45th AIAA Aerospace Sciences Meeting and Exhibit
8 - 11 January 2007, Reno, Nevada

AIAA 2007-336

Computations of Multiphase Fluid Flows Using Marker-Based Adaptive, Multilevel Cartesian Grid Method

Eray Uzgoren[*], Jaeheon Sim[†], Wei Shyy[‡]

Department of Aerospace Engineering
University of Michigan, Ann Arbor, MI 48109 USA

Addressing the dynamics of multiphase fronts is crucial for successful practices of many engineering applications that involve both gas and liquid constituents. Numerical simulations of such flows need to resolve the location and the shape of the front, where interfacial conditions are satisfied as part of the solution. In this work, immersed boundary method on adaptively refined Cartesian grid is utilized to capture the dynamics of the multiphase front. Marker based interface representation is extended to handle the complex solid geometries that exist in the flow field. The entire computational algorithm is applied to simulate steady state and time dependent liquid plug problem and assessed against existing theoretical analyses. It is then used to simulate time dependent draining flow problems, motivated by spacecraft fuel tank operations where experimental guidance is available. Distinct interfacial characteristics and fluid physics associated with different flow regimes are highlighted.

I. Introduction

FLOWS involving interactions between liquids and gases can be observed in a wide range of engineering applications. Some examples include the applications of drug delivery through the respiratory system or the design of a fuel tank of a space shuttle. Despite the diversity in their application areas, the common nature of both problems is the existence of a multiphase front between the liquid and the gas state, which may move in response to the flow dynamics. The interplay between the interface and the flow dynamics determines the outcome of the application mechanics.

Numerical simulations of interfacial flows are required to resolve the location of the front to handle, which arise as a result of surface tension forces and distinctive material properties of the constituents. Techniques that involve in front tracking are grouped into three main categories, namely the Lagrangian, the Eulerian, and the combined Eulerian-Lagrangian methods[1]. In Lagrangian methods, the interface location is tracked explicitly by advecting and deforming the computational grid with the moving interface[2,3,4]. On the other hand, Eulerian methods such as volume of fluids[5,6] and level set[7,8] methods employ a scalar function to extract the interface location on stationary grids. The marker based front tracking algorithm falls into the mixed Eulerian-Lagrangian category as it tracks the interface using set of markers moving on a stationary computational grid[9,10,11,12,13].

One of the primary characteristics of multiphase flows is that pressure and viscous stresses show discontinuities across the interface as a result of the surface tension forces. These jump conditions are incorporated with flow computations via models which can be grouped into two distinguished methods; sharp interface methods (SIM), and continuous interface methods (CIM). SIM satisfies the jump condition across interface explicitly and thus, gives more accurate solution[9,14,15]. CIM smoothes the fluid properties and surface tension forces across interface, and solves a single set of equations for the entire domain[5,6,7,8,10,11,12]. The present study considers CIM due to the concerns related to computational efficiency and relative difficulties with sharp interface methods via marker based tracking. Specifically, overall method is referred to immersed boundary method (IBM)[10,11,12,13].

[*] Post Doctoral Research Fellow, Member AIAA.

[†] Ph.D. Student.

[‡] Clarence L. “Kelly” Johnson Collegiate Professor and Chair, Fellow AIAA.

In recent years, interface tracking via triangulated elements have improved its most obvious drawback of algorithmic complexity for the problems in which topological changes occur. This is achieved by means of level-contour construction algorithm without the need of the connectivity information[16]. On the other hand, keeping the connectivity information helps to reduce the computational effort as the reconstruction algorithm can be applied locally where it is required[12]. Additional algorithmic improvements for front tracking are required for handling contact line problems, in which liquid and solid phases meet at the computational boundaries[17], and for handling infinitely long bubbles at the inflow/outflow regions. The behavior of the multiphase front at such conditions can be modeled to capture the multiphase front dynamics.

Practices of fluid flow simulations oftentimes require capabilities of handling complex geometries within the flow field. The marker based data structure which has already been defined for tracking the multiphase fronts can be utilized to apply the prescribed conditions on the embedded boundary. Consequential technique falls into the category of non-boundary conforming techniques, which are relatively easier to generalize, computationally efficient, and can be used along with Cartesian grids. Yang and Balaras[18] demonstrated its capabilities via validation studies including the laminar flow induced by oscillating cylinder, a flow over a cylinder and turbulent flow (modeled by LES) over a traveling wavy wall. A similar approach is utilized for the numerical simulations of flapping wings and insect flight[19].

Additions of each of these components stress on the one of the most obvious challenges in developing physical numerical tools for multiphase flow simulations: the cost of computations. Mechanisms including convection, diffusion, pressure, body force and interfacial conditions create multiple time/length scales, which can amplify the resolution requirements to impracticable levels due to the limitations in computational resources. The present work considers two remedies to overcome such difficulties, adaptive grid refinement and multi-domain techniques for parallel computing purposes. Adaptive grid refinement optimizes the computational data size while maintaining a desirable quality solution by resolving the required length scales to a desired level of resolution, which can be controlled locally[20]. This refinement procedure is utilized on Cartesian grids due to the nature of uncomplicated grid generation and data-structure.

Large discontinues in the transport properties results in stiff linear systems. The convergence characteristics of solvers are usually different when they are utilized in parallel computations. Multi-domain methods provide a technique that ameliorates such difficulties by handling the linear system in smaller subsets of the overall computational domain. Its success on parallel environment is based on its capability of reducing the frequency of the data exchange between the sub-regions while maintaining a reasonable convergence rate. The objectives of a multi-domain method include a balanced distribution of computational load, minimal communication between processors, and minimal number of iterations for convergence. The unstructured nature of adaptively refined grids requires a partitioning algorithm to meet these objectives, which can be attained using a cost-effective strategy via space filling curves[21].

The present study employs the immersed boundary method, which uses marker points to track the interface location, and continuous interface methods to model interfacial conditions. The large transport property jumps across the interface, the resulting computational stiffness, and moving boundaries are treated on adaptively refined 3D Cartesian grid. A domain decomposition method and a partitioning strategy for adaptively refined grid are developed to enable parallel computing capabilities. Specifically, the approach consists of multilevel additive Schwarz method for domain decomposition, and Hilbert space filling curve ordering for partitioning. The issues related to load balancing, communication and computation, convergence rate of the iterative solver in regard to grid size and the number of sub-domains, and interface shape deformation are studied. Furthermore, interfacial representation using marker points is extended to model complex solid geometries for single and two-phase flows. Finally, we probe the numerical accuracy of the combined computational capabilities, and simulate the front propagation associated with the liquid plug problem, and draining fluid flow related to spacecraft fuel tank operations.

II. Numerical Methods

A. Immersed Boundary Method

The immersed boundary method (IBM) is employed to track the interface location and compute the surface tension forces, which are smeared around the interface over a finite thickness. The main strength of IBM lies in its ability to allow simulations to be carried out using a single set of equations for the entire flow field. The interface is explicitly tracked using marker points. The governing equations of the flow are solved on the background adaptive Cartesian grid.

The incompressible Navier-Stokes equations for mass and momentum conservation are given in Eqs. (1) and (2).

$$\nabla \cdot u = 0 \tag{1}$$

$$\frac{\partial \rho u}{\partial t} + \nabla \cdot \left(\rho uu \right) = -\nabla p + \nabla \cdot \left(\mu \nabla u + \mu \nabla^T u \right) + F_s + \rho g \tag{2}$$

where u is the velocity vector, p is the pressure, ρ is the density, μ is the viscosity, and F_s is the source term, which includes the interfacial forces arising from the surface tension (σ) and the curvature (κ) as shown in Eq. (3).

$$F_s = \int_S \sigma \kappa \hat{n} \delta \left(\mathrm{x} - \mathrm{X} \right) dS \tag{3}$$

The surface force in Eq. (3) is computed using the Lagrangian marker points (X) and is translated into an Eulerian quantity (x) via the approximate discrete Dirac delta function ($\delta_h \left(r \right)$) in Eq. (4), in which r is the distance between the marker point and the cell center scaled by the grid spacing.

$$\delta_h(r) = \begin{cases} \frac{1}{8}\left(5 - 2|r| - \sqrt{-7 + 12|r| - 4|r|^2} \right), & 1 \le |r| \le 2 \\ \frac{1}{8}\left(3 - 2|r| + \sqrt{1 + 4|r| - 4|r|^2} \right), & 0 \le |r| \le 1 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Approximate Dirac delta function is also used for obtaining the marker velocity field to move marker points for obtaining the new geometric surface representation. As the interface evolves, markers are added or deleted to maintain interface resolution. Capabilities of handling topological changes can be included with reconstruction algorithms[16].

$$\mathrm{u}^n(\mathrm{X}) = \int_v \mathrm{u}^n \left(\mathrm{x} \right) \delta_h \left(\mathrm{x} - \mathrm{X} \right) dv \tag{5}$$

The material properties, i.e. density and viscosity, are discontinuous across the interface. The variation needs to be smoothed in order to facilitate a single fluid formulation. An indicator function, *I*, that varies smoothly from a value of unity outside the interface to zero inside the interface is employed for this purpose. It is computed on the background grid using the Dirac delta function by Eq.(6).

$$\nabla^2 I \left(x \right) = \nabla \cdot \left(\int_s \hat{n} \delta \left(\mathrm{x} - \mathrm{X} \right) dS \right) \tag{6}$$

The transport properties are smeared across the interface using Eq.(7).

$$\beta = \beta_2 + (\beta_1 - \beta_2) I \tag{7}$$

The term β represents a material property such as density or viscosity. The subscripts 1 and 2 represent fluids outside and inside the body respectively.

B. Modeling Complex Solid Boundaries

The present work adopts the technique described by Yang and Balaras[18] on adaptively refined Cartesian grids. This approach employs the prescribed boundary conditions on an arbitrary interface by reconstructing a velocity field at cells near the interface. This process doesn't involve smearing and can be seen as a sharp interface method.

In this approach, cells around a solid body are marked as forcing, pseudo-fluid and solid cells. Forcing points are used for defining the imposed boundary conditions on the flow field, whereas pseudo-fluid points are required in case solid body can freely move. Solid points are turned off for velocity and pressure field computations. As the flow field can have multiple fluid components and complex solid bodies, a function of *I* smoothly varying from zero to two is formed using Eq. (6). Certain ranges are marked to indicate if the cell lies in the vicinity of a different constituent than the bulk material. This is illustrated in Fig. 1 with the ranges used in the computations.

