 FDMP, vol.3, no.1, pp.21-36, 2007

Improvements for calculating two-phase bubble and drop motion using an adaptive sharp interface method.

Mark Sussman[1], **and Mitsuhiro Ohta**[2]

(*Communicated by John Lowengrub*)

Abstract: In this paper, we describe new techniques for numerically approximating two-phase flows. Specifically, we present new techniques for treating the viscosity and surface tension terms that appear in the Navier-Stokes equations for incompressible two-phase flow. Our resulting numerical method has the property that results computed using our two-phase algorithm approach the corresponding "one-phase" algorithm in the limit of zero gas density/viscosity; i.e. the two-phase results approach the one-phase free-boundary results in the limit that the gas is assumed to become a uniform pressure void. By grid convergence checks and comparison with previous experimental data, we shall demonstrate the advantages of our new proposed discretizations.

Keyword: volume-of-fluid method, sharp interface, multiphase flow, surface tension, viscosity

1 Introduction

The accurate numerical treatment of surface tension and viscous effects in two-fluid problems has many applications. For example, Cristini and Renardy (2006) computed flows applicable to monodisperse emulsions. Jimenez, Sussman, and Ohta (2005) computed bubble motion in viscoelastic flows, which has applications in the study of bioreactors and chemical separators.

In this paper, we present new techniques for treating the surface tension terms and viscosity terms as they appear in the Navier-Stokes equations for two-phase flows. The governing equations for incompressible, immiscible two-phase flow can be written according to Chang, Hou, Merriman, and Osher (1996) as:

$$\rho \frac{DU}{Dt} = \nabla \cdot (-pI + 2\mu D) + \rho g \hat{z} - \sigma \kappa(F) \nabla H, \quad (1)$$

$$\nabla \cdot U = 0,$$

$$\frac{D\phi}{Dt} = 0, \quad (2)$$

$$\frac{DF}{Dt} = 0, \quad (3)$$

$$\rho = \rho_L H(\phi) + \rho_G (1 - H(\phi)),$$

$$\mu = \mu_L H(\phi) + \mu_G (1 - H(\phi)),$$

$$\kappa(\phi) = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \quad (4)$$

or,

$$\kappa(F) = \nabla \cdot \frac{\nabla F}{|\nabla F|}, \quad (5)$$

$$H(\phi) = \begin{cases} 1 & \phi \geq 0 \\ 0 & \phi < 0 \end{cases} \quad (6)$$

ϕ is a level set function which is positive in liquid and negative in gas. F is a volume-of-fluid

[1] Florida State University, Dept. of Math, Tallahassee, FL, USA.

[2] Muroran Institute of Technology, Dept. of Applied Chemistry, Muroran, Hokkaido, Japan

function which, for each computational cell, represents the fraction of fluid that is liquid. i.e. $F = 1$ if a cell contains only liquid, $F = 0$ if a cell contains only gas and $0 < F < 1$ otherwise. In the numerical algorithm we present in this paper, we shall couple the solutions of ϕ and F in the same manner as described by Sussman and Puckett (2000).

The other variables are defined as follows: U is the velocity vector, p is the pressure, $D = \frac{1}{2}(\nabla U + \nabla U^T)$ is the rate of deformation tensor, g corresponds to the acceleration due to gravity, μ_L (μ_G) is the viscosity of liquid (gas), ρ_L (ρ_G) is the density of liquid (gas), κ is the curvature, and σ is the coefficient of surface tension.

Numerical algorithms for calculating the surface tension force term,

$$\sigma\kappa\nabla H,$$

have been presented from the perspective of front tracking algorithms, volume-of-fluid algorithms, level set methods, and various hybrid methods. Example implementations of surface-tension in the context of front tracking methods are presented by Nobari, Jan, and Tryggvason (1993), Unverdi and Tryggvason (1992), Udaykumar, Rao, and Shyy (1996), Ye, Shyy, and Chung (2001), and Popinet and Zaleski (1999). Examples in the context of volume-of-fluid methods are presented by Brackbill, Kothe, and Zemach (1992), Francois, Cummins, Dendy, Kothe, Sicilian, and Williams (2006), Aleinov and Puckett (1995), Rudman (1998), and Gueyffier, Li, Nadim, Scardovelli, and Zaleski (1999). Examples in the context of the level set method are presented by Sussman, Smereka, and Osher (1994),Kang, Fedkiw, and Liu (2000), Liu, Krishnan, Marella, and Udaykumar (2005). Finally, examples in the context of various hybrid methods have been presented by Sussman and Puckett (2000), Sussman (2003), and Enright, Marschner, and Fedkiw (2002). The ability to accurately calculate surface tension can be important for modeling the impact of drops on surfaces, contact line dynamics, bubble motion, and the break-up of liquid jets. In previous work by Sussman (2003), a second order coupled level set and volume of fluid method was presented for calculating bubble growth and collapse. In that work, the "height fraction" technique, as originally described by Helmsen, Colella, and Puckett (1997), was employed to accurately calculate curvature directly from volume fractions. In this work we demonstrate that the "height fraction" technique can be extended to calculate curvature to *any* order of accuracy.

Conventional wisdom would have it that only a level set representation of an interface is capable of having a very high order accurate method for extracting the interface curvature. For example, a "spectral" level set approach was presented by Sussman and Hussaini (2003). Previous methods for extracting curvature from volume fractions have been proposed by Chorin (1985) (osculating circle technique), Poo and Ashgriz (1989), Aleinov and Puckett (1995) (convolution technique), Williams, Kothe, and Puckett (1998), Renardy and Renardy (2002) ("PROST"), Helmsen, Colella, and Puckett (1997) (height fraction), Sussman (2003) (height fraction), Francois, Cummins, Dendy, Kothe, Sicilian, and Williams (2006) (height fraction), and Cummins, Francois, and Kothe (2005) (height fraction). The challenge in accurately calculating curvature from the volume-of-fluid function F is the fact that F transitions sharply from 0 (computational cell containing only gas) to 1 (computational cell containing only liquid). Of all the approaches for calculating curvature from volume fractions, the "height fraction" approach is the most direct, and, in contrast to convolution techniques, it is "localized."

In this paper we present a systematic approach to extending the "height fraction" approach to any order of accuracy (i.e. higher than second order). We show specifically in this paper how to calculate curvature up to fourth order accuracy and provide guidelines of how to compute curvature to higher orders. We remark that for higher than fourth order accuracy, one would have to retain an ever increasing number of terms in the Taylor series expansion of the height function, but we believe this procedure can be automated if necessary to get very large orders of accuracy.

In addition to presenting new techniques for discretizing the curvature, we shall also present new "sharp-interface" techniques for discretizing the viscous force terms,

$$\nabla \cdot (2\mu D).$$

In Sussman, Smereka, and Osher (1994) and Kang, Fedkiw, and Liu (2000), the viscous force terms were treated explicitly. An explicit treatment is unacceptable for handling a wide range of bubble or drop deformation problems. For example, many non-Newtonian fluids also have large values for the viscosity coefficient, in these cases the explicit viscous timestep constraint is too restrictive. In Liu, Krishnan, Marella, and Udaykumar (2005), the viscous terms were split into two parts; one part reduced to the Laplacian of the velocity, the other part is related to the jump in stress at the liquid/gas boundary. The first part (Laplacian of velocity) was treated implicitly and the second part was treated explicitly. For cases in which the gas viscosity equals the liquid viscosity, there would be no time step constraints on the method proposed by Liu, Krishnan, Marella, and Udaykumar (2005), but for cases where there is a jump in viscosity, one would have the same time-step constraints as a method that treated the viscous terms explicitly.

