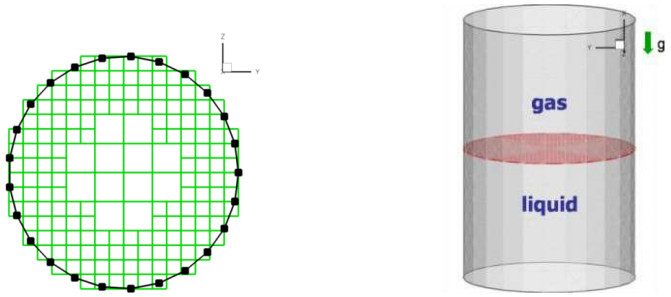
3 Computational assessment

The presented approach is tested using the following case studies: (i) Capillary tube simulation at zero gravity with various contact angle and parameters (ii) liquid plug problem, and (iii) 3D simulation of spacecraft fuel tank dynamics including liquid fuel



(a) (b)

Figure 15: Intersecting fluid and solid interfaces. (a) intersection between elements (b) Snapping procedure (point of view is tangential direction to the solid element. Marker on the open edge is marked with red.



(a) (b)

Figure 16: The computational configuration for 3D Capillary tube simulations. (a) adaptive Cartesian grids and tube wall represented by solid interface, (b) configuration with initial flat interface.

draining at a tilted micro-gravity and liquid sloshing motion under a sudden change of acceleration direction and/or magnitude. These cases are presented in the following.

3.1 Capillary tube simulation

Uzgoren et al. [53] simulated the Capillary tube with initially flat interface at zero-gravity condition varying contact angles and non-dimensional parameters. The initial flat interface between gas and liquid is deformed into the curved steady interface shape with the given contact angle by the contact line force applied on the tri-junction point between gas, liquid, and solid wall. The grid convergence and accuracy for various contact angles were first studied in axisymmetric domain. Then, 3D simulations were conducted by representing the circular tube wall using solid interface with triangular element in Cartesian adaptive background grid. These configuration and initial flat interface in a

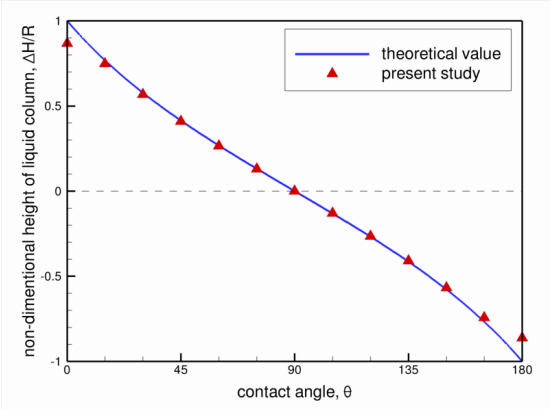


Figure 17: Comparison of non-dimensional height of liquid column between theoretical values and present axisymmetric computation. $La=10^4$, and a maximum of 129 grid points are used along radial direction.

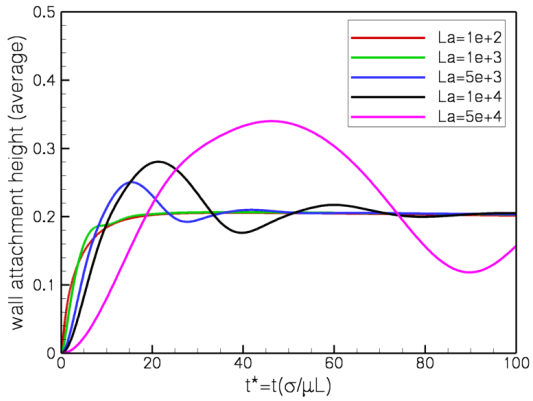
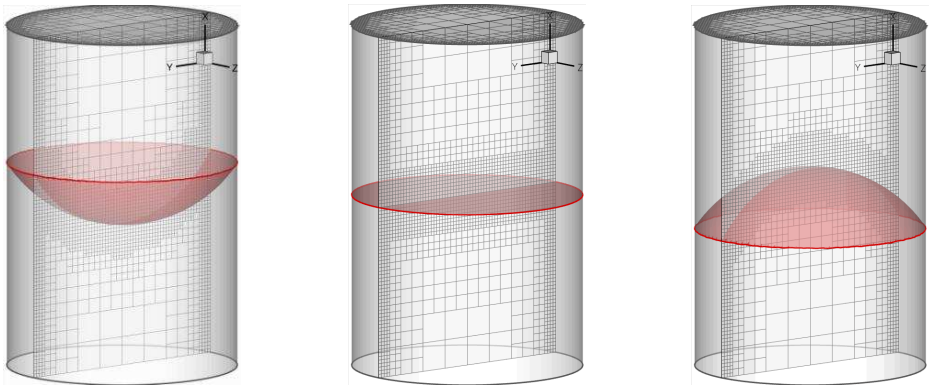


Figure 18: The wall attachment point of 3D Capillary tube with 30° contact angle. The Laplace number varies from 100 to 50,000.

Capillary tube are shown in Fig. 16. The density ratio and viscosity ratio are 1000 and 100 respectively. The Bond number, $Bo=\rho g L^2/\sigma$, is zero for zero-gravity condition, and it is tested by varying Laplace number, $La=\rho\sigma L/\mu^2$, from 10^2 to 10^5.

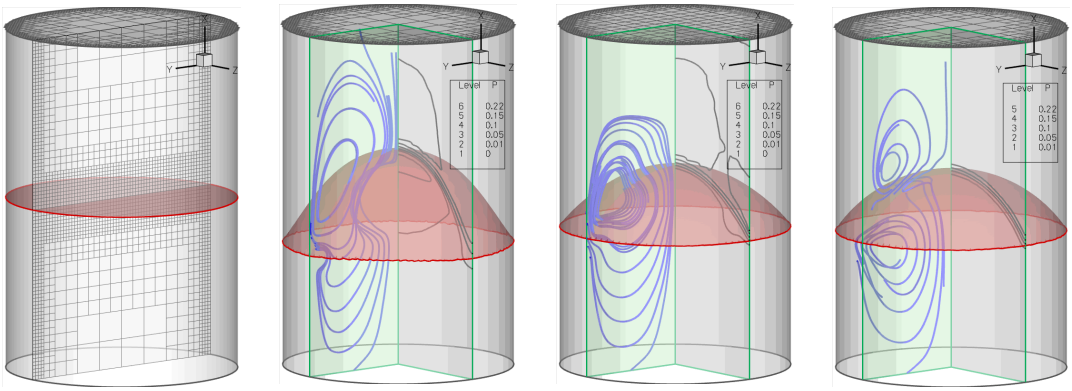
The height of liquid column (ΔH), the difference between wall attachment point and centerline location, is normalized by the radius of tube (R) and chosen for verifying the performance of present contact angle model. Fig. 17 shows the comparison between the analytical solution [49] and the present axisymmetric computation results with various contact angles from 0° to 180° at $La=10^4$. Maximum 129 grid points are used along tank radius based on the grid convergence test. The current contact angle model works very well with various contact angles, and shows good consistency with theoretical values especially from 15° to 165°, where the difference is just around one cell distance.

