A Coupled Level Set and Volume of Fluid Method for computing 3d and axisymmetric Incompressible two-phase flows

Mark Sussman; Elbridge Gerry Puckett;

29 October 1998

Key words and phrases: level set, volume of fluid, curvature, incompressible
1991 Mathematics Subject Classification: 65M06, 76D05, 76T05.
*Work supported in part by NSF # DMS 97-06847 and the Dept. of Math, UCD
†Work supported in part by NSF # DMS 97-06847 and the Dept. of Math, UCD
Proofs to be sent to:
Mark Sussman
Florida State University
Tallahassee, FL 32306
sussman@math.fsu.edu
850-644-7194 (office)
Abstract

We present a coupled level set and volume of fluid method (CLS) for computing 3d and axisymmetric incompressible two-phase flows. The (CLS) method combines some of the advantages of the level set approach (LS) with that of the volume of fluid approach (VOF). We do direct comparisons with computations using the level set method, volume of fluid method, and the boundary integral method. We also compare our computations to experimental results for a rising gas bubble in liquid. Our comparisons focus on flows in which surface tension forces and changes in topology are present in the flow.
1 Introduction

In this work we present a coupled level set and volume of fluid method (CLS) for computing 3d and axisymmetric incompressible two-phase flows. Incompressible two-phase flow, e.g. air and water, can be difficult to model since the density ratio at the free-surface between the gas and liquid can be 1000:1 and the viscosity ratio can be 100:1. Furthermore, complications may arise when surface tension is present and/or when a change of topology occurs. An example of the flows we compute is shown in figure 15 where we show the results of a 4mm air bubble rising to the surface of an air/water interface and then bursting due to stiff surface tension effects. The jet then breaks up due to capillary instabilities, emitting satellite drops.

A motivation for our development of the coupled level set and volume of fluid method (CLS) is for modeling three dimensional ink-jet devices. In an ink-jet device, ink is ejected from a nozzle. The ink forms into a lead drop followed by a long thin cylindrical tail. Surface tension is the driving force that will break the long tail into smaller droplets (or cause the tail to “recoil” into a second, slightly smaller, satellite drop), therefore it is important to model surface tension accurately. Throughout this paper, we will be comparing the CLS method to either a straight level set method (LS) or straight volume of fluid method (VOF) keeping the application of the ink-jet problem in the back of our minds. We note, that although we will be focusing on CLS, LS, or VOF type methods, there are many other numerical models proposed for solving incompressible free-surface problems. Some of them are, boundary integral methods [8, 21, 9], front tracking methods [38], and particle-in-cell methods [27].

In the level set method[22, 36, 35, 32, 17], a smooth level set function $\phi(r, z, t)$ is used to represent the free surface. Liquid regions are the regions in which $\phi(r, z, t) > 0$ and gas regions are regions in which $\phi(r, z, t) < 0$. The free-surface is “implicitly” represented by the set of points in which $\phi(r, z, t) = 0$. The advantages that the level set method carries is in its simplicity; whether it be axisymmetric flows[35] or three dimensional flows[32]. One never has to explicitly reconstruct the free-surface when using the level set method; thus avoiding complicated interface reconstruction algorithms. Another advantage of the level set method is that one can accurately compute the local curvature of a free-surface using the level-set representation for the interface. In the implementation by Sussman et al.[36], the level set function is the signed normal distance to the free surface; i.e. $\phi(r, z, t) = -d$ in the gas and $\phi(r, z, t) = +d$ in the liquid where $d$ is the shortest distance from the point $(r, z)$ to the free surface at time $t$. From such a representation of the free-surface, the interfacial normal and mean curvature can be accurately represented as

$$\frac{\nabla \phi}{|\nabla \phi|}$$ (1)

and

$$\nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}$$ (2)
respectively. A disadvantage of the level set method is the fact that the discretization of the equation to advect the level set function,

\[ \phi_t + u \phi_r + v \phi_z = 0 \]  

(3)

where \((u, v)\) is the underlying velocity field, is prone to more numerical error than front-tracking methods or VOF methods. A common symptom of this error is loss of mass. Other related problems, reported by Rider and Kothe [26], can occur if the interface undergoes severe stretching and tearing. One motivation for developing the CLS method is the loss of mass problem. When computing a three dimensional ink-jet problem, we are limited to relatively coarse computational grids. On a coarse grid, a level set method will tend to lose enough mass that the tail will disappear before it breaks up into satellite drops. Before developing the CLS method, we tried computing with the velocity field on a coarse grid and the level set function on a fine grid. Unfortunately, this is an inefficient idea since the redistance step would have to be done on the fine grid and the surface tension time step constraint would have to derived relative to the mesh size on the fine grid.

Besides the level set approach, we have considered the volume of fluid method (VOF). In the volume of fluid method (VOF) [1, 11, 25], the volume fraction \(F(\Omega, t)\) is used to represent the free surface. Typically, \(\Omega\) represents a computational cell \(\Omega_{ij}\); e.g. \(\Omega_{ij} = (r_i \leq r \leq r_{i+1} \text{and } z_j \leq z \leq z_{j+1})\). If \(F(\Omega, t) = 1\), then the region represented by the set of points \(\Omega\) is all liquid. If \(F(\Omega, t) = 0\), then the region represented by the set of points \(\Omega\) is all gas. If \(0 < F(\Omega, t) < 1\), then \(\Omega\) contains both gas and liquid. One can define the volume fraction function \(F(\Omega, t)\) in terms of the level set function \(\phi(r, z, t)\). Since we have \(\phi > 0\) in the liquid and \(\phi < 0\) in the gas, one can define \(F(\Omega, t)\) as

\[ F(\Omega, t) = \frac{1}{|\Omega|} \int_{\Omega} H(\phi(r, z, t))r \, dr \, dz \]  

(4)

where \(H\) is the Heaviside function,

\[ H(\phi) = \begin{cases} 
1 & \text{if } \phi > 0 \\
0 & \text{otherwise.} \end{cases} \]  

(5)

An advantage of representing the free surface as volume fractions is the fact that one can write accurate algorithms for advecting the volume fraction function so that mass is conserved while still maintaining a sharp representation of the interface. For example, in the work of Piliod and Puckett [23] and [28], second order volume of fluid advection methods were presented which can accurately compute the rotation of a notched disk and maintain mass conservation. A disadvantage of the VOF method is the fact that it is difficult to compute accurate local curvatures from volume fractions. This is because the volume fractions transition sharply in regions of the free surface. Standard VOF methods compute the curvature by first mollifying the volume fractions in a special way (see [1, 11]). We have experienced difficulty with this approach. If one does not smooth enough, then the curvature for even a circle
will be highly oscillatory. This would spell disaster for the “stationary bubble test” (see section 6.3). If one smooths too much then the numerical algorithm will not “see” changes in curvature along the free surface (since too much smoothing has the effect of making the curvature constant along the free surface). This can spell disaster for dynamic problems such as the Zero Gravity Drop Oscillation problem (see section 6.2). In the CLS method, we do not smooth the curvature at all, instead the curvature is obtained via finite differences of the level set function which in turn is derived from the level set function and volume of fluid function at the previous time step.

In this work, we couple the level set method with the volume of fluid method; similarly as was done in the work of Bourlioux[10]. We extend upon Bourlioux’s Algorithm by combining the coupled volume of fluid level set (CLS) advection method with the equations for incompressible two-phase flow. We also modify Bourlioux’s algorithm in how we couple the level set method with the volume of fluid method. We show that we have comparable mass conservation properties as with other second order Volume-of-Fluid advection methods [23] and we also show that we can accurately compute surface tension driven flows by coupling the level set method with the volume of fluid method.

2 Governing Equations

The equations of motion for incompressible two phase flow can be written as,

\[
U_t + U \cdot \nabla U = -\frac{\nabla p}{\rho(\phi)} + \frac{1}{\rho(\phi)} \nabla \cdot (2\mu(\phi)D) - \frac{1}{\rho(\phi)} \gamma_k(\phi) \nabla H(\phi) + F
\]

\[
\nabla \cdot U = 0,
\]

and

\[
\phi_t + U \cdot \nabla \phi = 0.
\]

\(\phi\) is the level set function which is positive in the liquid and negative in the gas. The governing equation for the level set function (8) states that \(\phi\) remains constant on particle paths; i.e. if the zero level set is initialized as the free-surface between the liquid and gas, then the zero level set will always represent the free-surface. From the level set function, one can derive the density \(\rho(\phi)\), viscosity \(\mu(\phi)\), and curvature \(\kappa(\phi)\). Density and viscosity are written as,

\[
\rho(\phi) = \rho_g(1 - H(\phi)) + \rho_l H(\phi) \mu(\phi) = \mu_g(1 - H(\phi)) + \mu_l H(\phi),
\]

where \(H(\phi)\) is the Heaviside function defined by (5). The local mean curvature can be written as

\[
\kappa(\phi) = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}.
\]

\(D\) is defined as the rate of deformation tensor,

\[
D = (\nabla U) + (\nabla U)^T.
\]
\( \mathbf{F} \) is a body force; in our implementations, \( \mathbf{F} \) represents the force due to gravity \( \mathbf{F} = (0, g) \). \( \rho_g, \rho_l, \mu_g, \mu_l, \) and \( \gamma \) are defined to be the gas density, liquid density, gas viscosity, liquid viscosity, and surface tension coefficient respectively.

The fact that the surface tension force,

\[
\frac{1}{\rho(\phi)} \gamma \kappa(\phi) \nabla H(\phi)
\]

(12)

can be cast as a body force is due to the work of Brackbill, Zemach and Kothe [11] and more recently, for the level set method, due to the work of [12].

We remark, that when we discretize the level set equation (8), we shall simultaneously solve the following equation for the volume of fluid function \( F \),

\[
F_i + \nabla \cdot (\mathbf{U} F) = 0.
\]

(13)

At \( t = 0 \), \( F \) will be initialized in each computational cell \( \Omega_{ij} \),

\[
\Omega_{ij} = (r, z) | r_i \leq r \leq r_{i+1} \text{and} z_j \leq z \leq z_{j+1},
\]

(14)

to be,

\[
F_{ij} = \frac{1}{\Delta r \Delta z} \int_{\Omega_{ij}} H(\phi(r, z, 0)) r dr dz.
\]

(15)

Here, \( \Delta r \) and \( \Delta z \) are defined as \( r_{i+1} - r_i \) and \( z_{i+1} - z_i \) respectively.

### 2.1 Projection Methodology

The method used to solve for velocity and pressure is a variable density approximate projection method described by [4, 25]. We rewrite (6) as

\[
\mathbf{U}_t + \frac{1}{\rho(\phi)} \nabla p = \mathbf{V}(\mathbf{U}, \phi).
\]

(16)

We then take the divergence of both sides of (16) and use the fact that \( \nabla \cdot \mathbf{U}_t = 0 \) in order to reduce (6) and (7) into a single equation for pressure,

\[
\nabla \cdot \frac{1}{\rho} \nabla p = \nabla \cdot \mathbf{V}.
\]

(17)

After solving (17) for \( \nabla p \) the updated value for \( \mathbf{U}_t \) is

\[
\mathbf{U}_t = \mathbf{V} - \nabla p / \rho.
\]

(18)

For future reference, we define the projection operator \( \mathbf{P} \) as

\[
\mathbf{U}_t \equiv \mathbf{P}(\mathbf{V}).
\]

(19)

Combining (19) and (18) yields

\[
\nabla p / \rho = \mathbf{V} - \mathbf{U}_t \equiv \mathbf{V} - \mathbf{P}(\mathbf{V}) \equiv (I - \mathbf{P})(\mathbf{V}).
\]

(20)
3 The CLS advection algorithm

In this section, we describe how to advance the free surface using the coupled level set volume of fluid (CLS) advection algorithm. We shall describe the details for the axisymmetric case. The 3d algorithm follows analogously. We shall discretize our variables on a uniform grid with grid spacing of $\Delta r = \Delta z$. The discrete level set function $\phi_{i,j}^n$ and discrete volume fraction $F_{i,j}^n$ are located at cell centers. The motion of the free surface is determined by the velocity field derived from the equations for incompressible two-phase flow. The discrete velocity field is defined at cell edges $u_{i+1/2,j}$ and $v_{i,j+1/2}$, and satisfies the discrete divergence free condition,

$$D^{MAC}U = \frac{(ru)_{i+\frac{1}{2},j} - (ru)_{i-\frac{1}{2},j}}{r_i \Delta r} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta z}. \tag{21}$$

A diagram of where the discrete variables are located in relation to the computational grid is shown in Figure 1. $J$ represents the index of the computational cell closest to the top physical boundary.