Sussman, Almgren, Bell, Colella, Howell, and Welcome (1999) and Li, Renardy, and Renardy (2000) treated the viscous terms implicitly and semi-implicitly respectively. Both approaches, Sussman, Almgren, Bell, Colella, Howell, and Welcome (1999) and Li, Renardy, and Renardy (2000), had no time-step constraint related to the viscous terms, but neither of these methods was for a "sharp-interface" approach. The method proposed by Sussman, Almgren, Bell, Colella, Howell, and Welcome (1999) led to a coupled system that was simultaneously solved for all three components of the velocity; the resulting matrix equation for the coupled system necessitated a complicated matrix iterative solver which was not guaranteed to converge. In recent work, Sussman, Smith, Hussaini, Ohta, and Zhi-Wei (2007) presented a two-phase *sharp-interface* approach in which the viscous terms were handled implicitly, but similarly as in Sussman, Almgren, Bell, Colella, Howell, and Welcome (1999), the resulting matrix system was not guaranteed to converge. Another drawback to the method proposed by Sussman, Smith, Hussaini, Ohta, and Zhi-Wei (2007), is that it is difficult to extend the matrix iterative solver from solving problems on a single, fixed, uniform rectangular grid, to a composite matrix iterative solver for solving problems on an adaptive hierarchy of grids.

In this paper, we shall extend upon the ideas of Li, Renardy, and Renardy (2000) for treating the viscous force terms. In other words, we still will not have a time-step constraint associated with the viscous force terms, and at the same time, the resulting matrix system will be decoupled and diagonally dominant. Therefore the method we present here for discretizing the viscous force terms will be more robust than that presented by Sussman, Almgren, Bell, Colella, Howell, and Welcome (1999) and Sussman, Smith, Hussaini, Ohta, and Zhi-Wei (2007). We incorporate the same splitting techniques as Li, Renardy, and Renardy (2000) but in a sharp interface framework instead of a smoothed interface framework; and on an adaptive hierarchy of grids instead of a single uniform grid. By treating the coupled viscous force terms explicitly and the remainder of the viscous force terms implicitly, we can use previously developed matrix iterative solvers that have been designed for an adaptive hierarchy of grids. For the results in this paper, we use a solver developed for an adaptive hierarchy of grids that exploits multigrid techniques across adaptive levels, and the multigrid preconditioned conjugate gradient on a specific level; we refer the reader to the work of Sussman (2005). In previous work, the composite solver described by Sussman (2005) was applied to the pressure projection equations. In this work, we use this composite solver for each component of the viscous force terms as well.

Our discretization of the viscous force terms has the property that in the limit of zero gas density, zero gas viscosity, our discretization of the viscous force terms reduces to the discretization that would appear if one were just computing the solution to the liquid only, and treating the gas pres-

sure as spatially constant.

2 Surface Tension: Curvature discretization using "height fractions"

In this section we describe high order discretization techniques for the curvature κ, which appears in the surface tension force term,

$$\sigma\kappa\nabla H.$$

The curvature of an interface is computed to second or fourth order accuracy from the volume fractions. The method is based on reconstructing the "height" function, Helmsen, Colella, and Puckett (1997), from the volume fractions. The first step in determining the curvature is to first determine whether the interface is more aligned horizontally (along the x-y plane), vertically in the x-z plane, or vertically in the y-z plane. Since we represent the interface using the coupled levelset and volume-of-fluid method, the orientation is determined by looking at the maximum component of the gradient of the levelset function. Without loss of generality, we assume that the free surface is oriented more horizontal than vertical; i.e. we assume that $|\phi_z| \geq |\phi_x|$ and $|\phi_z| \geq |\phi_y|$ where,

$$(\phi_z)_{i,j,k} \approx \frac{\phi_{i,j,k+1} - \phi_{i,j,k-1}}{2\Delta z}$$

For a second order curvature algorithm, a 3x3x7 stencil of volume fractions is constructed about cell (i,j,k). For the fourth order algorithm, a 5x5x13 stencil of volume fractions is constructed about cell (i,j,k). The following 3x3 (5x5 for fourth order) vertical sums, $F_{i',j'}$, $i' = -1\ldots1$, $j' = -1\ldots1$, are calculated as (e.g. second order case),

$$F_{i',j'} = \sum_{k'=-3}^{3} F_{i'+i,j'+j,k'+k},$$

where $F_{i',j'}$ (2 indices) is defined as a vertical sum and $F_{i,j,k}$ (3 indices) is the volume fraction of liquid in cell (i,j,k). Please see Figure 1 for a two dimensional illustration of a valid 3x7 stencil on the left and an invalid 3x3 stencil on the right. As long as the interface does not exit the bottom or top of the 3x3x7 second order stencil (or 5x5x13 fourth order stencil), the vertical sums are exact integrals of the height function $h(x,y)$ (up to a constant); i.e.

$$F_{i',j'}\Delta x\Delta y\Delta z = \int_{x_{i+i'-1/2}}^{x_{i+i'+1/2}} \int_{y_{j+j'-1/2}}^{y_{j+j'+1/2}} h(x,y)dxdy + C. \quad (7)$$

It can be shown that $\Delta z(F_{1,0} - F_{-1,0})/(2\Delta x)$ is a second order approximation to $h_x(x_i,y_j)$ and that $\Delta z(F_{1,0} - 2F_{0,0} + F_{-1,0})/\Delta x^2$ is a second order approximation to $h_{xx}(x_i,y_j)$. In general, one expands $h(x,y)$ in a Taylor series,

$$\begin{aligned} h(x,y) =& h(x_i,y_j) \\ &+ (x-x_i)h_x(x_i,y_j) \\ &+ \frac{1}{2}(x-x_i)^2 h_{xx}(x_i,y_j) \\ &+ \frac{1}{6}(x-x_i)^3 h_{xxx}(x_i,y_j) \\ &+ \frac{1}{24}(x-x_i)^4 h_{xxxx}(x_i,y_j) \\ &+ (y-y_j)h_y(x_i,y_j) \\ &+ \frac{1}{2}(y-y_j)^2 h_{yy}(x_i,y_j) \\ &+ \frac{1}{6}(y-y_j)^3 h_{yyy}(x_i,y_j) \\ &+ \frac{1}{24}(y-y_j)^4 h_{yyyy}(x_i,y_j) \\ &+ (x-x_i)(y-y_j)h_{xy}(x_i,y_j) \\ &+ \frac{1}{2}(x-x_i)^2(y-y_j)h_{xxy}(x_i,y_j) \\ &+ \frac{1}{2}(x-x_i)(y-y_j)^2 h_{xyy}(x_i,y_j) \\ &+ \frac{1}{4}(x-x_i)^2(y-y_j)^2 h_{xxyy}(x_i,y_j) \\ &+ \text{higher order terms.} \end{aligned}$$

After integrating the Taylor series expansion of

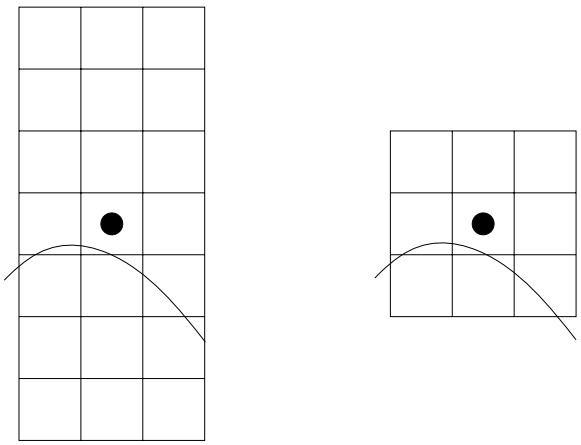

Figure 1: Two dimensional illustration of a valid 3x7 stencil on the left and an invalid 3x3 stencil on the right for calculating curvature using the height function approach. The vertical sums on the right are invalid since the interface exits the bottom of the 3x3 stencil.

