A MONOLITHIC MASS TRACKING FORMULATION FOR BUBBLES IN INCOMPRESSIBLE FLOW

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Mridul Aanjaneya
May 2013

This dissertation is online at: http://purl.stanford.edu/pk959jq5605

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Ronald Fedkiw, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Patrick Hanrahan

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Radek Grzeszczuk

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

This dissertation presents a novel method for treating bubbles in free surface incompressible flow that relies on the conservative advection of bubble mass and an associated equation of state in order to determine pressure boundary conditions inside each bubble. It is shown that executing this algorithm in a traditional manner leads to stability issues similar to those seen for partitioned methods for solid-fluid coupling. Therefore, the problem is reformulated monolithically. This is accomplished by first proposing a new fully monolithic approach to coupling incompressible flow to fully nonlinear compressible flow including the effects of shocks and rarefactions, and then subsequently making a number of simplifying assumptions on the air flow removing not only the nonlinearities but also the spatial variations of both the density and the pressure. The resulting algorithm is quite robust, has been shown to converge to known solutions for test problems, and has been shown to be quite effective on more realistic problems including those with multiple bubbles, merging and pinching, etc. Notably, this approach departs from a standard two-phase incompressible flow model where the air flow preserves its volume despite potentially large forces and pressure differentials in the surrounding incompressible fluid that should change its volume. The proposed method allows bubbles to readily change volume according to an isothermal equation of state.

This method is then extended to model both large and small scale bubble dynamics. Small under-resolved bubbles are evolved using Lagrangian particles that are monolithically two-way coupled to the surrounding flow in a manner that closely approximates the analytic bubble oscillation frequency while converging to the analytic

volume as predicted by the well-known Rayleigh-Plesset equation. A novel scheme is presented for interconverting between these under-resolved Lagrangian bubbles and the larger well-resolved Eulerian bubbles. A novel seeding mechanism is also presented to realistically generate bubbles when simulating fluid structure interaction with complex objects such as ship propellers. The proposed framework for bubble generation is general enough to be incorporated into all grid-based as well as particle-based fluid simulation methods.

Acknowledgments

I am deeply grateful to my advisor, Ron Fedkiw, who has had a profound impact on my attitude towards research as well as life. Quite frankly, I view him as *'Perfection Incarnate'*! I have learnt a lot from him about the traits in a good researcher and a good person during our several interactions in the last few years. He has taught me the values of sacrifice and hard work; that to make a difference one should choose 'the road less traveled', and that every individual is capable of achieving great feats provided he/she has a strong will to do so. I am especially proud of what I could accomplish under his guidance and mentorship.

The rest of Ron's research group has also played a large part in making the last few years both fun and educational. I would like to thank my co-authors Saket Patkar, Dmitriy Karpman and Michael Lentine. I would especially like to thank Saket Patkar with whom I worked very closely in the last year and also enjoyed a lot. I would also like to thank other members in Ron's group: Nipun Kwatra, Jón Grétarsson, Jonathan Su, Craig Schroeder, Avi Robinson-Mosher, Elliot English, Wen Zheng, Matthew Cong, Linhai Qiu, Rahul Sheth, Bo Zhu, Yue Yu, Like Gobeawan and Wenlong Lu. Special thanks to Nipun Kwatra who is like an elder brother to me and has been a constant source of encouragement, support and motivation.

During my stay at Stanford, I also had the privilege of working at the Nokia Research Center. I would like to thank the following people: Radek Grzeszczuk, Timo Pylvänäinen, Ramakrishna Vedantham, Jérôme Berclaz, Thommen Korah, Varsha Hedau, Vasu Parameswaran, Gabriel Takacs, Vijay Chandrasekhar, Sam Tsai and

David Chen. I am grateful to all of them for making this collaboration enjoyable and productive as well as teaching me the values of team spirit. I would especially like to thank Radek Grzeszczuk and Timo Pylvänäinen for closely mentoring my work at Nokia and helping me develop a deeper understanding of geometry and vision.

I would like to thank my reading committee members, Pat Hanrahan and Radek Grzeszczuk as well as my other committee members, Gianluca Iaccarino and Kunle Olukotun for their time and effort.

I am also grateful to Leo Guibas for initially recruiting me to Stanford. I have learnt a lot about geometry and topology by working and interacting with several members of his group. Thanks to Dmitriy Morozov, Kyle Heath, Daniel Chen, Niloy Mitra, Maks Ovsjanikov, Jian Sun, Mirela Ben-Chen, Adrian Butscher, Nikola Milosavljevic, Natasha Gelfand, Qixing Huang, Omprakash Gnawali, Ian Downes, HyungJune Lee, Emilio Antunez, Xiaoye Jiang, Arik Motskin and Justin Solomon. I would also like to thank Frédéric Chazal and Marc Glisse from INRIA Saclay - Île-de-France.

I would like to thank my undergraduate advisors, Sudebkumar Prasant Pal and Arijit Bishnu, for encouraging me to pursue graduate studies and constantly boosting my morale during ups and downs of graduate life. The exposure I gained has certainly been an eye opener both in terms of new knowledge as well as my outlook on life.

I would like to invoke the blessings of my father who, though no more, is responsible for instilling the 'scientific temper' in me. I am grateful to my elder brother who has always been a shining example before me, always ready with his valuable and apt suggestions. This work would not have been possible without the constant love, encouragement, moral support and guidance of my mother. I am grateful to her for all the sacrifices she made to help me come this far. This thesis is dedicated to her.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Bubbles are ubiquitous in most underwater scenes and embellish the otherwise lifeless water providing visual cues to the virtually imperceptible velocity field. They also provide a mechanism for sound generation [62, 48]. These sounds are a consequence of the volumetric changes that bubbles undergo during their temporal evolution - volumetric changes which can be substantial when bubbles rise a significant distance, or when fast moving objects such as ship propellers interact with water. In such fluid structure interactions, lower pressure regions are generated near the objects causing some of the water to instantly vaporize through the compressible phenomena of cavitation [5]. Since the density of water is a thousand times larger than that of air, the vaporized water forms bubbles that quickly expand in size becoming visible to the naked eye. Thus, to realistically simulate both bubble sounds and dynamics, it is important to design numerical methods that allow bubbles to change in volume - contrary to the traditional approach of treating the air inside bubbles as incompressible flow, e.g. [4, 60, 58, 28, 26, 47, 44, 63, 3].

Although a completely compressible treatment of this problem is possible, it is computationally expensive since water has a very high density compared to that of air and the larger sound speed in water imposes a very strict time step restriction on both phases based on the typical CFL condition. On the other hand, the air phase

is neglected when simulating free surface incompressible flows under the assumption that the inertia of air is very small compared to that of water. Hence, such solvers are unsuitable for modeling effects such as air entrainment and bubble dynamics. To prevent the air bubbles from collapsing unnaturally, methods have been proposed for computing a pressure inside these bubbles by tracking their volume over time and using an equation of state [56]. But volume can change radically if a bubble rises a long distance, merges with other nearby bubbles, or breaks up into smaller bubbles.

Several Lagrangian methods have also been proposed for modeling bubble dynamics [20, 49, 9, 59, 35, 6, 36, 30]. An early approach to particle-based bubble simulation was proposed by [20] who generated bubbles based on escaped particles from the particle level set method of [13] similar to the approach for spray in [16] (see also [18]). Later, a number of authors proposed additional Lagrangian techniques including [35, 6, 36, 30]. Although [35] did propose using a variable density Poisson solver for approximating the average bubble motion, this only gives very limited two-way interactions and ignores changes in the bubble's volume.

Instead of representing each bubble as a single particle, one can use a collection of particles to model a single bubble. SPH-based methods are good candidates for such an approach [49, 9, 59]. This can aid in modeling a bigger range of bubble dynamics obtaining a wider variety of topological shapes that real bubbles exhibit. However, fully Eulerian grid-based methods still seem preferable for these larger bubbles. Moreover, some of the most compelling methods for simulating fluids tie together multiple scales as can be seen in [45, 29, 39, 46].

This dissertation proposes a mass tracking formulation for simulating bubble dynamics based on the observation that the total mass of air inside water is conserved over time (ignoring mass exchange across the interface such as vaporization). This avoids the computational difficulties associated with tracking the volume when bubbles undergo complex topological deformations during merging or splitting. The method is devised by first proposing a rather straightforward approach based on mass tracking in Chapter 2. This approach suffers from stability issues which have characteristics

similar to partitioned (as opposed to monolithic) methods for solid-fluid coupling, see for example [22, 52, 21] and the references therein. These issues can be alleviated using outer iterations on the partitioned solver, as discussed in Chapter 3, although this can require ten or more Poisson solves per time step and is thus computationally expensive. Instead, we take a more traditional monolithic approach for the air-water problem similar to the solid-fluid coupling in [22, 52, 21] as motivated by [38]. We begin by revisiting the partitioned solver for incompressible and compressible flow from [7] and devise a monolithic solver using the ideas from [38] to couple together incompressible flow with fully nonlinear compressible flow including shocks and rarefactions. The results of this method are shown in Chapter 4 for both the gamma gas law and an isothermal equation of state. Then in Chapter 5 we simplify this approach greatly so that it is in line with the straightforward bubble simulation method of Chapter 2. This is achieved by setting both the bubble density and the bubble pressure to be spatially constant, although time-varying. Chapter 7 presents a detailed summary of the fully Eulerian approach developed so far referring to the appropriate equations throughout the text and highlights its efficacy with various examples - including the ability to treat multiple bubbles which may also split and merge. Since the level set loses volume and cannot keep track of sub-grid scale details, Chapter 8 proposes to track these under-resolved bubbles using Lagrangian particles that are also monolithically two-way coupled to the surrounding fluid. To seamlessly transition between these smaller Lagrangian bubbles and larger well-resolved Eulerian bubbles, Chapter 9 proposes a novel scheme for interconverting between the two representations. Chapter 10 proposes a novel seeding mechanism to realistically generate bubbles when simulating fluid structure interaction with complex objects such as ship propellers. The proposed framework for bubble generation is general enough to be incorporated into all grid-based as well as particle-based fluid simulation methods. Finally, we conclude in Chapter 11 with some interesting avenues for future work.

Chapter 2

A partitioned approach

We begin with a straightforward approach that conservatively advects the bubble mass and uses the isothermal equation of state $p = B\rho$ to compute a pressure inside the bubble which is subsequently used as a Dirichlet boundary condition for making the incompressible velocities divergence free. As we will see, this approach lacks stability properties because the current pressure in the bubble is not a good predictor of what the pressure would be in the next time step after applying the incompressible flow pressure to generate a new velocity field that changes the size of the bubble. If one couples the air pressure as a degree of freedom instead of as a Dirichlet boundary condition, then the incompressible Poisson solver can better react to the anticipated changes in the bubble volume. Thus, after presenting our straightforward approach in this section (which we call partitioned), we present a monolithic solver in Chapter 4.

2.1 Incompressible flow

The incompressible Navier-Stokes equations are given by

$$\vec{v}_t + (\vec{v} \cdot \nabla)\vec{v} + \frac{\nabla p}{\rho_I} = \frac{\nabla \cdot (\mu \nabla \vec{v})}{\rho_I} + \vec{f} \tag{2.1}$$

$$\nabla \cdot \vec{v} = 0 \tag{2.2}$$

where ρ_I is the density, $\vec{v}$ is the velocity, μ is the coefficient of viscosity and $\vec{f}$ is the net body force acting on the incompressible fluid. These equations are discretized on a MAC grid using the projection method [8], where we first explicitly update

$$\frac{\vec{v}^\star - \vec{v}^n}{\Delta t} + (\vec{v} \cdot \nabla)\vec{v} = \frac{\nabla \cdot (\mu \nabla \vec{v})}{\rho_I} + \vec{f} \tag{2.3}$$

and then solve for the pressure via

$$\nabla \cdot \frac{\nabla p}{\rho_I} = \frac{\nabla \cdot \vec{v}^\star}{\Delta t} \tag{2.4}$$

in order to update the intermediate velocity $\vec{v}^\star$ as follows

$$\frac{\vec{v}^{n+1} - \vec{v}^\star}{\Delta t} + \frac{\nabla p}{\rho_I} = 0 \tag{2.5}$$

We use the level set method [50] to track the interface between the bubbles and the incompressible fluid. Before updating the incompressible velocities through advection, they are extrapolated across the interface in order to define ghost node values. This could be accomplished using constant extrapolation normal to the interface by solving the equation $I_\tau + \vec{N} \cdot \nabla I = 0$, in fictitious time τ for each component I of $\vec{v}$. We instead compute the steady state solution using the fast extension method of [1]. The incompressible velocities are then advected using semi-Lagrangian advection which can be made second order accurate using a MacCormack-style method as in [53]. The level set function ϕ is advected using the particle level set method of [11] and the semi-Lagrangian advection scheme of [12]. To keep the level set a signed distance function we use the modified fast marching method proposed in [43].

The treatment of viscosity for multiphase incompressible flow with appropriate jump conditions at the interface is discussed in [33]. However, viscosity is solved for explicitly in [33] which has a severe time step restriction of $\Delta t \propto O(\Delta x^2)$. In order to take large time steps, we consider an implicit treatment of viscosity. As discussed in [27], if all the jump conditions are treated implicitly then the equations for all components of the velocity are coupled together. Although one could take an approach similar

to [51] for spatially varying viscosity where the coupling terms are treated explicitly and other terms are treated implicitly in order to get decoupling of various components, there can be some time step restrictions based on the jump conditions. For the simulation of bubbles, we assume that the dynamics inside the bubbles contain little momentum, hence, they cannot absorb any viscous momentum from the liquid. Thus, we enforce Neumann boundary conditions at the interface that the derivative of each component of the incompressible velocity is zero. Thus, the jump in pressure due to viscosity is also zero since the normal component of the viscous stress vanishes across the interface. Finally, as we assume constant viscosity in the incompressible fluid, the equations for the different components of the incompressible velocity decouple as well. In two spatial dimensions, equation (2.3) can be written component-wise as

$$\frac{v_1^\star - v_1^n}{\Delta t} + (\vec{v} \cdot \nabla) v_1 = \frac{\nabla \cdot (\mu \nabla v_1^\star)}{\rho_I} + f_1 \tag{2.6}$$

$$\frac{v_2^\star - v_2^n}{\Delta t} + (\vec{v} \cdot \nabla) v_2 = \frac{\nabla \cdot (\mu \nabla v_2^\star)}{\rho_I} + f_2 \tag{2.7}$$

The advection and external forces can be applied first to obtain $\hat{v}_1$ and $\hat{v}_2$ followed by a viscous solve of the form

$$\frac{v_1^\star - \hat{v}_1}{\Delta t} = \frac{\nabla \cdot (\mu \nabla v_1^\star)}{\rho_I} \tag{2.8}$$

$$\frac{v_2^\star - \hat{v}_2}{\Delta t} = \frac{\nabla \cdot (\mu \nabla v_2^\star)}{\rho_I} \tag{2.9}$$

Since $\hat{v}_1$ and $\hat{v}_2$ are not divergence free and the viscous update equations have been derived assuming the divergence free condition, $\hat{v}_1$ and $\hat{v}_2$ are sometimes first projected to be divergence free before applying viscosity. However, note that the pressure projection is not idempotent in the presence of a non-zero pressure gradient across the incompressible fluid. In this case we do not project $\hat{v}_1$ and $\hat{v}_2$ to be divergence free before the viscous update. Note that the advection terms are computed in a thin band of ghost cells so that there are adequate values when the interface moves, however, the viscous terms can only be updated interior to the level set due to the need to prescribe interface boundary conditions. Therefore, the level set must be

moved to its new time t^{n+1} location before applying the viscous update.

We use the second order cut-cell method of [19] for computing the pressure in equation (2.4) to make the incompressible velocities divergence free, where the pressure inside the bubbles and the outside air are used as Dirichlet boundary conditions. In the presence of surface tension, the term $\sigma\kappa$ is added to the Dirichlet pressure, where σ is the coefficient of surface tension and κ is the curvature of the interface, computed using the level set method [50]. See also Chapter 6.

The incompressible time step Δt_I is computed by enforcing the following inequality at every cell center, as described in [33],

$$\Delta t_I \left(\frac{C_{\mathsf{cfl}} + \sqrt{C_{\mathsf{cfl}}^2 + 4S_{\mathsf{cfl}}^2 + 4F_{\mathsf{cfl}}^2}}{2} \right) \leq 1 \tag{2.10}$$

Here, C_{cfl} accounts for the convection terms where v_1 and v_2 have been averaged to the cell center,

$$C_{\mathsf{cfl}} = \frac{|v_1|}{\Delta x} + \frac{|v_2|}{\Delta y} \tag{2.11}$$

S_{cfl} accounts for the surface tension forces,

$$S_{\mathsf{cfl}} = \sqrt{\frac{\sigma\kappa}{\rho_I(\min\{\Delta x, \Delta y\})^2}} \tag{2.12}$$

and F_{cfl} accounts for the body forces $\vec{f}$

$$F_{\mathsf{cfl}} = \sqrt{\frac{|f_1|}{\Delta x} + \frac{|f_2|}{\Delta y}} \tag{2.13}$$

Note that the time step restrictions due to viscosity are not present as it is treated implicitly.

2.2 The oscillating bubble problem

We consider a model oscillating bubble problem in one spatial dimension as shown in Figure 2.1, where an air bubble of radius $r^0 = .1$ m with initial density $\rho^0 = 1.1$ kg/m^3 is inside a water "sphere" of radius $r_w^0 = .4$ m. The computational domain is [0 m, 1 m] which gives .1 m of free air on each side of the water region. Figure 2.2 shows the problem in two spatial dimensions where an air bubble of radius $r^0 = .1$ m with initial density $\rho^0 = 1.1$ kg/m^3 is inside a water sphere of radius $r_w^0 = .4$ m. The computational domain is [0 m, 1 m] $\times$ [0 m, 1 m]. The setup for the problem in three spatial dimensions is defined similarly. For simplicity, there is no gravity, surface tension or viscosity in the system. Since the bubble is slightly compressed with density $\rho^0 = 1.1$ kg/m^3, there will be a larger pressure p^0 inside the bubble than in the ambient air which is taken to be a free surface condition of $p_{\mathsf{atm}} = 101,325$ Pa and therefore, the bubble will start to expand, subsequently vibrating back and forth. The appendices derive a second order ODE given by equations (A.4), (A.11) and (A.18). In all three equations, we take the standard approach of solving for $\ddot{R}(t)$, rewriting the second order equation as a first order system, subsequently integrating in time using third order accurate TVD Runge-Kutta, and refining the time step until the solutions converge to obtain data that we use for the "exact" solutions when these equations are considered.

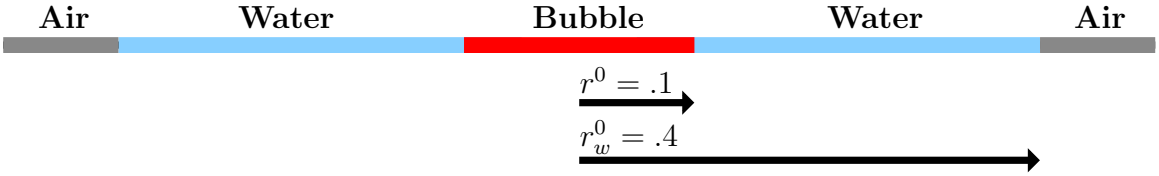


Figure 2.1: Setup for the oscillating bubble problem in one spatial dimension.

For this problem, we modify the time step restriction of Section 2.1 to account for velocities near zero when the bubble volume is at an extrema, i.e., when it has maximum or minimum volume. To prevent the time step from becoming excessively large in these cases, we add a term to Δt_I that estimates the change in velocity over a time

step, similar in spirit to what was done in [33] for body forces and [38] for compressible flow - see also Section 4.4 for a quick summary. Essentially what is needed is an estimate for $|\nabla p|$ which will influence the velocity. We do this by computing l_x and l_y as the minimum thicknesses of the water region in the x and y directions, and approximate $|p_x|$ as $|p_{\mathsf{atm}} - p^n|/l_x$ and $|p_y|$ as $|p_{\mathsf{atm}} - p^n|/l_y$, where p^n is the pressure inside the bubble at time t^n. Then we write

$$F_{\mathsf{cfl}} = \sqrt{\frac{|p_{\mathsf{atm}} - p^n|}{\rho^n l_x \Delta x} + \frac{|p_{\mathsf{atm}} - p^n|}{\rho^n l_y \Delta y}} \tag{2.14}$$

and include this in equation (2.10) analogous to equation (4.30).

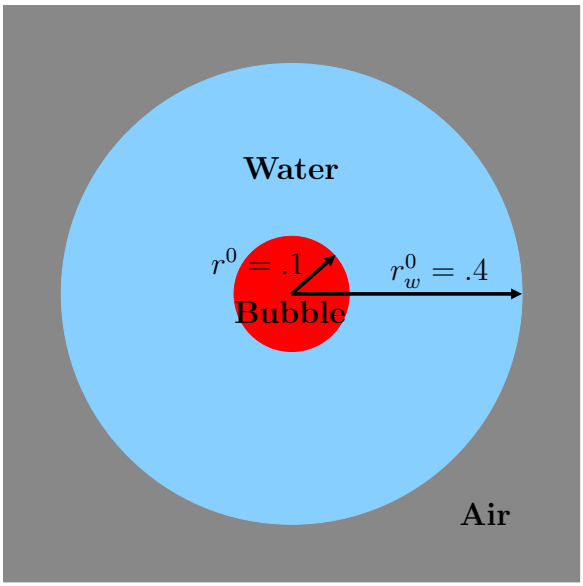


Figure 2.2: Setup for the oscillating bubble problem in two spatial dimensions.

2.3 Treating bubbles

Initially, each bubble is assigned a density (or mass), and the density is advected using the unconditionally stable conservative semi-Lagrangian scheme of [41]. This scheme is especially effective in keeping track of small bubbles, since the level set loses

volume over time and cannot keep track of sub-grid level details. Although one could advect the bubble density using velocities extrapolated from the incompressible flow, we use a more accurate approach where air velocities are constructed and maintained in a separate velocity field $\vec{u}$. The velocity used for level set advection is a hybrid between the incompressible flow velocity and the air velocities. Note that the effects of viscosity and pressure projection from time t^n are already present in this hybrid velocity field, although the air component of these velocities is inviscid. For greater stability, one could advect the time t^n level set function with the hybrid velocity field at time t^{n+1}, although we do not follow this approach. Despite accurate velocities, numerical smearing and other errors will cause the location of the zero level set and the location of the non-zero bubble densities to drift apart over time. We address this as follows. First we compute the total mass that belongs to a bubble as the sum of all the mass inside the bubble and all the mass closest to that bubble. Then we use a flood fill algorithm on that bubble to identify all grid cells belonging to that connected component. The volume of this connected component is carefully computed using a piecewise linear reconstruction of the level set as outlined in [40]. The mass is then uniformly redistributed inside the bubble to obtain a spatially constant density ρ_{b}. The Dirichlet boundary condition used for the incompressible Poisson solve given in equation (2.4) is computed using the equation of state $p = B\rho$ which simplifies to $p = BM/V$, where M and V are the total mass and total volume of a bubble respectively. Here, B is taken to be p_{atm} m^3/kg, so that a density of 1 kg/m^3 yields a pressure of $p_{\mathsf{atm}} = 101,325$ Pa.

Air velocities are treated in a manner similar to the incompressible flow velocities. Ghost node values are defined using the fast extension method of [1] exactly as is done for the incompressible velocities, and then the air velocities are advected using the MacCormack method [53]. Since the air is not strictly volume preserving as bubbles can expand and contract, we do not make the air flow incompressible as would be done for a truly two-phase incompressible flow. Instead, we solve a modified form of equations (2.4) and (2.5) for each bubble

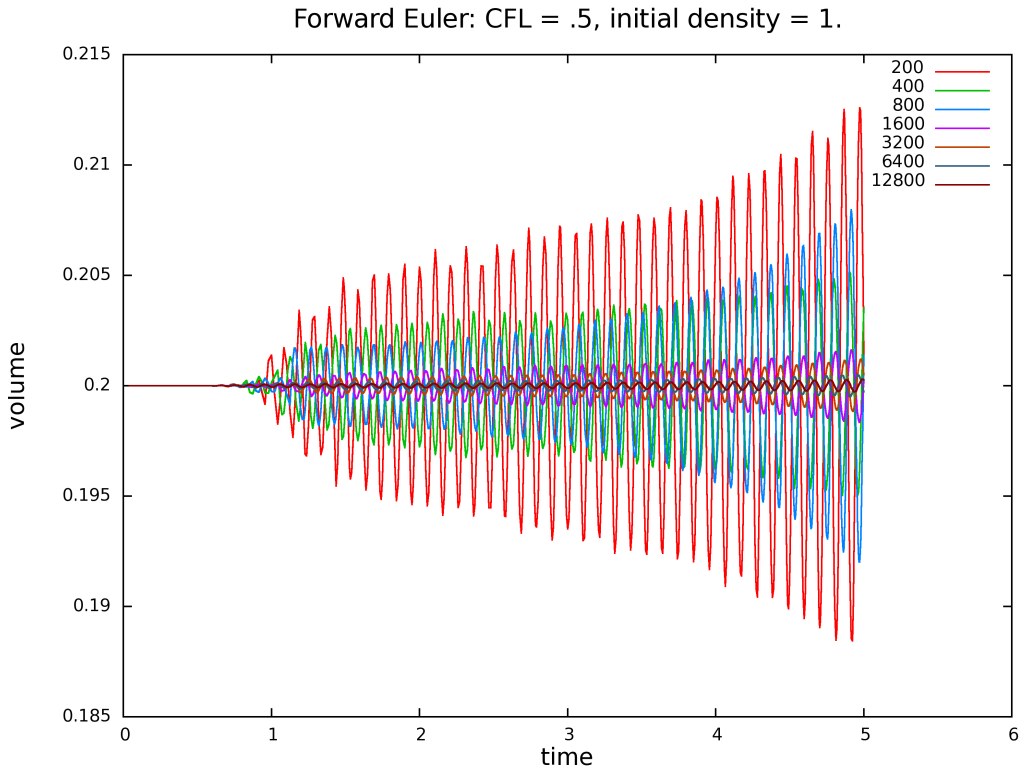


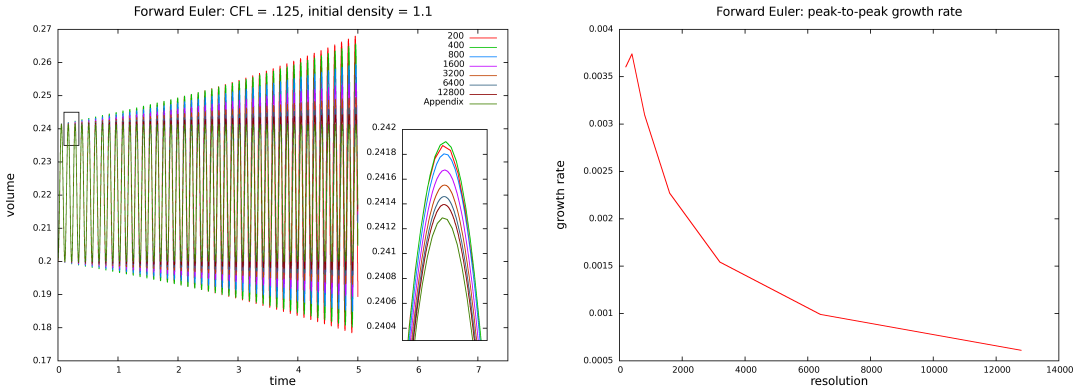
Figure 2.3: Numerical profiles generated by the partitioned scheme in one spatial dimension for bubble volume over time for the stationary bubble problem. Note the forward Euler characteristic of the partitioned scheme which causes instability in the computed solutions at lower grid resolutions. For higher grid resolutions, the nonphysical growth is slower as expected.

$$\nabla \cdot \nabla \tilde{p} = \nabla \cdot \vec{u}^{\star} - \nabla \cdot \vec{u}^{n+1} \tag{2.15}$$

$$\vec{u}^{n+1} - \vec{u}^{\star} + \nabla \tilde{p} = 0 \tag{2.16}$$

where $\tilde{p} = \Delta t p / \rho_{\mathsf{b}}$, ρ_{b} is the spatially constant air density inside the bubble (which could be different per bubble), and $\vec{u}^{\star}$ is the post-advected air velocity. Equations (2.4) and (2.5) are solved first for the full water volume using Dirichlet boundary conditions, after which the resulting water velocities surrounding the bubble are used as Neumann boundary conditions to solve equation (2.15). Since the integral of the

incompressible velocity around the surface of a bubble may not be zero, we compute the net divergence for the boundary of the bubble summing all these velocities divided by the number of cells in the bubble, and use that value for $\nabla \cdot \vec{u}^{n+1}$. This allows bubbles to expand and contract and otherwise change volume as they follow and are enslaved by the surrounding incompressible flow which has much higher momentum. Also note that the Poisson matrix resulting from equation (2.15) has a rank-deficiency of 1 due to the full Neumann boundary conditions and although the addition of $\nabla \cdot \vec{u}^{n+1}$ guarantees that the right hand side is in the range of the Poisson matrix, one still needs to take care to compute the minimum norm solution during the conjugate gradient solve.



(a) (b)

Figure 2.4: (a) Numerical profiles generated by the partitioned scheme in one spatial dimension for bubble volume over time for the oscillating bubble problem, (b) peak-to-peak growth rates under grid refinement. Note that the partitioned scheme shows instability at lower grid resolutions because of its forward Euler characteristic, although converges to the "exact" solution under grid refinement.

In Figures 2.3 and 2.4 we present numerical profiles generated by the partitioned scheme. Consider the setup shown in Figure 2.1 and assume the bubble is initially at rest, i.e., the initial air density inside it is $\rho^0 = 1$ kg/m^3. Analytically, the bubble should just stay at rest and the bubble volume should remain constant over time. We refer to this problem as the *stationary bubble problem*. Figure 2.3 shows the bubble volume profiles over time with the proposed partitioned scheme. Note that although the initial solution computed by this scheme is indeed constant, it quickly

goes unstable at lower grid resolutions because of the forward Euler characteristic of the partitioned scheme. For higher grid resolutions, the nonphysical growth is slower as expected. For the one dimensional oscillating bubble problem, Figure 2.4(a) shows the resulting volume profiles over time under grid refinement. The peak-to-peak growth rates are shown in Figure 2.4(b) and the respective convergence orders are shown in Table 2.1. Again, note that the partitioned scheme shows instability at lower grid resolutions while converging to the "exact" solution under grid refinement. The results for the two dimensional simulations are similar. Note that throughout the paper the peak-to-peak growth/decay rate is defined as the slope of the best fit line to the first few peaks in the bubble volume profile, whereas the convergence order is computed for the peaks shown in the inset zoom ins.

resolution1	resolution2	resolution3	convergence order
400	800	1600	-.4076
800	1600	3200	.1035
1600	3200	6400	.3797
3200	6400	12800	.5168

Table 2.1: Convergence orders for the volume profiles generated by the partitioned scheme for the oscillating bubble problem.

Chapter 3

An iterative approach

Although the partitioned scheme proposed in Chapter 2 converges to correct analytical solutions under grid refinement, it shows instability at lower grid resolutions because of its forward Euler characteristic. The most logical next step is to try using TVD Runge-Kutta methods for greater stability [54]. We approach this by applying second order accurate TVD Runge-Kutta on the bubble pressure p_{b} which is used as a Dirichlet boundary condition in equation (2.4). Specifically, we take two full steps of our method to compute the spatially constant density $\hat{\rho}_{\mathsf{b}}^{n+2}$ inside the bubble and use the average $(\rho_b^n + \hat{\rho}_b^{n+2})/2$ for computing the Dirichlet pressure $p_{\mathsf{b}} = B(\rho_{\mathsf{b}}^n + \hat{\rho}_b^{n+2})/2$. Subsequently, we rewind the simulation to the beginning of the time step and use this Dirichlet pressure boundary condition in the incompressible flow solve in equations (2.4) and (2.5) in order to obtain the divergence free incompressible velocity field. Note that intermediate substeps can result in velocities that dictate a smaller step size than that chosen at the beginning of the time step. Ignoring this can result in inaccurate solutions, so if this occurs we revert to the beginning of the time step and start over using smaller step size. Figure 3.1 shows a comparison between the forward Euler scheme, second order accurate TVD Runge-Kutta on the Dirichlet pressure p_{b} without modifying the time step, and the modified second order accurate TVD Runge-Kutta which reverts the simulation to the beginning of the time step if

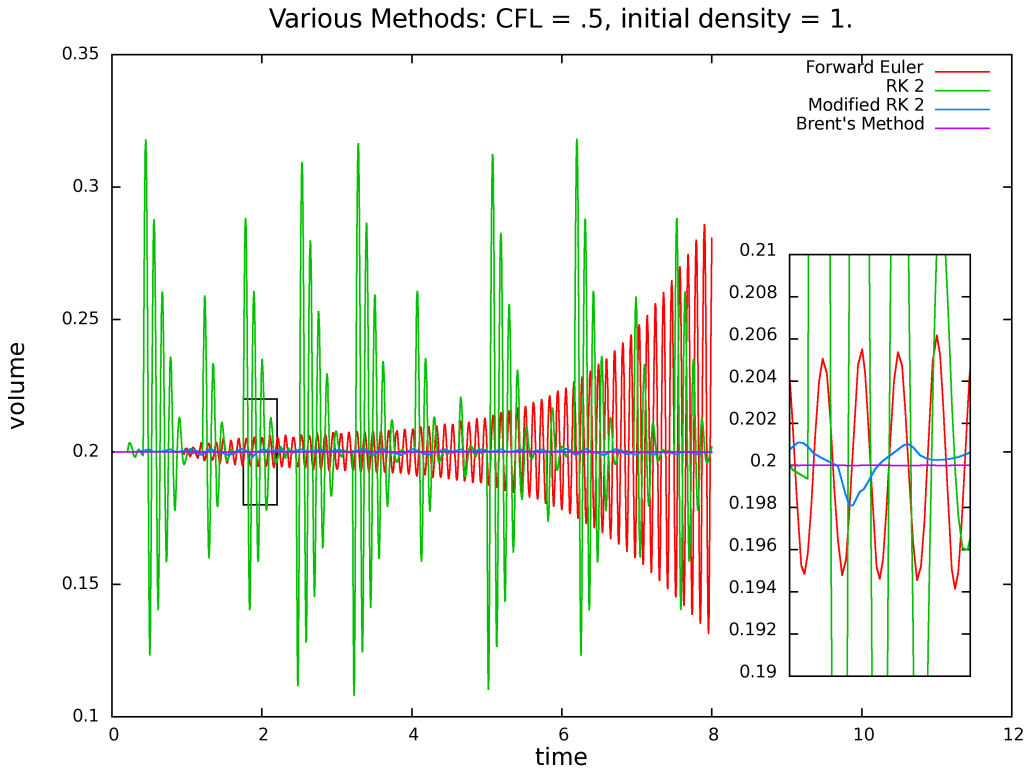


Figure 3.1: A comparison between numerical profiles for the one dimensional stationary bubble problem generated using the forward Euler scheme (red), second order TVD Runge-Kutta (green), modified second order TVD Runge-Kutta which takes time step restrictions imposed by intermediate velocities into account (blue), and Brent's method (magenta).

the CFL condition is violated. Note the intermittent spikes generated by the standard second order accurate TVD Runge-Kutta scheme, which are not present in the modified rewinding version as it obeys the CFL time step restriction even at intermediate steps. Also note that the modified second order accurate TVD Runge-Kutta scheme does a much better job at tracking the constant solution than the forward Euler version of the scheme.

