NUMERICAL METHODS FOR SIMULATING DROPLET DYNAMICS IN MICROFLUIDIC DEVICES

by

Eric Charles Cauble

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Mechanical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

May, 2016

©COPYRIGHT

by

Eric Charles Cauble

2016

All Rights Reserved

DEDICATION

I dedicate this thesis to my best friend and beautiful wife Jude and our two beautiful daughters Roma and Marilla. I am truly grateful for their overwhelming support and encouragement throughout the durartion of this research.

TABLE OF CONTENTS

TABLE OF CONTENTS - CONTINUED

LIST OF FIGURES

LIST OF FIGURES - CONTINUED

ABSTRACT

Proton exchange membrane (PEM) fuel cells are of beneficial interest due to their capability of producing clean energy with zero emissions. An important design challenge hindering the performance of fuel cells is controlling water removal to maintain a hydrated membrane while avoiding excess water that may lead to channel blockage. Fuel cell water management requires a detailed knowledge of multiphase flow dynamics within microchannels. Direct observation of these gas-liquid flows is difficult due to the small scale and viewing obstructions of the channels within the fuel cell. Instead, this work uses a computational fluid dynamics (CFD) approach to compute the dynamics and formation of droplets in fuel cell channels by imposing contact angles and computing the curvature at the interface along the wall boundary.

This thesis focuses on developing and implementing two methods to assist in computing the dynamical behavior of droplets in PEM fuel cells. The contact angle method leverages a conservative volume-of-fluid (VOF) formulation coupled with novel methodologies to impose and track dynamic contact angles. In particular, it is shown that variation of the contact hysteresis angle influences the wetting properties of the droplet and significantly impacts water transport throughout a fuel cell channel.

The proposed curvature scheme employs the method of least squares to fit an implicit polynomial function to a cloud of points created at the interface. The points are constructed from the volume of fluid (VOF) representation of the phase interface and the curvature is computed explicitly from this best fit polynomial function. It is shown that the new curvature method is second-order accurate and can be used to compute the curvature of a droplet along a boundary with prescribed contact angles.

This thesis presents details of the numerical approaches used to implement contact angles and to compute the curvature based on novel methods The developed numerical methods are then used to simulate a droplet contained in a channel with an incoming gas flow and discuss the results relevant to water management in PEM fuel cells.

1

INTRODUCTION

Understanding multiphase flows are important to many processes and encompass a wide range of engineering disciplines. Droplet dynamics [1], coating processes [2], atomization [3], and gas flow simulations [4] are a few examples where multiphase flow models are used. A topic that has become increasingly popular recently is how water interacts with the gas flow and solid channel walls within Proton Exchange Membrane (PEM) fuel cells [5].

Proton exchange membrane fuel cells have become a popular alternative energy generating device due to their efficiency to produce energy at low operating costs and zero emissions [6]. Energy consumption is increasing dramatically while carbon resources across the globe are depleting rapidly [7]. Carbon-based fuels emit toxic greenhouse gases and are responsible for adverse impacts on public health and environmental effects [8]. Clean energy sources such as PEM fuel cells hope to eventually replace carbon-based energy sources.

PEM fuel cells produce electricity through a chemical reaction between a fuel, usually hydrogen gas (H_2), and an oxidizing agent such as air. As a result, an oxygen reduction reaction (ORR) occurs producing water within the fuel cell channel, Fig. 1.1. While some water is beneficial for the hydration of the membrane, which assists in the transportation of protons, an accumulation of water within the channel can cause channel flooding which restricts the flow of gas and greatly reduces the efficiency of the fuel cell [9]. The pressure drop associated with the accumulation of water in a fuel cell channel has been studied recently [10] and can be predicted based on certain

flow conditions but it is difficult to predict the behavior of the water within the fuel cell's channels.

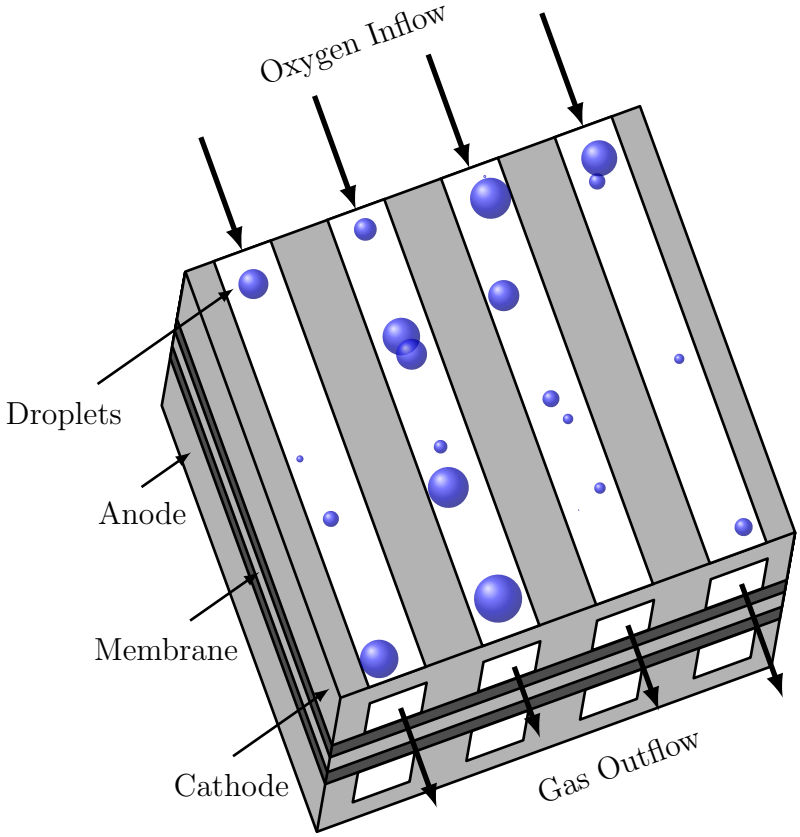


Figure 1.1: Schematic of fuel cell including droplets in the fuel cell channels.

Several attempts to understand the flow and interactions of the water within the channels have been conducted [11, 12, 13, 14]. These methods prove to be very costly and require specialized equipment. One such example was performed by a team at the University of Waterloo in Ontario Canada. They observed the amount of water accumulating within the channel using neutron radiography due to its ability to accurately detect hydrogen atoms in the water [15].

The difficulty in visualizing the flow of water in PEM fuel cell channels using specialized techniques [5] has prolonged the advancement of fuel cell development

and has hindered validation to proposed models. Numerical simulations of the flow are now being considered to hopefully better understand the transport of water flow through the channels. While numerical simulations have the potential to assist in understanding this phenomena, further work is needed in order to accurately depict the flow and the interactions with the gas and the porous membrane surfaces. The difficulty in simulating the flow through a channel in a microfluidic device is the relative scale of the flow. Most simulations are not able to accurately delineate the flow at the microscale due to the macroscale assumptions that are imposed into the fundamental equations governing fluid flow.

Thesis Outline

The following chapter will carry on an inquiry of the some of the key features related to droplet flow in a microfluidic devices and the theories used to describe contact angles. Following that discussion, Chapter 4 will present the numerical methods used to model the flow of water in microfluidic devices, such as fuel cell channels, and the altercations necessary to accurately model fluid flow on a microscale. The results and discussion of the implementation of the methods will be then be presented. Thereafter, ideas for future work will be proposed.

4

BACKGROUND

PEM Fuel Cells

Water management is crucial for a efficient operating PEM fuel cell. One such instance where water is important is the hydration of the polymer electrolyte membrane. If water is depleted from the membrane the resistance to proton conduction increases which limits the flow of protons to the cathode which are needed for the ORR (oxygen reduction reaction). However, an excess amount of water can also decrease the efficiency of the fuel cell by accumulating within the fuel cell channels which can block the flow of gas. Therefore, an understanding of the dynamic behavior of droplets in a gas flow channel is essential to determining the optimum gas flow and reaction times needed to optimize the fuel cells's efficiency.

Multiphase Flow

Multiphase flow is defined as a flow consisting of more than one state or phase. The phase or components of the flow are assumed to be well defined within the flow and are not taken to be well mixed.

No Slip Condition

A phenomenon that occurs in Newtonian liquids flows at the molecular liquid-solid interface is the no slip condition. The no slip condition states that the velocity of the fluid at the interface is equal to that of the solid. However, it is difficult to prove that this phenomenon occurs experimentally and is usually implemented in numerical models in order to simplify computations [16]. This is normally done by specifying

the slip length of the fluid. The slip length is defined as the minimum distance from the liquid-solid interface in which the velocity of the fluid is equal to that of the solid.

Contact Angles

Understanding the relationship between the motion of a droplet and contact lines is critical to accurately representing the flow of a droplet. Consider the simple sessile drop in Fig. 2.1, the adhesion of particles separating the phases can be described as a balance of forces acting on the droplet known as Young's equation. The triple point is defined as the point where the liquid, gas and solid interfaces intersect. Starting from the triple point, a line drawn tangent to the liquid interface represents the contact angle as shown by θ.

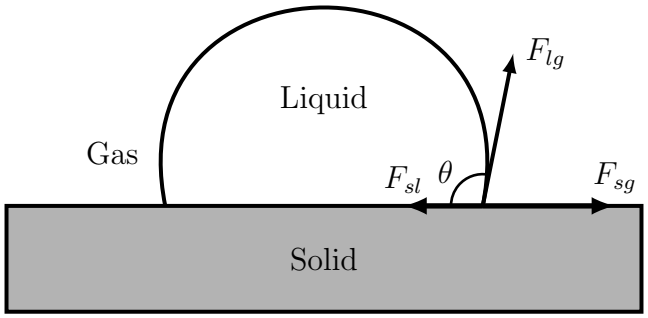


Figure 2.1: Triple point and contact line schematic of a sessile drop.

A simple explanation describing the orientation of a dynamic droplet on a solid surface can be done by applying a force balance on the water molecules. Near the interface of the droplet, the water molecules are influenced by intermolecular forces due to other phases such as gases or solids. One way to quantify this force is to observe the angle the interface of the droplet makes at a determined length scale and drawing a line tangent to the liquid interface which extends to the intersection of the liquid and solid interfaces. This angle is known as the contact angle and the

point where this angle intersects the solid boundary is known as the contact point. The wetting properties of a liquid droplet on a solid surface can be quantified by the contact angle. If the contact angle is obtuse, the droplet is said to be non-wetting and the solid is hydrophobic. If the contact angle is acute, the droplet is said to be wetting and the solid is hydrophilic.

A static droplet in equilibrium can be estimated using Young's equation

$$\cos(\theta) = \frac{F_{sl} - F_{sg}}{F_{lg}} \tag{2.1}$$

where θ is the contact angle and F represents the surface forces acting on the different phases: $s, l,$ and$g$ for solid, liquid, and gas. If $\cos(\theta)$ equals one, the liquid is described as being completely wetting (hydrophilic) and the energy of the solid is considered to be a high-energy solid. Similarly, if $\cos(\theta)$ equals zero, the liquid is said to be completely dewetting (hydrophobic). One of the major drawbacks of Eq. (2.1) is that it is only valid for macroscopic droplets [17] and it assumed that the droplet will exhibit the same contact angle characteristics for every experiment. This has been shown not to be true and the contact angle can vary 20° between experiments [18]. Instead, the static contact angle can take a range of values that vary between the advancing and receding angles known as the hysteresis angle. A modified Young's equation has been proposed which considers the line tension and radius of the droplet [17]. While this approach may be ideal for static droplets it is not ideal for dynamic contact angles.

A dynamic droplet is defined by an advancing contact angle (θ_{adv}) and a receding contact angle (θ_{rec}) as shown in Fig. 2.2. The difference between these angles is known as the hysteresis angle. For a droplet on an incline, the wetting area will remain unchanged until the contact angle exceeds the bounds given by θ_{adv} and θ_{rec}. If the

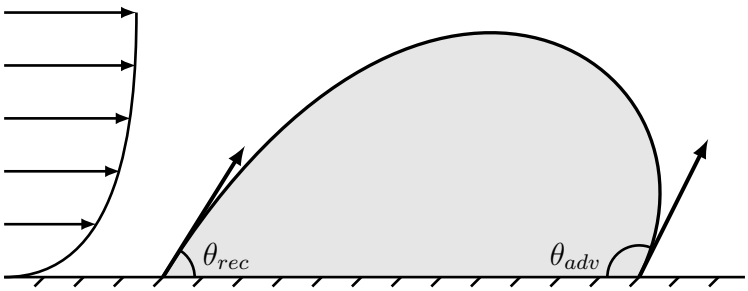


Figure 2.2: Advancing and receding contact angles for a dynamic droplet in a gas flow.

bounds are exceeded, the wetting area will change and the contact angle will equal that found experimentally for θ_{adv} and θ_{rec}. The advancing and receding contact angles are not related to the contact angles defined for a static droplet. Three sources that influence the hysteresis angle are: surface roughness, chemical contamination and solutes in the liquid [19].

Due to the no slip condition described above, the motion of a droplet across a solid surface is thought to be a rolling motion where new liquid particles make contact with the solid surface at the leading edge and remain in static contact with the solid surface throughout the duration of the wetting area until the receding edge. At the receding edge, the liquid particles are sheared from the solid surface [20]. However, if the force acting on the liquid particles is greater than the intermolecular forces acting on the liquid molecules, the droplet may leave a thin film at the trailing edge.

Accurately quantifying contact angles on solids are difficult due to all the influences that affect the contact angle. However, an accurate explication of dynamic contact angles are important to describing the behavior of a droplet. A difference in dynamic contact angles of a few degrees can have large impacts on the results of the flow. Macro scale flows are not as dependent on contact angles since larger forces such as gravity and other body forces dominate the behavior of the fluid.

However, for microscale flows, these body forces do not have the same impact on the flow and the flow is dominated mostly by surface tension and interfacial forces. Therefore, it is necessary to introduce another parameter, such as the dynamic contact angle, that describes the behavior of a droplet. As mentioned earlier, the underlying physics describing and calculating contact angles are vague and are based on the balance of cohesive forces at the microlevel. The simplest way to implement contact angles is to assign boundary conditions to the maximum and minimum contact angles based on experimental tests. There are three popular methods to experimentally measure contact angles: simple sessile droplet [21], Wilhelmy plate method [22], and axisymmetric drop shape analysis-profile (ADSA-P) [23].

9

CONTACT ANGLE THEORY

As important as contact lines are to describing the flow within a fuel cell channel, a complete understanding of the physics behind contact lines is not fully understood. With the recent advancements in technology, scientists and engineers are able to gain a better insight into this phenomena using neutron and other imaging techniques [5]. But how does one accurately model a dynamic droplet in a simulation so that the results can be used to design more efficient fuel cells?

The simplest method to describe the advancing and receding dynamic contact angles is to pre-define them as constant parameters throughout the entire flow. This assumption can produce large errors and greatly decrease the accuracy of the results. However, this method relies heavily on experimental data which is useful if the data exists for the particular phases one is trying to simulate.

Currently, there are two well known theories that attempt to explain the motion of the contact line and the dynamic contact angle. The first of which is the hydrodynamic theory on the mesoscopic layer which focuses on the energy losses due to viscous deformation [24]. The second is the molecular-kinetic theory at the microscopic layer which considers the absorption/desorption of fluid particles at the solid-liquid interface [25]. Both theories have their flaws and many attempts to relinquish these flaws have been attempted. One such approach is to combine the two theories in order to relate the contact line at the microscale to the macroscale [26].

Hydrodynamic Theory

A popular method used to describe the dynamics of the contact line of a droplet is the hydrodynamic theory. The dissipation of energy is said to be due to viscous

deformation at the liquid-gas interface at the mesoscale [27]. The advancing contact angle of the liquid at the microscale is assumed to be constant based on the influence of van der Waals forces between the liquid and solid. The droplet moves by deforming in a rolling type motion so that new liquid molecules overlap the water molecules that were previously in contact with the solid so that zero slip occurs at the interface. Similarly, the receding contact angle is a result of the seperation of van der Waals forces between the liquid and solid. Figure 3.1 shows the definition of the scales where the microscopic contact angle (θ_m), deformation, and dynamic advancing contact angle (θ_{adv}) are defined. The dashed line in Fig. 3.1 represents the extrapolated macroscopic contact angle to the solid surface.