$h(x,y)$ and using (7), one has,

$$\begin{aligned} F_{i',j'}\Delta z =& h(x_{i+i'},y_{j+j'}) \\ &+ \frac{1}{24}\Delta x^2 h_{xx}(x_{i+i'},y_{j+j'}) \\ &+ \frac{1}{(16)(120)}\Delta x^4 h_{xxxx}(x_{i+i'},y_{j+j'}) \\ &+ \frac{1}{24}\Delta y^2 h_{yy}(x_{i+i'},y_{j+j'}) \\ &+ \frac{1}{(16)(120)}\Delta y^4 h_{yyyy}(x_{i+i'},y_{j+j'}) \\ &+ \frac{1}{(24)(24)}\Delta x^2\Delta y^2 h_{xxyy}(x_{i+i'},y_{j+j'}) \\ &+ \text{higher order terms.} \end{aligned} \tag{8}$$

For a horizontally orientated surface, the curvature is written as,

$$\kappa = \nabla \cdot n$$

where,

$$n = \left(-\frac{h_x}{\sqrt{1+h_x^2+h_y^2}}, -\frac{h_y}{\sqrt{1+h_x^2+h_y^2}}, \frac{1}{\sqrt{1+h_x^2+h_y^2}} \right).$$

For a fourth order approximation to the curvature, we must approximate h_x, h_y, h_{xx}, h_{yy} and h_{xy} with fourth order accuracy. We assume the discretization of each of these terms has the form,

$$\frac{\partial^{l+m} h(x_i,y_j)}{\partial x^l \partial y^m} = \Delta z \sum_{i'=-2}^{2}\sum_{j'=-2}^{2} A_{i',j'}^{l,m} F_{i',j'}$$

where $l = 0,1,2$ and $m = 0,1,2$. The coefficients, $A_{i',j'}^{l,m}$ ($i' = -2\ldots2$ and $j' = -2\ldots2$), are determined by the "method of undetermined coefficients" in which one uses the relation (8) to relate $F_{i',j'}$ to h, and the fact that our discretization for the derivatives of h should be exact for the polynomials $h_{p,q}(x,y) = (x-x_i)^p (y-y_j)^q$ where $p = 0\ldots4$ and $q = 0\ldots4$. As a result, for each derivative (l,m) of h, one constructs a matrix system of equations with 25 equations and 25 unknown coefficients $A_{i',j'}^{l,m}$, $i' = -2\ldots2$, $j' = -2\ldots2$. Each row in the matrix system corresponds to a different $h_{p,q}(x,y)$, $p = 0\ldots4$, $q = 0\ldots4$. The coefficients on a particular row are determined by (8). For example the 7th row, 1st column of the matrix is given by $F_{-2,-2}$ which is determined by plugging $h_{1,1}(x,y)$ into (8). The 7th row entry of the right hand side of the matrix system corresponds to

$$\frac{\partial^{l+m} h_{1,1}(x_i,y_j)}{\partial x^l \partial y^m}.$$

In two dimensions, one has the following fourth order approximations,

$$\frac{\partial h(x_i)}{\partial x} \approx \frac{\Delta y}{\Delta x}\left(\frac{5}{48}(F_{-2}-F_2) + \frac{17}{24}(F_1 - F_{-1}) \right)$$

$$\frac{\partial^2 h(x_i)}{\partial x^2} \approx \frac{\Delta y}{\Delta x^2}\left(\frac{-1}{8}(F_{-2}+F_2) + \frac{3}{2}(F_1+F_{-1}) - \frac{11}{4}F_0 \right)$$

Our derivation above for approximating curvature from volume fractions is dependent on the fact that a column sum of volume fractions corresponds to the associated integral of the height function (7). If the interface should pass through any 1x1x13 column (4th order method) of data more than once, then the height function is not well defined. In this case we implement a fix

that does not sacrifice the order of accuracy of our method. Given a 1x1x13 column of data, denoted as F_k, $k = -6 \ldots 6$, our fix is implemented as follows:

- construct a temporary level set function, $\phi_k = F_k - 1/2$.
- Determine the critical index k^* that is closest to zero among the indices of cells where the temporary level set function changes sign: i.e.

$$k^* = sign(k) \min_{\phi_k \phi_{k+1} \leq 0} |k|.$$

- Determine the zero crossing k_{cross} between k^* and $k^* + 1$:

$$k_{cross} = k^* + \frac{\phi_{k^*}}{\phi_{k^*+1} - \phi_{k^*}}$$

- Reinitialize ϕ_k to be the distance from k_{cross}. For $k = -6 \ldots 6$,

$$\phi_k = \begin{cases} k - k_{cross} & z_{k^*+1} > z_{k^*} \\ k_{cross} - k & z_{k^*+1} < z_{k^*} \end{cases}$$

- Fix the column data F_k,

$$F_k = \begin{cases} 0 & \phi_k < -2 \\ 1 & \phi_k > 2 \\ F_k & \text{otherwise} \end{cases}$$

2.1 Numerical validation of curvature discretization for a circle

We check our curvature discretization algorithm for a circle in 2d or a sphere in 3d. In 2d, we have a unit circle located at the point (2,2) in a 4x4 domain. In 3d, we have a unit sphere located at the origin in a 2x2x2 domain. Symmetric boundary conditions are used at the borders of the domain. As demonstrated by tables 1, and 2, we get the appropriate order of accuracy for our high order height fraction curvature discretization schemes.

Table 1: Convergence study for computing curvatures from volume fractions of a unit circle in 2d. Errors for the second order and fourth order discretizations are reported.

Δx	E^{2nd}_{max}	E^{4th}_{max}	E^{2nd}_{avg}	E^{4th}_{avg}
1/16	0.0031	0.00104	0.0019	0.00016
1/32	0.0007	4.21E-5	0.0005	7.9E-6

Table 2: Convergence study for computing curvatures from volume fractions of a unit sphere in 3d. Errors for the second order and fourth order discretizations are reported.

Δx	E^{2nd}_{max}	E^{4th}_{max}	E^{2nd}_{avg}	E^{4th}_{avg}
1/16	0.050	0.03288	0.0035	0.00079
1/32	0.010	0.00084	0.0009	2.13E-5

2.2 Parasitic currents

In this section we test our implementation of surface tension for the problem of a static 2d drop with diameter D. We assume the drop is surrounded by gas with uniform pressure and zero viscosity. The exact solution for such a problem is that the velocity u is identically zero. If we scale the Navier-Stokes equations by the time scale $T = D\mu/\sigma$, and by the velocity scale $U = \sigma/\mu$, then the non-dimensionalized Navier-Stokes equations in the drop become,

$$\frac{Du}{Dt} = -\nabla p + Oh^2 \Delta u - Oh^2 \kappa \nabla H.$$

where the Ohnesorge number Oh is defined as,

$$Oh = \frac{\mu}{\sqrt{\sigma \rho D}}.$$

The numerical simulation uses the coupled levelset and volume-of-fluid (CLSVOF) algorithm described in Sussman (2003). We investigate the maximum velocity of our numerical method for varying grid resolutions at the dimensionless time $t = 250$. The dimensions of our computational grid are $[0,1]$x$[0,1]$ with symmetric boundary conditions at all boundaries. A unit diameter drop is placed at the origin of our domain. In other

Table 3: Convergence study for static droplet with surface tension (parasitic currents test). Droplet is surrounded by gas with uniform pressure and zero viscosity. Maximum velocity at $t = 250$ is shown. $1/Oh^2 = 12000$. The liquid density is one. Results for the second order discretization of curvature and the fourth order discretization are reported.

