UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Improvements in Computing Multiple Phase Flows (Ph.D. Thesis)

Brian James Miller

December 1997

CAM Report 97-55

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

UNIVERSITY OF CALIFORNIA

Los Angeles

Improvements in Computing Multiple Phase Flows

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Mathematics

by

Brian James Miller

1997

The dissertation of Brian James Miller is approved.

T.H.K Frederking

Tony Chan

Chris Anderson

Stanley Osher, Committee Chair

University of California, Los Angeles

1997

To my wife Amy, without your support and encouragement this would have been impossible.

Table of Contents

List of Figures

List of Tables

VITA

1970	Born, Wausau, Wisconsin, USA.
1992	B.S. (Mathematics), MTU
1992-1997	Teaching Assistant, Mathematics Department, UCLA
1994	M.S. (Mathematics), UCLA
1992-1997	Research Assistant, Mathematics Department, UCLA

PUBLICATIONS

David L. Brown, M. Gregory Forest, Brian J. Miller, and N. Anders Petersson, "Computation and Stability of Fluxons in a Singularly Perturbed Sine-Gordon Model of the Josephson Junction", *SIAM Journal of Applied Mathematics*, **54**:1048-1066, 1994

B. Miller, B. Merriman, K. Rafique, S. Osher, N. Noushkam, A. Spletzer, and T.H.K. Frederking, "Vapor-Liquid Interfacial Dynamics and Related Liquid Nitrogen Boiling on Perforated Plates", to appear in *Space Cryogenics Workshop, Eugene, OR, Aug. 4-5, 1997*

ABSTRACT OF THE DISSERTATION

Improvements in Computing Multiple Phase Flows

by

Brian James Miller

Doctor of Philosophy in Mathematics

University of California, Los Angeles, 1997

Professor Stanley Osher, Chair

Improvements are made to a level set method for solving the incompressible, immiscible Navier-Stokes equations for fluids separated by an interface. A variable density projection method combined with a TVD Runge-Kutta scheme is used to advance the computed solution in time. The improvements allow the method to be applied in three dimensions without difficulty, give more freedom in choosing boundary conditions, and enable the application of new fast solver techniques. The use of the level set formulation allows easy treatment of merging and breaking flows. We examine typical problems such as air bubbles in water and water drops in air. Comparisons are made with other numerical methods and with theoretical results.

CHAPTER 1

Introduction

In this presentation, we will develop improvements to an existing formulation of the incompressible, immiscible Navier-Stokes equations for multiple fluids separated by an interface [Sus94]. In that thesis, the incompressibility constraint was enforced by taking advantage of the two dimensional nature of the problem. This also forced the non-physical choice of no-slip velocity boundary conditions. In this work, we will instead solve the equation for pressure to enforce incompressibility. This method has some advantages over the previous method. Namely, it generalizes to three dimensions easily, allows the correct velocity boundary conditions to be used, and is a natural first step if one would like to use the method described in [HLO97].

Chapter 2 summarizes the equations of motion. Section 2.6 presents the new projection algorithm, and Section 2.3 presents the new surface tension formulation. The discretization of the problem in treated in Chapter 3. The discretization of the new projection method is described in 3.4. A summary of the application of the Immersed Interface Method [HLO97] to this problem is shown in Chapter 4. Some numerical results are presented in Chapter 5. New applications of this method are described in Chapters 6 and 7. The figures for all chapters are contained in Chapter 8.

CHAPTER 2

Algorithm

2.1 Continuum Equations

The equations of motion we solve are the incompressible Navier-Stokes equations for flows involving two immiscible fluids. Denote the density and viscosity inside one fluid by ρ_1 and μ_1, and in the other fluid as ρ_2, and μ_2. The equations of motion and boundary condition on the interface are

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} = \mathbf{g} + \frac{1}{\rho}(-\nabla P + \nabla \cdot (2\mu \mathbf{D})) \tag{2.1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2.2}$$

$$(P_1 - P_2 + \alpha\kappa)\, n_i = 2\left(\mu_1 \mathbf{D}_{1,ij} - \mu_2 \mathbf{D}_{2,ij}\right) n_k + \frac{\partial \alpha}{\partial x_i} \tag{2.3}$$

where $\mathbf{u} = (u, v)$ is the fluid velocity, $\rho = \rho(\mathbf{x}, t)$ is the fluid density, $\mu = \mu(\mathbf{x}, t)$ is the fluid viscosity. The viscous stress tensor $\mathbf{D}$ is defined

$$\mathbf{D}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

$\mathbf{g}$ is the gravitational acceleration, and P is the fluid pressure. In the surface stress boundary condition (2.3), derived in [LL87], α is the surface tension coefficient which we assume is constant on the interface, κ is the local mean curvature of the interface, and $\mathbf{n}$ is the unit normal to the interface pointing into liquid 2. We also define τ to be the unit tangent to the interface. The gradient of α in (2.3) is really a surface gradient, since α is only defined on the interface.

Since we assume the fluids are immiscible, the density and viscosity simply convect along with the fluid

$$\rho_t + (\mathbf{u} \cdot \nabla)\rho = 0 \tag{2.4}$$

$$\mu_t + (\mathbf{u} \cdot \nabla)\mu = 0 \tag{2.5}$$

We specify boundary conditions on $\mathbf{u}$. For example, at a solid wall boundary, we choose the standard Dirichlet condition.

$$\mathbf{u} = 0 \tag{2.6}$$

2.2 Use of Level Set Functions

In our problems of interest there typically are large jumps in the fluid density and/or viscosity across the interface. For this reason, we try to avoid taking finite differences of ρ or μ across the interface when solving (2.4) and (2.5). The use of level set functions helps avoid doing this. The level set function $\phi(\mathbf{x}, t)$ is chosen to be positive in region 2 and negative in region 1. Therefore, the zero level set defines the interface. We choose ϕ initially to be the distance to the initial interface, with the appropriate choice of sign. If we now advect $\phi(\mathbf{x}, t)$ according to

$$\phi_t + (\mathbf{u} \cdot \nabla)\phi = 0 \tag{2.7}$$

then when we need information about ρ or μ, we evaluate ϕ at that point, and the sign of ϕ tells us the value of ρ and μ. The advantage here, is that $\phi(\mathbf{x}, t)$ is a smooth function, even near the interface so standard numerical methods will give accurate, non-oscillatory solutions.

One important point to remember is that initially, ϕ is a distance function, i.e. $|\nabla\phi| = 1$. At later times, this is not necessarily true. Periodically during

the time integration, we execute a procedure called reinitialization that forces $|\nabla\phi| = 1$ while leaving the zero level of ϕ unchanged. This procedure is detailed in Section 2.7.