The equations governing the interface motion are

$$\phi_t + \nabla \cdot (U \phi) = 0 \tag{22}$$

and

$$F_i + \nabla \cdot (UF) = 0. \tag{23}$$

Remark: The conservative formulation of the level set equation (22) is equivalent to (3) since $\nabla \cdot U = 0$.

We shall assume that the level set function $\phi_{i,j}^0$ is initialized as the signed normal distance from the initial position of the free surface. The volume fraction function $F_{i,j}^0$ shall be initialized as the fraction of liquid fluid contained in cell $(i,j)$. In other words,

$$F_{i,j}^0 = \frac{1}{\Delta r \Delta z} \int_{\Omega_{ij}} H(\phi(r,z,0))r dr dz. \tag{24}$$

where

$$\Omega_{ij} = (r,z) | r_i \leq r \leq r_{i+1} \text{and} z_j \leq z \leq z_{j+1}. \tag{25}$$

Given $\phi_{i,j}^n$, $F_{i,j}^n$ and $U$, we use a “coupled” second order conservative operator split advection scheme in order to find $\phi_{i,j}^{n+1}$ and $F_{i,j}^{n+1}$. For axisymmetric flows, the operator split algorithm for a general scalar $s$ follows as

$$\tilde{s}_{i,j} = s_{i,j} + \frac{\Delta t}{r_i \Delta r} (r_{i-1/2} G_{i-1/2,j} - r_{i+1/2} G_{i+1/2,j}) \tag{26}$$

$$s_{i,j}^{n+1} = \tilde{s}_{i,j} + \frac{\Delta t}{\Delta z} (G_{i,j-1/2} - G_{i,j+1/2}) + \tilde{s}_{i,j}(v_{i,j+1/2} - v_{i,j-1/2}). \tag{27}$$
where \( G_{i+1/2,j} = s_{i+1/2,j} u_{i+1/2,j} \) denotes the flux of \( s \) across the right edge of the \((i,j)\)th cell and \( \tilde{G}_{i,j+1/2} = \tilde{s}_{i,j+1/2} v_{i,j+1/2} \) denotes the flux across the top edge of the \((i,j)\)th cell. For 3d flows, the operator split algorithm for a general scalar \( s \) follows as

\[
\begin{align*}
\tilde{s}_{i,j,k} &= s_{i,j,k}^n + \frac{\Delta t}{\Delta x} (G_{i-1/2,j,k} - G_{i+1/2,j,k}) - \frac{\Delta t}{\Delta x}(u_{i+1/2,j,k} - u_{i-1/2,j,k}), \\
\hat{s}_{i,j,k} &= \tilde{s}_{i,j,k} + \frac{\Delta t}{\Delta y} (\tilde{G}_{i,j-1/2,k} - \tilde{G}_{i,j+1/2,k}) - \frac{\Delta t}{\Delta y}(v_{i,j+1/2,k} - v_{i,j-1/2,k}), \\
\bar{s}_{i,j,k} &= \hat{s}_{i,j,k} + \frac{\Delta t}{\Delta z} (\hat{G}_{i,j,k-1/2} - \hat{G}_{i,j,k+1/2}) - \frac{\Delta t}{\Delta z}(w_{i,j,k+1/2} - w_{i,j,k-1/2}), \\
s_{i,j,k}^{n+1} &= s_{i,j,k} - \Delta t (\frac{\partial}{\partial x} (u_{i+1/2,j,k} - u_{i-1/2,j,k}) + \frac{\partial}{\partial y} (v_{i,j+1/2,k} - v_{i,j-1/2,k}) + \frac{\partial}{\partial z} (w_{i,j,k+1/2} - w_{i,j,k-1/2})) \tag{31}
\end{align*}
\]

Remarks:

- Although (26) and (27) are not in conservation form, the scalar \( s \) is still conserved since \( u_{i+1/2,j} \) and \( v_{i,j+1/2} \) satisfy (21). The form that we use to difference the fluxes in (26) and (27) was used by [25]. The 3d analogue represented by (28) thru (31) reduces to the two dimensional case when any one of the discrete derivatives \( u_x, v_y \) or \( w_z \) are zero.

- The operator split procedure described above is made second order accurate by alternating the starting sweep direction at each time step; i.e. by employing “Strang splitting” [31].

The scalar flux \( s_{i+1/2,j} \) is computed differently depending on whether \( s \) represents the level set function \( \phi \) or the volume fraction function \( F \). For the case when \( s \) represents the level set function \( \phi \) we have the following representation for \( s_{i+1/2,j} \) when \( u_{i+1/2,j} > 0 \),

\[
s_{i+1/2,j} = s_{i,j}^n + \frac{\Delta t}{2} (1 - u_{i+1/2,j} \frac{\Delta t}{\Delta r}) \frac{\partial}{\partial r} s_{i+1,j}^n - s_{i-1,j}^n \tag{32}
\]

and when \( u_{i+1/2,j} < 0 \),

\[
s_{i+1/2,j} = s_{i+1,j}^n - \frac{\Delta t}{2} (1 + u_{i+1/2,j} \frac{\Delta t}{\Delta r}) \frac{\partial}{\partial r} s_{i+2,j}^n - s_{i,j}^n.
\]

The above discretization is motivated by the predictor corrector method described in [5] and the references therein. The scalar flux \( s_{i+1/2,j} \) is obtained by extrapolating \( s \) in both space and time. Below, we show an example for the case when \( u_{i+1/2,j} > 0 \),

\[
s_{i+1/2,j} \approx s_{i,j} + \frac{\Delta r}{2} r_{r,ij} + \frac{\Delta t}{2} s_{l,ij}. \tag{33}
\]
For an operator split algorithm we only solve for one direction at a time. This means, for example, that we are solving

\[ s_t + us_r = 0. \]

We can substitute \( s_{t,i,j} = -us_{r,i,j} \) into (33) in order to obtain,

\[ s_{i+1/2,j} \approx s_{i,j} + \frac{\Delta r}{2} (1 - u \frac{\Delta t}{\Delta r}) s_{r,i,j}. \]

If we replace \( u \) with \( u_{i+1/2,j} \) and \( s_{r,i,j} \) with \( s_{r,i,j}^n - s_{r,i,j}^{n-1/2} \), then we recover (32).

For the case when \( s \) represents the volume of fluid function \( F \) we have the following representation for \( s_{i+1/2,j} \) when \( u_{i+1/2,j} > 0 \),

\[
s_{i+1/2,j} = \frac{\left( \int_{z_{i+1/2}}^{z_{i-1/2}} \int_{r_{i+1/2}}^{r_{i-1/2}} H(\phi_{i-1/2,j}^n(r,z)) r dr dz \right)}{(r_{i+1/2} - \frac{1}{2} u_{i+1/2,j} \Delta t) u_{i+1/2,j} \Delta t} \tag{34}
\]

and when \( u_{i+1/2,j} < 0 \),

\[
\left( \int_{z_{i+1/2}}^{z_{i-1/2}} \int_{r_{i+1/2}}^{r_{i-1/2}} H(\phi_{i-1/2,j}^n(r,z)) r dr dz \right) \frac{(r_{i+1/2} - \frac{1}{2} u_{i+1/2,j} \Delta t)}{u_{i+1/2,j} \Delta t} \tag{35}
\]

The term \( \phi_{i,j}^n(R, z) \) found in (34) and (35) represents the linear reconstruction of the interface in cell \((i,j)\). In other words, \( \phi_{i,j}^n(R, z) \) has the form

\[
\phi_{i,j}^n(R, z) = a_{i,j}(r - r_i) + b_{i,j}(z - z_j) + c_{i,j}. \tag{36}
\]

The coefficients \( a_{i,j}, b_{i,j} \) and \( c_{i,j} \) are first chosen so that (36) represents the best fit line for the piece of the zero level set passing thru cell \((i,j)\). In other words, \( a, b \) and \( c \) minimize the following error:

\[
E_{i,j} = \int_{r_{i-1/2}}^{r_{i+1/2}} \int_{z_{i-1/2}}^{z_{i+1/2}} |H'(\phi)(\phi - a_{i,j}(r - r_i) - b_{i,j}(z - z_j) - c_{i,j})|^2. \tag{37}
\]

In order to solve for \( a, b, \) and \( c \), we minimize the discretized error,

\[
E_{i,j}^\Delta = \sum_{\nu = i-1}^{i+1} \sum_{j' = j-1}^{j+1} w_{\nu,j'} H'(\phi_{\nu,j'})(\phi_{\nu,j'} - a_{i,j}(r_{\nu} - r_i) - b_{i,j}(z_{j'} - z_j) - c_{i,j})^2. \tag{38}
\]

The discrete weights \( w_{\nu,s} \) are chosen so that (38) is an approximation to (37). For the computations we show, we have \( w_{r,s} = 16 \) for \( r = s = 0 \) and \( w_{r,s} = 1 \) for \( r \neq 0 \) or \( s \neq 0 \). We have tried other values for \( w_{r,s} \) with little effect on the accuracy of the computation. \( H'(\phi) \) represents the smoothed Heaviside function with thickness \( \epsilon \); in our computations, we always have \( \epsilon = \sqrt{2} \Delta r \). The resulting equations for \( a, b, c \) as a result of minimizing (38) is a 3x3 linear system.
The intercept $c_{i,j}$ is corrected so that the line represented by (36) cuts out the same volume in cell $(i,j)$ as specified by $F^n_{i,j}$. In other words, the following equation is solved for $c_{i,j}$:

$$\int_{z_{i-1/2}}^{z_{i+1/2}} \int_{r_{i-1/2}}^{r_{i+1/2}} H(a_{i,j}(r - r_i) + b_{i,j}(z - z_j) + c_{i,j}) r dr dz = F^n_{i,j}$$

(39)

Since $H$ is a Heaviside function defined as $H(\phi) = 1$ if $\phi > 0$ and $H(\phi) = 0$ otherwise, we solve 39 by use of the Newton iteration method. We remark that the algorithm is simplified by first rotating the grid axis so that the normal represented by $a_{i,j}$ and $b_{i,j}$ points away from the lower left hand corner of the $(i,j)$ computational cell. The coefficients $a_{i,j}, b_{i,j}$ and $c_{i,j}$ are also rescaled so that $a_{i,j}^2 + b_{i,j}^2 = 1$ and the new intercept represents the normal distance to the lower left hand corner of the computational cell.

The integrals in (34) and (35) are evaluated by finding the volume cut out of the region of integration by the line represented by (36).

Remarks:

- In comparison to setting,

$$a_{i,j} = \frac{1}{2\Delta r}(\phi_{i+1,j} - \phi_{i-1,j})$$

(40)

$$b_{i,j} = \frac{1}{2\Delta z}(\phi_{i,j+1} - \phi_{i,j-1})$$

(41)

the minimization procedure described above proved to be more accurate for simple tests such as the translation of a circle or Zalesak’s problem.

- In our numerical procedure, we disallow partial volume fractions in cells in which $|\phi| > \Delta r$. This guarantees that we only reconstruct the interface in cells in which $|\phi| \leq \Delta r$; thus the linear system that results from minimizing (38) will always have a solution.