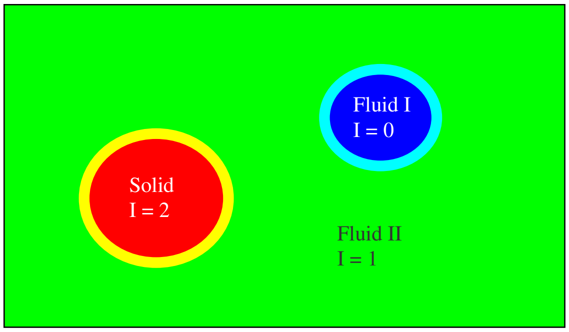


Figure 1. Indicator function varying smoothly from zero to two.

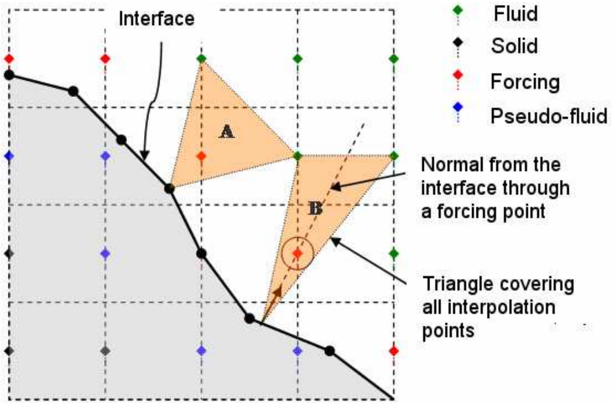

Interface
Fluid
Solid
Forcing
Pseudo-fluid
A
B
Normal from the interface through a forcing point
Triangle covering all interpolation points

Figure 2. Definition of faces around the solid interface for u-velocity.

The velocity field is constructed at the forcing points to yield the boundary condition defined on the solid boundary. Forcing points are defined on the fluid phase side, which have at least one neighboring point inside the solid interface. The velocity components are defined at the faces of a cell.

Figure 2 shows a representation for this flagging process on X-oriented faces. The triangle shaded areas represent the points considered for reconstruction of forcing point velocities. The construction involves an interpolation schema that uses one point on the interface, which does not necessarily have to be a marker point, and two points from the liquid faces. The possible interpolation schemas are presented by triangular shaded regions. Schema “A” involves in two faces that are closest to the construction point[18], whereas schema “B” uses a probe in the normal direction that passes through the construction point to utilize a bilinear interpolation[19].

The present interpolation procedure employs scheme “A”, which assumes linear variation of any variable ϕ. In Eqs. (8) and (9), x_i, y_i represents the corners of the triangle presented in Fig. 2. For stationary objects, the coefficients can be obtained once and can then be used for reconstructing the velocity field at each time step. On the other hand, the system has to 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.

$$\phi = b_1 + b_2 x + b_3 y \tag{8}$$

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \tag{9}$$

C. Adaptive Grid Based Computations

Immersed boundary method solves a single set of equations for all the constituents, which brings rapid variations to flow variables and material properties around the interface. Resolving the whole domain to the desired length scale may result in unnecessary resolutions at regions far from the interface. This would require an enormous amount of memory and computational power especially for large-scale simulations. One remedy is to employ adaptive grid refinement, which offers a way to adjust the grid sizes to the desired length scale locally.

An isotropically adaptive Cartesian grid is employed for solution of the flow governing equation. The grid is initialized in a uniform structure which has prescribed number of cells in each coordinate direction. The cell that needs to be refined is split into four smaller and equal cells. This procedure brings additional levels of grid in which new coordinates in computational space is assigned to the level that cells reside. To ensure a smooth variation in grid cell-size for quality and simplicity, cells sharing a face are not allowed to differ by more than one level of refinement. This procedure is called cell-by-cell adaptation and illustrated in Fig. 3. The implementation details are described in the work Singh and Shyy[12].

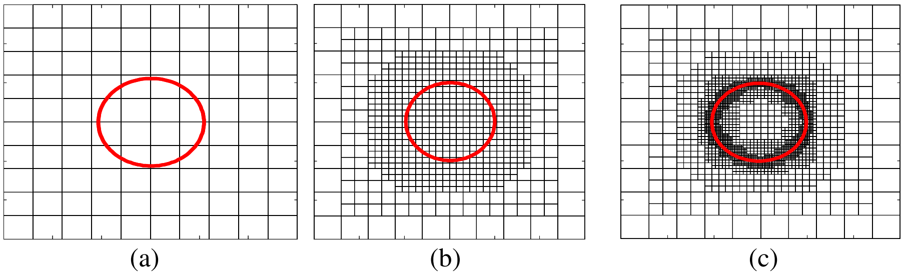


Figure 3. Steps of interface location based adaptation with 4~5 layers of cells across the interface refined uniformly, (a) Initial 10x10 base grid, (b) grid after two level adaptation, (c) grid after four levels adaptation.

The fractional step method, originally developed by Chorin[22], is used to solve the governing equations as it offers efficient solutions to the Navier-Stokes equations for incompressible flows. The fractional step method decouples the pressure field by integrating an incomplete form of the momentum equations. Equation (10) shows advection-diffusion equation for the original fractional step method.

$$\frac{\rho \mathrm{u}^{*} - \rho \mathrm{u}^{n}}{\Delta t} + \nabla \cdot \left(\rho \mathrm{uu} \right) = \nabla \cdot \left(\mu \nabla u \right) + F_{s} \tag{10}$$

Equation (10) is discretized using second order Runge-Kutta and Crank-Nicholson scheme for convection and viscous terms in order to obtain an approximate velocity field, which is usually not divergence free. The pressure field is solved using the continuity equation that is obtained by taking the divergence of the momentum equation considering only pressure term as shown in Eq. (11).

$$\frac{\nabla \cdot \rho \mathrm{u}^{*}}{\Delta t} = \nabla \cdot \left(\nabla p \right) \tag{11}$$

The intermediate velocity field is corrected using the pressure gradient to make the final velocity field divergence free (Eq. (12)).

$$\mathrm{u}^{n+1} = \mathrm{u}^{*} - \Delta t \nabla p \tag{12}$$

D. Multi-Domain Techniques for Pressure Poisson Equation

The linear system arising from the pressure Poisson Equation (Eq. (12)) has slow convergence properties, especially for high density ratios[23]. For efficient parallelism, the implementation accounts for a balanced load distribution, minimal communication between processors, and minimal number of iterations for convergence. As the present study employs adaptively refined Cartesian grids, achieving these requirements can be possible by employing domain decomposition method with a state-of-art partitioning algorithm.

Multi-domain methods offer a way to cope with these tradeoffs in parallel algorithms for solving linear systems. Its essence is to divide a large problem into smaller pieces, each of which is then solved independently before being combined to obtain the global solution of the original problem. In the literature, domain decomposition methods are recognized in two different classes; namely, Schur complement methods and Schwarz methods[24]. The present study employs Additive Schwarz method, which divides the computational domain into sub-regions, possibly overlapping. Each of these sub-regions forms subsystems of linear equations that are solved locally and they are then coupled with other sub-systems to obtain global solution. It couples sub-systems using successive exchange of Dirichlet conditions at the overlap regions.

The decision for choosing sub-problems for domain decomposition is referred to partitioning, which is a crucial ingredient of efficient parallel scientific computations. Once the decomposition is marked in the data, sub-problems are mapped onto a number of processors. For this purpose, space filling curve based ordering is used for mapping the physical space on a 1-D line, which is a special function with the property of locality[21].

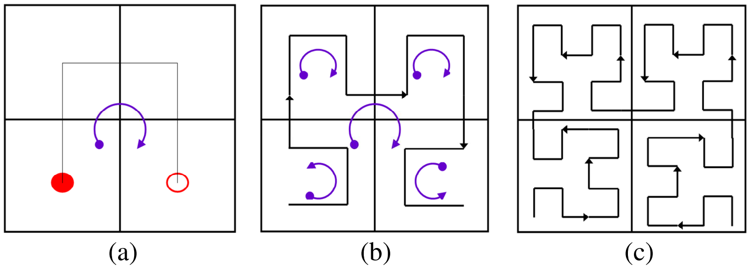


Figure 4. Construction of the Peano-Hilbert curve. (a) 1-order, (b) 2-order, (c) 3-order.

In two dimensions, the basic building block of the Hilbert curve is a "U" shaped line which visits each of 4 cells in a 2x2 block. Each subsequent level further divides the previous level's cells, creating sub-quadrants, visited by rotated "U" shaped curves. In three dimensions, the curves follow the same basic construction rules, but the basic building block extends to three-dimensions with additional "U" shaped turns. As the resolution increases, the curve fills the space. Consequently, it can be used for mapping n-dimensional space on a 1D line. Figure 4 illustrates the Hilbert curves on a unit domain with different resolution, which can be recursively defined by substituting a straight line segment by a certain pattern of lines. The curve fills the whole domain as this process is repeated infinitely. Repeating this sub-division for k times will yield in an approximation of a Hilbert curve of k-order, which indicates the resolution.

Algorithms for encoding and decoding work on a domain, which is either a unit-square in 2D or a unit-cube in 3D, can only be applied for cell-ordering on a computational domain with equal number of cells in each direction. Moreover, the number of cells is required to be multiples of 2^n. Otherwise, some cells would not conform to the 2^n nature of the space filling curve. In order to overcome such issues, the initial box grid, which is defined by the user as an input parameter, is assigned a base orientation and a base Hilbert key. Because the nature of the isotropic adaptation is similar to the Hilbert curve, which is based on division of a square/cube cell to four/eight sibling cells, the function for Hilbert curve can operate independently inside each of these boxes to create Hilbert keys depending on the refinement. During the cell-ordering, the independent Hilbert keys can be combined with the base values of orientation and resolution parameters to preserve the overall ordering. Figure 5 illustrates possible grid arrangements for channel flows with the initial and adapted curves. The initially set base values are inherited by the sibling cells during the adaptation.

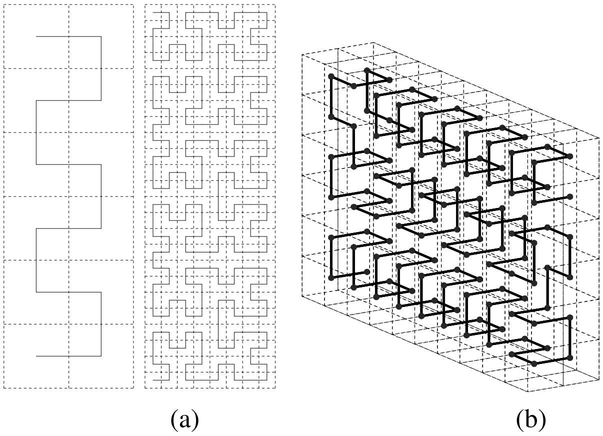


Figure 5. Initial cell-ordering for a computational domain with different number of cells assigned in different directions. (a) Two-dimensional domain, (b) Three-dimensional domain.