Δx	max. velocity (2nd)	max. velocity (4th)
1.0/32	1.4E-6	3.6E-7
1.0/64	1.5E-7	1.1E-8

words, we only compute with a quarter of the drop in the domain, and assume symmetry for the solution of the rest of the drop. Our tolerance for the pressure solver and viscous solver is $1.0E-12$. In table 3 we display results of our grid refinement study for $1/Oh^2 = 12000$. We used both the second order height fraction algorithm and the fourth order height fraction algorithm for calculating curvature. Our results indicate at least second order convergence when the second order curvature discretization is used, and at least fourth order convergence when the fourth order curvature discretization is used. For a reference of previous results for parasitic currents, we refer the reader to work by Popinet and Zaleski (1999) and Francois, Cummins, Dendy, Kothe, Sicilian, and Williams (2006).

As a remark, the overall accuracy of our numerical method is first order since the temporal discretization approach for the viscous force terms (backwards Euler for the decoupled terms) is only first order accurate. But, for problems which are dominated by surface tension effects, the improved accuracy in calculating curvature can have a positive impact on the overall error since the error for the discretization of the curvature dominates the overall error anyway. In our results here for parasitic currents, and also our results in section 6.1 for a 2D drop in shear flow, the improved accuracy in discretizing the curvature reduces the overall error significantly, even though the formal overall order of our method is only first order.

3 Viscosity: An adaptive, sharp interface treatment for the viscous force terms

We present in this section a simple and robust adaptive method for computing the viscous forces as they appear in (1),

$$\nabla \cdot (2\mu D). \tag{9}$$

Our algorithm follows the same strategy as proposed by Li, Renardy, and Renardy (2000) in which we discretize the coupling terms in (9) explicitly and the remainder of the terms implicitly. It was shown by Li, Renardy, and Renardy (2000) that the following temporal discretization for the viscous forces is stable for any time step:

$$\rho \frac{u^{**} - u^*}{\Delta t} = \nabla \cdot \mu \nabla u^{**} + (\mu u_x^{**})_x + (\mu v_x^*)_y + (\mu w_x^*)_z \tag{10}$$

$$\rho \frac{v^{**} - v^*}{\Delta t} = \nabla \cdot \mu \nabla v^{**} + (\mu v_y^{**})_y + (\mu u_y^*)_x + (\mu w_y^*)_z$$

$$\rho \frac{w^{**} - w^*}{\Delta t} = \nabla \cdot \mu \nabla w^{**} + (\mu w_z^{**})_z + (\mu u_z^*)_x + (\mu v_z^*)_y$$

The non-coupling terms are discretized using standard finite volume techniques. For example, we approximate the terms $\nabla \cdot \mu \nabla u^{**} + (\mu u_x^{**})_x$ in (10) as,

$$\frac{2\mu_{i+1/2,j,k}(u_{i+1,j,k} - u_{i,j,k}) - 2\mu_{i-1/2,j,k}(u_{i,j,k} - u_{i-1,j,k})}{\Delta x^2} + \frac{\mu_{i,j+1/2,k}(u_{i,j+1,k} - u_{i,j,k}) - \mu_{i,j-1/2,k}(u_{i,j,k} - u_{i,j-1,k})}{\Delta y^2} + \frac{\mu_{i,j,k+1/2}(u_{i,j,k+1} - u_{i,j,k}) - \mu_{i,j,k-1/2}(u_{i,j,k} - u_{i,j,k-1})}{\Delta z^2}$$

where we define the face centered viscosity $\mu_{i+1/2,j}$ "sharply" as,

$$\mu_{i+1/2,j} = \frac{1}{\frac{\theta_{i+1/2,j}}{\mu_L} + \frac{1-\theta_{i+1/2,j}}{\mu_G}}.$$

$\theta_{i+1/2,j}$ is the height fraction, also used by Gibou, Fedkiw, Cheng, and Kang (2002) and Liu, Krishnan, Marella, and Udaykumar (2005), which is

given by,

$$\theta_{i+1/2,j}(\phi) = \begin{cases} 1 & \phi_{i+1,j}, \phi_{i,j} \geq 0 \\ 0 & \phi_{i+1,j}, \phi_{i,j} < 0 \\ \frac{\phi^+_{i+1,j}+\phi^+_{i,j}}{|\phi_{i+1,j}|+|\phi_{i,j}|} & \text{otherwise} \end{cases} \quad (11)$$

The "+" superscript stands for the "positive part:" i.e., $a^+ \equiv \max(a,0)$.

The (combined) density is also defined "sharply" as,

$$\rho = \begin{cases} \rho^L & \phi \geq 0 \\ \rho^G & \phi < 0 \end{cases}$$

In previous work, the coupling terms were discretized as follows: e.g. for the term $(\mu v^*_x)_y$ in 2d, one has,

$$\frac{1}{4\Delta x \Delta y} \cdot \begin{pmatrix} \mu_{i,j+\frac{1}{2}}(v_{i+1,j+1}+v_{i+1,j}-v_{i-1,j+1}-v_{i-1,j}) \\ -\mu_{i,j-\frac{1}{2}}(v_{i+1,j}+v_{i+1,j-1}-v_{i-1,j}-v_{i-1,j-1}) \end{pmatrix}$$

Unfortunately, with the above discretization for the coupling terms, velocities in gas cells ($\mu_G = 0$) could be accidentally included in the discretization of the coupling terms in liquid cells. When the gas viscosity is zero, there should be no effect on the viscous force terms from the gas.

In this paper we discretize the explicit coupling terms, e.g. $(\mu v^*_x)_y$ or $(\mu u^*_y)_x$, as follows (in two dimensions),

$$(\frac{\partial \mu v_x}{\partial y})_{ij} \approx \begin{pmatrix} \mu_{i+1/2,j+1/2}(v_x)_{i+1/2,j+1/2-} \\ \mu_{i+1/2,j-1/2}(v_x)_{i+1/2,j-1/2+} \\ \mu_{i-1/2,j+1/2}(v_x)_{i-1/2,j+1/2-} \\ \mu_{i-1/2,j-1/2}(v_x)_{i-1/2,j-1/2} \end{pmatrix} \Big/ (2\Delta y)$$

$$(\frac{\partial \mu u_y}{\partial x})_{ij} \approx \begin{pmatrix} \mu_{i+1/2,j+1/2}(u_y)_{i+1/2,j+1/2-} \\ \mu_{i-1/2,j+1/2}(u_y)_{i-1/2,j+1/2+} \\ \mu_{i+1/2,j-1/2}(u_y)_{i+1/2,j-1/2-} \\ \mu_{i-1/2,j-1/2}(u_y)_{i-1/2,j-1/2} \end{pmatrix} \Big/ (2\Delta x)$$

The viscosity at a node is given by

$$\mu_{i+1/2,j+1/2} = \begin{cases} \mu_L & \theta_{i+1/2,j+1/2} = 1 \\ \mu_G & \theta_{i+1/2,j+1/2} = 0 \\ 0 & \mu_G = 0 \text{ and } 0 < \theta_{i+1/2,j+1/2} < 1 \\ \mu_{LG} & \text{otherwise,} \end{cases}$$

where,

$$\mu_{LG} = \frac{\mu_G \mu_L}{\mu_G \theta_{i+1/2,j+1/2} + \mu_L(1-\theta_{i+1/2,j+1/2})}.$$

$\theta_{i+1/2,j+1/2}$ is a "node fraction" defined as,

$$\theta_{i+1/2,j+1/2}(\phi) = \begin{cases} 1 & \phi_{i+1,j}, \phi_{i,j}, \phi_{i,j+1}, \phi_{i+1,j+1} \geq 0 \\ 0 & \phi_{i+1,j}, \phi_{i,j}, \phi_{i,j+1}, \phi_{i+1,j+1} < 0 \\ \theta_{ND} & \text{otherwise.} \end{cases}$$