The 3D Capillary tube geometry in Fig. 18 was tested with various Laplace number for validating the 3D performance of contact angle and solid interface. At the same grid



(a) $\theta=30^\circ$ (b)$\theta=90^\circ$ (c) $\theta=150^\circ$

Figure 19: The steady interface shape of 3D Capillary tube for different contact angles. (a) 30°, (b) 90°, (c) 150°



(a) $t^*=0.0$ (b)$t^*=20.0$ (c) $t^*=40.0$ (d) $t^*=100.0$

Figure 20: Evolution of streamlines and pressure of 3D Capillary tube with 150° contact angle.

resolution, the 3D computation has same accuracy as 2D, where the interface location is tracked with an accuracy of one cell distance. The change of wall attachment point at 30° contact angle is tracked in time from initial flat interface shape to the final steady state in Fig. 18. With a modest Laplace number, for example, $La=100$, the interface shape evolves smoothly without overshooting, but a large Laplace number can cause oscillations in interface shapes before it reaches the steady state solution because the relatively large surface tension and small viscosity create a larger contact line force, and thus movement. Fig. 19 shows the 3D sample snapshot of deformed interface shape for given contact angles 30°, 90°, and 150°. The Cartesian grid is locally adapted dynamically tracking the interface location for effective computation. The streamlines and pressure contour in a plane of 3D domain are shown in Fig. 20.

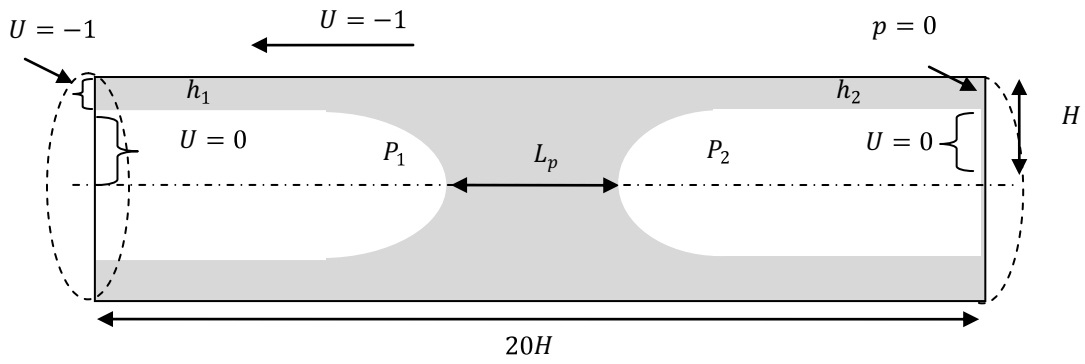


Figure 21: Computational setup and the boundary conditions.

3.2 The liquid plug flows

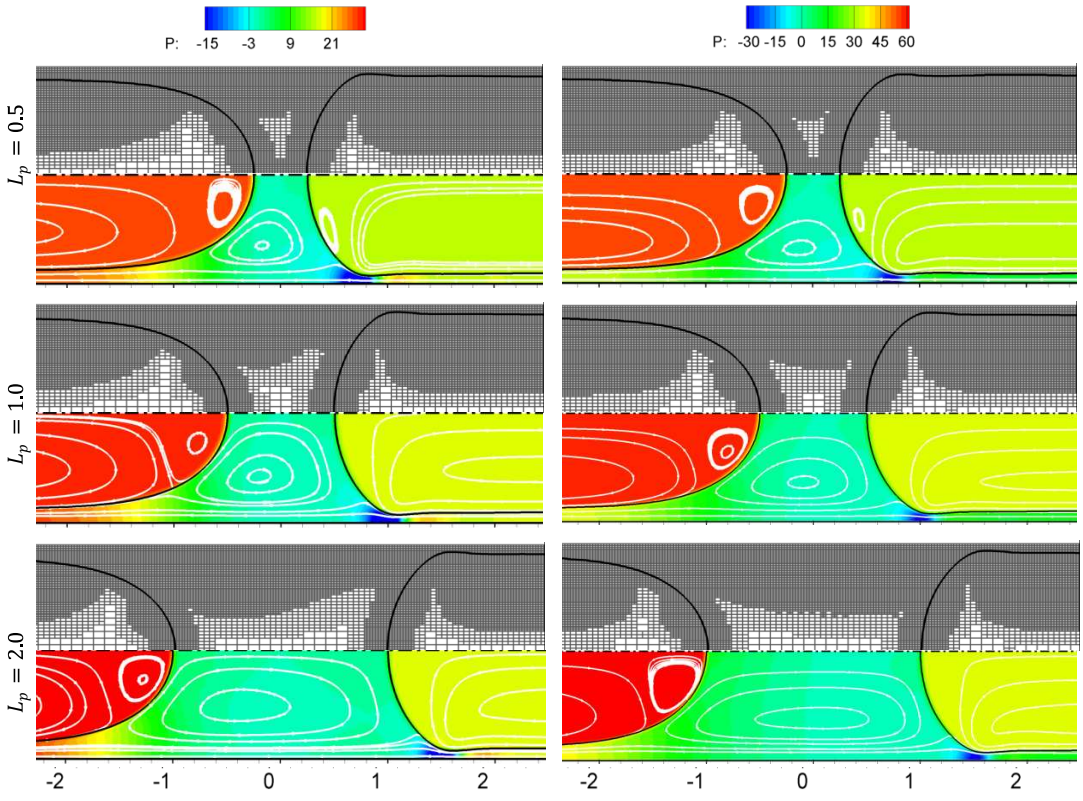
The lung airways are coated with a thin liquid film, which can become thicker during various situations, such as congestive heart failure, asthma, cystic fibrosis, delivery of drugs [54]. This liquid film can become unstable and form airway closure, or liquid plug, which can initiate undesired outcomes for the patient. In addition to having a role of blocking gas exchange, liquid plugs can cause damage to pulmonary epithelial cells because of the additional mechanical stresses they bring [55]. These factors become more significant especially when liquid plug experiences a rupture (reopening of the airway), causing a sudden change in air pressure. Many factors can lead to a rupture, including the liquid properties, gravity, propagation speed, presence of downstream plugs, airway geometry, plug size, and interfacial activity [56].