The enhanced stability shown by the use of a second order accurate TVD Runge-Kutta scheme on the Dirichlet pressure boundary condition p_{b} used for the bubble in solving equations (2.4) and (2.5) motivates consideration of a technique similar

to [38] where our goal is to replace the $p = BM/V$ Dirichlet boundary condition with the pressure the bubble would have in the next time step after solving for the incompressible velocities and advecting the bubble forward in time. Consider the pressure evolution equation [15],

$$p_t + \vec{u} \cdot \nabla p = -\rho c^2 \nabla \cdot \vec{u} \tag{3.1}$$

Since we assume that the air density inside the bubble is spatially constant, it follows from $p = B\rho$ that $\nabla p = 0$ and thus

$$p_t = -\rho c^2 \nabla \cdot \vec{u} \tag{3.2}$$

For a gas governed by the equation of state $p = B\rho$, the sound speed c is defined as

$$c = \sqrt{p_\rho + \frac{pp_e}{\rho^2}} = \sqrt{B} \tag{3.3}$$

implying that it is constant in time and space. Discretizing equation (3.2) in time gives

$$p^{n+1} = p^n - \Delta t \rho^{n+1} B \nabla \cdot \vec{u} \tag{3.4}$$

where we set ρ to time t^{n+1}. As mass is conserved over time $M^{n+1} = M^n = M$. Using $p = B\rho$, $\rho^n = M/V^n$ and $\rho^{n+1} = M/V^{n+1}$, we obtain

$$\frac{BM}{V^{n+1}} - \frac{BM}{V^n} = -\frac{BM}{V^{n+1}} \Delta t \nabla \cdot \vec{u} \tag{3.5}$$

or

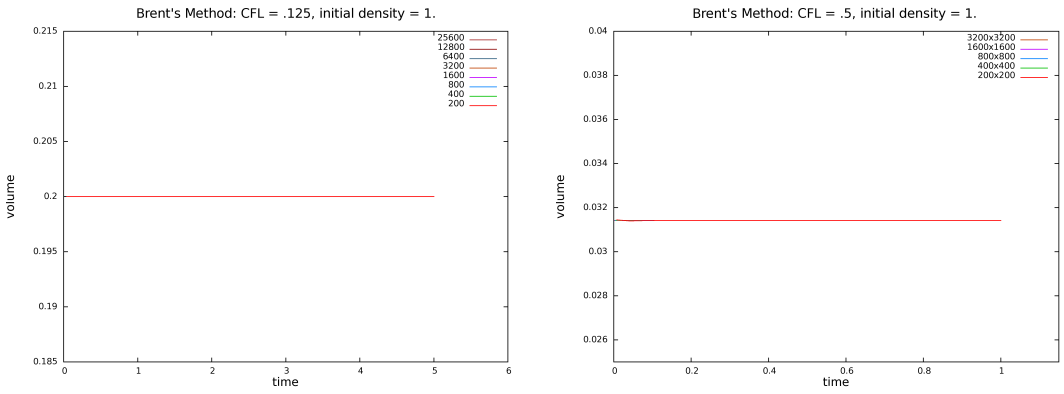
$$\Delta V = V^n \Delta t \nabla \cdot \vec{u} \tag{3.6}$$

where $\Delta V = V^{n+1} - V^n$. Using the divergence theorem, $V^n \nabla \cdot \vec{u}$ inside the bubble is equivalent to $\oint \vec{u} \cdot \vec{n} dS$. Letting $\bar{u}$ be the average normal velocity on the boundary of

the bubble and $\mathcal{P}$ be the perimeter of the bubble allows us to write

$$\Delta V = \Delta t \mathcal{P} \bar{u} \tag{3.7}$$

Note that we use a piecewise linear reconstruction of the level set as described in [40] for computing $\mathcal{P}$ and $\bar{u}$.



(a) (b)

Figure 3.2: Numerical profiles generated by the iterative scheme for bubble volume over time under grid refinement for the stationary bubble problem in, (a) one spatial dimension, (b) two spatial dimensions.

The initial guess for the iterative solver sets $p = BM/V$ as a Dirichlet boundary condition for projecting the incompressible velocities. These velocities are then used to guess the bubble's volume V^{n+1} using equation (3.7). Since mass is constant, this predicts a new $p^{n+1} = B\rho^{n+1} = BM/V^{n+1}$. This pressure can again be set as a Dirichlet boundary condition to project the incompressible velocities and improve the guess for p^{n+1}. Through the iterative solver, we are looking for a fixed point for this "function", i.e., $p = f(p)$ or a root of $g(p) = f(p) - p$. Note that if the input pressure is too large, the bubble expands and the predicted pressure drops, and similarly if the input pressure is too small, the bubble contracts and the predicted pressure increases. This allows us to place bounds on the solution. Basically, we start with $p^n = BM/V^n$ and if $f(p)$ is bigger, the initial guess for p is the left end-point of our interval. Otherwise, if $f(p)$ is smaller, this is taken as the right end-point of the interval. As the iteration proceeds, we eventually identify both left and right

end-points. Once we find a bounding interval we use Brent's method [2] to find the root.

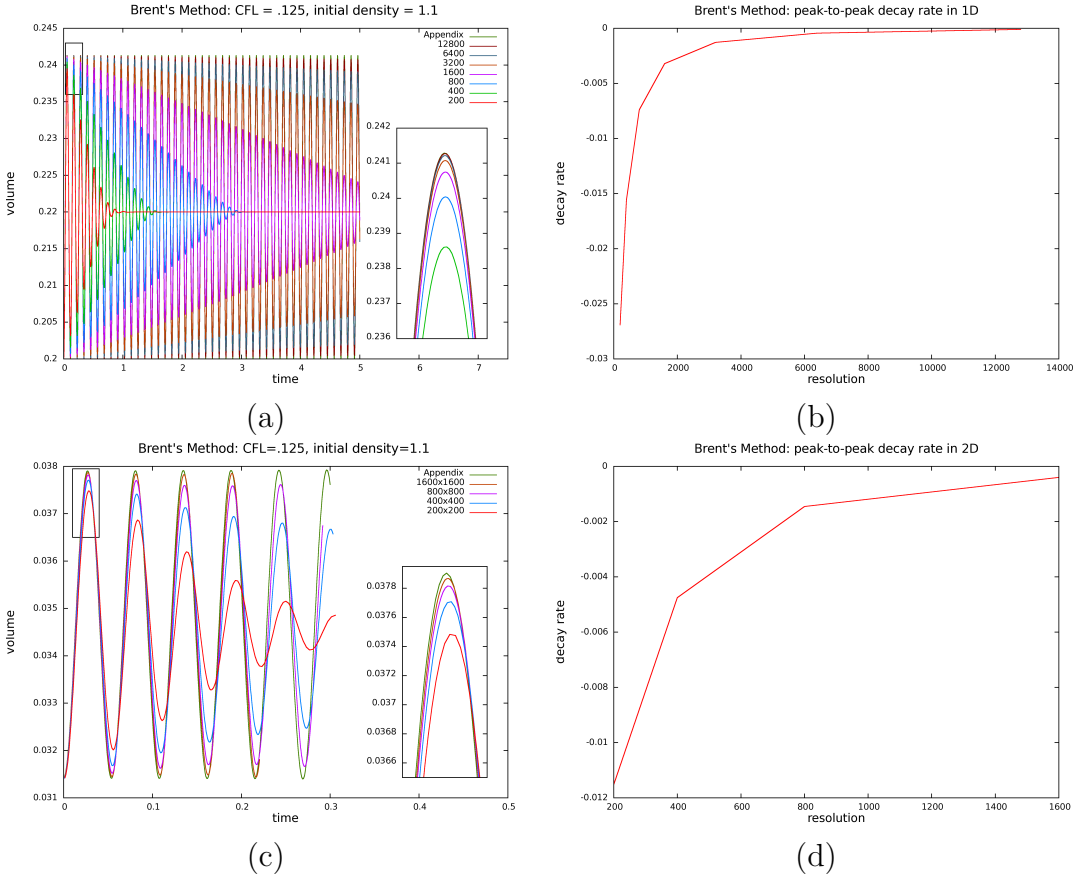


Figure 3.3: Numerical profiles generated by the iterative scheme for bubble volume over time under grid refinement for the oscillating bubble problem in, (a) one spatial dimension, and (c) two spatial dimensions. The peak-to-peak decay rates are shown in (b) one spatial dimension, and (d) two spatial dimensions. Note the improved stability achieved by the iterative method as compared to the forward Euler scheme. Also note that the highest resolution simulation in (c) has not been run for the entire time length because the bubble broke up into multiple bubbles due to Kelvin-Helmholtz instability, breaking our underlying assumptions and making further computation meaningless.

Figures 3.2(a) and 3.2(b) show the numerical profiles generated by the iterative solver for the bubble volume over time with increasing grid resolutions in one and two spatial dimensions for the stationary bubble problem. Note the improved stability achieved

resolution1	resolution2	resolution3	convergence order
200	400	800	.9043
400	800	1600	1.0153
800	1600	3200	1.1083
1600	3200	6400	1.2136
3200	6400	12800	1.3847

Table 3.1: Convergence orders for the volume profiles generated by the iterative scheme for the oscillating bubble problem in one spatial dimension.

resolution1	resolution2	resolution3	convergence order
200	400	800	1.0411
400	800	1600	1.1084

Table 3.2: Convergence orders for the volume profiles generated by the iterative scheme for the oscillating bubble problem in two spatial dimensions.

by the iterative solver. For the sake of comparison, the iterative solver is also shown labeled as Brent's method in Figure 3.1. Next consider the oscillating bubble problem. Figures 3.3(a) and 3.3(c) show the corresponding profiles for the bubble volume over time under grid refinement. The peak-to-peak decay rates for both one and two spatial dimensions are shown in Figures 3.3(b) and 3.3(d) while the respective convergence orders are shown in Tables 3.1 and 3.2. Comparing Figures 2.4 and 3.3, note that the explicit method diverges for low grid resolutions and eventually converges to the "exact" solutions as the grid is refined. In contrast, the implicit solution overly damps the phenomena on coarse grids while still converging to the "exact" solutions as the grid is refined. Obviously one would prefer the implicit approach over the explicit one so that although unresolved phenomena are not accurately resolved, i.e., the solution is overdamped, they at least do not explode and corrupt the entire solution.

Chapter 4

Compressible-incompressible coupling

Motivated by the method in [38] which proposes an elliptic solver for pressure for compressible flow, our goal is to develop a fully monolithic solver which solves for air and water pressures together. However, before doing this we first develop a full monolithic solver that couples incompressible flow with fully non-linear compressible flow including contact discontinuities like shocks and rarefactions building upon the partitioned solver of [7]. They used a fully explicit scheme for the compressible fluid, and the pressure in the compressible region was used as a Dirichlet boundary condition when solving the incompressible Poisson equation for updating the incompressible velocities. Near the interface, the ghost fluid method (GFM) [14] was used to treat the boundary conditions in a manner that admitted sharp discontinuities while still allowing for smooth discretizations across the interface. To achieve this, the interface values of pressure and normal velocity were carefully determined noting that, although these variables are continuous, they may possess kinks across the interface. Small errors in the normal velocity of the incompressible fluid create small errors in its divergence, which in turn can lead to large spurious pressure oscillations in the incompressible region. While small errors in the velocity of the compressible fluid

cause the same small errors in its density, these have little effect on the gas since the gamma gas law equation of state is rather robust. Again, since the incompressible flow's pressure response is rather stiff, one can expect large variations in the incompressible pressure near the interface which in turn can lead to poor predictions of the interface pressure. While these errors in the interface pressure have a relatively small effect on the heavier incompressible fluid, they can have a rather large effect on the lighter compressible gas. Conversely, since the gamma gas law equation of state is rather robust, the compressible pressure tends to be smooth near the interface and is therefore a good candidate for the interface pressure. In view of these statements, [7] proposed using the incompressible region to determine the interface normal velocity and the compressible region to determine the interface pressure. In the presence of surface tension, the compressible pressure is not used directly, but is first modified according to the appropriate $[p] = \sigma\kappa$ jump condition. Although the method of [7] works well, it suffers from a strict time step restriction because the sound speed in the compressible fluid dictates the size of the time step. In addition, the method of [7] utilizes a partitioned coupling approach which can suffer from stability issues.

4.1 A semi-implicit formulation for compressible flow

Let ρ be the density of the compressible fluid, $\vec{u}$ its velocity, E the total energy, p the pressure, and $\mathbf{U} = (\rho, \rho\vec{u}, E)$ the compressible state vector. The inviscid compressible Euler equations in multiple spatial dimensions are as follows

$$\mathbf{U}_t + \nabla \cdot \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho \\ \rho\vec{u} \\ E \end{pmatrix}_t + \nabla \cdot \begin{pmatrix} \rho\vec{u} \\ \rho\vec{u} \otimes \vec{u} + p \\ (E+p)\vec{u} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{4.1}$$

A semi-implicit formulation for solving these equations was recently proposed in [38], where the flux vector $\mathbf{F}(\mathbf{U})$ was split into an advection part and a nonadvection part

$$\mathbf{F}_1(\mathbf{U}) = \begin{pmatrix} \rho\vec{u} \\ \rho\vec{u}\otimes\vec{u} \\ E\vec{u} \end{pmatrix}, \mathbf{F}_2(\mathbf{U}) = \begin{pmatrix} 0 \\ p \\ p\vec{u} \end{pmatrix} \tag{4.2}$$

The advection part $\mathbf{F}_1(\mathbf{U})$ was integrated explicitly to get intermediate values $\rho^\star$, $(\rho\vec{u})^\star$ and $E^\star$, and since pressure does not affect the continuity equation, it follows that $\rho^{n+1} = \rho^\star$. The original method of [38] used a modified ENO scheme for the explicit update to avoid Gibbs phenomena. Recently, [21] proposed using the standard ENO scheme [55] and a different method for computing the post-advected pressure (p_a, see below) which avoids this issue. We follow this improved approach. The nonadvection momentum and energy updates are

$$\frac{(\rho\vec{u})^{n+1} - (\rho\vec{u})^\star}{\Delta t} = -\nabla p \tag{4.3}$$

$$\frac{E^{n+1} - E^\star}{\Delta t} = -\nabla\cdot(p\vec{u}) \tag{4.4}$$

Motivated by the standard incompressible flow formulation, equation (4.3) is divided by ρ^{n+1} to obtain

$$\vec{u}^{n+1} = \vec{u}^\star - \Delta t\frac{\nabla p^{n+1}}{\rho^{n+1}} \tag{4.5}$$

and its divergence is taken to obtain

$$\nabla\cdot\vec{u}^{n+1} = \nabla\cdot\vec{u}^\star - \Delta t\nabla\cdot\left(\frac{\nabla p^{n+1}}{\rho^{n+1}}\right) \tag{4.6}$$

Then the pressure evolution equation (3.1) is semi-discretized by fixing $\nabla\cdot\vec{u}$ to time t^{n+1} through the time step and by treating the advection terms explicitly. Let $e = E/\rho - \vec{u}\cdot\vec{u}/2$ denote the internal energy per unit mass, then the advected pressure is computed as $p_a = p^\star = p(\rho^\star, e^\star)$ using the equation of state. Substituting $p^\star$ into the

semi-discretized form of equation (3.1), we obtain

$$p^{n+1} = p^{\star} - \Delta t \rho c^2 \nabla \cdot \vec{u}^{n+1} \tag{4.7}$$

Eliminating $\nabla \cdot \vec{u}^{n+1}$ by combining equations (4.7) and (4.6) and rearranging the terms gives

$$p^{n+1} - \Delta t^2 \rho^n (c^2)^n \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = p^{\star} - \Delta t \rho^n (c^2)^n \nabla \cdot \vec{u}^{\star} \tag{4.8}$$

where the term ρc^2 has been fixed to the time t^n value. By composing the $\rho^n (c^2)^n$ terms into a diagonal matrix $P = [\Delta t^2 \rho^n (c^2)^n]$ and discretizing the gradient and divergence operators, we obtain the following system of equations

$$[P^{-1} + G^T (\hat{\rho}^{n+1})^{-1} G] \hat{p}^{n+1} = P^{-1} \hat{p}^{\star} + G^T \hat{u}^{\star} \tag{4.9}$$

where G is the discretized gradient operator, $-G^T$ the corresponding discretized divergence operator. The pressure is scaled by Δt, i.e., $\hat{p} = p \Delta t$, $\hat{\rho}$ is the density interpolated to cell faces and $\hat{u}$ denotes a density-weighted averaged face velocity, i.e.,

$$\hat{\rho}^{n+1}_{i+1/2} = \frac{\rho^{n+1}_i + \rho^{n+1}_{i+1}}{2}, \quad \hat{u}^{\star}_{i+1/2} = \frac{(\rho u)^{\star}_i + (\rho u)^{\star}_{i+1}}{\rho^{n+1}_i + \rho^{n+1}_{i+1}} \tag{4.10}$$

Note that the resulting matrix in equation (4.9) has an identity term in it, which allows fast solvers like preconditioned conjugate gradient (PCG) to converge in relatively few iterations. After solving equation (4.9) to obtain cell-centered pressure values, they are applied in a conservative flux-based manner to update the intermediate momentum and energy. Face pressures are computed using density-weighted averaging, i.e.,

$$\hat{p}^{n+1}_{i+1/2} = \frac{\rho^{n+1}_i \hat{p}^{n+1}_{i+1} + \rho^{n+1}_{i+1} \hat{p}^{n+1}_i}{\rho^{n+1}_i + \rho^{n+1}_{i+1}} \tag{4.11}$$

and face velocities are computed by rewriting equation (4.5) using face-averaged quantities as defined above.

$$\hat{u}_{i+1/2}^{n+1} = \hat{u}_{i+1/2}^{\star} - (\hat{\rho}_{i+1/2}^{n+1})^{-1} G_{i+1/2} \hat{p}^{n+1} \tag{4.12}$$

Here, $G_{i+1/2}$ denotes the row of G corresponding to face $i + 1/2$. The flux-based implicit update then takes the form

$$(\rho\vec{u})_i^{n+1} = (\rho\vec{u})_i^{\star} - \frac{\hat{p}_{i+1/2}^{n+1} - \hat{p}_{i-1/2}^{n+1}}{\Delta x}, \quad E_i^{n+1} = E_i^{\star} - \frac{\hat{p}_{i+1/2}^{n+1}\hat{u}_{i+1/2}^{n+1} - \hat{p}_{i-1/2}^{n+1}\hat{u}_{i-1/2}^{n+1}}{\Delta x} \tag{4.13}$$

4.2 Explicit coupling step

We use the level set method to track the interface between the compressible and incompressible fluids. For the explicit part of the method, the interface boundary conditions are treated as described in [7]. Various quantities need to be extrapolated across the interface in either direction to define ghost node values. This is accomplished using constant extrapolation normal to the interface by solving the equation $I_\tau + \vec{N} \cdot \nabla I = 0$, in fictitious time τ for the different quantities I - we use the fast extension method of [1]. To compute the compressible ghost node state we decompose the extrapolated state vector into entropy, pressure and velocity, compute the cell centered incompressible velocity, and replace the normal component of the compressible velocity field with the normal component of the incompressible cell centered velocity field before reassembling entropy, pressure and velocity to obtain a ghost state for the compressible flow. The explicit update for the compressible fluid consists of applying only the advection fluxes from equation (4.2) using these ghost node values. In order to properly handle uncovered cells a band of ghost node values is also updated in time.

The incompressible flow update proceeds similarly as to what was described in Section 2.1. The full incompressible velocity field is extrapolated across the interface into the compressible region to obtain ghost node values, the advection terms are updated

to obtain intermediate velocities, and the viscous terms are updated noting that the compressible fluid is inviscid and thus cannot absorb any viscous stress.

Note that when applying viscosity to the incompressible fluid, the time $\hat{t}$ velocities $\hat{v}_1$ and $\hat{v}_2$ are sometimes first projected to be divergence free, as noted in Section 2.1 after equations (2.8) and (2.9). In our monolithic approach, the divergence free projection is two-way coupled between the compressible and incompressible flow as given in equations (4.20) and (4.21) below in Section 4.3. Thus, our strategy is to apply this two-way coupled projection first and then use the resulting incompressible state to add the effects of the viscous terms to the incompressible fluid. Interestingly, note that this first coupled projection solve essentially gives the answer one would obtain if the incompressible flow was inviscid. Thus for the viscous case, we essentially obtain the inviscid solution first, rewind the compressible state vector $\mathbf{U}$ to its pre-projected state, apply the viscous update to the incompressible region, and then apply the coupled projection once again to obtain a final solution which includes the effects of viscosity on the incompressible fluid.

4.3 Implicit coupling step

For the sake of exposition, we describe the implicit step in one spatial dimension. Multiple spatial dimensions are handled in a straightforward manner using a dimension-by-dimension approach. Consider the situation depicted in Figure 4.1. Let p_{int} denote the interface pressure and $\theta = |\phi(x_2)|/(|\phi(x_2)| + |\phi(x_3)|)$ be the cell fraction between the interface and the center of cell 2. Discretizing the incompressible flow Poisson equation (2.4) for cell 3, we obtain

$$-\frac{\Delta t}{\Delta x}\left(\frac{p_4^{n+1} - p_3^{n+1}}{\rho_I \Delta x} - \frac{p_3^{n+1} - p_{\mathsf{int}}}{\rho_I(1-\theta)\Delta x}\right) = -\nabla \cdot \vec{v}^{\star} \tag{4.14}$$

Recall from Sections 2.1 and 4.1 that $\vec{v}$ refers to the face centered incompressible velocity while $\vec{u}$ refers to the cell centered compressible velocity. Let I and C subscripts

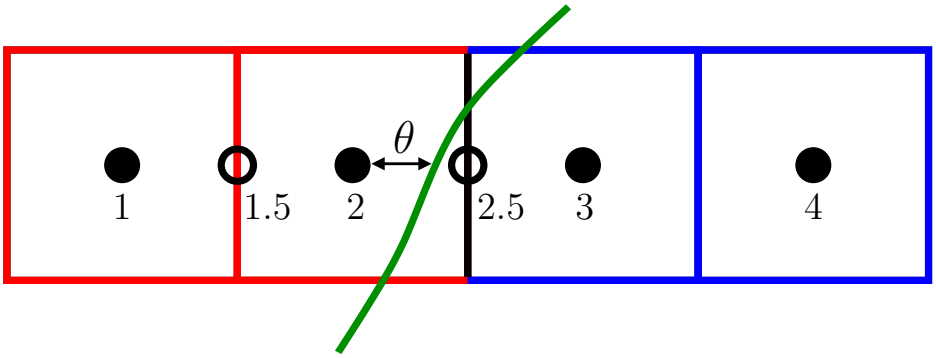


Figure 4.1: A global Poisson solve for coupling together compressible and incompressible fluids. Compressible cells are shown with red borders, incompressible cells are shown with blue borders. The interface is shown in green and the shared face is colored black.

represent values from the incompressible and compressible sides of the interface. For inviscid flow $\nabla p/\rho$ is continuous across the interface [33] satisfying

$$\frac{\nabla p_I}{\rho_I} = \frac{\nabla p_C}{\rho_C} \tag{4.15}$$

Although [33] showed this for two-phase incompressible flow, one can still derive equation (2.1) for compressible flow from the equations for conservation of mass and momentum and so it also holds for inviscid compressible flow and mixed compressible-incompressible flow (as long as the strong form of the equations hold and derivatives exist). For viscous flows things can be more complex, however, in our case we assume the compressible flow is inviscid and that there is no viscous momentum transfer from the incompressible to the compressible flow as mentioned in Section 2.1 (see also Section 4.2) - allowing equation (4.15) to still hold. Approximating equation (4.15) with one-sided differences, we obtain

$$\frac{p_{\mathsf{int}} - p_2^{n+1}}{\rho_C \theta \Delta x} = \frac{p_3^{n+1} - p_{\mathsf{int}}}{\rho_I (1-\theta) \Delta x} \tag{4.16}$$

where we take $\rho_C = \rho_2^{n+1}$ (one might also conceivably use $\hat{\rho}_{2.5}^{n+1}$). Solving equation (4.16) for the interface pressure p_{int} gives

$$p_{\mathsf{int}} = \frac{\theta\rho_C p_3^{n+1} + (1-\theta)\rho_I p_2^{n+1}}{\theta\rho_C + (1-\theta)\rho_I} \tag{4.17}$$

Writing

$$\tilde{\rho} = \theta\rho_C + (1-\theta)\rho_I \tag{4.18}$$

allows us to write

$$\frac{p_{\mathsf{int}} - p_2^{n+1}}{\rho_C\theta\Delta x} = \frac{p_3^{n+1} - p_2^{n+1}}{\tilde{\rho}\Delta x} = \frac{p_3^{n+1} - p_{\mathsf{int}}}{(1-\theta)\rho_I\Delta x} \tag{4.19}$$

and thus, equation (4.14) becomes

$$-\frac{\Delta t}{\Delta x}\left(\frac{p_4^{n+1} - p_3^{n+1}}{\rho_I\Delta x} - \frac{p_3^{n+1} - p_2^{n+1}}{\tilde{\rho}\Delta x}\right) = -\nabla\cdot\vec{v}^{\star} \tag{4.20}$$

Next consider the compressible Poisson equation in cell 2. Note that cell 3 has a valid compressible state vector after advection because we also update a band of ghost cells near the interface (see Section 4.2). Let $\hat{\rho}_{2.5}^{n+1}, \hat{\rho}_{1.5}^{n+1}$ denote the interpolated face density obtained by averaging cell-centered values. Discretizing the compressible Poisson equation (4.9) for cell 2 and using equation (4.19) gives

$$\frac{p_2^{n+1}}{\rho_2^n(c_2^2)^n\Delta t} - \frac{\Delta t}{\Delta x}\left(\frac{p_3^{n+1} - p_2^{n+1}}{\tilde{\rho}\Delta x} - \frac{p_2^{n+1} - p_1^{n+1}}{\hat{\rho}_{1.5}^{n+1}\Delta x}\right) = \frac{p_2^{\star}}{\rho_2^n(c_2^2)^n\Delta t} - \nabla\cdot\hat{u}^{\star} \tag{4.21}$$

Note that equations (4.20) and (4.21) together form a symmetric positive definite (SPD) system, allowing for the use of fast Poisson solvers such as PCG.

Besides using $\tilde{\rho}$ in equations (4.20) and (4.21) to enforce the balance condition of equation (4.15) we also desire a unique velocity at the shared face location $x_{2.5}$ (shown in black in Figure 4.1) for computing the term $\nabla\cdot\vec{v}^{\star}$ in equation (4.20) and the term $\nabla\cdot\hat{u}^{\star}$ in equation (4.21). Since we have separate velocity fields for compressible and

incompressible fluids, the values $\hat{u}^{\star}_{2.5}$ and $v^{\star}_{2.5}$ at the shared face need not be equal, even though theoretically the normal component of the velocity field is supposed to be continuous across the interface. Thus, at the shared face we apply a force λ to the incompressible fluid and an equal and opposite force $-\lambda$ to the compressible fluid so that the resulting time $t^{\star\star}$ velocities at the shared face are equal, i.e.,

$$\hat{\rho}^{n+1}_{2.5}\hat{u}^{\star\star}_{2.5} = \hat{\rho}^{n+1}_{2.5}\hat{u}^{\star}_{2.5} - \lambda\Delta t, \quad \rho_I v^{\star\star}_{2.5} = \rho_I v^{\star}_{2.5} + \lambda\Delta t \tag{4.22}$$

Setting $\hat{u}^{\star\star}_{2.5} = v^{\star\star}_{2.5}$, and solving for λ gives

$$\hat{u}^{\star\star}_{2.5} = v^{\star\star}_{2.5} = \frac{\rho_I v^{\star}_{2.5} + \hat{\rho}^{n+1}_{2.5}\hat{u}^{\star}_{2.5}}{\rho_I + \hat{\rho}^{n+1}_{2.5}} \tag{4.23}$$

as our velocity at the shared face (note that since the incompressible density tends to be much bigger than the compressible density, one could also use the incompressible velocity and obtain similar results - we have tested this numerically). Also note that the compressible face velocity is not an actual degree of freedom since the degrees of freedom for compressible flow lie at cell centers. Hence, we add the momentum $-\lambda\Delta t$ to both the compressible cell 2 and the compressible ghost cell 3, i.e., $(\rho u)^{\star\star}_2 = (\rho u)^{\star}_2 - \lambda\Delta t$ and $(\rho u)^{\star\star}_3 = (\rho u)^{\star}_3 - \lambda\Delta t$, so that the first equation of equation (4.22) is satisfied. This is equivalent in spirit to adding $-\lambda\Delta t$ to cell 2 and then re-extrapolating into ghost cell 3. Finally, after changing the compressible state vector at cell 2, the velocity $\hat{u}^{\star}_{1.5}$ is recomputed to make it consistent with the new state in cell 2. In multiple spatial dimensions, if a compressible cell borders multiple incompressible cells then it would be updated multiple times, leading to the compressible velocity at the shared face not matching the incompressible velocity. We use the updated incompressible velocity $v^{\star\star}_{2.5}$ at the shared face in this case for greater accuracy.

After solving the coupled projection solve of equations (4.20) and (4.21), the incompressible velocity at the shared face is updated to time t^{n+1} via

$$v^{n+1}_{2.5} = v^{\star\star}_{2.5} - \frac{\hat{p}^{n+1}_3 - \hat{p}^{n+1}_2}{\tilde{\rho}\Delta x} \tag{4.24}$$

For updating the compressible momentum at cell 2, we compute the face pressure $\hat{p}_{2.5}^{n+1}$ using the linear interpolant through p_2^{n+1} and p_{int}, i.e., $p_{2.5}^{n+1} = p_2^{n+1} + (p_{\mathsf{int}} - p_2^{n+1})/2\theta$. If θ is too small for the denominator, we instead compute the slope using equation (4.19) and obtain $p_{2.5}^{n+1} = p_2^{n+1} + \rho_2^{n+1}(p_3^{n+1} - p_2^{n+1})/(2\tilde{\rho})$. The compressible momentum at cell 2 is then updated as

$$(\rho\vec{u})_2^{n+1} = (\rho\vec{u})_2^{\star\star} - \frac{\hat{p}_{2.5}^{n+1} - \hat{p}_{1.5}^{n+1}}{\Delta x} \tag{4.25}$$

noting that $(\rho u)_2^{\star\star}$ includes the momentum update $\lambda\Delta t$. For the energy update, equation (4.13) is still used with the face pressure $\hat{p}_{2.5}^{n+1}$ as computed above and the face velocity $\hat{u}_{2.5}^{n+1}$ at the shared face computed using $\hat{u}_{2.5}^{\star\star}$ in equation (4.12). Note that in the presence of surface tension, we modify the Poisson equations for both the compressible cell 2 and the incompressible cell 3 by taking the jump $\sigma\kappa$ into account, as described in Section 2.1 (see also [33, 42]). Also note that in the presence of surfactants, the surface tension coefficient σ can be variable and our method can easily handle this scenario.

