Conf-9509183--3

UCRL-JC-121224
PREPRINT

A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow II

Mark Sussman
Center for Computational Sciences & Engineering
P.O. Box 808, L-316
Lawrence Livermore National Laboratory
Livermore, CA 94551

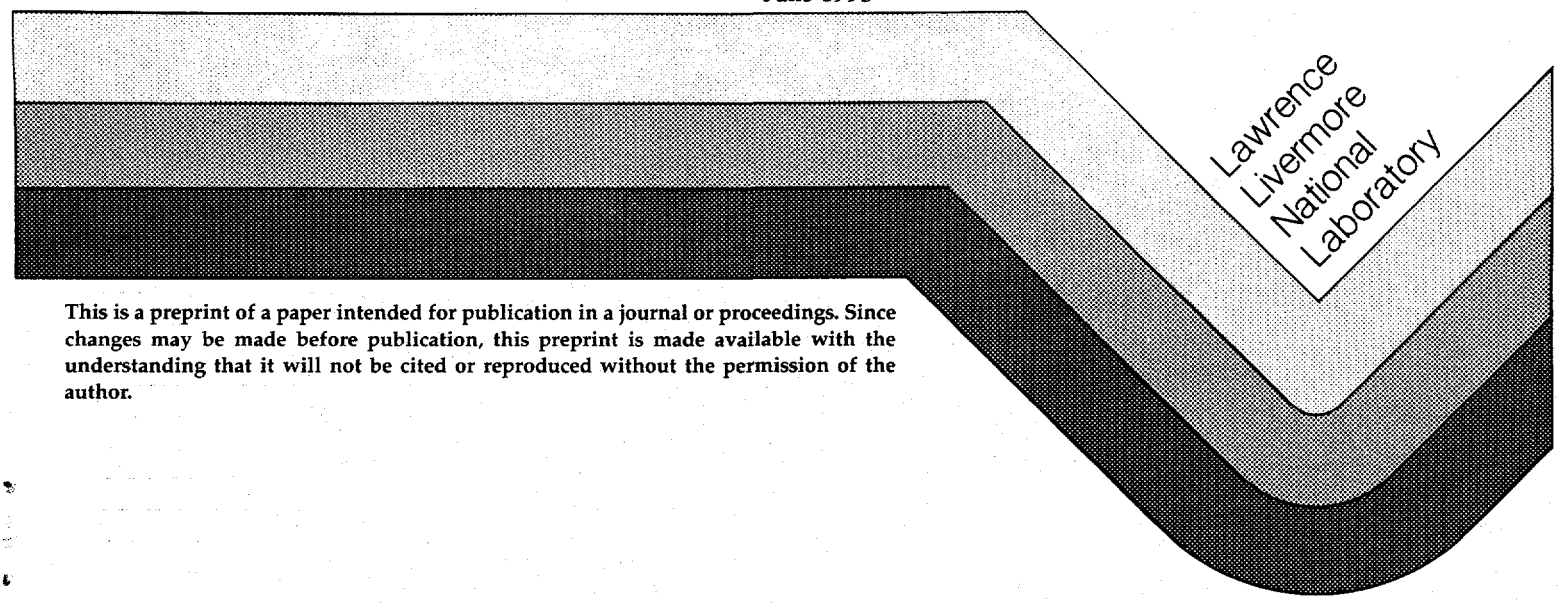
Emad Fatemi
University of California at Los Angeles
Department of Math
Los Angeles, CA 90024-1555

Peter Smereka
University of Michigan
Department of Math
Ann Arbor, MI 48109

Stanley Osher
University of California at Los Angeles
Department of Math
Los Angeles, CA 90024-1555

This paper was prepared for publication in the
Proceedings of the Sixth International Symposium on
Computational Fluid Dynamics
held September 4-8, 1995, Lake Tahoe, NV

June 1995



A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow II

Mark Sussman*, Emad Fatemi†, Peter Smereka‡, Stanley Osher§

June 28, 1995

Abstract

A level set method for capturing the interface between two fluids is combined with a variable density projection method to allow for computation of two-phase flow where the interface can merge/break and the flow can have a high Reynolds number. A distance function formulation of the level set method enables one to compute flows with large density ratios (1000/1) and flows that are surface tension driven; with no emotional involvement. Recent work has improved the accuracy of the distance function formulation and the accuracy of the advection scheme. We compute flows involving air bubbles and water drops, to name a few. We validate our code against experiments and theory.