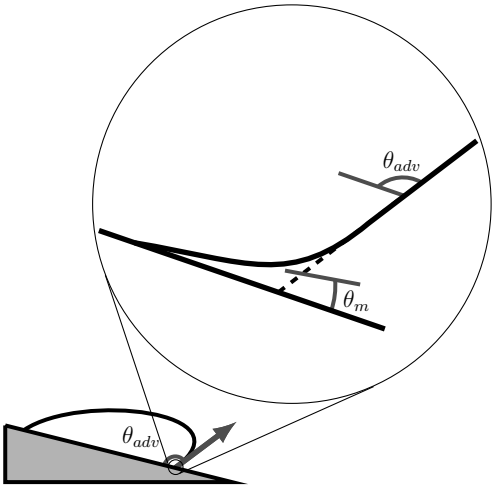


Figure 3.1: Schematic of the Hydrodynamic Theory.

A well known analytical correlation that was proposed to relate the advancing contact angle to distinguishable paramaters is known as the Hoffman-Jiang correlation [28]:

$$\frac{\cos(\theta_S) - \cos(\theta_{adv})}{\cos(\theta_S) + 1} = \tanh(4.96\ \mathrm{Ca}^{0.702}) \tag{3.1}$$

where θ_S is the static equilibrium contact angle, θ_{adv} is the advancing dynamic contact angle and Ca is the capillary number. The capillary number is given by

$$\mathrm{Ca} = \frac{\eta V}{\gamma_{lg}} \tag{3.2}$$

where η is the viscosity, V is the velocity and γ_{lg} is the liquid-gas interface tension.

An issue that arises in the hydrodynamic model is due to the definition of the contact line. The dynamic contact line in Fig. 3.1 is extrapolated to the solid surface and the angle is based on the amount of dissipation due to viscous deformation. This geometry can then lead to singularities in the surface tension force at the liquid interface since the gas velocity creates arbitrarily small contours near the triple point and the interface of the gas is undefined in this region [24].

Molecular-Kinetic Theory (MKT)

One of the major differences between the molecular-kinetic theory and the hydrodynamic theory is that MKT ignores energy dissipation due to viscous deformation [26]. Instead, MKT considers energy dissipation due to the absorption/desorption of fluid particles at the solid-liquid interface at the contact line [29]. As a result, the induced singularities due to the no slip condition along the wall is avoided. Another difference is the length scales for which the contact line is defined. MHK is based solely at the molecular scale and takes a statistical approach to predict the dynamics of the molecules at the triple point [30].

The MHK contact angle, which is considered a function of velocity for all scales, is a result of the disturbances due to the accumulation of molecules at the interface

of the liquid at the triple point [27]. This induces a surface tension force at the liquid interface given by

$$F_W = \gamma_{lg}(\cos(\theta_S) - \cos(\theta_D)) \ , \tag{3.3}$$

where γ_{lg} is the liquid-gas interfacial energy, θ_S is the equilibrium static contact angle and θ_D is the dynamic contact angle. The velocity of the contact line can then be determined by

$$U = 2\kappa^0\lambda \sinh\left(\frac{\gamma_{lg}\lambda^2(\cos(\theta_S) - \cos(\theta_D))}{2\mathrm{K_B}T}\right) \tag{3.4}$$

where λ is the distance between adsorption sights, k_B is the Boltzmann constant, T is the absolute temperature and κ^0 is the equilibrium frequency of the molecular displacements which is found experimentally. Solving Eq. (3.4) for the dynamic contact angle yields:

$$\cos(\theta_D) = \cos(\theta_S) \mp \frac{2\mathrm{K_B}T}{\gamma_{lg}\lambda^2} \sinh^{-1}\left(\frac{U}{2\kappa^0\lambda}\right) \ . \tag{3.5}$$

The negative and positive signs correspond to the advancing and receding contact line, respectively.

One of the disadvantages of the molecular-kinetic theory is that many of the parameters depend on experimental data. The accuracy of these parmaters is therefore dependent on the experimental equipment used and the method used to extrapolate this data.

So which theory is the preferred method? Many attempts and variations to both theories have been performed to try and solve this question but still there are controversies and discussions debating this question. The hydrodynamic model obeys

a no slip condition at the interface and it has been shown that in fact the surface of the liquid does translate in such a way that modeling the deformation as a rolling motion does hold true to some extent [31]. However, the hydrodynamic model is limited to energy losses due to viscous deformation. There has been some attempts to extend the theory to include other energy losses, such as Cox's law [27] but the fact that the contact line moves with a no slip boundary condition is said to violate the no slip condition [32].

The molecular-kinetic theory has some drawbacks as well. The biggest challenge is that there isn't a method included in the theory to relate the molecular scale to the hydrodynamics of the system [27]. One approach to alleviate this issue is to combine the molecular-kinetic theory and the hydrodynamic theory. Instead of assuming that microscopic contact angle (θ_m) is constant, MKT is used to predict θ_m based on the interactions of the solid and liquid as a function of velocity.

The combination of theories offers a way to describe the contact line of a droplet from the molecular scale to the mesoscale without neglecting dissipation of energy due to viscous deformation while also satisfying the no slip condition of the dynamic contact line. Another benefit to this combination is the control over parameters such as: temperature, wetting line friction, and viscosity.

Multiphase flows in microchannels are difficult to simulate accurately due to the scale of the flow, molecular forces, and energy losses within the flow on a microscale. However, with the right model, such as the combined model theory, and being able to adjust parameters to create an environment similar to a fuel cell, realistic results can be achieved. However, by adding these parameters to the model, the overall complexity of the model increases and it may be exceedingly difficult to accurately verify results needed for the definition of the parameters.

14

NUMERICAL METHODS

This chapter begins by discussing the Navier-Stokes equation and how it can be used to describe both the gas and liquid flows in a domain through jump conditions at the interface. The methods used to reconstruct the interface based on the volume of fluid (VOF) capturing scheme are discussed in detail thereafter. The implementation of the contact angle and a new curvature method are proposed based on methods used to reconstruct the interface.

Navier-Stokes Equations

The Navier-Stokes equations are used to describe the motion of a fluid in a domain based on conservation principles. For an incompressible flow the Navier-Stokes equation can be written as:

$$\frac{\partial \rho_i \boldsymbol{u}_i}{\partial t} + \nabla \cdot (\rho_i \boldsymbol{u}_i \otimes \boldsymbol{u}_i) = -\nabla p_i + \nabla \cdot \mu_i \big(\nabla \boldsymbol{u}_i + \nabla \boldsymbol{u}_i^T\big) + \boldsymbol{f}_i \tag{4.1}$$

where $\boldsymbol{u} = [u, v, w]$ is the velocity vector in three dimensions, t is time, ρ is density, p is pressure, μ is the dynamic viscosity, and $\boldsymbol{f}$ includes surface and body forces. The subscript i designates the phase of the fluid and takes values of g or l for gas and liquid, respectively.

The equation above has been written in both the gas and liquid phases. The difference in parameters of the phases are connected through jump conditions at the phase interface. For example, the jump in density and viscosity at the interface Γ are written as:

$$[\rho]_\Gamma = \alpha_{i,j}\rho_l + (1 - \alpha_{i,j})\rho_g \tag{4.2}$$

$$[\mu]_\Gamma = \alpha_{i,j}\mu_l + (1 - \alpha_{i,j})\mu_g, \tag{4.3}$$

respectively. In the absence of a phase change the velocity field is continuous

$$[\boldsymbol{u}]_\Gamma = 0. \tag{4.4}$$

The pressure is discontinuous due to contributions from surface tension and the normal component of the viscous stress, i.e.,

$$[p]_\Gamma = \sigma\kappa + 2[\mu]_\Gamma \boldsymbol{n}^{\mathrm{T}} \cdot \nabla \boldsymbol{u} \cdot \boldsymbol{n}, \tag{4.5}$$

where σ is the surface tension coefficient and κ is the interface curvature.

Volume-of-Fluid Scheme

The volume-of-fluid (VOF) method [33, 34] is a common interface capturing scheme. The volume of fluid scheme is an interface capturing algorithm which determines the interface by calculating the ratio of the liquid volume to cell volume [34]. Unlike interface tracking algorithms such as the level-set[35] or MAC methods[36], which determines the interface location based on a continuous level-set function or a distribution of marker particles throughout a flow [37], the VOF method captures the location of the interface by determining the volume of fluid in a cell and reconstructs the interface based on the liquid volume fraction $\alpha_{i,j}$ which is bounded by $0 \leq \alpha_{i,j} \leq 1$.

The advection of the fluid is based on the conservation of mass for an incompressible flow and is expressed as:

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\boldsymbol{u}_i \alpha_i) = 0 \tag{4.6}$$

where α is the volume fraction which tracks the amount of the liquid phase present in a computational cell and stores the ratio of liquid volume to cell volume. In a gas-liquid interfacial flow, α is unity if the cell is composed entirely of liquid and zero if the cell is composed entirely of gas. If α is greater than zero and less than unity, an interface exists which separates the phases within the cell.

Interface reconstruction of the fluid within a cell containing an interface is done by using the piecewise linear interface calculation method (PLIC) [33, 38], which is second-order accurate. The interface of the fluid within the cell is approximated by a straight line (2D) or plane (3D) and the position of this segment is dependent on the volume fraction of the fluid and the orientation of the interface is determined based on the approximated unit normal, Fig. 4.1, determined by ELVIRA [39].

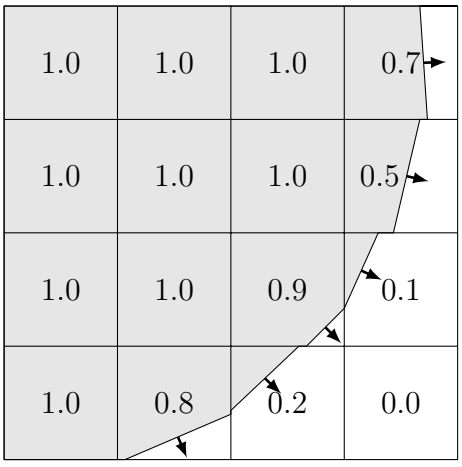


Figure 4.1: Example of a PLIC reconstruction of the interface in 2D.

The efficient least squares volume-of-fluid interface reconstruction algorithm (ELVIRA) method is used to calculate the normal of the linear interface based on volume fractions, [39]. ELVIRA determines the slope of the interface based on finite difference approximations of the liquid volume fraction contained in the neighboring cells.

Contact Angle

A way to introduce contact angles into a code is by defining them as upper and lower bounds which limit the maximum and minimum angle the interface of the liquid can make with the wall boundary. The upper bound represents the advancing contact angle (θ_{adv}) and the lower bound represent the receding contact angle (θ_{rec}). The angles between the upper and lower bounds giving by θ_{adv} and θ_{rec} represent the hysteresis angle. The hysteresis angles defines a range of contact angles that the interface of the liquid can make with the solid wall where the displacement of the liquid interface along the boundary remains stationary, Fig. 4.2.

The implementation of the contact angle was done by inspecting the VOF along the boundary and determining whether the computational cell contained an interface. If an interface was found to exist in a cell, α_{adv} and α_{rec} were computed based on the inputs θ_{adv} and θ_{rec}, the contact point, and n_x and n_z calculated using ELVIRA. If the volume fraction was within α_{max} and α_{min}, the contact point and the given contact angle would remain unchanged. Figure 4.3 shows the bounds given by the advancing and receding contact angles and a fixed contact point. The normal points in the direction of the gas phase from the liquid phase. If α was found to fall outside the bounds given by α_{adv} and α_{rec}, the contact point is advected accordingly in order to match α and either θ_{adv} or θ_{rec} depending if α was greater than α_{adv} or less than α_{rec}, respectively, Fig. 4.4.

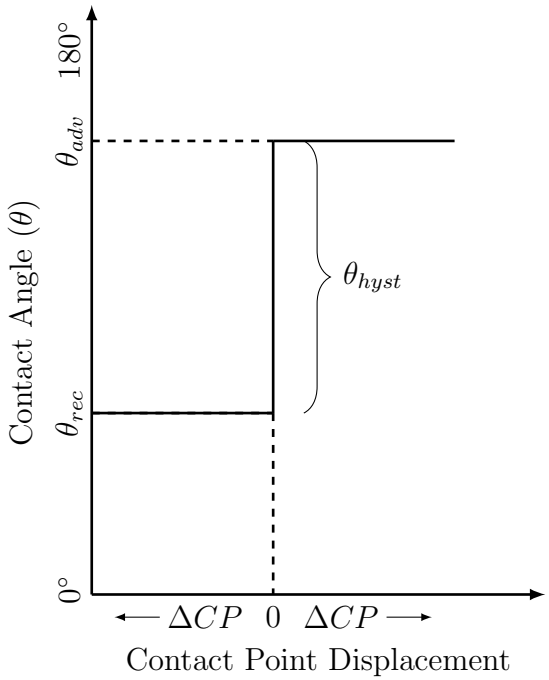


Figure 4.2: Hysteresis angle and the displacement of the contact point.

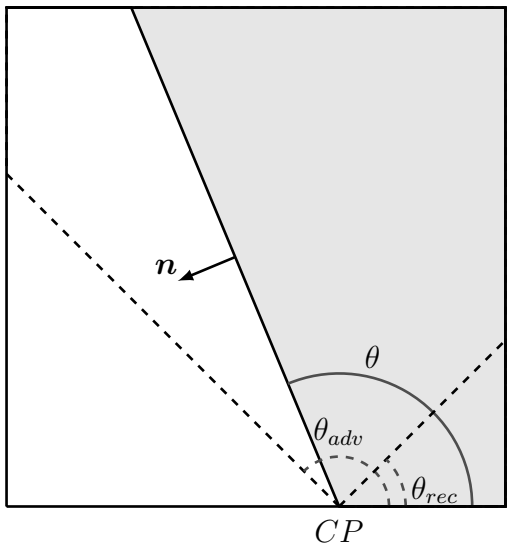


Figure 4.3: Contact angle specification. Grey represents the liquid.

Contact Angle Implementation

If the volume fraction (α) exceeds the bounds given by α_{max} and α_{min} based on θ_{adv} and θ_{rec}, respectively, then a new position of the contact point and normal in the y-direction is calculated. The new normal is based on the specification of either the advancing or receding contact angle and is determined by finding the xy-trace (Fig. 4.5):

$$\cos(\theta) = \frac{\boldsymbol{n} \cdot \boldsymbol{j}}{|\boldsymbol{n}|\,|\boldsymbol{j}|} \tag{4.7}$$

where $\boldsymbol{n} = \langle n_x, n_y, n_z \rangle$ is the normal vector and $\boldsymbol{j}$ is the unit normal in the y-direction. Solving Eq. (4.7) for n_y yields:

$$n_y = \frac{\sqrt{n_x^2 + n_z^2}}{tan(\theta)} \tag{4.8}$$

where θ is either the advancing or receding contact angle.

The PLIC is then reconstructed with the new normal using analytical solutions [40] based on the normal and the liquid volume fraction to determine the equation describing the plane:

$$n_x x + n_y y + n_z z = d \tag{4.9}$$

where d is the minimum distance from the plane to the origin. Based on the equation of the plane a new contact point is determined.

Contact Point

In order to advect the PLIC along the boundary, a contact point (CP) is specified. As mentioned previously, if α is greater than α_{max} then θ equals θ_{adv} and the contact point is displaced until α_{max} plus the change in volume from the new

contact point to the old contact point equals α. Figure 4.4 shows a situation when θ is greater than θ_{adv}. The shaded area, beginning at the CP_{old}, bounded between θ and θ_{adv} is the difference in volume fractions α and α_{max}. This difference is then used to advect the CP until the volume fraction of the shaded rectangular region equals the difference in volume fractions while keeping θ equal θ_{adv}.