4.4 Time step restriction

The size of the overall time step is computed as the minimum of the incompressible and the compressible time steps, i.e.,

$$\Delta t = \alpha \min\{\Delta t_I, \Delta t_C\} \tag{4.26}$$

where α denotes the CFL number. The incompressible time step Δt_I is computed as described in Section 2.1. The compressible time step Δt_C is computed as described in [38]. In order to prevent Δt_C from becoming infinite for near zero velocities u^n, the term $\nabla p/\rho$ is added which estimates the change in velocity at the end of a time

step. Hence, the CFL condition becomes

$$\Delta t_C \left(\frac{|u^n|_{\max} + \frac{|p_x|}{\rho_C}\Delta t_C}{\Delta x} \right) \leq 1 \tag{4.27}$$

Equation (4.27) is a quadratic in Δt_C with two solutions

$$\frac{-|u^n|_{\max} - \sqrt{|u^n|^2_{\max} + 4\frac{|p_x|}{\rho_C}\Delta x}}{2|p_x|/\rho_C} \leq \Delta t_C \leq \frac{-|u^n|_{\max} + \sqrt{|u^n|^2_{\max} + 4\frac{|p_x|}{\rho_C}\Delta x}}{2|p_x|/\rho_C} \tag{4.28}$$

Note that the lower limit in equation (4.28) is non-positive, and as $\Delta t_C \geq 0$, only the upper bound needs to be enforced. As $p_x \to 0$, both the numerator and the denominator vanish obtaining the typical bound of $\Delta x/|u^n|_{\max}$ which is problematic when $|u^n|_{\max}$ is small. We obtain a more convenient time step restriction which is not plagued by either small $|p_x|$ or small $|u^n|_{\max}$ by replacing the *second* Δt in equation (4.27) with the right hand bound from equation (4.28) to obtain

$$\frac{\Delta t_C}{2} \left(\frac{|u^n|_{\max}}{\Delta x} + \sqrt{\left(\frac{|u^n|_{\max}}{\Delta x} \right)^2 + 4\frac{|p_x|}{\rho_C \Delta x}} \right) \leq 1 \tag{4.29}$$

In two spatial dimensions, the following CFL restriction is obtained

$$\frac{\Delta t_C}{2} \left(\frac{|u_1^n|_{\max}}{\Delta x} + \frac{|u_2^n|_{\max}}{\Delta y} + \sqrt{\left(\frac{|u_1^n|_{\max}}{\Delta x} + \frac{|u_2^n|_{\max}}{\Delta y} \right)^2 + 4\frac{|p_x|}{\rho_C \Delta x} + 4\frac{|p_y|}{\rho_C \Delta y}} \right) \leq 1 \tag{4.30}$$

4.5 Numerical Results

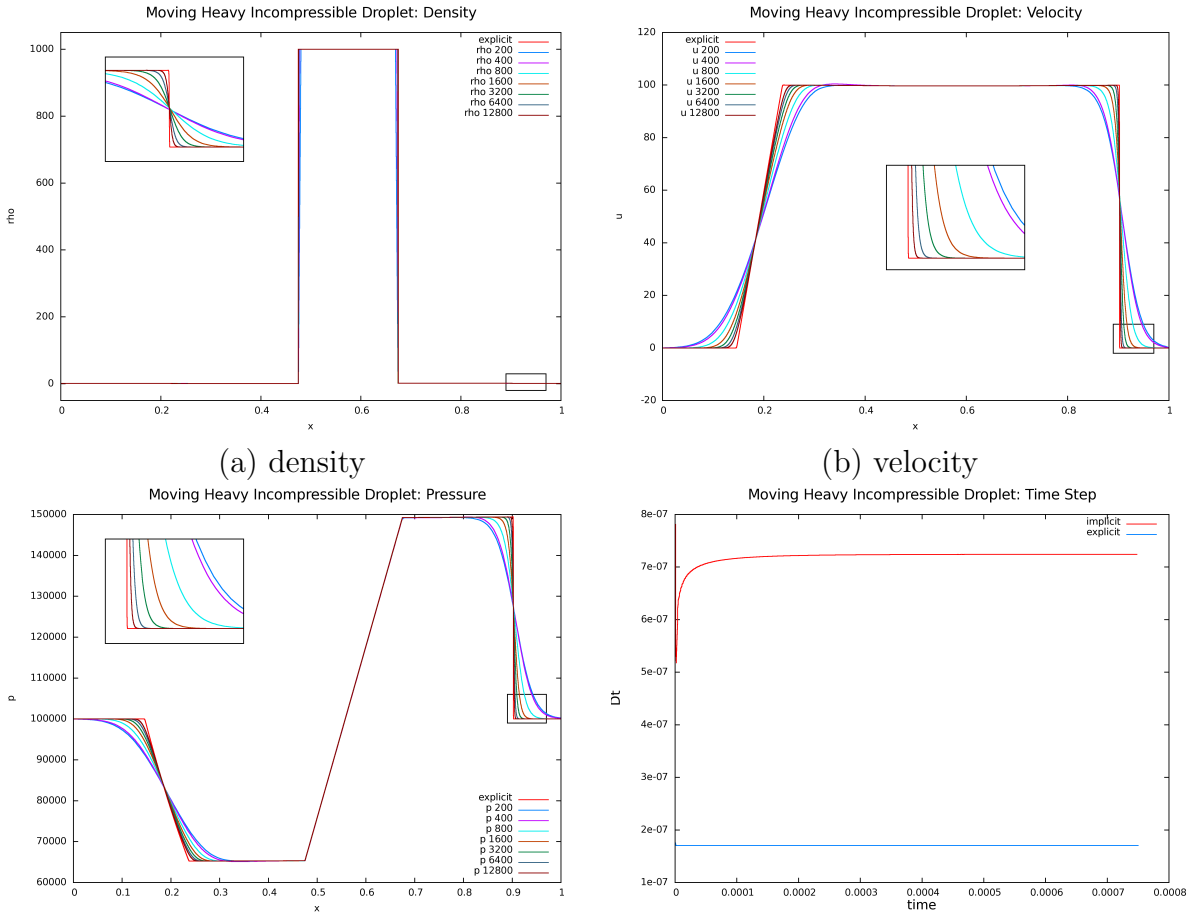
We used the gamma gas law equation of state $p = (\gamma - 1)\rho e$ and an outer loop of third order TVD Runge-Kutta in all these examples, noting that although the implicit treatment of terms inside the RK loop generally leads to a loss of third order time accuracy, greater stability is achieved. All examples use a CFL number of .5.

4.5.1 One dimensional examples

Consider a computational domain of [0 m, 1 m]. The domain is filled with a compressible gas with $\rho = 1.226$ kg/m^3, $u = 0$ m/s, and $p = 10^5$ Pa. An incompressible droplet of length .2 m is located at the center of the domain with $\rho = 1000$ kg/m^3, $u = 100$ m/s, and $p = 10^5$ Pa. Since the incompressible droplet is moving rightwards in an initially stationary gas, a shock wave forms in the gas ahead of it and a rarefaction wave forms in the gas behind it. Figure 4.2 shows the density, velocity, pressure and time step profiles, along with the fully explicit method of [7] run on the finest grid of resolution 12800 for the sake of comparison. (Note that our implementation of [7] gave a different result for this example, although agrees with [7] for all other examples that they ran in both one and two spatial dimensions, leading us to believe that there is probably a typo in the description of this example in [7].) Note that our method converges to the highly refined explicit result as the grid is refined. Figure 4.3 shows similar results when the density of the incompressible droplet is $\rho = 10$ kg/m^3. The compressible gas slows down the lighter droplet faster and as a result secondary rarefaction waves stretch between the droplet and the shock and rarefaction waves.

Note that our method can take a time step that is four times larger, however, the cost of each time step is slightly larger because the compressible degrees of freedom have been added to the incompressible Poisson solver. The overall speedup in wall clock time will generally depend on the ratio of increased cost per time step as compared to the decrease in the number of time steps - in this particular example, the code was approximately three times faster. For some problems the speedups can be significantly larger especially when one cares about phenomena that occur after the compressible flow relaxes to smaller velocities - in this case the time steps could increase by several orders of magnitude.

Consider a computational domain of [0 m, 1 m] filled with a compressible fluid with density $\rho = 1.58317$ kg/m^3, velocity $u = 0$ m/s, and pressure $p = 98066.5$ Pa. An incompressible droplet of length .2 m is initially located at the center of the domain with $\rho = 1000$ kg/m^3, $u = 0$ m/s, and $p = 98066.5$ Pa. A shock wave is initially

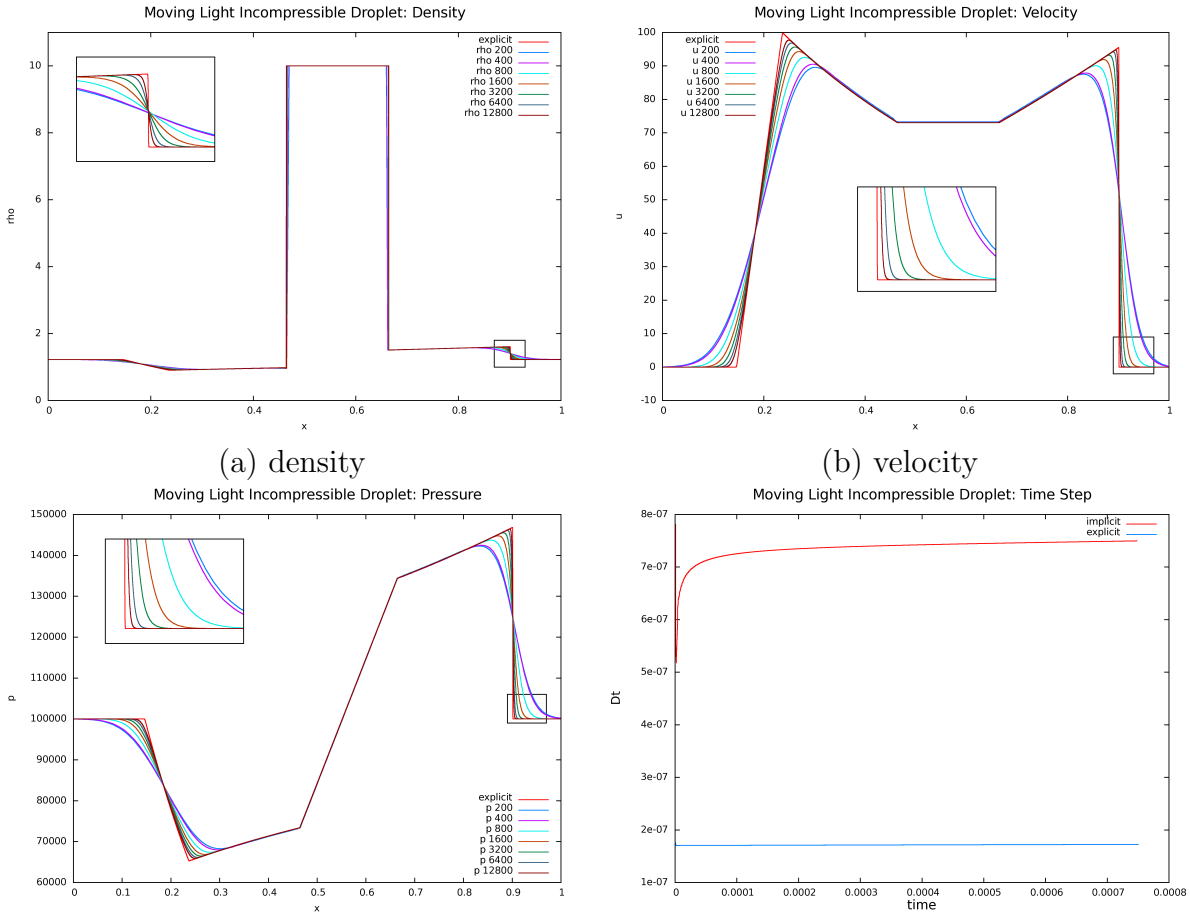


(a) density (b) velocity

(c) pressure (d) time step

Figure 4.2: Numerical results for the moving incompressible droplet example, where a droplet of density 1000 kg/m^3 is travelling to the right in an initially stationary compressible fluid at $t = 7.5 \times 10^{-4}$ s. Note that the profiles converge to those generated using the partitioned method of [7].

located at $x = .1$ m with a post-shock state of $\rho = 2.124$ kg/m^3, $u = 89.981$ m/s, and $p = 148407.3$ Pa to the left of $x = .1$ m. The shock wave travels to the right impinging on the incompressible droplet, causing both reflected and transmitted waves as shown in Figure 4.4 at $t = 1.75 \times 10^{-3}$ s. Note that the transmitted wave is too weak to be seen in this example, although it can be clearly seen in Figure 4.5 where the incompressible droplet has density $\rho = 10$ kg/m^3.



(a) density (b) velocity

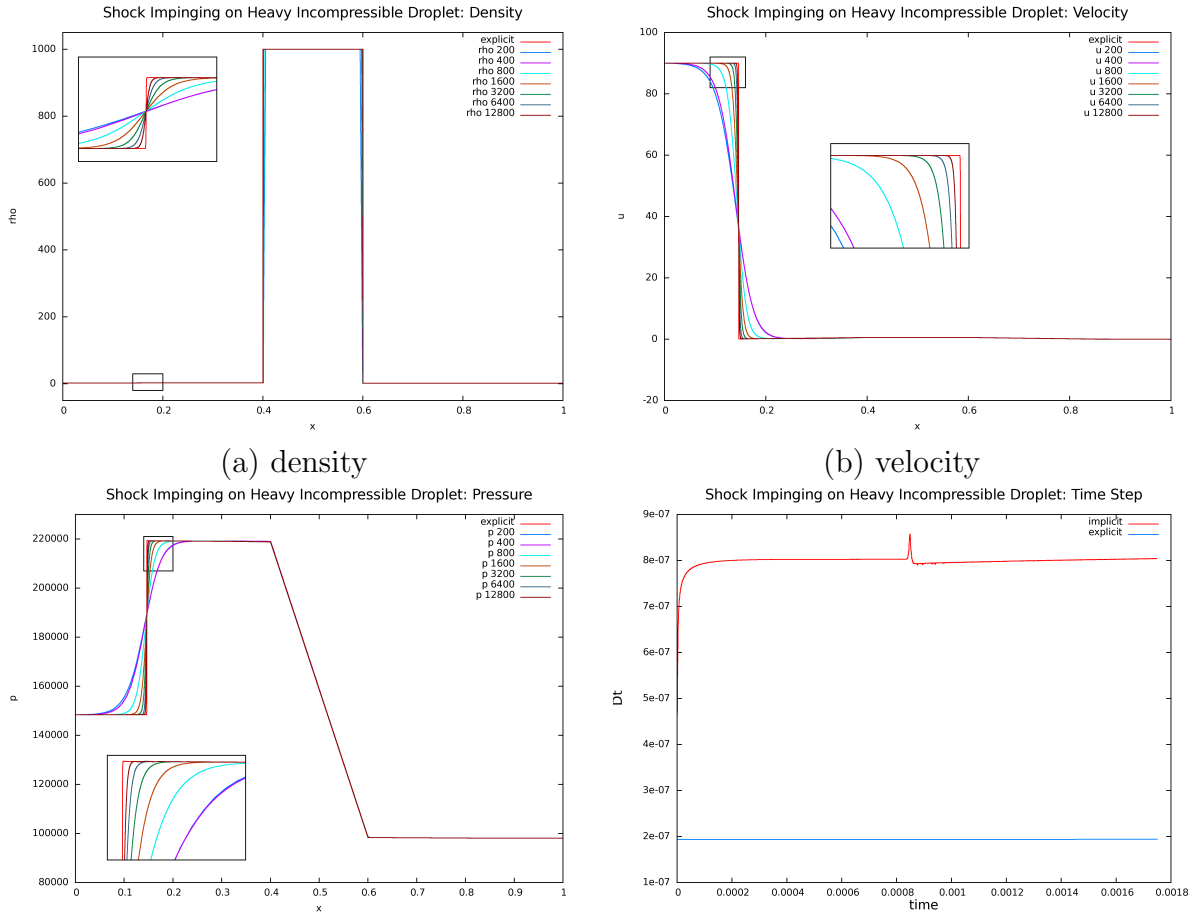
(c) pressure (d) time step

Figure 4.3: Numerical results for the moving incompressible droplet example, where a droplet of density 10 kg/m^3 is travelling to the right in an initially stationary compressible fluid at $t = 7.5 \times 10^{-4}$ s. Note that the profiles converge to those generated using the partitioned method of [7].

4.5.2 Two dimensional examples

All two-dimensional examples include the effects of viscosity and surface tension with coefficients $\mu = .001137$ kg/ms and $\sigma = .0728$ kg/s^2. These effects are not present in the one dimensional examples shown in Section 4.5.1 because the incompressible flow has constant velocity and the interface has no curvature.

Consider a computational domain of [0 m, 1 m]×[0 m, 1 m]. Similar to the one

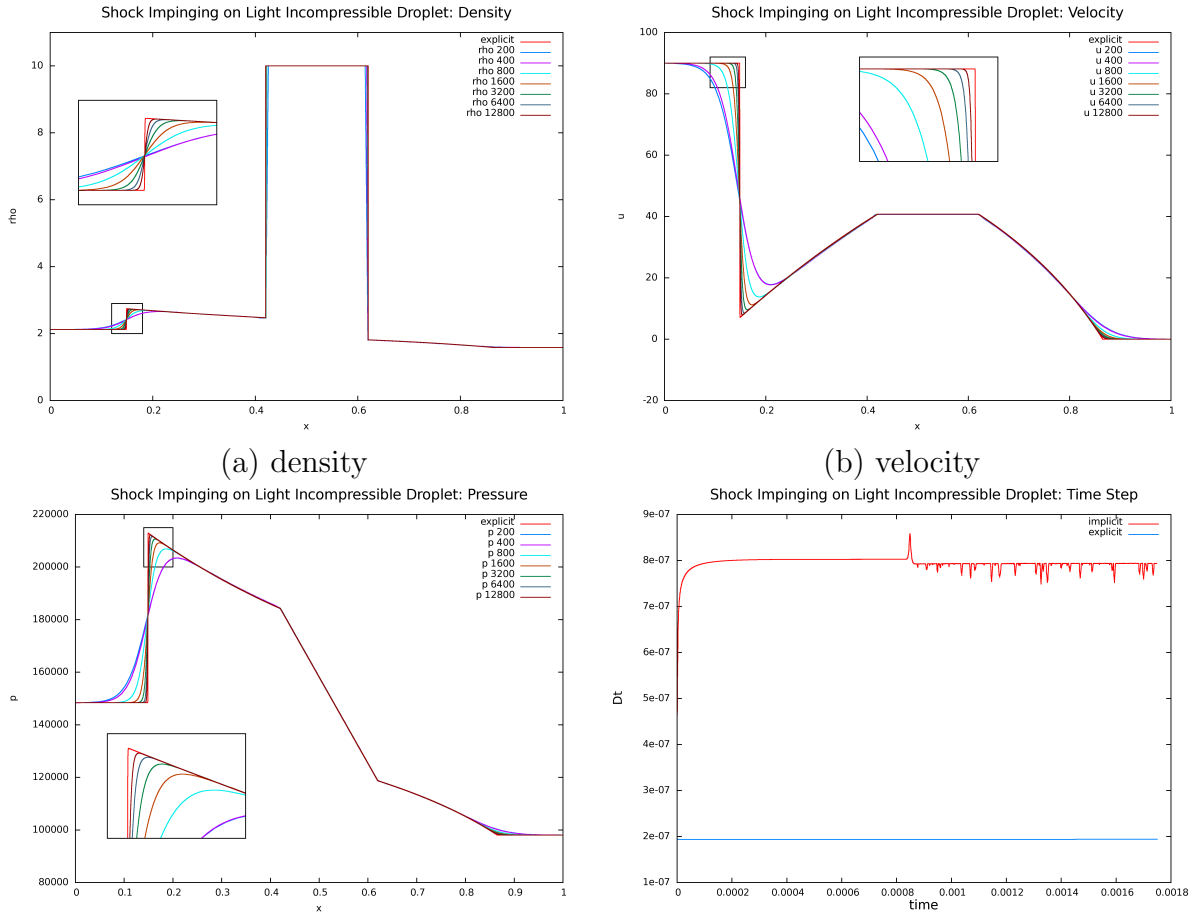


(a) density (b) velocity

(c) pressure (d) time step

Figure 4.4: Numerical results for the shock impinging on a heavy incompressible droplet example at $t = 1.75 \times 10^{-3}$ s, where a shock wave initially located at $x = .1$ m travels to the right impinging on an incompressible droplet of density 1000 kg/m^3 generating both reflected and transmitted waves. Note that the profiles converge to those generated using the partitioned method of [7].

dimensional case, the domain is filled with a compressible gas with $\rho = 1.226$ kg/m^3, $u = v = 0$ m/s, and $p = 10^5$ Pa. An incompressible droplet of radius .2 m is located at the center of the domain with $\rho = 1000$ kg/m^3, $u = 100$ m/s, $v = 0$ m/s, and $p = 10^5$ Pa. Since the compressible gas is initially stationary and the droplet is moving rightwards, a shock wave forms in the gas in front of it, and a rarefaction wave forms in the gas behind it. Figure 4.6(a) shows 50 equally spaced pressure



(a) density (b) velocity

(c) pressure (d) time step

Figure 4.5: Numerical results for the shock impinging on a light incompressible droplet example at $t = 1.75\times10^{-3}$ s, where a shock wave initially located at $x = .1$ m travels to the right impinging on an incompressible droplet of density 10 kg/m^3 generating both reflected and transmitted waves. Note that the profiles converge to those generated using the partitioned method of [7].

contours between $.75 \times 10^5$ Pa and 1.5×10^5 Pa on a 1600×1600 grid at $t = 5 \times 10^{-4}$ s. Figure 4.6(b) shows the pressure contour of 1.1×10^5 at various grid resolutions to show that the numerical profiles generated using our method converge to those generated using the fully explicit method of [7] under grid refinement. The velocity field is shown in Figure 4.6(c) where the incompressible velocities are shown in blue and the compressible ones are shown in red. Figure 4.6(d) shows the initial location

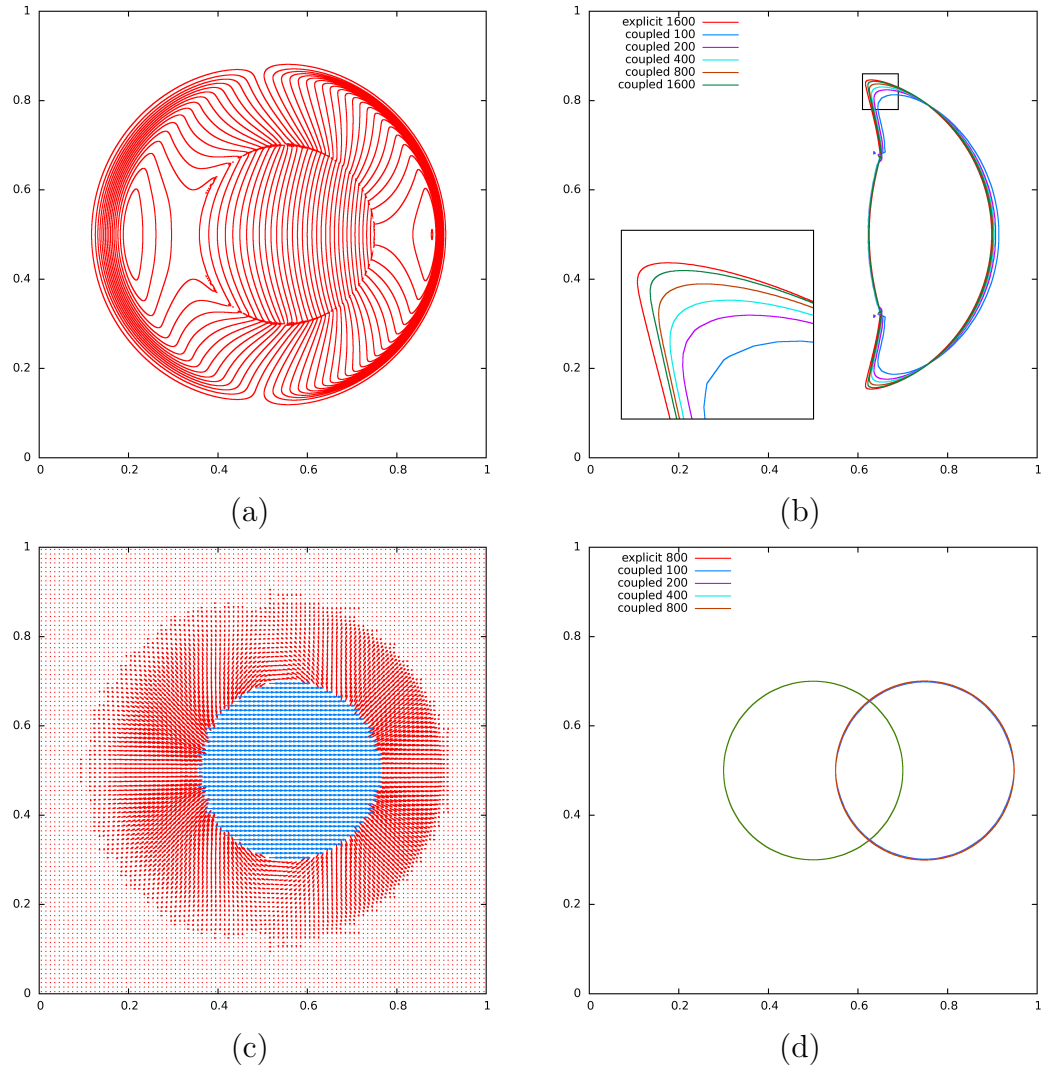


Figure 4.6: The moving incompressible droplet example in two spatial dimensions inside a computational domain of [0 m, 1 m]×[0 m, 1 m], where an incompressible droplet of density 1000 kg/m^3 and initial radius .2 m is travelling to the right in an initially stationary compressible fluid. (a) 50 equally spaced pressure contours between $.75 \times 10^5$ Pa and 1.5×10^5 Pa on a 1600×1600 grid at $t = 5 \times 10^{-4}$ s, (b) pressure contour of 1.1×10^5 at $t = 5 \times 10^{-4}$ s under grid refinement to illustrate convergence to the result generated using the partitioned method of [7], (c) velocity field at $t = 5 \times 10^{-4}$ s where the incompressible velocities are shown in blue, and compressible velocities are shown in red, and (d) the zero level set under grid refinement at $t = 2.5 \times 10^{-3}$ s as compared to its initial location.

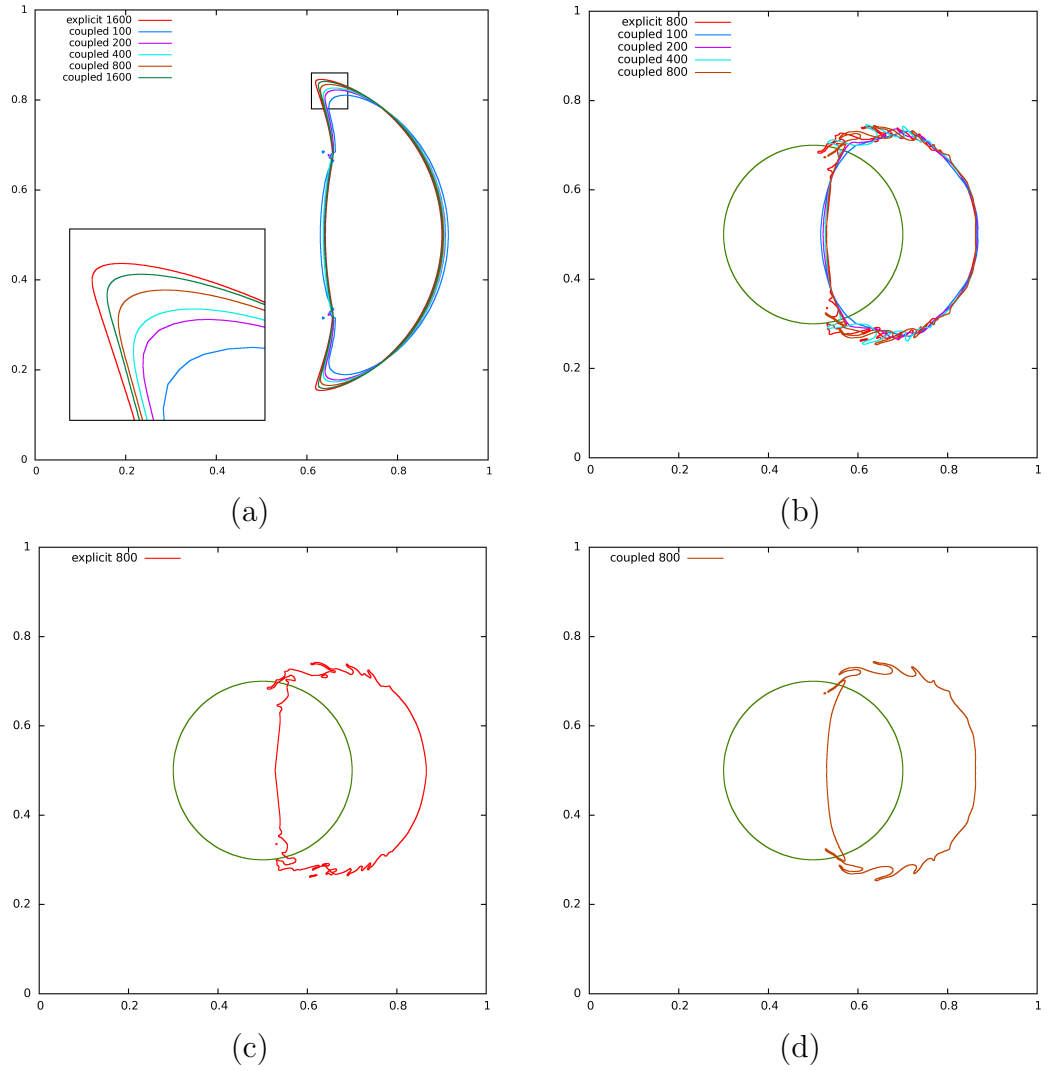
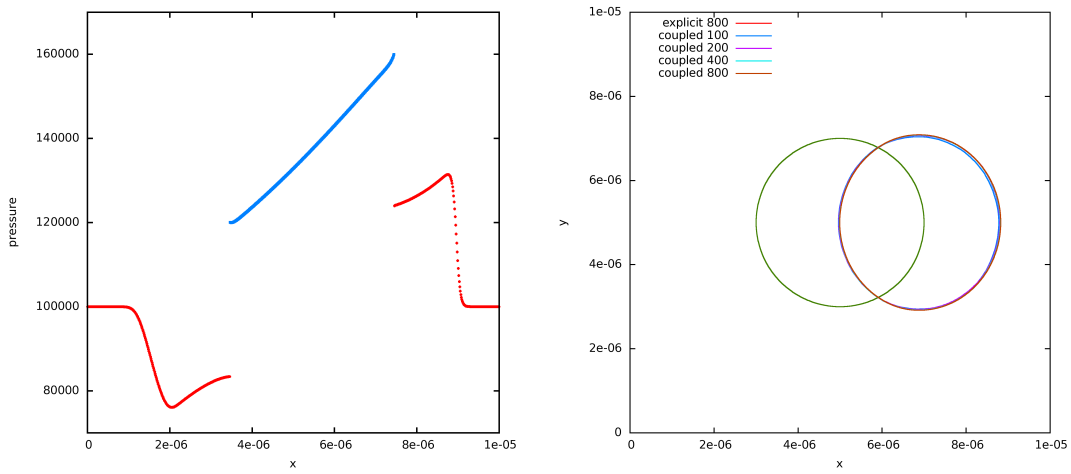


Figure 4.7: The moving incompressible droplet example in two spatial dimensions inside a computational domain of [0 m, 1 m]×[0 m, 1 m], where an incompressible droplet of density 10 kg/m^3 and initial radius .2 m is travelling to the right in an initially stationary compressible fluid. (a) pressure contour of 1.1×10^5 at $t = 5\times10^{-4}$ s under grid refinement to illustrate convergence to the result generated using the partitioned method of [7], (b) the zero level set under grid refinement at $t = 2.5\times10^{-3}$ s as compared to its initial location, (c) the zero level set on a grid of resolution 800×800 at $t = 2.5\times10^{-3}$ s as compared to its initial location using the fully explicit partitioned solver of [7], and (d) the zero level set on a grid of resolution 800×800 at $t = 2.5 \times 10^{-3}$ s as compared to its initial location using our monolithic solver.

of the zero level set as compared to its location at $t = 2.5 \times 10^{-3}$ s. Figure 4.7(a)-(d) show the results for the case when the incompressible droplet has $\rho = 10$ kg/m^3. Note that the lighter droplet undergoes larger deformation and also slows down at a faster rate. Also note that, as observed in [7], the computations for the lighter droplet show signs of Kelvin-Helmholtz instability as is apparent by the wiggles in the interface location shown in Figure 4.7(b). This effect is less apparent on coarser grids because of the artificial damping due to numerical viscosity.



(a) (b)

Figure 4.8: The moving incompressible droplet example in two spatial dimensions inside a computational domain of [0 m, 1×10^{-5} m]$\times$[0 m, 1×10^{-5} m], where an incompressible droplet of density 10 kg/m^3 and initial radius $.2 \times 10^{-5}$ m is travelling to the right in an initially stationary compressible fluid. (a) one dimensional cross-section of the pressure on an 800×800 grid at $t = 5 \times 10^{-9}$ s, where the pressure in the incompressible region is shown in blue and that in the compressible region is shown in red, and (b) the zero level set under grid refinement at $t = 2.5 \times 10^{-8}$ s as compared to its initial location.

To demonstrate the effects of surface tension and viscosity, we also shrunk the domain to [0 m, 1×10^{-5} m]$\times$[0 m, 1×10^{-5} m] for the case when the incompressible droplet has density $\rho = 10$ kg/m^3. Figure 4.8(a) shows a one dimensional cross-section of the pressure at $t = 5 \times 10^{-9}$ s, where the pressure in the incompressible region is shown in blue and that in the compressible region is shown in red. Note the jump in pressure across the interface due to surface tension effects. Figure 4.8(b) shows

the initial location of the zero level set as compared to its location at $t = 2.5 \times 10^{-8}$ s. Note that the smaller droplet is deformed less and has a more spherical shape as compared to the larger droplet shown in Figure 4.7(d).