Figure 6 summarizes the partitioning process for grids with multiple levels. First, the order of the Hilbert curve is set to the maximum level that exists in the grid. This allows us to use the quick sort algorithm based on Hilbert keys, which has $N\log N$ operations. Then, the separation points are determined on the ordered cell list to include equal number of cells on every Hilbert curve segments. This is to ensure the computational load is distributed on uniformly on the sub-domains, which are determined by these segmented curves. A global[§] array is created in order to track the identification of the governing processor for each cell. This enables us to create overlapping cell/face lists to be utilized by the domain decomposition technique.

Hilbert curve based ordering is also utilized to aid geometric multigrid solver, which is a challenging task on unstructured grids. Multigrid is one of the best iterative linear solvers for elliptic equations[26,27]. Multigrid achieves such computational efficiency by damping high and low frequency error components at the same rate. This is an attractive property as the additive Schwarz algorithm has a strong dependence on the overlap grid size.

[§] The term "global" is referred to data available to all processors, whereas the term "local" is referred to data available only to the processor of ownership.

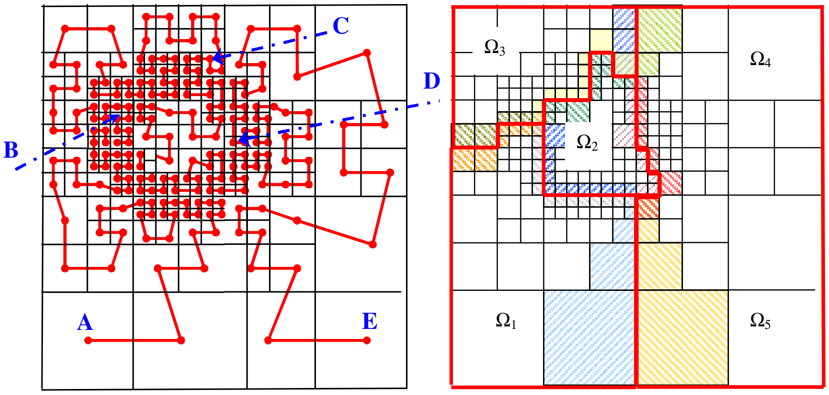


Figure 6. Partitioning based on cell-ordering. New partitions start at the capital letters.

The creation of coarser grids is handled by traveling on the ordered cells and combining four cells in 2D and eight cells in 3D together wherever possible. If all candidate cells to be combined are at the same level, all represent a coarser cell in the new level. This process is repeated until base uniform grid is reached unless desired number of levels is specified. Coarse grid generation is illustrated in Fig. 7.

Geometric based coefficient matrix assembly is based on conservation laws. Coefficient matrix is created and stored for all grid levels, which are updated once after grid adaptation. This avoids the computational cost of the matrix assembly during the time steps between adaptations at the cost of 30% increase in memory allocation. The coefficient matrix at this stage is in a template form involving only grid related information without the density field which changes with the location of the interface. Density field can be incorporated to the template during the solution of the pressure Poisson equation.

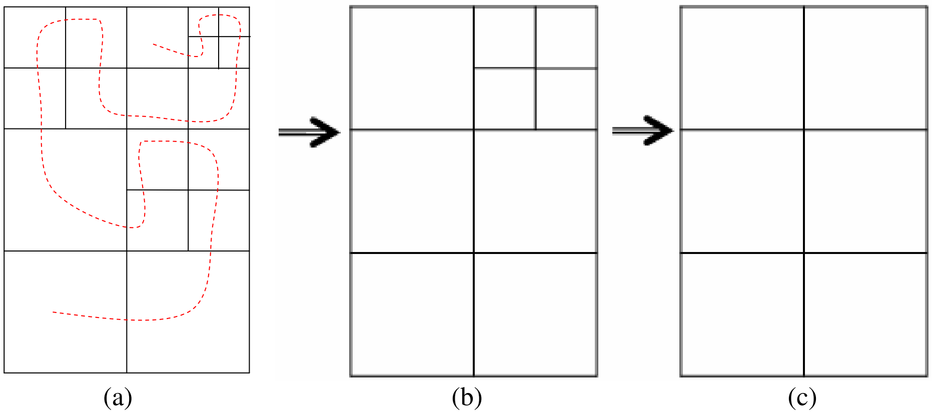


Figure 7. Results of the coarsening process. (a) Level = 2 illustrated with Hilbert cell-ordering, (b) Level = 1, (c) Level = 0 (initial box grid).

Restriction operator, *R*, relates the variables that are at a fine level, *f*, to a coarser level, *c*. On the other hand, prolongation operator, *P*, relates the variables that are at a coarse level, *c*, to a finer level, *f*. They are utilized in the process through the following relations:

$$R^C = \frac{1}{4}\left(R_1^f + R_2^f + R_3^f + R_4^f\right) \tag{13}$$

$$P_1^f = P_2^f = P_3^f = P_4^f = P^C \tag{14}$$

The multigrid algorithm is recursively defined and accepts three parameters, γ, μ_1, and μ_2 which sets the nature of the cycle as V-cycle or W-cycle. μ_1 and μ_2 denote the number of sweeps for pre-smoothing and post-smoothing, respectively. When γ*=1*, $\mu_1 = \mu_2 > 0$, multigrid follows a V-Cycle and when γ*=2*, $\mu_1 = \mu_2 > 0$, it follows a W-cycle.

III. Results

A. Multigrid Solver

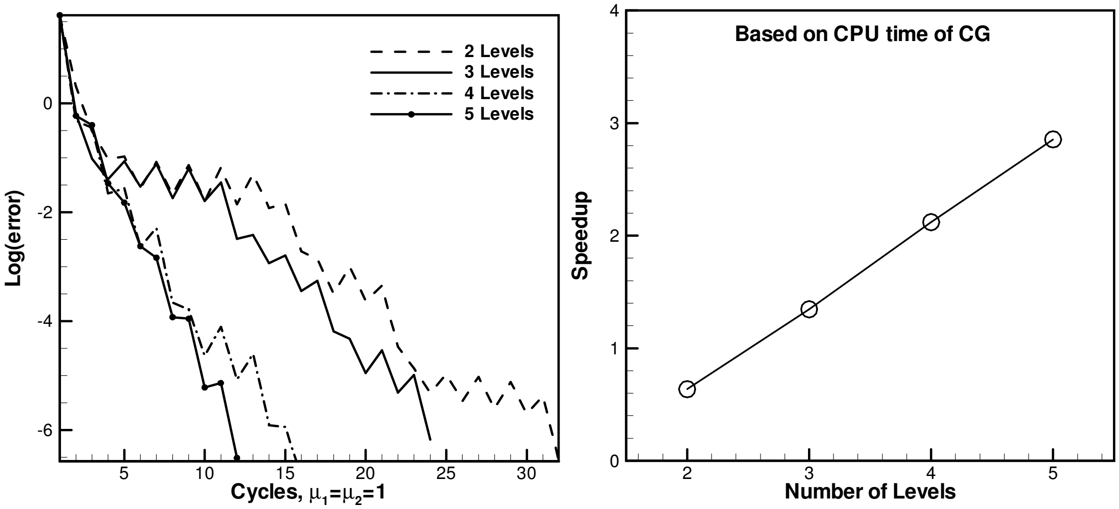
The developed multigrid solver is compared with the conjugate gradient solver, which is another efficient solver widely used for sparse and unstructured linear systems. The performance comparison is performed on the Poisson equation as described in Eq. (15).

$$\nabla \cdot \left(\frac{\nabla p}{\rho} \right) = f\left(x, y\right)$$
$$f\left(x, y\right) = -2\pi^2 \cos\left(\pi x\right) \cos\left(\pi y\right) \tag{15}$$

Equation (16) states the boundary conditions applied to Eq. (15),

$$f(-0.5, y) = f(0.5, y) = f(x, -0.5) = f(x, 0.5) = 0 \tag{16}$$

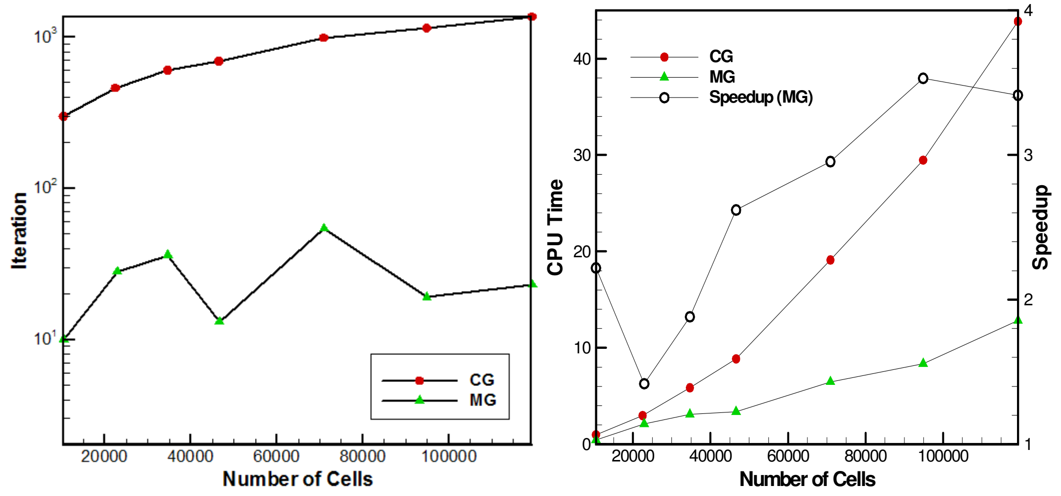
The multigrid solver's performance is first tested on the uniform grids, in which all the cells in the grid can be coarsened. Figure 8 illustrates that the multigrid performance increases with the number of levels employed. Furthermore, the computation time, scaled with respect to the total time spent by the conjugate gradient algorithm, is found to be in favor of the multigrid solver. Only the case of multigrid with two levels is found to be slower than CG, while a speedup of three is achieved when all the available levels are employed by the algorithm.



(a) (b)

Figure 8. Performance of multigrid when different number of levels is employed. (a) Iteration history, (b) Speedup based on the CPU-time of the conjugate gradient algorithm.