1 Introduction

In [9] an Eulerian scheme was described for computing incompressible two-fluid flow where the density ratios across the interface is large (e.g. air/water) and surface tension,viscous effects are included. In this paper, we modify our scheme improving both the accuracy and efficiency of the algorithm. We use a level set function to "capture" the air/water interface thus allowing us to efficiently compute flows with complex interfacial structure. In [9], a unique iterative process was devised to allow us to compute with stiff surface tension effects and steep density gradients. We have since imposed a new "constraint" on the iterative process improving the accuracy of this process. We have also upgraded our scheme to using higher order ENO for spatial derivatives, and high order Runga Kutta for the time discretization (see [7]).

An example of the problems we wish to solve is illustrated in figure 1. A water drop splashes down against a pool of water creating a large crater. As the water recovers, surface tension effects cause a gas bubble to be entrained inside the water. It is a very difficult problem involving much interfacial complexity and stiff surface tension effects. The density ratio at the interface is about 1000/1. In [12], the boundary integral method was used to compute the drop splash problem and compare with experimental results. The boundary integral is a very good method for inviscid air/ water problems because, as a Lagrangian based scheme, only points on the interface need to be discretized. Unfortunately, if one wants to include the merging and breaking up of fluid mass, the boundary integral scheme can be very difficult to use. In our example, we automatically handle the actions of the drop hitting the water, and of the air bubble being completely submerged. We also compute flows involving wind over water (see figure 8), which necessitates an Eulerian scheme.

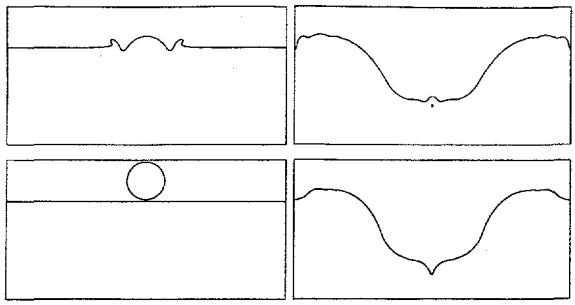
In order to compute flows with steep density ratios and surface tension, we give the interface a time independent thickness of only a few grid points wide. This is similar to what is done in [8], who also computed two fluid flow involving air and water. As shown in [9], a uniform thickness is crucial in accurately computing surface tension driven flows with steep density gradients. We show examples where we accurately compare with the boundary integral scheme for problems with 1000/1 density ratios and varying values of surface tension (see figure 3).

*Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48. Support under contract No. W-7405-Eng-48 was provided by the Applied Mathematical Sciences Program of the Office of Energy Research. Center for Computational Sciences and Engineering, LLNL, Livermore, CA 94550, Email: sussman@bigbird.llnl.gov

†work supported in part by NSF # DMS 94-04942, DARPA URI-ONR-N00014-92-J-1890. Dept. of Math, UCLA, Los Angeles, CA 90024-1555

‡Dept. of Math, Univ. of Michigan, Ann Arbor, MI 48109

§work supported in part by NSF # DMS 94-04942, DARPA URI-ONR-N00014-92-J-1890. Dept. of Math, UCLA, Los Angeles, CA 90024-1555



t=0.0 and t=1.6 t=24.0 and t=25.6

Figure 1: r=1.75mm, U=2m/s, 256x128 air bubble entrainment due to water drop hitting pool of water.

2 Numerical Formulation

2.1 Equations of Motion; level set formulation

In the work of [6, 15], the zero level set ($\{(x,y)/\phi(x,y)=0\}$) of a smooth function ϕ that is positive in one fluid and negative in the other was used to represent a sharp interface between two fluids. It was shown in [6] that the equation $\phi_t + \vec{u}\cdot\vec{\nabla}\phi$ accurately moves the zero level set according to the velocity field $\vec{u}$ even through the merging and breaking up of fluid mass.

As described in [9], there are many reasons to formulate the Navier-Stokes equations in the level set formulation. Computing spatial derivatives for ϕ, such as in the advection equation and for computing curvature, is more accurate than computing those values for a non-smooth function. Secondly, we maintain the level set function as a smooth distance function allowing us to give the interface a thickness fixed in time. Density and surface tension both depend on the level set function being a distance function.