The scalar flux $\tilde{s}_{i,j+1/2}$ is computed in the same manner as $s_{i+1/2,j}$. For the case when $s$ represents the level set function $\phi$, we have the following representation for $\tilde{s}_{i,j+1/2}$ when $v_{i,j+1/2} > 0$,

$$\tilde{s}_{i,j+1/2} = \tilde{s}_{i,j} + \frac{\Delta z}{2}(1 - v_{i,j+1/2}\frac{\Delta t}{\Delta z})\tilde{s}_{i,j+1} - \tilde{s}_{i,j-1}$$

(42)

and when $v_{i,j+1/2} < 0$,

$$\tilde{s}_{i,j+1/2} = \tilde{s}_{i+1,j} - \frac{\Delta z}{2}(1 + v_{i,j+1/2}\frac{\Delta t}{\Delta z})\tilde{s}_{i,j+2} - \tilde{s}_{i,j}$$

(43)

For the case when $\tilde{s}$ represents the volume of fluid function $F$ we have the following representation for $\tilde{s}_{i,j+1/2}$ when $v_{i,j+1/2} > 0$,

$$\tilde{s}_{i,j+1/2} = \frac{\left(\int_{z_{i+1/2}}^{z_{i+1/2}} \int_{r_{i-1/2}}^{r_{i+1/2}} H(\phi^R_{i,j}(r,z)) r dr dz\right)}{r_i v_{i,j+1/2}\Delta t \Delta r}$$

(44)
and when \( v_{i,j+1/2} < 0 \),

\[
\tilde{s}_{i,j+1/2} = \frac{\text{\(J_{z_i+j+1/2}^{z_i+j+1/2} r_{t_i+j+1/2}^{t_i+j+1/2} H(\tilde{\phi}_{i,j}^R(r,z))rdz\)}}{r_i |v_{i,j+1/2}| \Delta t \Delta r} \tag{45}
\]

The linear reconstruction \( \tilde{\phi}_{i,j}^R(r,z) \) found in (44) and (45) has an analogous form as (36),

\[
\tilde{\phi}_{i,j}^R(r,z) = a_{i,j}(r - r_i) + b_{i,j}(z - z_j) + c_{i,j}. \tag{46}
\]

After \( \phi^{n+1} \) and \( F^{n+1} \) have been updated according to (26) and (27) we “couple” the level set function to the volume fractions by assigning the level set function \( \phi^{n+1} \) to be the exact signed normal distance to the reconstructed interface. The algorithm to find the signed normal distance in a strip of \( K \) cells about the reconstructed interface is as follows:

1. Truncate the volume fractions:

\[
F_{i,j}^{n+1} = \begin{cases} 
0 & \text{if } F_{i,j}^{n+1} \leq 0 \text{ or } \phi_{i,j}^{n+1} < -\Delta r \\
1 & \text{if } F_{i,j}^{n+1} \geq 1 \text{ or } \phi_{i,j}^{n+1} > \Delta r \\
F_{i,j}^{n+1} & \text{otherwise}
\end{cases} \tag{47}
\]

2. Tag all computational cells \((i,j)\)

3. In each computational cell \((i,j)\), check to see if

\[
\phi_{i,j}^{n+1} \phi_{i',j'}^{n+1} \leq 0
\]

for some \(|i - i'| \leq 1, |j - j'| \leq 1\); if there is a \((i', j')\) such that (48) is satisfied, then perform the following steps:

(a) if

\[
0 < F_{i,j}^{n+1} < 1
\]

and

\[
\phi_{i,j}^{n+1} (\phi_{i,j}^{n+1} + \phi_{i',j'}^{n+1}) \leq 0 \quad \text{for some } |i - i'| \leq 1, |j - j'| \leq 1, \tag{50}
\]

then construct the linear reconstruction \( \phi_{i,j}^{n+1,R}(r,z) \) (36),

\[
\phi_{i,j}^{n+1,R}(r,z) = a_{i,j}(r - r_i) + b_{i,j}(z - z_j) + c_{i,j}. \tag{51}
\]

If (49) or (50) is not satisfied then mark all of cell \((i,j)\) face centroids and corners as either “positive” or “negative” depending on the sign of \( \phi_{i,j}^{n+1} \).

If both (49) and (50) are satisfied, mark all of cell \((i,j)\) face centroids and corners according to the sign of \( \phi_{i,j}^{n+1,R}(r,z) \) evaluated at the face centroids and corners.
(b) For each cell \((i', j')\), \((i' - i)^2 + (j' - j)^2 < K^2\) and \((i' - i)^2 + (j' - j)^2 < (|\phi_{\ell,j'}^{n+1}|/\Delta r + 2)^2\) do the following steps; we refer the reader to the diagram in Figure 3.

i. Determine the closest point on the boundary of cell \((i, j)\) to \((r_{i'}, z_{j'})\) (this point will always either be at the corner or face centroid of the cell boundary). If the sign of the level set function at the closest point is opposite of \(\phi_{i,j}^{n+1}\), then set \(d\), the shortest distance associated with cells \((i, j)\) and \((i', j')\), equal to the distance from \((r_{i'}, z_{j'})\) to the closest point on the boundary of cell \((i, j)\). If the sign of the level set function at the closest point is the same as \(\phi_{i,j}^{n+1}\) and \((49,50)\) are both satisfied, then let \(d\) be the shortest distance between \((r_{i'}, z_{j'})\) and the line segment represented by \(\phi_{i,j}^{n+1,R}(r, z)\).

ii. Update \(\phi_{i',j'}^{n+1}\) using \(d\):

\[
\phi_{i',j'}^{n+1} = \begin{cases} 
\text{sign}(\phi_{i',j'}^{n+1})d & \text{if } d < |\phi_{i,j}^{n+1}|\text{ or cell } (i', j') \text{ is tagged} \\
\phi_{i,j}^{n+1} & \text{otherwise}
\end{cases}
\]

iii. Untag cell \((i', j')\).

4. For cells \((i, j)\) which are still tagged, we have

\[
\phi_{i,j}^{n+1} = \begin{cases} 
-K\Delta r - \Delta r & \text{if } \phi_{i,j}^{n+1} < 0 \\
K\Delta r + \Delta r & \text{if } \phi_{i,j}^{n+1} > 0
\end{cases}
\]  

(52)

Remarks:

- The coupling between the level set function \(\phi\) and the volume of fluid function \(F\) occurs when computing the normal of the reconstructed interface (36) and also when assigning the level set function with the exact signed normal distance to the reconstructed interface.

- In order to find the shortest distance between the cell center \((i', j')\) and the line segment represented by \(\phi_{i,j}^{n+1,R}(r, z)\) (51), one first re-scales (51) so that \(a_{i,j}^2 + b_{i,j}^2 = 1\). The distance is then \(d = \phi_{i,j}^{n+1,R}(r_{i'}, z_{j'})\). The point \(x_c = (r_{i'}, z_{j'}) - d\nabla \phi_{i,j}^{n+1,R}\) is the point where the normal extension from \((i', j')\) to \(\phi_{i,j}^{n+1,R}(r, z)\) intersects \(\phi_{i,j}^{n+1,R}(r, z)\). If \(x_c\) falls outside of cell \((i, j)\), then the shortest distance between \((i', j')\) and \(\phi_{i,j}^{n+1,R}(r, z)\) must be the distance from \((i', j')\) to one of the end points of the line segment represented by \(\phi_{i,j}^{n+1,R}(r, z)\). In three dimensions, it becomes only slightly more complicated. The step for finding the normal distance to the plane represented by \(\phi_{i,j,k}^{n+1,R}(x, y, z)\) is simply \(d = \phi_{i,j,k}^{n+1,R}(x_{i'}, y_{j'}, z_{k'})\) (assuming the coefficients are appropriately scaled). As for the axisymmetric case, \(x_c = (x_{i'}, y_{j'}, z_{k'}) - d\nabla \phi_{i,j,k}^{n+1,R}\). The only added complication in three dimensions is finding the shortest distance to the end points of the reconstructed plane. But this can be handled by projecting the point \((x_{i'}, y_{j'}, z_{k'})\) onto the plane that coincides with each of cell \((i, j, k)\)'s faces and
then finding the distance from the projected point to the intersection between the reconstructed plane and the cell face. The procedure in this case reduces to the procedure for the 2d case.

- One difference between our algorithm and the one presented by Bourlioux [10] for coupling $\phi$ to $F$ is in how the new level set function $\phi^{n+1}$ is reinitialized using the volume of fluid function $F^{n+1}$. In [10], the level set function $\phi_{i,j}^{n+1}$ is modified to be the intercept $c_{i,j}$ (36) of the reconstructed interface $\phi_{i,j}^{n,R}(r, z)$ in cells that contain a piecewise linear reconstruction. For cells that do not contain the piecewise linear reconstruction, Bourlioux used the resistence iteration proposed by [36]. In our computations we have tried both approaches, and found more accurate results when assigning $\phi^{n+1}$ to be the exact distance from the reconstructed interface. In our implementation, we only need $\phi^{n+1}$ to be the exact distance within $K$ cells where $K < 5$ ($K$ is dependent on the “Interface Thickness”, see section 5). The speed for our algorithm is thus $O(K^2N)$ where $N$ is the number of computational cells that contain a volume of fluid reconstruction. This is the same speed estimate as for a redistance iteration within a tube $K$ cells thick about the zero level set.

- In the previous remark, it was recommended that one find the exact distance to the reconstructed interface when implementing the CLS algorithm. It is not recommended that one reconstruct the interface if one is only using the level set approach for moving the free surface. In this case, the redistance iteration [34, 33] is recommended since interface reconstructions using only the level set function without the aid of volume fractions will incur considerably more error than the iteration approach outlined in [34, 33].

4 Temporal Discretization

Our discretization procedure for approximating (6) is based on the variable density projection method described by Bell et al. [5], Bell and Marcus [7], Almgren et al. [4], and Puckett et al. [25]. The discrete velocity field $U_{i,j}^n$, level set function $\phi_{i,j}^n$ and volume fractions $F_{i,j}^n$ are located at cell centers. The pressure $p_{i+1/2,j+1/2}^{n-1/2}$ is located at cell corners. A diagram of where the discrete variables are located in relation to the computational cell closest to the top physical boundary.

The time stepping procedure is based on the Crank-Nicholson method. At the beginning of each time step, we are given the velocity $U^n$, level set function $\phi^n$, and volume fractions $F^n$ at time $t^n$. We are also given the lagged pressure gradient $p_{i+1/2,j+1/2}^{n-1/2}$. The density $\rho^n = \rho(\phi^n)$, viscosity $\mu^n = \mu(\phi^n)$ and Heaviside function $H^n = H(\phi^n)$ are given at time $t^n$ since they are functions of $\phi^n$.

We discretize (6) and (8) in time using the steps below; for details of the spatial discretization of the nonlinear terms, viscous terms, surface tension terms and projection step, see the appendix (section 9).
1. Predict the edge based velocity field $u_{i+1/2,j}^{ADV,n+1/2}$ and $v_{i,j+1/2}^{ADV,n+1/2}$ using an explicit predictor-corrector scheme. The edge based velocity field shall be discretely divergence-free; i.e. they satisfy (21). A description of how $u_{i+1/2,j}^{ADV,n+1/2}$ and $v_{i,j+1/2}^{ADV,n+1/2}$ are formed is described in the appendix (section 9.1).

2. Given $\phi^n$, $F^n$ and $U^{ADV,n+1/2}$, apply the CLS advection algorithm described in section 3 in order to determine $\phi^{n+1}$ and $F^{n+1}$.