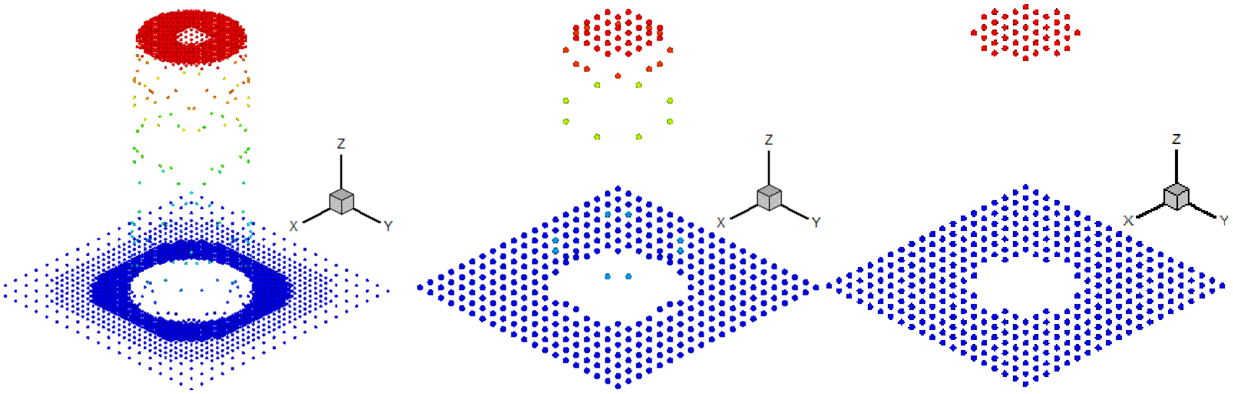
On the other hand, in adaptively refined grids, coarsening process only includes those cells that can be coarsened in adaptively refined grids. Figure 9 shows the performance of conjugate gradient and multigrid solvers on adaptively refined grids. Grid size is modified by changing the base level grid size while keeping the level of refinement in order to employ same number of levels in the multigrid solver. Total iteration number to reduce the maximum error to a tolerence limit is observed to remain in the same order for multigrid while it is found to increase linearly for the conjugate gradient algorithm. Similar to the uniform grid case, the computational cost of multigrid algorithm is less than conjugate gradient solver for all grid sizes and the speedup values vary between 1.5 and 3.4.



(a) (b)

Figure 9. Performance of multigrid and conjugate gradient algorithms on non-uniform grids. (a) Iteration number. (b) Computational time and speedup.

The density variation in the flow field affects the coefficient matrix at all levels. At the finest grid level, density varies across the interface smoothly as it is smeared using indicator function. One way to incorporate density values at a coarse level is to employ the restriction operator on the fine level. Another way we consider is to use interface location and employ the density variation sharply only at coarser levels. Original density field with smooth variation and both options for density field at coarse levels are presented in Fig. 10.



(a) (b) (c)

Figure 10. Treatment of the density field in multigrid algorithm. (a) Smeared density field at the finest level, (b) Smooth variation at intermediate levels, (c) Sharp variation at intermediate levels.

Figure 11 shows the performance results at different density ratios for conjugate gradient algorithm and multigrid algorithm with smooth and sharp density treatment. Materials inside and outside the interface is chosen in a way to simulate the problems of bubble and drop dynamics. Ratios less than one corresponds to a bubble (lighter fluid inside the interface) while ratios greater than one corresponds to a drop.

The convergence rate and hence the computational cost of multi-grid algorithm for both density treatments stayed advantageous over conjugate gradient for moderate density variations. All algorithms of MG and CG have shown a significant increase for the iteration numbers when the density ratios are in the order of 1000. Such an increase in iteration number is observed to be higher in cases resembling droplets than the ones of bubbles. Especially, the multi-grid algorithm with smooth density variations in coarse levels requires under-relaxation factor to achieve convergence at the ratio of 1000. On the other hand, convergence is achieved without employing under-relaxation with the sharp treatment for density variation achieved convergence without under-relaxation even though the iterations for convergence still shows a significant increase when compared to moderate density ratios. Figure 11 is produced using 4 sweeps for pre- and post-smoothing inside the multigrid algorithm. When the number of sweeps is increased from 4 to 8 for the case of sharp treatment of density variation, the iteration number for convergence decreased to back to 19. Also, sharp treatment achieved convergence at a reasonable rate for density ratios of 10,000 when the number of sweeps is set 24, while smooth treatment failed to converge for ratios above 1,000.

Figure 12 illustrates total computation time for CG and MG for sharp and smooth treatment. The number of sweeps for smoothing operators is consistent with the iterations presented in Fig. 12 (4 sweeps). Sharp variation for multigrid achieves better performance for the ratios of 0.001, 0.01, 0.1, 1, 10 and 100 than the CG and MG with smooth treatment. The choice of underrelaxation parameters and number of sweeps in smoothing steps of multigrid modifies the overall performance significantly for density ratios in the order of 1000 or higher.

Multilevel additive Schwarz method is explored for its performance on multiple domains. It is similar to the multigrid method presented in a single domain. As it can be seen from Fig. 13, grids at the coarser levels are also decomposed into multiple subdomains provided that they are greater than a threshold value for the number of cells in a single domain. If it is lower than the threshold value or it is the coarsest level, then partitioning is not employed. This is mainly because the hidden latency for the vector-matrix multiplication would become ineffective for the communications on very coarse grids.

Choice for the smoother can affect the performance of the multigrid solver and parallelization procedure. Ghost cells at each grid level are required to communicate as many as the number of sweeps when conjugate gradient is used as a smoother. This is referred to MG-CG algorithm. This algorithm can further be enhanced to yield a better performance by utilization of AS preconditioner to the conjugate gradient algorithm. Another approach is to employ Gauss-Seidel as both pre- and post smoother. This arrangement, referred to MG-GS, requires single communication at the beginning of the sweep and uses that value to smooth the solution field.

Figure 14 shows the number of iterations for each of the algorithm for density ratio of 100. In this figure, coarse granular solver with Gauss-Seidel smoothing has similar characteristics of single level additive Schwarz algorithm. On the other hand, multigrid solver with CG smoother can achieve convergence rates independent of number of subdomains. When employed as a preconditioner, iteration number for convergence decreases.

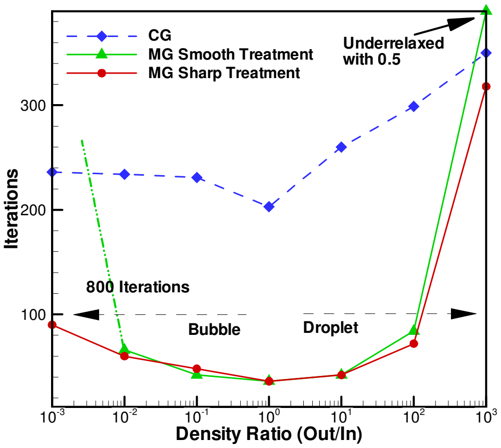


Figure 11. Iteration numbers for convergence of multigrid and conjugate gradient at various density ratios.

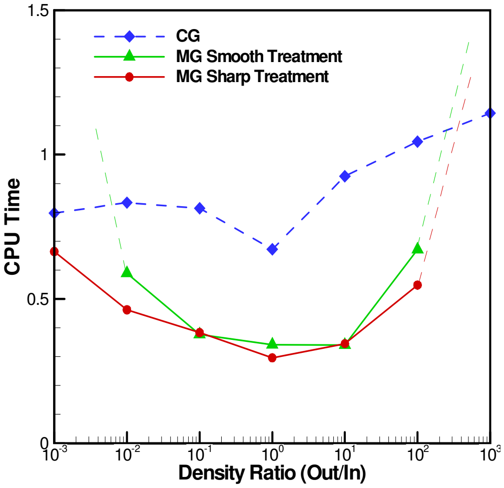


Figure 12. Total CPU time for convergence of MG and CG for various density ratios.

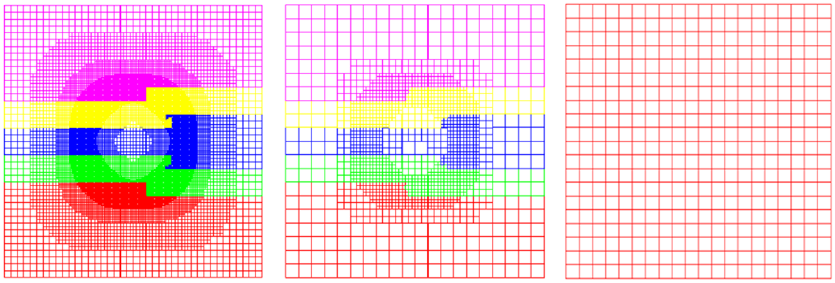


Figure 13. Creation of coarse grid levels and the way they are partitioned into five sub-domains.

B. Liquid Plug Problem

The theory of elongated bubbles is a classical problem in fluid mechanics, referred to as the Bretherton problem or liquid plug flow. Figure 15 illustrates an infinitely long air finger in a channel. It has been shown[1] that the thickness between the film and the channel wall, or the film thickness, decreases with a decrease Capillary number Ca=Uμ/σ, where U is the speed of the air finger, μ is the viscosity and σ is the surface tension. The asymptotic analysis by Bretherton[1] showed that at the limit of Capillary number approaching zero, the film thickness is proportional to $Ca^{2/3}$ (Eq.(17)). Cox[28] showed that the film thickness reaches a finite value for large Capillary numbers.

$$h = 1.3375 Ca^{2/3} \qquad (17)$$

The Bretherton problem has been investigated numerically for intermediate values of Capillary number. Giavedonni and Saita[29] and Heil[30] studied the effects of inertia and showed that at a fixed Capillary number small the film thickness decreases with increasing Reynolds number for viscous dominating flows. This behavior is reversed at high Reynolds number. Heil[30] showed that the formation of closed vortices at the bubble tip causes a significant increase in the pressure rise in that region. It is also concluded that this is most significant for flows at low Ca and high Re, which would cause a rupture[31]. Such a phenomenon can occur in the lungs during a medical treatment process, i.e. drug delivery. The liquid plug forms in the trachea just before inspiration. It propagates during the inspiration while leaving a trailing film thickness and may eventually rupture. In addition, the length of the liquid plug instilled in the trachea is shorter than the length of the trachea. As the plug propagates down the airway tree, the film thickness behind is usually thicker than the film ahead. In order to understand how liquid distributes in the airways throughout the lung, Fujioka and Grotberg[32] studied the interaction of the effect of the plug propagation speed and plug length in both the Stokes flow limit and for finite Reynolds number. They concluded that the film thickness decreases as the plug length decreases below the channel width, and for sufficiently short plugs, there is a significant interaction between the leading and the trailing menisci and their local flow effects.