We solve the following equations for incompressible flow including gravitational,viscous, and surface tension effects. The equations in dimensionless form are

$$H(\phi) \equiv \begin{cases} 1 & \text{if } \phi > 0 \\ 0 & \text{if } \phi < 0 \\ 1/2 & \text{if } \phi = 0 \end{cases} \tag{1}$$

$$\delta(\phi) = \partial H/\partial\phi \tag{2}$$

$$\kappa(\phi) = \vec{\nabla}\cdot\frac{\vec{\nabla}\phi}{|\vec{\nabla}\phi|} \tag{3}$$

$$\rho(\phi) = H(\phi) + (\rho_2/\rho_1)(1-H(\phi)) \tag{4}$$

$$\mu(\phi) = H(\phi) + (\mu_2/\mu_1)(1-H(\phi)) \tag{5}$$

$$\phi_t = -\vec{u}\cdot\vec{\nabla}\phi \tag{6}$$

$$\vec{u}_t = -\vec{u}\cdot\vec{\nabla}\vec{u} - \vec{\nabla}p/\rho + \frac{1}{Fr}\hat{y} + \frac{1}{R}\vec{\nabla}\cdot(2\mu D)/\rho + \frac{1}{W}\kappa(\phi)\vec{\nabla}H(\phi)/\rho \tag{7}$$

$$\vec{\nabla}\cdot\vec{u} = 0 \tag{8}$$

The dimensionless groups used are Reynolds number ($R = \frac{\rho_1 LU}{\mu_1}$), Froude number ($Fr = \frac{U^2}{gL}$), Weber number ($W = \frac{\rho_1 LU^2}{\sigma}$), and Bond number ($B = \frac{W}{Fr}$).

We will assume solid wall boundaries with the free-slip condition

$$\vec{u}\cdot\vec{n} = 0$$

2.2 Projection

In [1] a variable density projection method was described. We use a projection scheme coupled with high order upwind differencing of the convective terms in order to handle high Reynolds number flow. Briefly, we may write eqn (7) as:

$$\vec{u}_t + \vec{\nabla}p/\rho = L(\vec{u},\phi) \tag{9}$$

We use the fact that $\vec{u}_t$ is divergence free and hence for two-dimensional flow, we can write it as:

$$\vec{u}_t = \vec{\nabla}\times s_t$$

If we multiply both sides of eqn (9) by ρ and take the curl of both sides:

$$-\vec{\nabla}\rho\vec{\nabla}s_t = \vec{\nabla}\times L \tag{10}$$

The above equation eliminates pressure from eqn (7).

3 Discretization

We describe the actual numerical discretization of equations derived in the previous section. The outline of our scheme is as follows:

given ϕ_n, $\vec{u}_n$, defined at cell centers, we solve for ϕ_{n+1},$\vec{u}_{n+1}$.

1. We compute $L(\vec{u}_n, \phi_n)$ (see eqn (9)) and $\vec{u}\cdot\vec{\nabla}\phi$ using high order upwinded differencing for the convective terms and central differencing for the viscous and curvature terms (see [9]). A description of the upwind difference scheme will be presented below.
2. Solve eqn (10) for $\vec{u}_{tn}$. Equation 10 is solved using a MILU PCG scheme (details in [9]).
3. We advance in time using second and third order Runga Kutta methods found in [7]. The time step k is determined by CFL condition, viscous, and surface tension constraints (see [9]). Let F_{tn} represent $\vec{u}_{tn}$ and ϕ_{tn} and F_n represent $\vec{u}_n$ and ϕ_n. For second order RK we have:
$$\begin{aligned} \bar{F}_{n+1} &= F_n + kF_{tn} \\ F_{n+1} &= F_n + (k/2)(\bar{F}_{tn+1} + F_{tn}) \end{aligned}$$
4. We perform a "redistance" update on ϕ_{n+1}. Given $\phi^{(0)}_{n+1} \equiv \phi_{n+1}$ as initial data, we solve the equation $\phi_t = sign(\phi^{(0)}_{n+1})(1 - |\vec{\nabla}\phi|)$ for $t = 0$ to $t = \alpha h$ where αh is the thickness of our interface. The new solution $\phi^{(\alpha)}_{n+1}$ will represent the signed distance from the zero level set of $\phi^{(0)}_{n+1}$ for points within αh of the interface. We use a new constraint for improving the accuracy of the above operation; details are presented below.
5. We now let our new ϕ_{n+1} value be $\phi^{(\alpha)}_{n+1}$.