The dynamics of a liquid plug rupture for a small plug length can be affected by the van der Waals attraction forces. Furthermore, the van der Waals forces can be the major source determining the threshold of the rupture. The critical length is determined based on non-dimensional form of Navier-Stokes equations, which is consistent with prior considerations [47,57] focusing on thin liquid films.

The steady state solutions are obtained at various plug lengths to compare the present approach with prior studies of Fujioka and Grotberg [55], who employed a Lagrangian type interface tracking in a 2D channel. In addition to planar results, we extend their study for axisymmetric computations.

Fig. 21 shows the computational setup based on the channel geometry and the liquid plug. Two air-bubbles containing different pressure levels drive the liquid plug at a constant speed. The liquid plug length, denoted by L_p, is the distance between air bubbles. The liquid film thickness for air bubbles are denoted as h_1 and h_2.

The flow conditions are set by taking the characteristic length as the radius, denoted with H, and characteristic velocity as the bubble tip velocity, denoted with U. Focusing on the liquid plug region, no-slip wall condition at the pipe wall is employed by changing the computational framework to follow the bubble tip, which propagates at a constant



(a) (b)

Figure 22: Two-dimensional steady state solutions of the liquid plug problem with plug lengths: $L_P = 0.5$, $L_P = 1.0$, and $L_P = 2.0$, (a) planar geometry, (b) axisymmetric geometry. Colors indicate pressure levels.

speed. Using the described computational framework, axial velocity at the channel walls is set to negative unity yielding a zero-velocity for the bubble tip. The boundary conditions at the channel entrance and exit are designed to honor the infinitely long channel assumption. No air is allowed to enter or escape from the channel by setting the velocity profiles inside the air bubble to zero. On the other hand, a linear velocity profile is assigned to liquid at the entrance based on velocity profiles on the interface and the no-slip wall. Outflow conditions based on constant pressure leveled at zero is assigned at the channel exit. Lastly, symmetry conditions are considered at the centerline to facilitate half-domain solutions to the planar and axisymmetric numerical simulations.

Based on the parameters presented in Fig. 21, the non-dimensional parameters based on liquid's density and viscosity are given in Eq (3.1).

$$Re = \frac{\rho_l H U}{\mu_l}; \quad Ca = \frac{\mu_l U}{\sigma}. \tag{3.1}$$

The flow properties of steady state plug propagation are obtained via time marching

from an initial condition of a zero velocity field and an arbitrary interfacial geometry, which is chosen to be a combination of half circles in the vicinity of the liquid plug and straight lines to resemble infinitely long air bubble with no variation in film thickness. The shape of the plug and the mechanical stresses at the channel wall at $Re = 50$ and $Ca = 0.05$ are considered for three different plug lengths, i.e. $L_P = 0.5$, $L_P = 1.0$ and $L_P = 2.0$.

Fig. 22 shows the pressure contours and the streamlines for three different values of L_P in a two-dimensional channel and axisymmetric circular tube along with the steady state grid distributions. The pressure drop between the air bubbles drives the liquid plug through the channel. A circulation zone at the plug core is observed for each of the solution, causing the film thickness at the upstream of the liquid plug to be at a minimum. This location is also marked as the minimum liquid pressure location, which is accompanied by large pressure gradients on each side. A qualitative comparison between the axisymmetric and the planar geometries suggest that the flow structures are similar to determine the critical locations of wall stresses, while the interface shapes, hence the trailing film thickness, h_1, and the magnitude of the pressure drop across the plug are quantitatively different, as presented in Table 2. For instance, the axisymmetric geometry is observed to have smaller film thickness than the planar cases for all plug lengths. In addition, the trailing film thickness remains the same for plug lengths of 1.0 and 2.0, in spite of an increasing trend seen in planar cases.

Table 1: Steady state trailing film thickness and the pressure drop across the plug, at $Re = 50$ and $Ca = 0.05$.

	Axisymmetric			Planar					
	Present Computations			Present Computations			Fujioka and Grotberg [55]		
L_p	0.5	1.0	2.0	0.5	1.0	2.0	0.5	1.0	2.0
h_1	0.111	0.115	0.115	0.114	0.119	0.123	0.113	0.119	0.123
$\Delta P/L_P$	46.0	24.0	14.7	21.7	12.0	7.1	23.0	13.0	8.0

In Table 1, the film thickness length and the pressure drop across the plug for planar geometries are compared with the prior numerical study by Fujioka and Grotberg [55]. Their numerical approach adopts a Lagrangian method to identify the moving air bubble shape with a single fluid formulation, which assumes constant pressure inside the air bubble. In present approach, this assumption is replaced by assigning material properties to air, i.e. density ratio as 50 and viscosity ratio as 10. Despite the difference in formulations, both approaches agree on the characteristics of the flow. For instance, both 2D planar results show that the steady state trailing film thickness, h_1, decreases with a decreasing plug length for the two-dimensional channel and the steady state pressure drop across the plug, $\Delta P/L_P$, shows the same decreasing trend for larger plug lengths.

Fig. 23 compares the wall shear stress for both single fluid formulation [55] and the present two-fluid formulation carried at three different grid levels. The results at these grid levels are found to be consistent and agree with the reference study [55] except the location where the film thickness is at minimum. This difference at the peak values of

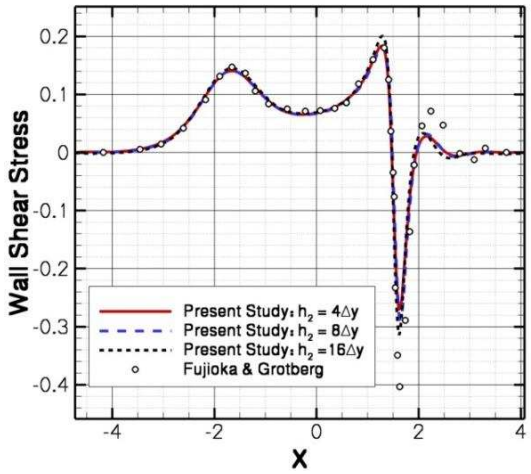


Figure 23: Steady state wall shear stress obtained by the present study and Fuijoka & Grotberg [50]. $Re=40$, $Ca=0.05$ and $L_P=2.0$.

the shear stress can be related to the assumption of constant pressure for the single fluid formulation because the corresponding region in two-fluid formulation variation in pressure inside the air-bubble, which, in return, effects the shape and the shear stress distribution on the wall.