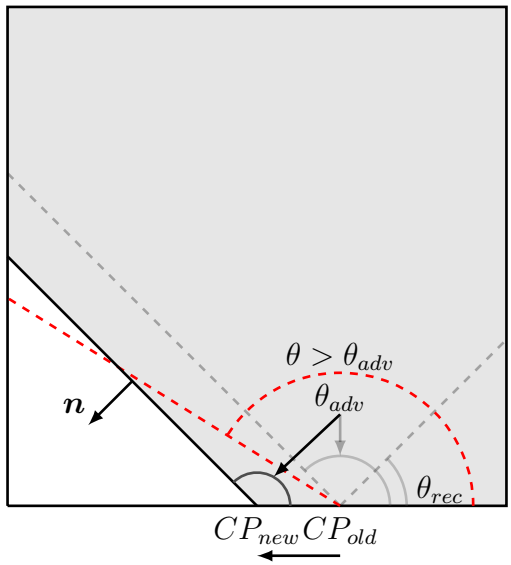


Figure 4.4: Contact point advection when α is greater than α_{max}.

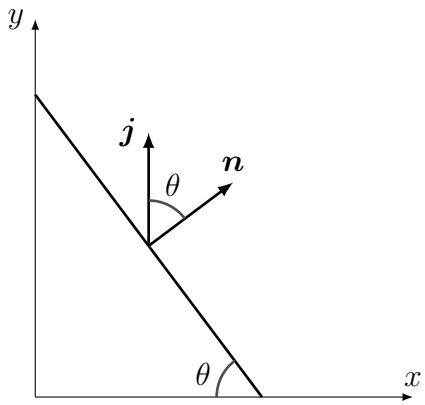


Figure 4.5: Direction angle between $\boldsymbol{j}$ and $\boldsymbol{n}$.

The location and advection of the contact point is determined by calculating the minimum distance between a point located on the PLIC and the center of the cell at $y = y_{jmin}$. If the xz plane is considered in Fig. 4.6, the PLIC along the bottom wall is then used to define the contact point as shown in Fig. 4.7. The vector $\boldsymbol{a}$ is fixed based on the normal given by ELVIRA and represents the orientation of the PLIC in the x and z directions and the vector $\boldsymbol{b}$ is a vector from the point on the PLIC on the leftmost cell boundary to the center of the cell O. Point B is selected based on the magnitude of the normals n_x and n_z and the location of the cell edge. If n_x is greater than n_z then the PLIC is extrapolated to the rightmost cell boundary where $x = x_{i+1}$, otherwise the PLIC is extrapolated to the uppermost boundary where $z = z_{k+1}$. The unknown points at A and B are determined by solving Eq. (4.9) for x and z given the general equation of the plane determined by ELVIRA. Points A and B represent the location where the PLIC intercepts the cell boundaries.

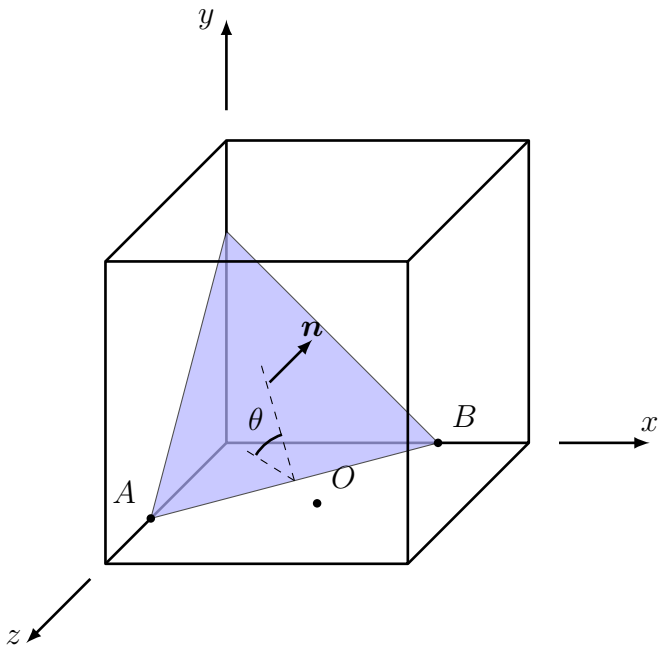


Figure 4.6: Determination of the contact point in three dimensions.

The contact point (CP) location is determined by calculating the minimum distance between a point on the PLIC and the cell center. This can be determined by projecting the vector $\boldsymbol{b}$ onto $\boldsymbol{a}$. The projection of $\boldsymbol{b}$ onto $\boldsymbol{a}$ can be expressed as:

$$\boldsymbol{c} = \left(\frac{\boldsymbol{b}\cdot\boldsymbol{a}}{|\boldsymbol{a}|^2}\right)\boldsymbol{a} = \left(\frac{\boldsymbol{b}\cdot\boldsymbol{a}}{\boldsymbol{a}}\right). \tag{4.10}$$

The vector $\boldsymbol{d}$ is then computed by:

$$\boldsymbol{d} = \boldsymbol{b} - \boldsymbol{c} \tag{4.11}$$

The location of the contact point can then be easily determined based on $\boldsymbol{d}$ and the center of the cell O.

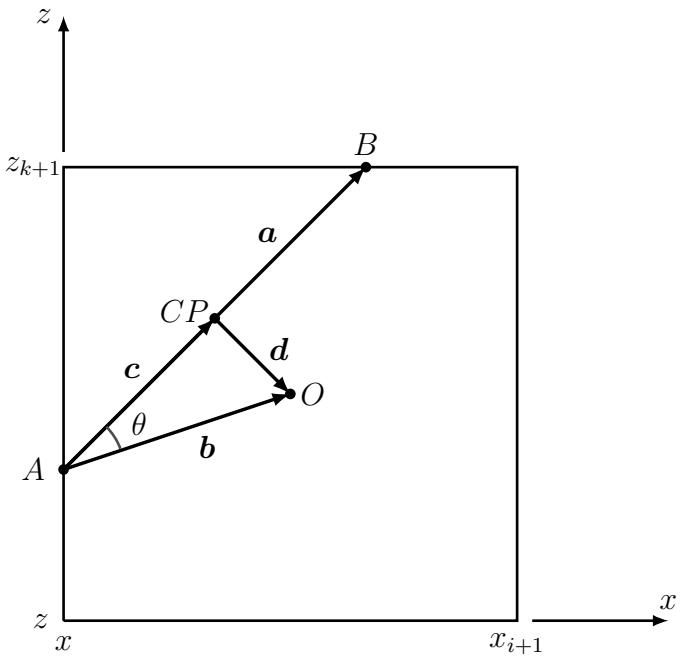


Figure 4.7: Determination of the contact point.

By specifying the contact point in this manner, the orientation of the PLIC along the wall can rotate freely about the contact point per each time step iteration

if α is between α_{adv} and α_{rec}. The advection of the contact point is only determined if α is greater than α_{adv} or less than α_{rec}. If α is less than α_{adv} and greater than α_{rec} then the n_y is determined using the bisection interpolation method to compute the general equation of the PLIC based on the volume fraction and the contact point. As the volume fraction increases or decreases, the contact point remains fixed and as a result the only variable that may change is n_y which will either increase or decrease depending on if the liquid volume fraction in the cell is either increasing or decreasing, respectively. As to our knowledge, an analytical solution does not exist that determines the general equation of the plane based on a fixed orientation of the PLIC in the n_x and n_z directions and the known liquid volume fraction.

Curvature

The dependence of an accurate representation of gas-liquid interfaces in a multiphase flow simulation is important to completely understanding the flow dynamics of a multiphase simulation. Since the surface tension force is proportional to the curvature, an accurate determination of the curvature is an essential representation of the flow behavior and characteristics. Surface forces affect: droplet and fluid velocities, droplet size distributions [41, 42], fluid deformations, and spray characterization [43] and understanding how surfaces forces are defined in a simulation and their limitations is critical to being able to predict and understand the characteristics of a flow.

The characteristics of droplets in a microscale flow are dominated by surface tension forces and the greater the surface tension force, the smaller the droplet sizes [44]. As the droplets are reduced in size, they become inherently difficult to accurately represent using current methods to compute the forces acting on the surface. As such, an accurate representation of the interface and the curvature is critical in

flows involving droplets. A challenge common to all VOF schemes is computing the interface curvature from the liquid volume fractions.

The height function method is a popular method used to compute the curvature in VOF schemes, [45, 46]. The height function method determines local height functions by extending the reconstructed interfaces along a boundary into the "ghost cells" and integrating the liquid volume either vertically or horizontally depending on the direction of the interface normal, Fig. 4.8. Most commonly, a 7x3x3 stencil is used to accurately compute the curvature at the cell center in a cell containing an interface by:

$$\kappa = \frac{h_{xx}}{(1 + h_x^2)^{3/2}} \tag{4.12}$$

where h_x and h_{xx} is the first and second derivative of the height function, respectively. It has been shown that the height function method, using a 7x3x3 stencil, is second-order accurate [47].

An issue that arose using the height function method was the extension of the interface into the ghost cells with defined contact angles. This created a challenging problem because the volume fractions needed to be defined in these cells and if several interfaces extended into a single ghost cell, an accurate determination of the liquid volume fraction became exceedingly difficult. The ghost cells are computational cells outside the simulation domain that are used to aid in defining boundary conditions. In order to use the height function method, the amount of liquid volume contained in the ghost cells must be determined. In two-dimensions this can be simply done by extending the interface into the ghost cells and determining the liquid volume fraction based on the interface normal. However, this becomes exceedingly difficult when multiple interfaces are merging into a single ghost cell as shown with merging

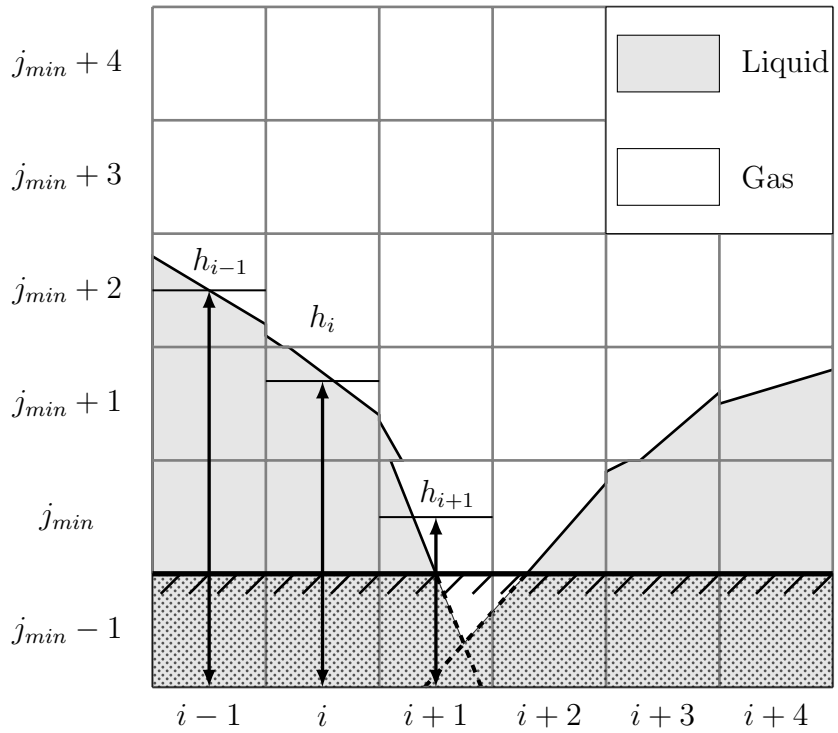


Figure 4.8: Issue with height function when multiple droplets are merging.

droplets in Fig. 4.8. In three-dimensions, a single ghost cell may experience multiple interfaces converging into a single ghost cell and determining the amount of liquid volume contained in this cell becomes non-trivial.

The next section focuses on developing a method to compute the curvature of an interface based on the reconstruction of the liquid interface using an implicit polynomial function fitted to a cloud of points obtained from VOF. Using the information gathered from the polynomial reconstruction of the interface, we are then able to easily compute the corresponding curvature of the interface. The purposed method aims to provide a simpler solution to compute the curvature in cells containing a contact angle while also providing a converging curvature even when only a second-order VOF representation is available.

Curvature Scheme

Utilizing the methods discussed above, the interface in each cell is represented by a straight line or a plane (PLIC). In order to compute the interface curvature of the liquid, a polynomial function is determined that best fits a set of points defined on the PLIC reconstruction. This function is then used to determine the corresponding curvature of the interface.

For each cell that contains an interface, a set of points are assigned to the PLIC one third the distance to the centroid from the points where the PLIC intersects the cells edges. Figure 4.9 shows an example of the points distributed on a reconstructed interface of PLIC. Similar definitions are used if the PLIC has more intersections with the cell. These points along with the points located in the adjacent neighboring cells in each direction are combined to create a cloud of points on the interface in the vicinity of the cell of interest.

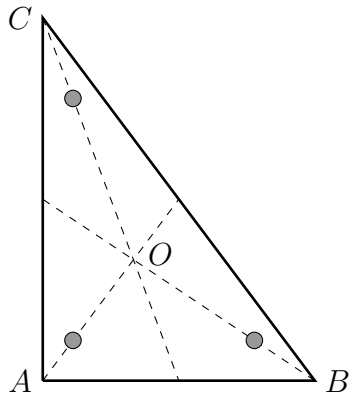


Figure 4.9: Triangulate point distribution

The cloud of points is fitted with an implicit polynomial, such as the second order function:

$$f(x,y,z) = a_{000} + a_{100}x + a_{010}y + a_{001}z + a_{200}x^2 + a_{110}xy + a_{020}y^2 + a_{101}xz + a_{011}yz + a_{002}z^2 \tag{4.13}$$

defined in three dimensions, which consists of 10 monomials shown in Eq. (4.14) and 10 coefficients, Eq. (4.15):

$$[\boldsymbol{m}] = [1 \; x \; y \; z \; x^2 \; xy \; y^2 \; xz \; yz \; z^2]^{\mathrm{T}} \tag{4.14}$$

$$[\boldsymbol{a}] = [a_{000} \; a_{100} \; a_{010} \; a_{001} \; a_{200} \; a_{110} \; a_{020} \; a_{101} \; a_{011} \; a_{002}]^{\mathrm{T}}. \tag{4.15}$$

The least-square method is a fitting technique used to estimate a function based on a series of data points $\{(x_1, y_1, z_1), \ldots, (x_N, y_N, z_N)\}$. The error associated with the fitted function can be linearly defined as:

$$E = \boldsymbol{m}^{\mathrm{T}}\boldsymbol{a} \tag{4.16}$$

The best fit function is found by minimizing the error of the estimated function:

$$E = \sum_{i=1}^{N}\sum_{j=1}^{N}(\boldsymbol{m}_{i,j}^{\mathrm{T}}\boldsymbol{a}_j)^2 \tag{4.17}$$

where N is the number of points used to construct the cloud of points, $\boldsymbol{m}$ is the vector of the monomials evaluated at each data point and $\boldsymbol{a}$ represents the number of unknown coefficients.

In order to fit the data points and determine an implicit polynomial curve for $f(x, y, z)$, the 3L algorithm was used [48]. The 3L algorithm was chosen because it is computationally efficient and does not require a high degree polynomial which may lead to oscillations for relatively simple surfaces such as a sphere while also providing accurate results to complex surfaces which may include singularities.

If the circle in Fig. 4.10 is considered and points are applied to the perimeter of this circle, these points can then be fitted using the 3L algorithm. The 3L algorithm

works by defining additional points Γ_{+c} and Γ_{-c}. These additional points are created by moving $+c$ and $-c$, respectively, in the normal direction from the original points Γ_o,Fig. 4.11. The enlarged cloud of points containing Γ_o, Γ_{+c}, and Γ_{-c} define three iso-surfaces of the 2D curv. The value of the iso-surfaces (values of f at Γ_o, Γ_{+c}, and Γ_{-c}) are set to 0, $+c$, and $-c$ respectively to enforce a unity gradient near the interface.

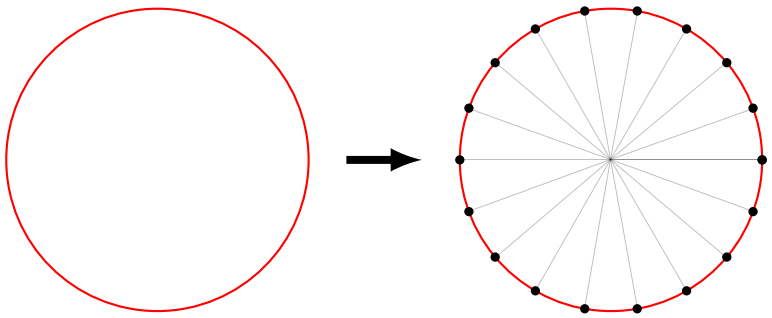


Figure 4.10: Example of applying points to the interface of a curve.