3.1 convective terms

In steps (1) and (2) above, one needs to compute $\vec{u}\cdot\vec{\nabla}\phi$ and $\vec{u}\cdot\vec{\nabla}\vec{u}$. We discretize them as:

$$\begin{aligned} \vec{u}\cdot\vec{\nabla}\phi &= u_{ij}(\phi_{i+1/2j} - \phi_{i-1/2j}) + v_{ij}(\phi_{ij+1/2} - \phi_{ij-1/2}) \\ \vec{u}\cdot\vec{\nabla}\vec{u} &= u_{ij}(\vec{u}_{i+1/2j} - \vec{u}_{i-1/2j}) + v_{ij}(\vec{u}_{ij+1/2} - \vec{u}_{ij-1/2}) \end{aligned}$$

As a note, the above equations represent the result of subtracting off the divergence free part from the conservative formulation. Since we use the stream function formulation for the projection (see [9]) we can write u_{ij} and v_{ij} as

$$\begin{aligned} u_{ij} &= (\tilde{u}_{i+1/2j} + \tilde{u}_{i-1/2j})/2 \\ v_{ij} &= (\tilde{v}_{ij+1/2} + \tilde{v}_{ij-1/2})/2 \\ \tilde{u}_{i+1/2j} &\equiv (s_{i+1/2j+1/2} - s_{i+1/2j-1/2})/h \\ \tilde{v}_{ij+1/2} &\equiv -(s_{i+1/2j+1/2} - s_{i-1/2j+1/2})/h \end{aligned}$$

The edge values $\phi_{i+1/2j}$, $\vec{u}_{i+1/2j}$, ...are computed using a high order ENO procedure derived in [7].

3.2 re-distancing operation

As mentioned earlier, we give our interface a mesh dependent thickness in order to compute flows with 1000/1 density ratios and surface tension effects. We incorporate our thickness through the definition of the heaviside function:

$$\begin{aligned} H_\epsilon(\phi) &\equiv \begin{cases} 1 & \phi > \epsilon \\ 0 & \phi < -\epsilon \\ \frac{1}{2}(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin(\pi\phi/\epsilon)) & |\phi| \le \epsilon \end{cases} \\ sign_\epsilon(\phi) &\equiv 2(H_\epsilon(\phi) - 1/2) \end{aligned}$$

For ease of notation, in this section we will denote $\phi^{(0)}_{n+1}$ as ϕ_0, $\phi^{(k)}_{n+1}$ as ϕ_k, $\phi^{(k+1)}_{n+1}$ as ϕ_{k+1}, and so on.
In [9], we used an iteration procedure using the equation

$$\phi_t = L(\phi_0, \phi) = sign(\phi_0)(1 - |\vec{\nabla}\phi|) \tag{11}$$

for maintaining ϕ as the signed distance from the interface; therefore preserving the interfacial thickness.

The above equation does not change the position of the zero level set of ϕ_0, unfortunately in numerical computation this may not be true. We use the fact that

$$\partial_t \int_{\Omega_{ij}} H(\phi) = 0 \tag{12}$$

In every grid cell $\Omega_{ij} = ((x,y)/x_i - 1/2 < x < x_i + 1/2 \text{and} y_i - 1/2 < y < y_i + 1/2)$. That is, since the interface should not move, the volume should not change either. We modify equation 11:

$$\phi_t = L(\phi_0, \phi) + \lambda_{ij} f(\phi) \tag{13}$$

λ_{ij} is *constant* in each cell Ω_{ij} determined using,

$$\partial_t \int_{\Omega_{ij}} H(\phi) = \int_{\Omega_{ij}} H'(\phi)\phi_t = \int_{\Omega_{ij}} H'(\phi)(L(\phi_0, \phi) + \lambda_{ij} f(\phi)) = 0$$

λ_{ij} is calculated to be

$$\lambda_{ij} = \frac{-\int_{\Omega_{ij}} H'(\phi)L(\phi)}{\int_{\Omega_{ij}} H'(\phi)f(\phi)} \tag{14}$$

In our calculations we choose

$$f(\phi) \equiv H'(\phi)|\vec{\nabla}\phi|.$$

This insures that we only correct at the interface without disturbing the distance function property away from the interface.