To investigate the rupture dynamics, the flow is initiated using the steady state solutions obtained in the absence of the van der Waals attraction forces. These forces are the main mechanism of a reopening as they are reversely proportional with distance between the air bubbles. The liquid plug length corresponds to the film thickness in the computations of the van der Waals forces. Hence, the smaller the plug length, the more dominant they become over other forms of forces, i.e. surface tension. Eventually the liquid plug will rupture causing bubble interfaces to merge removing the airway blockage.

We consider the van der Waals attraction forces on the axisymmetric configuration for the plug length of $L_P=0.5$. The potential of the attraction forces at an interface location is defined based on the minimum distance to another fluid interface. This distance is not only used to compute the potential but also utilized to determine the time of the rupture, at which the film becomes unstable. Once the liquid plug length becomes smaller than a critical plug length, rupture is numerically initiated. The critical plug length is determined by the balance between the surface tension and the attraction forces using Eq. (2.31). The attraction forces defined by the Hamaker constant, A, is introduced to the computations as unity, whereas all other parameters including boundary conditions are kept the same as the steady state cases. According to these parameters, non-dimensional time and critical liquid plug length that would cause a rupture is given in Eq. (3.2).

$$t=\frac{A\rho}{\sigma\mu}t^*; \quad h_{critical}=\sqrt{\frac{A}{\sigma}}=0.22. \tag{3.2}$$

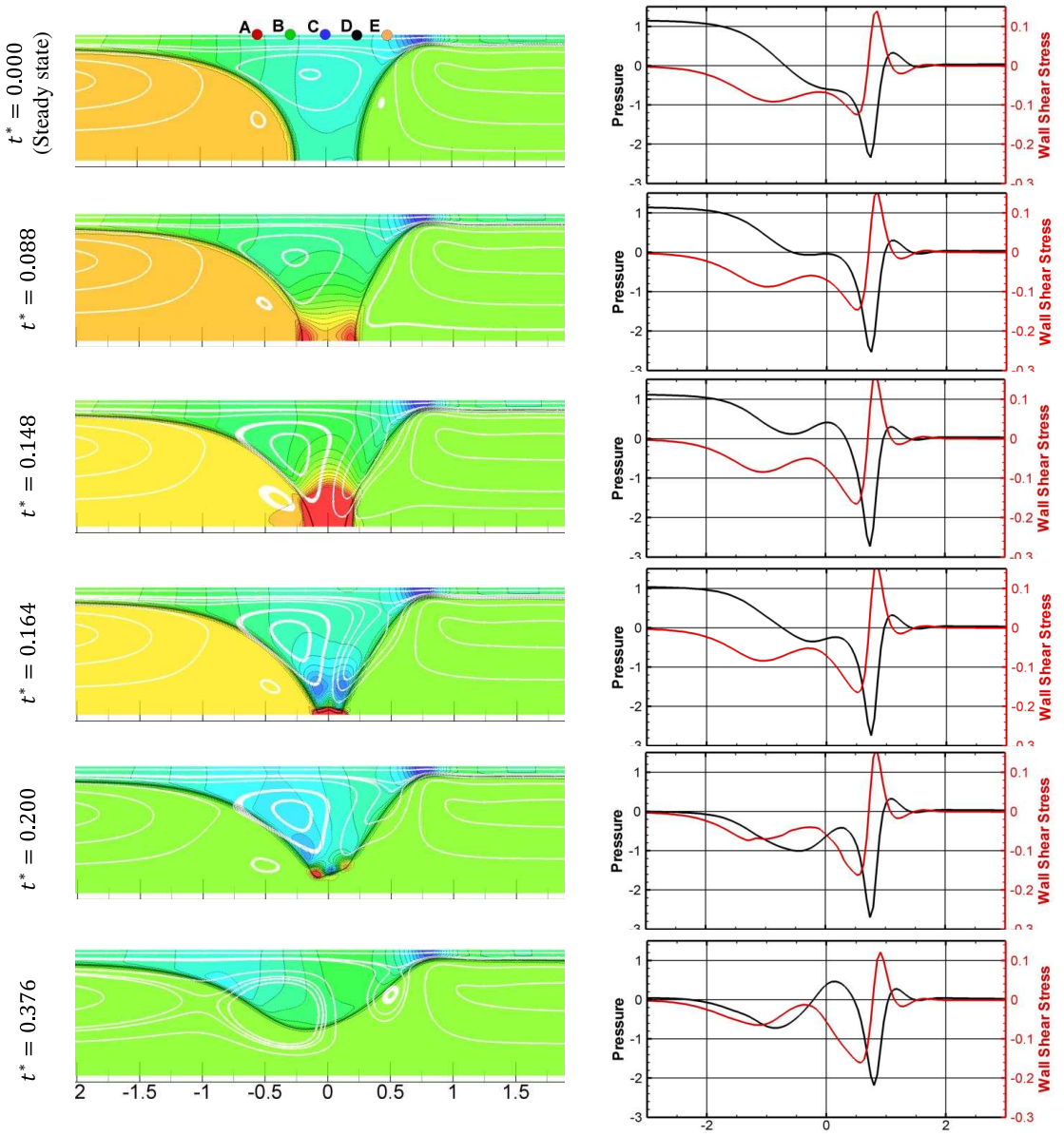


Figure 24: Time instants of the flow field during the reopening of the airway at $Re = 50$, $Ca = 0.05$ with an initial plug length of 0.5. (a) Pressure contours levels (color) and streamlines (white lines). (b) Wall pressure and wall shear stress.

Fig. 24 shows the time history of the pressure contours and streamlines before and after the rupture. The rupture is observed to happen between $t^* = 0.148$ and $t^* = 0.164$. Before rupture happens, the pressure starts to build up at the liquid plug as its length

kept decreasing. At the rupture time, $t^* \cong 0.16$, the length goes below the critical plug length and the interfaces experience a topological change, which causes a sudden change in pressure. Afterwards, the surface tension becomes locally dominant in this region, causing the freshly merged interface to move towards the channel walls. The sudden change in pressure level due to reopening is felt at the walls.

3.3 Spacecraft fuel tank dynamics

The dynamics of the liquid fuel contained in a tank is of interest especially for spacecraft applications. Considering micro-gravity conditions, capillary and viscous effects can become as significant as inertia effects, as opposed to normal gravity conditions. In such conditions, the flow characteristics may change rapidly in response to the spacecraft motion as a result of altered magnitude and direction of the body forces. These factors not only determine the amount of fuel delivered to the combustion chamber but also influence the spacecraft dynamics because the flow motion in the tank can disturb the dynamics of the vehicle by changing its center of mass [58].