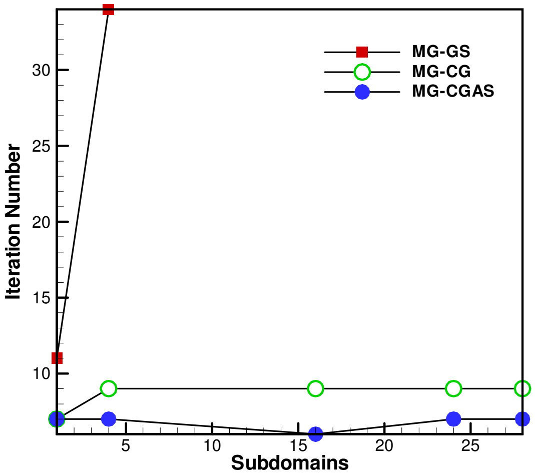


Figure 14. Iteration numbers for different number of sub-domains.

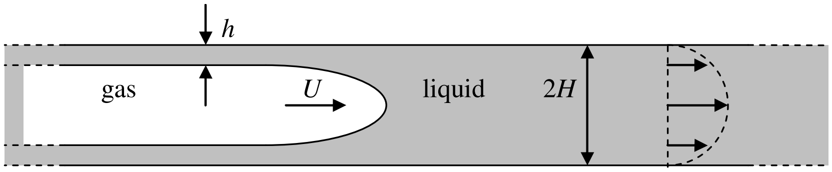


Figure 15. Illustration of the Bretherton problem.

Numerical investigations of the Bretherton problem have so far considered only the steady state using free-surface boundary conditions while neglecting the viscous effects of the air finger[29, 30, 31]. These authors solved the governing equations using body fitted grids, which can deform on the bubble interface. Such an approach is referred to Lagrangian methods which are known to be ineffective at small deformations due to the difficulties in preserving mesh quality. Further investigations of the liquid plug dynamics for the practical purposes require a flexible interface tracking method that can handle the large deformations with a feasible computational cost, especially in the case of understanding the mechanisms of rupture and the interactions for liquid plug problem. Immersed boundary method is a very good candidate for such problems since it has more attractive features when compared to methods that use body fitted grids as it includes the effects of all the constituents in the flow field.

Figure 16 illustrates the computational setup for the Bretherton problem, in which the bubble pressure drives an air-finger into the liquid. A mixed boundary condition is imposed at the inlet, where the constituents are separated by the interface. As the trailing film thickness remains constant further from the inlet condition, the marker point at the boundary is allowed to move only in the vertical direction setting the location for the mixed boundary conditions.

The non-dimensional parameters of the flow are defined as in Eq. (17).

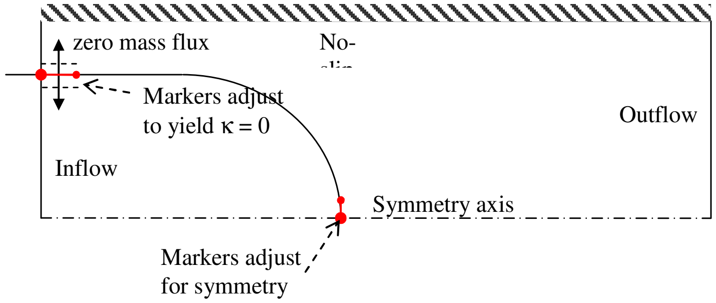


Figure 16. Illustration of computational setup for Bretherton problem.

$$\mathrm{Re} = \frac{\rho_l U H}{\mu_l}, \quad Ca = U\frac{\mu_l}{\sigma}, \quad We = \frac{\mathrm{Re}}{Ca} = \frac{\rho_l \sigma H}{\mu_l^2} \tag{17}$$

In Eq. (17), *H* is the half of the channel height, *U* is the velocity at which the bubble propagates, ρ_l and μ_l are the density and viscosity of the liquid and σ is the surface tension. As the solution of the lower and the upper half of the channel are identical, the study is carried on the upper half.

1. *Steady Propagation of a Liquid Plug*

At the steady state, the finger propagates at a speed of *U* and leaves a film thickness of *h* behind. Therefore, a frame of reference is chosen so that the computational domain moves at a constant speed of *U*. No-slip condition at the channel walls are applied as u=(-1,0). Far right from the bubble tip, open boundary condition is imposed by specifying the pressure gradient. A mixed boundary condition is used on the far left from the bubble tip by enforcing u=(0,0) for air and u=(-1, 0) for the liquid.

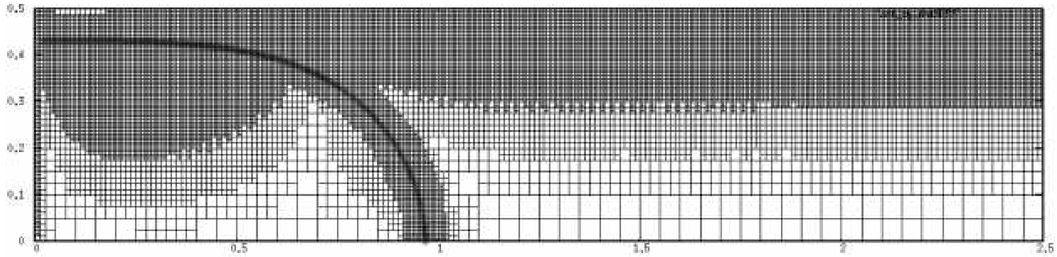
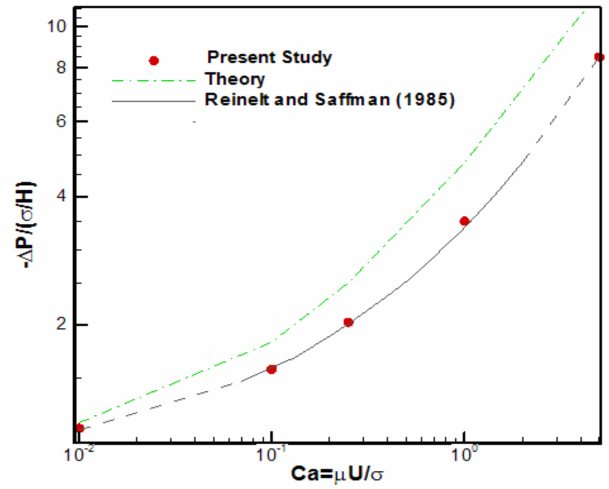


Figure 17. Snapshot of adaptively refined grid (at 3 levels) for Re=0 and Ca=0.05 at steady state.

Figure 17 is an illustration for the adaptively refined grid at *Re*=0 and *Ca*=0.05. The film region, between the channel wall and the interface at the inlet, is required to contain minimum of 10 cells to be able to resolve the flow field. Grid in Fig. 15 is obtained using a base grid of 50x10 and three levels of refinement.



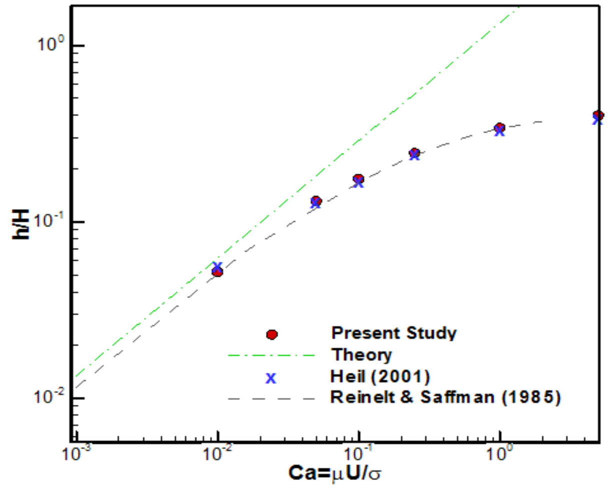


Figure 18. Pressure drop at the bubble tip for various Ca.

Figure 19. Film thickness at various Ca.

Figures 18 and 19 illustrate the variation of the film thickness and the pressure drop at the bubble tip for various Capillary numbers. Flow conditions explored in the present study include *Ca*=0.01, 0.05, 0.1, 0.25, 1.0, and 5.0. Figure 16 illustrates that our results are in good agreement with the prior studies of prior numerical studies[29,30,33,32,34]. Variation from the theory[27] is due to an increase in the capillary number as the approach is applicable to small Capillary numbers. The corresponding bubble shapes at the steady state are also presented in Fig. 20.

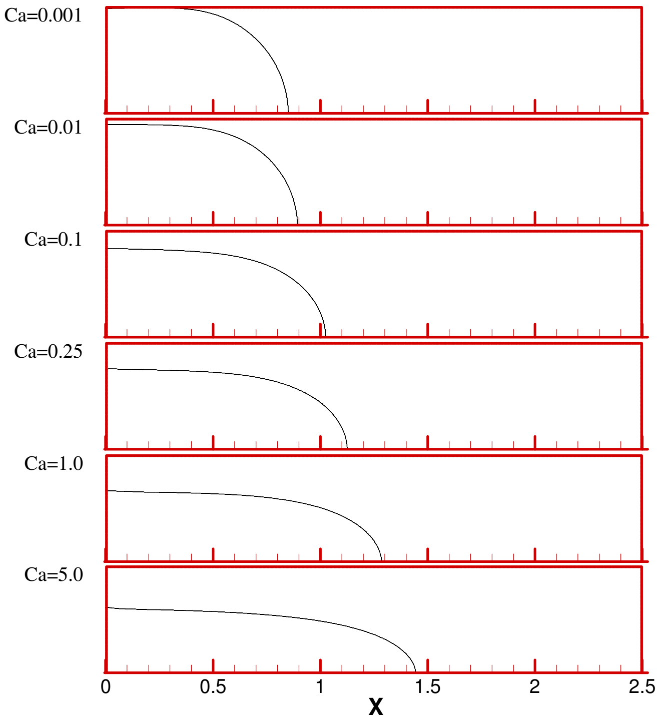


Figure 20. Steady state shapes of the air-finger for various Capillary numbers.