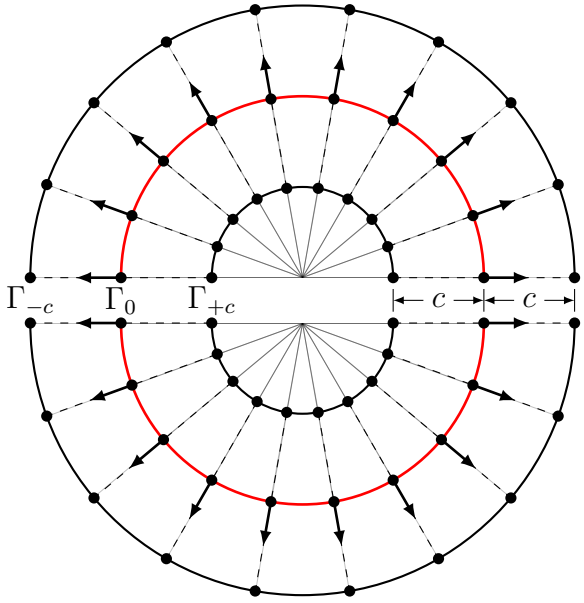


Figure 4.11: Example of the 3L points used in the x-y plane.

The evaluated monomials for each level set are used to generate the 3L block matrix, Eq. (4.18), and the block vector $\boldsymbol{b}$, Eq. (4.19):

$$[\boldsymbol{M_{3L}}] = \begin{bmatrix} \boldsymbol{M_{\Gamma_{-c}}} \\ \boldsymbol{M_{\Gamma_0}} \\ \boldsymbol{M_{\Gamma_{+c}}} \end{bmatrix} \tag{4.18}$$

$$[\boldsymbol{b}] = \begin{bmatrix} -\boldsymbol{c} \\ 0 \\ +\boldsymbol{c} \end{bmatrix} \tag{4.19}$$

where $\boldsymbol{M_{\Gamma_{-c}}}, \boldsymbol{M_{\Gamma_0}}, \boldsymbol{M_{\Gamma_{+c}}}$ are $(N \times \text{size}(\boldsymbol{m}))$ matrices. N is the total number of points used on each level set and $\boldsymbol{m}$ is the vector of monomials. The least-square solution for the coefficients of the polynomial are obtained by solving:

$$\boldsymbol{a} = (\boldsymbol{M_{3L}^T M_{3L}})^{-1} \boldsymbol{M_{3L}^T b} \tag{4.20}$$

The implicit polynomial is only fitted to points within the specified stencil. An example of a point distribution along a circle fitted with a 2^{nd} order polynomial is shown in Fig. 4.12. The grey surface represents the PLIC, the dots represent the distribution of points using a three neighbor stencil, and the red surface denotes the 2^{nd} order fitted polynomial function. Figure 4.12 shows good correlation between the fitted function and the interface of the circle in the region of the stencil. The curvature determined by the fitted polynomial is computed only in the cell containing an interface and not its neighbors. The curvature in the remaining neighboring cells are then determined individually if an interface exists following the same procedure but using different neighbors to complete the entire stencil.

A similar point distribution on a three-dimensional sphere is shown in Fig. 4.13. Even for a three-dimensional surface, the fitted polynomial to a set of points shows a relatively good fit and an accurate description of the curvature of the interface, Fig. 4.14 and Fig. 4.15.

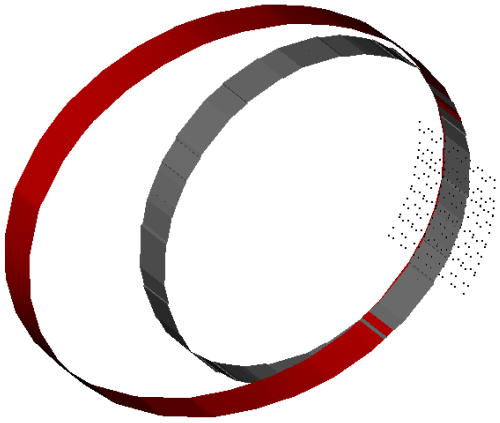


Figure 4.12: Three neighbor stencil of points assigned to the PLIC (grey) and fitted with 2^{nd} order polynomial function (red). The points represent the data points used to form the cloud of points on the interface.

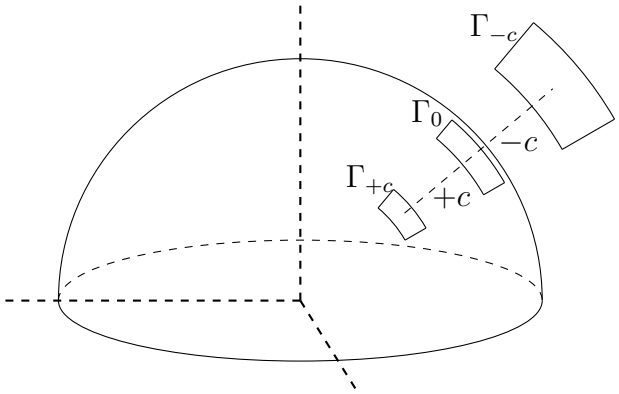


Figure 4.13: Ribbon used in 3L algorithm on a three-dimensional surface.

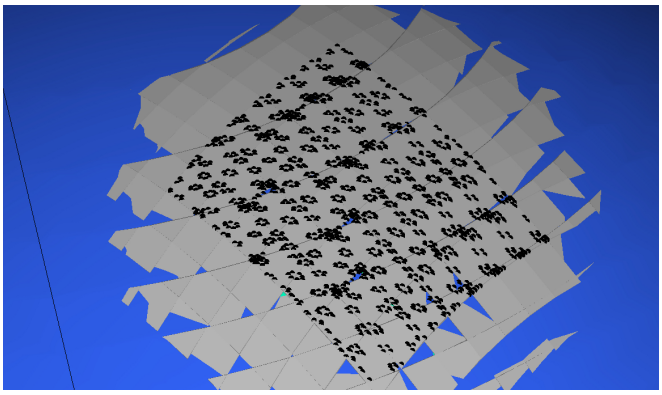


Figure 4.14: Two neighbor stencil of points assigned to a 3D surface. The gray surface represents the PLIC and the blue surface is the fitted 2^{nd} order polynomial function. The points represent the data points used to form the cloud of points on the interface.

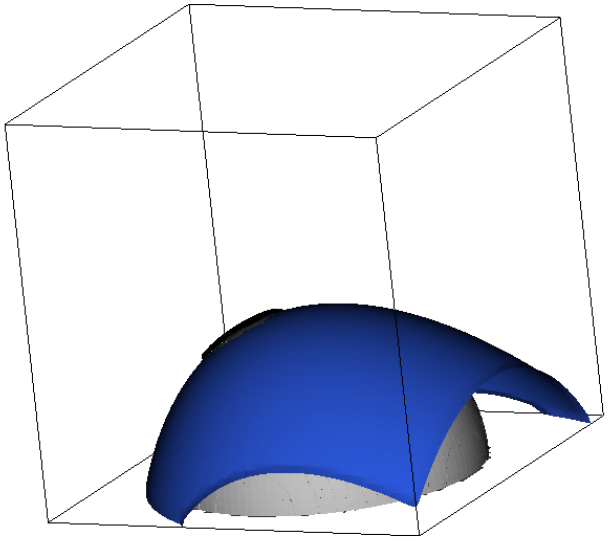


Figure 4.15: Points fitted with 2^{nd} order polynomial function

In order to determine the curvature of the liquid interface, the curvature is defined in terms of the computed polynomial function:

$$\kappa = \nabla \cdot \left(\frac{\nabla f(x,y,z)}{|\nabla f(x,y,z)|} \right) \tag{4.21}$$

The corresponding curvature is then calculated at the average of the points on the PLIC in the cell containing an interface. If the curvature has a large magnitude, the curvature of the function in that cell is large and if the curvature is zero, the curvature is straight. Essentially, the curvature depicts how much a surface deviates from a plane.

The 3L method can also be extended to other orders of polynomials as well. 2^{nd}, 3^{rd}, and 4^{th} order polynomials were considered in the initial tests. Several other variables were also considered during the initial tests which include: number of points per PLIC, the size of the stencil, and the distance to the inner and outer level sets (c).

CURVATURE RESULTS

The new curvature method was tested extensively in two and three dimensions in order to verify that the method worked and to determine the accuracy of this new method. Several variations of variables and different routines were tested when developing the new curvature method. Some of these had little effect on the solution or had a negative impact on the solution. These alterations and the corresponding results are discussed below in the miscellaneous section for completeness to the overall development of the new curvature method.

The following simulations were performed using the NGA computational platform [49].

Circle Test Case

Several tests were conducted using a circle defined in two dimensions in order to verify the accuracy of the new curvature scheme and to gain a better understanding of how the size of the stencil and the order of the fitted polynomial affect the computed curvature. These results are compared to the results obtained using the mesh decoupled height function method [50] using the exact analytical solution for the curvature a circle.

For the exact test case, a circle was centered in a square unit domain and had a radius of 0.2. For the second order VOF approximation test, the center of the circle was randomly shifted in order to prevent perfectly alignment of the VOF with the computational cells and to replicate more realistic VOF conditions. For a circle with radius R, the exact curvature is computed by:

$$\kappa = \frac{1}{R}\,. \tag{5.1}$$

Two tests were conducted. The first was performed by assigning points along the entire interface using the exact radius of a circle, Fig. 5.1. These points were used to compute the corresponding curvature to minimize any error that may occur by computing the curvature located on a point not directly on the circle. Finally, exact normals were used to reduce any sources of error outside of fitting the implicit polynomial and computing the corresponding curvature. Similarly, a perfect test case for the height function method was created by specifying exact liquid volume fractions to define the circle.

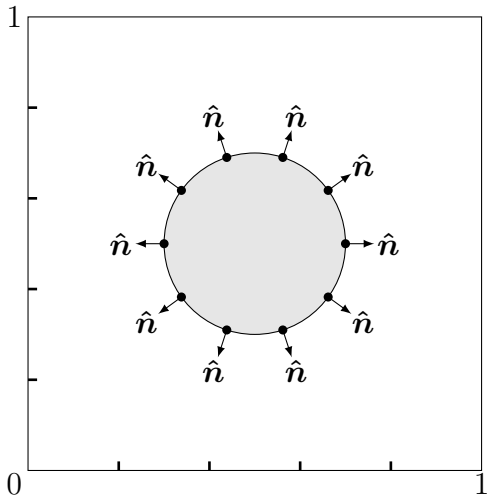


Figure 5.1: Exact circle test case.

The next 2D test case was performed using a second-order VOF approximation of the interface. This is a common situation because most VOF transport schemes are second-order accurate. The volume fractions were computed using the trapezoidal rule to approximate the volume integrals. Points were then assigned to the PLIC and ELVIRA was used to specify the direction normal to the interface. The location of the center of the circle was shifted randomly by a small amount and the results over 50 tests were then averaged. A value of $c = \Delta x$, the cell size, is used.

Test Evaluation

In order to test mesh convergence, several tests were conducted in a uniform mesh. The number of computational cells across the domain was increased in increments of 5 beginning at $nx = ny = nz = 10$ until $nx = ny = nz = 100$ and then the increments were increased by 10 until $nx = ny = nz = 180$. Two error metrics were performed after each test, specifically the L_2 and L_∞ errors defined in terms of the curvature κ:

$$L_2(\kappa) = \frac{\sqrt{\sum_{i=1}^{N_{cells}} (\kappa_i - \kappa_E)^2}}{\sqrt{\sum_{i=1}^{N_{cells}} \kappa_E^2}} \tag{5.2}$$

$$L_\infty(\kappa) = \max_{i=1\ldots N_{cells}} \left| \frac{\kappa_i - \kappa_E}{\kappa_E} \right| \tag{5.3}$$

where κ_i and κ_E represent the computed and exact curvatures, respectively. L_2 computes the average error in curvature across surface while L_∞ determines the maximum curvature error on the surface.

Circle Test Case Results

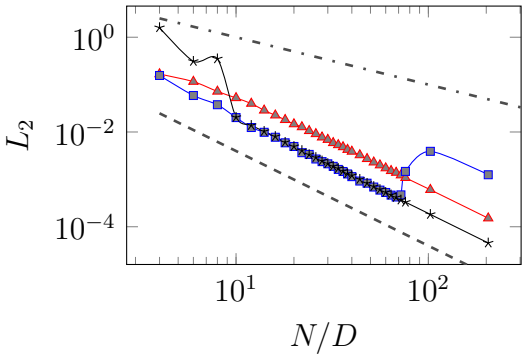
The results shown in Fig. 5.2 show the errors calculated for the exact circle test case when using a 1-3 neighbor stencil. The errors are plotted as a function of the number of cells (N) across the diameter of the droplet (D). The results show that both the 2^{nd} and 3^{rd} order polynomials are second-order accurate when using 1-2 neighbors. The 2^{nd} order polynomial fails to converge with second-order accuracy when using 3 neighbors. The 3^{rd} order polynomial becomes numerically unstable as the mesh is refined when using 1-2 neighbors. This is partly due to the characteristics of the 3^{rd} order polynomial and its inability to accurately fit a tight grouping of points

due to the number of inflection points. When the number of neighbors increased, the span of the distribution of points increased which improved the accuracy of the results.

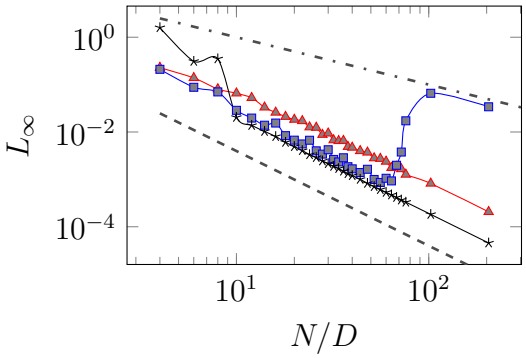
The results of the height function method showed a second-order convergence using an exact VOF which was expected [47].

The 2^{nd} order polynomial produced the best results overall, Fig. 5.3, and produced lower errors overall than compared to the height function method. Both the 2^{nd} and 3^{rd} order polynomials are seen to be converging for course meshes but both results display noise as the mesh is refined. The noise is reduced as the number of neighbors is increased. However, even with the extra noise, the results using the 2^{nd} order polynomial and 3 neighbors are more accurate than the mesh decoupled height function method which fails to converge for relatively course meshes, Fiq. 5.3e.

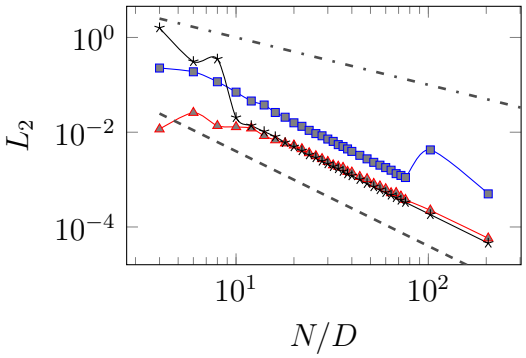
In summary, for a two-dimensional circle, the purposed scheme provides a converging curvature on meshes as fine as $N/D \approx 60$. The height function only converges up to $N/D \approx 15$. This extra range of convergence will improve the accuracy of many engineering simulations that resolve liquid structures with between 15 and 60 points.