2.3 Inclusion of Surface Tension

In the continuum equations, surface tension forces appear by imposing a jump in the stress tensor across the interface (2.3). We find this jump condition difficult to apply in practice and would like to reformulate it as a volume force that is applied only near the interface. In [CHM96] this was done in the context of vortex sheet problems. While in [BKZ], a similar result is arrived at using a more physical approach. Our derivation will be closer to that of [CHM96] in nature.

We define a region Ω divided by a section of the interface, Γ, in it's interior. Label the two parts Ω_1 and Ω_2, see Figure 8.1. Consider the conservation of momentum equation for Ω

$$\int_\Omega \rho \frac{D\mathbf{u}}{Dt} d\mathbf{x} = \int_\Omega \rho \mathbf{g} d\mathbf{x} + \int_{\partial\Omega} \sigma_{i,j} n_j ds \tag{2.8}$$

Where $\sigma = P\mathbf{I} - 2\mu\mathbf{D}$ is the stress tensor, and $D/Dt = \partial/\partial t + \mathbf{u} \cdot \nabla$ is the Lagrangian time derivative. If we now integrate the final term around $\partial\Omega_1$ and $\partial\Omega_2$ separately, we find that

$$\int_{\partial\Omega} \sigma_{i,j} n_j ds = \int_{\partial\Omega_1} \sigma_{i,j} n_j ds + \int_{\partial\Omega_2} \sigma_{i,j} n_j ds + \int_\Gamma [\sigma_{i,j} n_j] \, ds$$

Where $[\sigma_{i,j} n_j]$ is the jump in the normal component of the stress tensor across the interface. Now using (2.3) and Gauss' Theorem, we can simplify the above equation to

$$\begin{aligned} \int_{\partial\Omega} \sigma_{i,j} n_j ds &= \int_{\Omega_1} \frac{\partial \sigma_{i,j}}{\partial x_j} ds + \int_{\Omega_2} \frac{\partial \sigma_{i,j}}{\partial x_j} ds + \int_\Gamma \alpha\kappa\mathbf{n} ds \\ &= \int_\Omega \frac{\partial \sigma_{i,j}}{\partial x_j} ds + \int_\Gamma \alpha\kappa\mathbf{n} ds \end{aligned}$$

We now claim that

$$\int_\Gamma \alpha\kappa \mathbf{n} ds = \int_\Omega \alpha\kappa(\phi(\mathbf{x}))\delta(\phi(\mathbf{x}))\nabla\phi d\mathbf{x} \tag{2.9}$$

For the proof, we define an auxiliary function $\psi(x,y)$ satisfying $\nabla\phi\cdot\nabla\psi = 0$, and $|\nabla\psi| \neq 0$. We now make a change of variables $\eta = \psi(x,y)$, $\xi = \phi(x,y)$, which has Jacobian

$$\begin{aligned} det\left(\begin{bmatrix} \eta_x & \eta_y \\ \xi_x & \xi_y \end{bmatrix}\right) &= \eta_x\cdot\xi_y - \xi_x\cdot\eta_y = (\xi_y, -\xi_x)\cdot(\eta_x,\eta_y) \\ &= (\phi_y,-\phi_x)\cdot(\psi_x,\psi_y) = \phi_y\psi_x - \phi_x\psi_y \end{aligned} \tag{2.10}$$

Some simple algebra then reveals that

$$|\nabla\phi|^2\cdot|\nabla\psi|^2 - (\nabla\phi\cdot\nabla\psi)^2 = \phi_x^2\psi_y^2 - 2\phi_x\psi_x\phi_y\psi_y + \psi_x^2\phi_y^2 = (\phi_y\psi_x - \phi_x\psi_y)^2$$

so,

$$(\phi_y,-\phi_x)\cdot(\psi_x,\psi_y) = |\nabla\phi|\cdot|\nabla\psi| \neq 0$$

Therefore the Jacobian is non-zero and the coordinate transformation is invertible. We now apply this transformation to the right hand side of (2.9).

$$\begin{aligned} \int_\Omega \alpha\kappa(\phi(\mathbf{x}))\delta(\phi(\mathbf{x}))\nabla\phi d\mathbf{x} &= \int_{\Omega'} \alpha\kappa(\xi)\delta(\xi)\nabla\phi\cdot\frac{1}{|\nabla\phi||\nabla\psi|}d\xi d\eta \\ &= \int_{\phi=0} \alpha\kappa\frac{\mathbf{n}}{|\nabla\psi|}d\eta \end{aligned} \tag{2.11}$$

Since $\nabla\phi/|\nabla\phi| = \mathbf{n}$. Finally, if we parameterize the interface $\Gamma = (\bar{x}(s),\bar{y}(s))$ by arc length s, then

$$\begin{aligned} d\eta = d\psi(\bar{x}(s),\bar{y}(s)) &= (\psi_x\bar{x}_s + \psi_y\bar{y}_s)ds \\ &= (\psi_x,\psi_y)\cdot(\bar{x}_s,\bar{y}_s)ds = |\nabla\psi|ds \end{aligned}$$

Since (ψ_x, ψ_y) is in the same direction as the unit tangent to Γ, $(\bar{x}_s, \bar{y}_s)$. So we see

$$\int_{\phi=0} \alpha\kappa \frac{\mathbf{n}}{|\nabla\psi|} d\eta = \int_\Gamma \alpha\kappa \mathbf{n} ds$$

and the proof is complete. The construction of the appropriate ψ function is completed in [CHM96].

We have now transformed the boundary condition (2.3) defined only across the interface Γ to an equivalent volume forcing term

$$\alpha\kappa(\phi)\delta(\phi)\nabla\phi \tag{2.12}$$

valid throughout the whole domain.

We would like to make one additional simplification. The surface tension forcing (2.12) will be multiplied by $1/\rho$ in the continuum equations. In the derivation above, we could have multiplied each integrand in (2.9) by

$$\frac{\rho}{\langle\rho\rangle}$$

where $\langle\rho\rangle = (1 + \rho_2/\rho_1)/2$. The proof above is unchanged and the result is that the term in the continuum equations will be independent of ρ, depending only on $\langle\rho\rangle$. Finally, our new surface tension term becomes

$$\frac{\rho}{\langle\rho\rangle}\alpha\kappa(\phi)\delta(\phi)\nabla\phi \tag{2.13}$$

2.4 Smoothing of the Interface

There are two problems with the formulation as it stands. First, despite all the care we have taken to avoid differencing ρ or μ across the interface, sometimes we have no choice. For instance, in (2.22), if we treat ρ as a discontinuous function,

the linear system we will have to solve will be very ill-conditioned. The other difficulty we face is that in Section 2.3 we have defined a volume force which contains a Dirac δ function.