The "+" superscript stands for the "positive part" (i.e., $a^+ \equiv \max(a,0)$), and

$$\theta_{ND} = \frac{\phi^+_{i+1,j}+\phi^+_{i,j+1}+\phi^+_{i,j}+\phi^+_{i+1,j+1}}{|\phi_{i+1,j}|+|\phi_{i,j+1}|+|\phi_{i,j}|+|\phi_{i+1,j+1}|}.$$

The components of the parts of the deformation tensor that are handled explicitly, e.g. the coupled terms, $(u_y)_{i+1/2,j+1/2}$ in the equation for v and $(v_x)_{i+1/2,j+1/2}$ in the equations for u, are calculated at nodes using standard central differencing, i.e.,

$$(u_y)_{i+1/2,j+1/2} = \frac{u_{i+1,j+1}-u_{i+1,j}+u_{i,j+1}-u_{i,j}}{2\Delta y}.$$

Our discretization of the viscous terms have the following important properties:

1. if the gas viscosity is zero, the velocity in gas cells, $\phi_{i,j} < 0$, is never used. This enables our two-phase method to be equivalent to the corresponding one-phase method in the limit of zero gas density and zero gas viscosity (i.e. gas treated as a vacuum with uniform pressure).

2. The resulting matrix system for each velocity component can be written in the following form,

$$\alpha(x)p + \beta \nabla \cdot (A(x,y)\nabla p) = f(x,y), \quad (12)$$

where (12) is solved for p. A is a diagonal matrix. We solve (12) on an adaptive hierarchy of grids as described in section 4.

4 Matrix solver for adaptive hierarchy of grids

The equations that result when calculating the viscous force terms or when calculating the pressure projection step can be written generally as,

$$\alpha(x)p+\beta\nabla\cdot(A(x,y)\nabla p)=f(x,y)+\alpha(x)q+\beta\nabla\cdot V, \quad (13)$$

where A is a diagonal matrix and V is a given vector field (e.g., for the pressure projection step, V is the velocity to be projected and for the calculation of the viscous forces, $V\equiv 0$).

We discretize (13) as,

$$\alpha_{ij}p_{ij}+\beta(DGp)_{ij}=f_{ij}+\alpha_{ij}q_{ij}+\beta(DV)_{ij} \quad (14)$$

where, in 2d,

$$(Gp)_{i+1/2,j}=(A^{11})_{i+1/2,j}\frac{p_{i+1,j}-p_{ij}}{\Delta x}$$

$$(Gp)_{i,j+1/2}=(A^{22})_{i,j+1/2}\frac{p_{i,j+1}-p_{ij}}{\Delta y}$$

$$(DV)_{ij}=\frac{u_{i+1/2,j}-u_{i-1/2,j}}{\Delta x}+\frac{v_{i,j+1/2}-v_{i,j-1/2}}{\Delta y}$$

We solve (14) with a combination of the multigrid method and the multigrid preconditioned conjugate gradient method (MGPCG). We refer the reader to the work of Tatebe (1993) in regards to MGPCG. Our adaptive composite solver will converge to any specified tolerance ε, so long as each time the outer multigrid solver visits a level, we compute the solution on that level exactly. In our implementation, we compute the solution on each level (using MGPCG) to a tolerance of εx10^{-2}.

Our Multigrid+MGPCG algorithm for solving (14) follows the same "V-cycle" procedure as outlined by Briggs, Henson, and McCormick (2000):

1. Put (14) in residual correction form. For levels $\ell=0$ to ℓ_{max},

$$V^\ell = V^\ell - Gp^\ell_{predict}$$

$$q^\ell = q^\ell - p^\ell_{predict}$$

$$p^\ell = p^\ell_{predict}$$

2. For $\ell=\ell_{max}\ldots 0$,

1. if $\ell<\ell_{max}$, restrict $V^{\ell+1}$ to V^ℓ.
2. Calculate $R^\ell = f^\ell+\alpha q^\ell+\beta DV^\ell$.
3. if $\ell<\ell_{max}$, restrict $R^{\ell+1}$ to R^ℓ.
4. save V^ℓ to V^ℓ_{save}, q^ℓ to q^ℓ_{save}, and R^ℓ to R^ℓ_{save}.
5. $p^\ell_{correct}=0$.
6. iterate until the residual on level ℓ is less than the tolerance εx10^{-2}. Iterate using MGPCG,

   $$\alpha p^\ell_{correct}+\beta DGp^\ell_{correct}=R^\ell_{save}$$

7.

   $$R^\ell = R^\ell_{save}-\alpha p^\ell_{correct}-\beta DGp^\ell_{correct}$$

   $$V^\ell = V^\ell - Gp^\ell_{correct}$$

   $$q^\ell = q^\ell - p^\ell_{correct}$$

3. For $\ell=0\ldots\ell_{max}$,

1. if $\ell>0$, prolongate $p^{\ell-1}_{correct}$,

   $$p^\ell_{correct}=p^\ell_{correct}+I^\ell_{\ell-1}p^{\ell-1}_{correct}$$

2. iterate until the residual on level ℓ is less than the tolerance εx10^{-2}. Iterate using MGPCG,

   $$\alpha p^\ell_{correct}+DGp^\ell_{correct}=R^\ell_{save}$$

3.

   $$V^\ell = V^\ell_{save}-Gp^\ell_{correct}$$

   $$q^\ell = q^\ell_{save}-p^\ell_{correct}$$

   $$p^\ell = p^\ell+p^\ell_{correct}$$

4. repeat steps (2) and (3) until convergence.

5 Overall numerical method for two-phase flow

Prior to each timestep we are given a liquid velocity, $U^{L,n}$, and a total velocity U^n. $U^{L,n}$ corresponds to U^n except on gas faces, where we replace the gas velocity in $U^{L,n}$ with the extrapolated liquid velocity. $U^{L,n}$ is then used to calculate the nonlinear advective terms in the liquid, and also used to advance the free surface.

Prior to each time step, we are also given a level set function, ϕ^n, and a volume-of-fluid function, F^n. The level set function, and volume-of-fluid function are stored at cell centers. The velocity is stored at both cell centers and face-centers.

An outline of our numerical algorithm is as follows:

step 1: CLSVOF interface advection:

$$\phi_{ij}^{n+1} = \phi_{ij}^n - \Delta t [U^L \cdot \nabla \phi]_{ij}$$

$$F_{ij}^{n+1} = F_{ij}^n - \Delta t [U^L \cdot \nabla F]_{ij}$$

step 2: Calculate (cell centered) advective force terms:

$$\mathscr{A}_{ij}^L = [U^L \cdot \nabla U^L]_{ij}^n$$

$$\mathscr{A}_{ij} = [U \cdot \nabla U]_{ij}^n$$

The nonlinear advective terms are discretized using upwind, second order Van-Leer slope limiting; see e.g. van Leer (1979) and Sussman (2003) for a description of slope limiting.

step 3: Calculate (cell centered) semi-implicit viscous forces:

$$U_{ij}^n = \begin{cases} U_{ij}^{L,n} & \phi_{ij} \geq 0 \\ U_{ij}^n & \phi_{ij} < 0 \end{cases}$$

$$\mathscr{A}_{ij} = \begin{cases} \mathscr{A}_{ij}^L & \phi_{ij} \geq 0 \\ \mathscr{A}_{ij} & \phi_{ij} < 0 \end{cases}$$

$$U_{ij}^* = U_{ij}^n + \Delta t \left(-\mathscr{A}_{ij} + g\hat{z}\right).$$

$$\rho \frac{U^{**} - U^*}{\Delta t} = L^{**,uncoupled} + L^{*,coupled}$$

The term $L^{**,uncoupled}$ represents the uncoupled viscous force terms which are handled implicitly and the term $L^{*,coupled}$ represents the coupled viscous force terms which are handled explicitly (see section 3). The viscous coefficient μ is evaluated as a function of the updated levelset function ϕ^{n+1}.

