

3-D Adaptive Eulerian-Lagrangian Method for Gravity- and Capillarity-Induced Flows

Jaeheon Sim* and Wei Shyy†
University of Michigan, Ann Arbor, MI, 48109

Delivery of cryogenic propellants from a spacecraft fuel tank is complicated due to the low and fluctuating gravity level and its interaction with the capillary, convective, and diffusive mechanisms. The present effort is aimed at developing suitable computational modeling techniques capable of offering adequate resolution of moving interfacial dynamics, topological changes due to break-up and merger of the fluid objects, and interactions between phase boundaries and complex solid boundaries. A 3-D adaptive Eulerian-Lagrangian method is developed, utilizing the stationary (Eulerian) frame to resolve the flow field, and the marker-based triangulated moving (Lagrangian) surface meshes to treat the fluid interface. The multiphase fluid boundary is modeled using a continuous interface method, and the solid boundary is treated by a sharp interface method along with the ghost cell method. The performance of the present framework is assessed using several test cases of different challenges, including the (i) sloshing liquid motion by a sudden reduction of acceleration exhibiting substantial variations in the shape and the location of the phase boundary, and (ii) stability of the liquid-gas interface dynamics due to vertically oscillating gravitational acceleration of varying frequency and amplitudes, resulting in complex surface wave patterns.

I. Introduction

UNDERSTANDING the dynamics of the cryogenic propellants in a spacecraft fuel tank is crucial in designing and operating a spacecraft. Considering micro-gravity conditions, gravitational, capillary, convective, and viscous effects are important and compete with one another. The influence of these mechanisms determine the amount of fuel delivered to the combustion chamber as well as influence the spacecraft dynamics due to the shift in its center of mass.^{1,2} For example, during spacecraft landing/docking maneuver and/or engine shutdown/restart, the acceleration can decrease or increase suddenly, and consequently, large sloshing motion of cryogenic propellants appears. While the understanding the flow motions in a liquid fuel tank and its influence on the spacecraft dynamics are crucial for spacecraft applications, opportunities for experimental investigations are limited due to the requirements for test conditions such as large acceleration change from high- to micro-gravity.

Numerical simulations of interfacial flows can play instrumental roles in helping improve our capabilities of resolving these issues. The present effort is aimed at developing suitable computational modeling techniques capable of offering adequate resolution of moving interfacial dynamics, topological changes due to break-up and merger of the objects, and interactions between phase boundaries and complex solid boundaries. As reviewed by multiple authors,³⁻⁸ 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^{9,10} which update the grid to match the interface location at every time instant,
- Eulerian methods^{6,11-15} which extract the interface location, as a post-processing step, with the help of a scalar function on a stationary grid,
- Eulerian-Lagrangian methods^{7,16-20} which utilize a separate set of grid representing the interface on a stationary grid used to compute the flow field.

* Graduate Student, Department of Aerospace Engineering, Student member AIAA.

† Clarence L. "Kelly" Johnson Collegiate Professor and Chair, Department of Aerospace Engineering, Fellow AIAA.

Once the location is known, various methods are usually employed to handle the interfacial conditions, including the pressure and viscous stresses across the interface of the multi-fluid, to balance the surface tension effect. Regarding the Eulerian-Lagrangian method, two main categories exist:

- Continuous interface methods^{7, 16, 17, 21, 22} (CIM) which solve one set of equations for the entire flow field by smearing out the flow properties around the interface,
- Sharp interface methods^{14, 23-28} (SIM) which impose the conditions directly on the interface by considering different sets of equations for different phases.

During the last decade, various combinations of the interface tracking methods and interfacial conditions models have been proposed in order to capture the interfacial dynamics. As reviewed by Uzgoren et al.³, the difficulties for modeling interfacial flows are limited to not only the algorithmic challenges, but also the compromise made between the computational cost and accuracy for practically interesting problems. The goal of the present study is to develop a cost-effective method for a unified three-dimensional multiphase flow solver that is capable of capturing the interfacial flow dynamics,

- around complex solid geometries,
- with wetting surface as part of a computational boundary,
- with fronts representing multiple fluid constituents in various phases,
- with topological changes, i.e., possibilities of a merger or a break-up of fluid objects.

In order to achieve the goals listed above, the present study utilizes a marker-based interface tracking algorithm utilizing an Eulerian-Lagrangian method. To treat the interfacial flow, a continuous interface method (CIM) is adopted to utilize a single set of equations for the entire fluid flow field. The stationary grid based on the Cartesian mesh along with multiple levels of local grid adaptation²⁹ is adopted to fully resolve the flow features in the bulk flow field and near the multiphase front. The method in the literature has sometimes been referred to as the Immersed Boundary Method.^{3, 7, 16, 21, 22}

However, the present approach has added capabilities of handling complex solid geometries. Practices of fluid flow simulations oftentimes require capabilities of handling complex geometries needed for engineering problems. In this regard, Yang and Balaras³⁰ reconstructed fluid velocities close to embedded boundary and demonstrated its capabilities via validation studies including the laminar flow induced by oscillating cylinder and turbulent flow over a traveling wavy wall. A similar approach is utilized for the numerical simulations of flapping wings and insect flight.³¹ Mittal et al.³² implemented finite-difference-based ghost cell methodology, in the context of the sharp interface immersed boundary method, by reconstructing velocities at the solid phase to satisfy no-slip condition there. However, their approaches have been validated only for the single phase flow on the solid surface. In the present study, a finite-volume-based ghost cell method is developed to reconstruct the velocity fields by accounting for the surface tension at the fluid interface. A local velocity correction algorithm is used to conserve mass around solid boundary.

When moving multiphase front comes in the vicinity of a wetted surface, a tri-junction location forms 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-Laplace 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 the Young-Laplace 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.²² 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. Especially considering the balance between the viscous and surface tension forces in that region, there is a possibility that this high curvature region can create instability for simulations with large Laplace numbers. Instead of imposing the contact angle, another possibility is to let the interface evolve into the equilibrium condition by modeling the contact line force.

Another issue is that markers representing the multiphase front are tracked with the connectivity information corresponding to a triangulated surface. This connectivity information allows us to compute 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,^{20, 33} to identify the elements intersecting. Another benefit of using the indicator function is to

allow a single set of equation formulation of the continuous interface methods by treating the fluid properties such as density and viscosity across an interface over a finite thickness.

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.³⁴ 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.²⁰

We highlight the capabilities of the techniques presented in our recent papers^{3, 16, 17, 20} by exploring the following flow problems:

- Sloshing motion of the liquid fuel contained in a spacecraft fuel tank subjected to a sudden reduction in acceleration,
- Liquid fuel surface stability due to vertically oscillating thrust, and the relation between threshold acceleration and forcing frequency.

II. Numerical Methods

The present marker-based method employs both Eulerian and Lagrangian variables in order to perform the interfacial flow computations efficiently. The bulk flow variables are solved on the stationary (Eulerian) background grid, whereas interface variables are handled by moving (Lagrangian) markers. A single fluid formulation for all fluid phases is made possible by smoothing out the properties across the interface. Non-dimensionalized incompressible Navier-Stokes equations for mass and momentum conservation are given in Eqs. (1) and (2) respectively, which accounts for the interfacial dynamics as a forcing term. In Eqs. (1) and (2), \mathbf{u} is the velocity vector, p is the pressure, \mathbf{F}_f represents the interfacial dynamics due to surface tension effects of fluid interfaces, and \mathbf{F}_s , represents the forcing function to establish no-slip condition on solid boundaries.

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = -\nabla p + \frac{1}{Re} \nabla \cdot (\mu \nabla \mathbf{u} + \mu \nabla^T \mathbf{u}) + \frac{1}{Fr} \rho \mathbf{g} + \frac{1}{We} \mathbf{F}_f + \mathbf{F}_s \quad (2)$$

Here, all variables are non-dimensionalized by a characteristic velocity (U) and length scale (L), standard gravity (g_0), and liquid material properties (ρ_l , μ_l , and σ_l). The non-dimensional parameters of Reynolds, Froude, and Weber numbers in Eq. (2) are defined as, $Re = \rho_l U L / \mu_l$, $Fr = U^2 / (g_0 L)$, and $We = \rho_l U^2 L / \sigma_l$. In addition, the bond number can be defined as $Bo = We / Fr = \rho_l g_0 L^2 / \sigma_l$.

In general, the pressure and viscous stresses show discontinuities across a fluid interface due to the surface tension force. Equation (3) relates the jump condition of the flow properties in the normal direction, \mathbf{n} , with the surface tension forces, which is directly proportional with the surface tension, σ , and the curvature, κ .

$$(p_2 - p_1) - \mathbf{n} \cdot (\boldsymbol{\tau}_2 - \boldsymbol{\tau}_1) \cdot \mathbf{n} = \sigma \kappa \quad (3)$$

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

Equations (1) and (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 following sequences of steps are used for flow computation.

Step 1: Predictor-step

Solve the momentum equation for an intermediate velocity field \mathbf{u}^* using Eq. (4) where all the known values such as the surface tension source term, forcing function for solid geometries, gravitation, convection, and old time-step viscous term due to Crank-Nicholson method are lumped into \mathbf{S}^n . The term, \mathbf{a}_v , corresponds to the other half of Crank-Nicholson method and represents the coefficients of the stiffness matrix ($\mathbf{f}_{visc}^* = \mathbf{a}_v \mathbf{u}^*$). Temporal discretization is facilitated by the 2nd 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 $d\mathbf{A}$. Subsequently, remove the effect of pressure term by shifting the velocity field back to obtain another intermediate velocity field \mathbf{u}^{**} using Eq. (5).