One solution to these problems is to define a small region about the interface and make all transitions smooth from one fluid to another. Numerically, we give this region a width of a few grid points. We define ϵ to be the width of the transition region and define a smoothed Heaviside function.

$$H_\epsilon(\phi) = \begin{cases} 0 & \phi < -\epsilon, \\ \frac{1}{2}\left(1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin(\pi\phi/\epsilon)\right) & |\phi| < \epsilon, \\ 1 & \phi > \epsilon \end{cases} \tag{2.14}$$

We also define the derivative with respect to ϕ of this function and define it as an approximate delta function, since it satisfies the definition of the Dirac delta function in the limit $\epsilon \to 0$.

$$\delta_\epsilon(\phi) = \begin{cases} \frac{1}{2\epsilon}\left(1 + \cos(\pi\phi/\epsilon)\right) & |\phi| < \epsilon, \\ 0 & |\phi| \geq \epsilon \end{cases} \tag{2.15}$$

Then we redefine $\rho(\phi) = \rho_1 \cdot H_\epsilon(\phi) + \rho_2(1 - H_\epsilon(\phi))$, μ is redefined likewise. With this definition, ρ and μ are now continuous and the transition region ϵ can be chosen to be $\mathcal{O}(h)$. We also modify (2.12) to use our newly defined smooth delta function. The forcing due to the surface tension is now

$$\mathbf{F}_{ST} = \frac{\rho}{\langle\rho\rangle}\alpha\kappa(\phi)\delta_\epsilon(\phi)\nabla\phi \tag{2.16}$$

We now re-state the continuum equations we will solve using the simplifications we have derived.

$$\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} = \mathbf{g} + \frac{1}{\rho}\left(-\nabla P + \nabla\cdot(2\mu\mathbf{D})\right) + \frac{1}{\langle\rho\rangle}\alpha\kappa(\phi)\delta_\epsilon(\phi)\nabla\phi \tag{2.17}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{2.18}$$

2.5 Nondimensionalization

To non-dimensionalize the equations, we need to choose a characteristic speed U and length R. For most interface motion problems, a convenient choice for the characteristic length is an initial dimension of one of the phases. For problems without an obvious choice of characteristic speed, we may choose $U = \sqrt{Rg}$. After non-dimensionalization, the equations of motion (2.17) become

$$\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{Fr}\hat{y} + \frac{1}{\rho}\left(-\nabla P + \frac{1}{Re}\nabla\cdot(2\mu\mathbf{D})\right) + \frac{1}{We}\frac{1}{\langle\rho\rangle}\kappa\delta_\epsilon(\phi)\nabla\phi \tag{2.19}$$

Which we express in a simplified notation as

$$\mathbf{u}_t = \mathcal{L}(\mathbf{u}) - \frac{1}{\rho}\nabla P \tag{2.20}$$

The dimensionless density and viscosity have value 1 in region 1 and ρ_2/ρ_1, μ_2/μ_1 respectively inside region 2. Our dimensionless parameters are the Reynolds Number ($Re = RU\rho_1/\mu_1$), the Weber Number ($We = \rho_1 RU^2/\alpha$), and the Froude Number ($Fr = U^2/(Rg)$).

2.6 Projection

The presence of the pressure term in (2.19) forces the velocity to remain divergence free. We can use this to construct the equation that the pressure must satisfy. If we take the divergence of $\mathbf{u}_t = \mathcal{L}(\mathbf{u}) - \frac{1}{\rho}\nabla P$,

$$\nabla\cdot\mathbf{u}_t = \nabla\cdot\mathcal{L}(\mathbf{u}) - \nabla\cdot(\frac{1}{\rho}\nabla P) \tag{2.21}$$

and notice that $\nabla\cdot\mathbf{u}_t = \partial(\nabla\cdot\mathbf{u})/\partial t = 0$, then

$$\nabla\cdot(\frac{1}{\rho}\nabla P) = \nabla\cdot\mathcal{L}(\mathbf{u}) \tag{2.22}$$

Similarly, we can define Neumann boundary conditions on P

$$\frac{1}{\rho}\frac{\partial P}{\partial \mathbf{n}} = \mathcal{L}(\mathbf{u}) \cdot \mathbf{n} \tag{2.23}$$

We have specified an elliptic equation for P with Neumann boundary conditions. It is well known that the solution to this problem only exists if a compatibility condition between the forcing and boundary conditions is satisfied. Even when the solution does exist, it is not unique; any constant added to P also solves (2.22) and (2.23). This non-uniqueness makes the solution of discrete pressure equation problematic.

2.6.1 Contribution of Surface Tension Force

When we compute the forcing term for the pressure equation (2.22), we need to take the divergence of (2.13). We can make use of the fact that

$$\delta(\phi)\nabla\phi = \nabla H(\phi)$$

It is not too difficult to show that

$$\begin{aligned}\frac{1}{We}\frac{1}{\langle\rho\rangle}\nabla\cdot(\kappa\nabla H) &= \frac{1}{We}\frac{1}{\langle\rho\rangle}\left[\nabla\kappa\cdot\nabla H + \kappa\nabla\cdot\nabla H\right] \\ &= \frac{1}{We}\frac{1}{\langle\rho\rangle}\left[\kappa H''\right]\end{aligned} \tag{2.24}$$

Where we have used the fact that $|\nabla\phi| = 1$ near the interface for a distance function. See Appendix A for details used in the above derivation. When we compute this term, we can use the analytic formula for $H_\epsilon(\phi)$, (2.14) to compute the second derivative in (2.24).

2.7 Re-initialization of ϕ

In the previous section, we relied heavily on the assumption that ϕ is a distance function, at least near the interface. There are many references about an iterative

procedure to force ϕ to satisfy $|\nabla\phi| = 1$ near the interface without moving the location of the interface[Sus94, CHM96]. To summarize, given a level set function which is not a distance function $\phi_0(\mathbf{x})$, we iterate the following problem to steady state

$$\phi_\tau = S(\phi_0)\left(1 - \sqrt{\phi_x^2 + \phi_y^2}\right) \tag{2.25}$$

$$\phi(\mathbf{x}, \tau = 0) = \phi_0(\mathbf{x}) \tag{2.26}$$

where S is a smoothed sign function

$$S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + h^2}}$$

This iteration gives a function $\phi(\mathbf{x})$ that has the same zero level set as $\phi_0(\mathbf{x})$, but is also a distance function near the interface. In practice, we only need to iterate a few times to achieve the distance function property within the transition region.

2.8 Algorithm Summary

- Initialize $\phi(\mathbf{x}, 0)$ to be the signed distance to the initial interface. Set velocity $\mathbf{u} = 0$.
- Advance $\mathbf{u}$ and ϕ one time step using a TVD Runge-Kutta scheme with the projection method.
- Reinitialize ϕ.

CHAPTER 3

Discretization

We limit our domain to be rectangular. Furthermore, we only allow equal grid spacing in each dimension. This is not a limit of the method, it is only done for easier programming.