Consider a computational domain of [0 m, 1 m]×[0 m, 1 m]. Similar to the one dimensional case, the domain is filled with a compressible fluid with $\rho = 1.58317$ kg/m^3, $u = v = 0$ m/s, and $p = 98066.5$ Pa. An incompressible droplet with initial state $\rho = 10$ kg/m^3, $u = v = 0$ m/s, and $p = 98066.5$ Pa with radius .2 m is located at the center of the domain. A shock wave is initially located at $x = .1$ m with a post-shock state of $\rho = 2.124$ kg/m^3, $u = 89.981$ m/s, $v = 0$ m/s, and $p = 148407.3$ Pa to the left of $x = .1$ m. The shock wave travels to the right impinging on the incompressible droplet, generating both reflected and transmitted waves. Figure 4.9(a) shows 50 equally spaced pressure contours between 1×10^5 Pa and 1.8×10^5 Pa on a 1600×1600 grid at $t = 1.25 \times 10^{-3}$ s. Figure 4.9(b) shows the pressure contour of 1.62×10^5 Pa under grid refinement to illustrate convergence to those generated using the fully explicit method of [7]. Figure 4.9(c) shows the velocity field, where the incompressible velocities are shown in blue, and the compressible velocities are shown in red. Figure 4.9(d) shows the initial location of the zero level set as compared to its location at $t = 2.5 \times 10^{-3}$ s under grid refinment. Note that the computations on the finer grids also show signs of Kelvin-Helmholtz instability.

4.6 Constant temperature formulation

We make an isothermal assumption, where the equations for conservation of mass and momentum form a closed system and the equation for conservation of energy decouples. In this case, the gamma gas law equation of state can be rewritten as $p = B\rho$, where B is essentially $(\gamma - 1)e$ which we set equal to the atmospheric pressure, i.e., $B = p_{\mathsf{atm}}/\rho_{\mathsf{atm}}$ m^3/kg, as described in Chapter 2. For this equation of

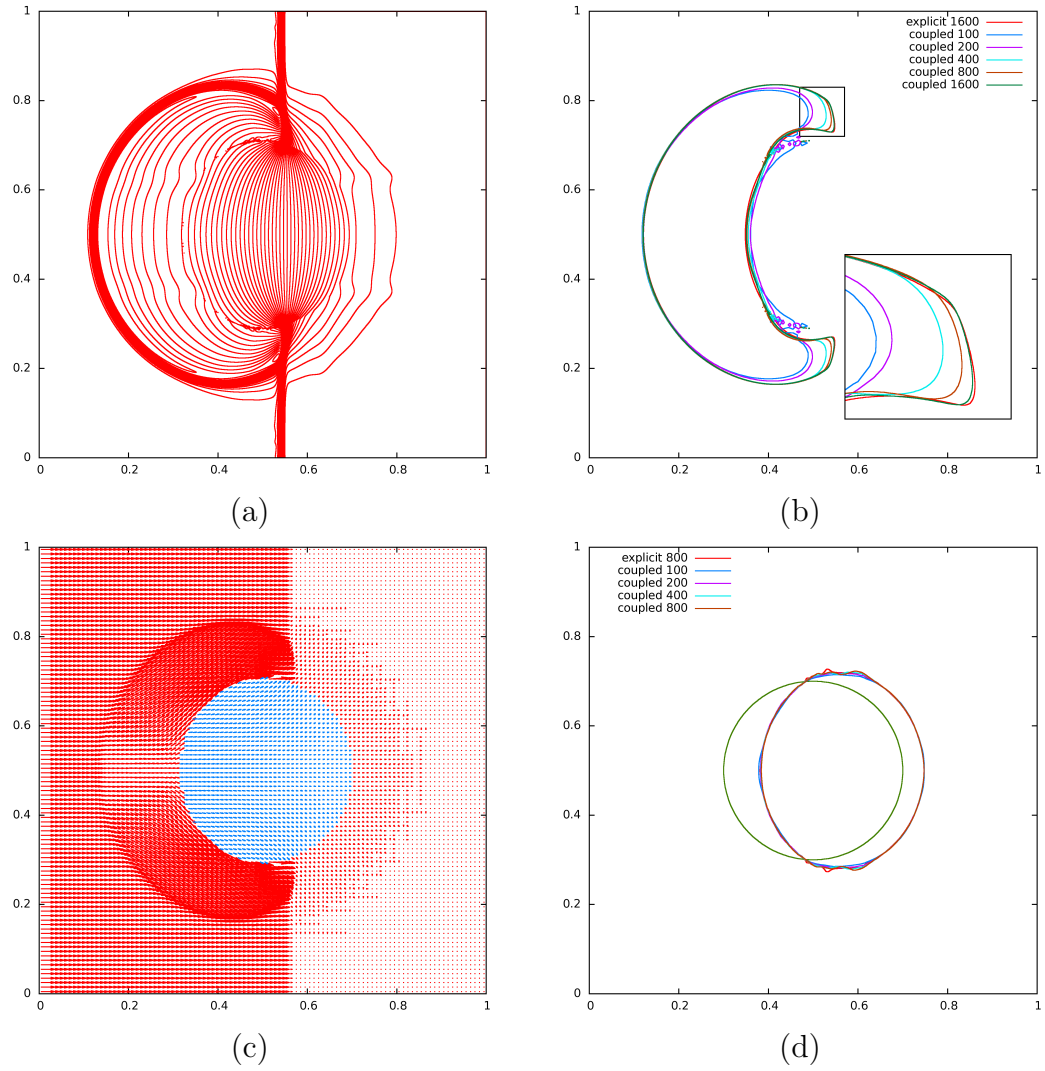


Figure 4.9: The shock impinging on an incompressible droplet example in two spatial dimensions inside a computational domain of [0 m, 1 m]×[0 m, 1 m], where a shock wave initially located at $x = .1$ m travels to the right impinging on an incompressible droplet with $\rho = 10$ kg/m^3 and initial radius .2 m generating both reflected and transmitted waves. (a) 50 equally spaced contours between 1×10^5 Pa and 1.8×10^5 Pa on a 1600×1600 grid at $t = 1.25 \times 10^{-3}$ s, (b) pressure contour of 1.62×10^5 Pa under grid refinement at $t = 1.25 \times 10^{-3}$ s to illustrate convergence to the result generated using the partitioned method of [7], (c) velocity field at $t = 1.25 \times 10^{-3}$ s where the incompressible velocities are shown in blue, and the compressible velocities are shown in red, and (d) the zero level set under grid refinement at $t = 2.5 \times 10^{-3}$ s as compared to its initial location.

state, substituting equation (3.3), equation (4.8) becomes

$$p^{n+1} - \Delta t^2 B\rho^n \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) = B\rho^\star - \Delta t B\rho^n \nabla \cdot \hat{u}^\star \tag{4.31}$$

where $p^\star$ has been replaced with $B\rho^\star$. Note that, as mentioned in Section 4.1, $\rho^\star = \rho^{n+1}$, and thus $p^\star = B\rho^\star = B\rho^{n+1}$. Furthermore, if ρc^2 in equation (4.8) is taken to be at time t^{n+1} instead of t^n, we can replace ρ^n with ρ^{n+1} in equation (4.31) and divide by $B\rho^{n+1}$ to obtain

$$\frac{1}{\Delta t^2 B\rho^{n+1}} \hat{p}^{n+1} - \nabla \cdot \left(\frac{\nabla \hat{p}^{n+1}}{\rho^{n+1}} \right) = \frac{1}{\Delta t} - \nabla \cdot \hat{u}^\star \tag{4.32}$$

which is the continuous analog of equation (4.9) for this equation of state.

4.6.1 Numerical results

Now reconsider the examples presented in Section 4.5.1 simulated using $p = B\rho$ as the equation of state. We use the same ambient conditions for density and velocity in all the examples, noting that the ambient pressures will be different since pressure depends on density. Also, for the examples with prescribed shocks we choose to match the shock speed prescribing a post-shock state of $\rho = 1.97705$ kg/m^3 and $u = 70.4023$ m/s. Figures 4.12-4.15 show the numerical profiles generated using equation (4.32) - a high resolution comparison to the fully explicit method of [7] is also shown in the results. Note that the incompressibility assumption confines shock waves to the compressible fluid, however, if one wishes to study phenomena such as when shock waves impinge upon the compressible fluid or shock-induced bubble collapse then fully two-phase compressible flow models should be used [32, 31].

Next consider the oscillating bubble problems introduced in Section 2.2. Here we use the two-way coupled simulation techniques proposed in this section which couple the incompressible flow solver to a full compressible flow solver that includes shocks and rarefactions, albeit a somewhat simplified isothermal $p = B\rho$ equation of state.

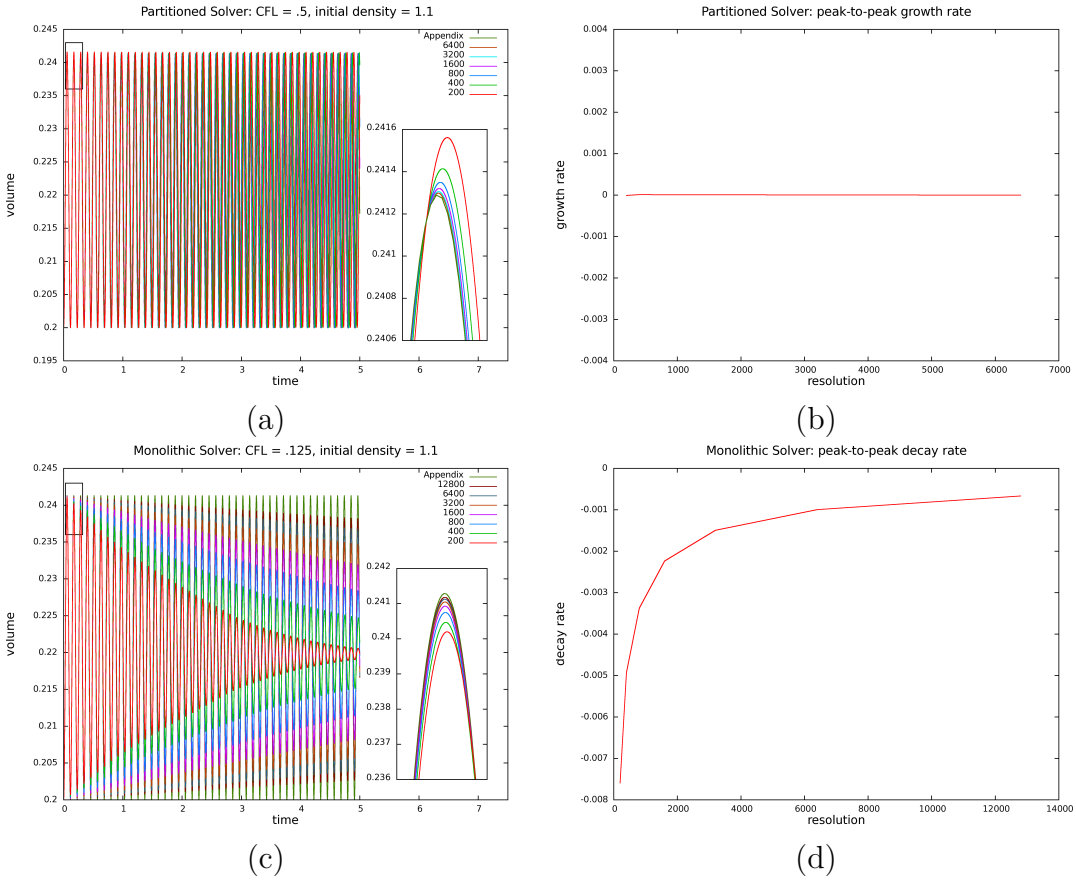


Figure 4.10: Numerical profiles for bubble volume over time along with peak-to-peak growth/decay rates for the oscillating bubble problem in one spatial dimension with equation of state $p = B\rho$ generated by (a,b) the partitioned method of [7], and (c,d) the monolithic solver.

resolution1	resolution2	resolution3	convergence order
200	400	800	1.1968
400	800	1600	1.1644
800	1600	3200	1.0506
1600	3200	6400	1.0000

Table 4.1: Convergence orders for the volume profiles generated by the partitioned method of [7] for the oscillating bubble problem in one spatial dimension using the equation of state $p = B\rho$.

resolution1	resolution2	resolution3	convergence order
200	400	800	-.08905
400	800	1600	.6740
800	1600	3200	.5688
1600	3200	6400	.6031
3200	6400	12800	.6033

Table 4.2: Convergence orders for the volume profiles generated by the monolithic solver for the oscillating bubble problem in one spatial dimension using the equation of state $p = B\rho$.

resolution1	resolution2	resolution3	convergence order
100	200	400	3.3236
200	400	800	2.5883

Table 4.3: Convergence orders for the volume profiles generated by the partitioned method of [7] for the oscillating bubble problem in two spatial dimensions using the equation of state $p = B\rho$.

resolution1	resolution2	resolution3	convergence order
100	200	400	.9151
200	400	800	.6104

Table 4.4: Convergence orders for the volume profiles generated by the monolithic solver for the oscillating bubble problem in two spatial dimensions using the equation of state $p = B\rho$.

Figure 4.10(a) shows the numerical profiles for the bubble volume over time generated using the partitioned method of [7], while Figure 4.10(c) shows the profiles generated using our proposed monolithic solver in one spatial dimension. Figures 4.10(b) and (d) show the corresponding peak-to-peak growth/decay rates under grid refinement while Tables 4.1 and 4.2 show the respective convergence orders. Note that the results converge to the "exact" solution under grid refinement in both cases. Figure 4.11 shows the results for a two dimensional oscillating bubble. The respective convergence orders are shown in Tables 4.3 and 4.4. Note that, unlike the one dimensional case, the method of [7] also damps the solution on coarser grids, although converging to

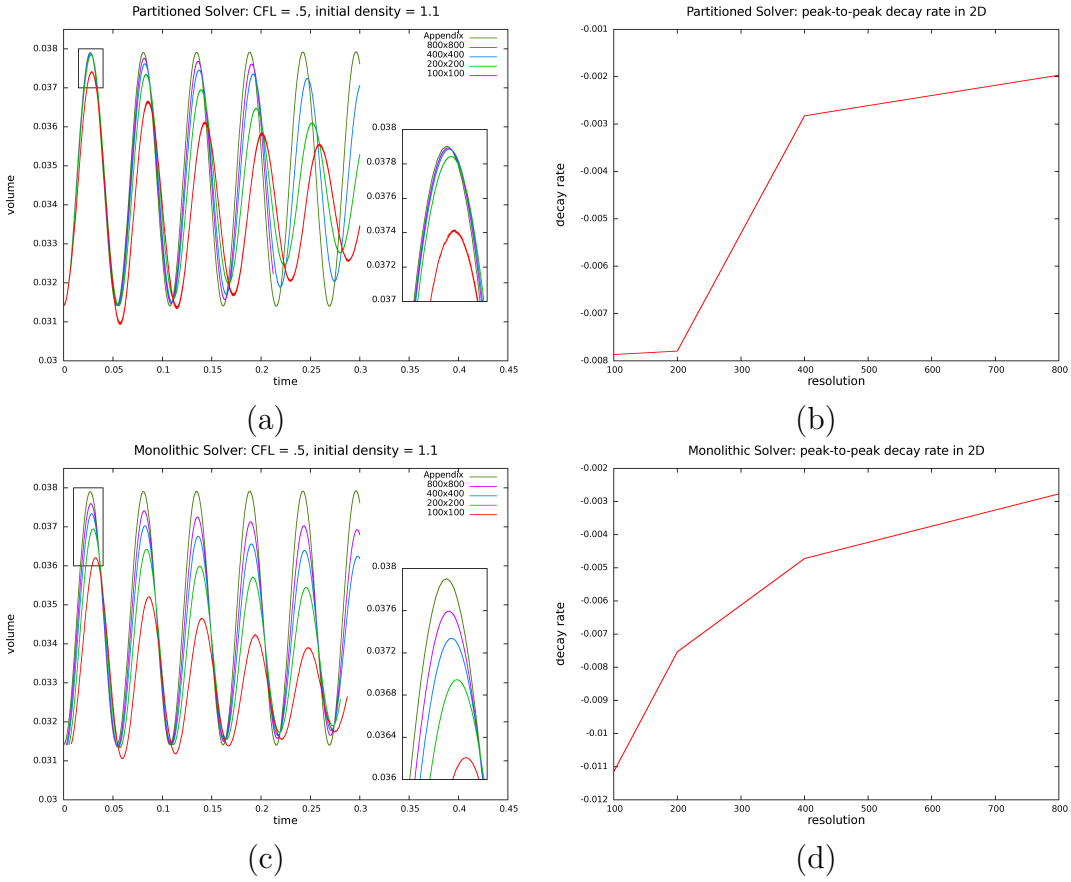
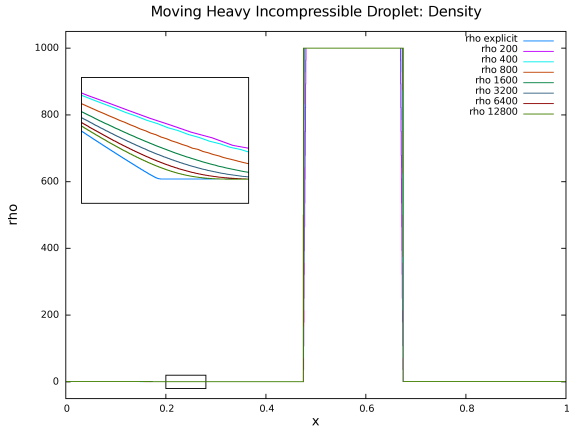
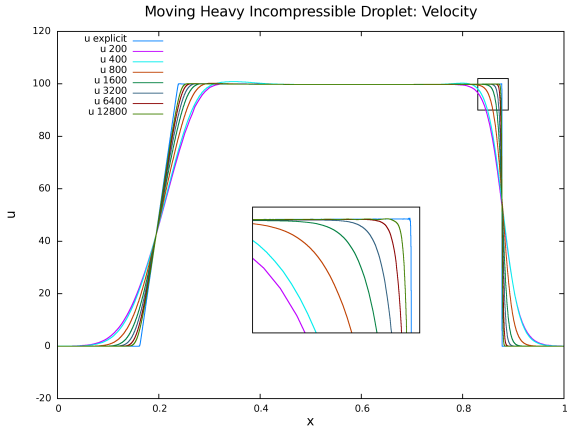


Figure 4.11: Numerical profiles for bubble volume over time along with peak-to-peak decay rates for the oscillating bubble problem in two spatial dimensions with equation of state $p = B\rho$ generated by (a,b) the partitioned method of [7], and (c,d) the monolithic solver. Note that the highest resolution simulation in (a) has not been run for the entire time length because of the significant computational overhead incurred by the partitioned scheme of [7].

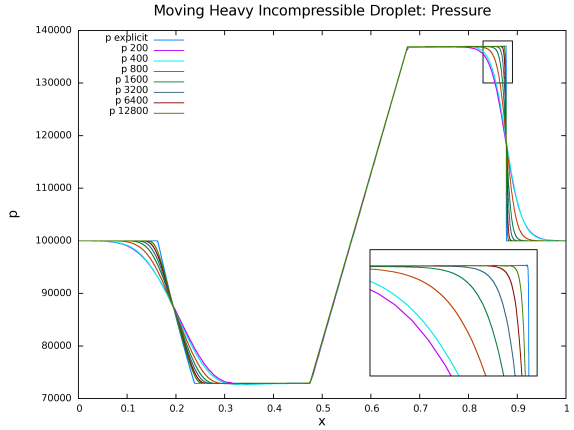
the "exact" solution under grid refinement.



(a) density

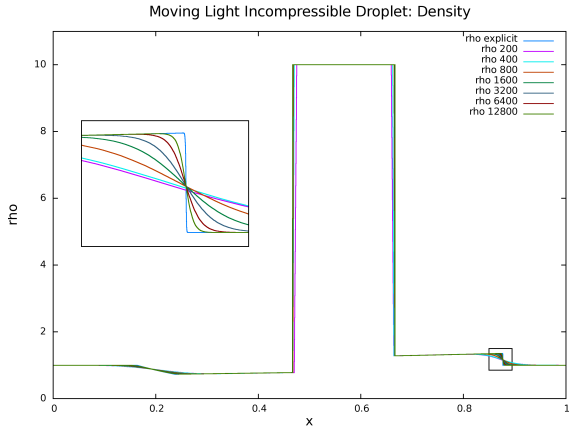


(b) velocity

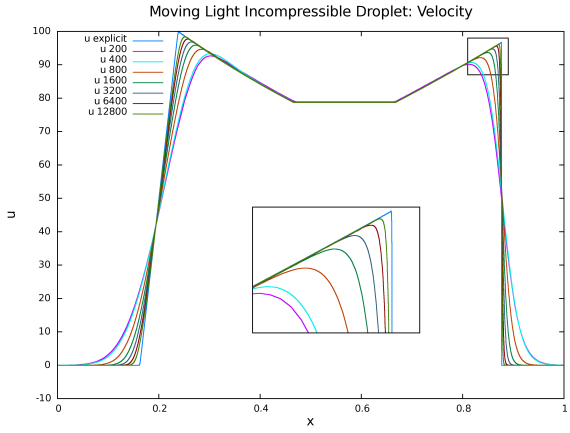


(c) pressure

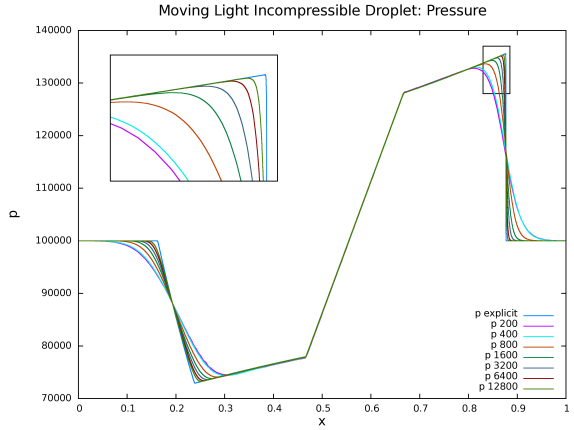
Figure 4.12: Numerical results for the moving incompressible droplet example, where a droplet of density 1000 kg/m^3 is travelling to the right in an initially stationary compressible fluid with equation of state $p = B\rho$ at $t = 7.5 \times 10^{-4}$ s. Note that the profiles converge to those generated using the partitioned method of [7].



(a) density

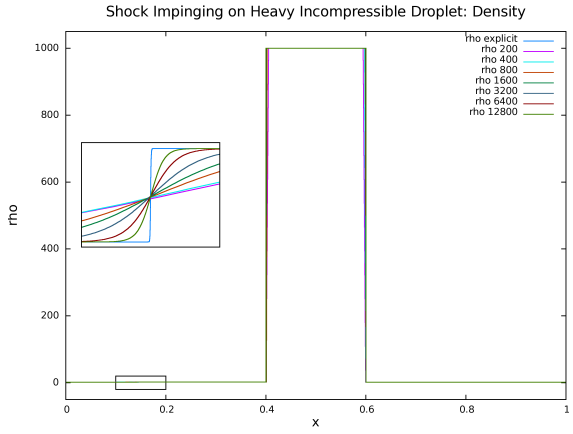


(b) velocity

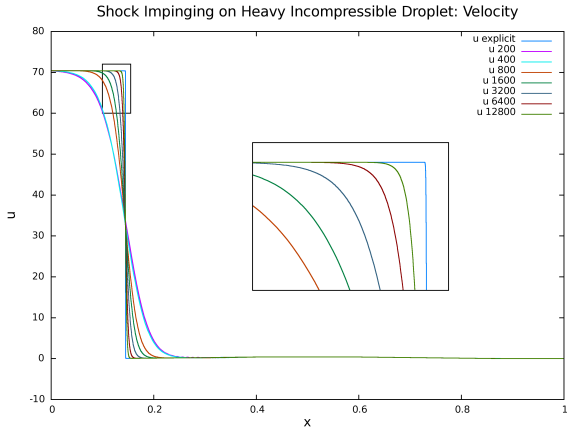


(c) pressure

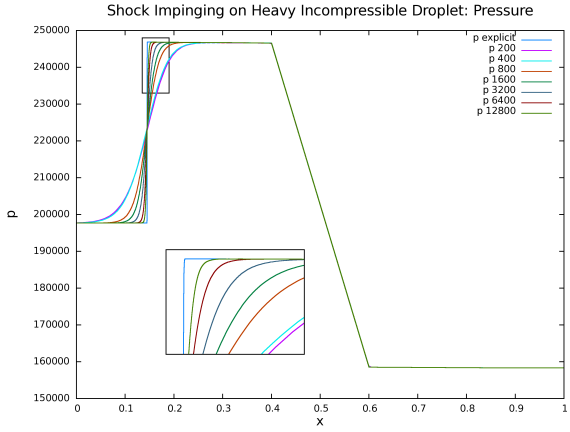
Figure 4.13: Numerical results for the moving incompressible droplet example, where a droplet of density 10 kg/m^3 is travelling to the right in an initially stationary compressible fluid with equation of state $p = B\rho$ at $t = 7.5 \times 10^{-4}$ s. Note that the profiles converge to those generated using the partitioned method of [7].



(a) density

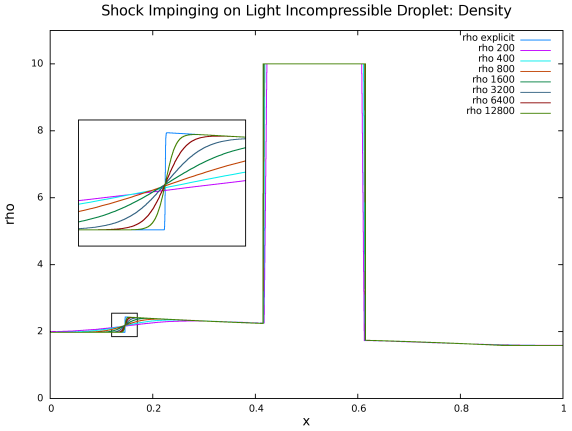


(b) velocity

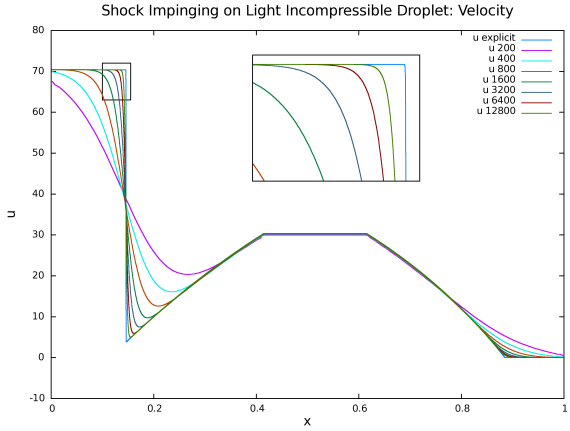


(c) pressure

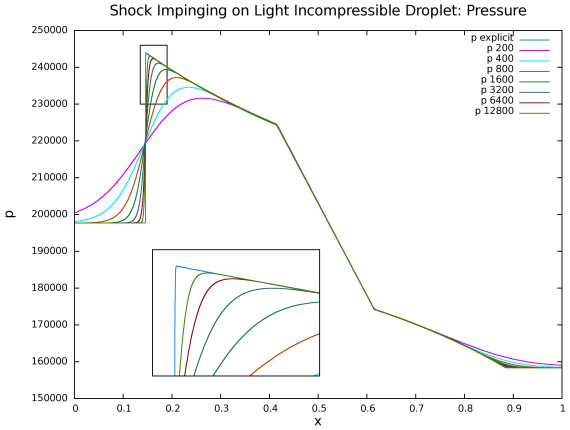
Figure 4.14: Numerical results for the shock impinging on a heavy incompressible droplet example with equation of state $p = B\rho$ at $t = 1.75 \times 10^{-3}$ s, where a shock wave initially located at $x = .1$ m travels to the right impinging on an incompressible droplet of density 1000 kg/m^3 generating both reflected and transmitted waves. Note that the profiles converge to those generated using the partitioned method of [7].



(a) density



(b) velocity



(c) pressure

Figure 4.15: Numerical results for the shock impinging on a heavy incompressible droplet example with equation of state $p = B\rho$ at $t = 1.75 \times 10^{-3}$ s, where a shock wave initially located at $x = .1$ m travels to the right impinging on an incompressible droplet of density 10 kg/m^3 generating both reflected and transmitted waves. Note that the profiles converge to those generated using the partitioned method of [7].

Chapter 5

Constant density and pressure

In this section continuing on from the isothermal assumption from Section 4.6, we additionally make constant density and constant pressure assumptions to arrive at our final method for simulating bubbles.

5.1 Constant density

We achieve constant density by redistributing the density in each bubble as the average density per bubble before the implicit pressure solve, exactly as is done for the partitioned solver in Section 2.3. Note that a spatially constant density field does not imply that pressure inside the bubbles is spatially constant as well.

Figure 5.1 shows the numerical profiles for the bubble volume over time along with peak-to-peak decay rates under grid refinement for the oscillating bubble problems. The respective convergence orders are shown in Tables 5.1 and 5.2. Note that the profiles converge to the "exact" solutions under grid refinement.

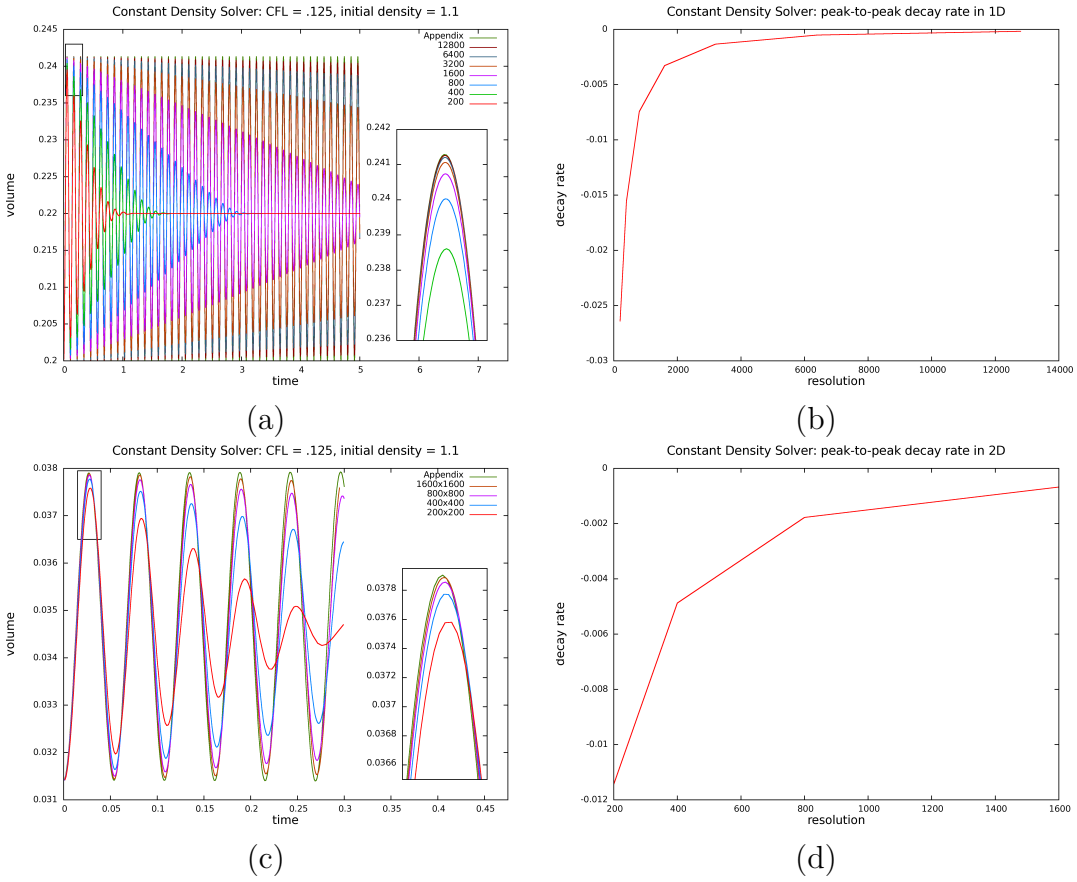


Figure 5.1: Numerical profiles for the bubble volume over time along with peak-to-peak decay rates under grid refinement generated using the monolithic solver with density redistribution to obtain constant density as well as an equation of state $p = B\rho$ for the oscillating bubble problem in (a,b) one spatial dimension, and (c,d) two spatial dimensions.