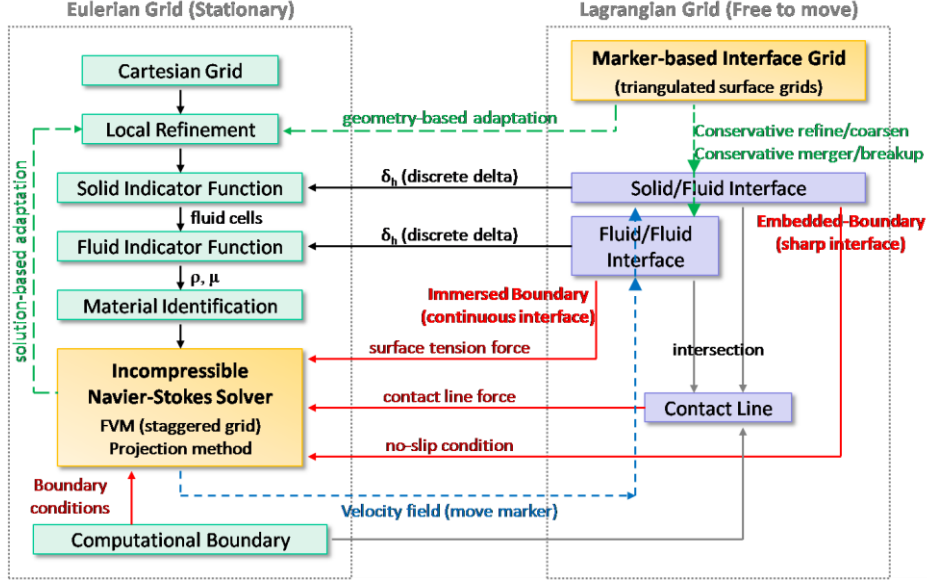


Figure 1. Summary of the numerical method and the interactions between Eulerian and Lagrangian descriptions.

$$\left[\Delta V \frac{\rho}{\Delta t} - \mathbf{a}_v \right] \mathbf{u}^* = - \int_{dA} \nabla P^n \cdot d\mathbf{A} + \mathbf{f}_{visc}^n + \mathbf{S}^n \quad (4)$$

$$\mathbf{u}^{**} = \mathbf{u}^* + \frac{\Delta t \nabla P^n}{\rho^{n+1}} \quad (5)$$

Step 2: Corrector-step

Correct the predicted velocity field (\mathbf{u}^{**}) using Eq. (6). The pressure field for this correction is computed by enforcing the velocity-divergence condition and solving the Poisson equation of Eq. (7). The divergence of the new velocity field \mathbf{u}^{n+1} is zero due to incompressible flow.

$$\mathbf{u}^{n+1} = \mathbf{u}^{**} - \frac{\Delta t \nabla P^{n+1}}{\rho^{n+1}} \quad (6)$$

$$\sum_{face} \left(\frac{\nabla P^{n+1}}{\rho^{n+1}} \right) \cdot d\mathbf{A} = \frac{1}{\Delta t} \sum_{face} \mathbf{u}^{**} \cdot d\mathbf{A} \quad (7)$$

A. Marker-Based Interface Tracking

In the marker-based tracking method, 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 Figure 2. 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. Figure 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 a positive 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.

The marker locations, denoted by \mathbf{X} in Lagrangian frame, for the surface grid are computed from the velocities at its location, similarly denoted by $\mathbf{U}(\mathbf{X})$, in Eq. (8).

$$\frac{\partial \mathbf{X}}{\partial t} = \mathbf{U}(\mathbf{X}) \quad (8)$$

Fluid interfaces use the background solution field to compute the marker velocities as shown in Eq. (9). In this equation, the discrete Dirac delta function, $\delta(\mathbf{x} - \mathbf{X})$, is employed for converting the Eulerian velocity field, $\mathbf{u}(\mathbf{x})$, to Lagrangian form, $\mathbf{U}(\mathbf{X})$. On the other hand, solid interfaces use the prescribed velocity field to advance the marker points using Eq. (8).

$$\mathbf{U}(\mathbf{X}) = \int_v \mathbf{u}(\mathbf{x}) \delta(\mathbf{x} - \mathbf{X}) dV \quad (9)$$

As the marker points advance to a new position with time, it can lead to an unevenly distributed marker points on the interface surface. 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 quantities. For fluid interfaces, Eq. (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 \mathbf{X} and an adjacent marker located at \mathbf{X}_{ngbr} as a function of Eulerian grid spacing, Δ , given in Eq. (10).

$$\frac{\Delta}{3} < |\mathbf{X} - \mathbf{X}_{ngbr}| < \Delta \quad (10)$$

When the distance between two marker points violate the upper bound of Eq. (10), a new marker is placed on the mid-point of the edge between marker \mathbf{X} and \mathbf{X}_{ngbr} . The connectivity information is updated in the vicinity of the 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. (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 adjusting 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.¹⁷ for further information on restructuring of triangulated surface representation.

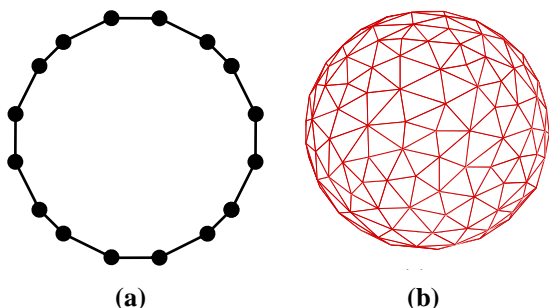


Figure 2. Interface representation by marker points. (a) Line segments in 2D, (b) Triangular elements in 3D.

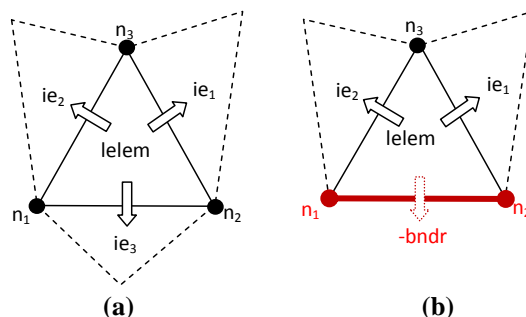


Figure 3. Connectivity information through element edges. (a) inner element with three neighbors, (b) boundary element with two neighbors.

B. 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 continuous interface method for the whole domain, a smooth variation of discontinuous material properties across interfaces is 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. (11).

$$\varphi = \varphi_2 + (\varphi_1 - \varphi_2)I \quad (11)$$

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 the success of numerical simulation of 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. (12), while others utilize a discrete form of the Heaviside step function by integrating 1-D form of discrete Dirac delta function.

$$\nabla^2 I = \nabla \left(\int_A \delta(\mathbf{x} - \mathbf{X}) dA \right) \quad (12)$$

Obtaining a numerical solution to Eq. (12), in which the location of the interface is represented by the Lagrangian quantity, \mathbf{X} , while \mathbf{x} corresponds to the Eulerian description of the computational domain. Analytical form of the Dirac delta function is only non-zero at $\mathbf{x} = \mathbf{X}$, 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.^{21, 37, 38}

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. Figure 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. zero or one. This approach is computationally effective since we assign indicator function instead of computing them in most computation region. In this paper, discrete Dirac delta function, that supports the conservation rules dictated by zeroth, first and second moments as described in Peskin,²¹ is employed. In order to get indicator function, the Heaviside step function in Eq. (13) which utilizes the shortest distance value between the cell-center to the interface location is utilized by integrating the one-dimensional form of discrete Dirac delta function since it gives more accurate values on the boundary. More details and comparison between Poisson equation and Heaviside step function are found in our previous researches.³⁹

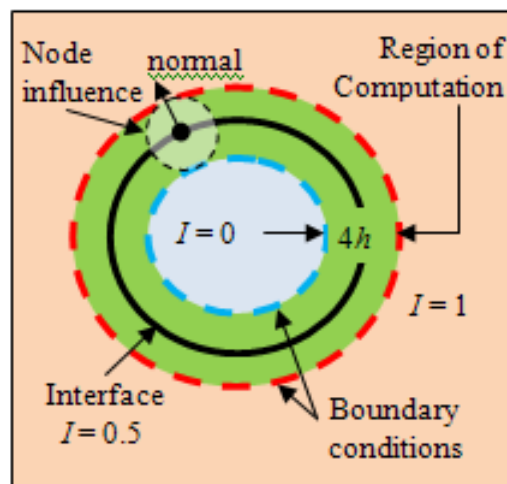


Figure 4. Region of computation and boundary conditions.

$$I(r) = \frac{1}{8} \begin{cases} 0 & r < -2 \\ \frac{23}{4} - 2\pi + 5r + r^2 - \left(\frac{r}{2} + \frac{3}{4}\right) \sqrt{-4r^2 - 12r - 7} - \frac{1}{2} \sin^{-1} \left(-\frac{2r+3}{\sqrt{2}} \right) & -2 < r \leq -1 \\ \frac{15}{4} - 2\pi + 3r + r^2 + \left(\frac{r}{2} + \frac{1}{4}\right) \sqrt{-4r^2 - 4r + 1} - \frac{1}{2} \sin^{-1} \left(-\frac{2r+1}{\sqrt{2}} \right) & -1 < r \leq 0 \\ \frac{17}{4} + 2\pi + 3r - r^2 + \left(\frac{r}{2} - \frac{1}{4}\right) \sqrt{-4r^2 + 4r + 1} - \frac{1}{2} \sin^{-1} \left(-\frac{2r-1}{\sqrt{2}} \right) & 0 < r \leq 1 \\ \frac{9}{4} + 2\pi + 5r - r^2 - \left(\frac{r}{2} - \frac{3}{4}\right) \sqrt{-4r^2 + 12r - 7} - \frac{1}{2} \sin^{-1} \left(-\frac{2r-3}{\sqrt{2}} \right) & 1 < r \leq 2 \\ 1 & r > 2 \end{cases} \quad (13)$$

C. 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. The details of the algorithm can be found Singh and Shyy.^{36, 40} Adaptation is performed based on the interface location and the flow solution quality.