3.3.1 Draining tank flow

The liquid fuel draining process depends on many factors, i.e. tank geometry, acceleration, surface tension, pressure difference applied and the material properties of the liquid fuel [14, 15]. The effects of these parameters have been studied experimentally [59, 60] under the effects of axial acceleration. When the acceleration or the gravitational force is aligned with axial axis of a symmetric fuel tank, the problem can be tacked in an axisymmetric formulation of the Navier-Stokes equations [14,15]. On the other hand, when the direction of the acceleration is different, the dynamics can only be captured using full three-dimensional approach.

The validity of the presented strategy in 3D configuration using previously reported axisymmetric results [14, 15]. Then, the draining dynamics at 45° tilted micro-gravity is investigated. Weber number, Bond number and Reynolds number are used to study the draining flow in a micro-gravity condition:

$$\text{Weber Number,}\quad We = \frac{\rho Q^2}{\pi^2 \sigma R^3}, \tag{3.3}$$

$$\text{Bond Number,}\quad Bo = \frac{We}{Fr} = \frac{\rho g R^2}{\sigma}, \tag{3.4}$$

$$\text{Reynolds Number,}\quad Re = \frac{\mu U(2R)}{\mu}. \tag{3.5}$$

In these equations, ρ is the density of the liquid, Q is the volume flow rate, σ is the surface tension of the interface between gas and liquid, g is the gravitational acceleration, and R is the characteristic length of the fuel tank, which is taken as the radius for cylindrical geometry in this study. Accordingly, non-dimensional time is defined as $t^* = tQ/\pi R^3$.

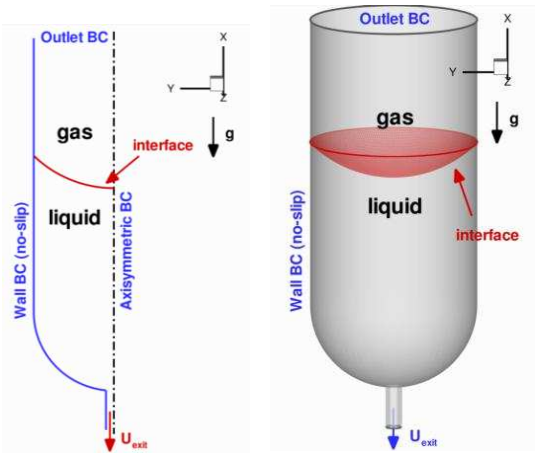


Figure 25: The computational geometry configuration of draining fuel tank with hemispherical bottom. (a) axisymmetric domain, (b) 3D computational domain.

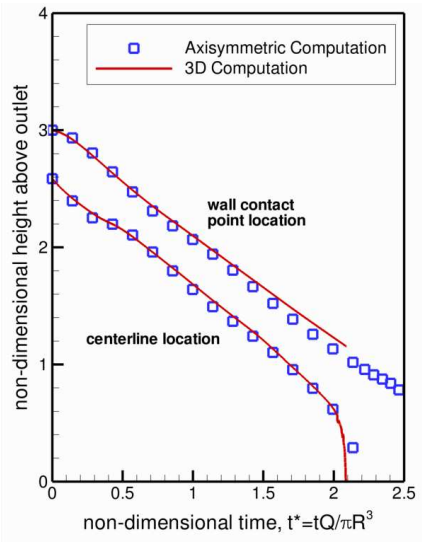
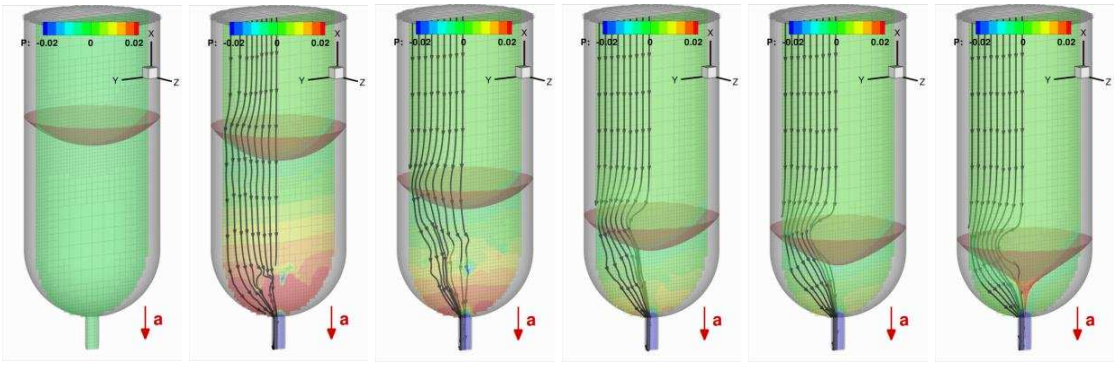


Figure 26: The comparison between axisymmetric and three dimensional computation for draining tank flow simulation. $Bo=5$, $We=0.97$, and $Re=216$.

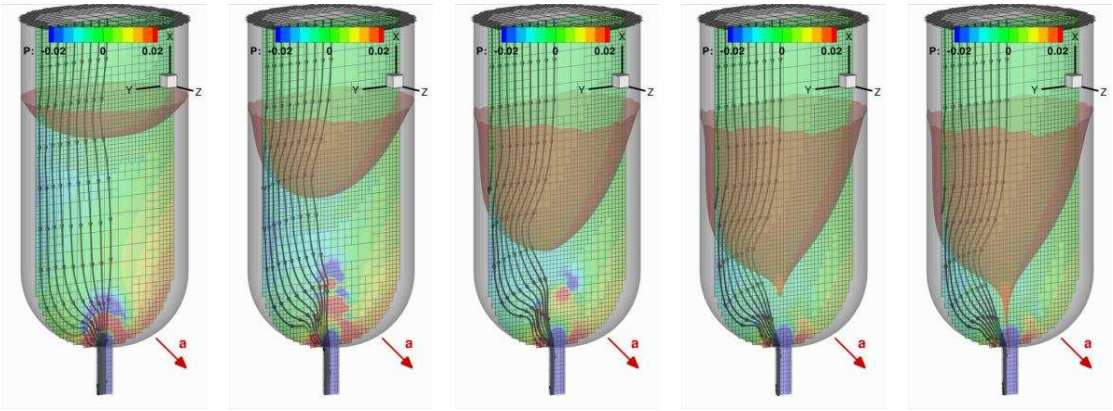
Fig. 25 shows the computational geometry configuration used for the verification of the three-dimensional computations. An axisymmetric fuel tank with a hemispherical bottom is considered to compare 3D computation with axisymmetric one. The initial fluid interface shape is assumed as circular arc with the given contact angle, 45°, at the tank wall. Bond number, Weber number, and Reynolds number are considered as 5, 0.97, and 216, respectively. Fig. 26 illustrates that non-dimensional centerline and wall attachment location of the interface for 3D computations are in agreement with the axisymmetric computations. The snapshots of adaptive grids and pressure contour for 3D computation are shown in Fig. 27, where maximum of 21 grid points are used along radius of tank. In spite of the fact that there are only 4 cells across the outlet pipe, the core dynamics of the draining process is captured.