Once $\phi^{n+1}$ is obtained, the following quantities are updated:

$$\phi^{n+1/2} = \frac{1}{2}(\phi^n + \phi^{n+1})$$  \hspace{0.5cm} (53)

$$\rho^{n+1/2} = \rho(\phi^{n+1/2})$$  \hspace{0.5cm} (54)

$$\mu^{n+1/2} = \mu(\phi^{n+1/2})$$  \hspace{0.5cm} (55)

3. Semi-implicit viscous solve for the intermediate velocity $U^*$:

$$\frac{U^* - U^n}{\Delta t} = -[(U \cdot \nabla)U]^{n+1/2} - \frac{Gp^{n-1/2}}{\rho^{n+1/2}} + \frac{L^* + L^n}{2\rho^{n+1/2}} - \frac{M^{n+1/2}}{\rho^{n+1/2}} + F. \hspace{0.5cm} (56)$$

$L$ is a second-order finite difference approximation to $\nabla \cdot (2\mu(\phi)D)$, $M$ is a finite difference approximation to $\gamma\kappa(\phi)\nabla H$ and $Gp$ is an approximation to $\nabla p$. In the appendix (section 9.2) we give a description of $Gp$, $L$ and $M$. The nonlinear advection term $[(U \cdot \nabla)U]^{n+1/2}$ is evaluated using an explicit predictor-corrector scheme and requires only the available data at $t^n$. In the appendix (section 9.1), we give a description of how $[(U \cdot \nabla)U]^{n+1/2}$ is discretized. The density $\rho$, viscosity $\mu$, Heaviside function $H$ and curvature $\kappa$ are constructed from the level set function calculated at time $t^{n+1/2}$ in the level set advection step (53). The lagged pressure gradient $Gp^{n-1/2}$ and force $F$ are treated as source terms.

Equation (56) when discretized results in a coupled parabolic solve for all velocity components of $U^*$. We use multigrid as an iteration method for solving (56).

4. Projection Step for $U^{n+1}$:

$$\frac{U^{n+1} - U^n}{\Delta t} = \mathcal{P} \left( \frac{U^* - U^n}{\Delta t} \right) \hspace{0.5cm} (57)$$

$$\frac{1}{\rho^{n+1/2}}Gp^{n+1/2} = \frac{1}{\rho^{n+1/2}}Gp^{n-1/2} + (I - \mathcal{P}) \left( \frac{U^* - U^n}{\Delta t} \right).$$

$\mathcal{P}$ represents the discretization of the projection operator (19). In the appendix (section 9.3) we give a description of $\mathcal{P}$. 

15
4.1 Timestep

The timestep $\Delta t$ at time $t^n$ is determined by restrictions due to the CFL condition, gravity, viscosity and surface tension ([36, 11]):

$$\Delta t < \min_{i,j} \left( \sqrt{\frac{(\rho_1 + \rho_2)}{8\pi \gamma}} \Delta r^{3/2} + 3 \frac{\rho^n \Delta r^2}{\mu^n}, \left| \frac{\Delta r}{u^n} \right| + \sqrt{\left| \frac{u^n}{2} \right|^2 + \epsilon^2} \right),$$

where

$$\mathcal{F}^n = \left| - \frac{Gp^{n-1/2}}{\rho^n} + \frac{\mu^n}{\rho^n} - \frac{\mathcal{M}^n}{\rho^n} + \mathcal{F} \right|.$$ 

The last time step constraint is justified through the following simplified analysis. If we consider the simplified equation,

$$u_t = F$$

$$u(t_n) = u_n$$

then the solution at $t_{n+1}$ is

$$u(t_{n+1}) = u_n + \Delta t F.$$ 

We require a “CFL” type condition,

$$u(t_{n+1}) \Delta t < \Delta r$$

The resulting equation for $\Delta t$ is

$$(u_n + \Delta t F) \Delta t < \Delta r.$$ 

5 Interface Thickness

We shall give the interface a thickness as was done in the work of [38] and [36]. Numerically, we substitute the smoothed Heaviside function $H_c(\phi)$ for the sharp Heaviside function $H(\phi)$. The smoothed Heaviside function is defined as

$$H_c(\phi) = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \frac{1}{2} [1 + \frac{\phi}{\epsilon} + \frac{1}{\pi} \sin(\pi \phi/\epsilon)] & \text{if } |\phi| \leq \epsilon \\ 1 & \text{if } \phi > \epsilon \end{cases}$$ (58)

Assume that $\phi$ represents the signed normal distance to the free surface. The 1/2 contour of the sharp Heaviside function $H(\phi)$ will show up on a contour plot with jagged or staircase contours. Although $\phi$ is smooth, $H(\phi)$ has a jump at the zero level set. A contour plot of $H_c(\phi)$ where $\epsilon = \alpha \Delta r$ will not show up having a jagged
shape when \( \alpha > 1 \). By giving the interface a thickness of 2\( \epsilon \) we eliminate problems when solving (89) and also when discretizing the surface tension term

\[
\gamma \frac{\kappa(\phi) \nabla H(\phi)}{\rho(\phi)}.
\]

In our algorithm, the front will have a uniform thickness since the level set function \( \phi \) will always represent the signed normal distance to the free surface due to our CLS advection algorithm (see section 3).

6 Results

In this section we test the CLS method on 3d and axisymmetric problems in which surface tension effects and changes in topology are present. We compare the CLS method with the level set method [36, 35], the VOF method [1, 23, 20] and the boundary integral method [35]. We shall show that our coupled level set volume of fluid method (CLS) has comparable accuracy to the level set method for computing most surface tension driven flows. We also show that our CLS method conserves mass to within a fraction of a percent for all of our test cases.

6.1 Validation of the CLS advection algorithm

We consider two problems in this section; the translation of a circle in a periodic domain and the rotation of a notched disk (Zalesak’s problem [39]). These problems are 2-d problems and not axisymmetric problems.

For the translation of a circle, we have a 4x4 periodic domain and a unit circle initialized at the center of the domain. The velocity field is specified as \( u = 1 \) and \( v = 0 \). In Tables 1 and 2, we display the error at \( t = 4 \) for grid resolutions ranging from 32x32 to 256x256. These errors are compared to the errors reported by [23] when using the ELVIRA method for volume of fluid reconstruction. The error is measured as

\[
E(t) = \frac{1}{L} \sum_{i,j} \int_{\Omega_{ij}} |H(\phi_e(t)) - H(\phi_c(t))| d\mathbf{x}
\]  \hspace{1cm} (59)

where \( \phi_e \) is the level set function representing the exact solution and \( \phi_c \) is the level set function representing the computed solution. \( L \) is the perimeter of the interface. The integral in (59) is approximated by partitioning \( \Omega_{ij} \) into 128x128 rectangles and then applying the midpoint rule. The values of \( \phi_e \) and \( \phi_c \) at the midpoint of each rectangle are obtained via bilinear interpolation.

In spite of the fact that we truncate the volume fractions which are farther than \( \Delta x \) from the zero level set, the maximum mass fluctuation for the translating circle problem is \( 1.0E-9 \). The mass is measured as

\[
V(t) = \sum_{i,j} F_{i,j} \Delta r \Delta z.
\]  \hspace{1cm} (60)
Table 1: The error $E(1.0)$ after translating a unit circle one unit in time with CFL number 1

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>CLS</th>
<th>ELVIRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/8</td>
<td>0.000523</td>
<td>0.000610</td>
</tr>
<tr>
<td>1/16</td>
<td>0.000128</td>
<td>0.000160</td>
</tr>
<tr>
<td>1/32</td>
<td>0.000031</td>
<td>0.000040</td>
</tr>
<tr>
<td>1/64</td>
<td>0.000008</td>
<td>0.000010</td>
</tr>
</tbody>
</table>

Table 2: The error $E(1.0)$ after translating a unit circle one unit in time with CFL number 1/32

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>CLS</th>
<th>ELVIRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/8</td>
<td>0.001110</td>
<td>0.001380</td>
</tr>
<tr>
<td>1/16</td>
<td>0.000327</td>
<td>0.000437</td>
</tr>
<tr>
<td>1/32</td>
<td>0.000097</td>
<td>0.000125</td>
</tr>
<tr>
<td>1/64</td>
<td>0.000029</td>
<td>0.000038</td>
</tr>
</tbody>
</table>

For the rotation of a notched disc, we have a 1x1 periodic domain and a notched circle of radius 0.15 units positioned at (0.5, 0.75). The width of the notch is 0.05 units and the height of the notch is 0.25 units. The velocity field is initialized as

$$u = \frac{\pi}{3.14} (0.5 - y)$$

$$v = \frac{\pi}{3.14} (x - 0.5).$$

In Table 3 we display the error $E(628)$ for grid resolutions ranging from 100x100 to 400x400. These errors are compared to the errors reported by [23] when using the ELVIRA method for volume of fluid reconstruction. In Figure 4 we display the interface profile for the 100x100 computation.

Table 3: The error $E(628.0)$ for Zalesak’s test problem.

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>CLS</th>
<th>ELVIRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/16</td>
<td>0.00572</td>
<td>0.00567</td>
</tr>
<tr>
<td>1/32</td>
<td>0.00252</td>
<td>0.00262</td>
</tr>
<tr>
<td>1/64</td>
<td>0.00106</td>
<td>0.00121</td>
</tr>
</tbody>
</table>
6.2 Zero Gravity Drop Oscillation

We compute axisymmetric zero-gravity drop dynamics using the CLS method and compare our results to those using the level set method. According to the linearized results derived by Lamb (1932, §275) [19], the interfacial position of the drop is shown to be

\[ R(\theta, t) = a + \epsilon P_n(\cos(\theta)) \sin(\omega_n t), \]

where

\[ \omega_n^2 = \gamma \frac{n(n-1)(n+1)(n+2)}{a^3(\rho_1(n+1) + \rho_3 n)} \]

and \( P_n \) is the Legendre polynomial of order \( n \). \( \theta \) runs between 0 and 2\( \pi \) where \( \theta = 0 \) corresponds to \( r = 0 \) and \( z = a \). If viscosity is present, the amplitude (Lamb (1932, §355)) is proportional to \( e^{-t/\tau} \), where

\[ \tau = \frac{a^2 \rho_1}{\mu_1(2n+1)(n-1)}. \]

We compute the evolution of a drop with \( a = 1 \), \( g = 0 \), \( \mu_1 = 1/200 \), \( \mu_1/\mu_g = 100 \), \( \gamma = 1/2 \), \( \rho_1 = 1 \) and \( \rho_1/\rho_g = 100 \). The initial interface is given by \( R(\theta, \pi/2) \), with \( \epsilon = .05 \) and \( n = 2 \). With these parameters we find \( \omega_2 = 2.00 \) and \( \tau = 38.3 \). The fluid domain is \( \Omega = \{(r, z)|0 \leq r \leq 1.5 \text{ and } 0 \leq z \leq 1.5\} \) and we compute on grid sizes ranging from 32 x 32 to 128 x 128. Symmetric boundary conditions are imposed at \( r = 0 \) and \( z = 0 \). The interfacial thickness parameter \( \alpha \) is two grid cells. The results of our computations using the CLS method are compared to those of the level set method in Figure 5 where we display the perturbation in the major axis for the grid size 128 x 128. The period is 3.17 and the expected linearized period is \( \pi \). In table 4, we display the relative error between succeeding resolutions. The \( L^1 \) error is measured as

\[ \int_0^{\pi} |R_h(\pi/2, t) - R_{2h}(\pi/2, t)| dt, \]

and the \( L^\infty \) error is measured as

\[ \max_{0 \leq t \leq \pi} |R_h(\pi/2, t) - R_{2h}(\pi/2, t)|. \]

In order to compute the error numerically, we subdivide the time interval into \( n \) equally spaced intervals where \( n \) is the number of time steps at the fine grid resolution specified by \( h \). The values on the coarse grid are interpolated in time and then compared with that on the fine grid. The major amplitude \( R_h(\pi/2, t) \) is measured from \( \phi \).
Table 4: Convergence study using the CLS algorithm for zero gravity drop oscillations \( \gamma = 1/2, \mu_i = 1/200, \mu_i/\mu_g = 100, \rho_i/\rho_g = 100 \) and \( \alpha = 2 \).