3.3 Discretization of re-distancing operation

We may put eqn (11) in the form

$$\phi_t + \vec{w} \cdot \nabla\phi = sign(\phi_0) \tag{15}$$

$$\vec{w} = (\nabla\phi / \mid \nabla\phi \mid) sign(\phi_0) \tag{16}$$

Equation 15 has the form of an advection equation with speed $\vec{w}$ so we use high order upwinded ENO type schemes (see [7, 17]) to approximate the spatial derivatives. We use second order Runga-Kutta for discretizing (15) in time. The discretization is a higher order generalization of a first order scheme presented in [5] for the case $\phi \geq 0$.

We now describe how the constraint is applied. Given ϕ_n, we compute $\tilde{\phi}_{n+1}$ as described above. We then compute λ_{ij} (see eqn (14)) where:

$$\begin{aligned} L(\phi) &\approx (\tilde{\phi}_{n+1} - \phi_0)/(t_{n+1} - t_0) \\ H'(\phi) &\approx \partial H_h(\phi_0)/\partial\phi \end{aligned}$$

Our new updated ϕ_{n+1} is

$$\phi_{n+1} = \tilde{\phi}_{n+1} + (t_{n+1} - t_0)\lambda_{ij} f(\phi_0)$$

If ϕ_0 is close to a distance function, we have

$$\begin{aligned} \int_{\Omega_{ij}} H_h(\phi_{n+1}) - H_h(\phi_n) &= \int_{\Omega_{ij}} H_h'(\phi_0)(\phi_{n+1} - \phi_0) + \int_{\Omega_{ij}} H_h''(\phi_0)(\phi_{n+1} - \phi_0)^2/2 + \ldots = \\ &\int_{\Omega_{ij}} H_h''(\phi_0)(\phi_{n+1} - \phi_0)^2/2 + \ldots = O(h^4) \end{aligned}$$

In all our problems, we use a distance-iterate time step of $\frac{h}{2}$ which satisfies the CFL condition (since $|\vec{w}| \leq 1$). We only need to iterate 2α times if the interfacial thickness is αh. This is apparent because the characteristics (see eqn (15)) move away from the interface with speed one; so ϕ will be "corrected" at a distance d from the interface after time $t = d$. We are using a formally second order accurate re-distance algorithm along with our new constraint, so we expect $O(h^2)$ accuracy. For the drop collision problem (see figure 4), we see about three times improvement in mass conservation in comparison with the results of the same exact problem in figure 23 of [9].

4 Translating circle and Zalesak's problem

These examples will show the effectiveness of our advection scheme. The velocity field is pre-specified, so that only equations for ϕ are solved. An advantage of the level set scheme for advection of sharp interfaces is the fact that one can use arbitrarily high order schemes for solving equations 6 and 15. We compute the solution for a translating circle in a 4×4 periodic box:

$$u_0(x,y) = v_0(x,y) = 1 \tag{17}$$

$$\phi_0(x,y) = \sqrt{x^2 + y^2} - 1 \tag{18}$$

Δx	error	order
1/4	1.01E-3	N/A
1/8	2.08E-4	2.3
1/16	2.41E-5	3.1

Table 1: Convergence study: Diagonal translation of circle

Δx	error	order
1	1.267E-1	N/A
1/2	3.894E-2	1.7

Table 2: Convergence study: Rotation of cutout circle (Zalesak's problem)

We discretize equation 6 using fifth order ENO (see [7]) and third order Runga Kutta. The redistancing algorithm is discretized using third order ENO for the spatial derivatives and second order Runga Kutta. We run the above problem up to $t = 4$ and then measure the error. The error is measured as

$$\int_{\Omega} |H(\phi_{real}) - H(\phi)|/L \tag{19}$$

where L is the perimeter size of the initial interface. In table 1 we measured up to third order accuracy.

We now test our advection scheme for computing "Zalesak's problem" (see [16]). The domain size is 100×100 and it contains a slotted circle centered at (50,75) with slot width 15. We initialize $\vec{u}$ and ϕ as follows:

$$\begin{aligned} u_0 &= (\pi/314)(50 - y) \\ v_0 &= (\pi/314)(x - 50) \\ \phi_0 &= \text{signed distance from object} \end{aligned}$$

We compute for $t = 0$ to $t = 628$ (one full revolution) on a 100×100 grid; the same as that used in [16]. We then refine the grid in order to measure accuracy. In both cases, the time step is equal to Δx. We get an order of accuracy of 1.7 (see table 2) which is very good considering the sharp corners in the initial data. We overlayed the coarse grid results with the expected solution in figure 2. The maximum mass fluctuation was less than 0.7 percent on the coarse grid and less than 0.2 percent on the fine grid.