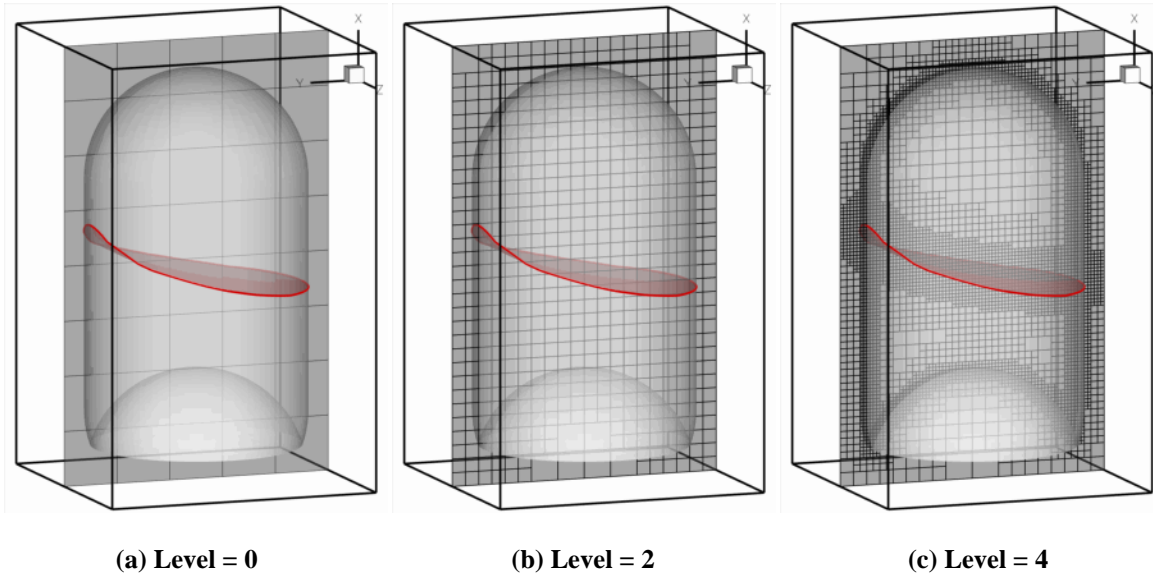


Figure 5. Snapshots of geometry-based grid adaptation for fluid and solid interfaces for spacecraft fuel tank with concave-shaped bottom. Different adaptation levels are shown in (a) level 0, (b) level 2, (c) level 4

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. Figure 5 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 delta function, triggers full refinement on vicinity including two-cell layers from the interface.

Cells, those are not already fully refined, are adapted based on the solution of the flow field. The present implementation uses a curl-based adaptation criterion⁴¹ that computes a parameter ξ for each cell as shown in Eq. (14). 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 of Eq. (15) using the criteria in Eqs. (16) and (17).

$$\xi_{cell} \sim |\nabla \otimes \mathbf{U}|^l \quad (14)$$

$$\sigma' = \frac{1}{N_{cell} \sum_i \xi_i^2} \quad (15)$$

$$\xi_{cell} > \sigma' \rightarrow \text{Refine cell} \quad (16)$$

$$\xi_{cell} < 0.1\sigma' \rightarrow \text{coarsen the cell} \quad (17)$$

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 \mathbf{U}^{**} .

D. Fluid Interface Treatment: Surface Tension

When interface separating fluid phases, the source term arises from the surface tension (σ) and the curvature (κ) as shown in Eq. (18).

$$\mathbf{F}_s = \int_A \sigma \kappa \delta(\mathbf{x} - \mathbf{X}) dA \quad (18)$$

The surface force is computed using the Lagrangian marker points, \mathbf{X} , and is translated into an Eulerian quantity, \mathbf{x} , via the approximate discrete Dirac delta function, $\delta(\mathbf{x} - \mathbf{X})$. 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. (19) where unit normal vector and curvature can be computed using curve fitting for two-dimensional interfaces^{22, 23, 42} and surface fitting for three-dimensional interfaces;⁴³ computation using a line integral form shown in Eq. (20) and fitting curves/surfaces to obtain normal and tangent vectors.^{7, 44}

$$\delta \mathbf{f} = \int_{\delta A} \sigma \kappa dA \quad (19)$$

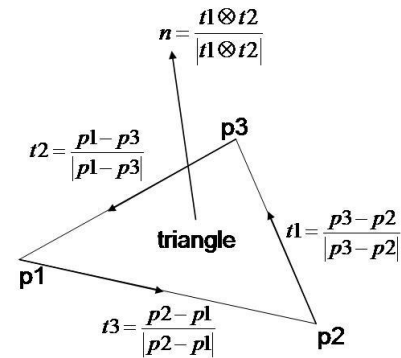
$$\delta \mathbf{f} = \int_{\delta A} \sigma (\mathbf{n} \times \nabla) \times \mathbf{n} dA = \int_s \sigma (\mathbf{t} \otimes \mathbf{n}) ds \quad (20)$$

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.^{22, 42, 43, 45} The use of Eq. (19) does not enforce conservation whereas the line-integral form, Eq. (20), does not require explicit curvature computation and maintains the conservation.

The approach developed by Singh³⁶ uses the line integral form and computes the local normal and tangent vectors along the triangle edges using the simple approach of Al-Rawahi^{44, 45} shown in Eq. (21) following Figure 6. If required, the curvature can be computed using Eq. (22). The overall accuracy of this approach to compute surface tension force and its modeling have already been demonstrated for boiling flows⁴⁶ and for dendritic solidification.⁴⁵

$$\delta \mathbf{f} = \sum_{edge=123} \sigma (\mathbf{t} \otimes \mathbf{n})_{edge} \Delta s \quad (21)$$

$$\kappa = \frac{\delta \mathbf{f} \cdot \mathbf{n}}{\sigma \Delta A} \quad (22)$$



triangle => p1, p2, p3
Figure 6. Computation of the unit normal and tangent vectors on interface triangles.

E. 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), the source term due to solid interfaces, \mathbf{F}_s , can be estimated at its non-zero locations with the help of the prescribed velocity at the interface, \mathbf{u}_{int} , and other forcing terms, i.e. gravity, \mathbf{g} , and surface tension forces, \mathbf{F}_s , as presented in Eq. (23).

$$\mathbf{F}_s = \rho \frac{\partial \mathbf{u}_{int}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u})_{int} - \nabla p - \frac{1}{Re} \nabla \cdot (\mu \nabla \mathbf{u} + \mu \nabla^T \mathbf{u})_{int} - \frac{1}{Fr} \rho \mathbf{g} - \frac{1}{We} \mathbf{F}_f \quad (23)$$

Following the prior naming by Yang and Balaras,³⁰ 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.^{47, 48} 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. (24) with the guidance of Figure 7.

$$\frac{P_E - P_P}{\Delta x} A_e + \frac{P_W - P_P}{\Delta x} A_w + \frac{P_N - P_P}{\Delta y} A_n = u_e^* A_e - u_w^* A_w + v_n^* A_n - v_{bndr}^{n+1} A_s \quad (24)$$

The last term of the right hand side term in Eq. (24), 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 using ghost cell methodology as shown in Figure 8. It should also be noted that, these forcing faces can be set in the fluid side, and the fluid velocities close to solid boundary are reconstruct directly.^{30, 31} However, this approach creates incorrect velocity fields around contact line where fluid interface meets solid surface by ignoring the influence of surface tension and contact line force from fluid interfaces although both approaches works well for the single phase flow on the solid surface. In the present study, the forcing velocity is reconstructed on the solid side using ghost cell methodology to solve multiphase flow around irregular solid geometry.

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. However, it may causes the same issue creating incorrect velocity fields around contact line when the forcing faces are located in the fluid side. In order to solve such an issue of staggered grid and to improve computational efficiency, ghost cell is defined right out of fluid boundary cells in Figure 8. The interpolation is conducted not on the forcing face itself but on the center of ghost cells, and thus all interpolation are

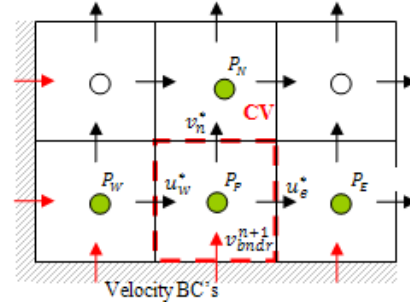


Figure 7. Pressure control volume at the computational boundaries

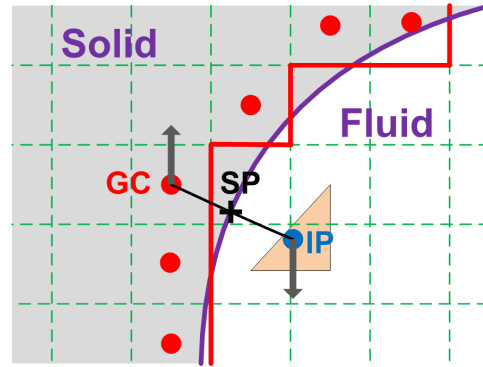


Figure 8. Identification of ghost cell (GC), solid point (SP), and imaginary point (IP). The linear velocity interpolation is conducted on the imaginary point first, then computed at the ghost cells based on the velocity of solid point.

conducted on the solid side not disturbing fluid velocities, and save computational memories and time since cell-based linear interpolation requires only one-third of linear equation sets for interpolation.

The first point on the interpolation scheme, the closest location, namely solid point (SP), on the solid interface from ghost cell, is found by comparing the distance normalized by the grid spacing for the elements in the vicinity of the ghost cells. Once determined, interpolation weights between points i and j , denoted by w_{ij} , can be computed based on inverse distance, Δ_{ij} , using Eq. (25) and following Figure 9. Then the prescribed condition on the interface can be obtained for any function, ϕ , using Eq. (26).

$$w_{ij} = \frac{1/\Delta_{ij}}{\sum_{i=1,3} (1/\Delta_{ij})} \quad (25)$$

$$\phi_j = \phi_1 w_{1j} + \phi_2 w_{2j} + \phi_3 w_{3j} \quad (26)$$

Next, the imaginary point (IP) is obtained by mirroring the ghost cell based on the solid point for interpolation in Figure 8. The remaining parts in the interpolation scheme are the fluid cells, identification of which is one of the most critical parts of the algorithm. The accuracy of the interpolation scheme improves when those fluid cells are selected as close as possible to the imaginary point 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 cell-centers included in the scheme, a short list of fluid cells is formed using the neighboring cells. This list is sorted using a *merge-sort* algorithm based on the distance values. The various combinations of cells 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 of Figure 8, and a tetrahedron in 3D.

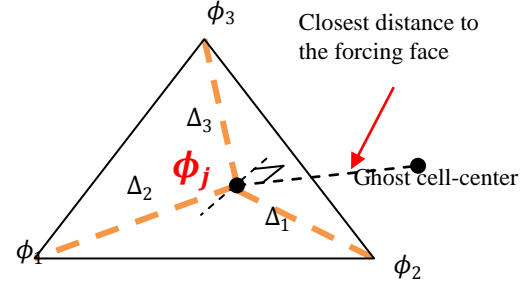


Figure 9. The closest interface element to a forcing face.