If our domain is $\{(x,y) : a < x < b, c < y < d\}$, and our grid spacing is h, with nx grid points in x and ny grid points in y, then our grid points are

$$\mathbf{x}_{i,j} = (a + (i - .5)h, c + (j - .5)h)$$
$$i = -1, 0, 1, 2, \ldots, nx, nx + 1, nx + 2$$
$$j = -1, 0, 1, 2, \ldots, ny, ny + 1, ny + 2$$

The grid points whose indices put them outside the domain are called "ghost points" and are used to enforce the boundary conditions and to allow for easy application of difference approximations. All unknowns $\mathbf{u}$, ϕ, and P are defined at these grid points. See Figure 8.2 for an example of a grid with one layer of ghost points outside the computational domain. The open circles denote the locations where the unknowns are defined.

3.1 Spatial Discretization

All non-linear and convection terms are discretized using a 2nd or 3rd order ENO method [SO89]. Suppose we wish to compute a second order approximation to

$u \cdot u_{\mathbf{x}}$, part of the convection term $\mathbf{u} \cdot \nabla \mathbf{u}$, evaluated at the grid point $\mathbf{x}_{i,j}$. For notational convenience, we drop the j index, since we are only concerned with variations in the i dimension.

There are two cases, if $u_i > 0$, then

$$u_x(\mathbf{x}_i) \approx \frac{u_i - u_{i-1}}{h} + \frac{h}{2} minmod\left(\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}, \frac{u_i - 2u_{i-1} + u_{i-2}}{h^2}\right)$$

and if $u_i < 0$, then

$$u_x(\mathbf{x}_i) \approx \frac{u_{i+1} - u_i}{h} - \frac{h}{2} minmod\left(\frac{u_{i+2} - 2u_{i+1} + u_i}{h^2}, \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}\right)$$

Where

$$minmod(a,b) = \begin{cases} sgn(a)\min(|a|,|b|) & a \cdot b > 0, \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Approximations of y derivatives are similar. To achieve higher order accuracy, we add additional terms to the approximation which are computed in the same manner.

The viscous term, $1/(\rho Re)\nabla \cdot (2\mu \mathbf{D})$, is discretized using centered differences for the derivatives of $\mathbf{u}$ and an averaging of $\mu(\mathbf{x})$. For instance, the discretization of the first component of the divergence of the viscous stress tensor term is

$$\begin{aligned} &2(\mu u_x)_x + (\mu u_y)_y + (\mu v_x)_y \approx \\ &\quad \{((\mu_{i+1,j} + \mu_{i,j})(u_{i+1,j} - u_{i,j}) - (\mu_{i,j} + \mu_{i-1,j})(u_{i,j} - u_{i-1,j}))/h^2 \\ &\quad + ((\mu_{i,j+1} + \mu_{i,j})(u_{i,j+1} - u_{i,j}) - (\mu_{i,j} + \mu_{i,j-1})(u_{i,j} - u_{i,j-1}))/(2h^2) \\ &\quad + (\mu_{i,j+1}(v_{i+1,j+1} - v_{i-1,j+1}) - \mu_{i,j-1}(v_{i+1,j-1} - v_{i-1,j-1}))/(4h^2)\} \\ &\quad /(\rho_{i,j} Re) \end{aligned}$$

The discretization of the second term in the divergence of the viscous stress tensor $(\mu u_y)_x + (\mu v_x)_x + 2(\mu v_y)_y$ is defined similarly. Let $\mathbf{L}(\mathbf{u})$ be the 2nd order accurate discretized approximation to $\mathcal{L}(\mathbf{u})$. Call the components of $\mathbf{L} = (L^1, L^2)$. We apply linear or quadratic extrapolation boundary conditions on $\mathbf{L}$ to define $\mathbf{L}$ on grid points outside the domain (the ghost points).

3.2 Time Discretization

We discretize in time using the Method of Lines. This enables us to use a TVD Runge-Kutta scheme to advance the solution in time. These TVD R-K schemes were derived in [SO88]. The advantage of these schemes is their low storage requirements and non-oscillatory nature, with the trade off of higher cost per iteration due to repeated pressure solves. For example, if we consider solving

$$\mathbf{u}_t = \mathbf{L}(\mathbf{u})$$

First order Runge-Kutta is just the Explicit Euler method

$$\mathbf{u}^{n+1} = \mathbf{u}^n + dt \cdot \mathbf{L}(\mathbf{u}^n)$$

Second order Runge-Kutta is defined, using an intermediate solution $\mathbf{u}^1$, by

$$\begin{aligned} \mathbf{u}^1 &= \mathbf{u}^n + dt \cdot \mathbf{L}(\mathbf{u}^n) \\ \mathbf{u}^{n+1} &= .5 \cdot (\mathbf{u}^n + \mathbf{u}^1) + .5 \cdot dt \cdot \mathbf{L}(\mathbf{u}^1) \end{aligned}$$

Third order TVD Runge-Kutta uses two intermediate solutions, $\mathbf{u}^1$, and $\mathbf{u}^2$

$$\begin{aligned} \mathbf{u}^1 &= \mathbf{u}^n + dt \cdot \mathbf{L}(\mathbf{u}^n) \\ \mathbf{u}^2 &= .75 \cdot \mathbf{u}^n + .25 \cdot \mathbf{u}^1 + .25 \cdot dt \cdot \mathbf{L}(\mathbf{u}^1) \\ \mathbf{u}^{n+1} &= \frac{1}{3} \cdot \mathbf{u}^n + \frac{2}{3} \cdot \mathbf{u}^2 + \frac{2}{3} \cdot dt \cdot \mathbf{L}(\mathbf{u}^2) \end{aligned}$$

3.3 Discretization of Reinitialization

The reinitialization equations (2.25) are discretized using a non oscillatory method for the spatial derivative terms. We choose to rewrite (2.25) in a form that looks more like a convection equation

$$\begin{aligned} \phi_\tau + H_1(\phi)\phi_x + H_2(\phi)\phi_y - S(\phi_0) = 0 \\ \phi(\mathbf{x}, \tau = 0) = \phi^n(\mathbf{x}) \end{aligned} \tag{3.2}$$

with

$$\begin{aligned} H_1(\phi) &= S(\phi^0)\frac{\phi_x}{\sqrt{\phi_x^2 + \phi_y^2}} \\ H_2(\phi) &= S(\phi^0)\frac{\phi_y}{\sqrt{\phi_x^2 + \phi_y^2}} \end{aligned}$$

To compute these terms numerically, we define forward and backward finite differences of ϕ, $\phi_x^+ = D_+^x\phi$, and $\phi_x^- = D_-^x\phi$, and corresponding approximations to H_1, $H_1^+ = S(\phi)\phi_x^+$, and $H_1^- = S(\phi)\phi_x^-$. Then the following selection procedure determines the approximation of ϕ_x, and H_1.