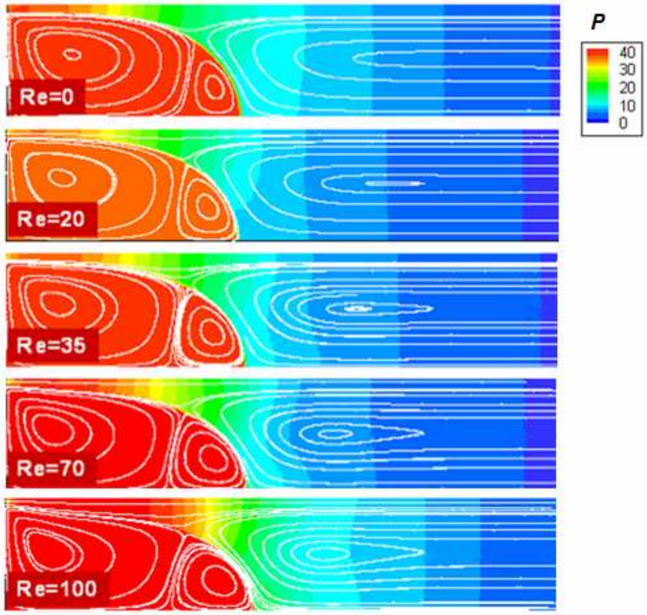


Figure 21. Pressure contours and streamline traces of the steady propagation of liquid plug (Ca=0.05).

Reynolds number at a given Capillary number is also investigated at the steady state. Figure 21 shows that the results obtained using the immersed boundary method are in good agreement with the benchmark solutions of Heil when *Ca*=0.05. In this figure, pressure contours are shown with colors and streamlines are presented via white lines. Similar to their observations, IBM solutions also captured the formation of a closed vortex field in the vicinity of the bubble tip for finite Reynolds numbers. This vortex field is not observed when the fluid inertia is not considered as in the case of *Re*=0 and it is observed to increase with Reynolds number.

Film thickness variation with *Re* at *Ca* =0.05 is shown in Fig. 22 comparing the results with the numerical study of Heil[30]. In the steady state simulations, the initial thickness is usually chosen to be close to the thickness suggested by the analytical theory or earlier numerical studies in order to avoid extensive computation time. However, the steady state solution is independent from the initial conditions and the simulations recover the trailing thickness value regardless of the choice of h_0.

Figure 23 illustrates results of a numerical experiment, in which two different initial values are used for h_0. Film thickness at the steady state for the case *Re*=100 and *Ca*=0.05 is studied on a base grid of 50x10 with two levels of refinement. The film thickness is known to be around 0.137 and the two values for initial thickness are set to 0.1 and 0.2. These values are chosen to approach the steady state value from different directions. The first point is at a closer point to the wall than the steady state, whereas the second choice of initial film thickness is further away from the wall. As it can be observed from Fig. 23, film thickness at the steady state is obtained regardless of the initial film thickness value. The variation from the initial stage to the steady state is derived by the pressure difference due to the boundary conditions. When the marker at the boundary adjusts itself to a new cell, the inlet velocity is readjusted in order to account for the change in the boundary conditions. The effect of this adjustment appears in the time history figures as flat regions, where the change in film thickness is small.

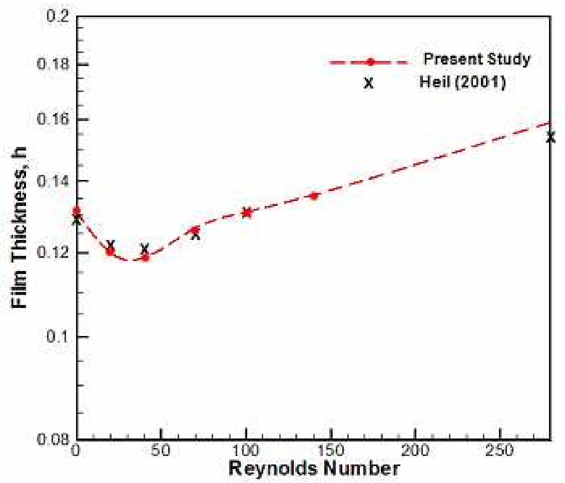


Figure 22. Comparison of the trailing film thickness at *Ca*=0.05 for various Reynolds number.

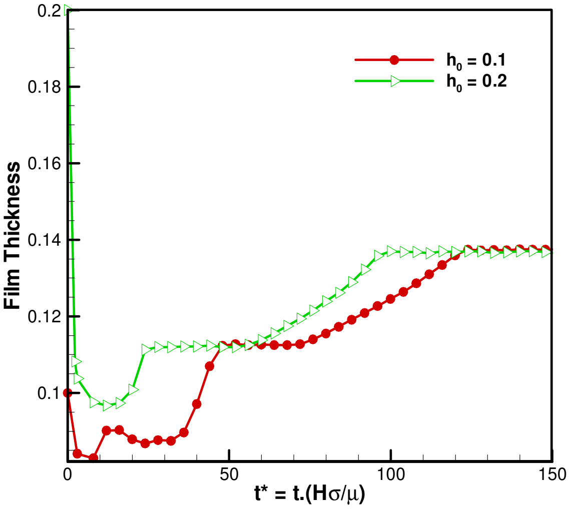


Figure 23. History of trailing film thickness for different initial conditions, i.e., h_0 = 0.1 and h_0 = 0.2 until steady state is achieved.

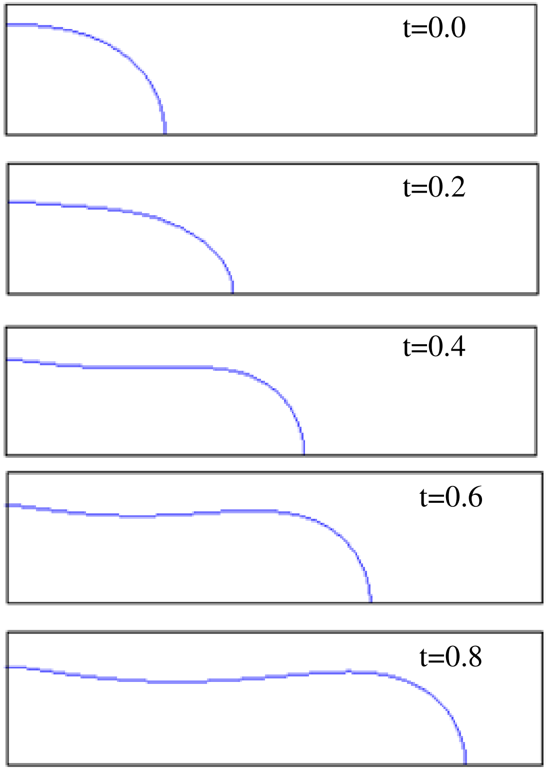


Figure 24. Evolution of air finger in a 2D channel for *Re*=100 and *Ca*=0.05.

2. *Time Dependent Plug Propagation*

Capabilities developed by the current study enable us to further investigate the evolution of the interface for the liquid plug problem. In this section, we focus on the growth of the air finger to capture the bubble shapes. Boundary conditions are chosen to be the same as the ones used in the steady state computations. Representation of the computational domain is shown in Fig. 24. On the other hand, computational frame is stationary and does not follow the bubble to observe the changes at the bubble tip. Imposition of the boundary conditions of unsteady computations is modified for its consistency with the steady state cases. Mixed inflow conditions are given on the left boundary where the liquid velocity is set to *u*=(0,0) and air to *u*=(1,0). No-slip condition is used at the top channel wall by enforcing *u*=(0,0). Outflow and symmetry conditions are imposed for the right and bottom boundaries.

Inlet condition brings extra mass into the computational domain. This condition will cause bubble volume to increase, which is computed using the area and velocity at the inlet. Volume conservation routine accounts for this change when it is required to conserve the mass enclosed by the interface.

Bubble shapes are also studied for *Re*=100 and *Ca*=0.05 for its development in a channel flow. Snapshots from the computations are shown in Fig. 24. As the air finger elongates in time, the liquid plug starts developing a neck between the tip of the air finger and the inlet conditions. This is due to the pressure rise at the film region.

Figure 25 shows the pressure contours at an instant, *t* = 0.8. As the pressure at the film region builds up, the interface

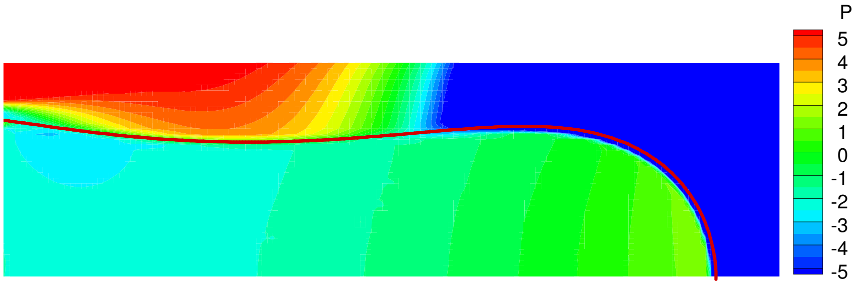


Figure 25. Pressure contours at *t* = 0.8. The cutoff values are limited to ±5 in order to highlight the pressure developing at the film region.

forms a neck by balancing the pressure drop with interfacial forces. If the air finger reaches a certain length, this mechanism would cause rupture as shown in the experiments by Cassidy et al.[35].

Having capabilities of handling complex solid geometries opens up the possibilities to further investigate the unsteady behavior of liquid plug flows in practical applications. One relevant example occurs in the medical treatment in the lung. Liquid plugs are transported through the airways in drug delivery applications. This process relies heavily on the liquid distribution in the lung, which has many branching airways. In order to understand the liquid distribution, it is important to understand the nature of liquid plugs at these airway bifurcations. Developed methodology is utilized to initiate the computational approach that can be directly applied to such a practical problem. Bifurcation point and branches at a predefined angle can be included in the computations via marker points. These markers are tagged as solid points to enforce the boundary conditions at these geometries. Another group of markers, tagged as fluid points, is utilized to define the interface between the air and the liquid phases.

Figure 26 shows snapshots of the air-finger shapes in the channel before its bifurcation point. The development of the air-finger during the initial stages occurs in a similar fashion to the liquid plug in the horizontal channel. However, air bubble, in contrast to the referred case, doesn't create a neck during elongation process.

This is because the bubble shape is affected by the existence of the bifurcation point. As a result, the film thickness gets smaller than the steady state solutions for the same flow parameters while the bubble tip region becomes more flat. Film thickness development in the branching channel is compared with the steady state solutions of Heil[30]. Figure 27 illustrates the variation of film thickness in time before the bifurcation point. According to Fig. 27, the present case study determines thickness to be 0.107 whereas steady state solution predicts the thickness to be around 0.129.