To illustrate the 3D capabilities, the draining fuel tank simulation is conducted at a 45° tilted micro-gravity condition. The ratio between the outlet radius, r, and the tank radius, R, is 1/10, and the tank height is 4.0 times of tank radius. All simulations are conducted in a micro-gravitational environment with 1.5% of normal gravity. The initial fluid interface shape is assumed as a circular arc with the given contact angle, 45°, at the tank wall. Bond number, Weber number, and Reynolds number are considered as 5, 0.97, and 21.6, respectively. The computational geometry configuration is shown in Fig. 25. The complex 3D tank wall is represented by solid interface on a Cartesian computational grid. Fig. 28 shows the snapshots of adaptive grid and pressure contour plot. Similar to the axial gravity case, maximum of 21 grid points along the tank radius is employed, which corresponds to 4 grid cells per the diameter of the outlet pipe.



(a) $t^*=0.0$ (b) $t^*=0.2$ (c) $t^*=1.0$ (d) $t^*=1.6$ (e) $t^*=1.8$ (f) $t^*=2.0$

Figure 27: The snapshots of 3D liquid fuel draining flow under axial-direction micro-gravity. Grids are locally adapted around interface in time. Black lines represent streamlines and colored contour shows pressure distribution.

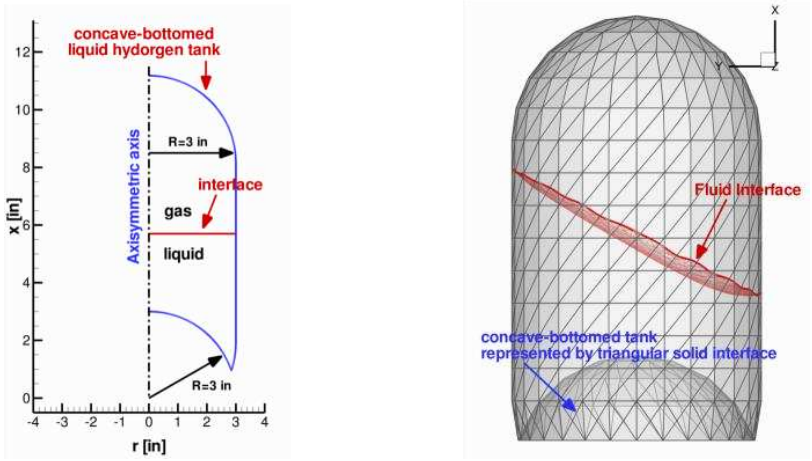


(a) $t^*=0.15$ (b) $t^*=0.60$ (c) $t^*=1.05$ (d) $t^*=1.28$ (e) $t^*=1.33$

Figure 28: The snapshots of 3D fuel tank draining with 45° tilted gravitational acceleration. The magnitude of acceleration is $0.015g_0$, and 45° contact angle is applied on the wall. ($Bo=5$, $We=0.97$, and $Re=21.6$).

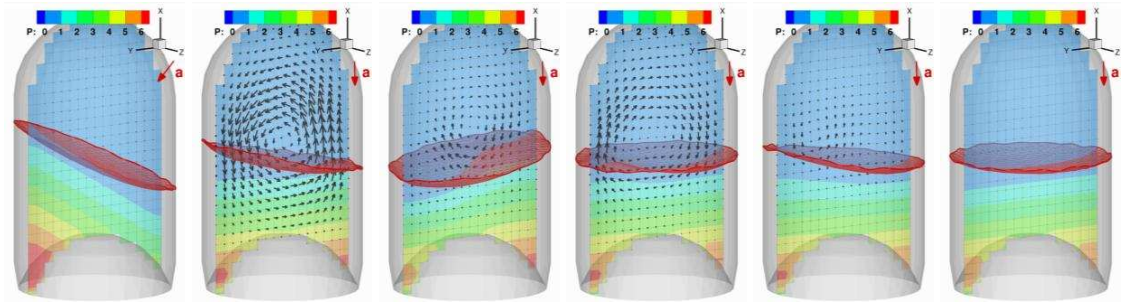
3.3.2 loshing liquid motion in a spacecraft fuel tank

The sloshing waves on the liquid fuel surface can influence the fuel draining dynamics affecting vapor ingestion and liquid fuel residuals. In addition, it can change the whole spacecraft dynamics due to the movement of center of mass of the liquid fuel during landing and/or docking maneuver. At normal or higher gravitational acceleration during powered flight, the fluid motion is remained damped, and shows moderate flow motion. On the other hand, even smaller potential energy change can cause large movements under micro-gravity conditions. For example, the sudden engine cutoff transforms the potential energy of liquid at a higher acceleration into kinetic energy, and thus, large sloshing fluid motion. Understanding the flow motions in a liquid fuel tank, and its



(a) (b)

Figure 29: The computational geometry configuration of draining fuel tank with hemispherical bottom. (a) axisymmetric domain, (b) 3D computational domain.



(a) $t^*=0.0$ (b) $t^*=1.0$ (c) $t^*=2.2$ (d) $t^*=3.4$ (e) $t^*=4.2$ (f) $t^*=20.0$

Figure 30: Sloshing motion of liquid fuel in a concave-bottomed tank with a sudden change of acceleration direction. Velocity vector and colored pressure contour are represented.

influence on the spacecraft dynamics are crucial for spacecraft applications.

The present review considers the pure sloshing motion of a liquid fuel in a spacecraft tank. Fig. 29(a) shows the configuration of the scale model of the Saturn V/S-IVB liquid hydrogen tank experimented by Toole et al. [61] This complex concave-shaped tank is represented by the triangular solid interface elements in Fig. 29(b) for 3D computation on the stationary Cartesian grids. Fig. 30 shows the flow field and instantaneous interface shapes during a sloshing flow inside this particular geometry undergoing a sudden change of direction of acceleration at 1% of the earth gravity level. The applied acceleration is initially 45° tilted for the axis in Fig. 30(a), where, the deformed interface with 60° contact angle between liquid and gas are shown. At $t^*=0.0$, the applied acceleration is abruptly changed into axial direction, and the difference of potential energy makes slosh-

ing motion. Bond number, which is defined as the ratio of gravitational force to surface tension forces, $Bo = 24$ is used in this simulation.