$$\begin{cases} \phi_x \approx \phi_x^-, H_1 \approx H_1^- & H_1^+ \geq 0 \text{ and } H_1^- \geq 0, \\ \phi_x \approx \phi_x^+, H_1 \approx H_1^+ & H_1^+ \leq 0 \text{ and} H_1^- \leq 0, \\ \begin{cases} \phi_x \approx \phi_x^-, H_1 \approx H_1^- & (H_1^+ - H_1^-)/(\phi_x^+ - \phi_x^-) > 0, \\ \phi_x \approx \phi_x^+, H_1 \approx H_1^+ & \text{otherwise} \end{cases} & H_1^- \geq 0 \text{ and } H_1^+ \leq 0, \\ \phi_x \approx 0, H_1 \approx 0 & H_1^- < 0 < H_1^+ \end{cases}$$

A similar selection procedure is used to approximate ϕ_y and H_2. Once all terms are computed, we complete the approximation of H_1 and H_2 by dividing each by $\sqrt{\phi_x^2 + \phi_y^2}$. Higher order accurate approximations can be derived, but we omit the details. Once the spatial terms are discretized, we the same 2nd or 3rd order

TVD Runge-Kutta method for advancing in τ as we defined for the continuum equations.

There are a few modifications we have added to the reinitialization procedure [Fed]. The first is a limiting procedure to ensure that $sgn(\phi)$ does not change at any point in the domain.

Each step in the TVD Runge-Kutta procedure looks like

$$\phi^{new} = \phi^{old} + dt \cdot G$$

where ϕ^{old} is some linear combination of ϕ's at previous steps, dt is some multiple of the pseudo-time step $d\tau$, and $G = H_1\phi_x + H_2\phi_y - S$ is the approximation to the spatial derivative term. We wish to ensure that $sgn(\phi^{old} - dt \cdot G) = sgn(\phi^{old})$, equivalently, $\phi^{old} \cdot \phi^{old} \geq dt \cdot \phi^{old} \cdot G$. If this condition is not satisfied at any grid point in the domain, we replace $G_{i,j}$ with $\epsilon S_{i,j}$, where ϵ was defined as some multiple of the spatial grid size, and

$$S_{i,j} = \frac{\phi^0}{\sqrt{(\phi^0)^2 + 4\epsilon^2}}$$

is a smooth approximation to $sgn(\phi^0)$.

If one examines the reinitialization equation (2.25), it is easy to see that information is propagated away from the interface with increasing τ. As this 'wave' approaches the boundary of the domain, numerical difficulties usually occur resulting in non smooth regions of ϕ near the boundary.

The second modification is the imposition of different boundary conditions on ϕ. After the completion of the TVD Runge-Kutta pseudo-time stepping, we must compute the value of ϕ^{n+1} at the ghost points outside the domain. We use the Dirichlet information on the boundary and linearly extrapolate the interior values to the ghost points. For a rectangular domain, we define the boundary

value of ϕ^{n+1} to be

$$\phi^{n+1}(x_i, yL) = \phi^{n+1}_{i,jL} + sgn\left(\phi^{n+1}_{i,jL}\right) \cdot \left|\phi^{n+1}_{i,jL} - \phi^{n+1}_{i,jL+1}\right|$$

where (x_i, yL) is a point on the lower boundary, and jL is the index of the row of grid points adjacent to that boundary. This choice of boundary condition ensures a smooth extrapolation of the interior values to the ghost points.

3.4 Discretization of the Projection

The discretization of the projection is the most difficult part of the problem. First define the following coefficients

$$\begin{aligned} d_1 &= 2.0/(\rho(i,j) + \rho(i+1,j)) \\ d_2 &= 2.0/(\rho(i,j) + \rho(i,j+1)) \\ d_3 &= 2.0/(\rho(i,j) + \rho(i-1,j)) \\ d_4 &= 2.0/(\rho(i,j) + \rho(i,j-1)) \end{aligned}$$

We now discretize (2.22) in a straightforward manner.

$$\begin{aligned} (d_1 + d_2 + d_3 + d_4)P_{i,j} - d_1 P_{i+1,j} - d_2 P_{i,j+1} - d_3 P_{i-1,j} - d_4 P_{i,j-1} = \\ -h^2(L^1_{i+1,j} - L^1_{i-1,j} + L^2_{i,j+1} - L^2_{i,j-1})/(2h) \end{aligned} \tag{3.3}$$

This equation is valid if (i,j) is in the interior of the domain. However, if the stencil crosses the boundary of the domain, we modify (3.3) by using the boundary conditions. For instance, if $i = 1, 1 < j < ny$, then the Neumann boundary condition is

$$d_4(P_{0,j} - P_{1,j})/h = -(L^2_{0,j} + L^2_{1,j})/2$$

The equation(3.3) then becomes

$$\begin{aligned}(d_1+d_2+d_3)P_{i,j}-d_1P_{i+1,j}-d_2P_{i,j+1}-d_3P_{i-1,j}=\\ -h^2(L^1_{i+1,j}-L^1_{i-1,j}+L^2_{i,j+1}-L^2_{i,j-1})/(2h)+h(L^2_{0,j}+L^2_{1,j})/2\end{aligned} \tag{3.4}$$

The modifications are similar for those grid points near the other boundaries.

We now put these equations into matrix form. Order the unknown P starting at the lower left corner, $\{(1,1),(1,2),\ldots,(1,nx),(2,1),\ldots,(ny,nx)\}$. There are $N = nx * ny$ equations with N unknowns. Define the resulting matrix A, and the forcing b. We now need to solve $Ax = b$, where x is the vector of pressure unknowns ordered as above.

Just as in the continuous case, the linear system has a non empty kernel. In particular the vector $\mathbf{e} = (1,1,\ldots,1)^t$ is in the null space of A. The corresponding compatibility condition is that $(b,e) = 0$, where $(\cdot,\cdot)$ is the usual vector inner product. If b does not satisfy this condition, we can modify b so that it does, by replacing b with

$$b = b - \frac{(b,e)}{(e,e)} = b - \frac{(b,e)}{N} \tag{3.5}$$

This will cause a shift in the solution by a constant, which does no harm. In fact, this correction will cause the PCG method to converge to the least squares solution. We must also choose our initial guess x^0 so that $(x^0,e) = 0$.

The matrix A is symmetric and positive semi-definite, so we can hope to apply some sort of preconditioned conjugate gradient method to the system. This problem is addressed in [LR80] in great detail.