<table>
<thead>
<tr>
<th>( \Delta r )</th>
<th>( L_1 )</th>
<th>( L_\infty )</th>
<th>period</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/64</td>
<td>N/A</td>
<td>N/A</td>
<td>3.18</td>
</tr>
<tr>
<td>3/128</td>
<td>0.00029</td>
<td>0.00066</td>
<td>3.17</td>
</tr>
<tr>
<td>3/256</td>
<td>0.00012</td>
<td>0.00032</td>
<td>3.17</td>
</tr>
</tbody>
</table>

6.3 Stationary bubble and spurious currents

We consider the simplified two-dimensional bubble problem when there is no gravity and the density ratio and viscosity ratio are both one. We assume that the bubble shape is a perfect circle; i.e. curvature of the bubble is constant. If the initial velocity is zero, then the velocity should stay zero for all time since the curvature of the bubble is constant and there are no driving forces. Unfortunately, the \textit{discretized} curvature of a circular bubble is not constant; so surface tension forces will cause the bubble to oscillate. We compare the maximum velocity (i.e. the error) for the CLS method to that of the level set method, and also to results found in work by Popinet and Zaleski [24]. We use dimensionless parameters as defined in [24]. The relevant dimensionless parameter for our problem is the Ohnesorge number \( Oh = \mu/\sqrt{\sigma D} \) where \( D \) is the diameter of the bubble. The width of the computational domain is \( 2D \). The resulting momentum equation in dimensionless form is

\[
\frac{DU}{Dt} = -\nabla p + \Delta U - \frac{1}{Oh^2} \kappa \nabla H.
\]

The smaller \( Oh \) is, the more stiff the problem will be. In order to compare with the data shown in [24], we used the exact same parameters as prescribed in their work. We set \( Oh^2 = 1/12000 \), we computed up until dimensionless time \( T^* \) equal to \( 250Oh^2 \), and we scaled the values for maximum velocity displayed in table 5 by \( Oh^2 \). In other words we display \( Oh^2 |U^*| \) which is the same as \( |U| \mu/\sigma \) where \( U^* \) is the dimensionless velocity, and \( U \) is the dimensional velocity. In table 5, we compare the maximum velocity for different grid resolutions and different methods. The results under the column “front track” were computations done by [24] using a front tracking method with a new “pressure gradient correction method.” The results under the column “PLIC/VOF” were computations done by J. Li[20] using the Piecewise Linear Interface Calculation (PLIC/VOF) code.

6.4 Drop impacting solid wall

Here we test the CLS method on the 2-d falling drop problem. The parameters are the same as that computed by Aleinov and Puckett [1] except that we compute with a density ratio of 816 : 1 as opposed to a density ratio of 10 : 1. Also, the viscosity ratio in our computations is 64 : 1 as opposed to constant viscosity. The
Table 5: Amplitude of spurious currents around a circular bubble. Maximum velocity $O h^2 |U^*|$ is displayed for front tracking method, Level set method, CLS method and VOF method. $\gamma = 12000$, $\mu_l = 1$, $\mu_l/\mu_g = 1$, $\rho_l/\rho_g = 1$ and $\alpha = 2$.

<table>
<thead>
<tr>
<th>grid size</th>
<th>front track</th>
<th>level set</th>
<th>CLS</th>
<th>PLIC/VOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>16$^2$</td>
<td>3.8E-5</td>
<td>3.8E-4</td>
<td>3.0E-4</td>
<td>1.5E-3</td>
</tr>
<tr>
<td>32$^2$</td>
<td>6.7E-6</td>
<td>2.2E-4</td>
<td>5.0E-4</td>
<td>3.3E-3</td>
</tr>
<tr>
<td>64$^2$</td>
<td>1.1E-6</td>
<td>1.0E-4</td>
<td>1.5E-4</td>
<td>3.3E-3</td>
</tr>
<tr>
<td>128$^2$</td>
<td>1.2E-7</td>
<td>3.4E-5</td>
<td>6.0E-4</td>
<td>4.1E-3</td>
</tr>
<tr>
<td>256$^2$</td>
<td>1.7E-8</td>
<td>1.0E-5</td>
<td>1.7E-4</td>
<td>4.1E-3</td>
</tr>
</tbody>
</table>

radius of the falling drop is 0.1cm. Other relevant parameters are $g = 980cm/s^2$, $\sigma = 73.2g/s^2$, $\mu_g = 1.78E - 4g/(cm s)$, $\mu_l = 1.137E - 2g/(cm s)$, $\rho_g = 0.001225g/cm^3$, and $\rho_l = 1.0g/cm^3$. The dimensions of our domain is 0.25cm by 1.0cm. In Figure 6 we show the evolution of the drop as it hits the bottom of our domain (mesh size $32 \times 128$). In figure 7, we show the same problem except with surface tension effects turned off. Without surface tension, the spray from the drop creeps up the walls.

In table 6, we show the relative errors in the position of the interface and the velocity ($t = 0.025$) for grid resolutions ranging from $16 \times 64$ to $64 \times 256$. The error for the position of the interface is measured similarly as in (59) except that we measure the relative error between succeeding grid resolutions since we do not know the exact solution for this problem:

$$E(t) = \sum_{i,j} \int_{\Omega_{ij}} |H(\phi_f(t)) - H(\phi_c(t))|d\mathbf{x}. \quad (63)$$

Here, $\phi_c$ is the level set function from a coarser computation and $\phi_f$ is the level set function from the refined computation. The relative error for the velocity is measured by the following equations,

$$E_{u,L1}(t) = \sum_{i,j} \sqrt{(u_{f,ij} - u_{c,ij})^2 + (v_{f,ij} - v_{c,ij})^2} \Delta x \Delta y \quad (64)$$

$$E_{u,max}(t) = \max_{i,j} \sqrt{(u_{f,ij} - u_{c,ij})^2 + (v_{f,ij} - v_{c,ij})^2}. \quad (65)$$

In table 6, we also display the position of the center of mass at $t = 0.025$.

6.5 Capillary instability

A motivation for our development of the CLS method stems from the modeling of micro-scale ink-jet devices [2, 16]. Typically, when a jet of ink is ejected from an ink jet device, a long “tail” of liquid forms in which capillary instability causes the long “tail” to break up into droplets. The ability to model surface tension is important in order to accurately model the break up of the “tail” into droplets. In this section, we
Table 6: Convergence study at $t = 0.025$ for falling 2d water drop in air. Radius of drop is $0.1\text{cm}$. The density ratio is $816:1$ and the viscosity ratio is $64:1$. Results computed using the CLS method $\alpha = 2$.

<table>
<thead>
<tr>
<th>$\Delta x$</th>
<th>$E(0.025)$ center of mass</th>
<th>$E_{u,L1}(0.025)$</th>
<th>$E_{u,\text{max}}(0.025)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1cm/64</td>
<td>N/A</td>
<td>0.4372</td>
<td>N/A</td>
</tr>
<tr>
<td>1cm/128</td>
<td>0.00202</td>
<td>0.4382</td>
<td>5.34</td>
</tr>
<tr>
<td>1cm/256</td>
<td>0.00058</td>
<td>0.4379</td>
<td>2.76</td>
</tr>
</tbody>
</table>

We consider an initially perturbed cylindrical column of water in air. The shape of the initial interface is,

$$r(z) = r_0 + \epsilon \cos(2\pi z/\lambda).$$  \hspace{1cm} (66)

We compute on an axisymmetric domain $\Omega = \{(r, z)|0 \leq r \leq \lambda/4 \text{ and } 0 \leq z \leq \lambda/2\}$. Symmetric boundary conditions are enforced at $r = 0$, $z = 0$ and $z = \lambda/2$. Outflow boundary conditions are enforced at $r = \lambda/4$. The relevant parameters for our test problem are $r_0 = 6.52$ microns, $\epsilon = 1.3$ microns, $\lambda = 60$ microns, $\mu_l = 1.138 \times 10^{-2}g/(\text{cms})$, $\mu_g = 1.77 \times 10^{-4}g/(\text{cms})$, $\rho_l = 1.0g/cm^3$, $\rho_g = 0.001225g/cm^3$, and $\gamma = 72.8\text{dynes/cm}$. In our computations we use dimensionless parameters where the Reynolds number $R = \rho L U / \mu_l$ is 7.5, the Weber number $W = \rho L U^2 / \gamma$ is 1, $L = 1$ micron, $U = 8.53m/s$ and the density and viscosity ratios are 816 and 64 respectively.

In Figure 8, we display the results of our computations using the CLS method for the capillary jet as it breaks up. As a comparison, we have also displayed computations using the level set approach (see Figure 9). In Tables 7 and 8, we measure the relative errors for the interface and velocity field for grid resolutions ranging from 32x64 to 128x256. The relative error for the interface is measured by (63). The relative error for the velocity is measured by the following equations,

$$E_{u,L1}(t) = \sum_{i,j} \sqrt{(u_{f,ij} - u_{c,ij})^2 + (v_{f,ij} - v_{c,ij})^2} r_i \Delta r \Delta z$$  \hspace{1cm} (67)

$$E_{u,\text{max}}(t) = \max_{i,j} \sqrt{(u_{f,ij} - u_{c,ij})^2 + (v_{f,ij} - v_{c,ij})^2}. \hspace{1cm} (68)$$

As shown by the tables, we obtain about first order accuracy before and after pinch off. We suspect that we have only first order accuracy for this problem because the density and viscosity jump across the interface. We recomputed the capillary jet problem with constant density $\rho_l = \rho_g = 1.0g/cm^3$ and constant viscosity $\mu_l = \mu_g = 1.138 \times 10^{-2}g/(\text{cms})$ and the relative errors for the recomputed problem are shown in Tables 9 and 10. The interface profiles for the constant density case are shown in Figure 10.
Table 7: Convergence study using the CLS algorithm for the capillary instability problem. All data in terms of dimensionless parameters. \( t = 80.0, \mu_l/\mu_g = 64, \rho_l/\rho_g = 816. \)

<table>
<thead>
<tr>
<th>grid</th>
<th>( E(80) )</th>
<th>( E_{u,L1}(80) )</th>
<th>( E_{u,max}(80) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16x32</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>32x64</td>
<td>36.88</td>
<td>268.7</td>
<td>0.064</td>
</tr>
<tr>
<td>64x128</td>
<td>21.13</td>
<td>167.5</td>
<td>0.050</td>
</tr>
<tr>
<td>128x256</td>
<td>8.14</td>
<td>94.2</td>
<td>0.034</td>
</tr>
</tbody>
</table>

Table 8: Convergence study using the CLS algorithm for the capillary instability problem. All data in terms of dimensionless parameters. \( t = 120.0, \mu_l/\mu_g = 64, \rho_l/\rho_g = 816. \)

<table>
<thead>
<tr>
<th>grid</th>
<th>( E(120) )</th>
<th>( E_{u,L1}(120) )</th>
<th>( E_{u,max}(120) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16x32</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>32x64</td>
<td>318.9</td>
<td>936.2</td>
<td>0.90</td>
</tr>
<tr>
<td>64x128</td>
<td>182.9</td>
<td>599.3</td>
<td>1.32</td>
</tr>
<tr>
<td>128x256</td>
<td>76.8</td>
<td>150.6</td>
<td>0.53</td>
</tr>
</tbody>
</table>

Table 9: Convergence study using the CLS algorithm for the capillary instability problem. All data in terms of dimensionless parameters. \( t = 120.0, \mu_l/\mu_g = 1, \rho_l/\rho_g = 1. \)

<table>
<thead>
<tr>
<th>grid</th>
<th>( E(120) )</th>
<th>( E_{u,L1}(120) )</th>
<th>( E_{u,max}(120) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16x32</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>32x64</td>
<td>52.76</td>
<td>68.55</td>
<td>0.1535</td>
</tr>
<tr>
<td>64x128</td>
<td>11.23</td>
<td>16.17</td>
<td>0.0339</td>
</tr>
<tr>
<td>128x256</td>
<td>2.82</td>
<td>4.24</td>
<td>0.0091</td>
</tr>
</tbody>
</table>

Table 10: Convergence study using the CLS algorithm for the capillary instability problem. All data in terms of dimensionless parameters. \( t = 160.0, \mu_l/\mu_g = 1, \rho_l/\rho_g = 1. \)

<table>
<thead>
<tr>
<th>grid</th>
<th>( E(160) )</th>
<th>( E_{u,L1}(160) )</th>
<th>( E_{u,max}(160) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16x32</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>32x64</td>
<td>78.3</td>
<td>125.3</td>
<td>0.143</td>
</tr>
<tr>
<td>64x128</td>
<td>15.3</td>
<td>47.6</td>
<td>0.177</td>
</tr>
<tr>
<td>128x256</td>
<td>3.0</td>
<td>12.9</td>
<td>0.090</td>
</tr>
</tbody>
</table>

23
Table 11: Convergence study (CLS) for axisymmetric inviscid rising air bubble in water. The density ratio is 1:816. \(Fr = 1\), \(We = 200\).