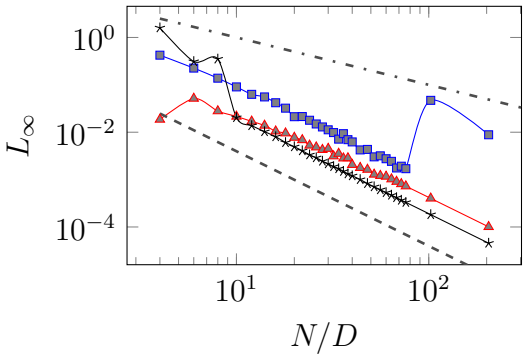
(a) 1 neighbor L_2 error



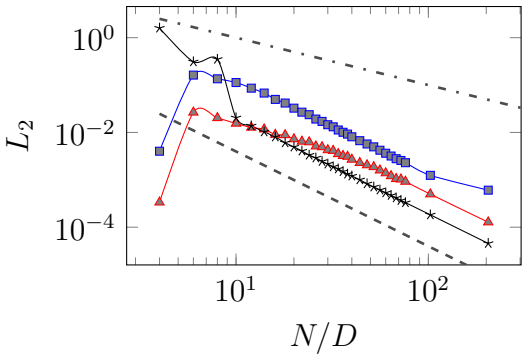
(b) 1 neighbor L_∞ error



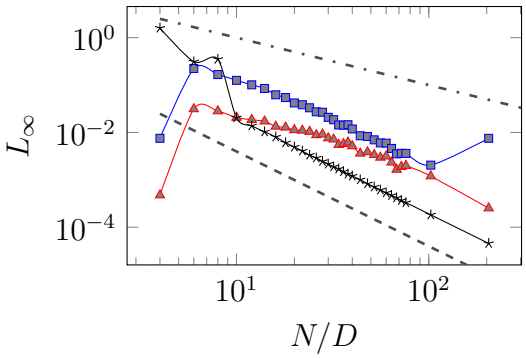
(c) 2 neighbors L_2 error



(d) 2 neighbors L_∞ error

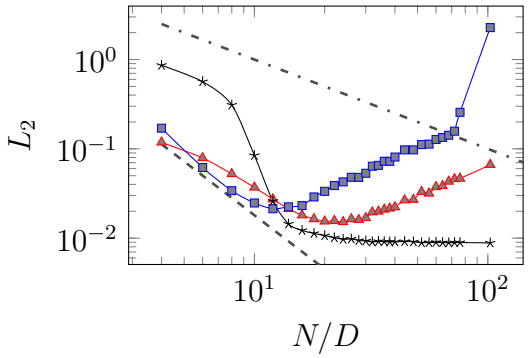


(e) 3 neighbors L_2 error

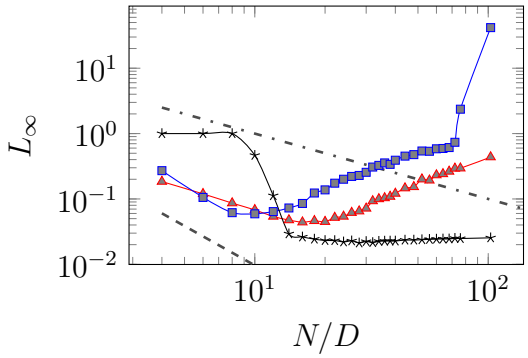


(f) 3 neighbors L_∞ error

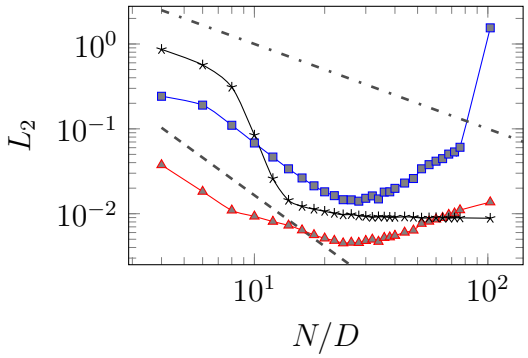
Figure 5.2: Exact circle curvature test case. The triangles and squares indicate a 2^{nd} and 3^{rd} order fitted polynomial functions, respectively. The stars represent the height function method and the dash-dotted and dashed-dashed lines show first and second order convergence for comparison.



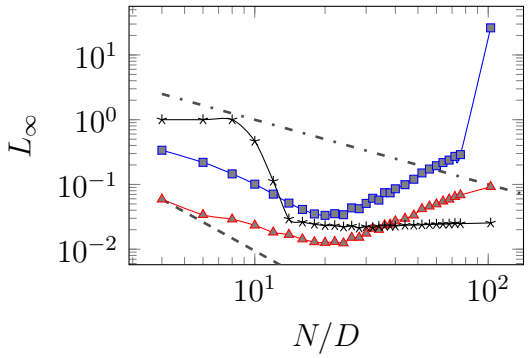
(a) 1 neighbor1 L_2 error



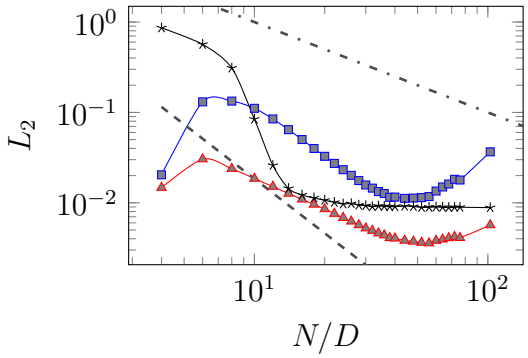
(b) 1 neighbor L_∞ error



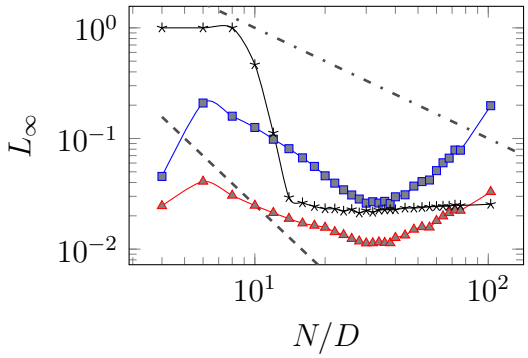
(c) 2 neighbors L_2 error



(d) 2 neighbors L_∞ error



(e) 3 neighbors L_2 error



(f) 3 neighbors L_∞ error

Figure 5.3: Second-order VOF approximation circle test case. Results were averaged over 50 simulations per each mesh level. The red triangles and blue squares indicate a 2^{nd} and 3^{rd} order fitted polynomial functions, respectively. The dash-dotted and dashed-dashed lines show first and second order convergence for comparison.

Sphere Test Case

A three-dimensional test was performed in order to compare the results from the new curvature method to the results obtained from the mesh decoupled height function method. A sphere with a radius of 0.4 was randomly shifted from the center of the domain, in a cubic domain with unit edge lengths, Fig. 5.4. For a sphere with a radius R, the exact curvature is computed by:

$$\kappa = \frac{2}{R}\,. \tag{5.4}$$

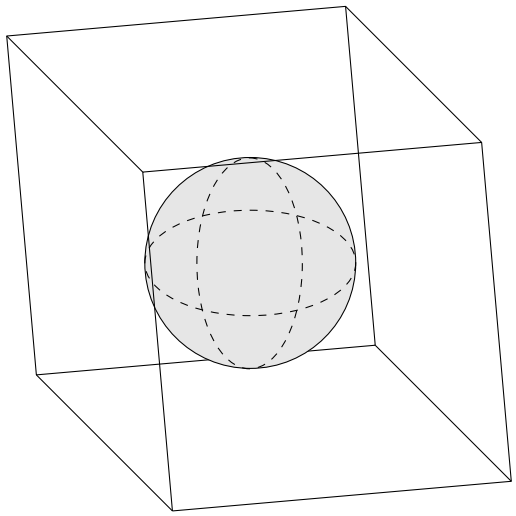


Figure 5.4: Sphere test case.

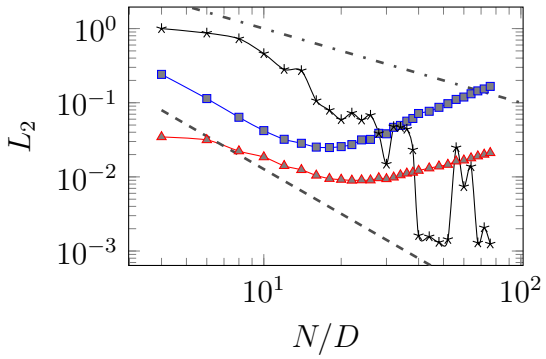
Sphere Test Case Results

The results for the sphere test case are shown in Fig. 5.5. Similar to the 2D circle case, a 2^{nd} order polynomial produced the best results overall. The results for all three neighbors show that both the 2^{nd} and 3^{rd} order polynomials are converging with at least first-order accuracy. Using 3 neighbors produced the best results overall,

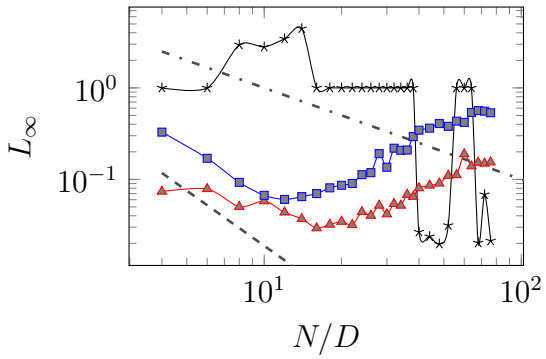
Fig. 5.5e. Both the 2^{nd} and 3^{rd} order polynomials are converging with at least first-order accuracy. The 3^{rd} order polynomial begins to stop converging around $N/D = 64$ while the 2^{nd} order polynomial continues to converge with first-order accuracy out to $N/D = 75$ which is the smallest mesh size tested in three-dimensions.

The results using the height function method showed a large fluctuation in the results. Some of this fluctuation is expected since the VOF is second-order accurate. However, the heigh function did produce peak L_2 errors which were significantly less than the errors produced using the new curvature method and the height function method would benefit greatly from taking the average of 50 results similar to what was done in the 2D test case. One interesting thing to note is how large the L_∞ errors are using the height function method. This error depicts the largest error in curvature that occurred across the entire surface of the sphere. Overall, the height function method produced L_∞ errors that were much greater than the L_∞ errors seen using the new curvature method. The significance of this is that the these inaccuracies would induce rather large pressure gradients in these cells which would impact the interface not only in this cell but in the neighboring cells as well and this error would create a ripple affect across the entire interface yielding unrealistic results.

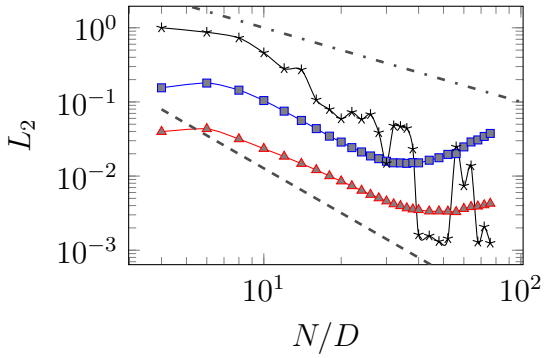
Figure 5.6 shows a comparison of the curvature results using the new curvature method and the height function method with an N/D of 10. The value of the curvature in each cell containing an interface is plotted across the interface of the sphere. This result clearly shows the consistency of computing the curvature across a sphere using the new curvature method. There is very little deviation between curvature values using the new curvature method. However, there is a large deviation of curvature values using the height function method. These deviations will produce unrealistic results of the interface and will lead to errors in a simulation which are difficult to detect.



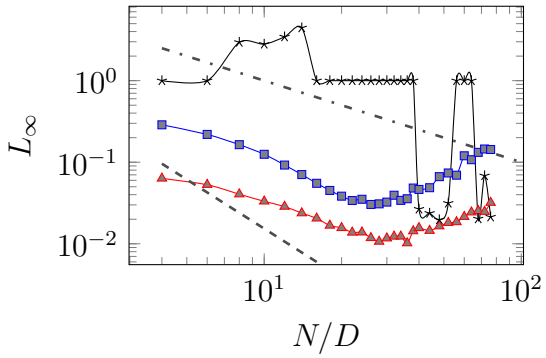
(a) 1 neighbor1 L_2 error



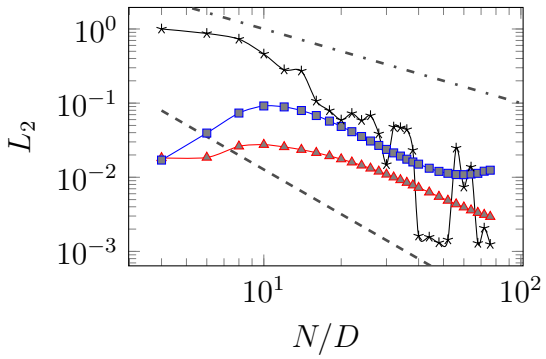
(b) 1 neighbor L_∞ error



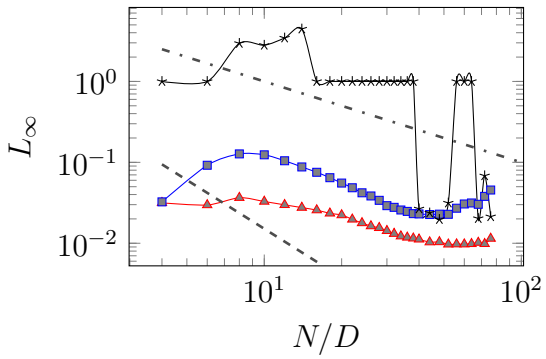
(c) 2 neighbors L_2 error



(d) 2 neighbors L_∞ error

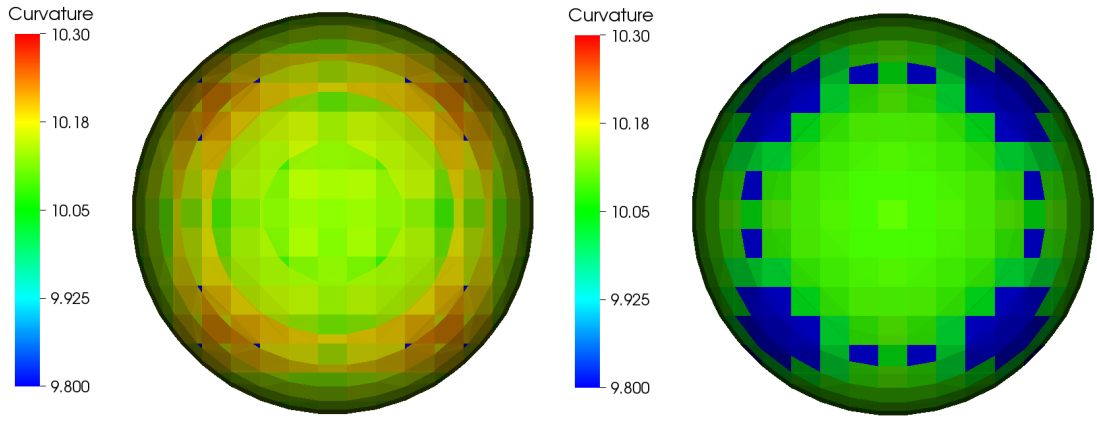


(e) 3 neighbors L_2 error



(f) 3 neighbors L_∞ error

Figure 5.5: Second-order VOF approximation for the 3D sphere test case. The red triangles and blue squares indicate a 2^{nd} and 3^{rd} order fitted polynomial functions, respectively. The dash-dotted and dashed-dashed lines show first and second order convergence for comparison.



(a) New Curvature method (b) Height function method

Figure 5.6: Comparison of the new curvature method results using a 2^{nd} order polynomial with 2 neighbors and the height function method results across a sphere with N/D=14. Exact curvature equals for this sphere equals 10.