The interpolation procedure is performed assuming a linear variation of any variable, ϕ . Equations (27) and (28) are the formulation of the procedure in 2D.

$$\phi = b_1 + b_2 x + b_3 y \quad (27)$$

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \quad (28)$$

In Eqs. (27) and (28), x_i and y_i represents the corners of the triangle presented in Figure 8. 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 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 ghost cell velocities can be computed simply since the solid point is the midpoint between ghost cell and imaginary point. The required velocities of forcing faces are averaged from the velocities of faced cells.

The above interpolation scheme is utilized in the estimation of the forcing term, \mathbf{F}_s , in Eq. (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, \mathbf{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. (24).

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. (24). 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.

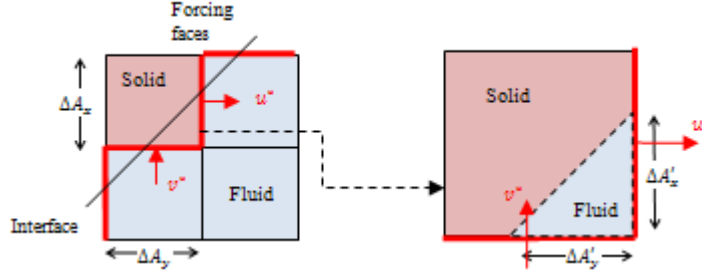


Figure 10. Region of correction to ensure divergence free velocity field.

Figure 10 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. (29) should be satisfied, resulting an error, e , when the full area for each face is utilized as given in Eq. (30).

$$u^* \Delta A'_x - v^* \Delta A'_y = 0 \quad (29)$$

$$u^* \Delta A_x - v^* \Delta A_y = e \quad (30)$$

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. (31), a correction term for the forcing face velocities to adjust the fluxes accordingly.

$$e = \epsilon(\Delta A_x + \Delta A_y) \quad (31)$$

$$(u^* - \epsilon) \Delta A_x - (v^* - \epsilon) \Delta A_y = 0 \quad (32)$$

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 Figure 11. 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 Figure 11, 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.

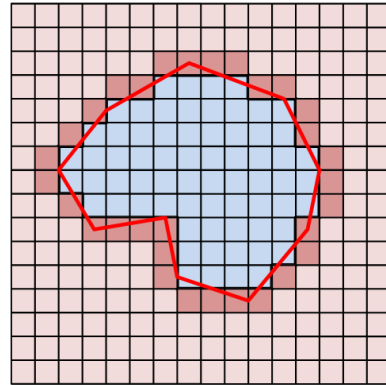


Figure 11. Global conservation with local correction.

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. (27) leaving the fluid faces at the u^n value. This corresponds to the first step in Table 1, which summarizes the algorithm advancing from time step $[n]$ to time step $[n + 1]$. The rest of predicted velocities (u^*) for fluid faces are accounted for on solving the advection-diffusion equation as appears in second step of Table 1. The correction of mass conservation for solid boundary is required only by the Poisson equation as mentioned above. 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. These procedures should be implemented during solving pressure Poisson equation in third step by modifying the velocity terms only for forcing faces. The fluid velocities at time step $[n + 1]$ will be updated from pressure gradients solved at the previous step in the last step of Table 1.

Table 1. Overall algorithm for solid boundary treatment

Apply forcing terms (only for forcing faces): $\mathbf{u}_i^* = \mathbf{b}_{i1} + \mathbf{b}_{i2}x_i + \mathbf{b}_{i3}y_i + \mathbf{b}_{i4}z_i \quad i: \text{only forcing faces}$
Solve for advection diffusion equation (only for fluid faces): $\left(\Delta V \frac{\rho^{n+1}}{\Delta t} - \mathbf{a}_v\right) \mathbf{u}^* = - \int_{dA} \nabla p^n \cdot \mathbf{n} dA + \mathbf{f}_{visc}^* - \mathbf{f}_{conv}^* + \mathbf{f}_f^* + \mathbf{g}$ $\mathbf{u}_i^{**} = \mathbf{u}_i^* + \frac{\Delta t (\nabla p^n)_i}{\rho^{n+1}} \quad i: \text{only fluid faces}$
Solve for continuity equation (pressure Poisson equation) $\sum_i \left(\frac{(\nabla p^{n+1})_i}{\rho^{n+1}}\right) \cdot \mathbf{n} dA = \frac{1}{\Delta t} \sum_i \mathbf{u}_i^{**} \cdot \mathbf{n} dA + \frac{1}{\Delta t} \sum_j (\mathbf{u}_j^* + \boldsymbol{\epsilon}_j) \cdot \mathbf{n} dA$ $i: \text{only fluid faces}$ $j: \text{only forcing faces}$
Correct velocity field: $\mathbf{u}_i^{n+1} = \mathbf{u}_i^{**} - \frac{\Delta t (\nabla p^{n+1})_i}{\rho^{n+1}} \quad i: \text{only fluid faces}$

F. 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^{49,50,51} is adopted in this study.

The angle at the contact line, shown in Figure 12, can be used for representing the balance of forces resulting from intermolecular forces between solid, liquid and gas phases. The tangential component of the resulting force, F_R , is shown in Eq. (33). In static equilibrium, Eq. (33) leads to the well-known Young-Laplace equation as given in Eq. (34).

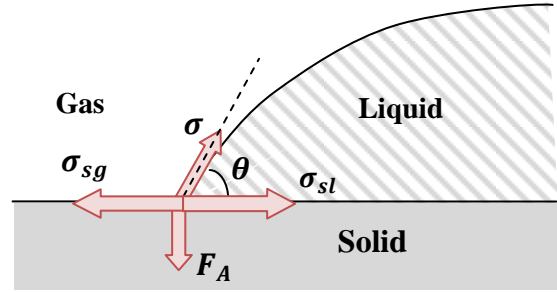


Figure 12. Forces at tri-junction (contact line). F_A represents the adhesive forces.

$$F_R = \sigma_{sg} - \sigma_{sl} - \sigma \cos \theta \quad (33)$$

$$\sigma_{sg} = \sigma_{sl} + \sigma \cos \theta_0 \quad (34)$$

In Eq. (33), 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 σ is the surface tension defined for liquid and gas.

In this work, we adopt an approach based on a simplified version of Huang et al.⁵¹ 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. (34) into Eq. (33), to estimate the force to recover the static contact angle.

$$F_R = \sigma(\cos \theta_0 - \cos \theta) \quad (35)$$

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, \mathbf{t}_s , instead of the curvature effects given in Eq. (20). The contact angle is enforced on open edges where the element connects to either a computational boundary or a solid interface by modifying Eq. (20) into Eq. (36). In Eq. (36), 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.

$$\delta \mathbf{f} = \sum_{\substack{i=\text{closed} \\ \text{edges}}} \sigma (\mathbf{t}_f \otimes \mathbf{n})_i \Delta s + \sum_{\substack{i=\text{open} \\ \text{edges}}} \sigma (\cos \theta_0 - \cos \theta)_i \mathbf{t}_s \Delta s \quad (36)$$

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. (9), we consider a one-sided averaging via Dirac function only for fluid side. This treatment is represented by Eq. (37).

$$\mathbf{U}(\mathbf{X}) = \frac{\sum_x \mathbf{u}(\mathbf{x}) \delta(\mathbf{X} - \mathbf{x})}{\sum_x \delta(\mathbf{X} - \mathbf{x})} \quad (37)$$

In Eq. (37), 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.

G. 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 B. 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 solid indicator function which is available on the Cartesian grid. The information of solid indicator function, I_s , is transferred onto fluid interface markers, \mathbf{X}_f , via Eq. (38) from the Cartesian grid cells, \mathbf{x} .

$$I_s(\mathbf{X}_f) = \int_v I_s(\mathbf{x}) \delta(\mathbf{x} - \mathbf{X}_f) dv \quad (38)$$

When the value of I_s at any marker is smaller than 0.5, which corresponds to the location of the solid interface, it is designated as out-of-fluid. In addition, elements having markers all to be out-of-fluid 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 element and 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, the open edge is extended or trimmed by moving markers on to the solid interface using the distance information as shown in Figure 13.

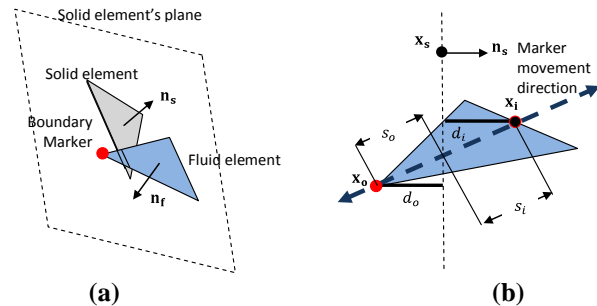


Figure 13. Intersecting fluid and solid interfaces. (a) intersection between elements (b) Snapping procedure (point of view is tangential direction to the solid element. Marker on the open edge is marked with red point.)

III. Computational Assessment

To highlight the performance of the present approach, case studies have been conducted for sloshing fluid dynamics and combined gravity- and capillarity-induced surface waves. They are directly relevant to the spacecraft fuel motion under a sudden reduction of the acceleration, and fuel interface stability under thrust oscillation. These cases are presented in the following.

A. Sloshing of cryogenic propellants in a spacecraft fuel tank

Figure 14(a) shows the configuration of the scale model of the Saturn V/S-IVB liquid hydrogen tank experimented by Toole et al.¹ Their concave-shaped tank is illustrated by the triangular solid interface elements in Figure 14(b) for 3D computation on the stationary Cartesian grids. Petroleum ether and air are utilized as a substitute of liquid fuel and pressurizing gas respectively as they are in the experiment. The properties of petroleum ether at 21°C are given in Table 2.

Based on this experimental investigation, a test case, 2F-19 with $Bo=24$ and $Fr=14.6$, is chosen to assess our computational modeling performance. These values correspond to an acceleration level of $0.01g_0$ and maximum fluid velocity of 0.33m/s. The acceleration history is shown in Figure 15. The key non-dimensional parameters are the *Bond number* and the *Froude number*, defined in Eqs. (39) and (40), which are related to the parameters in Eq. (2).

Table 2. Properties of petroleum ether at 21°C¹

density	6.382e+2 kg/m³
viscosity	2.489e-4 kg/m.s
surface tension	1.490e-2 N/m

$$\text{Bond Number, } Bo = \frac{\rho a R^2}{\sigma} \quad (39)$$

$$\text{Froude Number, } Fr = \frac{V_{max}^2}{aR} \quad (40)$$

One of the difficulties is to match Froude number with experiment since we don't have any information on the kinetic energy at a given interface shape. There is no record for the impacting force to make the initial sloshing motion, thus we pick a free surface shape from the snapshot with highest wall attachment point, then adjust the acceleration time to match the Froude number measuring the maximum velocity of a wall attachment point by trial and error.