<table>
<thead>
<tr>
<th>(\Delta x)</th>
<th>(E(1.3))</th>
<th>(E_{u,L1}(1.3))</th>
<th>(E_{u,max}(1.3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>32x64</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>64x128</td>
<td>0.167</td>
<td>3.39</td>
<td>2.09</td>
</tr>
<tr>
<td>128x256</td>
<td>0.049</td>
<td>2.06</td>
<td>2.34</td>
</tr>
<tr>
<td>256x512</td>
<td>0.016</td>
<td>1.19</td>
<td>3.76</td>
</tr>
</tbody>
</table>

6.6 Inviscid gas bubble

We consider the rise of an inviscid axisymmetric gas bubble in liquid. Here, the dimensionless parameters are \(Fr = 1\) and \(We = 200\). The density ratio is 1:816. We compute on an axisymmetric domain \(\Omega = \{(r, z)|0 \leq r \leq 3\) and \(0 \leq z \leq 6\}\). The interfacial thickness spreading parameter \(\alpha\) is 3 grid cells. In figure 11, we display results computed using the CLS method (thin lines) and compare them with the boundary integral method [35] (thick lines). As a note, in the boundary integral method, the density in the gas is assumed zero and the domain of computation is assumed to have infinite extent whereas in the CLS method, the density ratio is 1:816 and we use far-field boundary condition; i.e. the pressure on the walls is \(p = z/\text{Fr}\). The grid resolution for the CLS results is 128 \(\times 256\) whereas 240 points were used to discretize the interface for the boundary integral method.

In table 11 we display the relative errors for the inviscid rising bubble problem when the grid is successively refined. The error is computed at \(t = 1.3\), when the jet is about ready to break through the top of the bubble.

In figure 12, we overlay results using the level set method (thick line) with those of the CLS method (thin lines) at \(t = 1.3\). Here, the results are almost identical.

We have also compared CPU costs of the CLS method and the level set method for this problem. It takes about 3% of the total CPU time for level set advection versus 10% of the total CPU time for operator split CLS advection. The redistance procedure for the level set method took 9% of the total CPU time whereas the redistance procedure for the CLS method took 2.5% of the total CPU time.

6.7 Steady rising gas bubble

We compute the steady rise of an axisymmetric gas bubble rising in a viscous liquid. For this problem, the density ratio is 714:1 and the viscosity ratio is 6667:1. The relevant (dimensionless) parameters for this problem are \(Fr = 0.78\), \(Re = 9.7\), \(\mu_l/\mu_g = 6667\), \(\rho_l/\rho_g = 714\), and \(We = 7.6\). These parameters correspond to those used in bubble experiments by Hnat and Buckmaster [18] and used in steady bubble computations by Ryskin and Leal [29, 30]. Our computational domain is \(5 \times 20\) and we use far-field boundary condition (pressure on the walls is \(p = z/\text{Fr}\)). We discretize
our domain with $64 \times 256$ grid cells. The interfacial thickness parameter $\alpha$ is two grid cells. In figure 13, we show the results using the CLS method. In figure 14, we plot the position of the center of mass versus time. Also plotted is the best fit line for $2 < t < 10$. The expected slope as seen experimentally by Hnat and Buckmaster [18] is 1 and the computed slope here is 1.003. The maximum volume fluctuation of the gas bubble using the CLS method is 0.0009 %. As a remark, the maximum volume fluctuation using the level set method with comparable resolution is about 20% (see [32]).

6.8 Gas bubble bursting at a free surface

Here, we compute the problem of a gas bubble rising to the free surface of a liquid. When the bubble breaks through the surface, large surface tension forces are produced which ultimately cause a jet of liquid to be ejected. This jet of liquid can subsequently break up into drops.

This problem was studied by Boulton-Stone & Blake (1993) [9] using the boundary integral method and by Sussman & Smereka (1997) [35] using the level set method. In our CLS calculation, we consider a spherical 4mm air bubble released just below the surface. The domain (axisymmetric geometry) is $\Omega = \{(x, y)|0 \leq r \leq 3 \text{ and } 0 \leq z \leq 12\}$ and the mesh is $48 \times 192$. For these computations the density ratio is 1000:1 and the viscosity ratio is 100:1. The relevant (dimensionless) parameters for this problem are $Re = 474$, $\mu_i/\mu_g = 100$, $\rho_i/\rho_g = 1000$, $Fr = 0.64$ and $We = 1.0$. The interfacial thickness parameter $\alpha$ represents two grid cells. In figure 15 we show the evolution of the air bubble rising to the surface of the air/water interface and then bursting due to stiff surface tension effects. The jet then breaks up due to capillary instabilities, emitting satellite drops. The maximum volume fluctuation of the water for this problem is 0.007 % even through the complex changes in topology.

7 Three dimensional results

In Figure 16, we show the computation of the rise of a fully three dimensional inviscid air bubble in water. The density ratio is 816:1 and the dimensionless parameters used for this problem are $We = 200$ and $Fr = 1$. We use adaptive mesh refinement when computing this problem [32]. The dimensions of the domain are $4x4x8$ and the mesh size on the finest level of adaptivity is $\Delta x = \Delta y = \Delta z = 1/16$. We use far-field boundary conditions on all sides of the domain. In Figure 17, we display a cross-section of the bubble at $t = 1.24$ and $t = 1.48$ and compare these results with the results computed using the axisymmetric CLS algorithm with the same fine grid resolution $\Delta r = 1/16$. We remark, that the maximum mass fluctuation of the gas bubble for the 3d computation was 0.01%.

In Figure 18, we display the interaction of two viscous gas bubbles in liquid. For this problem, we start off with two gas bubbles whose centers are offset in the “x” direction by one bubble radii and offset in the “z” direction by 2.3 radii. The density
ratio is 20:1 and the viscosity ratio is 26:1. The dimensionless parameters we use for this problem are $We = 50$, $Fr = 1$ and $Re = 50^{3/4}$. These parameters correspond to Figure 12 in [38]. We use adaptive mesh refinement when computing this problem [32]. The dimensions of the domain are 4x4x8 and the mesh size on the finest level of adaptivity is $\Delta x = \Delta y = \Delta z = 1/16$. We use free-slip boundary conditions on all sides of the domain. Our results agree qualitatively with those in [38]. We attribute the difference, in part, due to the fact that our bubbles were initially offset by different values than were the bubbles in [38].

8 Conclusions

A coupled level set volume of fluid method (CLS) has been presented for representing the free surface of two-phase flows. We have tested our method on two-dimensional, axisymmetric, and fully three dimensional flows.

We find that for many problems, the CLS method gives comparable results as the level set method. For example, for the Rayleigh Capillary problem (see section 6.5) the CLS method and level set method give identical results. For certain problems, where the interface develops corners, or there is interfacial merging and pinching, the CLS method outperforms the level set method since the CLS method will conserve mass to a fraction of a percent while the level set method can lose as much as 20\% of mass. On the other hand, the level set method outperforms the CLS method for certain surface tension driven flows; specifically the stationary bubble problem where as the grid is refined, spurious currents go away with the level set method but they do not go away with the CLS method.

We also find that for many problems, the CLS method gives comparable results as standard VOF/PLIC type methods. For some problems, especially problems with surface tension, the CLS method outperforms standard VOF methods. The slope and curvature calculation is easier with the CLS method since these quantities are obtained directly from the level set function. The maximum magnitude of the velocity for the stationary bubble problem is almost an order of magnitude smaller for the CLS method than for a VOF/PLIC method. One disadvantage of a CLS method versus a VOF method is in implementation; one has to develop code to find the exact distance from a piecewise linear interface. Also, when surface tension is not present, there may not be any need for a standard VOF/PLIC method to mollify the interface for slope or curvature reconstruction, this would give the VOF/PLIC method a slight speed advantage over the CLS method (see section 6.6).

In the introduction, we commented that one motivation for developing the CLS method was for modeling ink-jet devices. We see from our results section that our motivations were satisfied in the sense that the CLS method conserves mass, and also computes most surface tension driven flows accurately. A case in point, is the Rayleigh Capillary computations done in section 6.5. Identical parameters as found in a typical jetting problem were used and we observed convergence before and after pinch off when using the CLS method.
9 Appendix

9.1 Approximation of the Advection Term

In this section, we describe the discretization of the advection term,

\[ [(U \cdot \nabla)U]^{n+1/2}. \]  

(69)

In the process of describing the discretization of (69) we shall also describe how the discretely divergence free advective edge velocities, \( u^{ADV,n+1/2}_{i+1/2,j} \) and \( v^{ADV,n+1/2}_{i,j+1/2} \), are formed.

The discretization of (69) in this algorithm is very similar to the discretization used by [25, 3]. It is a predictor-corrector method based on the unsplit Godunov method introduced by Colella [14].

In the predictor we extrapolate the velocity \( U \) to the cell faces at \( t^{n+1/2} \) using a second-order Taylor series expansion in space and time. The time derivative \( U_t \) is replaced using (6). For face \((i+1/2, j)\) this gives

\[
U^{n+1/2}_{i+1/2,j} = U^n_{ij} + \left( \frac{\Delta t}{2} - \frac{u^n_{ij} \Delta t}{2} \right) U^n_{x,ij} - \frac{\Delta t}{2} (v U)_y^{n}_{ij} - \frac{\Delta t}{2} (w U)_z^{n}_{ij} + \\
\frac{\Delta t}{2} \left( - \frac{G_{ij}^{n-1/2}}{\rho^n_{ij}} + \frac{C^n_{ij}}{\rho^n_{ij}} - \frac{M^n_{ij}}{\rho^n_{ij}} + F \right)
\]

(70)
extrapolated from cell \((i, j)\), and

\[
U^{n+1/2,R}_{i+1/2,j} = U^n_{i+1,j} - \left( \frac{\Delta t}{2} + \frac{u^n_{i+1,j} \Delta t}{2} \right) U^n_{x,ij} - \frac{\Delta t}{2} (v U)_y^{n}_{i+1,j} - \frac{\Delta t}{2} (w U)_z^{n}_{i+1,j} + \\
\frac{\Delta t}{2} \left( - \frac{G_{i+1,j}^{n-1/2}}{\rho^n_{i+1,j}} + \frac{C^n_{i+1,j}}{\rho^n_{i+1,j}} - \frac{M^n_{i+1,j}}{\rho^n_{i+1,j}} + F \right)
\]

(71)
extrapolated from cell \((i+1, j)\).

Analogous formulae are used to predict values at each of the other faces of the cell

\[ U^{n+1/2,T/B}_{i,j+1/2}. \]  

(72)
The first derivative normal to the face, \( U^n_z \) for the example in (70) and (71), is evaluated using a monotonicity-limited fourth-order slope approximation [13]. The limiting is done on each component of the velocity at \( t^n \) individually.

The transverse derivative term,

\[ \langle U_y \rangle, \]

is evaluated by first extrapolating \( U \) to the transverse faces from the cell centers on either side, using normal derivatives only, and then choosing between these states using the upwinding procedure as described in detail by Almgren et al. [3] and Puckett et al. [25].