Miscellaneous Testing

Constrained Least Square Method

The unconstrained least-square method is an approximation to fit a function to a set of data points and is not point dependent but finds the best fit function that minimizes the error between the function and the cloud of points being fitted. Errors in the computation of the curvature occurred if the PLIC in adjacent cells do not lie on roughly the same plane. The reason for this error is that the polynomial function may not pass through any of the points in a cell that the curvature is to be determined because the solution yields the best fit for the cloud of points. Therefore, the error may be substantial if the function does not represent the orientation of points on the PLIC in the cell the curvature is to be determined. In order to ensure that the polynomial function passes through a point in a cell where the curvature is to be determined, a constrained least square method was used [51]. The constrained least-

squares method is similar to the unconstrained method above except the minimization function, Eq. (4.17), is subject to an equality constraint:

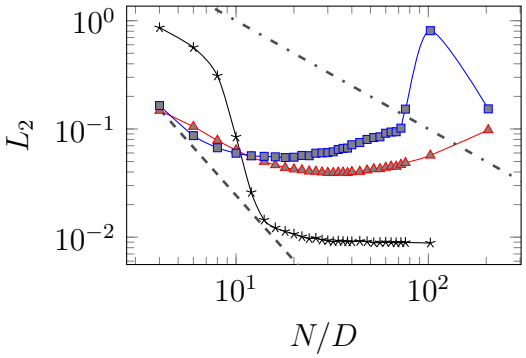
$$\boldsymbol{C}\boldsymbol{a} = d \tag{5.5}$$

where $\boldsymbol{C}$ is a $(1 \times \text{size}(\boldsymbol{m}))$ vector consisting of the monomials evaluated at a point in which the function is to constrained to. The point chosen to constrain the function was taken to be the average of the points located on the PLIC in a cell where the curvature was to be determined. The value of d is zero and $\boldsymbol{a}$ is then determined in order to satisfy the constraint applied to Eq. (4.17). The coefficients of the polynomial are then determined by solving:

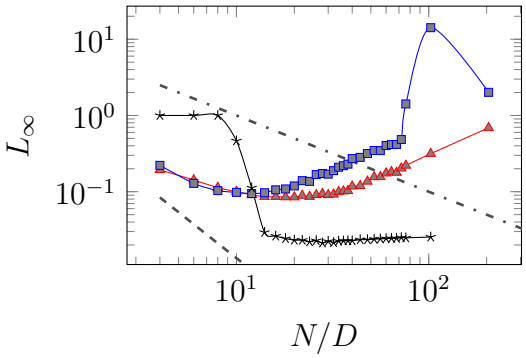
$$\begin{bmatrix} \boldsymbol{a} \\ z \end{bmatrix} = \begin{bmatrix} 2\boldsymbol{M}_{3L}^{\mathrm{T}}\boldsymbol{M}_{3L} & \boldsymbol{C}^{\mathrm{T}} \\ \boldsymbol{C} & 0 \end{bmatrix}^{-1} \begin{bmatrix} 2\boldsymbol{M}_{3L}^{\mathrm{T}}\boldsymbol{b} \\ 0 \end{bmatrix} \tag{5.6}$$

for $\boldsymbol{a}$.

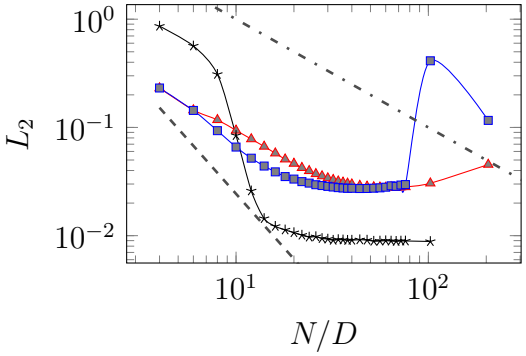
The main issue with the constrained least-square method is that the equality constraint ensures that the function passes through a point in the cell where the curvature is to be determined. This may not be the best fit for the entire set of data points in the stencil and the polynomial function may not pass through any points in the neighboring cells, however, the unconstrained method may pass through points located in the neighboring cells but not the points contained in a cell in which the curvature is to be determined. The constrained least-square method failed to converge in two-dimensions due to an induced error by being constrained to a single point and was therefore temporarily removed from the curvature routine. The constrained least-square method may be better suited to this curvature method if multiple points could be constrained instead of a single point.



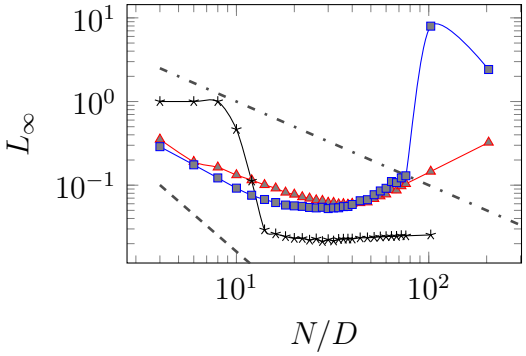
(a) 1 neighbor1 L_2 error



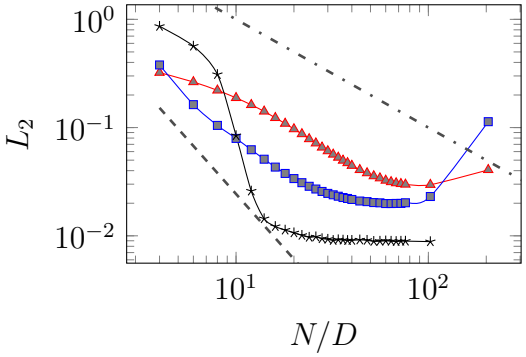
(b) 1 neighbor L_∞ error



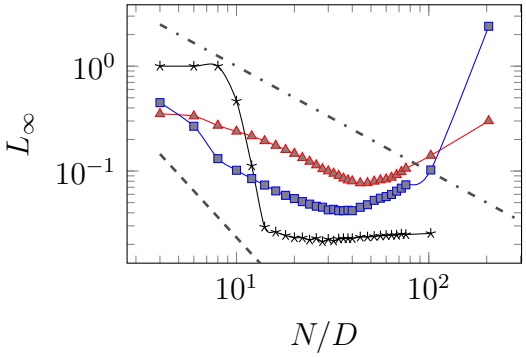
(c) 2 neighbors L_2 error



(d) 2 neighbors L_∞ error



(e) 3 neighbors L_2 error

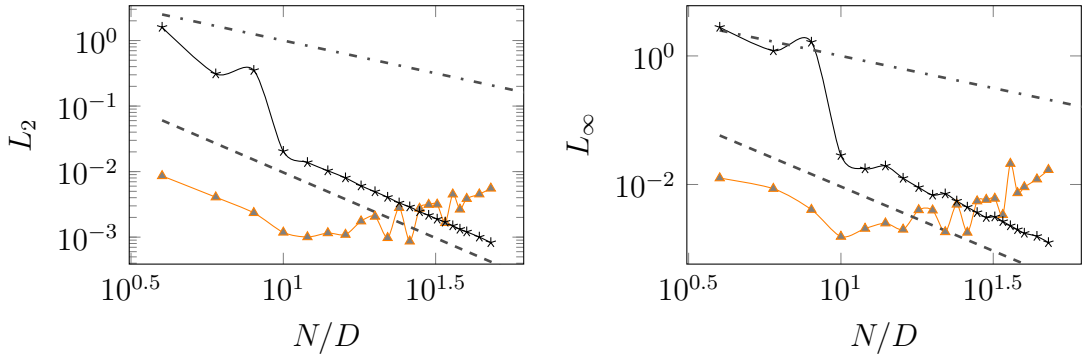


(f) 3 neighbors L_∞ error

Figure 5.7: Second-order VOF approximation for the 2D circle test case using a constrained least squares fit. Results were averaged over 50 simulations per each mesh level. The triangles and squares indicate a 2^{nd} and 3^{rd} order fitted polynomial functions, respectively. The dash-dotted and dashed-dashed lines show first and second order convergence for comparison.

Higher Order Polynomials

The highest order polynomial used in the new curvature method was a 4^{th} polynomial. For relatively simple surfaces such as the circle and the sphere, the 4^{th} polynomial became numerically unstable as the mesh size was reduced. This error was likely a result of trying to fit a tight grouping of points while having three inflection points in a short distance. Higher order polynomials would work much better if the points separated by greater distances and the surface that was being fitted was much more complicated. This conclusion can easily be seen in Fig. 5.8 for the exact VOF circle case. For course meshes, the overall accuracy of the 4^{th} polynomial is better than the results shown for a 2^{nd} or 3^{rd} order polynomial but as the mesh is refined further, the error increases drastically and becomes unstable.



(a) 3 neighbors L_2 error

(b) 3 neighbors L_∞ error

Figure 5.8: 4^{th} order polynomial curvature convergence results using an exact VOF. The triangles represent the 4^{th} order polynomial and the stars are the results obtained using the height function method.

Number of Points per PLIC

The number and location of the points located on the PLIC was altered in order to determine whether an even distribution of points gave the best results overall and the effect that the distribution of point had on the curvature results. An even

distribution of points is relatively difficult to achieve though because the number of points per PLIC is directly related to the area of the PLIC and therefore the distance between points is not constant. A test was conducted in order to see how the curvature was impacted by the number of points using a 1 neighbor stencil. The results for a quarter of a circle, using an $N/D = 40$, is shown in Fig. 5.9. There was not a noticeable trend in the results that would suggest that the fluctuation in the liquid volume fractions in the neighboring cells had a significant impact on the curvature results. A similar test was performed by varying the number of points per PLIC. Figure 5.10 shows that the number of points per PLIC also had very little affect on the solution and therefore a minimum number of points per PLIC was chosen.

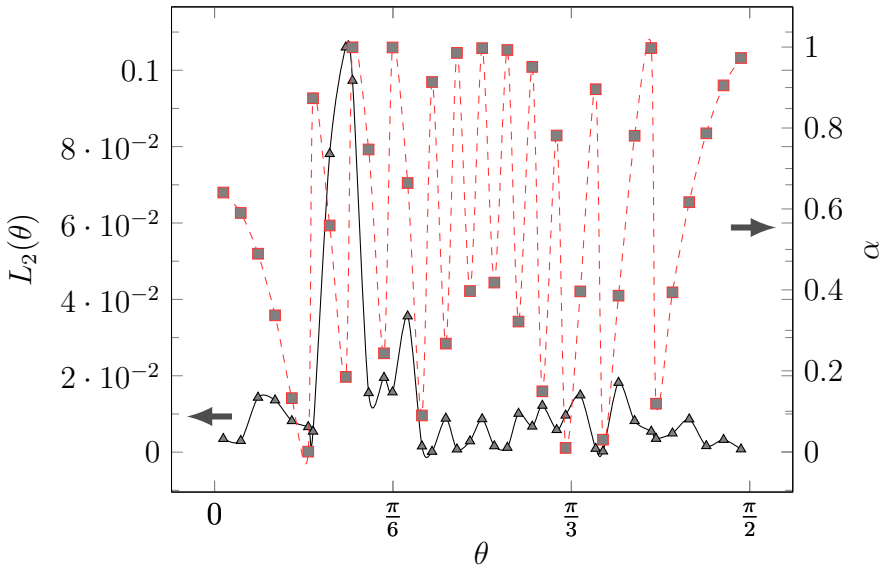


Figure 5.9: One neighbor stencil L_2 curvature error (line/triangles) and the volume fraction α (dashed/squares) plotted as a function of the angular position (θ) over a quarter of a circle.

Euclidean Distance

The parameter that had the largest impact on the overall curvature results was the Euclidean distance (c) or the distance normal to the interface on the zero level

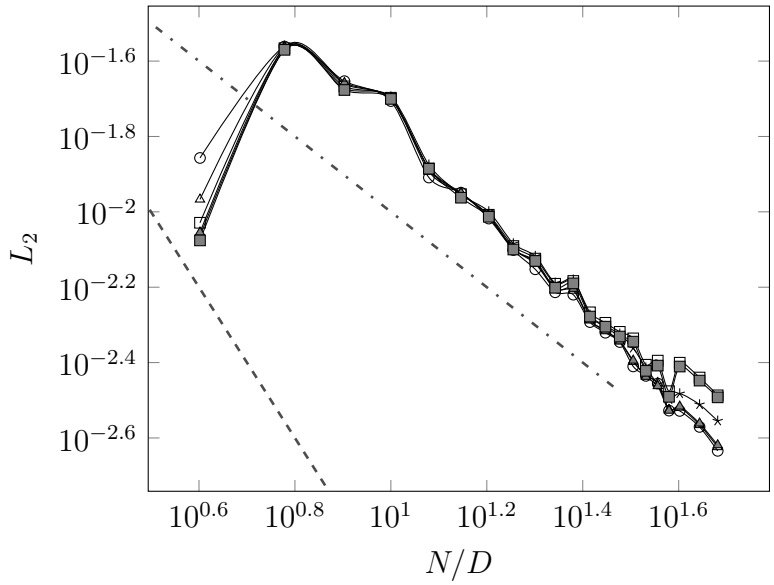


Figure 5.10: Points per PLIC comparison using an exact VOF and 3 neighbors.

set to the inner and outer ribbons. A series of tests was conducted by increasing and decreasing the distance c in order to determine how much of an influence this parameter had on the overall curvature solution based on the mesh size of the domain. The convergence solutions shown in Fig. 5.2, 5.3, 5.5 were based on a value of c equal to that of the corresponding mesh size where the mesh size is equal to length of the domain divided by the number of cells across the length of the domain, L_x/n_x .

From Fig. 5.11 it is clear that as the Euclidean distance distance c is increased, the distance separating the points on the inner level set Γ_{+c} is reduced inducing a singularity and it fails to approximate the actual surface given by Γ_0. The authors of the 3L algorithm recommend specifying c in the range of 2-4% of the total object's size that is being fitted [48].

Several tests were conducted based on different mesh sizes in order to visualize how the Euclidean distance affected the curvature results using a 2D circle. The Euclidean distance c was varied by varying the constant a times the corresponding

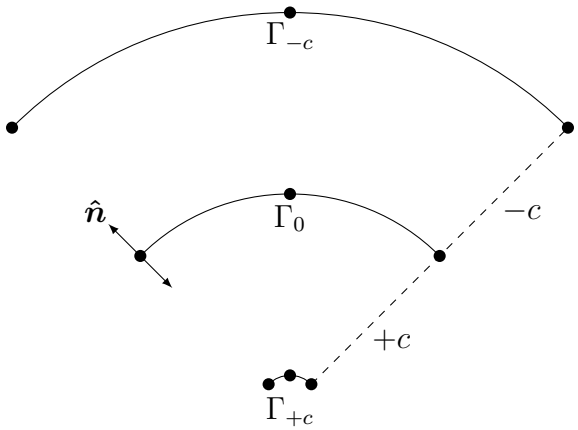


Figure 5.11: Image showing the affect that the Euclidean distance c has on the inner and outer level sets.

mesh size. The test cases were performed using a 2^{nd} order fitted polynomial and 3 neighbors for N/D equal to 16, 68, 76. When the value of c equals 1, the computed curvature is calculated using the same Euclidean distance as that used to compute the curvature results shown in Fig. 5.3e.

The results shown in Fig. 5.12 shows how the curvature L_2 error is affected when the Euclidean distance is varied for $N/D = 16$, $N/D = 68$, and $N/D = 76$. $N/D = 16$ is a relatively course mesh and as a result the distance separating the distribution of points is much greater than that using a finer mesh. As the mesh is refined the overall affect of decreasing c is reduced. One of the conclusions drawn from the results of the convergence plots in Fig. 5.2, 5.3, 5.5 was that as the mesh was refined, the distance separating the points decreased and as a result the curvature error increased. Therefore, this test was performed at these reduced mesh sizes in order to determine if increasing or decreasing Euclidean distance reduced the curvature error.

For a relatively simple surface, such as a circle, this result really only shows how much of an impact the Euclidean distance has on the result. For a much more

comprehensive study, this test should be performed using a surface which features steep troughs or peaks.

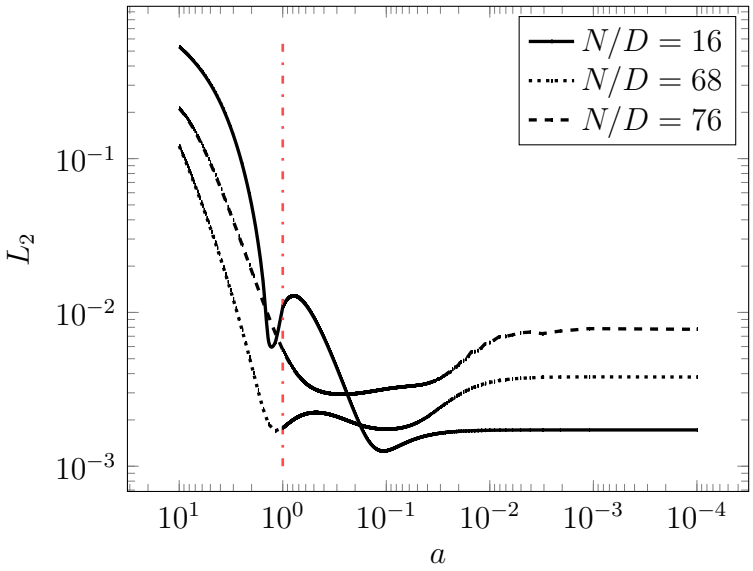


Figure 5.12: Curvature results while varying the Euclidean distance $c = a(L_x/n_x)$ for three different mesh sizes. The red dashed vertical line represents c equal to the mesh size.