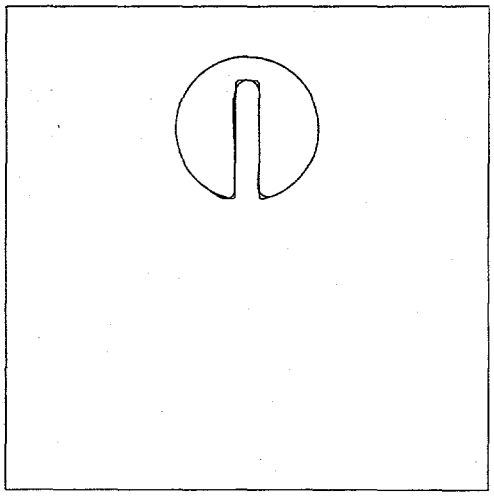


Figure 2: Zalesak's problem, $\Delta x = \Delta t = 1.0$

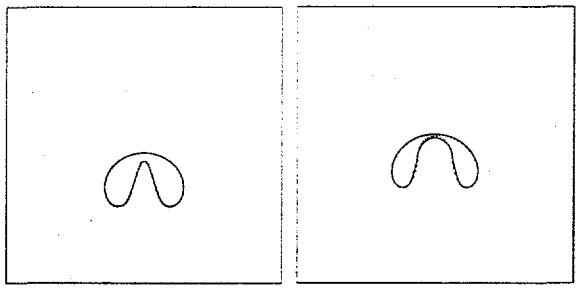
time	v/U	expected	aspect ratio	expected
5.0	0.98	1.00	2.7	2.7
6.0	0.98	1.00	2.8	2.7
7.0	0.98	1.00	2.8	2.7

Table 3: Viscous gas bubble: comparison with experiment

5 2d and 3d axisymmetric air/water flow

5.1 Bubble and Drop problems

The following 3d axisymmetric tests demonstrate our ability to accurately handle flows with steep density ratios (1/0.001225) and large surface tension effects. We have modified our 2D code in a similar manner as done in [1, 10] for handling 3d axisymmetric problems. In figure 3, we display rising gas bubbles with infinite Reynolds number and varying surface tension. We agree very closely to our boundary integral scheme and the tests run in [3].



t=1.2 Bond number 200 t=1.8 Bond number 10

Figure 3: $Re = \infty$ Rising 3d air bubble 120x240. dotted lines: boundary integral method

In figure 5, we display a rising gas bubble (density ratio 1/0.001225 viscosity ratio 1/0.01) that reaches a steady speed/shape due to viscous and surface tension forces. We use far-field boundary conditions in our computations in order to circumvent wall effects. In table 3, we show that our results agree very closely with the experiments of [13] (figure 1A). They had a dimensionless rise speed of 1.0 and an aspect ratio ($\frac{\text{major axis}}{\text{minor axis}}$) of 2.7. Our results also match closely to the computations of [11] (figure 6).

In order to illustrate our ability to compute with surface tension, we compute 3d axisymmetric zero gravity drop dynamics and compare with the low amplitude linearized drop oscillation solutions of [2] (ch. 275,355). We also compute large amplitude solutions and compare with [4]. In figure 6, we display $r(\theta = 0)$ of an initial "2-mode" perturbation and compare with the expected linearized viscous effects. The computed dimensionless period is 3.18 and the expected period is π. In figure 7, we display the evolution of a drop when given a large amplitude "4-mode" perturbation. The results agree very closely with [4] (figure 6).

5.2 Breaking Waves

There has been recent work done in trying to compute wave growth with wind. [1] We have done many preliminary tests including standing wave calculations and stokes wave computations. In figure 8, we see the effects of wind being blown over a 2d large amplitude stokes wave (see [14]) causing the wave break. Without the wind, in the moving frame of reference, the wave will maintain the same shape. In figure 9, we show good agreement with expected viscous effects for a low-amplitude standing wave (see [2] ch. 348).