To summarize, it turns out that a good preconditioner for $Ax = b$ can be formed using the matrix resulting from discretizing

$$\Delta P = F$$
$$\frac{\partial P}{\partial \mathbf{n}} = g$$

We call the matrix corresponding to the above problem, C. C also has a one dimensional null space. In fact, $\mathbf{e}$ is also a null vector of C. If we now define the matrix $D^{1/2}$ to be the diagonal matrix with $d_{kk} = \sqrt{a_{kk}/c_{kk}}$, then our preconditioner matrix $M = D^{1/2}CD^{1/2}$. In the preconditioned conjugate gradient (PCG), we solve $Mz = r$ repeatedly. This is easily done by

- compute $w = D^{-1/2}r$
- solve $Cu = w$ (using BLKTRI from SLATEC)
- compute $z = D^{-1/2}u$

We can simplify this process considerably by replacing the original system $Ax = b$ by

$$\hat{A}\hat{x} = \hat{b} \tag{3.6}$$

$$\hat{A} = D^{-1/2}AD^{-1/2} \tag{3.7}$$

$$\hat{b} = D^{-1/2}b \tag{3.8}$$

We solve $\hat{A}\hat{x} = \hat{b}$, using PCG with C as our preconditioner, then compute $x = D^{-1/2}\hat{x}$ to complete the solution procedure.

The remaining issue in this method is that we must force each iterate to be orthogonal to the null space of $\hat{A}$. An examination of the PCG method reveals that the direction vector is updated by taking a multiple of the old direction plus the solution to the preconditioner problem. So if we force the solution of $C\hat{z} = \hat{r}$ to be orthogonal to $\hat{\mathbf{e}} = D^{-1/2}\mathbf{e}$, by explicitly applying (3.5) to the update of the direction vectors in the PCG method, each iterate will remain orthogonal to $\hat{\mathbf{e}}$. The general PCG routine from SLATEC can be modified without much trouble to do this.

We can include the surface tension term into the forcing for (2.22) explicitly, by adding (2.24) to the forcing term computed above in (3.3).

3.5 CFL Condition

We compute the time step using [Sus94]. In addition to the usual time step restrictions for velocity and the viscous term, Δt_c, and Δt_v respectively, we need a restriction to ensure that surface tension driven capillary waves on the interface don't travel more than one computational cell per time step. The Δt_s term ensures this.

$$\Delta t_s = \sqrt{\frac{(1+\rho_b/\rho_e)We}{8\pi}} h^{1.5}$$

$$\Delta t_v = \frac{3}{14} min\left(\frac{\rho Re \cdot h^2}{\mu}\right)$$

$$\Delta t_c = min\left(\frac{h}{|\mathbf{u}|}\right)$$

$$\Delta t^{n+1} = .5 min(\Delta t_s, \Delta t_v, \Delta t_c)$$

CHAPTER 4

Immersed Interface Method

In his dissertation [Li94], Zhilin Li developed a method to solve elliptic equations of the form

$$\nabla \cdot (\beta \nabla u) = f \tag{4.1}$$

$$[u]_\Gamma = w \tag{4.2}$$

$$\left[\beta \frac{\partial u}{\partial n}\right]_\Gamma = v \tag{4.3}$$

$$\tag{4.4}$$

$[\cdot]_\Gamma$ denotes the jump in the quantity across the interface Γ, and $\beta(x, y)$ could be a discontinuous function. In his work at UCLA, Li developed a faster method [HLO97]. In either method, we need to be able to compute certain quantities about the interface: its curvature, unit tangent and normal, and derivatives of functions along the interface. In his original method, Li used a discrete set of points together with spline interpolation. However, one could use a level set to define the interface and also compute all the above quantities from the level set function. That is what we propose to do. In the Chapter 5 we have some preliminary results comparing the static elliptic solvers using both the smoothed interface and IIM method to solve the problems. Section 4.2 has an application of the IIM to a simple fluid flow problem

4.1 Level Set Implementation

The IIM is really a way to compute a modification to a finite difference scheme when the finite difference stencil crosses an interface. Using this method, one is able to retain second order accuracy in the inf-norm throughout the domain. We can label a stencil as "regular" if none of its 'arms' intersects the interface, and "irregular" otherwise. The IIM modifications need only be computed for irregular stencils. For each irregular stencil, we choose a point on the interface, usually the point on the interface closest to the stencil center. See Figure 8.7 for a sketch of an irregular stencil and its corresponding interface point.

There are several necessary interface quantities that are needed in an application of the IIM to an elliptic problem. First, we need a collection of points lying on the interface, usually computed by some interpolation from $\phi(x, y)$. At each of these points, we need the outward unit normal, local curvature, and the value of the jumps in the solution w, and v. It is important to note that all the geometric quantities needed in the IIM method are easily computed directly from the level set function with some combination of interpolation and finite differences.

Since we need to compute derivatives along the interface of w and v, it is essential that our collection of points lying on the interface are uniformly spaced. This can be accomplished in a number of ways. One could just compute many points on the interface and when computing derivatives, make sure not to use points that are too close together. We prefer to choose our points more carefully so that they are well spaced. This also reduces storage costs and search times, which are not insignificant in these problems.

4.2 Application to Vortex Sheet Problems

A simple application of the Immersed Interface Method is the numerical solution of a two dimensional vortex sheet problem. The problem formulation is derived in [HOS95]. We summarize. Consider a two dimensional domain, periodic in x. The fluid velocity $\mathbf{u}$ is incompressible with its vorticity confined to a one dimensional subset of the domain defined by the zero level of a smooth function $\phi(x, y, t)$. The stream function $\Psi(x, y, t)$ satisfies

$$\begin{aligned} \Delta\Psi &= -\delta(\phi) \\ \Psi(x, \pm y, t) &= 0 \\ \Psi(x, y, t) &\quad \text{periodic in } x \\ \Psi(x, y, 0) &= y + .05\sin(2\pi x) \end{aligned} \tag{4.5}$$

The velocity is then $\mathbf{u} = (-\Psi_y, \Psi_x)$. The level set function is advected according to $\phi_t + \mathbf{u} \cdot \nabla\phi = 0$. The vortex sheet strength is $1/|\nabla\phi|$. This problem can be easily cast in the form of the Immersed Interface Method by noting that

$$\begin{aligned} [\Psi]_\Gamma &= 0 \\ \left[\frac{\partial\Psi}{\partial n}\right]_\Gamma &= -\frac{1}{|\nabla\phi|} \end{aligned} \tag{4.6}$$

The IIM solution of the linear system arising from the discretization of (4.5) with (4.6) requires a modification of the (otherwise zero) right hand side at irregular points to accommodate (4.6). The solution of the system is then computed by one call of a fast elliptic solver.

Some care must be taken computing $(-\Psi_y, \Psi_x)$ because Ψ has a jump in normal derivative across the interface. As detailed in [Li94], one can derive a finite difference approximation that is second order accurate away from the

interface and first order accurate near the interface by using the known values of the jumps in Ψ and $\frac{\partial \Psi}{\partial n}$.