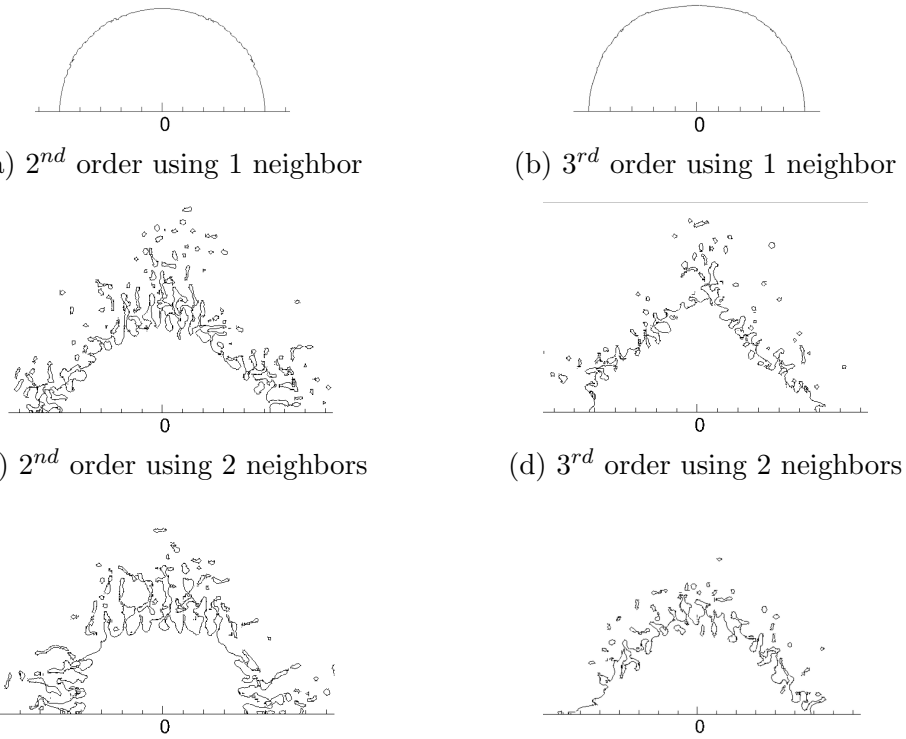
CONTACT ANGLE RESULTS

After the new curvature code was tested and verified that it was working correctly, the new curvature method was applied to a droplet on a surface in a channel with a incoming gas flow. The spreading and the displacement of a droplet was tested in two and three dimensions in a gas flow channel. Both the advancing and receding contact angles were varied in order to test how the droplet responds to different surface wettability characteristics and to verify that the droplet was behaving in a way that was expected.

One of the challenges in testing the new contact angle code was the lack of adjustable parameters in the numerical model in order to fine tune the results. Currently, the only user defined parameter that can be adjusted is the advancing and receding contact angles. The definition of the slip length is dependent on the volume fractions and the dynamic contact angles. Another parameter that is mesh dependent is the scale at which the contact angle is defined. Currently, the contact angle is defined in the bottom row of subcells in a cell that contains an interface. As the mesh is refined, the cells containing the contact line are refined therefore, the percentage of the interface that contains a contact angle is reduced and the contact angles has less of an impact on the motion and displacement of the droplet.

A 2D test case was performed using a variation of curvature parameters, similar to those used to perform the curvature convergent plots shown in Fig. 5.2, 5.3, 5.5, for a droplet resting on a wall. This test was performed using the unconstrained least square method. Figure 6.1 shows the results of the droplets after the simulation has completed 100 cycles. From these results it is concluded that the orientation of the fitted polynomial, meaning that the polynomial may be convex or concave, is highly dependent on the location of the points in the adjacent cells. Based on this idea,

the location where the curvature is computed may not lie directly on an isosurface of the polynomial function which may then induce large errors in the curvature of the interface.



(a) 2^{nd} order using 1 neighbor

(b) 3^{rd} order using 1 neighbor

(c) 2^{nd} order using 2 neighbors

(d) 3^{rd} order using 2 neighbors

(e) 2^{nd} order using 3 neighbors

(f) 3^{rd} order using 3 neighbors

Figure 6.1: Induced errors at the interface due to the curvature using an unconstrained least square methods after 100 cycles.

The explanation for these errors can be reasoned by examining Fig. 6.2. Because the least square method minimizes the error of the fitted function to a set of points in which it is fitting, the solution may not pass through a set of points in a cell containing an interface. If this occurs, the error of the curvature may not accurately represent the interface of the surface which induces unrealistic pressure gradients near the interface. If curvature was to be determined in the i, j cell shown in Fig. 6.2 based on a 2 neighbor stencil, the best fit may be similar to the curve shown. Because the

curve is not constrained to the points located in the i, j cell, this curve may not pass through any of the points located in this cell as shown and the curvature is then computed using points that do not lie directly on or near the isosurface. As a result, the computed curvature in that cell is positive, therefore a negative pressure will occur as a result. This induced negative pressure will force the liquid volume downwards leading to large fluctuations at the interface. If the interface in the i, j cell continues to experience positive curvatures, the liquid will continue to move downward and the error will increase per each iteration.

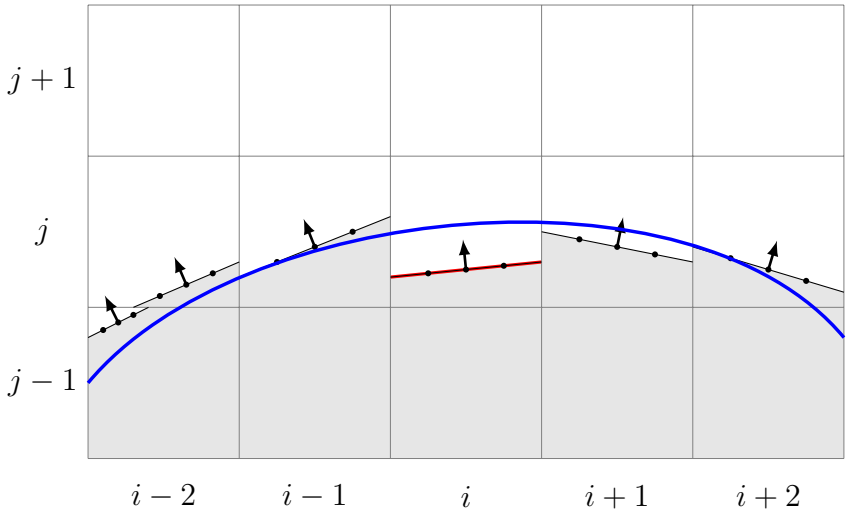


Figure 6.2: Pictorial depicting the possible errors occurring using a unconstrained least squares method fitted with a 2^{nd} order polynomial.

When initially testing the curvature method, tests using a constrained least square method were performed. The results of these tests revealed unpredictable curvature errors for a simple circle. As a result, the constrained least squares method was dropped from the initial testing. However, with the recent results using the unconstrained least squares method shown in Fig. 6.1, the constrained method was implemented with the idea that by forcing the isosurface through the cell in which the curvature was to be computed, the orientation of the polynomial would more

accurately represent the interface in this cell based on the location of its points along with the neighboring points. Using the same PLIC orientation example that was used for the unconstrained method shown in Fig. 6.2, a fitted polynomial using a constrained least squares method might resemble something similar to that shown in Fig. **??** where the isosurface is forced to intersect a point on the PLIC in the i, j cell. This orientation better represents the shape of the interface and the curvature in this cell is negative which will create a positive pressure gradient near the interface which will force the fluid upwards in the i, j cell. The issue now is that some information in the neighboring cells is lost using a 2^{nd} order polynomial and 2 neighbors. However, this error may be mitigated by using a 3^{rd} order polynomial instead. The simulation results, at the completion of 100 cycles, using a constrained least squares method is shown in Fig. 6.4. The errors at the interface have decreased significantly and the added constraint has improved the overall results.

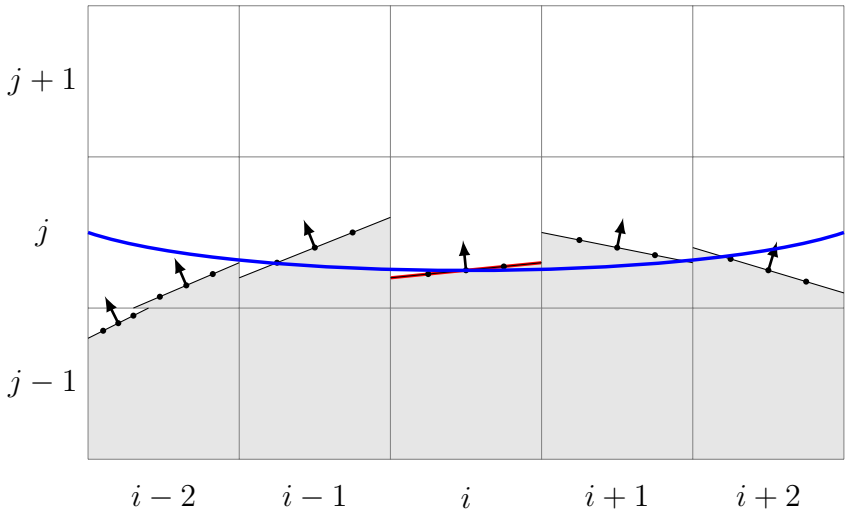
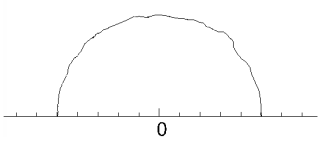
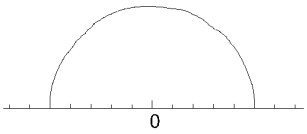


Figure 6.3: A possible solution using a constrained least squares method fitted with a 2^{nd} order polynomial.

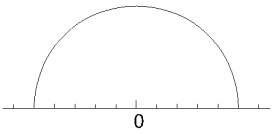
A simple 2D simulation was run to test the displacement of the contact point in two dimensions with a cross jet flow, Fig. 6.5. The red interface represents the PLIC



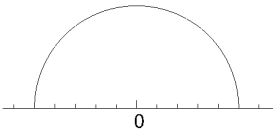
(a) 2^{nd} order using 1 neighbor



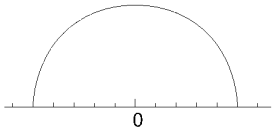
(b) 3^{rd} order using 1 neighbor



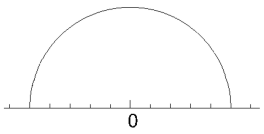
(c) 2^{nd} order using 2 neighbors



(d) 3^{rd} order using 2 neighbors



(e) 2^{nd} order using 3 neighbors



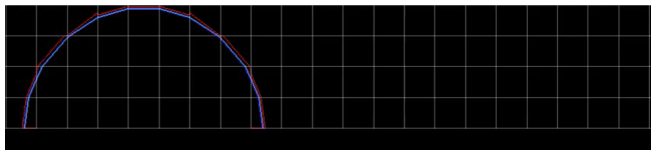
(f) 3^{rd} order using 3 neighbors

Figure 6.4: Simulation results after 100 cycles using a constrained least square method.

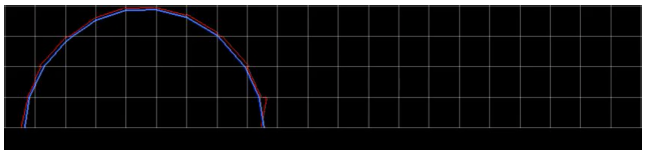
and the blue interface represents the average liquid volume fraction of the droplet. This simulation was run using the height function method to compute the curvatures. From Fig. 6.5, it is clear from the red interface that the PLIC maintains an advancing contact angle of 155° while displacing in the direction of the flow while the receding angle of the PLIC is maintains a constant 30°.

A 3D test case was performed using a single droplet in a channel with a gas crossflow as shown in Fig. 6.6. A 0.25mm droplet was defined in a $(1 \times 0.25 \times 0.6)$mm channel with a mesh size of $(320 \times 80 \times 192)$ which corresponds to the $(x \times y \times z)$ dimensions, respectively. The advancing and receding contact angles were 135° and

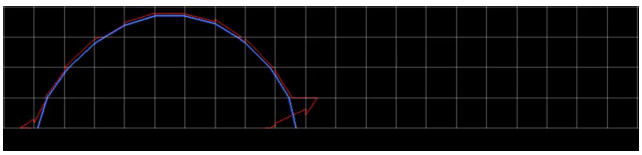
70°, respectively. A constrained 3^{rd} order polynomial and 1 neighbor was used to compute the curvature. As the simulation time progresses, the droplet begins deform in the direction of the flow due to the force acting at the interface due to the gas flow as time progresses and will eventually reach steady state as shown in Fig. 6.6d where the droplet will remain stationary. The droplet does not displace which could be due to the either the size of the droplet or the channel dimensions. What is interesting though, is that the defined contact angles are preventing the droplet from advecting in the direction of the flow because the computed contact angle of the interface is within the hysteresis angle. This result is promising in studying droplet formations in PEM fuel cells because if the droplet remains stationary due to the hydrophobicity of the solid, then channel blockage may occur as droplets merge within the channel.



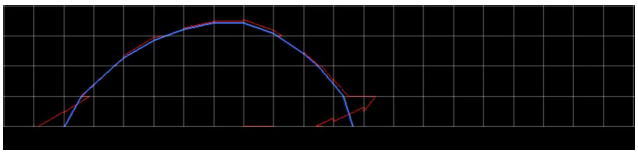
(a)



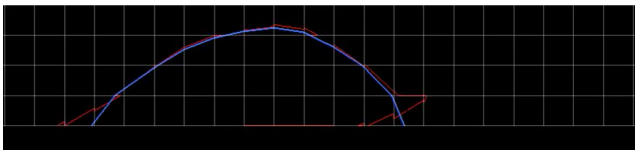
(b)



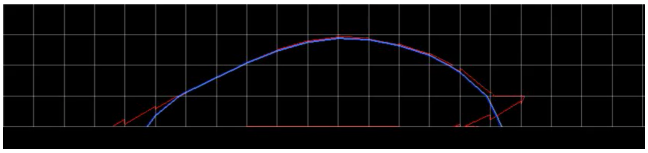
(c)



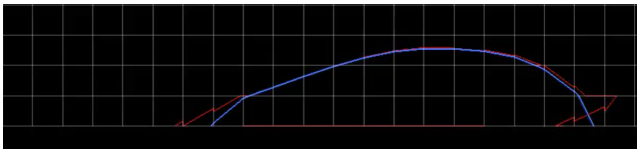
(d)



(e)

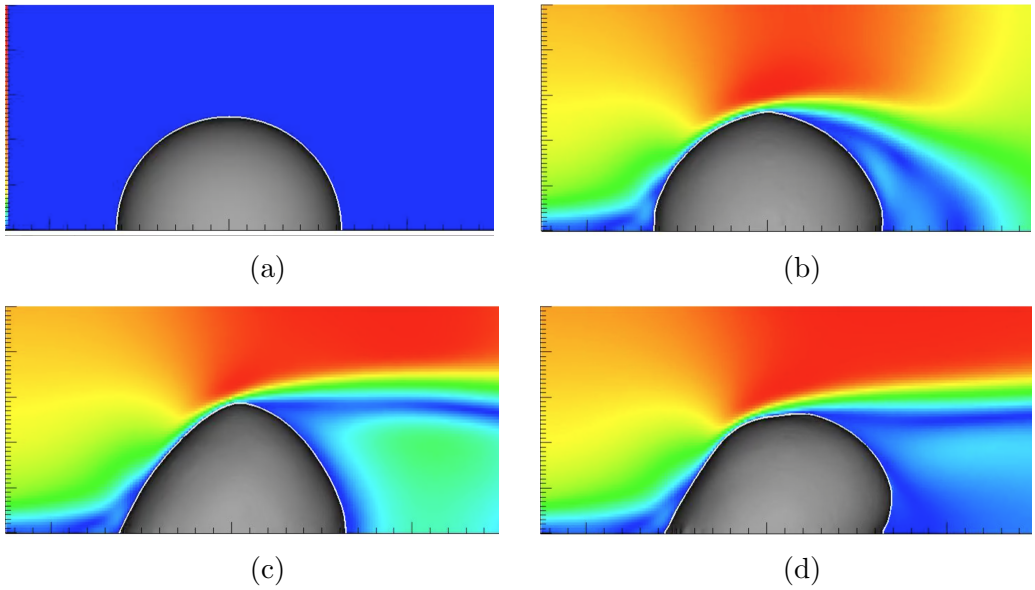


(f)



(g)

Figure 6.5: Progression of a droplet in two dimensions with defined contact angles. The red represents the PLIC and the blue represents the average VOF. The advancing and receding contact angles are 155° and 30°.