step 4: Interpolate cell centered forces to face centered forces, calculate the face centered surface tension force, and then form the face centered intermediate velocity to be projected, $V_{i+1/2,j}$:

$$\mathscr{A}_{i+1/2,j}^L = \frac{1}{2}(\mathscr{A}_{i+1,j}^L + \mathscr{A}_{i,j}^L)$$

$$\mathscr{A}_{i+1/2,j} = \frac{1}{2}(\mathscr{A}_{i+1,j} + \mathscr{A}_{i,j})$$

$$\mathscr{A}_{i+1/2,j} = \begin{cases} \mathscr{A}_{i+1/2,j}^L & \phi_{ij} \geq 0 \text{ or } \phi_{i+1,j} \geq 0 \\ \mathscr{A}_{i+1/2,j} & \text{otherwise} \end{cases}$$

$$L_{i+1/2,j} = \frac{\begin{pmatrix} L_{ij}^{*,coupled} + L_{ij}^{**,uncoupled} + \\ L_{i+1,j}^{*,coupled} + L_{i+1,j}^{**,uncoupled} \end{pmatrix}}{4}$$

$$U_{i+1/2,j}^n = \begin{cases} U_{i+1/2,j}^{L,n} & \phi_{ij} \geq 0 \text{ or } \phi_{i+1,j} \geq 0 \\ U_{i+1/2,j}^n & \text{otherwise} \end{cases}$$

$$V_{i+1/2,j} = U_{i+1/2,j}^n + \tag{15}$$

$$\Delta t \begin{pmatrix} -\frac{\mathscr{A}_{i+1/2,j}}{2} \\ +\frac{2}{\rho_{i+1,j}+\rho_{i,j}} L_{i+1/2,j} \\ -[\frac{\sigma \kappa \nabla H}{\rho}]_{i+1/2,j} + g\hat{z} \end{pmatrix} \tag{16}$$

The surface tension term,

$$\frac{\sigma \kappa_{i+1/2,j} (\nabla H)_{i+1/2,j}}{\rho_{i+1/2,j}},$$

is discretized as,

$$\frac{\sigma \kappa_{i+1/2,j} \frac{H(\phi_{i+1,j})-H(\phi_{ij})}{\Delta x}}{\rho_{i+1/2,j}}$$

where

$$H(\phi) = \begin{cases} 1 & \phi \geq 0 \\ 0 & \phi < 0 \end{cases}$$

and

$$\rho_{i+1/2,j} = \rho_L \theta_{i+1/2,j} + \rho_G (1 - \theta_{i+1/2,j})$$

The discretization of the height fraction, $\theta_{i+1/2,j}$, is described by equation (11).

The curvature $\kappa_{i+1/2,j}$ is computed with second-order or fourth-order accuracy from the volume fractions as described in Section 2.

step 5: Implicit pressure projection step:

$$\nabla \cdot \frac{\nabla p}{\rho} = \nabla \cdot V \tag{17}$$

$$U^{n+1}_{i+1/2,j} = V - [\frac{\nabla p}{\rho}]_{i+1/2,j}$$

We solve the resulting linear system using a composite solver for an adaptive hierarchy of grids (see section 4).

step 6: Liquid velocity extrapolation; assign $U^{L,n+1}_{i+1/2,j} = U^{n+1}_{i+1/2,j}$ and then extrapolate $U^{L,n+1}_{i+1/2,j}$ into the gas region.

step 7: Interpolate face centered velocity to cell centered velocity:

$$U^{L,n+1}_{ij} = \frac{1}{2}(U^{L,n+1}_{i+1/2,j} + U^{L,n+1}_{i-1/2,j})$$

$$U^{n+1}_{ij} = \frac{1}{2}(U^{n+1}_{i+1/2,j} + U^{n+1}_{i-1/2,j})$$

6 Results

6.1 2D Drop in Shear Flow

For this test problem, we are initially given a drop with radius a located at the coordinants $(L/2, L/2)$ in a LxL physical domain. The density ratio and viscosity ratio between the drop and its surrounding fluid is 1. The horizontal velocity at the top wall is fixed at $u(x,L,t) = U_0$ and the horizontal velocity at the bottom wall is fixed at $u(x,0,t) = -U_0$. The velocity at the side walls is $u(0,y,t) = u(64,y,t) = \dot{\gamma}(y - L/2)$. The initial velocity is $u(x,y,0) = \dot{\gamma}(y - L/2)$. $\dot{\gamma}$ is the shear rate and satisfies $\dot{\gamma} = 2\frac{U_0}{L}$.

We assume the surface tension coefficient is $\sigma = 200.0$, the viscosity coefficient is $\mu = 100.0$, $a = 1$, $L = 64$, $U_0 = 32.0$, and the density of both fluids is $\rho^L = \rho^G = 1$. Given these parameters, the Capillary number is,

$$Ca = \frac{\mu \dot{\gamma} a}{\sigma} = 1/2,$$

and the Reynolds number is,

$$Re = \frac{\rho \dot{\gamma} a^2}{\mu} = 1/100.$$

We compute solutions to this problem on an adaptive hierarchy of grids and compare the results to those reported in Zheng, Lowengrub, Anderson, and Cristini (2005). Our base coarse grid has dimensions 64x64 and we compute with either 4 levels of adaptivity (effective fine grid resolution 1024x1024, $\Delta x_{fine} = 1/16$) or 5 levels of adaptivity (effective fine grid resolution 2048x2048, $\Delta x_{fine} = 1/32$). The computation with 5 levels of adaptivity was restarted from the 4 level computation at time $t = 12.3$. The actual number of grid cells advanced per time step after the drop had attained a steady state were 8192 and 19136 for the four level and five level computations respectively.

In Figures 2 and 3, we show the droplet at time $t = 16$. In Figure 4, we compare the steady half-length (major axis divided by initial radius) for the 4-level computation (1.52), 5-level computation (1.57) and the 4-level computation using fourth order curvature (1.58). The drop half-length reported by Zheng, Lowengrub, Anderson,

and Cristini (2005) using the boundary integral method is 1.583.

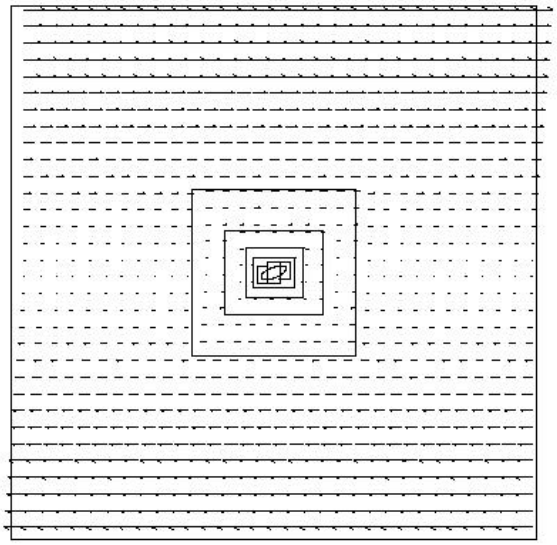

Figure 2: Steady shape of drop in a shear flow. $Ca = 0.5$, $Re = 1/100$. Five levels of adaptivity. Coarse level has 64x64 grid cells. Effective fine grid resolution 2048x2048. Drop half-length divided by a is 1.57; expected value is 1.583.