The equation for the advection of ϕ is discretized, as before, using a TVD Runge-Kutta method in time, and a 2nd or 3rd order ENO scheme for the convection term. The time step dt must satisfy the standard CFL condition $dt < .5 * h/|\mathbf{u}|$, where h is the uniform spatial grid spacing.

We show some numerical results using this method on the problem above. The domain is $\{(x,y)| -1 \leq x \leq 1, -1 \leq y \leq 1\}$. The number of grid points in each direction is 80. Our initial interface is a sinusoidal shape. The vortex sheet strength has a non dimensional value of 1.

Figure 8.8 shows the evolution of the vortex sheet over time. We see that over time the vortex sheet rolls up as it should. At the given grid resolution, we were not able to integrate much further in time beyond the time of the final subplot in the figure. The velocity at the center of the domain grows quite large forcing a correspondingly small time step.

Figure 8.9 shows how the Immersed Interface Method preserves the jump in the velocity. We can modify the code slightly to use the smoothed interface procedure instead of IIM. When that is done, it is quite obvious how the interface is smeared and the jump in velocity then becomes a smooth transition.

There is a trade off between the IIM and the smoothed interface methods. The smoothed interface methods are quite easy to program, while writing an IIM solver is quite complicated. There is a significant difference in the quality of solution produced by each method, the IIM method will carefully preserve discrete jumps in the solution. Since the IIM method will preserve jumps, finite difference methods must be more carefully implemented. Care must be taken when differentiating numerically across an interface. Fortunately for the vortex

sheet problem, we can use the knowledge of the value of the jump in velocity to make differentiating across the interface first order accurate.

CHAPTER 5

Tests and Numerical Results

5.1 Linear Solver Tests

We have run various tests solving elliptic problems like (4.1). We choose an exact solution $P(x,y) = cos(x)cos(y)$, and the density function

$$\rho(x,y) = \left(1 - \frac{\rho_b}{\rho_e}\right) H_\epsilon(\phi) + \frac{\rho_b}{\rho_e}$$

We choose $\phi(x,y) = \sqrt{(x-.5)^2 + (y-.5)^2} - .3$, and $\epsilon = 1.5h$. We ran two tests comparing three methods: Diagonally Scaled Incomplete Cholesky preconditioning, the inverse Laplacian preconditioner described in Section 3.4, and the fast IIM method described in [HLO97]. We test these methods with density ratios of .1 and .001. The results are presented in Tables 5.1 and 5.2.

T is time for the run measured in seconds, err is the $l2$ error once the error is shifted to have zero average. For the IIM method results, M is the number of discrete points on the interface. The rate of convergence is listed at the bottom of each column.

For the PCG methods, we see that the number of iterations to convergence does depend on the density jump across the interface. Hopefully, when this solver is embedded in the Navier-Stokes solver, using the previous solution as the initial guess for the next solution will help to reduce the number of iterations to convergence. Preliminary results seem to bear this out. We should only

Table 5.1: Convergence Study: Linear Solver, density ratio=.1

	DSICCG			Inverse Laplacian			IIM			
N	its	err	T	its	err	T	M	its	err	T
10	19	.104		22	.104		8	8	.0018	
20	31	.052		35	.052		24	11	.0004	
30	45	.0335		36	.033		32	11	.0002	
40	59	.025	8	46	.025	8	48	10	.0001	8
50	73	.020	12	51	.020	12	56	11	6.45e-5	11
80	116	.012	43	68	.012	35	96	10	2.65e-5	19
	1st Order			1st Order			2nd Order			

expect first order convergence in the PCG methods since the density function is smoothed across the interface.

The IIM method shows excellent convergence properties and has the additional feature that the number of iterations and the time to convergence is independent of the density jump. It is this property that we hope to exploit in a Navier-Stokes solver.

5.2 Surface Tension Test

A good check for the validity of the surface tension term is to consider a circular bubble in the absence of gravity and viscous effects. The jump in pressure is then given by Laplace's formula, equation in [LL87, 61.3]

$$P_e - P_b = -\frac{\kappa}{We} \tag{5.1}$$

the minus sign is needed because $P_b > P_e$ if the bubble is concave, and $\kappa = 1/r$ where r is the radius of the bubble. For the test, we choose $We = .1$, $\rho_b = .001$,

Table 5.2: Convergence Study: Linear Solver, density ratio=.001

	DSICCG			Inverse Laplacian			IIM			
N	its	err	T	its	err	T	M	its	err	T
10	19	12.1		32	12.1		8	8	.0015	
20	32	5.71		52	5.71		24	11	.00034	
30	46	3.65		76	3.65		32	12	.00016	
40	60	2.71	8	74	2.71	13	48	11	9.35e-5	8
50	75	2.14	11	100	2.14	22	56	12	5.85e-5	11
80	120	1.32	44	168	1.32	84	96	10	2.44e-5	20
	1st Order			1st Order			2nd Order			

$\rho_e = 1.0$, $\Omega = [-1,1] \times [0,2]$, $r = .5$. The center of the bubble is $(0,1)$, and we choose $\epsilon = 1.5 * h$. We ran this test with $nx = ny = 20$ and $nx = ny = 40$. The theoretical jump in pressure with this choice of parameters according to (5.1) is $[P_{exact}] = 20$. For $nx = ny = 20$, $[P] \approx 21.5$ and when $nx = ny = 40$, $[P] \approx 20.8$, verifying our formulation of the surface tension term.

5.3 Advection Test

We would also like a check of the time stepping - projection combination. We solve

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla P + \frac{\nu}{\rho}\Delta \mathbf{u} + \mathbf{F} \tag{5.2}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{5.3}$$

$$\mathbf{u} = \mathbf{g} \tag{5.4}$$

in the region $\Omega = [0,\pi] \times [0,\pi]$. As in [HK], we define $\nu = .01$, choose an exact solution and compute the forcing $\mathbf{F}(x,y,t)$, and boundary conditions $\mathbf{g}(x,y,t)$

from that solution. Let

$$\mathbf{u}_{exact}(x,y,t) = \begin{pmatrix} \sin^2(x)\sin(2y)\cos(2\pi t) \\ -\sin(2x)\sin^2(y)\cos(2\pi t) \end{pmatrix} \tag{5.5}$$

$$P_{exact}(x,y,t) = \sin(x)\sin(y)\cos(2\pi t) \tag{5.6}$$

$$\rho_{exact}(x,y,t) = 4 + x + y \tag{5.7}$$

We discretize this problem as outlined in Section 3. The errors reported below are computed at $t = .5$.