Figure 16 shows a comparison between the present computational result and the experiment. In the numerical result, the simulation started at $t=-0.12$ sec instead of -0.14 sec of Toole's snapshot to match the given Froude number. The fluid motion is accelerated from the initial stationary position as the potential energy is converted into kinetic energy by acceleration of normal gravity. At $t=0.00$ sec, the acceleration suddenly reduced into $0.01g_0$ as described

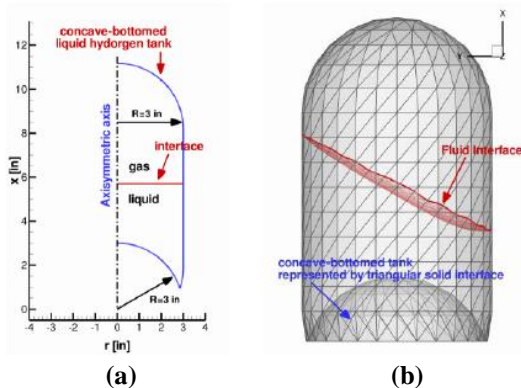


Figure 14. The computational geometry configuration of draining fuel tank with hemispherical bottom (a) axisymmetric domain, (b) 3D computational domain

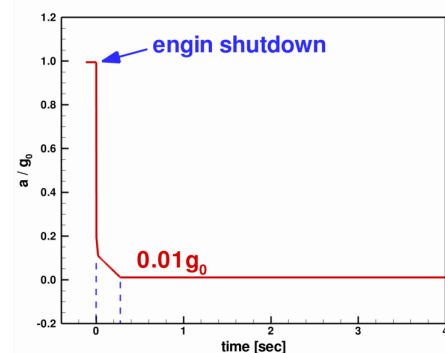


Figure 15. Acceleration history for simulation with engine shutdown at $t=0$. The acceleration decreases from normal gravity to micro-gravity ($0.01g_0$).

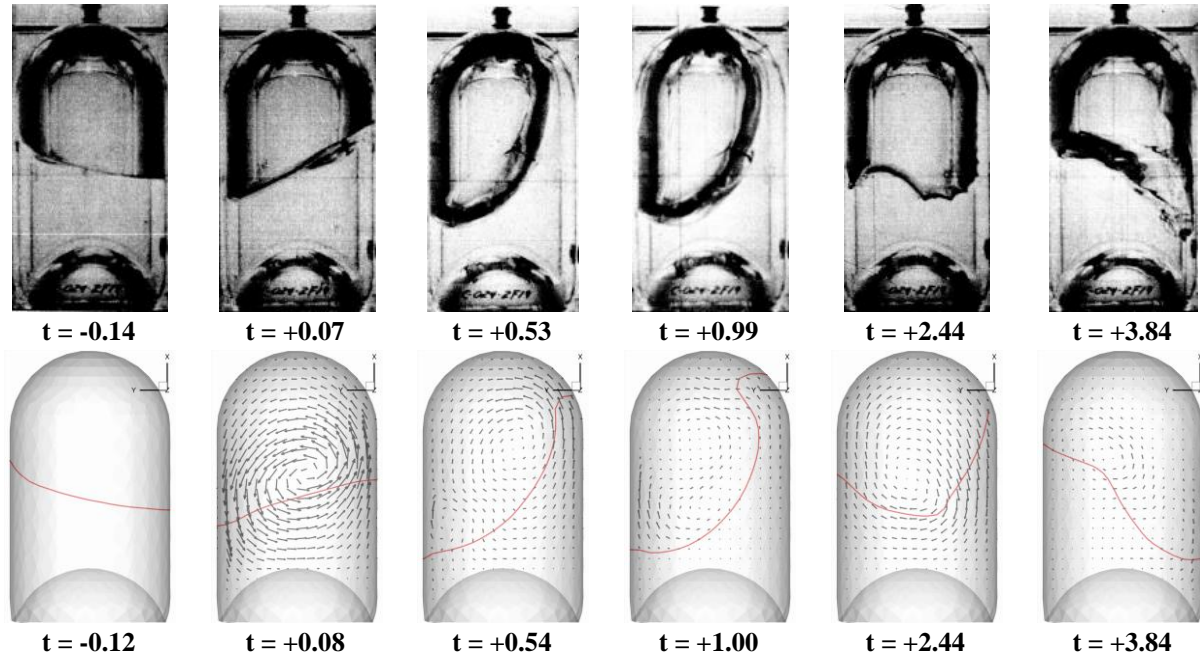


Figure 16. Comparison of present sloshing fuel tank simulation with experiment by Toole. The velocity vector and fluid surface are represented on a center-cut plane.

in Figure 15, and the high kinetic energy obtained at normal gravity is converted back into potential energy at micro-gravity. Smaller acceleration results in higher liquid fuel position to get same potential energy level. The liquid fuel reaches and stays at the top of the tank for the time being, and then, it retracts and makes the second sloshing motion. The numerical study is not perfectly identical with experiment since the kinetic energy distribution is different due to different mechanism making sloshing wave, but the overall liquid fuel motion and wave period are nearly identical and shows good agreement with experiment.

Figure 17 shows the change of the center of mass in time. Axial location plot indicates that the center of mass moves by 22% of tank radius, and the sloshing waves damps very fast. In Figure 17(b), the z-directional change can be negligible as expected, and y-directional change is up to 46% of tank radius. The huge change of the center of mass requires excessive demand on the whole vehicle dynamics and control systems.

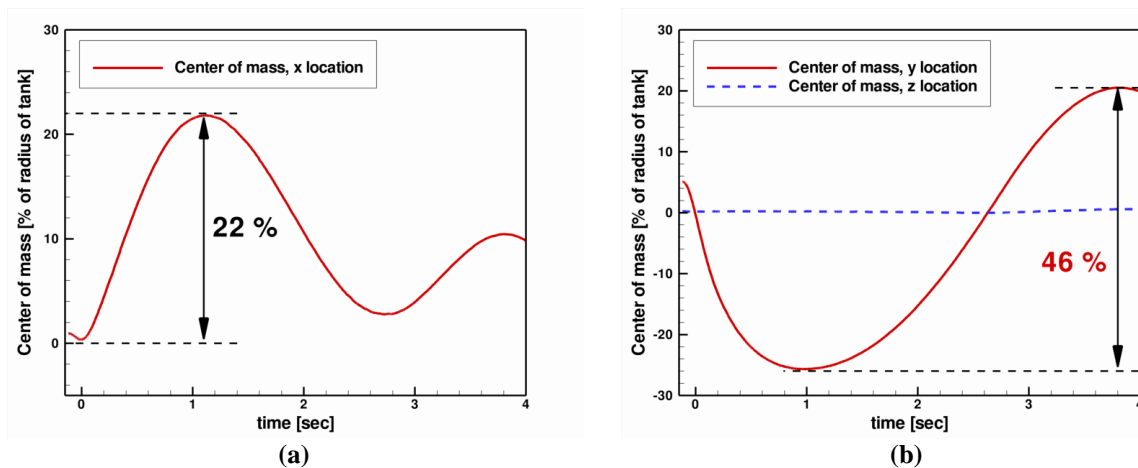


Figure 17. The center of mass of sloshing cryogenic fuel tank (a) axial x-location, (b) radial y and z-location

B. The liquid fuel surface stability of a spacecraft under thrust oscillation

Thrust oscillation is one of the most difficult problems in liquid and/or solid rocket designs since it has influence on the whole rocket system including astronauts' safety, and sometimes results in even rocket structure breakups. Active controls have applied to reduce such a dangerous pogo oscillation and a kind of shock absorbers have been installed for the safety of astronauts and payloads by mitigating thrust oscillation.⁵² However, the thrust oscillation is still one of the most difficult problems since it is related to engine pressure and fuel delivery system such as fuel/oxidizer pump and injectors and the study on the reason and the influence on the whole system is still required.

Here, the influence of thrust oscillation on the liquid fuel tank is investigated. In a practical rocket system, thrust oscillation is composed of many different frequencies and magnitudes produced by different parts of rocket.⁵² The present study focused on the parametrically excited liquid fuel surface wave, namely Faraday wave where the vertical sinusoidal oscillation in Eq. (41) is applied to liquid/air in an open container of Figure 18.

$$a = g_0(1 + \alpha \cdot \sin(2\pi ft)) \quad (41)$$

where, g_0 is standard gravity, α is oscillation magnitude ratio, and f is forcing frequency.

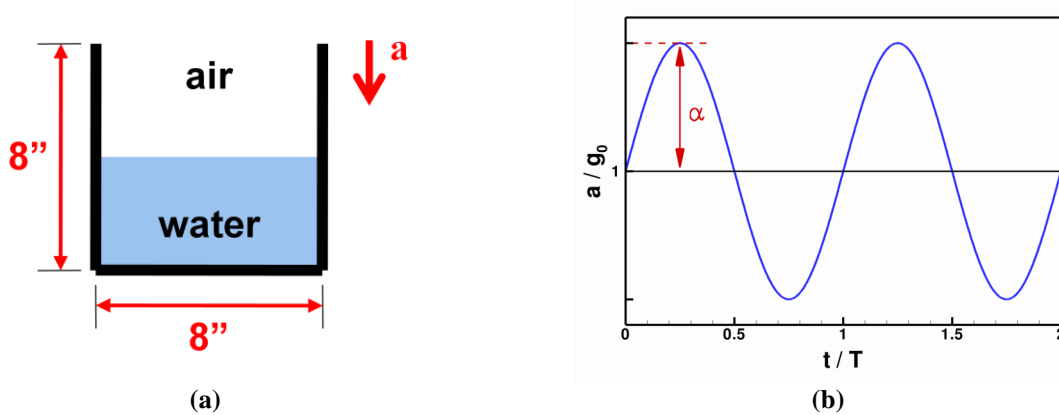


Figure 18. Geometric configuration of Faraday waves (a) 2D configuration (b) sinusoidal acceleration

According to dispersion relation for infinite depth periodic waves in Eq. (42), the surface waves become unstable by restoring forces of either gravitational effects or surface tension effects.⁵³

$$w^2 = gk + \left(\frac{\sigma}{\rho}\right)k^3 \quad (42)$$

Here, w is angular frequency of the surface waves, g is local acceleration, and k is the wave number. At low wave number, the gravitational effect dominates whereas surface tension effect does at high wave number. The crossover wave frequency, $w_c = (4g^3\rho/\sigma)^{1/4}$, is determined from the crossover wave number, $k_c = \sqrt{g\rho/\sigma}$, where the effects are equal. When the liquid fuel tank is parametrically excited with forcing angular frequency $w_f = 2\pi f$, the energy is pumped into waves of angular frequency $w = w_f/2, w_f, 3w_f/2, \dots$. The non-dimensionalized forcing frequency and forcing acceleration is defined from dimensional study in Eqs. (43) and (44).

$$w^* = \frac{w_0}{(\sigma/\rho)^{1/2}v^3} \quad (43)$$

$$a^* = \frac{a}{(\sigma/\rho)^{1/2}v^4} \quad (44)$$

An open container with a depth of 8" and a diameter (or width in 2D) of 8" in Figure 18 is simulated to compare the present study with experimental results conducted by Goodridge.⁵⁴⁻⁵⁶ Water and water-glycerin solutions are used to investigate the liquid surface stability changing the fluid viscosity. The contact angle between liquid and solid is assumed 0° as they are in the experiment. 2D simulations are first conducted to check the overall tendency of stability due to their computational efficiency, and 3D results are compared with 2D simulation at the same condition.