6.2 3D Axisymmetric bubble motion

We compute the steady state shapes of a gas bubble rising in a viscous Newtonian liquid. For comparison, we use the experimental results found in Bhaga and Weber (1981) and Hnat and Buckmaster (1976) and computational results in Ryskin and Leal (1984). In all of our bubble calculations, we computed on an adaptive hierarchy of grids with a base coarse grid of 24x72 and three levels of adaptivity. We have calculated these problems previously in the work of Sussman, Smith, Hussaini, Ohta, and Zhi-Wei (2007); the main difference is that in this paper, we treat the coupled viscous terms explicitly whereas in Sussman, Smith, Hussaini, Ohta, and Zhi-Wei (2007) they were treated implicitly. We find that the results whether the coupled viscous terms are treated explicitly or implicitly do not differ noticeably. The advantage of being able to treat the coupled viscous terms explicitly is that the matrix solver for solving the remainder of the viscous force terms is

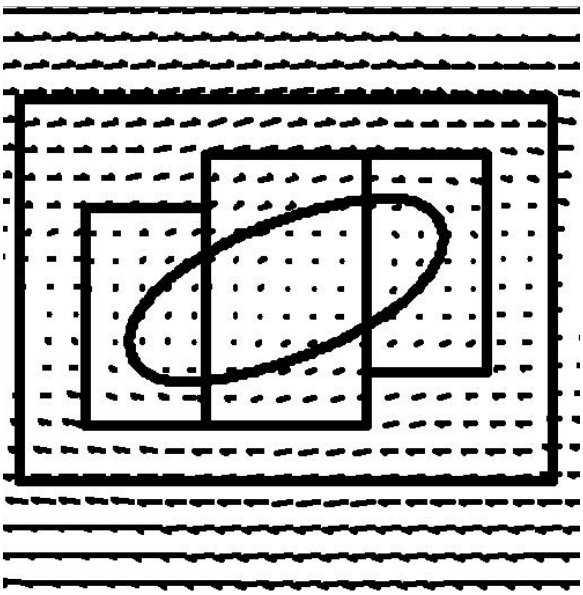

Figure 3: Magnification of Steady shape of drop in a shear flow. $Ca = 0.5$, $Re = 1/100$. Five levels of adaptivity. Coarse level has 64x64 grid cells. Effective fine grid resolution 2048x2048. Drop half-length divided by a is 1.57; expected value is 1.583.

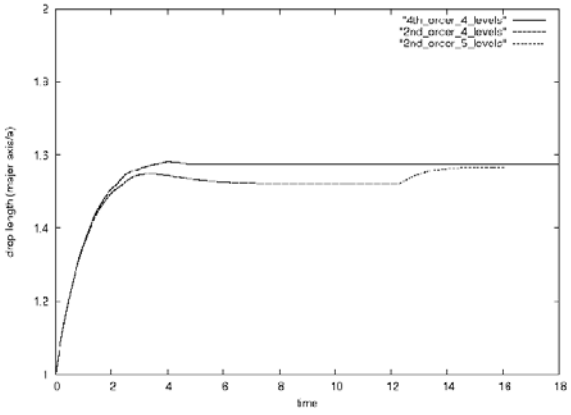

Figure 4: Graph of drop half-length divided by a vs. time. Coarse level has 64x64 grid cells. Drop half-length divided by a is 1.52 with 4 levels of AMR (effective fine grid 1024x1024). Drop half-length is 1.57 with 5 levels of AMR. Using 4th order curvature gives a drop half-length of 1.58 with 4 levels of AMR. $Ca = 0.5$, $Re = 1/100$. expected value is 1.583.

more robust and can handle problems with a very low Reynolds number.

As in Bhaga and Weber (1981) and Hnat and Buckmaster (1976), we present our computational results in terms of the following dimensionless groups. The Reynolds number R, the Eötvös number Eo, and the Morton number Mo are defined as follows

$$R = \frac{\rho L U}{\eta_L} \qquad Eo = \frac{g L^2 U}{\sigma} \qquad Mo = \frac{g \eta_L^4}{\rho \sigma^3}. \quad (18)$$

ρ is the liquid density, L is the bubble diameter, U is a characteristic velocity, η_L is the liquid viscosity, σ is the surface tension, and g is the acceleration of gravity. A comparison of computed terminal bubble shapes versus previous computational and experimental results are reported in Figures 5, 6, 7 and 8. Our comparisons include oblate ellipsoidal cap bubbles studied by Bhaga and Weber (1981) ($Eo = 243, Mo = 266$, and $R = 7.77$ for bubble figure 2(d) and $Eo = 116, Mo = 5.51$, and $R = 13.3$ for bubble figure 3(d)), spherical cap bubbles studied by Hnat and Buckmaster (1976) ($R = 19.38, Mo = 0.065$, and $C = 4.95$, where $C = \frac{r}{(\nu^2/g)^{1/3}}$), and a disk-bubble studied by Ryskin and Leal (1984) ($R = 100$ and $We = 10$).

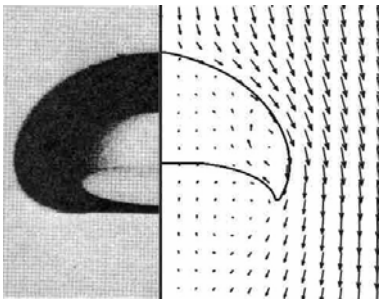

Figure 5: Comparison of our numerical results with experimental results found in paper by Bhaga and Weber (1981) (figure 2, bubble (d)).

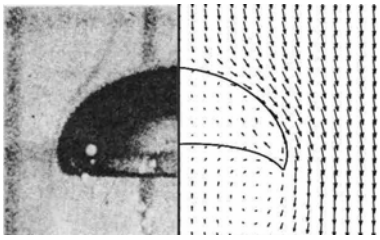

Figure 6: Comparison of our numerical results with experimental results found in paper by Bhaga and Weber (1981) (figure 3, bubble (d)).

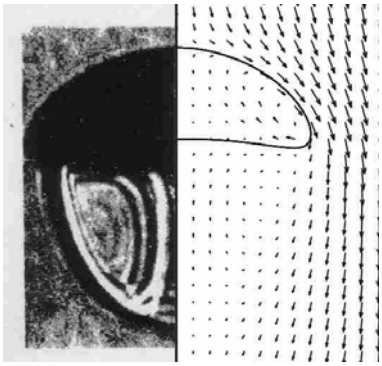

Figure 7: Comparison of our numerical results with experimental results found in paper by Hnat and Buckmaster (1976) where $Re = 19.38, Mo = 0.065$, and $C = 4.95$.

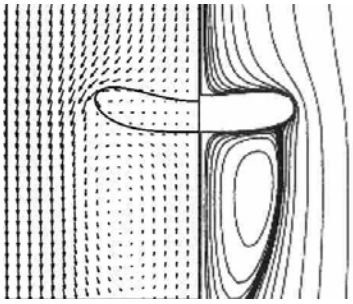

Figure 8: Comparison of our numerical results with computational results found in paper by Ryskin and Leal (1984) where $Re = 100$ and $We = 10$.

7 Conclusions

The "height fraction" approach for deriving curvature from volume fractions was extended from second order accurate to fourth order accurate. The improved accuracy was verified both analytically and through numerical tests. When applied to the "parasitic currents" test, there was a factor of 14 improvement of the fourth order method over the second order method. For the problem of droplet deformation under a shear flow, the 4th order method had 0.2% error whereas the 2nd order method gave a 4.0% error.