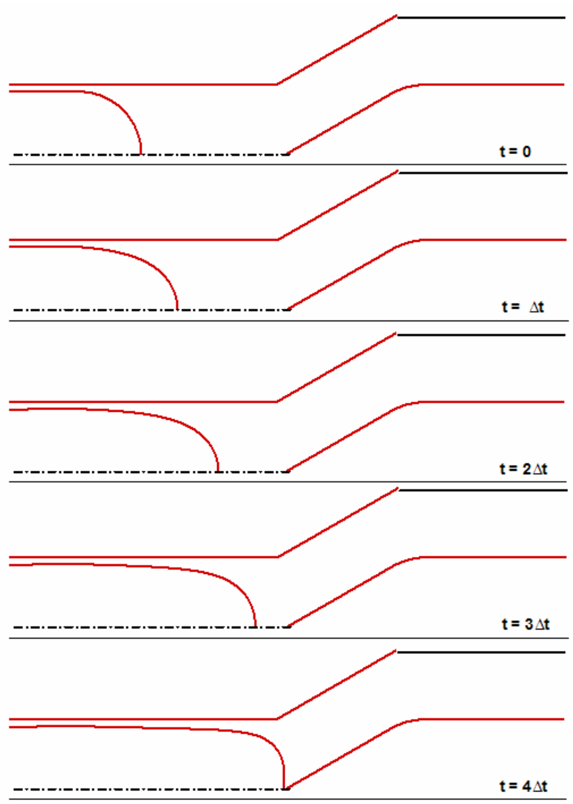


Figure 26. Time-dependent development of the bubble shape in branching channels.

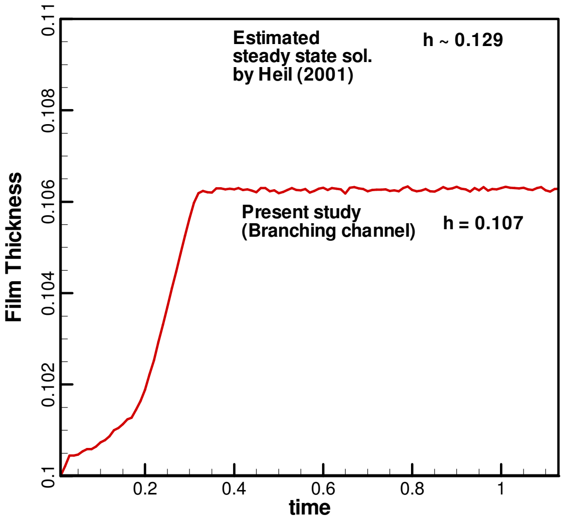


Figure 27. Variation of film thickness in time in the branching channel flow before bifurcation point is reached.

C. The Simulation of Liquid Fuel Tank in Micro-gravity

The dynamics of the fuel delivery at micro-gravity conditions are of interest for spacecraft applications. Some of the many parameters that influence the draining process include interfacial forces, mass flow rate, and tank's geometry. According to Symons[36], these conditions can be classified into three main categories; inertia-dominated, transition, and capillary-dominated regimes. These regimes depend on the draining parameter, which is the ratio of Weber number and Bond number as given in Eq. (18).

$$\text{Weber Number, } We = \frac{\rho Q^2}{\pi^2 \sigma R^3}\text{, Bond Number, } Bo = \frac{\rho g R^2}{\sigma}\text{, Draining Parameter} = \frac{We}{Bo+1} \tag{18}$$

In Eq. (18), 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 phases, g is the gravitational acceleration, and R is the characteristic length of the fuel tank, here chosen as the radius for cylindrical geometry, shown in Fig. 28. Accordingly, non-dimensional time is defined as:

$$t^* = \frac{tQ}{\pi R^3} \tag{19}$$

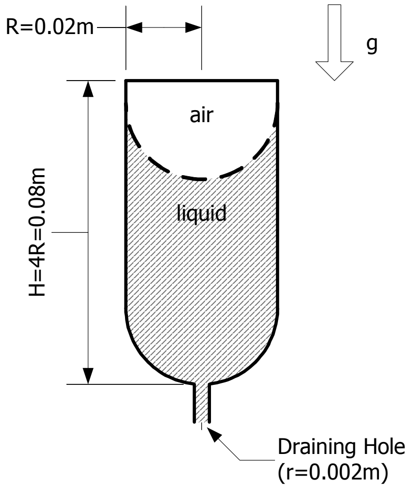


Figure 28. Geometric configuration of the fuel tank.

The draining parameter characterizes the interplay between the various operating factors, including the inertial and imposed pressure, gravity and capillary effects. The inertia-dominated regime is observed when the draining parameter is larger than unity. A main fluid flow feature of the inertia-dominated regime is that the interface centerline height moves towards the bottom of the tank at close to uniform velocities, which decrease away from the draining hole due to curved interface shape. On the other hand, when draining parameter becomes less than one, which corresponds to the transition regime, a few slosh waves with substantial amplitudes are present, causing the interface shape to be influenced by the wave propagation. Further reduction of the draining parameter to less than 0.11, many more slosh waves, with smaller amplitudes than those in the transition regime, appear. This regime is called the capillary-dominated regime.

In this study, we employ the adaptive grid and multigrid-based immersed boundary algorithm to investigate the draining flow characteristics in a hemispheric tank. First, we perform a validation study based on the experiments conducted by Simons[36]. The geometry of the tank utilized in the experiments is shown in Fig. 28, in which the ratio between the outlet radius, *r*, and the tank radius, *R*, is 1/10 and the height is 4*R*. Among the test cases at different regime conditions that the experiments are conducted by Simons, he presented data relevant to interface geometry in the transition regime, with the draining parameter is 0.18, with *We*=1.06 and *Bo*=5. Based on these parameters, we have performed simulations to validate the solutions and to probe the physical implications. In addition, we have conducted a simulation to the inertia-dominated flow regime by increasing *We* to 28.3 with other parameters unchanged. Under this flow conditions, the draining parameter becomes 4.72. For both cases, the initial fuel volume is set to fill 75% of the total tank height as the case was for the experiments. The initial interface geometry is approximated to yield an ellipsoidal shape as the condition for the experiment states the static conditions at the corresponding gravity level based on the zero-contact angle. In the experiment, the air reservoir pressure is determined by the desired flow rate, whereas we specify the same flow rate in the numerical simulations. .

Figure 29 shows the time dependent interface locations at the centerline and on the tank wall for both the transition and inertia-dominated regimes. In the initial draining stage, the interface height at the centerline shows a linear decrease in both regimes. At *t**=0.4, large amplitude sloshes start to form for the transition regime case, indicating the active role played by the capillary effect, whereas the interface in the inertia-dominated regime case continues its motion at the same speed until the vapor ingestion occurs. Comparison between the present numerical and Symons' experimental results for the transition regime case is shown in Fig. 29. Good agreement between experimental and numerical simulations is obtained. On the other hand, it is noted that no direct experimental measurement data of the time history of the interface locations was reported in Symons' experiment in regards to the inertia-dominated regime cases.

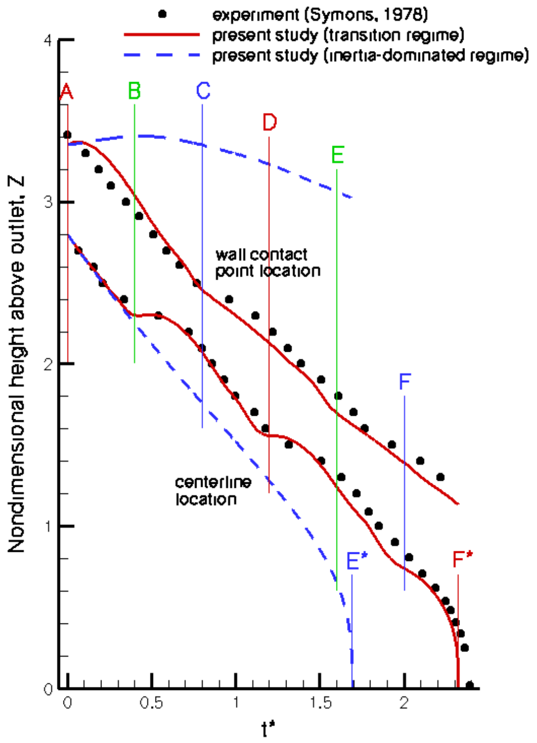
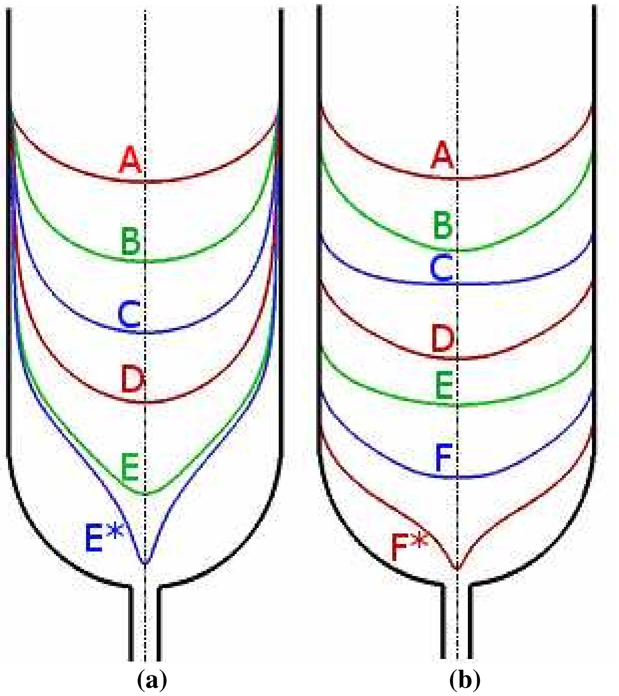


Figure 29. Non-dimensional height at the centerline and on the tank wall at inertia-dominated (*We*/(*Bo*+1)=4.72) and transitional regime (*We*/(*Bo*+1)=0.18). The experimental data was available only for transition regime. Capital letters indicate the time of snapshots of the interface shapes.



(a) **(b)**

Figure 30. Snapshots of interface shape during draining. (a) Inertia-dominated regime (*We*/(*Bo*+1)=4.72) , E* indicates before vapor ingestion, (b) Transition regime (*We*/(*Bo*+1)=0.18) , F* indicates before vapor ingestion.

Instantaneous interface shapes at different time steps are presented in Fig. 30(a) and Fig. 30(b) for inertia-dominated and transition regimes, respectively. In both regimes, the interface location on the centerline moves at approximately constant velocity until it approaches the tank-outlet. On the other hand, the wall attachment location moves in quite different manners between the two cases. In the inertia-dominated regime, the interface on the wall moves slowly, causing the interface to continue to deform and elongate, as shown in Figs. 29 and 30(a). The exact location of the wall attachment is influenced by the contact angle, which is assigned to be 0° here. With a different contact angle, the interface shape and location can behave differently in time[37]. Furthermore, with a dynamic contact angle, as is often observed in reality[38], the situation is more complicated. These aspects will be investigated in the future. For the transition regime case, the interface location on the wall moves from time to time significantly at different speeds in comparison to that of the center line. Consequently, the interface in this case tends to vary between curving up and flattening out, as illustrated in Fig. 30(b). The same characteristics in both regimes were observed in the prior studies[36,39].