Figure 19 shows the 2D liquid surface wave evolution in time for water/air fluids as oscillation magnitude increases at a given forcing frequency $f = 20 \text{ Hz}$. With small acceleration, it shows flat surface wave in Figure

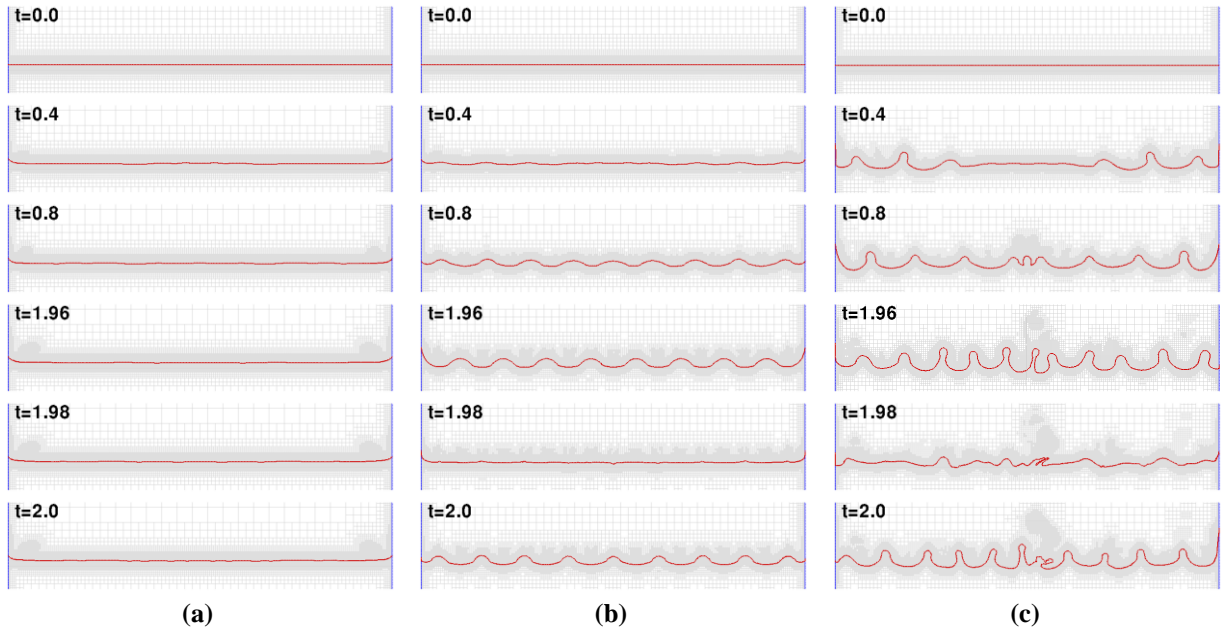


Figure 19. The features of surface wave stability according to forcing acceleration. Water/air fluids are vertically oscillated at a forcing frequency $f=20$ Hz. The flat liquid surface waves becomes periodic standing waves as acceleration increases, and shows the transition to unsteady aperiodic states with upward jets and droplets when the acceleration exceeds a certain threshold acceleration. (a) $\alpha=0.1$ (b) $\alpha=0.5$ (c) $\alpha=1.3$

19(a), but it becomes periodic standing wave in Figure 19(b) as forcing acceleration increases. The stable periodic surface waves are transformed to unstable aperiodic state, and shows some upward jets in Figure 19(c) when the acceleration exceeds a certain threshold level, which is called threshold acceleration. Goodridge defined the threshold acceleration as the acceleration level where two droplets detected within ten seconds⁵⁵ while Das defined it as jet forming or wave crest pinching off, followed by irregular motion.⁵⁷ In this study, jet forming is assumed as a criterion of threshold acceleration similar to Das's definition.

The threshold acceleration is investigated by changing the forcing frequency for fluids of different viscosity in Figure 20, where the threshold acceleration increases with forcing frequency. The present 2D simulations show higher threshold acceleration than Goodridge's experiment since 2D jets or droplets have larger size/mass, and thus require larger energy from parametrical excitation to form jets in 2D than 3D experiment. However, the overall trend of increasing threshold acceleration with increasing forcing frequency matches well with the experiments by Goodridge.⁵⁵ 80% Glycerin-water solution which has 50 times kinematic viscosity (ν) of water is also simulated to investigate the effects of viscosity. The higher viscosity of fluid makes the liquid surface waves more stable, and results in higher threshold acceleration to create unstable aperiodic wave states with upward jets. Another feature of higher viscosity fluid is it creates longer filament-like jets than lower viscosity fluid, which is shown in Figure 21.

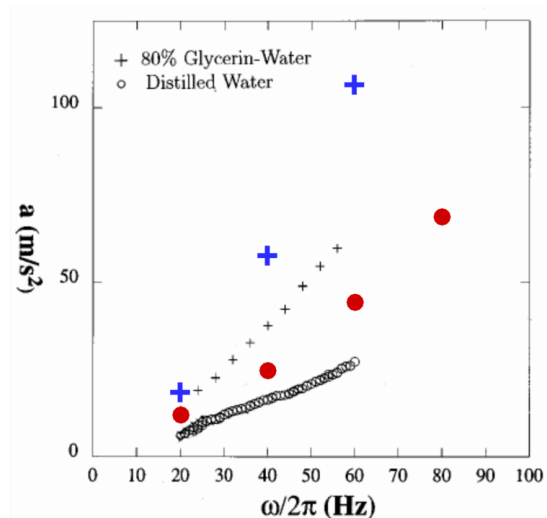


Figure 20. Threshold acceleration for fluids of different viscosity. The red and blue symbols are from the present 2D simulations.

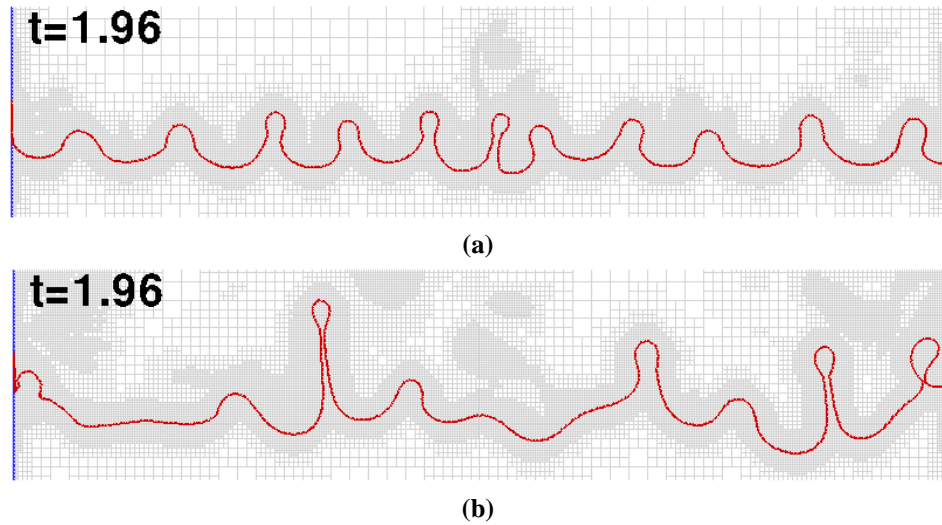


Figure 21. The effect of fluid viscosity at forcing frequency $f=20$ Hz. Higher viscosity of fluid shows longer filament-like jets with higher threshold acceleration. (a) water, $\alpha=1.3$ (b) 80% glycerin-water ($\nu = 50 \times \nu_{water}$), $\alpha=2.3$

In Figure 20, a 2D simulation shows higher threshold acceleration than 3D experiment. The comparison between 2D and 3D computations of the Faraday waves with the same acceleration ($\alpha = 0.3$) at the same forcing frequency ($f=10\text{Hz}$) is conducted and the difference is shown in Figure 22. At the same condition, 2D simulation results in periodic standing waves in Figure 22(a), but 3D simulation shows unsteady aperiodic state with very large upward jet on the centerline of container. The upwards jets require smaller energy in 3D since the mass to push up is smaller in 3D than 2D. Also, the concentration of waves at the center of axis makes largest upward jet first in the center of axis, and results in lower threshold acceleration in 3D.