5.2 Constant pressure

We now present a simplified monolithic solver which solves for a constant pressure p^{n+1} inside the bubble with a single degree of freedom, i.e., spatially constant but time-varying. In matrix terms, this corresponds to taking the Poisson matrix for the full system and collapsing all the rows and columns corresponding to cells in the same bubble into a single row and column, adding overlapping matrix elements. This can also be seen as summing equation (4.32) over the entire bubble to obtain

resolution1	resolution2	resolution3	convergence order
200	400	800	.9170
400	800	1600	1.0061
800	1600	3200	1.1145
1600	3200	6400	1.2446
3200	6400	12800	1.2756

Table 5.1: Convergence orders for the volume profiles generated by the monolithic solver with density redistribution to obtain constant density for the oscillating bubble problem in one spatial dimension using the equation of state $p = B\rho$.

resolution1	resolution2	resolution3	convergence order
200	400	800	1.1948
400	800	1600	1.3827

Table 5.2: Convergence orders for the volume profiles generated by the monolithic solver with density redistribution to obtain constant density for the oscillating bubble problem in two spatial dimensions using the equation of state $p = B\rho$.

$$\sum_{i\in\Omega} \frac{1}{\Delta t^2 B\rho^{n+1}}\hat{p}^{n+1} - \sum_{i\in\Omega} \nabla\cdot\left(\frac{\nabla\hat{p}^{n+1}}{\rho^{n+1}}\right) = \sum_{i\in\Omega}\frac{1}{\Delta t} - \sum_{i\in\Omega}\nabla\cdot\hat{u}^{\star} \tag{5.1}$$

The first term on each side of the equality simply sums over the number of cells N inside the bubble. The last term can be modified by multiplying and dividing by the volume of a cell V_{c} and converting the volume sum to a surface sum along the MAC grid cell faces that border the bubble. If the average normal velocity on all these faces is $\bar{u}$ and the perimeter of all these faces is $\mathcal{P}$, we obtain

$$\frac{N}{\Delta t^2 B\rho^{n+1}}\hat{p}^{n+1} - \sum_{i\in\Omega} \nabla\cdot\left(\frac{\nabla\hat{p}^{n+1}}{\rho^{n+1}}\right) = \frac{N}{\Delta t} - \frac{\bar{u}\mathcal{P}}{V_{\mathsf{c}}} \tag{5.2}$$

Most of the terms in the final summation vanish since ∇p is zero within the bubble, leaving only terms corresponding to MAC grid faces that surround the bubble volume. For each of these faces ρ^{n+1} equals $\tilde{\rho}$ as defined in equation (4.18). Note that the first

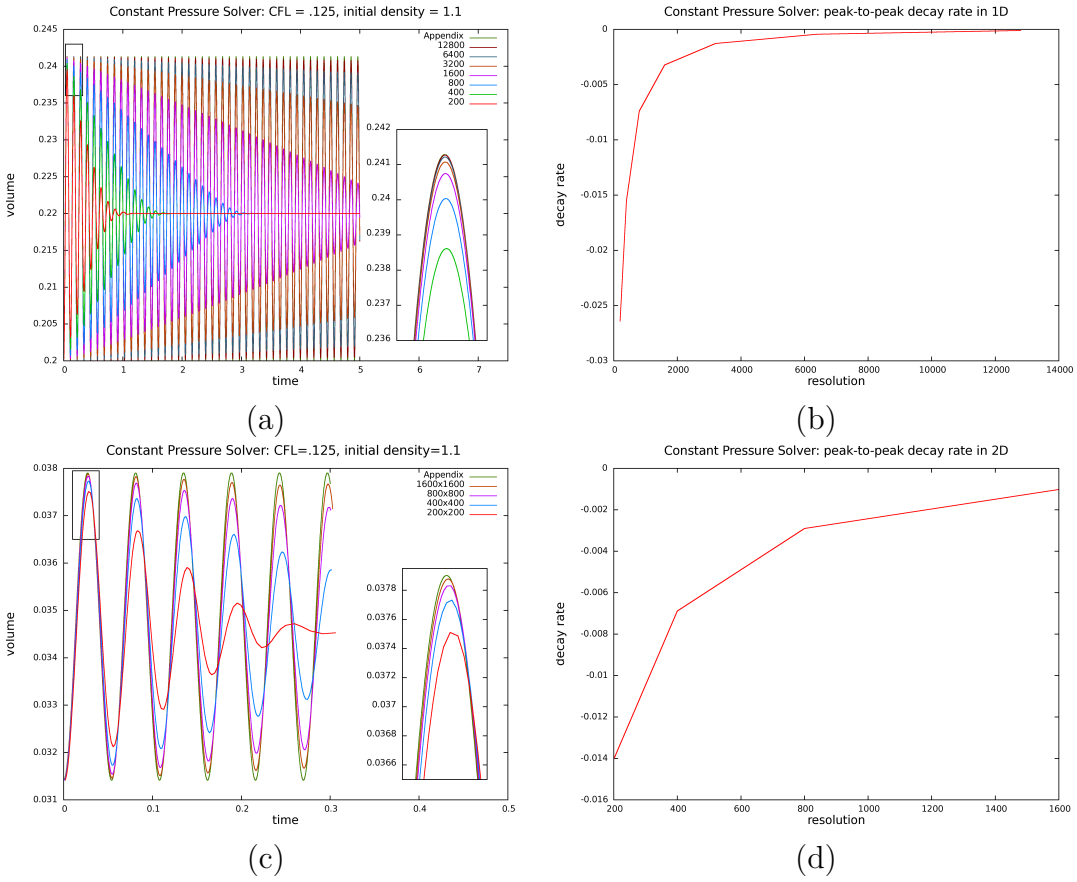


Figure 5.2: Numerical profiles for the bubble volume over time along with peak-to-peak decay rates under grid refinement generated by the constant pressure formulation where all pressure unknowns are collapsed to a single degree of freedom for each distinct bubble for the oscillating bubble problem in (a,b) one spatial dimension, and (c,d) two spatial dimensions.

term on each side of equation (5.2) is based on the number of cells within the bubble and the second term on each side of the equality is based on the number of MAC grid faces surrounding the bubble. Reminder, ρ^{n+1} in the first term denotes the bubble density.

Note that the constant pressure formulation does not produce pressure gradients within the bubble. Thus, the air velocities are updated through a second projection solve as per equations (2.15) and (2.16) using the boundary incompressible velocities,

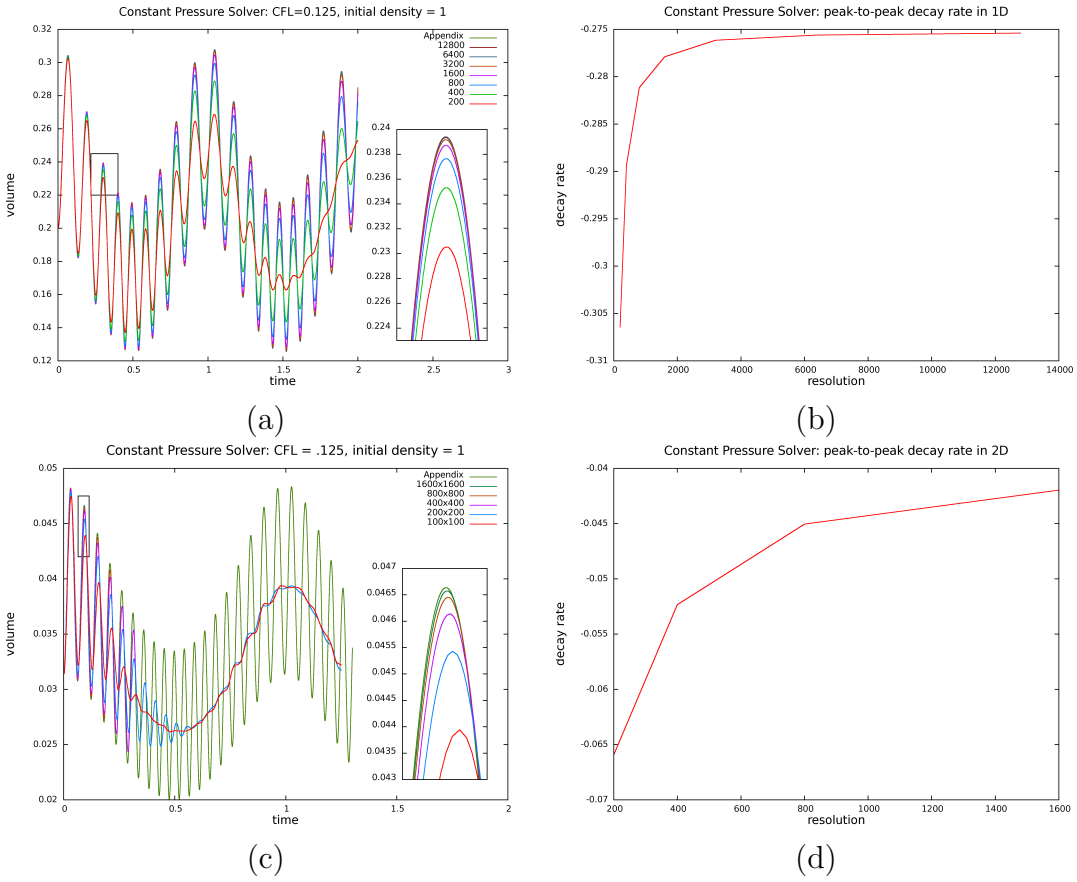


Figure 5.3: Numerical profiles for the bubble volume over time along with peak-to-peak decay rates under grid refinement generated by the constant pressure solver for the external pressure field problem in (a,b) one spatial dimension, and (c,d) two spatial dimensions. Note that the higher resolution simulations in (c) have not been run for the entire time length because the bubble breaks up into multiple bubbles due to Kelvin-Helmholtz instability, breaking our underlying assumptions and making further computation meaningless.

as outlined in Section 2.3. Figure 5.2 shows the numerical profiles for the bubble volume over time along with peak-to-peak decay rates for the oscillating bubble problems in one and two spatial dimensions respectively. Tables 5.3 and 5.4 show the respective convergence orders. Note that the profiles closely match those shown in Figures 3.3 and 5.1, verifying the correctness of the solver. We also consider the case where the initial density inside the bubble is 1 kg/m^3, and the outside air pressure is time-varying as $p_{\mathsf{atm}}(t) = Bf(t)$, where $f(t) = 1 - .2\sin(2\pi t + \pi/2)$. We refer to this

resolution1	resolution2	resolution3	convergence order
200	400	800	.9154
400	800	1600	1.0061
800	1600	3200	1.1145
1600	3200	6400	1.2136
3200	6400	12800	1.3847

Table 5.3: Convergence orders for the volume profiles generated by the constant pressure solver for the oscillating bubble problem in one spatial dimension.

resolution1	resolution2	resolution3	convergence order
200	400	800	1.1701
400	800	1600	1.0426

Table 5.4: Convergence orders for the volume profiles generated by the constant pressure solver for the oscillating bubble problem in two spatial dimensions.

problem as the *external pressure field* problem. Note that this problem is similar in spirit to the excitation of an isolated gas bubble from a planar sinusoidal wave, as studied in [17]. Figure 5.3 shows the resulting numerical profiles for the bubble volume over time along with peak-to-peak decay rates generated using our method in one and two spatial dimensions respectively, which converge to the "exact" solutions under grid refinement. Tables 5.5 and 5.6 show the respective convergence orders. Note that the "exact" solutions are computed using equations (A.4), (A.11) and (A.18) in the appendix which are also valid for time-varying external pressures.

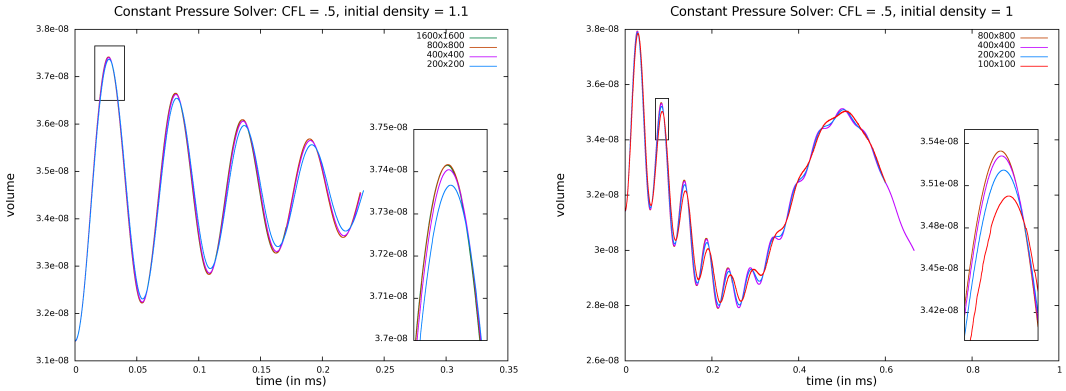
resolution1	resolution2	resolution3	convergence order
200	400	800	1.0208
400	800	1600	1.1272
800	1600	3200	1.2518
1600	3200	6400	1.3991
3200	6400	12800	1.5850

Table 5.5: Convergence orders for the volume profiles generated by the constant pressure solver for the external pressure field problem in one spatial dimension.

resolution1	resolution2	resolution3	convergence order
100	200	400	1.0661
200	400	800	1.1934
400	800	1600	1.3502

Table 5.6: Convergence orders for the volume profiles generated by the constant pressure solver for the external pressure field problem in two spatial dimensions.

For the oscillating bubble and external pressure field problems, we also considered the effects of surface tension and viscosity for a range of parameters from high to low, both individually and in combination. Figures 5.4(a) and (b) show the numerical profiles for the bubble volume over time for both these problems in two spatial dimensions with coefficients $\sigma = .0728$ kg/s^2 and $\mu = .2$ kg/ms. To heighten the effects of surface tension, the computational domain has been uniformly scaled down by a factor of 10^{-3}.



(a) (b)

Figure 5.4: Numerical profiles for the bubble volume over time under grid refinement generated by the constant pressure solver under the effects of surface tension and viscosity for the (a) oscillating bubble problem in two spatial dimensions, and (b) external pressure field problem in two spatial dimensions.

As further validation of our method, we also considered the rising bubble examples from [33]. Consider a computational domain of $[-1 \text{ m}, 1 \text{ m}] \times [-1 \text{ m}, 2 \text{ m}]$ which is initially filled with water except for a circular air bubble of radius 1/3 m centered at the origin with density $\rho = 1.226$ kg/m^3. The effects of surface tension and viscosity

are present with coefficients $\mu = .001137$ kg/ms and $\sigma = .0728$ kg/s^2. The edges of the computational domain have solid wall boundary conditions. Figure 5.5 shows the positions of the air bubble at $t = 0$, $t = .2$, $t = .35$ and $t = .5$ seconds under grid refinement. Note that the results are similar to those shown in Figures 3 and 4 from [33]. The finer grid computations at $t = .35$ s and $t = .5$ s show signs of Kelvin-Helmholtz instability, as noted in [33]. To demonstrate the effects of surface tension, we reduced the computational domain to $[-.01 \text{ m}, .01 \text{ m}] \times [-.01 \text{ m}, .02 \text{ m}]$ and the radius of the air bubble to 1/300 m. Figure 5.6 shows the positions of the air bubble at $t = 0$, $t = .02$, $t = .035$ and $t = .05$ seconds under grid refinement. Note that the results are similar to those shown in Figures 1 and 2 from [33].

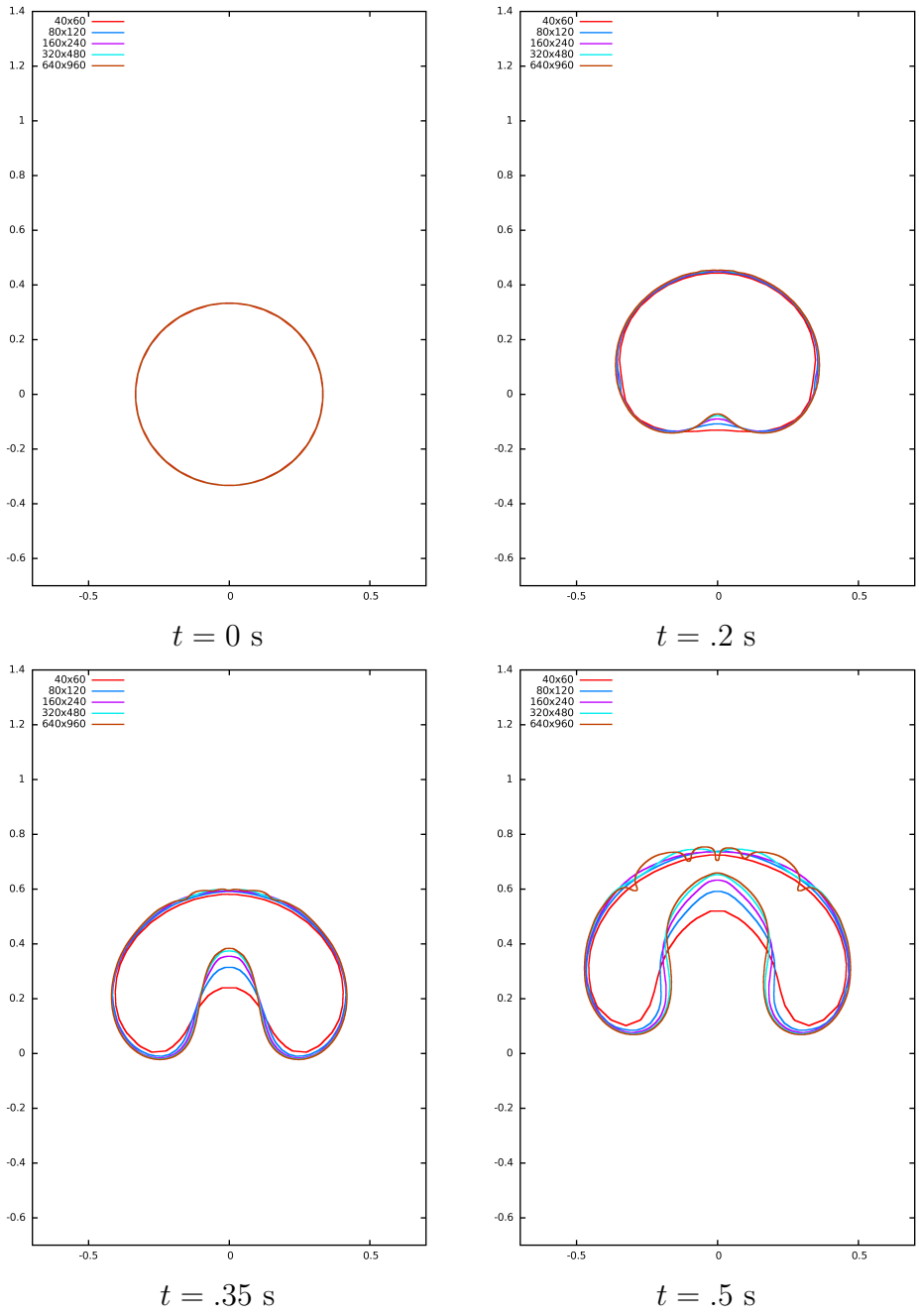


Figure 5.5: Level set profiles under grid refinement for an air bubble of initial radius 1/3 m rising inside a computational domain of $[-1 \text{ m}, 1 \text{ m}] \times [-1 \text{ m}, 2 \text{ m}]$ filled with water with solid wall boundary conditions. Note that the computations at time $t = .5$ s show signs of Kelvin-Helmholtz instability.

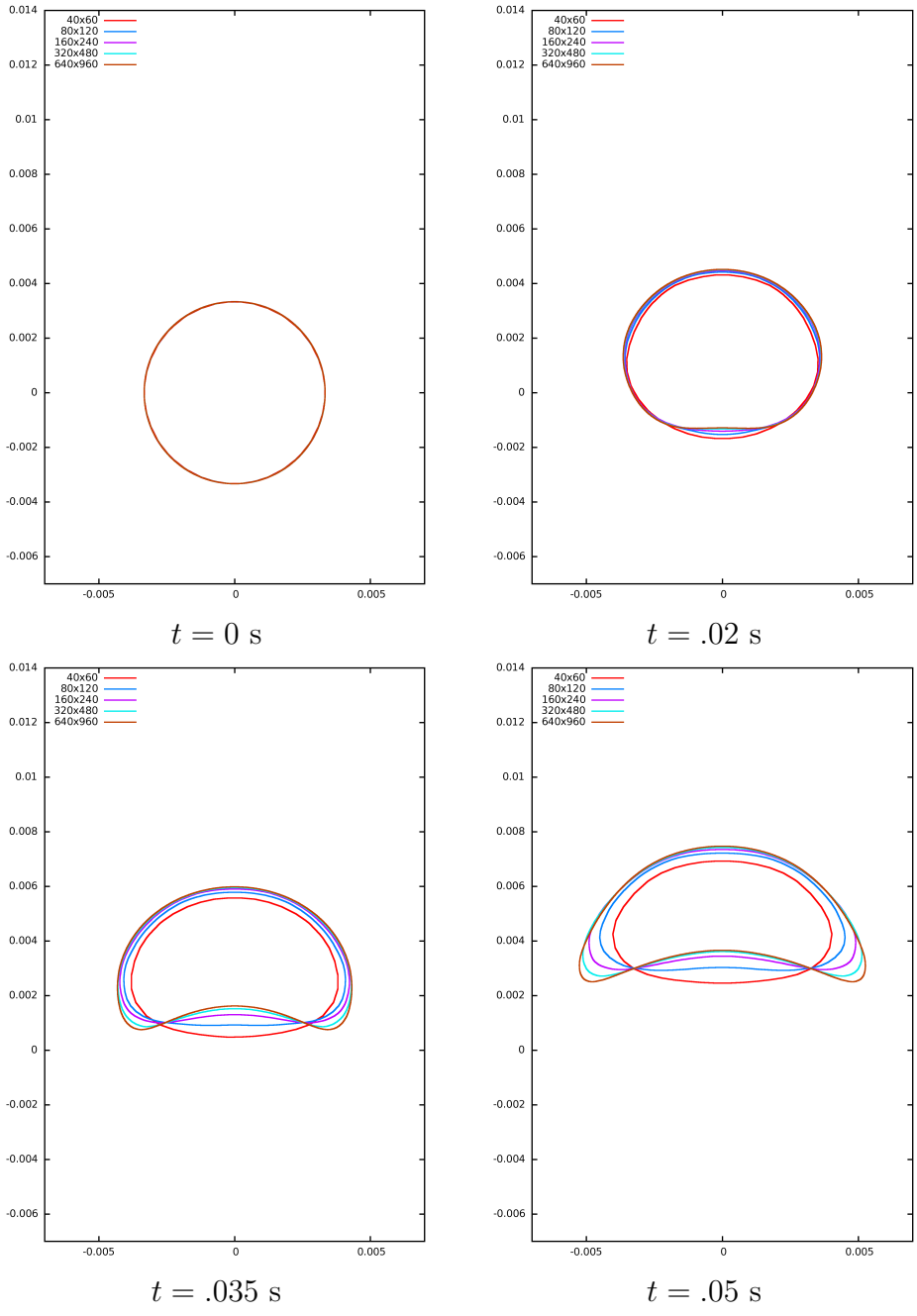


Figure 5.6: Level set profiles for an air bubble of initial radius 1/300 m rising under the effects of surface tension and viscosity inside a computational domain of $[-.01 \text{ m}, .01 \text{ m}] \times [-.01 \text{ m}, .02 \text{ m}]$ filled with water with solid wall boundary conditions.

Chapter 6

Momentum conservation

Consider an isolated incompressible droplet with no ambient pressure forces. When solving equation (2.4) to make the velocities divergence free, a constant pressure along the boundary leads to a net force of zero implying momentum conservation for the droplet. When updating the velocity degrees of freedom on regular MAC grid faces via equation (2.5), the interior cell-centered pressures are applied in a conservative fashion to velocities at faces bordering interior cells, and as there is no net pressure force along the boundary the projection step conserves momentum in each Cartesian direction independently. That is, momentum is conserved during the projection step for the velocity degrees of freedom that surround the cell-centered pressure degrees of freedom.

During the velocity extrapolation step, if one computes ϕ-values at faces by averaging cell-centered ϕ-values, then some of the faces involved in the momentum conserving projection step above may be deemed outside the droplet and overwritten. This violates momentum conservation and can be seen as moving the boundary inward by one grid cell replacing the proper exterior constant pressure Dirichlet boundary condition with a spurious internal pressure. Thus we do not use face-averaged ϕ-values, instead labeling every face adjacent to an interior cell center as interior to the droplet. Similarly, the viscous solver uses the same velocity degrees of freedom

with exterior Neumann boundary conditions. Currently, we do not use momentum conserving advection [41] as the largest errors for momentum conservation occur when applying surface tension.

In the presence of surface tension, there is additional momentum added for any velocity degree of freedom that lies between an interior and an exterior cell. We compute curvature at a face as a ϕ-weighted average of its cell-centered values, i.e.,

$$\kappa_f = \frac{\kappa_i|\phi_{i+1}| + \kappa_{i+1}|\phi_i|}{|\phi_i| + |\phi_{i+1}|} \tag{6.1}$$

The surface tension force per unit area at this face is $\mathsf{sign}(\pm 1)\sigma\kappa$, where $\mathsf{sign}(\pm 1)$ is chosen consistent with the outward unit normal in the Cartesian grid directions based on which cells are interior and exterior to the level set. To properly conserve momentum, the net force due to surface tension should sum to zero for each independent bubble and droplet independently along each of the Cartesian directions. To enforce this for each independent bubble and droplet, we compute the total surface tension force per unit area along each Cartesian direction as $\sigma\kappa^{\mathsf{total}} = \sum_f \mathsf{sign}(\pm 1)\sigma\kappa_f$, and subtract it off from the corresponding force for each face in a curvature-weighted fashion. That is, the new jump for each face becomes

$$\sigma\kappa_f^{\mathsf{new}} = \mathsf{sign}(\pm 1)\sigma\kappa_f - \frac{|\kappa_f|}{\sum_f |\kappa_f|}\sigma\kappa^{\mathsf{total}} \tag{6.2}$$

Obviously alternatives to equation (6.2) exist, and it is not the form of the correction but the fact that a correction needs to be made in order to conserve momentum that we stress. In our simulations we noticed that the net surface tension force along the boundary is close to zero for large well-resolved bubbles and water droplets. However, for under-resolved droplets which are a few grid cells wide, we noticed that the net surface tension force can be so far from zero that droplets can even change directions in mid-air violating conservation of momentum.

Although we only apply the surface tension correction to closed surfaces, it appears that one can use a similar strategy for open surfaces connected to boundaries as well.

Consider for example a single circular bubble, if this bubble is cut in half then it follows from the conservation of momentum that the force on the top half of the bubble must equal the force on the bottom half. However, if we redrew the bottom half of the bubble to any arbitrary curve, then the net force on this new bottom half must still be equal to the net force on the top half. Essentially, the net force on the bottom half of the bubble is independent of its shape and is somehow tied to the boundary conditions on the top half. Therefore, for an open surface connected to the boundary of the domain, with additional knowledge about contact angles, one could likely compute a consistent measure for the net force on the open surface.

Chapter 7

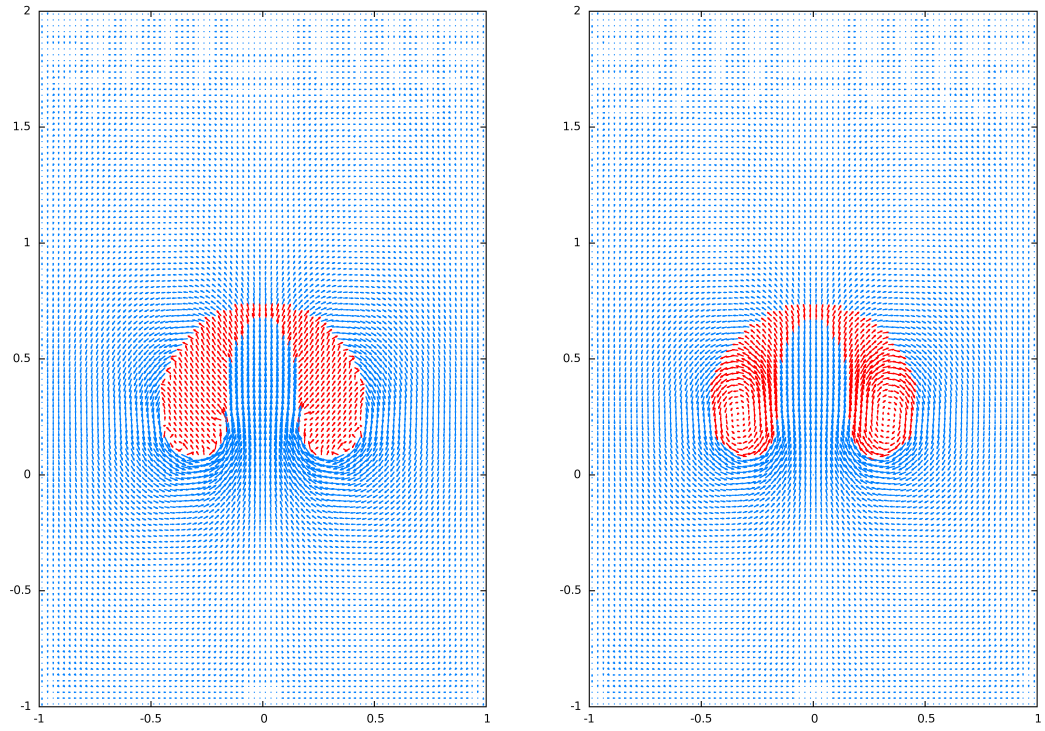
Complex bubble breakup

To summarize, our method for simulating bubbles in free surface incompressible flows is as follows. We use the level set method [50] to track the interface between the bubbles and the incompressible fluid. Initially, each bubble is assigned a density (or mass) which is advected using the unconditionally stable conservative semi-Lagrangian scheme of [41]. We advect the bubble density using air velocities which are constructed and maintained in a separate velocity field, although one could also use velocities extrapolated from the incompressible flow for increased efficiency but lower accuracy. The velocity used for level set advection is a hybrid between the incompressible flow velocity and the air velocity field. Even so, numerical smearing and other errors cause the location of the zero level set and the location of the non-zero bubble densities to drift apart over time. We address this issue as follows. First we compute the total mass that belongs to a bubble as the sum of all the mass inside the bubble and all the mass closest to that bubble. Then we use a flood fill algorithm on that bubble to identify all grid cells belonging to that connected component. The volume of this connected component is carefully computed using a piecewise linear reconstruction of the level set as outlined in [40]. The mass is then uniformly redistributed inside the bubble to obtain a spatially constant bubble density. Before advecting the incompressible velocities and the air velocities, they are extrapolated

across the interface in order to define ghost node values using the fast extension method of [1]. The two velocity fields are then independently advected using the second order accurate MacCormack method [53]. The level set function is advected using the particle level set method of [11] and the semi-Lagrangian advection scheme of [12]. To keep the level set a signed distance function we use the modified fast marching method of [43]. Note that we also compute the advection terms in a thin band of ghost cells so they are adequately defined when the interface moves. Also note that we passively advect the bubble mass that is not close to any bubble using the incompressible flow velocities in order to accurately track sub-grid level details. When a bubble is about to merge with the ambient air near the free surface, dynamic events can cause the bubble to open in one time step and close in the next. To robustly track the bubble density in these cases, we keep advecting the bubble mass even if the bubble merges with the ambient air near the free surface.

Viscosity in the water is treated implicitly using Neumann boundary conditions at the interface that the derivative of each component of the incompressible velocity is zero. As described in Section 2.1, this is based on the assumption that the dynamics inside the air bubbles contain little momentum and hence, they cannot absorb any viscous momentum from the liquid. Thus, the jump in pressure due to viscosity is also zero since the normal component of the viscous stress vanishes across the interface. Finally, as we assume constant viscosity in the incompressible fluid, the equations for the different components of the incompressible velocity decouple as in equations (2.6) and (2.7) for two spatial dimensions. Note that the level set is advected to its new time t^{n+1} position before the viscous update due to the need to prescribe interface boundary conditions.

For the implicit step of our method, we use a coupled solve between a single degree of freedom pressure for the bubble and the surrounding incompressible flow by solving equation (5.2), as described in Section 5.2. The ambient air is taken to be a Dirichlet boundary condition of $p_{\mathsf{atm}} = 101,325$ Pa for which we use the second order cut-cell method of [19]. In the presence of surface tension, the appropriate $\sigma\kappa$ jumps are added to pressure values near the interface. These pressure jumps are carefully



(a) (b)

Figure 7.1: Velocity field at $t = .5$ s on a 80×120 grid for the rising bubble example from [33] (and Figure 5.5) when the air velocities are (a) wiped out at the end of every time step and computed from the boundary incompressible velocities using a second projection solve, and (b) advected forward in time and updated using pressure gradients from a second projection solve. The incompressible velocities are shown in blue and the air velocities are shown in red. Note that the velocity field in (b) appears much more continuous and smooth compared to the velocity field in (a).

computed noting that the net surface tension force on each bubble and each water droplet must be zero, as described in Chapter 6. We use this pressure to update the incompressible velocities via equation (2.5). This provides a very stable monolithic coupling for interactions between bubbles and the surrounding incompressible flow. The air velocities inside the bubbles are computed from the boundary incompressible flow velocities using a second projection as in equations (2.15) and (2.16). One might consider wiping out the air velocities at the end of every time step assuming that the bubble has little momentum and thus, a very small influence on the incompressible

velocity field. However, this destroys the temporal continuity of the overall flow field as shown in Figures 7.1(a) and (b) for the rising bubble example from Figure 5.5. Note that all the vorticity in the air velocities is confined near the boundaries in Figure 7.1(a) which would give rise to a noisy level set as the simulation progresses. In contrast, Figure 7.1(b) has a much better behaved velocity field which maintains a smooth level set over time.

Finally, the size of the time step is computed using equation (2.10) where we add an estimate for $|\nabla p|$ using equation (2.14) which accounts for the change in velocity over a time step. This term prevents the time step from becoming excessively large when velocities are near zero and is similar in spirit to the idea proposed in [33] for body forces and [38] for compressible flow.

7.1 Numerical results

Consider a computational domain of $[-1 \text{ m}, 1 \text{ m}] \times [-1 \text{ m}, 2 \text{ m}]$ which is initially filled with water with density 1000 kg/m^3 where the free surface is located at $y = 1.5$ m. A circular air bubble of radius 1/3 m is centered at the origin with density 1.364 kg/m^3. The effects of surface tension and viscosity are absent. Figure 7.2 shows the level set at $t = 0$, $t = .45$, $t = .9$, $t = 1.2$, $t = 1.5$, $t = 1.8$, $t = 2.4$, $t = 2.7$ and $t = 3$ seconds for two grids of resolutions 320×480 and 640×960 respectively. Note the small scale details that our solver is able to resolve and accurately track over time. In order to show convergence of our solver under grid refinement, we reduced the computational domain to $[-.01 \text{ m}, .01 \text{ m}] \times [-.01 \text{ m}, .02 \text{ m}]$ and the radius of the bubble to 1/300 m, and included the effects of surface tension and viscosity with coefficients $\sigma = .0728$ kg/s^2 and $\mu = .001137$ kg/ms. The initial density of the air bubble is 1.227 kg/m^3. Figure 7.3 shows the level set at $t = 0$, $t = .02$, $t = .04$ and $t = .08$ seconds.