(a) (b)

(c) (d)

Figure 6.6: Progression of a droplet in three dimensions with defined contact angles over time.

CONCLUSION

Results

What started off being a relatively simple problem and that was to construct a method to define contact angles of a droplet on a surface in a gas flow turned out to be rather complex. There exists relatively little information about the implementation of contact angles in preexisting codes and as a result we decided to follow the simplest path in order to specify contact angles and that was to create bounds that specified the upper and lower bounds of the advancing and receding contact angles and implement a hysteresis angle which is more accurate than simply defining static contact angles to define a dynamic droplet. A major challenge of the implementation of a new contact angle method into the current computational platform was creating a new curvature method that was independent of the specification of the amount of liquid volume in the ghost cells and was just as accurate, if not more accurate, then the currently used height function method.

Many variables were considered in order to accurately determine the curvature along the interface of a droplet. Detailed tests were performed by altering these variables in order to determine the optimal settings that produced the best results. The results of these tests were then compared to the mesh decoupled height function method which has been shown to produce accurate results for well defined volume fractions. The height function method showed a strong second-order convergence when an exact liquid volume fraction was specified. However, the height function method failed to converge when the liquid volume fractions were second-order accurate. This becomes an issue for fine meshes and could have a large impact

on the final solution especially when the behavior of a liquid is highly dependent on interfacial and surface tension forces.

The new curvature method showed second-order convergence using exact points and normals. This method worked very well for course meshes when the volume fractions were not well defined and showed convergence even when a second-order VOF was used out to $N/D \approx 60$. This improved convergence on realistic VOF fields will provide better simulation results for many engineering problems involving droplets.

The 3D sphere test showed some interesting results. None of the cases or methods converged with second order accuracy. The height function method showed a lot of fluctuations in the results whereas the new curvature method was fairly consistent overall and showed at least first-order convergence.

When simulating a droplet in a gas flow with prescribed advancing and receding contact angles and using the new curvature method, a constrained least squares method was found to work best. While the unconstrained method was optimal for computing the initial curvature of a circle and a sphere, this method produced the largest errors when simulating a droplet in a gas flow as the simulation progressed.

The contact angle method still needs to be compared against experimental displacement data involving dynamic droplets. The greatest issue I see affecting the results of the contact angle code is defining the scale at which the contact angle is implemented relative to the radius of the droplet. The issue that arrises is that the contact angle code is only defined in the subcells along the bottom of the boundary and as the mesh is refined, the contact angle has less of an impact on the overall droplet. As a consequence, the results obtained by simulating a dynamic droplet along a boundary could vary considerably as the mesh is further refined.

Future Work

The final implementation of the contact angle code needs to be verified once the new curvature method is fully tested and compatible with larger and more complex multiphase simulations. The beginning of this work briefly tested the new curvature method in two-dimensions and a little bit in three dimensions using simple surfaces in order to verify that this curvature method was capable of being numerically accurate for refined mesh sizes. The next step would be to fully test the new curvature method in three-dimensions and to add in constraints which would allow the code to easily decipher between nearby interfaces such as droplets merging or branches that stem from a main flow similar to those found in atomization simulations.

The constrained least squares method worked well when simulating a droplet in a gas flow but it failed to produce accurate results of the interface during the initial testing. Another option to constrain the isosurface to a set of points is to add weight function to the least squares fit based on the distance to the points from the cell that the curvature is to be determined.

An important item still incomplete in the contact angle code is determining an analytical expression to solve for n_y given n_x and n_z, the liquid volume fraction (α), and the contact point. Currently, a bisection interpolation routine is used to solve for n_y which is slow and inaccurate unless the error is made to be within the machine accuracy and the number of iterations is increased dramatically which then decreases the overall performance of the routine.

Other future work in regards to the contact angle code is to modify the code in order to implement side wall wettability. Once the contact angle code has been validated against experiments, the contact angle code could be easily modified by

rotating the angles used in the current code so that n_z would be used to determine the contact angles.

Due to the implementation and extensive testing of the new curvature method, only relatively simple droplet cases using contact angles were simulated. The end goal of the contact angle work would be to prescribe contact angles to several droplets randomly oriented in a channel using random jet inputs that mimic droplets forming in a PEM fuel cell channel and study the behavior of the flow using actual gas flow inputs.

REFERENCES CITED

[1] S. Ravi Annapragada, J. Y. Murthy, and S. V. Garimella, "Prediction of droplet dynamics on an incline," *International Journal of Heat and Mass Transfer*, vol. 55, no. 5-6, pp. 1466–1474, 2012.

[2] S. J. Weinstein and K. J. Ruschak, "Coating Flows," *Annual Review of Fluid Mechanics*, vol. 36, pp. 29–53, 2004.

[3] M. Gorokhovski and M. Herrmann, "Modeling Primary Atomization," *Annual Review of Fluid Mechanics*, vol. 40, no. 1, pp. 343–366, 2008.

[4] K. W. Chu, B. Wang, D. L. Xu, Y. X. Chen, and A. B. Yu, "CFD-DEM simulation of the gas-solid flow in a cyclone separator," *Chemical Engineering Science*, vol. 66, no. 5, pp. 834–847, 2011.

[5] A. Bazylak, "Liquid water visualization in PEM fuel cells: A review," *International Journal of Hydrogen Energy*, vol. 34, no. 9, pp. 3845–3857, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.ijhydene.2009.02.084

[6] C. Sun, R. Hui, and J. Roller, "Cathode materials for solid oxide fuel cells: a review," *Journal of Solid State Electrochemistry*, vol. 14, no. 7, pp. 1125–1144, 2009. [Online]. Available: http://link.springer.com/10.1007/s10008-009-0932-0

[7] N. Abas, A. Kalair, and N. Khan, "Review of fossil fuels and future energy technologies," *Futures*, vol. 69, pp. 31–49, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.futures.2015.03.003

[8] A. J. McMichael, R. E. Woodruff, and S. Hales, "Climate change and human health: Present and future risks," *Lancet*, vol. 367, no. 9513, pp. 859–869, 2006.

[9] N. P. Brandon and D. J. Brett, "Engineering porous materials for fuel cell applications." *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 364, no. 1838, pp. 147–159, 2006.

[10] R. Anderson, E. Eggleton, and L. Zhang, "Development of Two-Phase Flow Regime Specific Pressure Drop Models for Proton Exchange Membrane Fuel Cells," pp. 1–29.

[11] M. Venkatraman, S. Shimpalee, J. W. Van Zee, S. I. Moon, and C. W. Extrand, "Estimates of pressure gradients in PEMFC gas channels due to blockage by static liquid drops," *International Journal of Hydrogen Energy*, vol. 34, no. 13, pp. 5522–5528, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.ijhydene.2009.04.039

[12] K. Tüber, D. Pócza, and C. Hebling, "Visualization of water buildup in the cathode of a transparent PEM fuel cell," *Journal of Power Sources*, vol. 124, no. 2, pp. 403–414, 2003.

[13] F. Chen, W.-M. Yan, H.-S. Chu, H.-Y. Wu, and C.-Y. Soong, "Analysis of thermal and water management with temperature-dependent diffusion effects in membrane of proton exchange membrane fuel cells," *Journal of Power Sources*, vol. 129, no. 2, pp. 127–137, 2004.

[14] V. R. Albertini, B. Paci, A. Generosi, S. Panero, M. A. Navarra, and M. di Michiel, "In Situ XRD Studies of the Hydration Degree of the Polymeric Membrane in a Fuel Cell," *Electrochemical and Solid-State Letters*, vol. 7, no. 12, pp. A519–A521, 2004. [Online]. Available: http://link.aip.org/link/?ESL/7/A519/1

[15] J. Park, X. Li, D. Tran, T. Abdel-Baset, D. S. Hussey, D. L. Jacobson, and M. Arif, "Neutron imaging investigation of liquid water distribution in and the performance of a PEM fuel cell," *International Journal of Hydrogen Energy*, vol. 33, no. 13, pp. 3373–3384, 2008.

[16] C. Neto, D. R. Evans, E. Bonaccurso, H.-J. Butt, and V. S. J. Craig, "Boundary slip in Newtonian liquids: a review of experimental studies," *Reports on Progress in Physics*, vol. 68, no. 12, pp. 2859–2897, 2005.

[17] U. O. M. Vázquez, W. Shinoda, P. B. Moore, C. C. Chiu, and S. O. Nielsen, "Calculating the surface tension between a flat solid and a liquid: A theoretical and computer simulation study of three topologically different methods," *Journal of Mathematical Chemistry*, vol. 45, no. 1, pp. 161–174, 2009.

[18] L. Gao and T. J. McCarthy, "Wetting 101," *Langmuir*, vol. 25, no. 24, pp. 14 105–14 115, 2009. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/la902206c

[19] P. G. De Gennes, "Wetting: Statics and Dynamics," 1985.

[20] L. Gao and T. J. McCarthy, "Wetting 101," *Langmuir*, vol. 25, no. 24, pp. 14 105–14 115, 2009.

[21] R. Tadmor and P. S. Yadav, "As-placed contact angles for sessile drops," *Journal of Colloid and Interface Science*, vol. 317, no. 1, pp. 241–246, 2008.

[22] E. Ramé, "The Interpretation of Dynamic Contact Angles Measured by the Wilhelmy Plate Method," *Journal of Colloid and Interface Science*, vol. 185, no. 1, pp. 245–251, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021979796945890

[23] O. Río and A. Neumann, "Axisymmetric Drop Shape Analysis: Computational Methods for the Measurement of Interfacial Properties from the Shape and Dimensions of Pendant and Sessile Drops." *Journal of colloid and interface science*, vol. 196, no. 2, pp. 136–147, 1997. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/9792739

[24] L. M. Pismen, "Mesoscopic hydrodynamics of contact line motion," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, vol. 206, no. 1-3, pp. 11–30, 2002.

[25] R. A. Hayes, J. Ralston, and B. Hydrodynamic, "The Molecular-Kinetic Theory of Wetting Robert A." *Langmuir*, vol. 10, no. 1, pp. 340–342, 1994.

[26] S. R. Ranabothu, C. Karnezis, and L. L. Dai, "Dynamic wetting: Hydrodynamic or molecular-kinetic?" *Journal of Colloid and Interface Science*, vol. 288, no. 1, pp. 213–221, 2005.

[27] T. D. Blake, "The physics of moving wetting lines," *Journal of Colloid and Interface Science*, vol. 299, no. 1, pp. 1–13, 2006.

[28] T.-S. Jiang, O. Soo-Gun, and J. C. Slattery, "Correlation for dynamic contact angle," *Journal of Colloid and Interface Science*, vol. 69, no. 1, pp. 74–77, 1979.

[29] Z. Keshavarz-Motamed, L. Kadem, and A. Dolatabadi, "Effects of dynamic contact angle on numerical modeling of electrowetting in parallel plate microchannels," *Microfluidics and Nanofluidics*, vol. 8, no. 1, pp. 47–56, 2010.

[30] T. D. Blake, A. Clarke, J. De Coninck, and M. J. de Ruijter, "Contact Angle Relaxation during Droplet Spreading: Comparison between Molecular Kinetic Theory and Molecular Dynamics," *Langmuir*, vol. 13, no. 7, pp. 2164–2166, 1997. [Online]. Available: http://dx.doi.org/10.1021/la962004g

[31] C. Chiquete, "Not even Hercules : the moving contact line problem," *Program*, pp. 1–17, 2007.

[32] C. Huh and L. Scriven, "Hydrodynamic model of steady movement of a solid/liquid/fluid contact line," *Journal of Colloid and Interface Science*, vol. 35, no. 1, pp. 85–101, 1971.

[33] R. B. DeBar, "Fundamentals of the KRAKEN code," *Technical Report*, no. March, pp. UCID–17 366, 1974. [Online]. Available: http://www.osti.gov/energycitations/product.biblio.jsp?osti{_}id=7227630$\backslash$nbackslash$nhttp://www.ntis.gov/search/product.aspx?ABBR=UCID17366

[34] C. Hirt and B. Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," *Journal of Computational Physics*, vol. 39, no. 1, pp. 201–225, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021999181901455

[35] M. Sussman, "A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles," *Journal of Computational Physics*, vol. 187, no. 1, pp. 110–136, 2003.

[36] B. Daly, "A technique for including surface tension effects in hydrodynamic calculations," *Journal of Computational Physics*, vol. 4, pp. 97–117, 1969. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021999169900424

[37] G. Tryggvason and J. Lu, *Direct Numerical Simulations of Multiphase Flows*, 2009.

[38] G. K. Karch, F. Sadlo, C. Meister, P. Rauschenberger, K. Eisenschmidt, B. Weigand, and T. Ertl, "Visualization of piecewise linear interface calculation," *IEEE Pacific Visualization Symposium*, pp. 121–128, 2013.

[39] J. E. Pilliod and E. G. Puckett, "Second-order accurate volume-of-fluid algorithms for tracking material interfaces," *Journal of Computational Physics*, vol. 199, no. 2, pp. 465–502, 2004.

[40] R. Scardovelli and S. Zaleski, "Analytical Relations Connecting Linear Interfaces and Volume Fractions in Rectangular Grids," *Journal of Computational Physics*, vol. 164, no. 1, pp. 228–237, 2000. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0021999100965677

[41] G. Malkawi, a. L. Yarin, and F. Mashayek, "Breakup mechanisms of electrostatic atomization of corn oil and diesel fuel," *Journal of Applied Physics*, vol. 108, no. 6, pp. 1–6, 2010.

[42] Z. T. Wang, A. M. Mitrašinović, and J. Z. Wen, "Investigation on Electrostatical Breakup of Bio-Oil Droplets," *Energies*, vol. 5, no. 11, pp. 4323–4339, 2012.

[43] a. Kourmatzis and J. S. Shrimpton, "Design and charge injection characteristics of an electrostatic dielectric liquid pulsed atomizer," *Journal of Electrostatics*, vol. 70, no. 3, pp. 249–257, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.elstat.2012.03.003

[44] a. Y. Vasil'ev and a. I. Maiorova, "Physical features of liquid atomization when using different methods of spraying," *High Temperature*, vol. 52, no. 2, pp. 250–258, 2014. [Online]. Available: http://link.springer.com/10.1134/S0018151X14010210

[45] M. Malik, E. S. C. Fan, and M. Bussmann, "Adaptive VOF with curvature-based refinement," *International Journal for Numerical Methods in Fluids*, vol. 55, no. 7, pp. 693–712, 2008.

[46] J. Hernández, J. López, P. Gómez, C. Zanzi, and F. Faura, "A new volume of fluid method in three dimensions - Part I: Multidimensional advection method with face-matched flux polyhedra," *International Journal for Numerical Methods in Fluids*, vol. 58, no. 8, pp. 897–921, 2008.

[47] G. Bornia, a. Cervone, S. Manservisi, R. Scardovelli, and S. Zaleski, "On the properties and limitations of the height function method in two-dimensional Cartesian geometry," *Journal of Computational Physics*, vol. 230, no. 4, pp. 851–862, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.jcp.2010.11.029

[48] M. M. Blane, "The 3L algorithm for fitting implicit polynomial curves and surfaces to data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. 298–313, 2000.

[49] O. Desjardins, G. Blanquart, G. Balarac, and H. Pitsch, "High order conservative finite difference scheme for variable density low Mach number turbulent flows," vol. 227, pp. 7125–7159, 2008.

[50] M. Owkes and O. Desjardins, "A mesh-decoupled height function method for computing interface curvature," *Journal of Computational Physics*, vol. 281, pp. 285–300, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999114007189

[51] S. a. Boyd, "Constrained least-squares," 2014. [Online]. Available: http://stanford.edu/class/ee103/lectures/constrained-least-squares/constrained-least-squares{_}slides.pdf