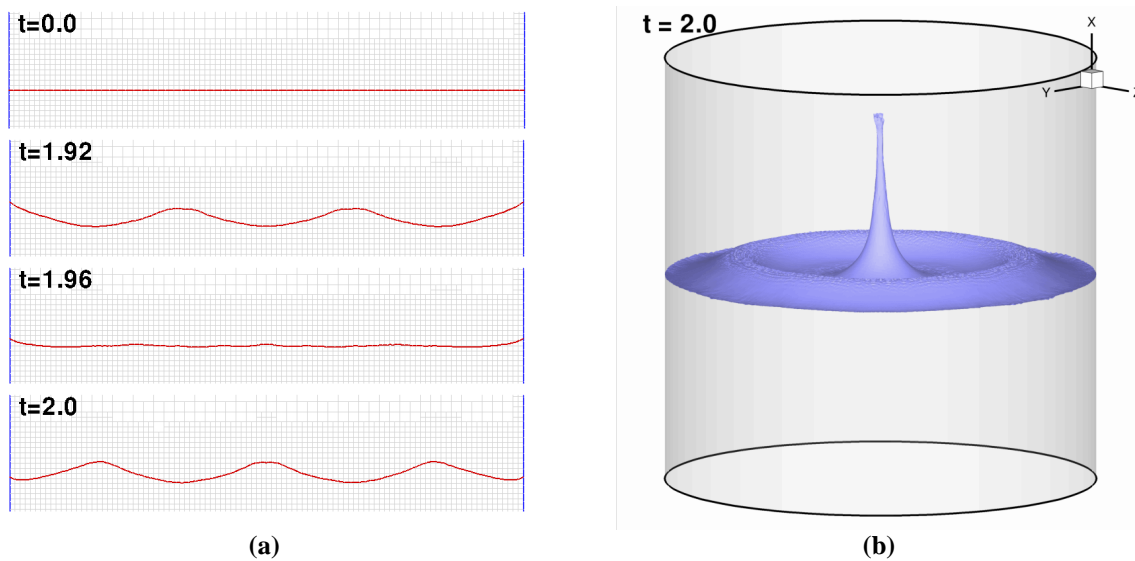


Figure 22. Comparison between 2D and 3D simulation at the same condition ($f=10$ Hz and $\alpha=0.3$) for water/air case. (a) 2D (b) 3D

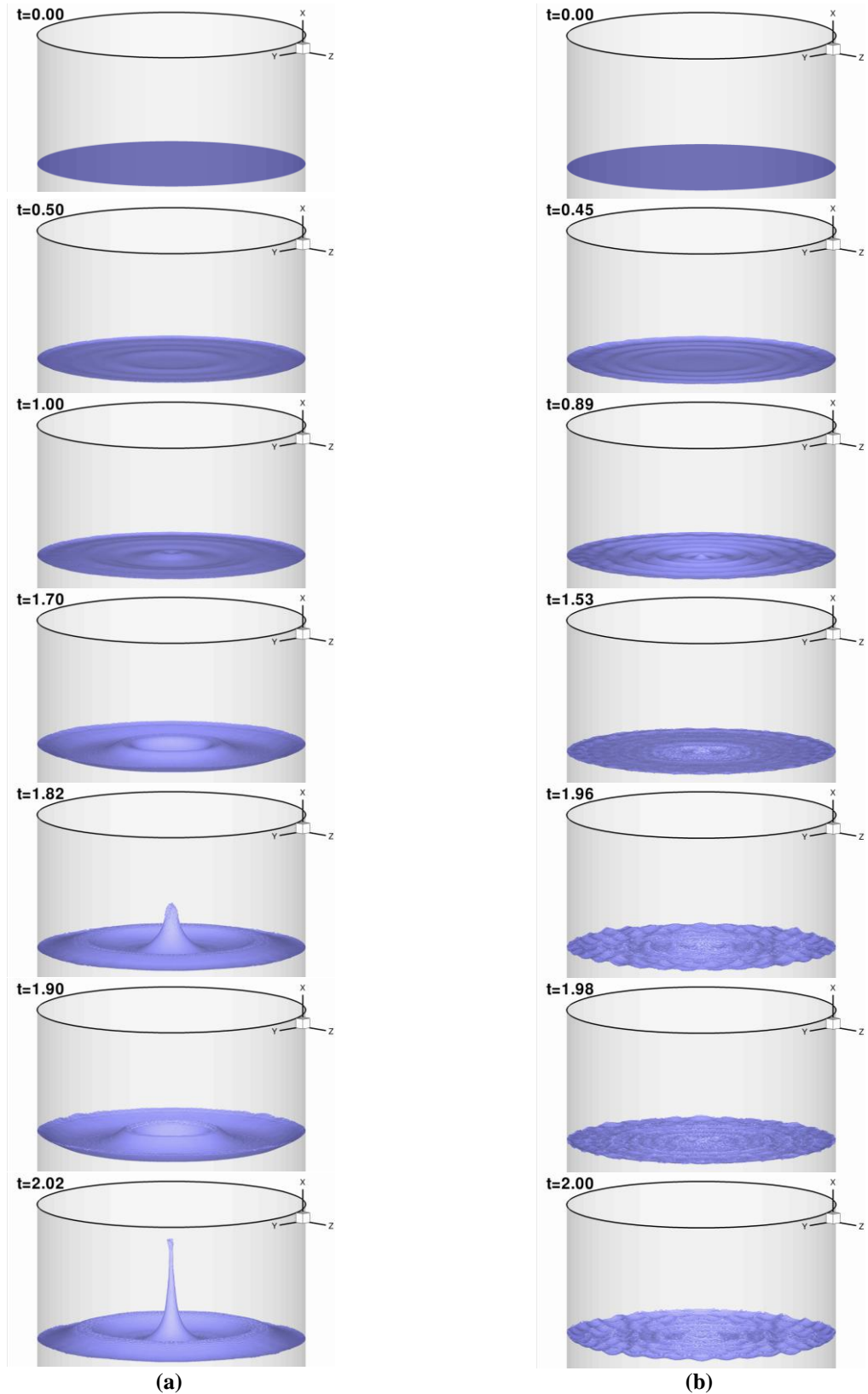


Figure 23. Comparison between gravity waves and Capillary-transition waves in 3D simulation for water/air case. (a) $f=10$ Hz and $\alpha=0.3$ (b) $f=20$ Hz and $\alpha=0.6$

According to the dispersion relation for infinite depth periodic waves in Eq. (42), the gravity wave at low angular frequency is transformed to Capillary wave where the surface tension effects become more important than gravity effects as forcing frequency increases. Figure 23(a) shows an example of gravity wave at low frequency ($f=10$ Hz) which is lower than the crossover forcing frequency ($f_c = 2w_c/(2\pi) = 27\text{Hz}$ in water/air case). The transition from gravity wave into Capillary wave is shown in Figure 23(b), where the large upward jet in the center of axis doesn't exist, and smaller multiple jets are observed in multiple locations. A maximum of 100 grid points along the tank diameter are used with locally adaptive grid to capture the surface wave motions. The present results show that 2D simulation is not enough to capture the non-axisymmetric phenomena, as shown in Figure 23(b), and finer 3D grids is required. It is noted that a higher forcing frequency results in a smaller wave length, and consequently smaller multiple jets.

The non-dimensionalized threshold acceleration ($a^* = av^4/(\sigma/\rho)^3$) is plotted with non-dimensionalized angular forcing frequency ($w^* = w_f v^3/(\sigma/\rho)^2$) in Figure 24 although 2D simulation shows lower threshold acceleration than 3D simulation. The present study shows the overall trend matches very well with the experiment by Goodridge, where the threshold acceleration shows $w_0^{4/3}$ dependence for lower-viscosity region ($w^* < 10^{-5}$), and $w_0^{3/2}$ dependence for higher-viscosity region ($w^* > 10^{-5}$).

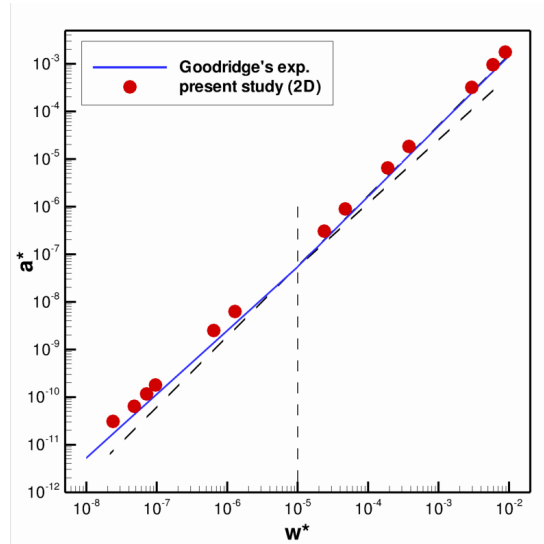


Figure 24. Nondimensionalized threshold acceleration as a function of nondimensionalized angular forcing frequency.

IV. Summary and Conclusions

In this paper, we report our efforts in developing a multi-scale three-dimensional adaptive Cartesian grid method for spacecraft fuel tank flow. The method utilizes Lagrangian, marker points representing a triangulated surface for both tracking interfaces separating fluid constituents and arbitrarily shaped solid geometries. The governing equations are solved on the Eulerian, Cartesian grid with automated local grid adaptation to capture the flow features with desirable resolution. These components are coupled to capture interfacial flow dynamics for various applications. In summary, the following key ingredients are developed and incorporated:

- (1) The marker-based data structure that represents and tracks fluid/fluid interface and/or solid boundary;
- (2) A smeared surface tension forces on interfaces separating fluid phases;
- (3) Modeling solid boundary using marker-based surface along with a ghost cell treatment around two- and three- dimensional irregular object;
- (4) An improved contact line modeling at tri-junction locations where fluid-liquid-gas phases all meet;
- (5) Local adaptive Cartesian grid method for effective computation.

To highlight these features, (i) sloshing motion and dynamics of cryogenic propellant in a spacecraft fuel tank and (ii) liquid fuel surface stability under spacecraft thrust oscillation by vertical sinusoidal oscillation of the gravitational acceleration are simulated. The simulations broadly agree with the experimental observation. Specifically, the following observations can be made.

- (1) The sloshing flow exhibits complex interface locations and movement, which significantly affects the handling of the fuel in a tank;
- (2) For the surface wave simulations:
 - (i) As the forcing oscillation magnitude increases, the surface wave exhibits periodic standing waves, and reaches aperiodic unstable states with upward jets and droplets when it exceeds a certain threshold value, namely threshold acceleration;
 - (ii) The threshold acceleration increases with forcing frequency;
 - (iii) The threshold acceleration increases with viscosity of fluid;
 - (iv) A higher viscosity fluid exhibits longer filament-like upward jets with higher threshold acceleration;

- (v) For higher viscosity fluids, the viscosity becomes an important parameter, and $a^* \sim (w^*)^{3/2}$ scaling is observed from both Goodridge's experiment and the present simulations;
- (vi) 2D simulation shows higher threshold acceleration than 3D simulation;
- (vii) In 3D simulation, the capillary wave is observed with high forcing frequency while the gravity wave is with low forcing frequency.