We also introduced a semi-implicit, sharp-interface, technique for treating the viscosity force terms on an adaptive hierarchy of grids. Our new algorithm allows us to solve problems which have strong viscous effects (a very small Reynolds number). Our computations compare very well to previous computations for the multi-phase problem of a rising gas bubble in liquid.

Acknowledgement: We thank Mr. D. Kikuchi and Mr. S. Yamaguchi for their help in preparing this manuscript. Work supported in part by NSF grant number 0242524 and by JSPS.

References

Aleinov, I.; Puckett, E. (1995): Computing surface tension with high-order kernels. In *Proceedings of the 6th International Symposium on Computational Fluid Dynamics*, Lake Tahoe, CA.

Bhaga, D.; Weber, M. (1981): Bubbles in viscous liquids: Shapes, wakes and velocities. *J. Fluid Mech.*, vol. 105, pp. 61–85.

Brackbill, J. U.; Kothe, D. B.; Zemach, C. (1992): A continuum method for modeling surface tension. *J. Comput. Phys.*, vol. 100, pp. 335–353.

Briggs, W. L.; Henson, V. E.; McCormick, S. (2000): *A Multigrid Tutorial, second edition*. SIAM, Philadelphia.

Chang, Y.; Hou, T.; Merriman, B.; Osher, S. (1996): Eulerian capturing methods based on a level set formulation for incompressible fluid interfaces. *J. Comput. Phys.*, vol. 124, pp. 449–464.

Chorin, A. J. (1985): Curvature and solidification. *J. Comput. Phys.*, vol. 57, pp. 472–490.

Cristini, V.; Renardy, Y. (2006): Scalings for droplet sizes in shear-driven breakup: Non-microfluidic ways to monodisperse emulsions. *FDMP: Fluid Dynamics & Materials Processing*, vol. 2(2), pp. 77–93.

Cummins, S.; Francois, M.; Kothe, D. (2005): Estimating curvature from volume fractions. *Computers and Structures*, vol. 83, pp. 425–434.

Enright, D.; Marschner, S.; Fedkiw, R. (2002): Animation and rendering of complex water surfaces. In *SIGGRAPH 2002*, volume ACM TOG 21, pp. 736–744.

Francois, M.; Cummins, S.; Dendy, E.; Kothe, D.; Sicilian, J.; Williams, M. (2006): A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *J. Comput. Phys.*, vol. 213(1), pp. 141–173.

Gibou, F.; Fedkiw, R.; Cheng, L.-T.; Kang, M. (2002): A second order accurate symmetric discretization of the poisson equation on irregular domains. *J. Comput. Phys.*, vol. 176, pp. 205–227.

Gueyffier, D.; Li, J.; Nadim, A.; Scardovelli, R.; Zaleski, S. (1999): Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *J. Comput. Phys.*, vol. 152, pp. 423–456.

Helmsen, J.; Colella, P.; Puckett, E. (1997): Non-convex profile evolution in two dimensions using volume of fluids. LBNL technical report LBNL-40693, Lawrence Berkeley National Laboratory, 1997.

Hnat, J.; Buckmaster, J. (1976): Spherical cap bubbles and skirt formation. *Physics of Fluids*, vol. 19 (2), pp. 182–194.

Jimenez, E.; Sussman, M.; Ohta, M. (2005): A computational study of bubble motion in newtonian and viscoelastic fluids. *FDMP: Fluid Dynamics & Materials Processing*, vol. 1(2), pp. 97–107.

Kang, M.; Fedkiw, R.; Liu, X.-D. (2000): A boundary condition capturing method for multiphase incompressible flow. *J. Sci. Comput.*, vol. 15, pp. 323–360.

Li, J.; Renardy, Y.; Renardy, M. (2000): Numerical simulation of breakup of a viscous drop in simple shear flow through a volume-of-fluid method. *Physics of Fluids*, vol. 12(2), pp. 269–282.

Liu, H.; Krishnan, S.; Marella, S.; Udaykumar, H. (2005): Sharp interface cartesian grid method ii: A technique for simulating droplet interactions with surfaces of arbitrary shape. *J. Computational Physics*, vol. 210, no. 1, pp. 32–54.

Nobari, M.; Jan, Y.; Tryggvason, G. (1993): Head on collision of drops; a numerical investigation. Technical Report ICOMP-93-45, NASA ICOMP Lewis Research Center, 1993.

Poo, J.; Ashgriz, N. (1989): A computational method for determining curvatures. *J. Comput. Phys.*, vol. 84, pp. 483–491.

Popinet, S.; Zaleski, S. (1999): A front-tracking algorithm for accurate representation of surface tension. *International Journal for Numerical Methods in Fluids*, vol. 30, no. 6, pp. 775–793.

Renardy, Y.; Renardy, M. (2002): Prost: A parabolic reconstruction of surface tension for the volume-of-fluid method. *J. Comput. Phys.*, vol. 183, no. 2, pp. 400–421.

Rudman, M. (1998): A volume tracking method for interfacial flows with large density variations. *Int. J. Numer. Methods Fluids*, vol. 28, pp. 357–378.

Ryskin, G.; Leal, L. (1984): Numerical solution of free boundary problems in fluid mechanics. part 2 buoyancy-driven motion of a gas bubble through a quiescent liquid. *J. Fluid Mech.*, vol. 148, pp. 19–35.

Sussman, M. (2003): A second order coupled levelset and volume of fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, vol. 187, pp. 110–136.

Sussman, M. (2005): A parallelized, adaptive algorithm for multiphase flows in general geometries. *Computers and Structures*, vol. 83, pp. 435–444.

Sussman, M.; Almgren, A.; Bell, J.; Colella, P.; Howell, L.; Welcome, M. (1999): An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.*, vol. 148, pp. 81–124.

Sussman, M.; Hussaini, M. (2003): A discontinuous spectral element method for the level set equation. *J. Scientific Computing*, vol. 19, pp. 479–500.

Sussman, M.; Puckett, E. (2000): A coupled level set and volume of fluid method for computing 3d and axisymmetric incompressible two-phase flows. *J. Comp. Phys.*, vol. 162, pp. 301–337.

Sussman, M.; Smereka, P.; Osher, S. (1994): A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, vol. 114, pp. 146–159.

Sussman, M.; Smith, K.; Hussaini, M.; Ohta, M.; Zhi-Wei, R. (2007): A sharp interface method for incompressible two-phase flows. *J. Comp. Phys.*, vol. 221(2), pp. 469–505.

Tatebe, O. (1993): The multigrid preconditioned conjugate gradient method. In *6th Copper Mountain Conference on Multigrid Methods*, Copper Mountain, CO.

Udaykumar, H.; Rao, M.; Shyy, W. (1996): Elafint - a mixed eulerian-lagrangian method for fluid flows with complex and moving boundaries. *Int. J. Numer. Meths. Fluids.*, vol. 22, no. 8, pp. 691–704.

Unverdi, S. O.; Tryggvason, G. (1992): A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, vol. 100, pp. 25–37.

van Leer, B. (1979): Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method. *J. Comput. Phys.*, vol. 32, pp. 101–136.

Williams, M.; Kothe, D.; Puckett, E. (1998): Convergence and accuracy of kernel-based continuum surface tension models. In *Proceedings of the Thirteenth U.S. National Congress of Applied Mechanics*, Gainesville, FL.

Ye, T.; Shyy, W.; Chung, J. (2001): A fixed-grid, sharp interface method for bubble dynamics and phase change. *J. Comp. Phys.*, vol. 174, pp. 781–815.

Zheng, X.; Lowengrub, J.; Anderson, A.; Cristini, V. (2005): Adaptive unstructured volume remeshing - ii: Application to two- and three-dimensional level-set simulations of multiphase flow. *Journal of Computational Physics*, vol. 208, pp. 626–650.