As further validation, we also simulated the rising bubble experiment corresponding to Fig 1A from [25] where a bubble of radius .0061 m and initial density 1 kg/m^3 rises in a liquid of density 875.5 kg/m^3. The computational domain is $[-.1464$ m,

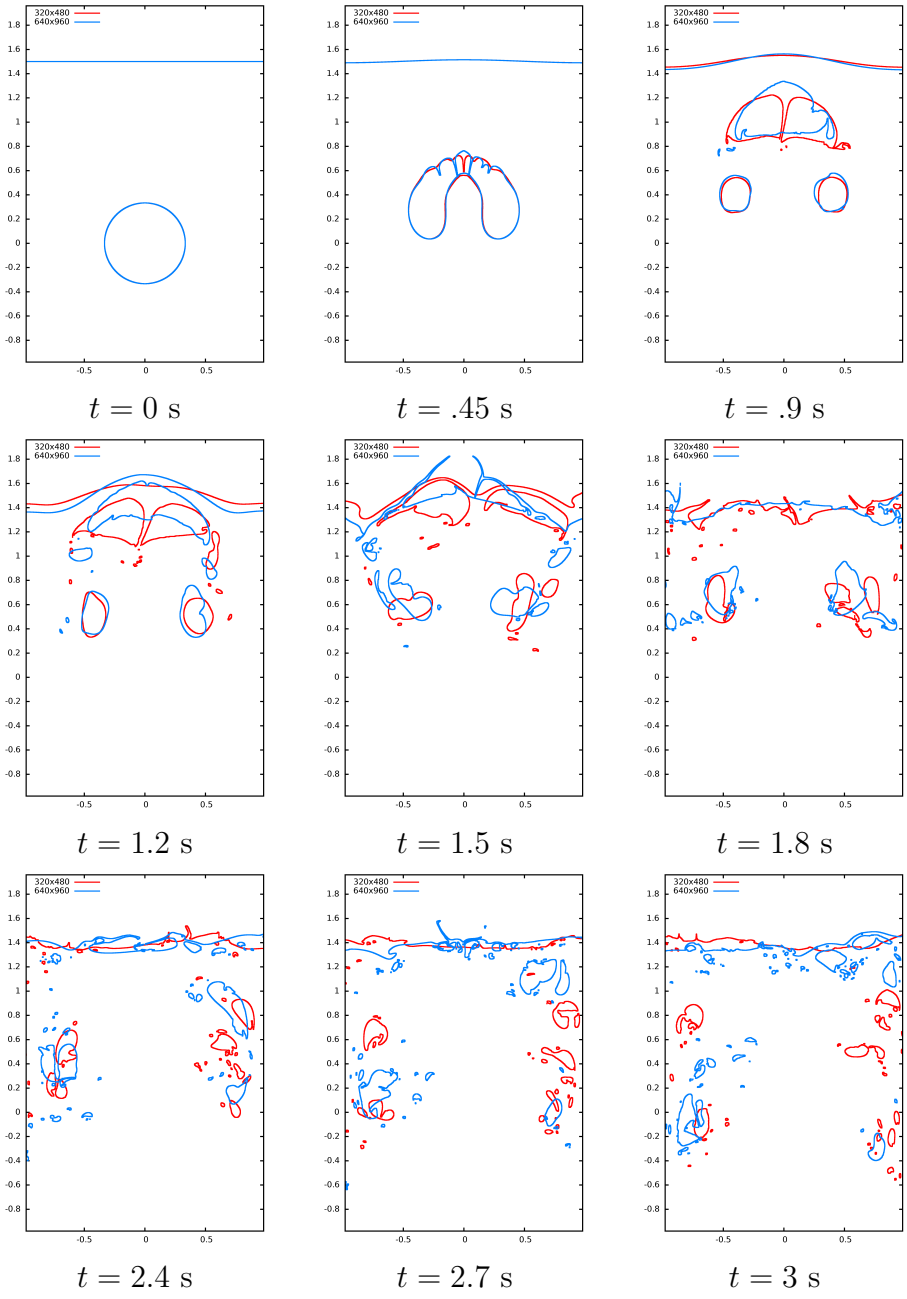


Figure 7.2: An air bubble of initial radius 1/3 m rising inside a computational domain of $[-1\text{ m}, 1\text{ m}] \times [-1\text{ m}, 2\text{ m}]$ filled with water with a free surface initially located at $y = 1.5$ m. The effects of surface tension and viscosity are absent. Note the small scale details that our solver is able to resolve and accurately track over time.

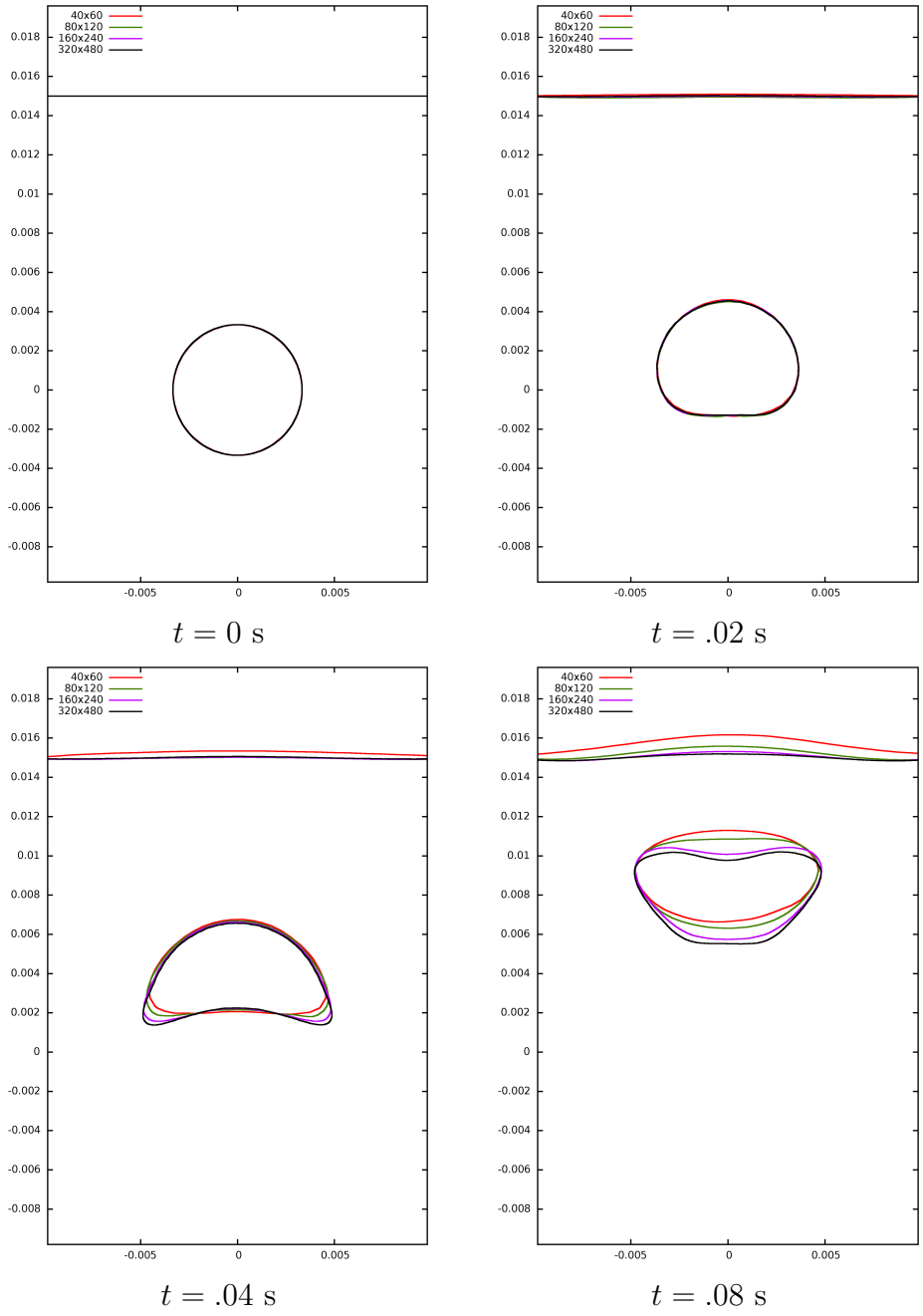
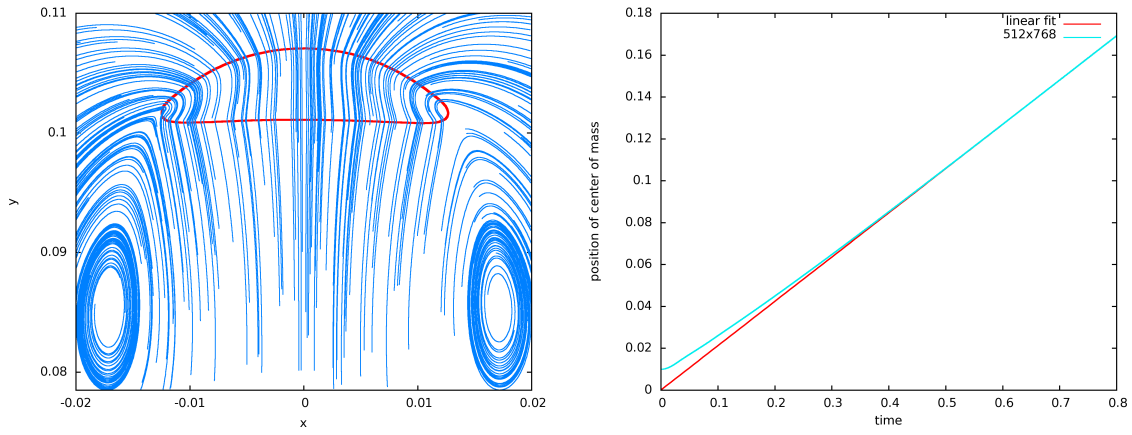


Figure 7.3: Level set profiles under grid refinement for an air bubble of initial radius 1/300 m rising inside a computational domain of $[-.01 \text{ m}, .01 \text{ m}] \times [-.01 \text{ m}, .02 \text{ m}]$ filled with water with a free surface initially located at $y = .015$ m. The effects of surface tension and viscosity are present.



(a) (b)

Figure 7.4: Rising bubble example from Fig. 1A of [25], (a) computed steady state bubble shape and the streamlines inside the bubble and those in its wake, and (b) time evolution of the position of the center of mass of the bubble.

.1464 m] $\times$ [$-$.0732 m, .366 m] and the bubble is initially located at the origin. The coefficient of viscosity is $\mu = .118$ kg/ms and the coefficient of surface tension is $\sigma = .0322$ kg/s^2. The edges of the computational domain have slip solid wall boundary conditions. This example was also simulated in [57, 24] and our solver gives similar results. Figure 7.4(a) shows the bubble at time $t = .6$ s on a 512×768 grid along with the streamlines both inside the bubble and in its wake. Figure 7.4(b) shows the time evolution of the position of the center of mass of the bubble. We compared this data with the linear best fit for $.6 \leq t \leq .8$ seconds. The slope of the linear best fit is .2114 m/s whereas the expected slope is .215 m/s. We note that the streamlines do not match the experimentally observed values due to the fact that a highly simplified single pressure degree of freedom model is used for the air, however, the steady state bubble rise speed is close to the experimentally measured value in [25] and the steady state bubble shape is similar to that observed in [25] and computed in [57, 24].

7.1.1 Object interaction

When updating the incompressible flow velocities during advection, in the presence of objects, we set velocity Dirichlet boundary conditions at cell faces whose centers lie inside an object as these cell faces have well-defined velocities determined by the object velocity. We then advect every face obtaining a well-defined incompressible flow velocity field at time $t^\star$. As we consider interaction with objects that are fully submerged initially, we initialize ϕ inside these objects to be the value when no objects were present, i.e., treat them as water. Subsequently when advecting ϕ, we also update ϕ at grid cells whose centers lie inside an object. Air velocities are handled similar to the incompressible flow velocities during advection with Dirichlet boundary conditions at cell faces inside objects. As described in [41] (see also [23]), for advecting the air density, we modify the forward and backward ray casting to stop when it hits an object. Interpolation weights are computed at the surface point, where weights coming from cells inside an object are discarded and the remaining weights are rescaled to sum to 1. During the viscous solve, we set Dirichlet boundary conditions at cell centers that lie inside objects. When solving for the pressure to make the incompressible flow velocities divergence free, we set Neumann boundary conditions at cell faces whose centers lie inside objects. Similarly, during the second projection step for updating the air velocities, we set Neumann boundary conditions at cell faces whose centers lie inside objects.

Consider a computational domain of $[-1 \text{ m}, 1 \text{ m}] \times [-1 \text{ m}, 2 \text{ m}]$ which is initially filled with water with density 1000 kg/m^3 where the free surface is located at $y = 1.5$ m. A circular air bubble of radius $1/3$ m is centered at the origin with density 1.364 kg/m^3. The domain has nine circular objects with four objects located at $(-.6 + .4k, .5)$, where $k = \{0, 1, 2, 3\}$ and the other five objects located at $(-.8 + .4k, .9)$, where $k = \{0, 1, 2, 3, 4\}$. Figure 7.5 shows the level set at $t = 0$, $t = .25$, $t = .5$, $t = .75$, $t = 1$, $t = 1.25$, $t = 1.5$, $t = 1.75$ and $t = 2.1$ seconds for two grids of resolutions 320×480 and 640×960 respectively. As can be seen, the objects break up the bubble into a large number of small bubbles which our solver is able to resolve and efficiently track over time. We also show an example where the larger bubbles deform

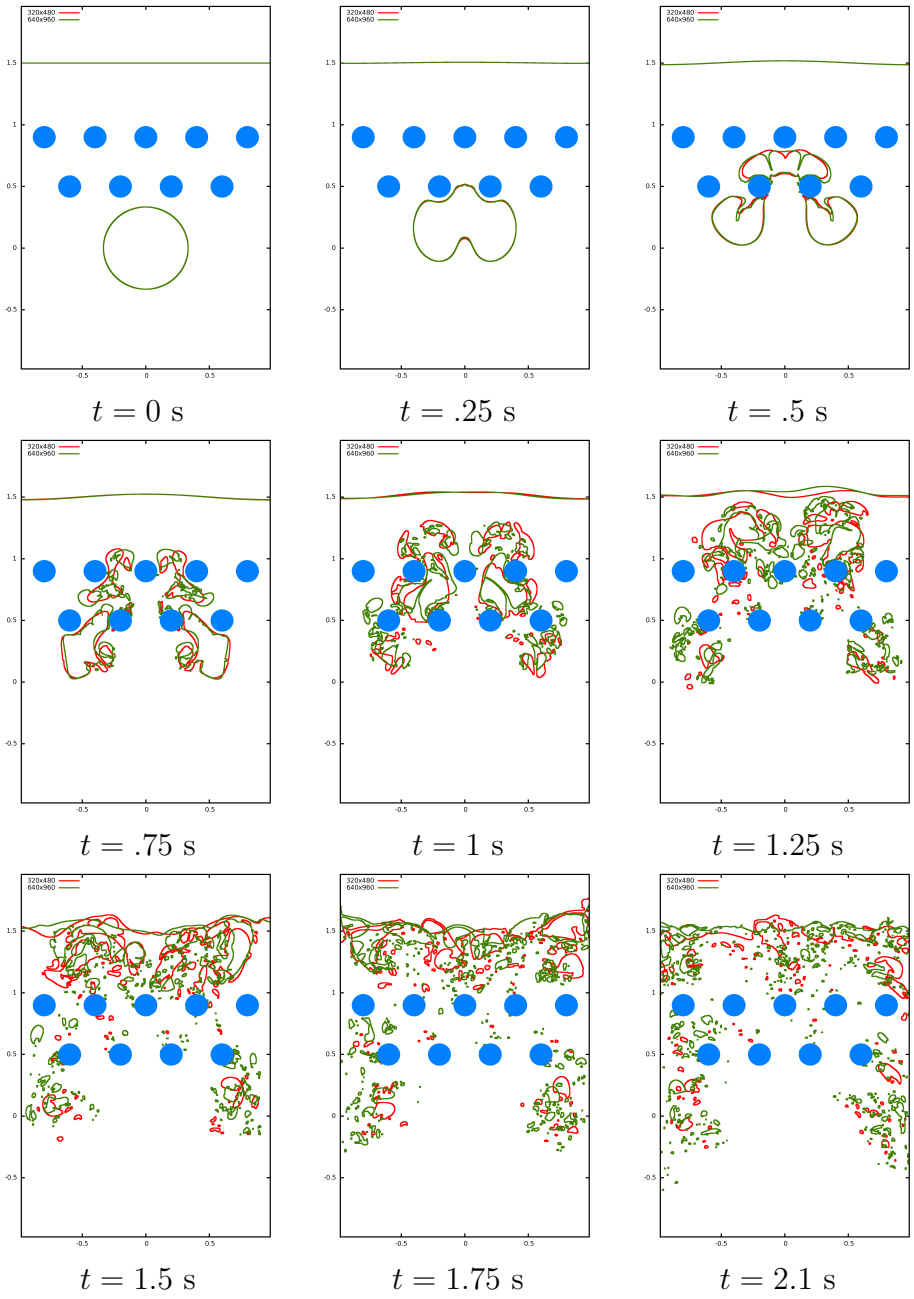


Figure 7.5: An inviscid air bubble of initial radius 1/3 m rising in the presence of objects inside a computational domain of $[-1\text{ m}, 1\text{ m}] \times [-1\text{ m}, 2\text{ m}]$ filled with water with a free surface initially located at $y = 1.5$ m. The objects break up the bubble into a large number of small bubbles which our solver is able to resolve and efficiently track over time.

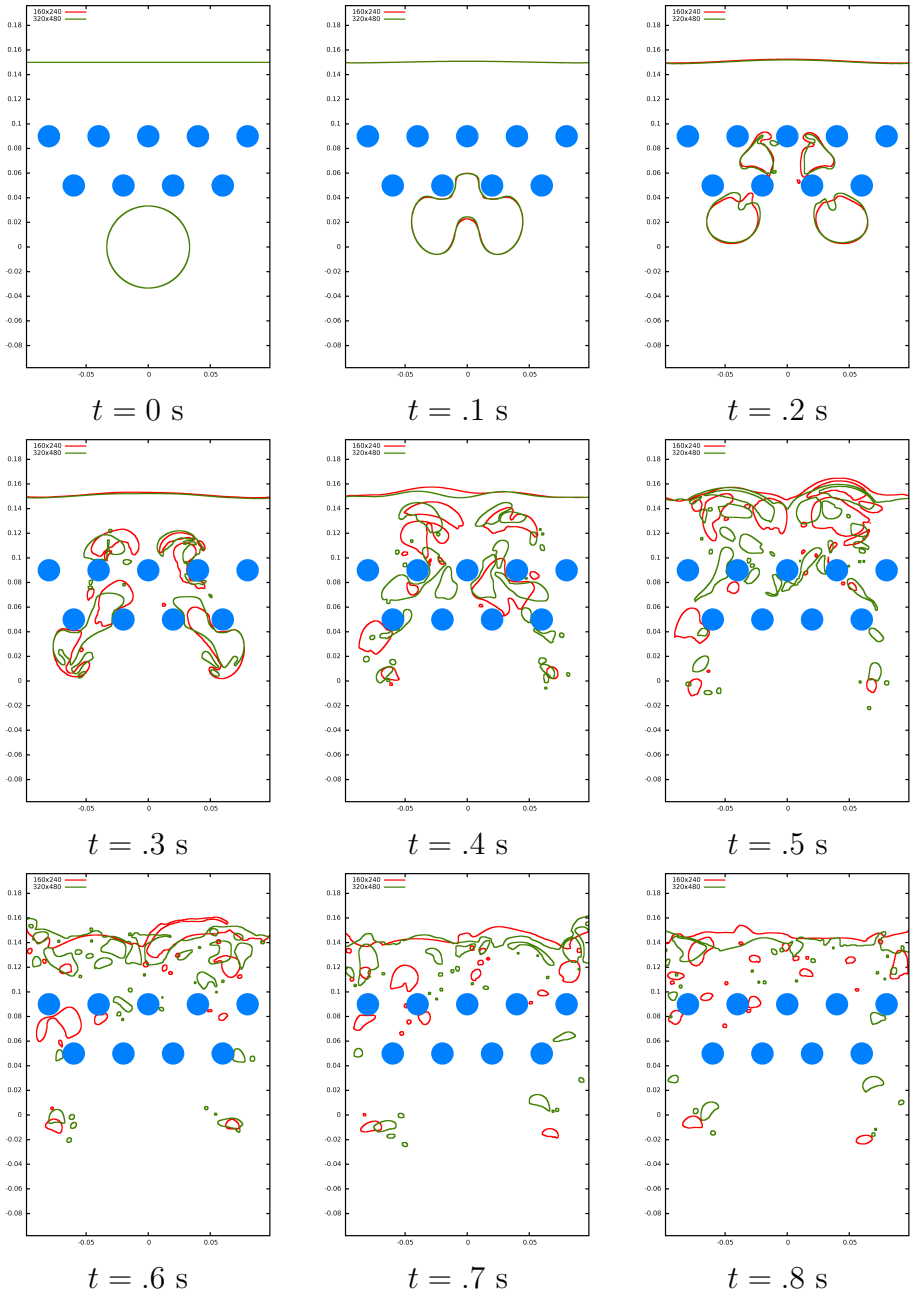


Figure 7.6: An air bubble of initial radius 1/30 m rising in the presence of objects inside a computational domain of $[-.1 \text{ m}, .1 \text{ m}] \times [-.1 \text{ m}, .2 \text{ m}]$ filled with water with a free surface initially located at $y = .15$ m. The effects of surface tension and viscosity are present. Note that the smaller bubbles remain spherical because of larger surface tension forces while the larger bubbles readily deform.

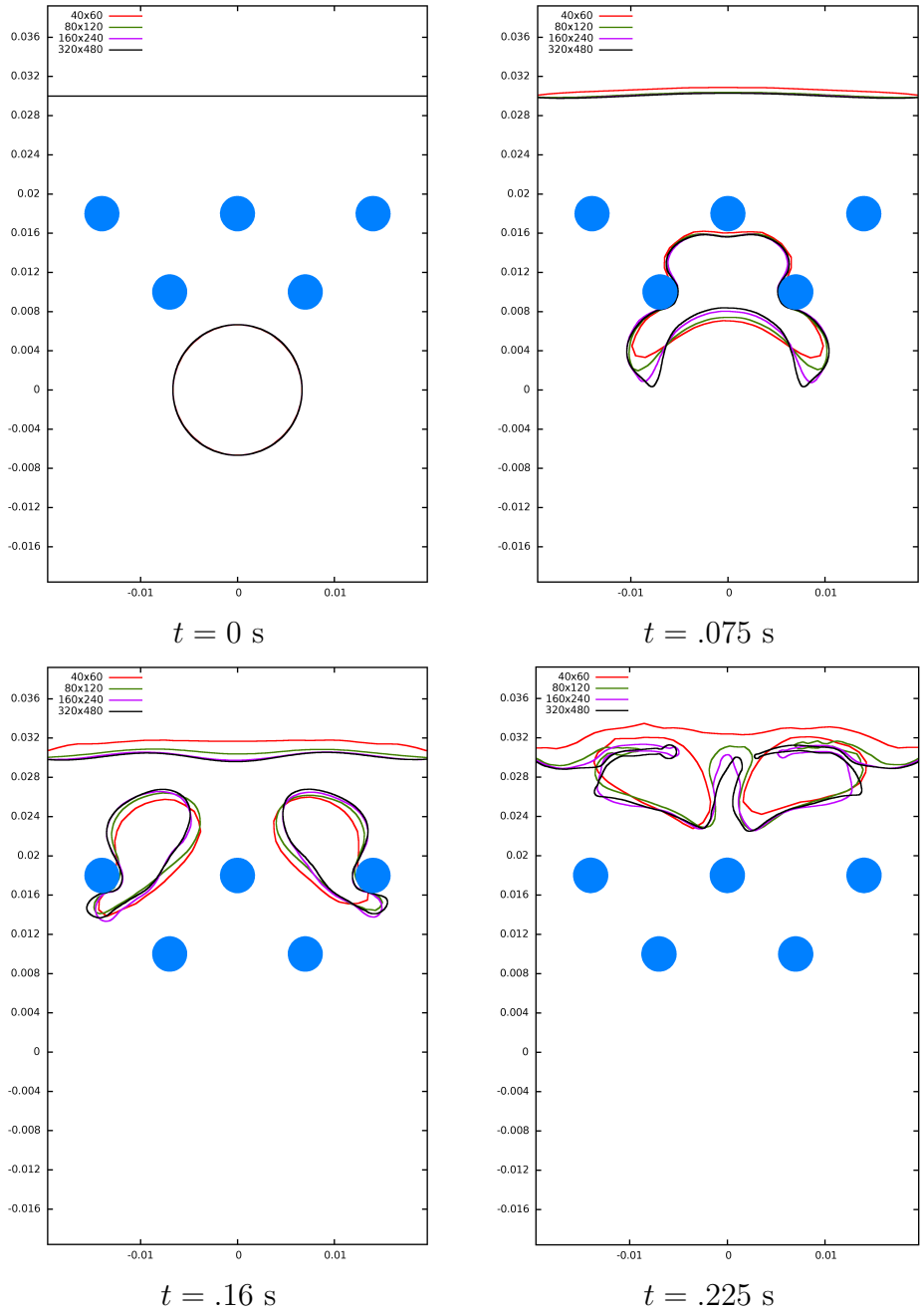
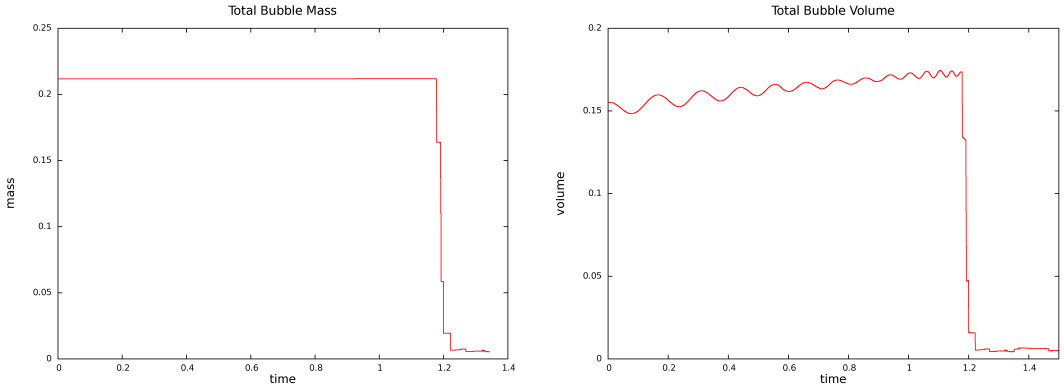


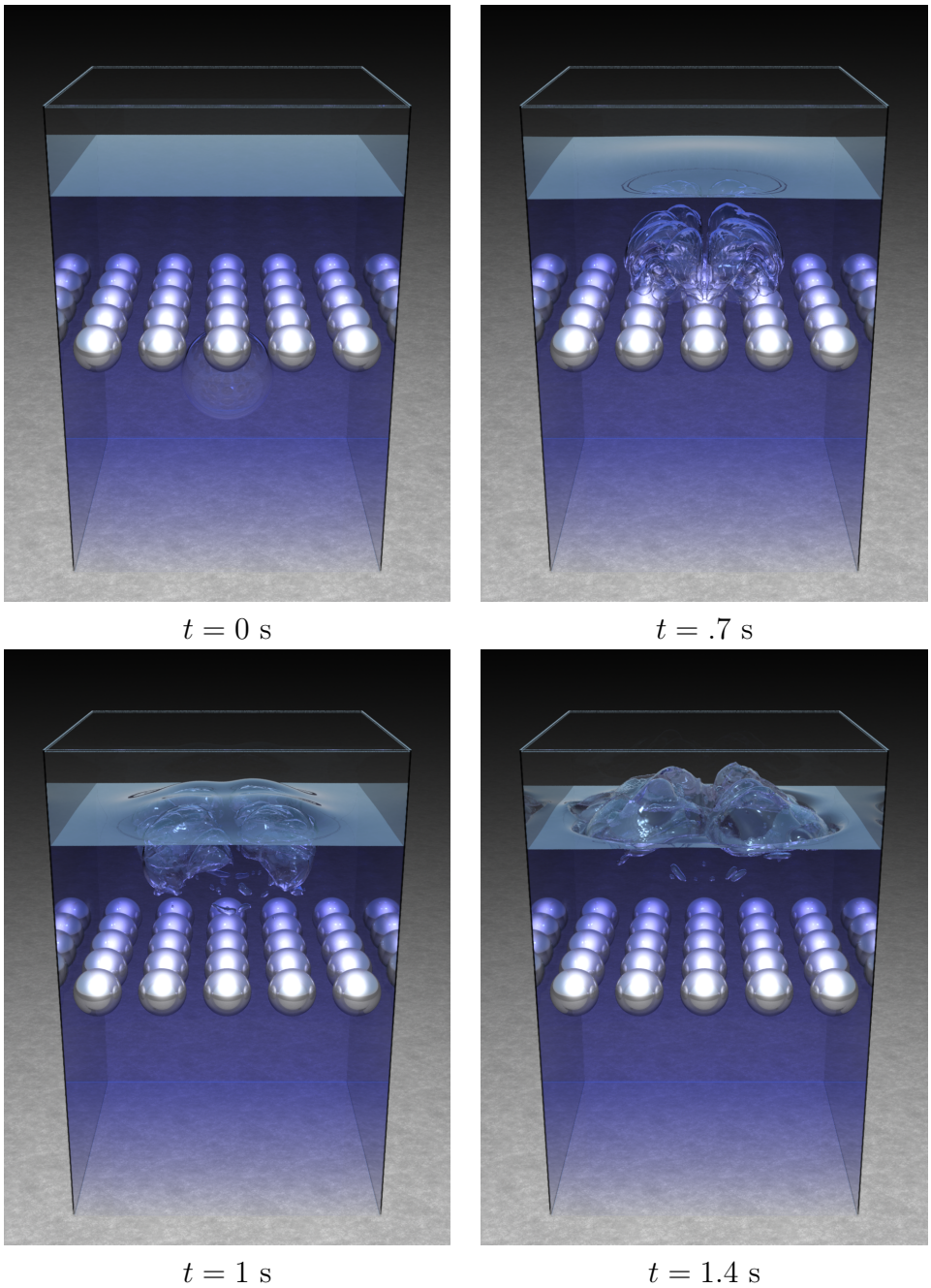
Figure 7.7: Level set profiles under grid refinement for an air bubble of initial radius 1/150 m rising in the presence of objects inside a computational domain of $[-.02$ m, .02 m$] \times [-.02$ m, .04 m] filled with water with a free surface initially located at $y = .03$ m. The effects of surface tension and viscosity are present.

while the smaller bubbles are able to preserve their shape due to surface tension forces by reducing the computational domain to $[-.1\text{ m}, .1\text{ m}] \times [-.1\text{ m}, .2\text{ m}]$, the radius of the bubble to 1/30 m with initial density 1.24 kg/m^3, and adding the effects of surface tension and viscosity with coefficients $\sigma = .0728$ kg/s^2 and $\mu = .001137$ kg/ms. Figure 7.6 shows the level set at $t = 0$, $t = .1$, $t = .2$, $t = .3$, $t = .4$, $t = .5$, $t = .6$, $t = .7$ and $t = .8$ seconds for two grids of resolutions 160×240 and 320×480 respectively. Note that the smaller bubbles remain spherical because of larger surface tension forces while the larger bubbles readily deform. In order to show convergence under grid refinement in the presence of objects, we reduced the computational domain to $[-.02\text{ m}, .02\text{ m}] \times [-.02\text{ m}, .04\text{ m}]$ and the radius of the bubble to 1/150 m, and reduced the number of objects to five. Two objects are located at $(-.007 + .014k, .01)$, where $k = \{0, 1\}$ and the other three are located at $(-.014 + .014k, .018)$, where $k = \{0, 1, 2\}$. The effects of surface tension and viscosity are present with the same coefficient values as above. The initial density of the air bubble is 1.229 kg/m^3. Figure 7.7 shows the level set at $t = 0$, $t = .075$, $t = .16$ and $t = .225$ seconds.



(a) (b)

Figure 7.8: Numerical profiles for the (a) total mass, and (b) total volume of the bubbles for the rising bubble example in three spatial dimensions.



$t = 0$ s $t = .7$ s

$t = 1$ s $t = 1.4$ s

Figure 7.9: An air bubble rising in a water column with a free surface in the presence of objects in three spatial dimensions. The effects of surface tension and viscosity are absent.