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 benefited from the discussion with John Peugeot and Jeff West of NASA Marshall Space Flight Center.

References

- ¹Toole, L.E. and L.J. Hastings, "An Experimental Study of the Behavior of a Sloshing Liquid Subjected to a Sudden Reduction in Acceleration," NASA TM X-53755, 1968.
- ²Veldman, A.E.P., J. Gerrits, R. Luppens, J.A. Helder, and J.P.B. Vreeburg, "The numerical simulation of liquid sloshing on board spacecraft," *Journal of Computational Physics*, 2007. 224(1): p. 82-99.
- ³Uzgoren, E., R. Singh, J. Sim, and W. Shyy, "Computational modeling for multiphase flows with spacecraft application," *Progress in Aerospace Sciences*, 2007. 43(4-6): p. 138-192.
- ⁴Shyy, W., 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). 2007, New York: Dover.
- ⁵Osher, S. and R.P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. 2002: Springer.
- ⁶Scardovelli, R. and S. Zaleski, "Direct numerical simulation of free-surface and interfacial flow," *Annual Review of Fluid Mechanics*, 1999. 31(1): p. 567-603.
- ⁷Tryggvason, G., 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," *Journal of Computational Physics*, 2001. 169(2): p. 708-759.
- ⁸Tryggvason, G. and A. Prosperetti, eds. *Computational Methods for Multiphase Flow*. 2007, Cambridge University Press.
- ⁹Wasekar, V.M. and R.M. Manglik, "Short-time-transient surfactant dynamics and Marangoni convection around boiling nuclei," *Journal of Heat Transfer*, 2003. 125(5): p. 858-866.
- ¹⁰Perot, B. and R. Nallapati, "A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows," *Journal of Computational Physics*, 2003. 184(1): p. 192-214.
- ¹¹Losasso, F., R. Fedkiw, and S. Osher, "Spatially adaptive techniques for level set methods and incompressible flow," *Computers and Fluids*, 2006. 35: p. 995-1010.
- ¹²Francois, M.M., 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," *Journal of Computational Physics*, 2006. 213(1): p. 141-173.
- ¹³Enright, D., F. Losasso, and R. Fedkiw, "A fast and accurate semi-Lagrangian particle level set method," *Computers and Structures*, 2005. 83(6-7): p. 479-490.
- ¹⁴Sussman, M., K.M. Smith, M.Y. Hussaini, M. Ohta, and R. Zhi-Wei, "A sharp interface method for incompressible two-phase flows," *Journal of Computational Physics*, 2007. 221(2): p. 469-505.
- ¹⁵Chang, C.H. and M.S. Liou, "A robust and accurate approach to computing compressible multiphase flow: Stratified flow model and AUSM+-up scheme," *Journal of Computational Physics*, 2007. 225(1): p. 840-873.
- ¹⁶Uzgoren, E., 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.
- ¹⁷Uzgoren, E., 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.
- ¹⁸Glimm, J., 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.*, 1998. 35(1).
- ¹⁹Glimm, J., J.W. Grove, X.L. Li, and D.C. Tan, "Robust Computational Algorithms for Dynamic Interface Tracking in Three Dimensions," *SIAM Journal on Scientific Computing*, 1999. 21(6): p. 2240-2256.
- ²⁰Singh, R. and W. Shyy, "Three-dimensional adaptive Cartesian grid method with conservative interface restructuring and reconstruction," *Journal of Computational Physics*, 2007.
- ²¹Peskin, C.S., "The immersed boundary method," *Acta Numerica*, 2003. 11: p. 479-517.
- ²²Francois, M. and W. Shyy, "Computations of Drop Dynamics with the Immersed Boundary Method, Part 1: Numerical Algorithm and Buoyancy-Induced Effect," *Numerical Heat Transfer: Part B: Fundamentals*, 2003. 44(2): p. 101-118.
- ²³Ye, T., R. Mittal, H.S. Udaykumar, and W. Shyy, "An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries," *Journal of Computational Physics*, 1999. 156(2): p. 209-240.

- ²⁴Leveque, R.J. and Z. Li, "The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources," *SIAM Journal on Numerical Analysis*, 1994. 31(4): p. 1019-1044.
- ²⁵Li, Z. and M.C. Lai, "The Immersed Interface Method for the Navier-Stokes Equations with Singular Forces," *Journal of Computational Physics*, 2001. 171(2): p. 822-842.
- ²⁶Liu, X.D., R.P. Fedkiw, and M. Kang, "A boundary condition capturing method for Poisson's equation on irregular domains," *Journal of Computational Physics*, 2000. 160(1): p. 151-178.
- ²⁷Kang, M., R.P. Fedkiw, and X.D. Liu, "A Boundary Condition Capturing Method for Multiphase Incompressible Flow," *Journal of Scientific Computing*, 2000. 15(3): p. 323-360.
- ²⁸Marella, S., S. Krishnan, H. Liu, and H.S. Udaykumar, "Sharp interface Cartesian grid method I: An easily implemented technique for 3D moving boundary computations," *Journal of Computational Physics*, 2005. 210(1): p. 1-31.
- ²⁹Aftosmis, M.J., "Solution adaptive Cartesian grid methods for aerodynamic flows with complex geometries," *Lecture notes, von Karman Institute for Fluid Dynamics, Series*, 1997. 2.
- ³⁰Yang, J. and E. Balaras, "An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries," *Journal of Computational Physics*, 2006. 215(1): p. 12-40.
- ³¹Gilmanov, A. and F. Sotiropoulos, "A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies," *Journal of Computational Physics*, 2005. 207(2): p. 457-492.
- ³²Mittal, R., H. Dong, M. Bozkurtas, F.M. Najjar, A. Vargas, and A.v. Loebbecke, "A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries," *J. Comput. Phys.*, 2008. 227(10): p. 4825-4852.
- ³³Shin, S., S.I. Abdel-Khalik, and D. Juric, "Direct three-dimensional numerical simulation of nucleate boiling using the level contour reconstruction method," *International journal of multiphase flow*, 2005. 31(10-11): p. 1231-1242.
- ³⁴Juric, D. and G. Tryggvason, "Computations of boiling flows," *Int. J. Multiphase Flow*, 1998. 24(3): p. 387-410.
- ³⁵Singh, R., 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.
- ³⁶Singh, R., *Three-dimensional marker-based multiphase flow computation using adaptive cartesian grid techniques*, in *Mechanical and Aerospace Engineering*. 2006, University of Florida: Gainesville. p. 144.
- ³⁷Stockie, J.M., *Analysis and Computation of Immersed Boundaries, with Application to Pulp Fibres*, in *Institute of Applied Mathematics*. 1997, University of British Columbia.
- ³⁸Engquist, B., A.K. Tornberg, and R. Tsai, "Discretization of Dirac delta functions in level set methods," *Journal of Computational Physics*, 2005. 207(1): p. 28-51.
- ³⁹Uzgoren, E., J. Sim, and W. Shyy. *Marker-Based, 3-D Adaptive Cartesian Grid Method for Multiphase Flow around Irregular Geometries*. in *46th AIAA Aerospace Sciences Meeting and Exhibit*. 2008. Reno, NV.
- ⁴⁰Singh, R., 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. Reno, NV.
- ⁴¹Wang, Z. and Z.J. Wang, "The Level Set Method on Adaptive Cartesian Grid For Interface Capturing," *AIAA Paper No*, 2004. 82.
- ⁴²Francois, M. and W. Shyy, "Micro-scale drop dynamics for heat transfer enhancement," *Progress in Aerospace Sciences*, 2002. 38(4): p. 275-304.
- ⁴³de Sousa, F.S., 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 multi-fluid flows with free surfaces," *Journal of Computational Physics*, 2004. 198(2): p. 469-499.
- ⁴⁴Al-Rawahi, N. and G. Tryggvason, "Numerical simulation of dendritic solidification with convection: two-dimensional geometry," *Journal of Computational Physics*, 2002. 180(2): p. 471-496.
- ⁴⁵Al-Rawahi, N. and G. Tryggvason, "Numerical simulation of dendritic solidification with convection: Three-dimensional flow," *Journal of Computational Physics*, 2004. 194(2): p. 677-696.
- ⁴⁶Shin, S. and D. Juric, "Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity," *Journal of Computational Physics*, 2002. 180(2): p. 427-470.
- ⁴⁷Patankar, S.V., *Numerical Heat Transfer and Fluid Flow*. 1980: McGraw-Hill.
- ⁴⁸Shyy, W., *Computational modeling for fluid flow and interfacial transport*. 1994: Elsevier New York.
- ⁴⁹Dussan, V.E.B., "The moving contact line: slip boundary conditions," *Journal of Fluid Mechanics*, 1977. 77: p. 665-684.
- ⁵⁰Hocking, L.M., "A moving interface on a rough surface," *Journal of Fluid Mechanics*, 1976. 76: p. 801-817.
- ⁵¹Huang, H., D. Lian, and B. Wetton, "Computation of a moving drop/bubble on a solid surface using a front tracking method," *Communications in Mathematical Physics*, 2004. 2(4): p. 535-552.
- ⁵²Yang, V. and W.E. Andersen, eds. *Liquid Rocket Engine Combustion Instability*. 1995, AIAA.
- ⁵³Landau, L.D. and E.M. Lifshitz, *Fluid Mechanics*. 1987, New York: Pergamon Press.
- ⁵⁴Goodridge, C.L., W.T. Shi, and D.P. Lathrop, "Threshold Dynamics of Singular Gravity-Capillary Waves," *Physical Review Letters*, 1996. 77(22): p. 4692.
- ⁵⁵Goodridge, C.L., W.T. Shi, H.G.E. Hentschel, and D.P. Lathrop, "Viscous effects in droplet-ejecting capillary waves," *Physical Review E*, 1997. 56(1): p. 472.
- ⁵⁶Goodridge, C.L., H.G.E. Hentschel, and D.P. Lathrop, "Breaking Faraday Waves: Critical Slowing of Droplet Ejection Rates," *Physical Review Letters*, 1999. 82(15): p. 3062.
- ⁵⁷Das, S.P. and E.J. Hopfinger, "Parametrically forced gravity waves in a circular cylinder and finite-time singularity," *Journal of Fluid Mechanics*, 2008. 599(-1): p. 205-228.