6 Conclusion

We have presented a robust scheme for handling 2d or 3d axisymmetric incompressible air/water flow. As a result of a new "constraint" in our re-distancing scheme, we see improved accuracy. We have done basic tests demonstrating the accuracy of the scheme and also tests validating the effects of surface tension and viscosity. We have shown many problems with density ratios of 1/0.001225 along with stiff surface tension effects, with good agreement with expected results. In the future, we would like to be able to improve the resolution of the scheme through adaptive

[1] Work done with Dan Marcus and Dave Chambers

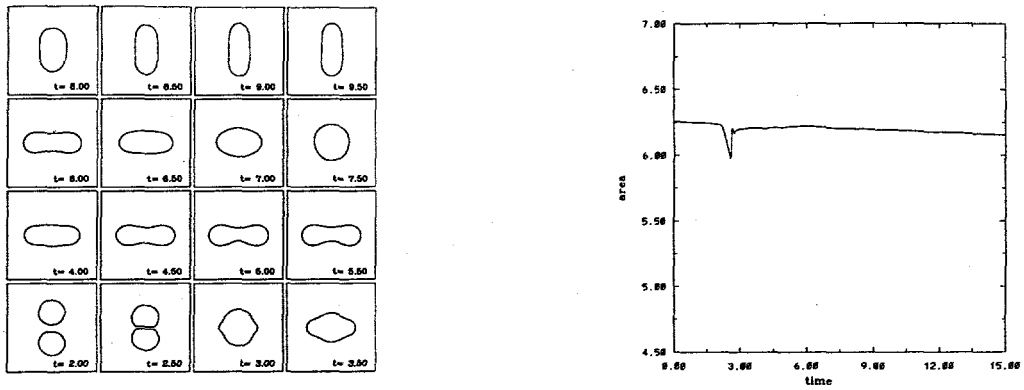


Figure 4: Re=20 B=2.0 density 1/14 44x44 two-dimensional drop collision

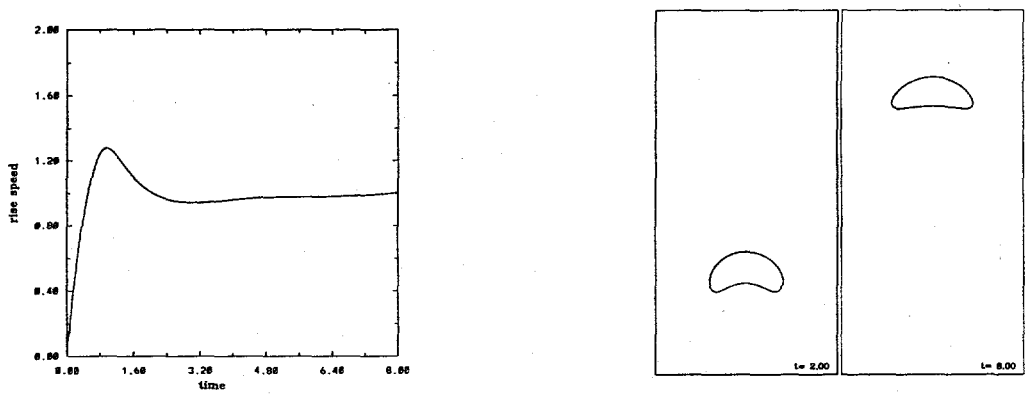


Figure 5: $Re = 9.8$ $We = 9.6$ $Cd = 3.44$ 32x128 steady rise of 3d air bubble

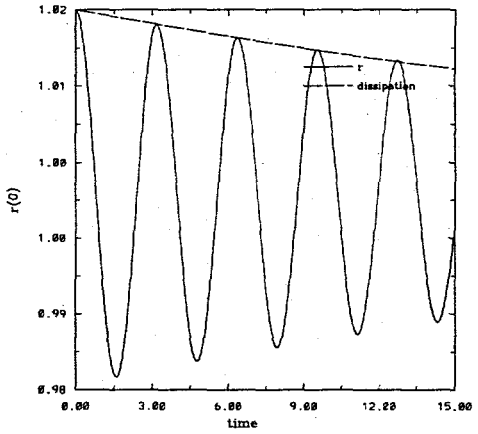


Figure 6: Mode=2, small amplitude oscillations of water drop due to surface tension, 3x3 domain, 50x100 grid $\epsilon = 0.02$ B=2.0