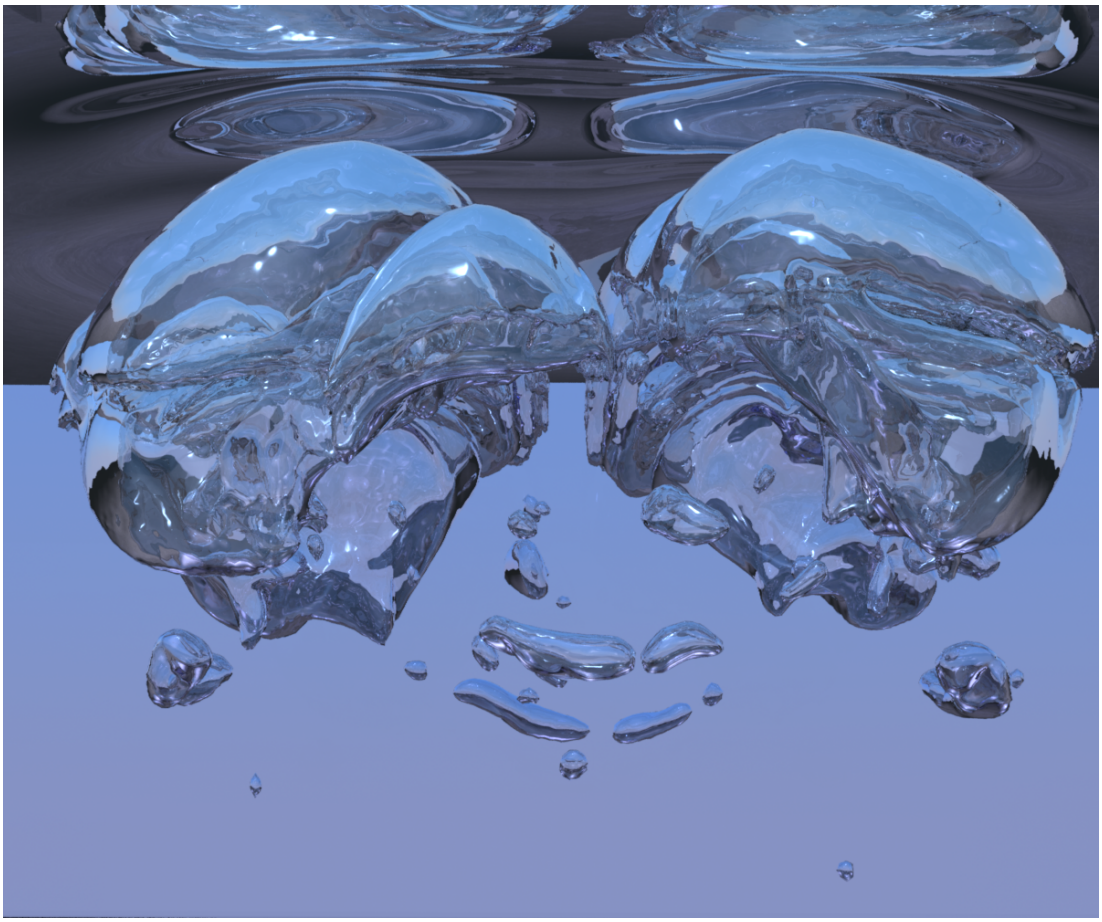


Figure 7.10: A close up of the three dimensional rising bubbles at $t = 1$ seconds (see Figure 7.9(c)). Note the large amount of topological detail that our solver is able to resolve and accurately track over time.

7.1.2 Three spatial dimensions

Consider a computational domain of $[-1 \text{ m}, 1 \text{ m}] \times [-1 \text{ m}, 2 \text{ m}] \times [-1 \text{ m}, 1 \text{ m}]$ which is initially filled with water with density 1000 kg/m^3 where the free surface is located at $y = 1.5$ m. An air bubble of radius 1/3 m and density 1.364 kg/m^3 is centered at the origin. The effects of surface tension and viscosity are absent. To break up the bubble into a large number of smaller bubbles, the domain also has 25 spherical objects of radius .15 m centered at $(-.8 + .4i, .5, -.8 + .4j)$, where $i, j = \{0, 1, 2, 3, 4\}$. Figure 7.9 shows the level set at $t = 0$, $t = .7$, $t = 1$ and $t = 1.4$ seconds simulated

using a grid of resolution $256 \times 384 \times 256$. A close up of the level set at $t = 1$ seconds is shown in Figure 7.10 illustrating the large amount of topological detail that our solver is able to resolve and accurately track over time. Figures 7.8(a) and (b) show the numerical profiles for the total bubble mass and volume demonstrating that the proposed method allows bubbles to readily change in volume while conserving the total bubble mass. This simulation took approximately two weeks for simulating 240 frames at 80 frames per second (i.e., a 3 second simulation) on a dual hexcore T7500 Dell workstation.

Chapter 8

Extensions to sub-grid scale

Although the Eulerian scheme proposed in Chapters 5 and 7 works for large well-resolved bubbles, it cannot keep track of sub-grid scale bubbles since the level set loses volume over time. To track these under-resolved bubbles, we propose a Lagrangian formulation which is still strongly coupled to the surrounding flow.

8.1 Sub-grid bubbles

We use the equation of state $P_b = B\rho_b$ for the sub-grid bubbles with the constant B chosen such that a density $\rho_b = 1.226$ kg/m^3 gives a pressure $P_b = 101{,}325$ Pa. We assume that the sub-grid bubbles are spherical in shape with radius r, have a single radial velocity degree of freedom v_r, and a single pressure degree of freedom P_b which is coupled to all the surrounding fluid degrees of freedom in a monolithic fashion.

When solving for the bubble volumes, monolithic approaches are preferable to partitioned approaches because they do not require additional relaxation techniques for stability and robustness (see e.g. [63, 34]). Therefore for stability reasons, we follow an approach similar to that described in Chapter 5 which modelled a well-resolved Eulerian level set bubble occupying N grid cells with the following equation,

$$\frac{N}{\Delta t^2 B \rho_b} P_b - \overline{\left(\frac{\nabla p}{\rho}\right)} \frac{\mathcal{P}^n}{V_c} = \frac{N}{\Delta t^2} - \frac{\bar{u}^n \mathcal{P}^n}{V_c \Delta t} \tag{8.1}$$

where Δt is the size of the time step, V_c is the volume of a grid cell, $\bar{u}^n$ is the average radial velocity of the bubble, $\mathcal{P}^n$ is the surface area of the bubble, and $\overline{\left(\frac{\nabla p}{\rho}\right)}$ is the average density-weighted pressure gradient across the bubble-water interface. We would like to use a similar equation for the sub-grid bubbles as well so that they have the same qualitative behavior as the level set bubbles and seamlessly convert into them when they grow large enough. A brute force approach for achieving this by creating a mesh for each sub-grid bubble would result in increased complexity and poor conditioning due to small control volumes. Instead, we make some approximations noting that our resulting scheme gives adequate results as illustrated in Figure 8.1.

First, we substitute $\bar{u}^n = v_r^n$ and $N = V_b^n / V_c$, where V_b^n is the volume of the bubble, and rewrite equation (8.1) as,

$$\frac{V_b^n}{V_c \Delta t^2 B \rho_b} P_b - \overline{\left(\frac{\nabla p}{\rho}\right)} \frac{\mathcal{P}^n}{V_c} = \frac{V_b^n}{V_c \Delta t^2} - \frac{v_r^n \mathcal{P}^n}{V_c \Delta t} \tag{8.2}$$

Notice as $\Delta t \to 0$, the first term on each side of the equation must balance indicating that the bubble pressure equals the equation of state pressure. Moreover, when the bubble pressure is identical to the equation of state pressure these terms cancel, and in order to remain at equilibrium with $v_r^n = 0$ the term $\overline{\left(\frac{\nabla p}{\rho}\right)} \frac{\mathcal{P}^n}{V_c}$ must also vanish. This means that the bubble pressure tries to match the average external pressure from the fluid when it is near radial equilibrium (n.b. equation (8.4)). Note that $\overline{\left(\frac{\nabla p}{\rho}\right)}$ is an area-weighted average where the weights are computed based on the fraction of the bubble's surface area visible to a neighboring fluid cell and that cell's pressure degree of freedom p_i. We *estimate* these weights w_i as the weights each of the neighboring eight cells would have in a tri-linear interpolation formula for the location of the center of a bubble. Then

$$\overline{\left(\frac{\nabla p}{\rho}\right)} \approx \sum_{i=1}^{8} w_i \frac{(p_i - P_b)}{\Delta x_i \rho} \approx \sum_{i=1}^{8} w_i \frac{(p_i - P_b)}{\Delta x \rho_b} \tag{8.3}$$

where Δx_i is the distance between the sub-grid bubble center and the center of the ith incompressible cell. We have found that we can make further approximations replacing Δx_i by a characteristic length Δx and replacing ρ by the bubble density ρ_b as seen in the rightmost term in equation (8.3). Here Δx is chosen as the length of a grid cell in the case of our uniform grid. Although these approximations might appear aggressive, they allow us to treat the sub-grid bubbles as point particles while keeping the equations well-defined even for degenerate cases where the sub-grid bubbles overlap each other or encompass a fluid degree of freedom. Note that

$$\sum_{i=1}^{8} w_i \frac{(p_i - P_b)}{\Delta x \rho_b} = \frac{p_{\mathsf{avg}} - P_b}{\Delta x \rho_b} \tag{8.4}$$

where p_{avg} is the incompressible flow pressure linearly interpolated to the bubble's

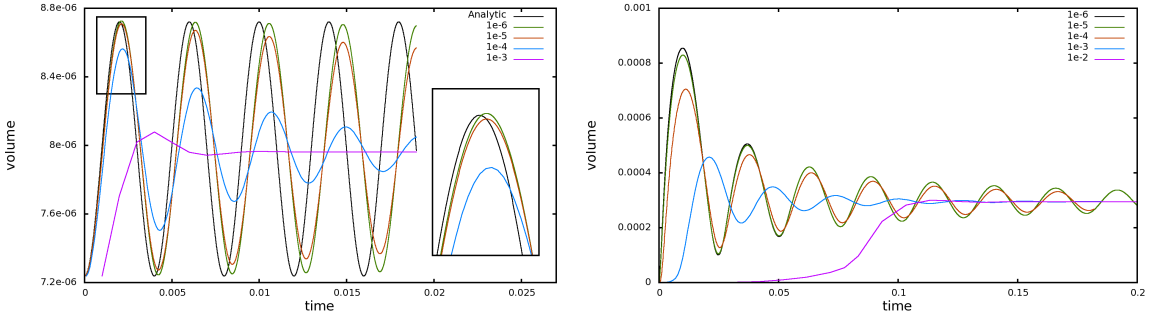


Figure 8.1: Using equations (8.5) and (8.11), we solve the oscillating bubble problem for a sub-grid bubble (radius = $.3\Delta x$) on a 25^3 grid of dimensions 1m^3 with (Left) an initial density of 1.1 kg/m^3, where the bubble starts with an initial volume of 7.238 cm^3, converges to the predicted volume of 7.962 cm^3, and closely approximates the analytic bubble oscillation frequency as the size of the time step is refined, and (Right) an initial density of 1,100 kg/m^3, where the bubble starts with an initial volume of .268 cm^3 and expands three orders of magnitude, remaining stable even when it grows beyond its incompressible neighbors. Note that the bubble remains stable at all time steps in both cases.

center. This provides some intuition as to why these approximations work. The term on the right is a reasonable approximation to the gradient between the incompressible flow pressure and the bubble's pressure using the same characteristic length scale.

In summary, we use the following equation when solving for the pressure of a sub-grid bubble,

$$\frac{V_b^n}{V_c \Delta t^2 B \rho_b} P_b - \sum_{i=1}^{8} w_i \frac{(p_i - P_b)\mathcal{P}^n}{\Delta x \rho_b V_c} = \frac{V_b^n}{V_c \Delta t^2} - \frac{v_r^n \mathcal{P}^n}{V_c \Delta t} \tag{8.5}$$

Even with these approximations, we converge to the analytic bubble volume at sub-grid resolutions and the simulation remains stable with large time steps. The denominator ρ in equation (8.3) controls the bubble's oscillation frequency. For any given radius, it can be tuned such that the model closely matches the analytic bubble oscillation frequency. Most of our sub-grid bubbles are seeded with radii in the interval $(.2\Delta x, .3\Delta x)$, and setting $\rho = \rho_b$ works quite well in this case as shown in Figure 8.1(left). For more accuracy such as when simulating fluid sounds, one could choose a better value for ρ or even make it a function of the bubble's radius. We leave this as future work since all our examples use large time steps and only rely on the sub-grid bubbles converging to the right volume while remaining stable.

Note that we treat all sub-grid bubbles independently of each other when coupling to the external fluid pressures. In addition, we set weights to zero when a neighboring fluid degree of freedom is inside a kinematic object or is subject to a free surface pressure boundary condition. This means that both kinematic objects and free air cannot see the pressure from the sub-grid bubble which is fine. When a neighboring fluid degree of freedom is inside a level set bubble, the level set pressure degree of freedom can be coupled to the sub-grid bubble pressures, except that this increases the density of the Poisson equation matrix repeatedly for every sub-grid bubble adjacent to a single level set region - which we have observed increases the number of iterations required by PCG for convergence. Therefore, we set weights to zero in this case as well.

As shown in [38], the divergence of the velocity field at time t^{n+1} is given by the second and fourth terms in equations (8.1) and (8.2), and thus equation (8.5). Since the volume-weighted divergence of a sub-grid spherical bubble is given by $v_r\mathcal{P}$, we can write

$$v_r^{n+1}\mathcal{P}^{n+1} = V_c D^{n+1} = v_r^n\mathcal{P}^n - \Delta t\mathcal{P}^n \sum_{i=1}^{8} w_i \frac{(p_i - P_b)}{\Delta x \rho_b} \tag{8.6}$$

After solving a monolithically coupled Poisson equation for all the fluid and bubble pressures, the right hand side of equation (8.6) is known. Using the definition of surface area $\mathcal{P}^{n+1} = 4\pi(r^{n+1})^2$ and writing $v_r^{n+1} = dr^{n+1}/dt$, we analytically integrate equation (8.6) from time t^n to t^{n+1} to obtain r^{n+1}. Once r^{n+1} is determined, we use a backward Euler discretization of $r^{n+1} - r^n = \Delta t v_r^{n+1}$ for computing v_r^{n+1}.

8.2 Coupling to Incompressible Flow

Consider an incompressible fluid containing many sub-grid bubbles with the inviscid Navier-Stokes equation given by

$$\vec{u}_t + (\vec{u}\cdot\nabla)\vec{u} + \frac{\nabla p}{\bar{\rho}} = \vec{g} \tag{8.7}$$

where $\bar{\rho}$ is the average density, $\vec{u}$ is the velocity, and $\vec{g}$ is the net body force acting on the fluid. We discretize these equations on a MAC grid where we first explicitly update

$$\frac{\vec{u}^\star - \vec{u}^n}{\Delta t} + (\vec{u}\cdot\nabla)\vec{u} = \vec{g} \tag{8.8}$$

with a semi-Lagrangian MacCormack method [53], and then solve for the pressure via

$$\nabla \cdot \frac{\nabla p}{\bar{\rho}} = \frac{\nabla \cdot \vec{u}^{\star}}{\Delta t} - \frac{\nabla \cdot \vec{u}^{n+1}}{\Delta t} \tag{8.9}$$

in order to update the intermediate velocity $\vec{u}^{\star}$ using equation (2.5).

We lump the divergences of all the sub-grid bubbles into a column vector $\vec{D}^{n+1}$ so that they can affect the divergences of the individual fluid cells via $W\vec{D}^{n+1}$, where W is a weight matrix with W_{jk} corresponding to the fraction of the divergence of bubble k that is added to the divergence of cell j. For simplicity of exposition, consider a single bubble where we only consider one column of the weight matrix which we index solely by the cell, i.e., for example W_j. Then, equation (8.9) can be written as

$$-\nabla \cdot \frac{\nabla p}{\bar{\rho}} = -\frac{\nabla \cdot \vec{u}^{\star}}{\Delta t} + \frac{W\vec{D}^{n+1}}{\Delta t} \tag{8.10}$$

With the aid of equation (8.6), we discretize equation (8.10) for cell j with faces f as follows

$$-\sum_{f=1}^{6} \frac{p_f - p_j}{\bar{\rho}\Delta x^2} + W_j \sum_{i=1}^{8} w_i \frac{(p_i - P_b)\mathcal{P}^n}{\Delta x \rho_b V_c} = -\frac{\nabla \cdot \vec{u}^{\star}}{\Delta t} + W_j \frac{v_r^n \mathcal{P}^n}{V_c \Delta t} \tag{8.11}$$

where p_f refers to the pressure on the other side of the face f. For multiple bubbles, the second and fourth terms in equation (8.11) must be summed over all the influencing bubbles k with W_j replaced by W_{jk}.

The weight matrix W can be chosen such that the resulting system of equations (8.5) and (8.11) is symmetric positive definite allowing for the use of fast solvers such as preconditioned conjugate gradient. In order to obtain symmetry, the coefficient of P_b in the second term in equation (8.11) must be the same as the coefficient of p_j in equation (8.5), and the coefficient of p_i (when $i \neq j$) must be the same as that for p_j in the corresponding equation for cell i. The first condition means

that $w_j = W_j \sum_{i=1}^{8} w_i$ or $W_j = w_j / \sum_{i=1}^{8} w_i$. Note that this relation automatically satisfies the second condition for symmetry as well. Typically, since w_i represents the interpolation weight, $\sum_{i=1}^{8} w_i = 1$ and we are simply using the interpolation weights once again to define W. However, as pointed out in Section 8.1, objects, level set bubbles and the free surface are not directly coupled to the sub-grid bubble's pressure discretization yielding weight values of zero and $\sum_{i=1}^{8} w_i \neq 1$. Technically, this means that our sub-grid bubbles are not directly coupled to objects, level set bubbles, or free surface pressure boundary conditions but are always assumed to be submerged in the neighboring fluid degrees of freedom that happen to be present.

Finally, after solving for the pressure and updating the fluid velocities in the usual manner, the translational velocity of the bubble is set to be the interpolated average fluid velocity $\vec{u}$ at the center of the bubble. One could make the bubble motion more lively by applying additional forces such as buoyancy, vorticity confinement or a random perturbation as done in [35] - we use buoyancy in our examples. For greater accuracy, this new velocity can then be subtracted from the average velocities to conserve the fluid momentum, although this step is not essential since the bubble momenta is very small.

Chapter 9

Bubble-bubble interactions

When two sub-grid bubbles overlap, we merge them into a single bubble adding their masses and volumes. The radial velocity of the bubble is chosen such that the net divergence is equal to the sum of the divergences of the original bubbles. Additionally, for increased realism, similar to [29] we apply an attraction force which is of the form $f_{\mathsf{attract}} = Km_1m_2/r^2$, where K is a constant, m_1, m_2 are the masses of the two bubbles and r is the distance between them. When a sub-grid bubble grows large enough such that its radius covers more than two grid cells, we convert it to a level set function. This is accomplished by rasterizing the sub-grid bubble onto the grid and adding its mass to the level set region by computing the appropriate density. Note that the divergence remains continuous during this process since the background fluid velocity already contains the bubble's divergence. Also, when a sub-grid bubble enters a level set bubble we delete the sub-grid bubble and add its mass to the mass of the level set region by modifying the density field.

If a level set bubble becomes smaller than a grid cell, it can lose mass because of numerical errors during advection. However, the bubble mass cannot disappear because it is advected conservatively using the method of [41]. This stray mass was distributed to the nearby bubbles as described in Chapter 2. However, such a scheme can sometimes move the bubble mass too far away in a non-physical manner. Instead,

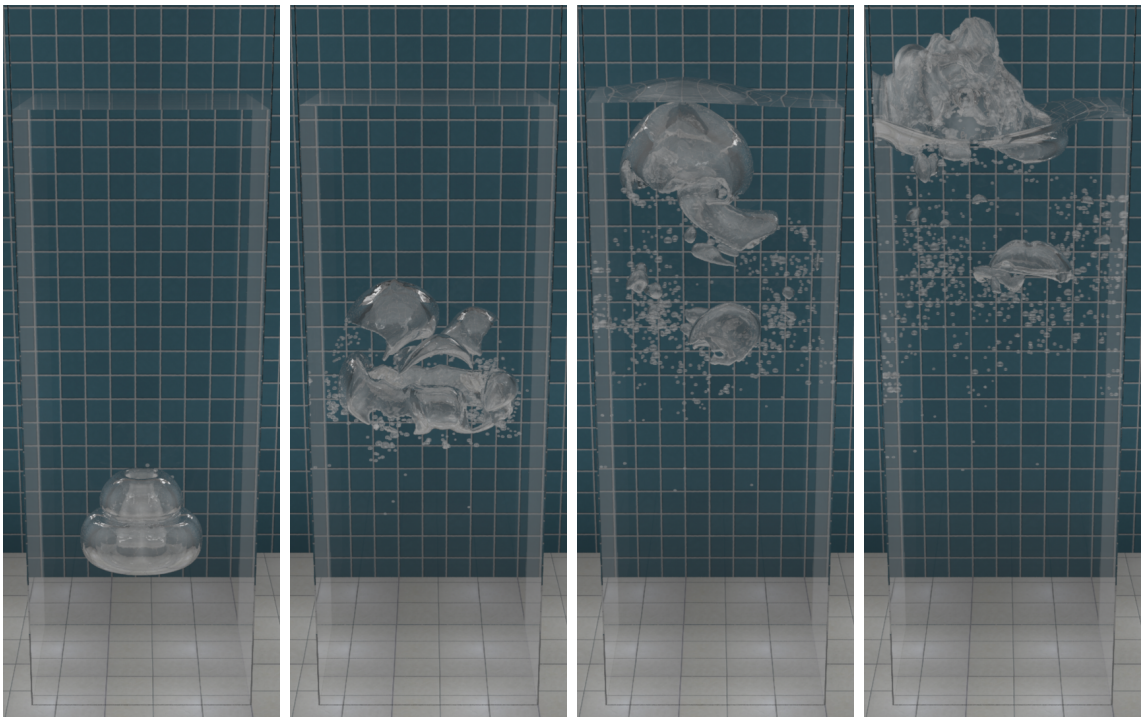


Figure 9.1: A single level set bubble rises in a tall domain, undergoing topological deformations and generating smaller level set as well as sub-grid bubbles during its temporal evolution (150×500×150 grid); about 1,200 sub-grid bubbles.

we propose to track this stray mass using sub-grid bubbles as shown in Figure 9.1. To achieve this, we first run a greedy condensation procedure on the stray density field by moving it in the direction of the gradient vectors for a few iterations. Then for every cell with density above some threshold we seed a sub-grid bubble with the appropriate mass. To correctly choose its volume, we set the steady state pressure $p = \rho_I g h$ (where ρ_I is the density of the incompressible fluid and h is the depth of the sub-grid bubble from the water surface) to be equal to the equation of state pressure $P_b = B\rho_b = BM_b/V_b$ and solve for V_b. Note that we do not use the incompressible pressure for computing the bubble's volume because it can oscillate wildly and even go negative at times during the course of the simulation due to small numerical errors in the velocity field - this is because of the well-known fact that the fluid pressure in incompressible flow is more of a Lagrange multiplier than an actual pressure. Finally note that even if our initial volume estimate has some errors, the monolithic coupling

keeps the scheme stable and the bubble readily changes volume to an appropriate value.

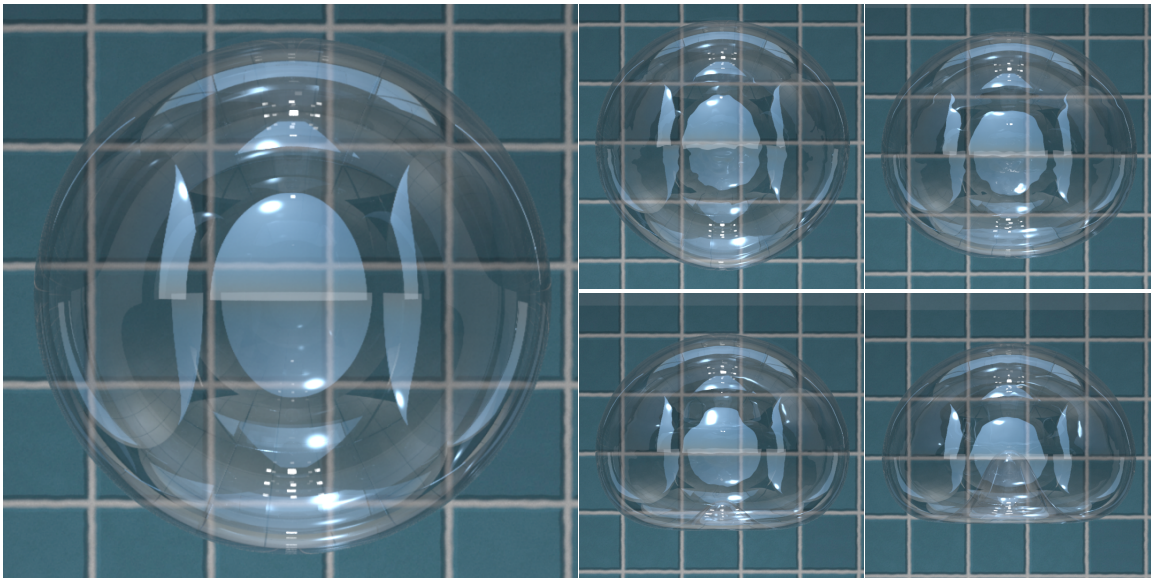


Figure 9.2: To preserve visual realism, we render a sub-grid bubble as a time-evolving level set function by maintaining a dictionary of level sets acquired from a coarse simulation and intersecting rays with them during the rendering process. Shown in the figure is a sub-grid bubble rising on a $6\times18\times6$ grid rendered (Left) as a sphere, and (Right) at different points in time using our level set dictionary.

9.1 Time-evolving proxy geometry

Although sub-grid bubbles are monolithically coupled to the surrounding fluid, rendering them as oscillating spheres next to fully deforming level set bubbles can look visually disturbing. To avoid this, we render them as time-evolving level set functions which have been pre-computed offline. This was achieved by maintaining a dictionary of level sets acquired from a rising bubble simulated on a coarse grid. During the rendering process these level sets are substituted within the bounding boxes of the sub-grid bubbles and intersected with the rays, as shown in Figure 9.2. Bubble shapes can also be handcrafted or created via superposition of certain basis functions [48] for

use during the rendering process. Using this method for rendering a simulation with hundreds of thousands of sub-grid bubbles can be computationally quite expensive. However, we found that using this method on the largest few thousand bubbles added negligible computational overhead while drastically increasing the visual realism, see Figure 9.3(far right). Note that in some cases we have rendered the sub-grid bubbles larger than their actual radii to increase the visual expression.

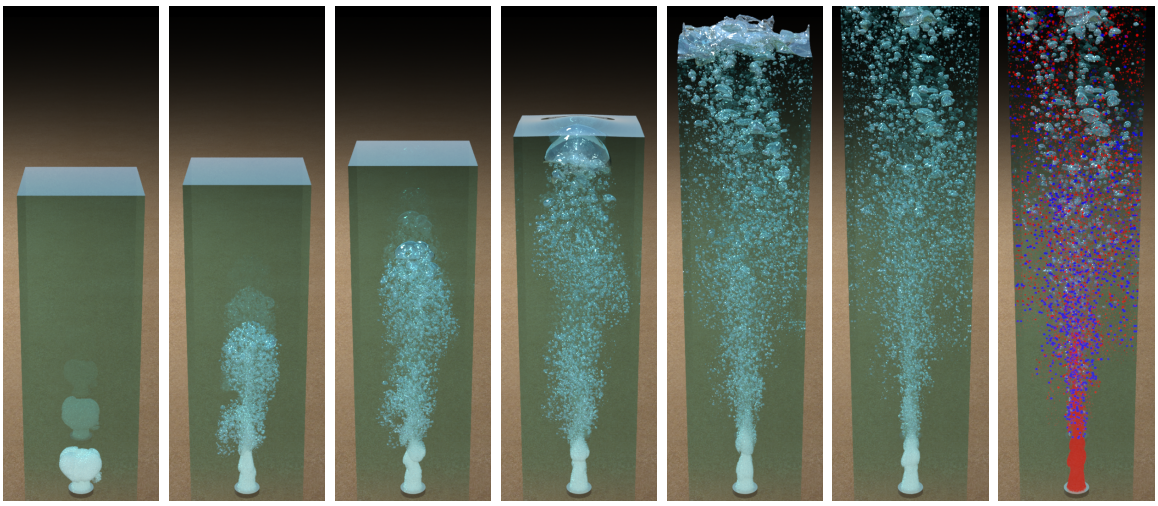


Figure 9.3: A cylindrical source inside a tall domain seeding tiny bubbles which grow as they rise and merge together due to attraction forces, ultimately forming large sub-grid and level set bubbles as they approach the surface (128×640×128 grid); about 500,000 sub-grid bubbles. (Far right) shows the sub-grid bubbles in red and blue, where red depicts the smaller spherical ones (Figure 9.2 left), and blue depicts those rendered using the time evolving level set dictionary (Figure 9.2 right).

9.2 Solid object interaction

As described in Chapter 7, when advecting the fluid velocities, the object velocity is set as a Dirichlet boundary condition at cell faces that lie inside the object. For level set advection, objects are treated as water. i.e., the level set function ϕ is initialized to be the value as if no objects were present and subsequently also updated at grid cells that lie inside the object. For advecting the air mass, the forward and backward

advection rays are clamped when they hit an object, as described in [23, 41]. The surface point is used for computing the interpolation weights, where weights coming from cells inside the object are discarded and the remaining weights are rescaled to sum to 1. Similarly, when advecting the sub-grid bubbles we clamp the advection rays to the object surface. During the pressure projection step, we set Neumann boundary conditions at cell faces that lie inside thick objects, while for thin shells we use visibility information as described in [23].

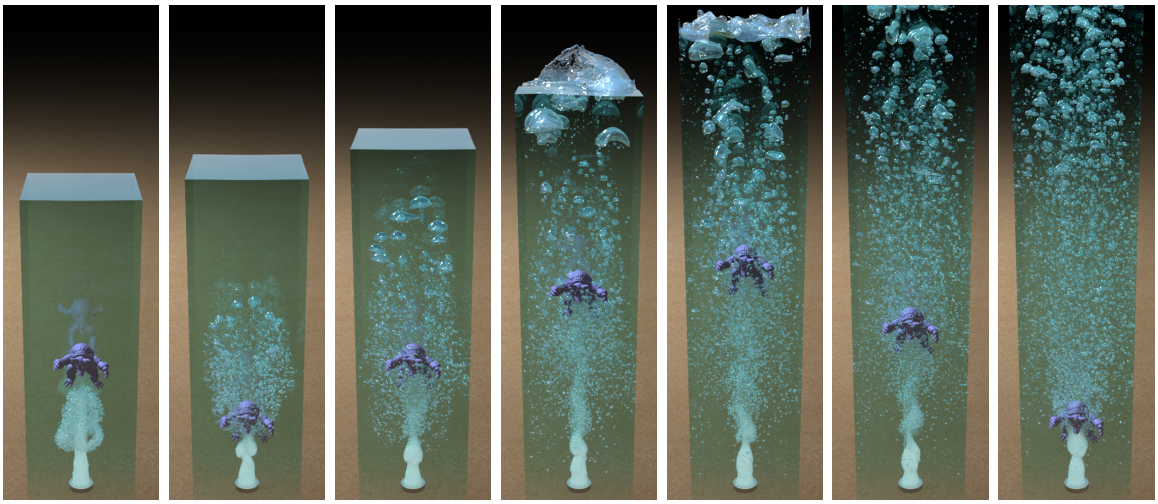


Figure 9.4: Same as Figure 9.3 except an armadillo moving up and down illustrating complex object interaction.

As discussed in Section 8.1, if a sub-grid bubble has a neighbor that lies inside an object then the weight w_i is set to zero. This means that sub-grid bubble pressures do not directly couple with solid objects. Although this is fine for kinematically coupled objects, the method should be modified in order to properly handle interactions with two-way coupled objects, increasing the overall complexity. We leave this as interesting future work. Figure 9.4 shows a kinematically coupled armadillo moving inside the underwater bubble simulation of Figure 9.3. Note that the larger level set bubbles form earlier in this example because of the merging of the sub-grid bubbles due to collisions with the armadillo.

Chapter 10

Bubble seeding

When considering fluid structure interactions with fast moving objects such as ship propellers, lower pressure regions are generated near the object and some of the water instantly vaporizes through cavitation [5] and forms bubbles. Since the density of water is a thousand times larger than that of air, these bubbles quickly expand in size becoming visible. The problem of modeling bubble generation has been addressed by various authors for phenomena such as boiling [47, 37] or air entrainment [20, 29, 46]. While the former schemes predict bubble seeding locations using temperature and objects, the latter set of schemes use the escaped level set particles. As a result, all these schemes are unsuitable for modeling cavitation. Note that it is extremely difficult to vaporize pure water due to strong cohesion forces between the water molecules. Thus, the major mechanism for cavitation is through nuclei that are very tiny bubbles (of the order of microns) already present in water, or which are generated near rough surfaces. When these bubbles enter lower pressure regions, they quickly grow in size becoming visible to the naked eye.