27
Once we have computed $u_{i+1/2,j}^{n+1/2,L/R}$ and $v_{i,j+1/2}^{n+1/2,T/B}$, we are in a position to construct the normal face-centered edge velocities at $t^{n+1/2}$:

$$u_{i+1/2,j}^{ADV}, v_{i,j+1/2}^{ADV}$$

Given $u_{i+1/2,j}^{n+1/2,L}$ and $u_{i+1/2,j}^{n+1/2,R}$, we use an upwinding procedure to choose $u_{i+1/2,j}^{n+1/2}$:

$$u_{i+1/2,j}^{n+1/2} = \begin{cases} 
  u^L & \text{if } u^L > 0 \text{ and } u^L + u^R > 0 \\
  0 & \text{if } u^L \leq 0 \text{ or } u^R \geq 0 \text{ or } u^L + u^R = 0 \\
  u^R & \text{if } u^R < 0 \text{ and } u^L + u^R < 0 
\end{cases} \quad (73)$$

Here, we suppress the $i + 1/2, j$ spatial indices on left and right states and we also suppress the $n + 1/2$ temporal index.

We follow a similar procedure as in (73) to construct $v_{i,j+1/2}^{n+1/2}$.

These normal velocities on cell faces at $t^{n+1/2}$,

$$u_{i+1/2,j}^{n+1/2}, v_{i,j+1/2}^{n+1/2} \quad (74)$$

are second-order accurate but do not, in general, satisfy the discrete divergence-free condition. In order to make these velocities divergence-free, we apply the MAC projection [6]. The equation

$$D^{MAC}(\frac{1}{\rho^n}G^{MAC}p^{MAC}) = D^{MAC}(U^{n+\frac{1}{2}}) \quad (75)$$

is solved for $p^{MAC}$, where

$$D^{MAC}U^{n+\frac{1}{2}} = \frac{r_{i+1/2}^{n+\frac{1}{2}}u_{i+1/2,j}^{n+\frac{1}{2}} - r_{i-1/2}^{n+\frac{1}{2}}u_{i-1/2,j}^{n+\frac{1}{2}}}{r_i \Delta r} + \frac{v_{i,j+1/2}^{n+\frac{1}{2}} - v_{i,j-1/2}^{n+\frac{1}{2}}}{\Delta z}$$

and $G^{MAC} = -(D^{MAC})^T$ so that

$$(G_x^{MAC}p^{MAC})_{i+\frac{1}{2},j}^{n+\frac{1}{2}} = \frac{p_{i+1,j}^{MAC} - p_{i,j}^{MAC}}{\Delta r}$$

with $G_y^{MAC}$ defined analogously. The resulting linear system (75) is solved using a multigrid preconditioned conjugate gradient solver [37].

The face-based advection velocities at $t^{n+1/2}$ are then defined by

$$u_{i+1/2,j}^{ADV} = u_{i+1/2,j}^{n+\frac{1}{2}} - \frac{1}{\rho_{i+1/2,j}^{n}}(G_x^{MAC}p^{MAC})_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \quad (76)$$

with $v_{i,j+\frac{1}{2}}^{ADV}$ defined analogously. The quantity $\rho_{i+1/2,j}^{n}$ in (76) is defined by

$$\rho_{i+1/2,j}^{n} = \frac{1}{2}(\rho_{i,j}^{n} + \rho_{i+1,j}^{n})$$

28
with $\rho_{i,j+1/2}^n$ defined analogously.

The next step, after constructing the advective velocities

$$u_{i+1/2,j}^{ADV}, v_{i+1/2,j}^{ADV},$$

is to choose the appropriate state $U_{i+1/2,j}^{n+1/2}$ given the left and right states in (70) thru (71):

$$U_{i+1/2,j}^{n+1/2} = \begin{cases} U^L \
\frac{1}{2}(U^L + U^R) \
U^R \end{cases} \text{ if } u^{ADV} > 0 \text{ if } u^{ADV} = 0 \text{ if } u^{ADV} < 0$$

We have

$$U_{i+1/2,j}^{n+1/2} = \begin{cases} U^L \
\frac{1}{2}(U^L + U^R) \
U^R \end{cases} \text{ if } u^{ADV} > 0 \text{ if } u^{ADV} = 0 \text{ if } u^{ADV} < 0$$

(77)

Here, we suppress the $i + 1/2, j$ spatial indices on left and right states and we also suppress the $n + 1/2$ temporal index.

We follow a similar procedure as in (77) to construct

$$U_{i,j+1/2}^{n+1/2}.$$

The advection term can now be defined by

$$[(U \cdot \nabla)U]_{i,j}^{n+1/2} = \frac{1}{r_i \Delta r} \frac{r_{i+1/2}u_{i+1/2,j}^{ADV} + r_{i-1/2}u_{i-1/2,j}^{ADV}}{2} (U_{i+1/2,j} - U_{i-1/2,j}) + \frac{1}{\Delta z} \frac{v_{i,j+1/2}^{ADV} + v_{i,j-1/2}^{ADV}}{2} (U_{i,j+1/2} - U_{i,j-1/2})$$

(78)

9.2 Discretization of Pressure Gradient, Viscous and Surface Tension terms

In this section we describe the finite difference approximation to the pressure gradient, $Gp$, viscous term, $\mathcal{L}$, and surface tension term, $\mathcal{M}$.

The discrete pressure gradient is defined by

$$(Gp)_{ij} \equiv \begin{pmatrix} \frac{p_{i+1/2,j+1/2} + p_{i+1/2,j-1/2} - p_{i-1/2,j+1/2} - p_{i-1/2,j-1/2}}{2r_i \Delta r} \
\frac{p_{i+1/2,j+1/2} - p_{i+1/2,j-1/2} + p_{i-1/2,j+1/2} - p_{i-1/2,j-1/2}}{2\Delta z} \end{pmatrix}$$

(79)

where $G$ here denotes a discrete gradient operator defined at cell centers but operating on nodal data.

The first component of the viscous term $\nabla \cdot 2\mu(\phi) \mathcal{D}$ is discretized as

$$\left(\mathcal{L}\right)_{ij} = \begin{pmatrix} \frac{2\mu_{i+1/2,j+1/2}(u_{i+1,j+1} - u_{i,j}) - 2\mu_{i-1/2,j+1/2}(u_{i,j+1} - u_{i,j+1})}{r_i \Delta r} \
\frac{\mu_{i,j+1/2}(u_{i,j+1} - u_{i,j}) - \mu_{i,j-1/2}(u_{i,j} - u_{i,j-1})}{\Delta z} + \frac{\mu_{i+1,j+1/2}(u_{i+1,j+1} - u_{i,j}) - \mu_{i+1,j-1/2}(u_{i+1,j} - u_{i+1,j-1})}{\Delta r \Delta z} - \mu_{i,j} u_{i,j} / r_i^2 \end{pmatrix}$$

(90)
where
\[
\mu_{i+\frac{1}{2},j} = \frac{1}{2}(\mu(\phi_{i,j}) + \mu(\phi_{i+1,j})) \quad \text{and} \quad \mu_{i,j+\frac{1}{2}} = \frac{1}{2}(\mu(\phi_{i,j}) + \mu(\phi_{i,j+1})).
\]  
(80)

Remarks:

- the discretization described by (80) is a basic discretization; improved discretizations for variable viscosity problems are described by Coward et al. [15].
- The second component of the viscous term, \((L)^2_{ij}\), is discretized in a similar manner as the first component.

The surface tension term \(\gamma \kappa(\phi) \nabla H(\phi)\) is discretized as
\[
(M)_{ij} \equiv \gamma(DN)_{ij}(GH^\text{node})_{ij}.
\]  
(81)

\(N_{i+1/2,j+1/2}\) is the discrete approximation of the level set normal \(\nabla \phi/|\nabla \phi|\),
\[
N_{i+1/2,j+1/2} \equiv \frac{(G\phi)_{i+1/2,j+1/2}}{|(G\phi)_{i+1/2,j+1/2}|},
\]  
(82)

where
\[
(G\phi)_{i+1/2,j+1/2} \equiv \left(\frac{\phi_{i+1,j+1}+\phi_{i+1,j-1+\phi_{i+1,j}+\phi_{i,j}}}{2\Delta r} \right).
\]  
(83)

Here we use \(G\) to refer to the discrete gradient operator defined on nodes but operating on cell-centered data.

We define the cell-based discrete divergence operator \(D\) by
\[
(DN)_{ij} \equiv \frac{r_{i+1/2}(n_{i+1/2,j-1/2}^1 + n_{i+1/2,j+1/2}^1 - n_{i-1/2,j-1/2}^1 + n_{i-1/2,j+1/2}^1)}{r_i \Delta r} + \frac{r_{i+1/2}(n_{i+1/2,j+1/2}^2 - n_{i+1/2,j-1/2}^2 - n_{i-1/2,j+1/2}^2 + n_{i-1/2,j-1/2}^2)}{r_i \Delta z}.
\]  
(84)

The node-based Heaviside function \(H^\text{node}_{i+1/2,j+1/2}\) is defined as
\[
H^\text{node}_{i+1/2,j+1/2} = H(\phi_{i+1,j} + \phi_{i,j} + \phi_{i+1,j+1} + \phi_{i,j+1}).
\]  
(85)

9.3 Discretization of the Projection

In this section we describe the discrete “approximate projection” \(P\), which is used in (57). \(P\) is an approximation to the projection operator \(P\) described in (19). We remark that a detailed description of the approximate projection is given by [4].

Given the discrete vector field
\[
\frac{U^* - U^n}{\Delta t},
\]  
(86)
we decompose (86) into an \textit{approximately} divergence free part
\[
\frac{U^{n+1} - U^n}{\Delta t}
\]
and the discrete gradient of a scalar \( q \) divided by density
\[
\frac{(Gq)_{ij}}{\rho_{ij}^{n+1/2}},
\]
where the discrete gradient \( G \) in (88) is defined in (79).

The approximate projection is computed by solving
\[
L_\rho q = D \left( \frac{U^* - U^n}{\Delta t} \right)
\]
for \( q \). The right hand side of (89) is an approximation to \( \nabla \cdot \mathbf{V} \) found in the right hand side of (17). The discrete divergence \( DU \) is
\[
(DU)_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{r_{i+1}(u_{i+1,j} + u_{i+1,j+1}) - r_i(u_{i,j} + u_{i,j+1})}{r_{i+1/2}\Delta r} + \frac{r_{i+1}(v_{i+1,j+1} - v_{i+1,j}) - r_i(v_{i,j} - v_{i,j+1})}{r_{i+1/2}\Delta z}.
\]
The left hand side of (89), \( L_\rho q \), is an approximation to \( \nabla \cdot \frac{1}{\rho} \nabla p \) found in the left hand side of (17). The discrete representation of \( L_\rho q \) is
\[
(L_\rho q)_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{6r_{i+1/2}\Delta t} \left( \frac{r_i}{\rho_{ij}} (2q_{i-\frac{1}{2},j-\frac{1}{2}} + q_{i+\frac{1}{2},j-\frac{1}{2}} + q_{i-\frac{1}{2},j+\frac{1}{2}} - 4q_{i+\frac{1}{2},j+\frac{1}{2}}) + \frac{r_i}{\rho_{i+1,j}} (2q_{i+1/2,j-\frac{1}{2}} + q_{i+\frac{1}{2},j-\frac{1}{2}} + q_{i+1/2,j+\frac{1}{2}} - 4q_{i+\frac{1}{2},j+\frac{1}{2}}) + \frac{r_i}{\rho_{i+1,j+1}} (2q_{i+1/2,j+1} + q_{i+\frac{1}{2},j+1} + q_{i+1/2,j-\frac{1}{2}} + q_{i+3/2,j+\frac{1}{2}} - 4q_{i+1/2,j+\frac{1}{2}}) \right).
\]
The operator \( L_\rho q \) (91) is derived from the variational form of (17),
\[
\int \frac{1}{\rho} \nabla q(x) \cdot \nabla \psi(x) \, dx = \int \frac{U^* - U^n}{\Delta t} \cdot \nabla \psi(x) \, dx, \quad \forall \psi(x),
\]
where \( dx \) is the volume element \( r \, dr \, d\theta \). The finite element basis functions \( \psi(x) \) represent standard piecewise bilinear functions.