4 Summary and conclusions

In this paper, we report our recent efforts in developing a multi-scale three-dimensional adaptive grid method for multiphase flow problems. The method utilizes marker points representing a triangulated surface for both tracking interfaces separating fluid constituents and arbitrarily shaped solid geometries. The transport equations are solved on the Cartesian grid with automated local grid adaptation to capture the flow features. The triangular mesh representing the Lagrangian framework is free to move on the Cartesian grid that represents the Eulerian framework. These components are coupled together to capture interfacial flow dynamics for various applications. In summary, the following key aspects offer added capabilities for handling 3D moving boundaries around complex solid boundaries:

(i) The marker-based data structure that allows contact lines on solids surfaces where surface tension forces on interfaces separating fluid phases, pressure, viscous stresses, as well as contact lines geometries between solid-liquid-gas phases are handled simultaneously;

(ii) Utilizing markers to represent not only fluid-fluid interfaces but also solid boundaries to allow the Cartesian grid based computations to handle three- dimensional problems of engineering interests;

(iii) Incorporating the indicator function based on a distance formulation to distinguish materials, and aid cost of computations relevant to geometric calculations;

(iv) Inclusion of multiple physical mechanisms including surface tension, van der Waals effects in the interface region, along with the full Navier-Stokes equations.

Geometric computations involving 3D moving and intersecting surfaces can be very time consuming. Using an indicator function for these computations provides an efficient approach for such problems. This is analogous to the level-contour-based reconstruction during a topological change. The methods utilized in this study shows that this approach can also be extended to the moving surfaces by checking the indicator functions on the marker points to identify the intersection points.

To highlight these features, the following case studies are presented:

(i) Capillary flows with various Laplace numbers and contact conditions.

(ii) Spacecraft fuel tank flow problems with gravitational effects and varied control parameters including the capillary and Bond numbers.

(iii) The liquid plug problem, motivated by issues associated by drug delivery into the lungs, is investigated using the present two-fluid formulation. The combined surface tension and van der Waals effects, interacting with different plug lengths and flow parameters, leading to rich physical phenomena including topological changes and distinct flow characteristics.

Acknowledgments

The work reported in this paper has been partially supported by NASA Constellation University Institutes Program (CUIP), Claudia Meyer and Jeff Rybak program managers. We have benefitted from communication with Jim Grotberg and Hideki Fujioka of the University of Michigan while investigating the liquid plug flow problems.

References

[1] E. Uzgoren, R. Singh, J. Sim and W. Shyy, Computational modeling for multiphase flows with spacecraft application, Prog. Aerosp. Sci., 43 (2007), 138-192.
[2] W. Shyy, H.S. Udaykumar, M.M. Rao and R.W. Smith, Computational Fluid Dynamics with Moving Boundaries, Taylor & Francis, Washington, DC, (1996, revised printing 1997, 1998 & 2001), New York: Dover, 2007.
[3] S. Osher and R.P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2002.
[4] R. Scardovelli and S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech., 31(1) (1999), 567-603.
[5] G. Tryggvason, B. Bunner, A. Esmaeeli, N. Al-Rawahi, W. Tauber, J. Han, Y.J. Jan, D. Juric and S. Nas, A front-tracking method for the computations of multiphase flow, J. Comput. Phys., 169(2) (2001), 708-759.
[6] G. Tryggvason and A. Prosperetti (Eds), Computational Methods for Multiphase Flow, Cambridge University Press, 2007.
[7] V.M. Wasekar and R.M. Manglik, Short-time-transient surfactant dynamics and Marangoni convection around boiling nuclei, J. Heat Trans., 125(5) (2003), 858-866.
[8] B. Perot and R. Nallapati, A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows, J. Comput. Phys., 184(1) (2003), 192-214.
[9] F. Losasso, R. Fedkiw and S. Osher, Spatially adaptive techniques for level set methods and incompressible flow, Comput. Fluids, 35 (2006), 995-1010.
[10] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian and M.W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, J. Comput. Phys., 213(1) (2006), 141-173.
[11] D. Enright, F. Losasso and R. Fedkiw, A fast and accurate semi-Lagrangian particle level set method, Comput. Struct., 83(6-7) (2005), 479-490.
[12] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta and R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, J. Comput. Phys., 221(2) (2007), 469-505.
[13] C.H. Chang and M.S. Liou, A robust and accurate approach to computing compressible multiphase flow: Stratified flow model and AUSM+-up scheme, J. Comput. Phys., 225(1) (2007), 840-873.
[14] E. Uzgoren, J. Sim and W. Shyy, Computations of multiphase fluid flows using marker-based adaptive, multilevel Cartesian grid method, in: 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007, Reno, NV.
[15] E. Uzgoren, R. Singh, J. Sim and W. Shyy, A unified adaptive cartesian grid method for solid-multiphase fluid dynamics with moving boundaries, in: 18th AIAA Computational Fluid Dynamics Conference, 2007, Miami, FL.
[16] J. Glimm, M.J. Graham, J. Grove, X.L. Li, T.M. Smith, D. Tan, F. Tangerman and Q. Zhang, Front tracking in two and three dimensions, Comput. Math. Appl, 35(7) (1998), 1-11.