The pressure variation and streamlines are presented at t^*=1,69 (marked as E* in Fig. 29), which is just before vapor ingestion for inertial regime in Fig. 31(a), and at t^*=2,00 (marked as F in Fig. 29) for the transition regime in Fig. 31(b). In the transition regime, a vortex is observed around the interface and affects the draining speed of the interface. This vortex is first formed near the wall attachment point and travels as a wave between the wall and the centerline creating the sloshes as a result. Such feature is not observed at inertia-dominated regime, where the fluid flow essentially follows the tank geometry.

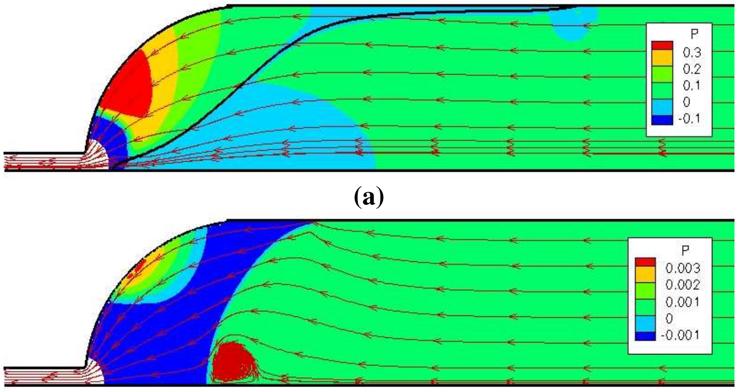
D. Summary and Conclusions

In this paper, we have presented a marker based multiphase front tracking algorithm that utilizes grid refinement and multi-domain methods to aid the computational cost that it comes with. The multi-domain technique utilizes space filling curves to partition the physical space into sub-regions considering the objectives of parallel computing; i.e. load balancing, reduced communication to computation ratio. Furthermore, solution procedure for pressure

Poisson equation is analyzed for its performance when multi-grid and conjugate gradient solvers are utilized. Single domain study compared both methods' convergence rate, total computational time on uniform and non-uniform grids. Its application with additive Schwarz domain decomposition method is found to satisfy another objective of parallel computations by achieving convergence rates independent of the number of sub-domains.

The present algorithm is practiced for problems of two diverse engineering applications, both of which need to resolve the location of the multiphase front within the computational domain. These are namely, liquid plug problem and the draining process of a fuel tank at micro-gravity conditions.

In the liquid plug problem, interactive boundary conditions due to the marker points at the inlet and the symmetry line are used to obtain the steady state solution field. These results are verified using the prior numerical studies. Furthermore, unsteady propagation of the liquid plug is studied in horizontal and branching channels. Since the branching channel flow simulation involves both solid and fluid interfaces, our computations demonstrate the method's capability of handling solid, liquid and gas objects all together.



(a)

(b)

Figure 31. Pressure contours and streamlines. (a) Inertia-dominated regime at *t=1.69 before vapor ingestion, (b) Transition regime showing vortex at *t**=2.00.**

In addition, two draining flow problems, motivated by the spacecraft fuel tank operation, are investigated and validated with the experimental results. Distinct flow characteristics and interface shapes are observed and reproduced satisfactorily by the numerical simulation.

Acknowledgment: The work reported in this paper has been partially supported by NASA Constellation University Institutes Program (CUIP), Ms. Claudia Meyer program director, and Marshall Space Flight Center, Mr. John Peugeot project monitor.

References

[1] Shyy, W., Udaykumar, H. S., Rao, M. M. and Smith, R. W., "Computational Fluid Dynamics with Moving Boundaries", Taylor & Francis, Washington, DC, 1996.

[2] Ryskin G. and Leal L. G., "Numerical solution of free-boundary problems in fluid mechanics", Journal of Fluid Mechanics, 1984, 148:1-43.

[3] Raymond F. and Rosant J.-M., "A numerical and experimental study of the terminal velocity and shape of bubbles in viscous liquids", Chemical Engineering Science, 2000, 55:943-955.

[4] Wasekar V. M. and Manglik R. M., "Short-time transient surfactant dynamics and Marangoni convection around boiling nuclei", Journal of Heat Transfer, 2003, 5: 858-866.

[5] Osher S. and Fedkiw R. P., "Level set methods: an overview and some recent results", Journal of Computational Physics, 2001, 169:463-502.

[6] Osher S. and Fedkiw R. P., "Level set methods and dynamics implicit surfaces", Applied Mathematical Science, Springer-Verlag, NewYork, 2003.

[7] Rider W. J. and Kothe D. B., "Reconstructing volume tracking", Journal of Computational Physics, 1998, 141:112-152.

[8] Scardovelli R. and Zaleski S., "Interface reconstruction with least-square fit and split Lagrangian-Eulerian advection", International Journal for Numerical Methods in Fluids, 2002, 41:251–274.

[9] Ye T., Shyy W. and Chung J. N., "A fixed-grid, sharp-interface method for bubble dynamics and phase change", Journal of Computational Physics, 2001, 174:781-815.

[10] Tryggvason G., Bunner B., Esmaeeli A.., Al-Rawahi N., Tauber W., Han J., Nas S. and Jan Y-J., "A front tracking method for the computations of multiphase flow", Journal of Computational Physics, 2001, 169:708-759.

[11] Peskin, C. S., "The immersed boundary method.", *Acta Numerica*, 11, 2003, 479-517.

[12] Singh R.K. and Shyy W., "Three-Dimensional Adaptive Grid Computation with Conservative, Marker-Based Tracking for Interfacial Fluid Dynamics", AIAA Paper-1523, presented at 44th Aerospace Science Meeting and Exhibit, Reno, NV, 2006.

[13] Singh, R.K. and Shyy, W., "Three-Dimensional Adaptive Cartesian Grid Method with Conservative Interface Restructuring and Reconstruction" accepted for publication in Journal of Computational Physics.

[14] Li Z. and Lai M-C, “The immersed interface method for the Navier-Stokes equations with singular forces”, Journal of Computational Physics, 2001, 171:822-842.

[15] Liu X.-D., Fedkiw R. and Kang M., “A boundary condition capturing method for Poisson’s equation on irregular domains”, Journal of Computational Physics, 2000, 160:151-178.

[16] Shin, S., and Juric, D., "Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity." Journal of Computational Physics, 180(2), 2002, pp. 427-470.

[17] Francois M. and Shyy W., “Computations of drop dynamics with the immersed boundary method; Part 2- Drop impact and heat transfer”, Numerical Heat Transfer, Part B, 2003, 44:119-143.

[18] Yang, J., and Balaras, E., "An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries." Journal of Computational Physics, 215(1), 2006, pp. 12-40.

[19] Gilmanov, A., and Sotiropoulos, F., "A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies." Journal of Computational Physics, 207(2), 2005, pp. 457-492.

[20] Aftosmis, M. J., "Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries." Lecture notes, von Karman Institute for Fluid Dynamics, Series, 2, 1997.

[21] Aftosmis, M. J., Berger, M. J., and Murman, S. M., "Applications of Space-Filling-Curves to Cartesian Methods for CFD." AIAA Paper, 1232, 2003-1237, 2004.

[22] Chorin, A., J., “Numerical Solution of the Navier Stokes Equations”, *Mathematics of Computation*, Vol. 22, pp. 745-762, 1968.

[23] Francois M., Uzgoren E., Jackson J. and Shyy W., “Multigrid computations with the immersed boundary technique for multiphase flows”, *International Journal of Numerical Methods for Heat and Fluid Flow*, 2004, 14:98-115.

[24] Smith, B., Bjørstad, P., and Gropp, W. (1996). Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge University Press, New York, NY

[25] Briggs, W. L., and McCormick, S. F., A multigrid tutorial, Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2000.

[26] Wesseling, P. (1982). "A robust and efficient multigrid method.".

[27] Bretherton, F. P., “The motion of long bubbles in tubes”, *J. Fluid. Mech.*, Vol. 10, 1961, pp. 166.

[28] Cox, B. G., ‘‘On Driving a Viscous Fluid out of a Tube,’’ *J. Fluid Mech.*, Vol. 14, 1962, pp. 81-96.

[29] Giavedoni, M. D., and Saita, F. A., ‘‘The Axisymmetric and Plane Cases of a Gas Phase Steadily Displacing a Newtonian Liquid—A Simultaneous Solution of the Governing Equations,’’ *Phys. Fluids*, 9(8), 1997, pp. 2420–2428.

[30] Heil, M., “Finite Reynolds number effects in the Bretherton Problem”, *Phys. Fluids*, Vol 13, pp. 2517, 2201.

[31] Howell, P. D., Waters, S., L., and Grotberg, J. B., “The propagation of a liquid bolus along a liquid-lined flexible tube*”, J. Fluid Mech.*, Vol. 406, 2000, pp. 309.

[32] Fujioka, H., and Grotberg J. B., “The steady state propagation of a Liquid Plug in Two-Dimensional Channel”, *Journal of Biomechanical Engineering*, Vol. 126, 2004, pp. 567-577.

[33] Reinelt, D. A., and Saffman, P. G. (1985). "The penetration of a finger into a viscous fluid into a channel and tube." *SIAM J. Sci. Stat. Comput*, 6, 542

[34] Halpern, D., and Gaver, D. P., “Boundary-element analysis of the time-dependent motion of a semi-infinitebubble in a channel., *J. Computational Phys.*, Vol. 115, pp. 366-375, 1994.

[35] Cassidy, K. J., Gavriely, N., and Grotberg, J. B. (2001). "Liquid Plug Flow in Straight and Bifurcating Tubes." *Journal of Biomechanical Engineering*, 123, 580.

[36] Symons, E. P., “Draining Characteristics of Hemispherically Bottomed Cylinders in a Low-Gravity Environment”, NASA TP-1279, 1978.

[37] Chato, D. J., Marchetta, J., Hochstein, J. I., and Kassemi M, “Approaches to Validation of Models for Low Gravity Fluid Behavior”, 42nd Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2004, AIAA-2004-1150.

[38] Francois, M. and Shyy, W., "Micro-Scale Drop Dynamics for Heat Transfer Enhancement", 2002, Progress in Aerospace Sciences, Vol. 38, pp.275-304.

[39] Berenyi, S. G. And Abdalla K. L., “Vapor Ingestion Phenomenon in Hemispherically Bottomed Tanks in Normal Gravity and In Weightlessness”, NASA TN D-5704, 1970.