After (89) is solved, we form \( (U^{n+1} - U^n)/\Delta t, \)
\[
\frac{U^{n+1} - U^n}{\Delta t} = \frac{U^* - U^n}{\Delta t} - \frac{Gq}{\rho^{n+1/2}},
\]
and \( p^{n+1/2}, \)
\[
p^{n+1/2} = p^{n-1/2} + q.
\]

Remarks:
• the discrete projection step presented here is slightly different from the continuous analogue presented in section 2.1 because we are solving for the difference in pressure \( q = p^{n+1/2} - p^{n-1/2} \) instead of the actual pressure \( p^{n+1/2} \).

• The discrete projection operator \( \mathcal{P} \) is called an approximate projection because the discrete divergence of (87),

\[
\left[ D \left( \frac{U^{n+1} - U^n}{\Delta t} \right) \right]_{i+1/2,j+1/2}
\]

is not identically zero. In order to see why (94) is not necessarily zero, we apply the discrete divergence \( D \) to both sides of (93) in order to arrive at

\[
\left[ D \left( \frac{U^{n+1} - U^n}{\Delta t} \right) \right]_{i+1/2,j+1/2} = \left[ D \left( \frac{U^* - U^n}{\Delta t} \right) \right]_{i+1/2,j+1/2} - \left[ D \frac{1}{\rho^{n+1/2}} Gq \right]_{i+1/2,j+1/2}
\]

The discrete operator \( D \frac{1}{\rho^{n+1/2}} Gq \) is not the same as \( L_\rho q \) which means (94) is not necessarily zero.
References


List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Diagram of where the discretely divergence-free edge velocity field $U^{ADV}$, level set function $\phi$ and volume of fluid function $F$ are located in relation to the computational grid and the physical boundary.</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>Diagram of where the discrete variables $U$, $p$, $\phi$ and $F$, are located in relation to the computational grid and the physical boundary.</td>
<td>39</td>
</tr>
<tr>
<td>3</td>
<td>The distances $d$ assigned to points $A$, $F$, and $E$ respectively are $</td>
<td>AB</td>
</tr>
<tr>
<td>4</td>
<td>Comparison with expected answer for Zalesak’s problem after one full revolution of notched disk. Grid resolution is 100x100.</td>
<td>41</td>
</tr>
<tr>
<td>5</td>
<td>Perturbation in major amplitude for zero gravity drop oscillations. $\mu_l = 1/200$, $\gamma = 1/2$, density ratio 100:1, viscosity ratio 100:1. Comparison of coupled level set method (CLS) with the level set method.</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>Falling 1mm 2d water drop in air; density ratio 816:1, viscosity ratio 64:1, number of grid cells 32 $\times$ 128.</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>Falling 1mm 2d water drop in air; surface tension turned off. Density ratio 816:1, viscosity ratio 64:1, number of grid cells 32 $\times$ 128.</td>
<td>44</td>
</tr>
<tr>
<td>8</td>
<td>Capillary Instability. $\rho_w/\rho_a = 816$, $\mu_w/\mu_a = 64$. Grid resolution is 64x128.</td>
<td>45</td>
</tr>
<tr>
<td>9</td>
<td>Capillary Instability; Level set method. $\rho_w/\rho_a = 816$, $\mu_w/\mu_a = 64$. Grid resolution is 64x128.</td>
<td>46</td>
</tr>
<tr>
<td>10</td>
<td>Capillary Instability. $\rho_w/\rho_a = 1$, $\mu_w/\mu_a = 1$. Grid resolution is 64x128.</td>
<td>47</td>
</tr>
<tr>
<td>11</td>
<td>Spherical gas bubble in liquid: density ratio 816:1, We=200. Results computed using the volume of fluid method (thin contour) are compared to results computed using the boundary integral method (thick contour).</td>
<td>48</td>
</tr>
<tr>
<td>12</td>
<td>Spherical gas bubble in liquid: density ratio 816:1, We=200. Results computed using the volume of fluid method (thin contour) are compared to results computed using the level set method (thick contour).</td>
<td>49</td>
</tr>
<tr>
<td>13</td>
<td>Rise of an initially spherical gas bubble in viscous liquid. Results computed using the CLS method. density ratio 714:1, viscosity ratio 6667:1, $Re = 9.7$, $We = 7.6$, $Fr = 0.78$.</td>
<td>50</td>
</tr>
<tr>
<td>14</td>
<td>Comparison of position of center of mass of rising viscous gas bubble in liquid to “best fit” straight line of the position. Expected slope as predicted by experiments is 1. Results computed using the CLS method. density ratio 714:1, viscosity ratio 6667:1, $Re = 9.7$, $We = 7.6$, $Fr = 0.78$.</td>
<td>51</td>
</tr>
<tr>
<td>15</td>
<td>Evolution of 4mm air bubble rising to an air/water interface. $\mu/\mu_g = 100$, $\rho/\rho_g = 1000$, $Fr = 0.64$, $Re = 474$, and $We = 1.0$. Results computed using CLS method.</td>
<td>52</td>
</tr>
</tbody>
</table>
16  Rise of inviscid air bubble in water. $We = 200$, effective fine grid 64x64x128. .................................................. 53
17  Spherical gas bubble in liquid; density ratio 8:1; $We=200$. Left: Cross section of three-dimensional results ($y=2$, x-z plane), effective fine grid 64x64x128, dimensions of domain: 4x4x8. Right: Axisymmetric results, effective fine grid 32x128, dimensions of domain 2x8. .................................................. 54
18  Non-Axisymmetric merging of two viscous gas bubbles, effective fine grid 64x64x128. .................................................. 55
Physical boundary

Gas or Liquid

$U^{ADV}_{i-3/2,j}, \phi, F$
$U^{ADV}_{i-1/2,j}, \phi, F$
$U^{ADV}_{i+1/2,j}, \phi, F$

$\psi^{ADV}_{i-1,j-1/2}$
$\psi^{ADV}_{i,j-1/2}$
$\psi^{ADV}_{i+1,j-1/2}$

$U^{ADV}_{i-3/2,j+1}, \phi, F$
$U^{ADV}_{i-1/2,j+1}, \phi, F$
$U^{ADV}_{i+1/2,j+1}, \phi, F$

$\psi^{ADV}_{i-1,j-3/2}$
$\psi^{ADV}_{i,j-3/2}$
$\psi^{ADV}_{i+1,j-3/2}$

$U^{ADV}_{i-3/2,j+2}, \phi, F$
$U^{ADV}_{i-1/2,j+2}, \phi, F$
$U^{ADV}_{i+1/2,j+2}, \phi, F$

$\psi^{ADV}_{i-1,j-5/2}$
$\psi^{ADV}_{i,j-5/2}$
$\psi^{ADV}_{i+1,j-5/2}$

Figure 1: Diagram of where the discretely divergence-free edge velocity field $U^{ADV}$, level set function $\phi$ and volume of fluid function $F$ are located in relation to the computational grid and the physical boundary.
Physical boundary

<table>
<thead>
<tr>
<th>Gas or Liquid</th>
<th>$\phi$, $F$</th>
<th>$\phi$, $F$</th>
<th>$\phi$, $F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_{i-1,J}$</td>
<td>$U_{i,J}$</td>
<td>$U_{i+1,J}$</td>
<td></td>
</tr>
<tr>
<td>$p_{i-3/2,J-1/2}$</td>
<td>$p_{i-1/2,J-1/2}$</td>
<td>$p_{i+1/2,J-1/2}$</td>
<td></td>
</tr>
<tr>
<td>$\phi$, $F$</td>
<td>$\phi$, $F$</td>
<td>$\phi$, $F$</td>
<td></td>
</tr>
<tr>
<td>$U_{i-1,J-1}$</td>
<td>$U_{i,J-1}$</td>
<td>$U_{i+1,J-1}$</td>
<td></td>
</tr>
<tr>
<td>$p_{i-3/2,J-3/2}$</td>
<td>$p_{i-1/2,J-3/2}$</td>
<td>$p_{i+1/2,J-3/2}$</td>
<td></td>
</tr>
<tr>
<td>$\phi$, $F$</td>
<td>$\phi$, $F$</td>
<td>$\phi$, $F$</td>
<td></td>
</tr>
<tr>
<td>$U_{i-1,J-2}$</td>
<td>$U_{i,J-2}$</td>
<td>$U_{i+1,J-2}$</td>
<td></td>
</tr>
<tr>
<td>$p_{i-3/2,J-5/2}$</td>
<td>$p_{i-1/2,J-5/2}$</td>
<td>$p_{i+1/2,J-5/2}$</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Diagram of where the discrete variables $U$, $p$, $\phi$ and $F$, are located in relation to the computational grid and the physical boundary.
Figure 3: The distances $d$ assigned to points $A$, $F$, and $E$ respectively are $|AB|$, $|FG|$ and $|EH|$. 
Figure 4: Comparison with expected answer for Zalesak’s problem after one full revolution of notched disk. Grid resolution is 100x100.
Figure 5: perturbation in major amplitude for zero gravity drop oscillations. $\mu_l = 1/200$, $\gamma = 1/2$, density ratio 100:1, viscosity ratio 100:1. Comparison of coupled level set method (CLS) with the level set method.
Figure 6: Falling 1mm 2d water drop in air; density ratio 816:1, viscosity ratio 64:1, number of grid cells $32 \times 128$. 
Figure 7: Falling 1mm 2d water drop in air; surface tension turned off. Density ratio 816:1, viscosity ratio 64:1, number of grid cells $32 \times 128$. 
Figure 8: Capillary Instability. $\rho_w/\rho_a = 816$, $\mu_w/\mu_a = 64$. Grid resolution is 64x128.
Figure 9: Capillary Instability; Level set method. $\rho_w/\rho_a = 816$, $\mu_w/\mu_a = 64$. Grid resolution is $64\times128$. 
Figure 10: Capillary Instability. $\rho_w/\rho_a = 1$, $\mu_w/\mu_a = 1$. Grid resolution is 64x128.
Figure 11: Spherical gas bubble in liquid: density ratio 816:1, We=200. Results computed using the volume of fluid method (thin contour) are compared to results computed using the boundary integral method (thick contour)
Figure 12: Spherical gas bubble in liquid: density ratio 816:1, We=200. Results computed using the volume of fluid method (thin contour) are compared to results computed using the level set method (thick contour).
Figure 13: Rise of an initially spherical gas bubble in viscous liquid. Results computed using the CLS method. density ratio 714:1, viscosity ratio 6667:1, $Re = 9.7$, $We = 7.6$, $Fr = 0.78$
Figure 14: Comparison of position of center of mass of rising viscous gas bubble in liquid to “best fit” straight line of the position. Expected slope as predicted by experiments is 1. Results computed using the CLS method. density ratio 714:1, viscosity ratio 6667:1, \( Re = 9.7, We = 7.6, Fr = 0.78 \)
Figure 15: Evolution of 4mm air bubble rising to an air/water interface. $\mu_l/\mu_g = 100$, $\rho_l/\rho_g = 1000$, $Fr = 0.64$, $Re = 474$, and $We = 1.0$. Results computed using CLS method.
Figure 16: Rise of inviscid air bubble in water. \( We = 200 \), effective fine grid 64x64x128.
Figure 17: Spherical gas bubble in liquid; density ratio 816:1; We=200. Left: Cross section of three-dimensional results (y=2, x-z plane), effective fine grid 64x64x128, dimensions of domain: 4x4x8. Right: Axisymmetric results, effective fine grid 32x128, dimensions of domain 2x8.
Figure 18: Non-Axisymmetric merging of two viscous gas bubbles, effective fine grid 64x64x128.