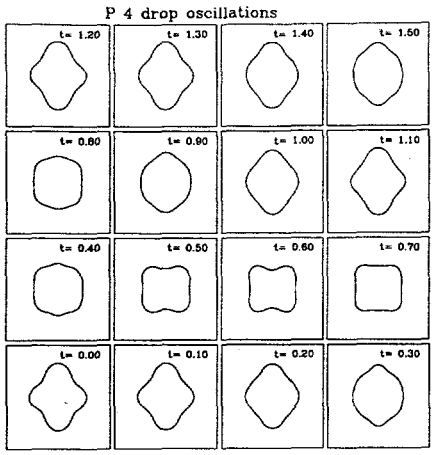
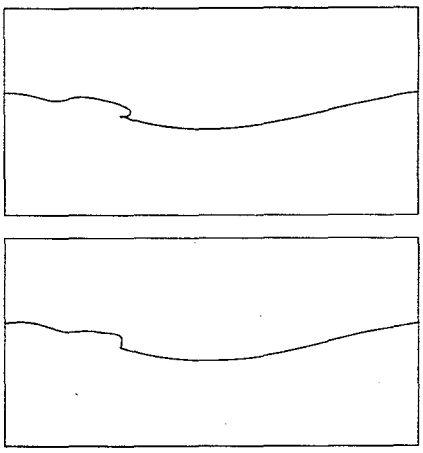


Figure 7: Mode=4, oscillating water drop due to surface tension, 4x4 domain, 64x128 grid $\epsilon = 0.3$ B=2.0



t=1.80 and t=1.95

Figure 8: 10m breaking wave, 3.4m/s wind initial slope=0.4 128x64

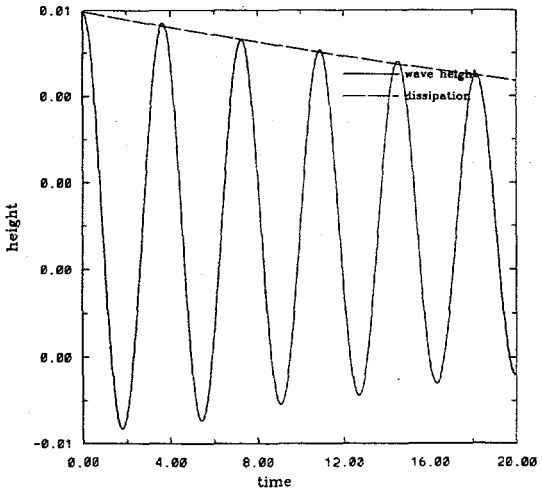


Figure 9: standing wave, 1.0x1.0 domain, 50x100 grid, $Re = 1000$, $\epsilon = 0.008$

mesh technology; thus enabling the simulation of fine scale behavior such as growth of the various modes of a wind driven wave.

References

[1] J.B. Bell and D.L. Marcus, J. Comp. Phys., 101, pp. 334-348, (1992).

[2] Lamb, H., *Hydrodynamics*, Dover Publications, 1945.

[3] Lundgren, T.S. and Mansour, N.N., J. Fluid Mech., 224, 177 (1991).

[4] Lundgren, T.S. & Mansour, N.N., J. Fluid Mech., 194, 479-510.

[5] Rouy, E. and Tourin, A., SIAM J. Numer. Anal., Vol. 29, No. 3, pp. 867-884, June 1992.

[6] Osher, S. and Sethian, J.A., J. Comp. Phys., 79,1, pp. 12-49, (1988).

[7] Shu, C.W. and Osher,S., J. Comp. Phys., 83, pp. 32-78, (1989).

[8] Unverdi, S.O. and Tryggvason, G., J. Comp. Phys., 100, pp. 25-37, (1992).

[9] Sussman, M., Smereka, P., & Osher, S.J., J. Comp. Phys., 114, pp. 146-159.

[10] Sussman, M., UCLA, Ph.D. thesis, June 1994.

[11] Ryskin, G. & Leal, L.G., J. Fluid Mech., 148, 19-35.

[12] Oguz, H.N & Prosperetti, A., J. Fluid Mech., 203, 143-179.

[13] Hnat, J.G.& Buckmaster, J.D., Phys. Fluids, 19, 182-194.

[14] Rienecker,M.M., and Fenton,J.D., J. Fluid Mech. (1981), 104, pp. 119-137.

[15] Mulder, W., Osher, S., and Sethian, J.A., J. Comp. Phys., 100, 209 (1992).

[16] Zalesak, S.T., J. Comp. Phys., 31, 335-362 (1979).

[17] Harten, A., J. Comp. Phys., 83, 148-184 (1989).