Although lower pressure regions might appear to be good candidates for seeding bubbles, this idea does not work well in practice because the incompressible pressure behaves like a Lagrange multiplier, as mentioned in Chapter 9. The incompressible flow velocities, on the other hand, are much more reliable. We observed that the

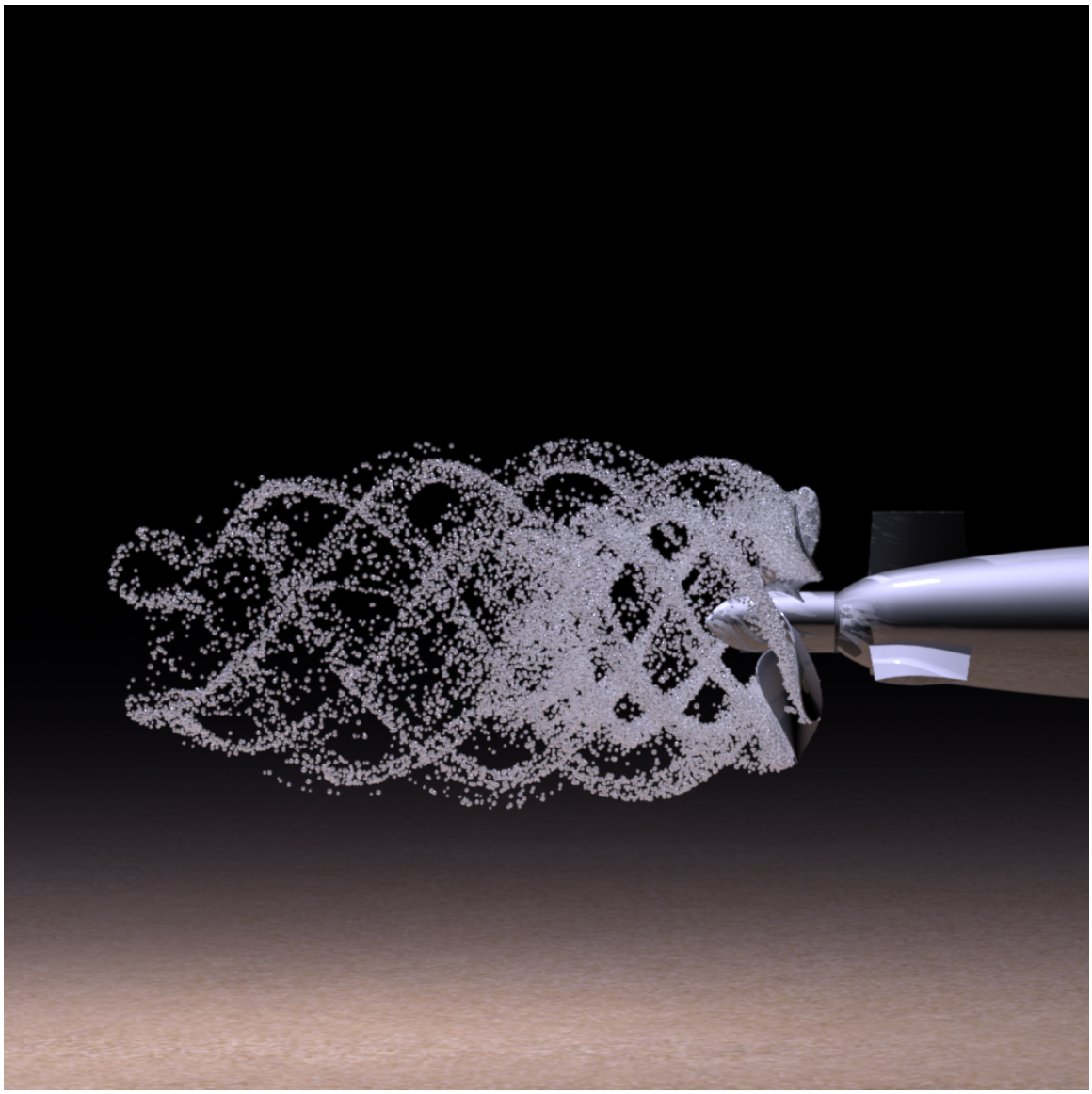


Figure 10.1: A cavitating propeller generates the characteristic helical pattern in its wake.

vorticity of the velocity field is a very good predictor for cavitating regions and thus, we determine such regions by thresholding the vorticity magnitude. Note that one should be careful when computing vorticity at fluid cells bordering objects because the vorticity magnitude can be erroneously high due to Neumann boundary conditions. To avoid this, we first extrapolate the fluid velocity inside objects and then compute vorticity. We seed bubbles with small radii, zero radial velocity, and use the steady state pressure for setting their mass, as described in Chapter 9. Once seeded, these bubbles stably grow to their correct volume in a few time steps because of the monolithic couping scheme. Figure 10.1 shows the characteristic helical pattern generated by a cavitating propeller simulated on a $256 \times 512 \times 256$ grid, and Figure 10.2 shows a cavitating hydrofoil generating the typical Von Karman vortex street.

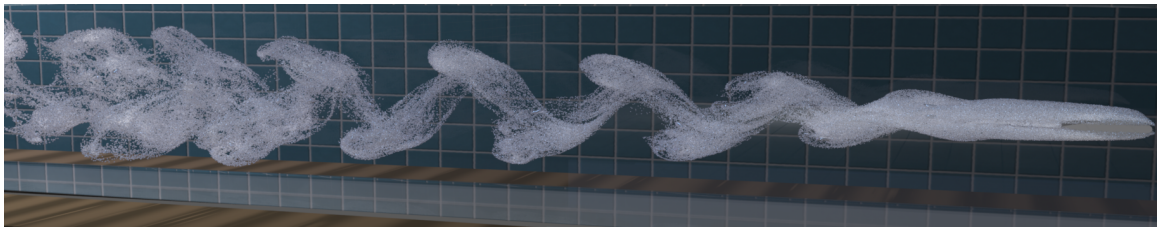


Figure 10.2: A fast moving hydrofoil generates the typical Von Karman vortex street in its wake through cavitation. The vortex street is generated because of the two-dimensional cross-sectional nature of the hydrofoil (1024×128×128 grid); about 600,000 sub-grid bubbles.

A nuclei in a lower pressure region keeps growing until it becomes large enough to affect the surrounding pressure. The number of cavitating bubbles and their size is determined by the nuclei density in water, a high nuclei density implying that there are many cavitation sites in lower pressure regions and so each nuclei can only grow by a small amount before it starts affecting the surrounding pressure - ultimately manifesting as a mist of small bubbles. Solid objects affect the nuclei density in proportion to their surface roughness, rougher surfaces generating more nuclei. Our method allows us to emulate different nuclei densities by varying the magnitude of the attraction forces between the sub-grid bubbles, a small magnitude implying a higher nuclei density as less bubbles merge together and vice-versa (see Figure 10.3).

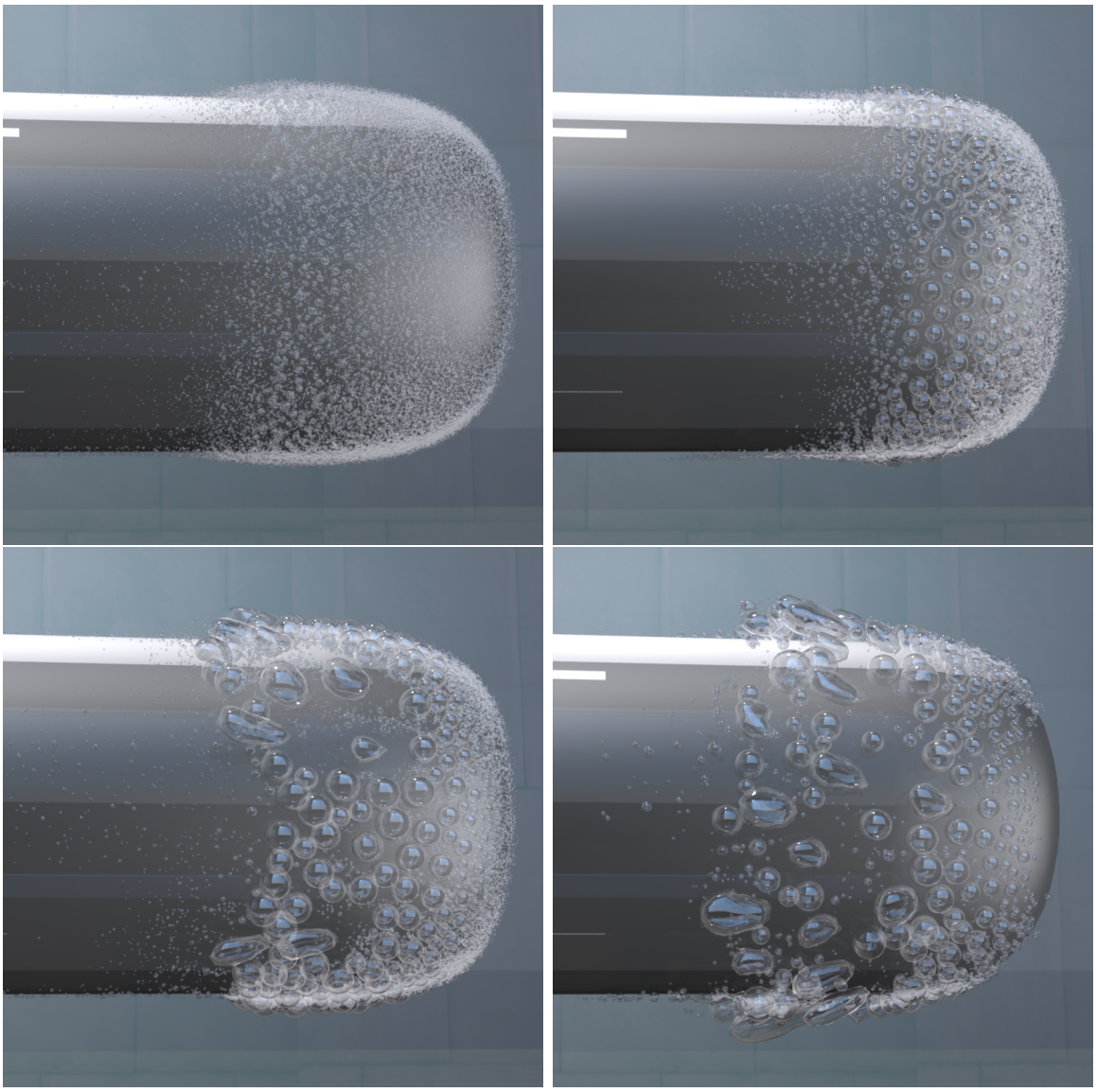


Figure 10.3: Headforms with varying surface roughness to illustrate different nuclei densities in water ranging from a large number of small bubbles to a few large bubbles (256×512×256 grid). Note that our results are qualitatively similar to the experimental results on bubble cavitation shown in Figure B.3 in [61].

Figure 10.4: Faucet pouring water showing air entrainment (200×400×200 grid); about 300,000 sub-grid bubbles. Note that the size of the sub-grid bubbles was accentuated to highlight the complex bubble interactions and the dynamic flow field.

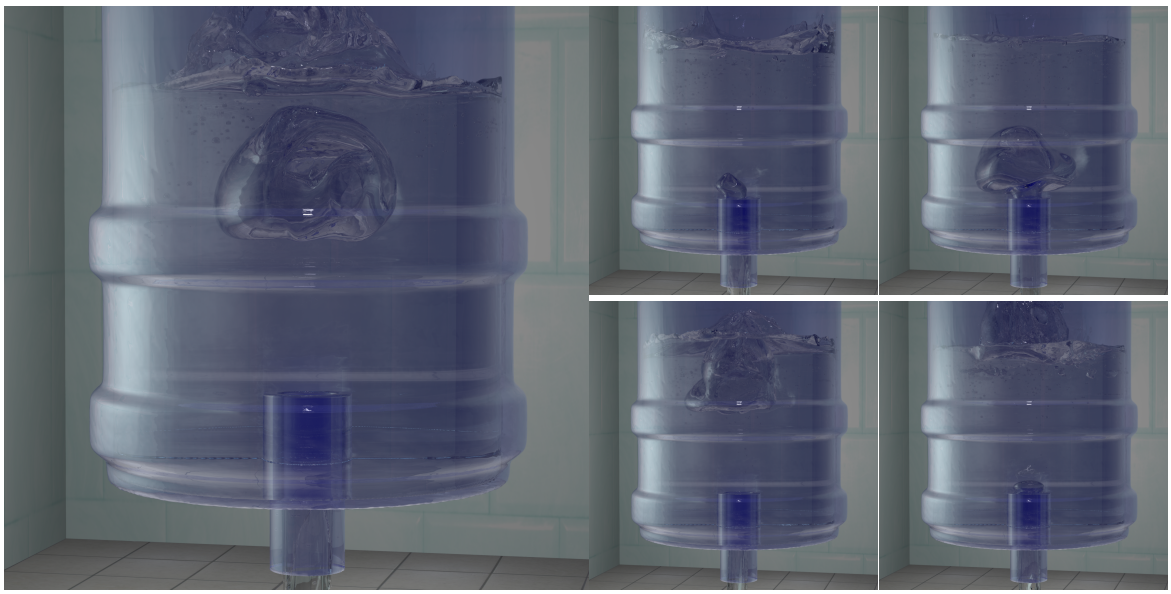


Figure 10.5: A fully simulated water dispenser (200×300×200 grid). As water exits the spout the air pressure above the free surface decreases and some air gets entrained from below to compensate for this pressure drop forming bubbles.

Although initially designed for simulating cavitation, we found that our vorticity-based seeding mechanism works well for simulating air entrainment as well because of high vorticity at the boundary between the faster impinging jet and the slower surrounding flow. Figure 10.4 shows a faucet pouring water into a container entraining bubbles. Figure 10.5 shows a fully simulated water dispenser where the free air above the water surface expands whenever water pours out decreasing the air pressure. To balance this pressure drop, some air gets entrained from below forming bubbles, and the process continues.

Chapter 11

Conclusion and Future Work

We designed a method for simulating air bubbles in free surface incompressible flows. To formulate our method, we first proposed a straightforward partitioned solver based on mass tracking. We showed that such an approach suffers from stability issues which have characteristics similar to partitioned (as opposed to monolithic) methods for solid-fluid coupling [22, 52, 21]. These issues can be alleviated using outer iterations on the partitioned solver, although the computation time increases drastically because each time step can require ten or more Poisson solves. Hence, we took a monolithic approach for the air-water problem similar to the solid-fluid coupling in [22, 52, 21] as motivated by [38]. To design this approach, we revisited the partitioned solver of [7] for coupling compressible and incompressible flow and devised a monolithic solver using the ideas from [38] to couple together incompressible flow with fully non-linear compressible flow including shocks and rarefactions. We then simplified this approach greatly to make this approach in line with our straightforward partitioned approach for simulating bubbles. This was achieved by setting both the bubble density and the bubble pressure to be spatially constant, although time-varying. We demonstrated the accuracy and robustness of this method on test problems as well as more realistic problems in both one, two and three spatial dimensions.

Since the level set loses volume and cannot keep track of sub-grid scale details, we

extended the above Eulerian formulation by tracking under-resolved bubbles using Lagrangian particles that are monolithically two-way coupled to the surrounding fluid. We showed that despite the aggressive approximations made in Section 8.1, the proposed sub-grid bubble model still closely approximates the analytic bubble oscillation frequency and converges to the analytic volume as predicted by the Rayleigh-Plesset equation while remaining stable even for large time steps. We proposed a novel scheme for seamlessly interconverting between these small Lagrangian bubbles and larger well-resolved Eulerian bubbles. We also introduced a novel seeding mechanism to realistically generate bubbles when simulating fluid structure interaction with complex objects such as ship propellers.

In the future, we would like to augment our solver with a sound simulation system. Since we have already shown that our proposed model closely approximates the analytic bubble oscillation frequency, we believe that such a system would produce realistic fluid sound effects. It would be interesting to track bubbles using deforming chimera grids [10] that also split and merge along with the bubbles to achieve higher level of detail near the bubble-water interface. Another interesting avenue for future work is to include lower-dimensional surface tension effects to model bubble-bubble interactions such as bubble stacking. Eventually, we would like to integrate our bubble solver into a full-fledged ship simulation to study effects of bubbles on ship movements by two-way coupling the bubbly liquid with solid objects.

Appendix A

Oscillating bubble problems

Consider an infinitesimal element with volume $d\Omega$ and force per unit volume ∇p, implying that the total force on this element is $\nabla p d\Omega$. The work done in displacing this element in an infinitesimal time interval dt is given by $\nabla p d\Omega \cdot d\vec{x} = \nabla p d\Omega \cdot \vec{v} dt$. The total work done dW in the time interval dt is the integral of this work over the entire domain, i.e.,

$$dW = \int_{\Omega} \nabla p \cdot \vec{v} d\Omega dt = \int_{\Omega} \nabla \cdot (p\vec{v}) d\Omega dt \tag{A.1}$$

since $\nabla \cdot \vec{v} = 0$. Using the divergence theorem, this integral is equivalent to the following surface integral

$$dW = \oint_{\partial\Omega} p\vec{v} \cdot \vec{n} dS dt \tag{A.2}$$

which we use in the following subsections.

Appendix-I

Consider the one dimensional oscillating bubble problem introduced in Section 2.2 and Figure 2.1. Since the system is symmetric about the midpoint we only consider the right-half of the domain. In one spatial dimension, water being incompressible has a spatially constant velocity. Since there is no mass transfer between the air bubble and the water this is also the velocity of the bubble-water interface, or the rate of change of the radius of the bubble. Thus at time t, $v(t) = \dot{R}(t)$ at the interface of the bubble of radius $R(t)$. If l is the length of the water region, then $\rho_I l$ is its mass. Let $p_{\mathsf{b}}(t)$ be the pressure inside the bubble and $p_{\mathsf{atm}}(t)$ be the pressure in the air at time t. The total force on the water is given by $p_{\mathsf{b}}(t) - p_{\mathsf{atm}}(t)$. This must equal the mass times the acceleration $\ddot{R}(t)$ of water, i.e.,

$$p_{\mathsf{b}}(t) - p_{\mathsf{atm}}(t) \quad = \quad \rho_I l \ddot{R}(t) \tag{A.3}$$

Using an equation of state $p = B\rho$, it follows that $p_{\mathsf{b}}(t) = B\rho_{\mathsf{b}}(t) = BM/R(t)$, where M is the constant mass of the bubble. Substituting this in equation (A.3) gives

$$\ddot{R}(t) = \frac{1}{\rho_I l}\left(\frac{BM}{R(t)} - p_{\mathsf{atm}}(t)\right) \tag{A.4}$$

Appendix-II

Consider the oscillating bubble problem in two spatial dimensions as shown in Figure 2.2. Since the total volume of water is conserved, the radius of the water sphere $R_w(t)$ is dependent on the radius of the bubble $R(t)$. Let R^0 be the initial radius of the bubble and R_w^0 be that of the water drop. Then conservation of volume yields

$$\begin{aligned} \pi((R_w^0)^2 - (R^0)^2) &= \pi(R_w(t)^2 - R(t)^2) \\ R_w(t) &= \sqrt{(R_w^0)^2 + R(t)^2 - (R^0)^2} \\ R_w(t) &= \sqrt{a^2 + R(t)^2}, \text{ where } a^2 = (R_w^0)^2 - (R^0)^2 \end{aligned} \tag{A.5}$$

In fact, the volume is conserved for any annulus with inner radius $R(t)$ and outer radius r, so that $\pi(r^2 - R(t)^2)$ is constant and

$$r\dot{r} = R(t)\dot{R}(t) \tag{A.6}$$

At any point in time the kinetic energy of an annulus of infinitesimal thickness dr and radius r is given by $\frac{1}{2}(\rho_I 2\pi r dr) \cdot v(r)^2$, where $v(r) = \dot{r} = R(t)\dot{R}(t)/r$ from equation (A.6). The total kinetic energy of water is obtained by integrating this from the radius of the bubble $R(t)$ to the outer radius of water $R_w(t)$, i.e.,

$$K.E.(t) = \rho_I \pi \int_{R(t)}^{R_w(t)} (R(t)\dot{R}(t)/r)^2 r dr = \pi \rho_I (R(t)\dot{R}(t))^2 \ln(R_w(t)/R(t)) \tag{A.7}$$

Equating the work done with the kinetic energy, and noting that equation (A.2) applied to our annulus gives two terms of the form $2\pi r\dot{r}$ results in

$$\int_0^t p_{\mathsf{b}}(\tau) 2\pi R(\tau)\dot{R}(\tau) d\tau - p_{\mathsf{atm}}(\tau) 2\pi R_w(\tau)\dot{R}_w(\tau) d\tau = \pi \rho_I (R(t)\dot{R}(t))^2 \ln(R_w(t)/R(t)) \tag{A.8}$$

From equation (A.6), $R_w(t)\dot{R}_w(t) = R(t)\dot{R}(t)$ for any time t and thus

$$2\int_0^t (p_{\mathsf{b}}(\tau) - p_{\mathsf{atm}}(\tau)) R(\tau)\dot{R}(\tau) d\tau = \rho_I (R(t)\dot{R}(t))^2 \ln(\sqrt{a^2 + R(t)^2}/R(t)) \tag{A.9}$$

Differentiating both sides with respect to t and simplifying gives

$$p_{\mathsf{b}}(t) - p_{\mathsf{atm}}(t) = \rho_I (\ddot{R}(t)R(t) + \dot{R}(t)^2) \ln(\sqrt{a^2 + R(t)^2}/R(t)) - .5a^2 \rho_I \dot{R}(t)^2/(a^2 + R(t)^2) \tag{A.10}$$

Substituting $p_{\mathsf{b}}(t) = B\rho_{\mathsf{b}}(t) = BM/(\pi R(t)^2)$ and rearranging gives

$$\begin{aligned}\rho_I \ddot{R}(t)R(t)\ln\left(\frac{\sqrt{a^2+R(t)^2}}{R(t)}\right) &+ \rho_I \dot{R}(t)^2\left(\ln\left(\frac{\sqrt{a^2+R(t)^2}}{R(t)}\right) - \frac{.5a^2}{a^2+R(t)^2}\right) \\ &- \frac{BM}{\pi R(t)^2} + p_{\mathsf{atm}}(t) = 0 \end{aligned} \tag{A.11}$$

Appendix-III

Finally, consider the three dimensional case. Equation (A.5) becomes

$$R_w(t) = \sqrt[3]{a^3 + R(t)^3}, \text{ where } a^3 = (R_w^0)^3 - (R^0)^3 \tag{A.12}$$

equation (A.6) becomes

$$r^2\dot{r} = R(t)^2\dot{R}(t) \tag{A.13}$$

and equation (A.7) becomes

$$K.E.(t) = 2\pi\rho_I \int_{R(t)}^{R_w(t)} (R(t)^2\dot{R}(t)/r^2)^2 r^2 dr = 2\pi\rho_I R(t)^4 \dot{R}(t)^2 (1/R(t) - 1/R_w(t)) \tag{A.14}$$

Equating the work done with the kinetic energy, and noting that equation (A.2) applied to our thickened shell gives two terms of the form $4\pi r^2\dot{r}$ results in

$$\int_0^t p_{\mathsf{b}}(\tau)4\pi R(\tau)^2\dot{R}(\tau)d\tau - p_{\mathsf{atm}}(\tau)4\pi R_w(\tau)^2\dot{R}_w(\tau)d\tau = 2\pi\rho_I R(t)^4\dot{R}(t)^2(1/R(t) - 1/R_w(t)) \tag{A.15}$$

From equation (A.13), $R_w^2(t)\dot{R}_w(t) = R^2(t)\dot{R}(t)$ for any time t and thus

$$2\int_0^t (p_{\mathsf{b}}(\tau) - p_{\mathsf{atm}}(\tau))R(\tau)^2\dot{R}(\tau)d\tau = \rho_I R(t)^4\dot{R}(t)^2\left(1/R(t) - 1/\sqrt[3]{a^3 + R(t)^3}\right) \tag{A.16}$$

Differentiating both sides with respect to t and simplifying gives

$$\begin{aligned} p_{\mathsf{b}}(t) - p_{\mathsf{atm}}(t) &= \rho_I\ddot{R}(t)\left(R(t) - \frac{R(t)^2}{\sqrt[3]{a^3 + R(t)^3}}\right) \\ &+ \rho_I\dot{R}(t)^2\left(\frac{3}{2} + \frac{R(t)^2}{\sqrt[3]{a^3 + R(t)^3}^2} - \frac{2R(t)}{\sqrt[3]{a^3 + R(t)^3}}\right) \end{aligned} \tag{A.17}$$

Substituting $p_{\mathsf{b}}(t) = B\rho_{\mathsf{b}}(t) = BM/(\frac{4}{3}\pi R(t)^3)$ and rearranging gives

$$\begin{aligned} \rho_I\ddot{R}(t)\left(R(t) - \frac{R(t)^2}{\sqrt[3]{a^3 + R(t)^3}^2}\right) &+ \rho_I\dot{R}(t)^2\left(\frac{3}{2} + \frac{R(t)^2}{\sqrt[3]{a^3 + R(t)^3}^2} - \frac{2R(t)}{\sqrt[3]{a^3 + R(t)^3}}\right) \\ &- \frac{BM}{\frac{4}{3}\pi R(t)^3} + p_{\mathsf{atm}}(t) = 0 \end{aligned} \tag{A.18}$$

Bibliography

[1] D. Adalsteinsson and J. Sethian. The fast construction of extension velocities in level set methods. J. Comput. Phys., 148:2–22, 1999.

[2] K. Atkinson. An Introduction to Numerical Analysis. John Wiley and Sons, 1989.

[3] Landon Boyd and Robert Bridson. Multiflip for energetic two-phase fluid simulation. ACM Trans. Graph., 31(2):16:1–16:12, 2012.

[4] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modelling surface tension. J. Comput. Phys., 100:335–353, 1992.

[5] Christopher E. Brennen. Cavitation and Bubble Dynamics. Oxford University Press, USA, 1995.

[6] Oleksiy Busaryev, Tamal K. Dey, Huamin Wang, and Zhong Ren. Animating bubble interactions in a liquid foam. ACM Trans. Graph., 31(4):63:1–63:8, 2012.

[7] R. Caiden, R. Fedkiw, and C. Anderson. A numerical method for two phase flow consisting of separate compressible and incompressible regions. J. Comput. Phys., 166:1–27, 2001.

[8] A.J. Chorin. Numerical solution of the Navier-Stokes Equations. Math. Comp., 22:745–762, 1968.

[9] Paul W. Cleary, Soon Hyoung Pyo, Mahesh Prakash, and Bon Ki Koo. Bubbling and frothing liquids. ACM Trans. Graph., 26(3), 2007.

[10] R.E. English, L. Qiu, Y. Yu, and R. Fedkiw. An adaptive discretization of incompressible flow using a multitude of moving cartesian grids. (submitted), 2012.

[11] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. J. Comput. Phys., 183:83–116, 2002.

[12] D. Enright, F. Losasso, and R. Fedkiw. A fast and accurate semi-Lagrangian particle level set method. Computers and Structures, 83:479–490, 2005.

[13] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. ACM Trans. Graph. (SIGGRAPH Proc.), 21(3):736–744, 2002.

[14] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). J. Comput. Phys., 152:457–492, 1999.

[15] R. Fedkiw, X.-D. Liu, and S. Osher. A general technique for eliminating spurious oscillations in conservative schemes for multiphase and multispecies euler equations. Int. J. Nonlinear Sci. and Numer. Sim., 3:99–106, 2002.

[16] N. Foster and R. Fedkiw. Practical animation of liquids. In Proc. of ACM SIGGRAPH 2001, pages 23–30, 2001.

[17] D. Fuster and T. Colonius. Modelling bubble clusters in compressible liquids. J. Fluid. Mech., 688:352–389, 2011.

[18] W. Geiger, M. Leo, N. Rasmussen, F. Losasso, and R. Fedkiw. So real it'll make you wet. In SIGGRAPH 2006 Sketches & Applications. ACM Press, 2006.

[19] F. Gibou, R. Fedkiw, L.-T. Cheng, and M. Kang. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. J. Comput. Phys., 176:205–227, 2002.

[20] S. T. Greenwood and D. H. House. Better with bubbles: enhancing the visual realism of simulated fluid. In Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim., pages 287–296, 2004.

[21] J. Grétarsson and R. Fedkiw. Fully conservative, robust treatment of thin shell fluid-structure interactions in compressible flows. (submitted), 2012.

[22] J.T. Grétarsson, N. Kwatra, and R. Fedkiw. Numerically stable fluid-structure interactions between compressible flow and solid structures. J. Comput. Phys., 230:3062–3084, 2011.

[23] E. Guendelman, A. Selle, F. Losasso, and R. Fedkiw. Coupling water and smoke to thin deformable and rigid shells. ACM Trans. Graph. (SIGGRAPH Proc.), 24(3):973–981, 2005.

[24] Denis Gueyffier, Jie Li, Ali Nadim, Ruben Scardovelli, and Stphane Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. Journal of Computational Physics, 152(2):423 – 456, 1999.

[25] J. G. Hnat and J. D. Buckmaster. Spherical cap bubbles and skirt formation. Physics of Fluids, 19:182–194, 1976.

[26] J.-M. Hong and C.-H. Kim. Discontinuous fluids. ACM Trans. Graph. (SIGGRAPH Proc.), 24(3):915–920, 2005.

[27] J.-M. Hong, T. Shinar, M. Kang, and R. Fedkiw. On boundary condition capturing for multiphase interfaces. J. Sci. Comput., 31:99–125, 2007.

[28] Jeong-Mo Hong and Chang-Hun Kim. Animation of bubbles in liquid. Comput. Graph. Forum (Eurographics Proc.), 22(3):253–262, 2003.

[29] Jeong-Mo Hong, Ho-Young Lee, Jong-Chul Yoon, and Chang-Hun Kim. Bubbles alive. ACM Trans. Graph., 27(3):48:1–48:4, 2008.

[30] Markus Ihmsen, Nadir Akinci, Gizem Akinci, and Matthias Teschner. Unified spray, foam and air bubbles for particle-based fluids. Vis. Comput., 28(6-8):669–677, 2012.

[31] E. Johnsen and T. Colonius. Numerical simulations of non-spherical bubble collapse. J. Fluid. Mech., 629:231–262, 2009.

[32] Eric Johnsen and Tim Colonius. Implementation of weno schemes in compressible multicomponent flow problems. J. Comput. Phys., 219(2):715 – 732, 2006.

[33] M. Kang, R. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. J. Sci. Comput., 15:323–360, 2000.

[34] Byungmoon Kim, Yingjie Liu, Ignacio Llamas, Xiangmin Jiao, and Jarek Rossignac. Simulation of bubbles in foam with the volume control method. ACM Trans. Graph., 26(3), 2007.

[35] Doyub Kim, Oh-young Song, and Hyeong-Seok Ko. A practical simulation of dispersed bubble flow. ACM Trans. Graph., 29:70:1–70:5, 2010.

[36] Po-Ram Kim, Ho-Young Lee, Jong-Hyun Kim, and Chang-Hun Kim. Controlling shapes of air bubbles in a multi-phase fluid simulation. Vis. Comput., 28(6-8):597–602, 2012.

[37] Theodore Kim and Mark Carlson. A simple boiling module. In ACM SIGGRAPH/Eurographics Symp. on Comput. Anim., pages 27–34, 2007.

[38] N. Kwatra, J. Su, J.T. Grétarsson, and R. Fedkiw. A method for avoiding the acoustic time step restriction in compressible flow. J. Comput. Phys., 228(11):4146–4161, 2009.

[39] Ho-Young Lee, Jeong-Mo Hong, and Chang-Hun Kim. Interchangeable sph and level set method in multiphase fluids. Vis. Comput., 25(5-7):713–718, 2009.

[40] M. Lentine, M. Cong, S. Patkar, and R. Fedkiw. Simulating free surface flow with very large time steps. In ACM SIGGRAPH/Eurographics Symp. on Comput. Anim. 2012, pages 107–116, 2012.

[41] M. Lentine, J.T. Grétarsson, and R. Fedkiw. An unconditionally stable fully conservative semi-lagrangian method. J. Comput. Phys., 230:2857–2879, 2011.

[42] Xu-Dong Liu, Ronald P. Fedkiw, and Myungjoo Kang. A boundary condition capturing method for poisson's equation on irregular domains. J. Comput. Phys., 160(1):151–178, 2000.

[43] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. Computers and Fluids, 35:995–1010, 2006.

[44] F. Losasso, T. Shinar, A. Selle, and R. Fedkiw. Multiple interacting liquids. ACM Trans. Graph. (SIGGRAPH Proc.), 25(3):812–819, 2006.

[45] F. Losasso, J. Talton, N. Kwatra, and R. Fedkiw. Two-way coupled sph and particle level set fluid simulation. IEEE TVCG, 14(4):797–804, 2008.

[46] V. Mihalef, D. N. Metaxas, and Mark Sussman. Simulation of two-phase flow with sub-scale droplet and bubble effects. Comput. Graph. Forum, 2009.

[47] V. Mihalef, B. Unlusu, D. Metaxas, M. Sussman, and M. Hussaini. Physics based boiling simulation. In SCA '06: Proc. of the 2006 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim., pages 317–324, 2006.

[48] William Moss, Hengchin Yeh, Jeong-Mo Hong, Ming C. Lin, and Dinesh Manocha. Sounding liquids: Automatic sound synthesis from fluid simulation. ACM TOG, 29(3):21:1–21:13, 2010.

[49] M. Müller, B. Solenthaler, R. Keiser, and M. Gross. Particle-based fluid-fluid interaction. In Proc. of the 2005 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim., pages 237–244, 2005.

[50] S. Osher and R. Fedkiw. Level Set Methods and Dynamic Implicit Surfaces. Springer-Verlag, 2002. New York, NY.

[51] N. Rasmussen, D. Enright, D. Nguyen, S. Marino, N. Sumner, W. Geiger, S. Hoon, and R. Fedkiw. Directable photorealistic liquids. In Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Comput. Anim., pages 193–202, 2004.

[52] A. Robinson-Mosher, C. Schroeder, and R. Fedkiw. A symmetric positive definite formulation for monolithic fluid structure interaction. J. Comput. Phys., 230:1547–66, 2011.

[53] A. Selle, R. Fedkiw, B. Kim, Y. Liu, and J. Rossignac. An Unconditionally Stable MacCormack Method. J. Sci. Comp., 35(2):350–371, 2008.

[54] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. J. Comput. Phys., 77:439–471, 1988.

[55] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes II (two). J. Comput. Phys., 83:32–78, 1989.

[56] M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. J. Comput. Phys., 187:110–136, 2003.

[57] M. Sussman and P. Smereka. Axisymmetric free boundary problems. J. Fluid Mech., 341:269–294, 1997.

[58] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. J. Comput. Phys., 114:146–159, 1994.

[59] N. Thürey, F. Sadlo, S. Schirm, M. Müller-Fischer, and M. Gross. Real-time simulations of bubbles and foam within a shallow water framework. In SCA '07: Proc. of 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 191–198, 2007.

[60] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multifluid flows. J. Comput. Phys., 100:25–37, 1992.

[61] T. J. C. van Terwisga. Cavitation on ship propellers. http://ocw.tudelft.nl/courses/marine-technology/cavitation-on-ship-propellers/.

[62] Changxi Zheng and Doug L. James. Harmonic fluids. ACM Trans. Graph. (SIGGRAPH Proc.), 28(3):37:1–37:12, 2009.

[63] W. Zheng, J.-H. Yong, and J.-C. Paul. Simulation of bubbles. In SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 325–333, 2006.