[17] J. Glimm, J.W. Grove, X.L. Li and D.C. Tan, Robust computational algorithms for dynamic interface tracking in three dimensions, SIAM J. Sci. Comput., 21(6) (1999), 2240-2256.
[18] R. Singh and W. Shyy, Three-dimensional adaptive Cartesian grid method with conservative interface restructuring and reconstruction, J. Comput. Phys., 224 (2007), 150-167.
[19] W. Shyy, Multiphase computations using sharp and continuous interface techniques for micro-gravity applications, Comptes Rendus, Mecanique 332(5-6) (2004), 375-386.
[20] C.S. Peskin, The immersed boundary method, Acta Numer., 11 (2003), 479-517.
[21] M. Francois and W. Shyy, Computations of drop dynamics with the immersed boundary method, Part 1: Numerical algoritm and buoyancy-induced effect, Numer. Heat Transfer: Part B: Fundamentals, 44(2) (2003), 101-118.
[22] T. Ye, R. Mittal, H.S. Udaykumar and W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys., 156(2) (1999), 209-240.
[23] R.J. Leveque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal., 31(4) (1994), 1019-1044.
[24] Z. Li and M.C. Lai, The immersed interface method for the navier-stokes equations with singular forces, J. Comput. Phys., 171(2) (2001), 822-842.
[25] X.D. Liu, R.P. Fedkiw and M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys., 160(1) (2000), 151-178.
[26] M. Kang, R.P. Fedkiw and X.D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput., 15(3) (2000), 323-360.
[27] S. Marella, S. Krishnan, H. Liu and H.S. Udaykumar, Sharp interface Cartesian grid method I: An easily implemented technique for 3D moving boundary computations, J. Comput. Phys., 210(1) (2005), 1-31.
[28] H.S. Udaykumar, , R. Mittal, P. Rampunggoon and A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, J. Comput. Phys., 174(1) (2001), 345-380.
[29] J.S. Sachdev and C.P.T. Groth, A mesh adjustment scheme for embedded boundaries, Commun. Comput. Phys., 2 (2007), 1095-1124.
[30] A. Gilmanov and F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies, J. Comput. Phys., 207(2) (2005). 457-492.
[31] J. Yang and E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, J. Comput. Phys., 215(1) (2006), 12-40.
[32] M.J. Aftosmis, Solution adaptive Cartesian grid methods for aerodynamic flows with complex geometries, von Karman Institute for Fluid Dynamics, Lecture notes, 2 (1997), A1-A105.
[33] S. Shin, S.I. Abdel-Khalik and D. Juric, Direct three-dimensional numerical simulation of nucleate boiling using the level contour reconstruction method, Int. J. Multiphase Flow, 31(10-11) (2005), 1231-1242.
[34] D. Juric and G. Tryggvason, Computations of boiling flows, Int. J. Multiphase Flow, 24(3) (1998), 387-410.
[35] R. Singh, N. N'Dri, E. Uzgoren, W. Shyy and M. Garbey, Three-dimensional adaptive, Cartesian grid method for multiphase flow computations, in: 43rd AIAA Aerospace Sciences Meeting and Exhibit, 2005.
[36] R. Singh, Three-dimensional marker-based multiphase flow computation using adaptive cartesian grid techniques, in: Mechanical and Aerospace Engineering, University of Florida: Gainesville, 2006, p. 144.

[37] J.M. Stockie, Analysis and Computation of Immersed Boundaries, with Application to Pulp Fibres, PhD Thesis, University of British Columbia, 1997.
[38] B. Engquist, A.K. Tornberg and R. Tsai, Discretization of Dirac delta functions in level set methods, J. Comput. Phys., 207(1) (2005), 28-51.
[39] R.K. Singh, J. Chao, M. Popescu, C.-F. Tai, R. Mei and W. Shyy, Multiphase/multidomain computations using continuum and lattice Boltzmann methods, ASCE J. Aerospace Eng., 19 (2006), 288-295.
[40] Z. Wang and Z.J. Wang, The level set method on adaptive Cartesian grid for interface capturing, AIAA Paper, 82, 2004.
[41] M. Francois and W. Shyy, Micro-scale drop dynamics for heat transfer enhancement, Prog. Aerosp. Sci., 38(4) (2002), 275-304.
[42] F.S. de Sousa, N. Mangiavacchi, L.G. Nonato, A. Castelo, M.F. Tomé, V.G. Ferreira, J.A. Cuminato and S. McKee, A front-tracking/front-capturing method for the simulation of 3D multifluid flows with free surfaces, J. Comput. Phys., 198(2) (2004), 469-499.
[43] N. Al-Rawahi and G. Tryggvason, Numerical simulation of dendritic solidification with convection: Two-dimensional geometry, J. Comput. Phys., 180(2) (2002), 471-496.
[44] N. Al-Rawahi and G. Tryggvason, Numerical simulation of dendritic solidification with convection: Three-dimensional flow, J. Comput. Phys., 194(2) (2004), 677-696.
[45] S. Shin and D. Juric, Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity, J. Comput. Phys., 180(2) (2002), 427-470.
[46] S. Shin, S.I. Abdel-Khalik, V. Daru and D. Juric, Accurate representation of surface tension using the level contour reconstruction method, J. Comput. Phys., 203(2) (2005), 493-516.
[47] T. Erneux and S.H. Davis, Nonlinear rupture of free films, Phys. Fluids A: Fluid Dynamics, 5 (1993), 1117.
[48] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, McGraw-Hill, 1980.
[49] W. Shyy, Computational Modeling for Fluid Flow and Interfacial Transport, Elsevier, New York, 1994.
[50] V.E.B. Dussan, The moving contact line: slip boundary conditions, J. Fluid Mech., 77 (1977), 665-684.
[51] L.M. Hocking, A moving interface on a rough surface, J. Fluid Mech., 76 (1976), 801-817.
[52] H. Huang, D. Lian and B. Wetton, Computation of a moving drop/bubble on a solid surface using a front tracking method, Commun. Math. Phys., 2(4) (2004), 535-552.
[53] E. Uzgoren, J. Sim and W. Shyy, Marker-based, 3D adaptive cartesian grid method for multiphase flow around irregular geometries, in: 45th AIAA Aerospace Sciences Meeting and Exhibit, Nevada: Reno, 2008.
[54] H. Fujioka and J.B. Grotberg, Steady liquid plug flow in a 2-D channel, Faseb J., 17(4) (2003), A87-A87.
[55] H. Fujioka and J.B. Grotberg, Steady propagation of a liquid plug in a two-dimensional channel, J. Biomech. Eng., 126 (2004), 567-577.
[56] Y. Zheng, H. Fujioka, J.C. Grotberg and J.B. Grotberg, Effects of inertia and gravity on liquid plug splitting at a bifurcation, J. Biomech. Eng., 128 (2006), 707.
[57] X. Jiang and A.J. James, Numerical simulation of the head-on collision of two equal-sized drops with van der Waals forces, J. Eng. Math., 59(1) (2007), 99-121.
[58] A.E.P. Veldman, J. Gerrits, R. Luppes, J.A. Helder and J.P.B. Vreeburg, The numerical simulation of liquid sloshing on board spacecraft, J. Comput. Phys., 224(1) (2007), 82-99.
[59] E.P. Symons, Draining characteristics of hemispherically bottomed cylinders in a low-

gravity environment, NASA, Technical Paper 1297, 1978.
[60] S.G. Berenyi, Vapor ingestion phenomenon in hemispherically bottomed tanks in normal gravity and in weightlessness, NASA, Report No TN D-5704, 1970.
[61] L.E. Toole and L.J. Hasting, An experimental study of the behavior of a sloshing liquid subjected to a sudden reduction in acceleration, NASA, 1968.