Table 5.3: Errors in Advection Test

N	$\lvert\mathbf{u} - \mathbf{u}_{exact}\rvert_2$	$\lvert u - uExact\rvert_\infty$	$\lvert v - v_{exact}\rvert_\infty$
50	.027	.0146	.0146
60	.0152	.0081	.0081
80	.0068	.0060	.0061
Order	2.93	1.82	1.82

We see second order convergence in the velocity as expected.

5.4 Comparison to Previous Method

5.4.1 Two Dimensional Flows

We have run various test cases of our method in 2-d Cartesian geometry. We would like to present some comparisons of the method outlined in Section 3 to the previous results in [Sus94]. We look at two rising bubble problems.

The first rising bubble simulation, Figure 8.3, has a very small surface tension and medium range Reynolds number (Re=100, We=200) with a density ratio of 1000:1. We see in the comparison, that the bubble with no surface tension deforms

rapidly while the bubble with surface tension stays more circular, verifying the effect that surface tension has on rising bubbles.

The second rising bubble problem Figure 8.4 has stronger surface tension (Re=168, We=45), density ratio =40:1. After more time steps than the previous example, the effects of surface tension are clear. While the bubble with no surface tension has broken into two smaller bubbles, the bubble with strong surface tension is still intact and has undergone only mild deformation.

5.4.2 Axisymmetric Flows

We can also solve these flow problems in axisymmetric coordinates. The derivation of the equations is presented in Appendix B. We examine two test cases for axisymmetric flow, a bubble rising in a viscous liquid, and a droplet falling into a liquid pool. In both problems we discretize on a 40x80 grid.

For the rising bubble problem, our non dimensional parameters are Re=10, We=100.0, Fr=.1, Figure 8.10 shows the evolution of the bubble. One can again see the effect of weak surface tension in the evolution by the large deformation of the bubble from circular.

The falling droplet problem has parameters Re=100, We=10, Fr=.1, the solution is plotted in Figure 8.11. This problem clearly illustrates the ease with which the level set methodology solves problems with merging interfaces. Nothing special was done to simulate the droplet merging with the liquid. Again, the presence of surface tension forces keep the falling droplet basically circular before impact with the surface of the liquid.

CHAPTER 6

Application to Micro-gravity Flows

One new application of this method we have investigated is the problem of simulating flows in micro-gravity. The problem origin is the simulation of vapor-liquid interactions in low gravity. Consider a heat exchanger unit on a satellite utilizing a liquid coolant. In a 1-g environment, buoyancy will cause vapor bubbles to rise away from the heat source ensuring a steady flow of heat from the source to the fluid. However, in low gravity environments if the heat exchanger is not properly designed, a layer of insulating vapor can develop between the heat source and the cooling fluid due to the lack of buoyancy. If this occurs, the flow rate of heat from the source is dramatically reduced. This can result in damage to the part being cooled if it sensitive to high heat.

We study this phenomenon experimentally and numerically in [MMR97]. The experimental setup is a perforated metal plate submerged in liquid nitrogen. A heat source is applied beneath one of the perforations causing a bubble of vapor to form inside the hole. As more heat is applied, the bubble expands and eventually separates from the plate and rises. We would like to model this situation using our level set method derived here and see agreement between our computer simulations and the results from the lab.

Our simplified numerical model problem is the injection of gaseous nitrogen into a rectangular box containing liquid nitrogen under very low gravity conditions. We choose different boundary conditions to simulate the experimental

problem. On the lower boundary of the domain, we specify a region called the inlet where we fix a vertical velocity and also maintain $\phi < 0$ corresponding to the vapor phase. On the top boundary of the domain, we specify an outlet boundary condition so bubbles rising to the top exit the domain. On each side of the domain, we specify either no slip or free slip boundary conditions depending on the solution method.

We first applied the method of [Sus94] to this problem. This required a few modifications. First, we must define a purely vertical velocity at the injection site. This makes the determination of the dimensionless constants, the Reynolds and Weber numbers, more straightforward. Second, and more importantly, the no slip boundary conditions no longer apply, we instead impose free slip boundary conditions.

Finally, the specification of the level set function is different. Initially, the box is entirely one fluid (liquid Nitrogen) and the other phase (Nitrogen vapor) is carried in through the inlet. We can accomplish this by setting the level set function to be zero on the inlet, the positive distance from the inlet in the interior of the box, and negative in the ghost points outside the box, below the inlet. During the time evolution, we force the ghost points beneath the inlet to have negative value explicitly.

In all our investigations, we choose to specify an inlet port on the lower side of the box with a purely vertical velocity profile across this inlet. Because of incompressibility, we need to specify an outlet port and outlet velocity so that

$$\int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} = 0 \tag{6.1}$$

In all our tests, we put the inlet on the bottom of the box and the outlet on top,

so the integral condition becomes

$$-\int_{inlet} v dx = \int_{outlet} v dx \tag{6.2}$$

Where v is the vertical component of velocity specified on the inlet or outlet.

In [Sus94], the Navier-Stokes equations are expressed as $\mathbf{u}_t = \mathbf{L}(\mathbf{u})$ where $\mathbf{L}(\mathbf{u})$ contains the convective, viscous, gravity, and surface tension terms. The incompressibility is enforced by finding the divergence free part of $\mathbf{L}$ at each time step [CM93]. This can done by solving

$$\nabla(\rho \nabla \Psi) = -\nabla \times \rho L \tag{6.3}$$

Where Ψ is the standard stream function. The free slip boundary conditions correspond to $\Psi = 0$. Since $d\Psi/dx = -v$ and $d\Psi/dy = u$, we can construct boundary conditions on Ψ for our injection problem as follows.

For simplicity, let our box $\Omega = [0,1] \times [0,1]$. $\Psi(0,0) = 0$, then

$$\begin{aligned}
\Psi(x,0) &= -\int_0^x v(\xi,0) d\xi \\
\Psi(1,y) &= -\int_0^1 v(x,0) dx + \int_0^y u(1,\eta) d\eta \\
\Psi(x,1) &= -\int_0^1 v(x,0) dx + \int_0^1 u(1,y) dy - \int_x^1 v(\xi,1) d\xi \\
\Psi(y,0) &= -\int_0^1 v(x,0) dx + \int_0^1 u(1,y) dy - \int_0^1 v(x,1) dx + \int_y^1 u(0,\eta) d\eta
\end{aligned}$$

If (6.2) is satisfied, then this specification of boundary conditions on Ψ will be consistent, that is, $\lim_{y \to 0} \Psi(y,0) = 0$.

We have run numerous cases with these boundary conditions and include some plots of the injection process evolution. Figure 8.5 with Re=170, shows the time evolution of the injection process in the absence of surface tension. The inlet velocity was specified with a purely vertical, parabolic profile. Note the effect of

no surface tension: the trail of vapor that follows the bubble into the domain and at later times, there is no breaking of the interface.

In contrast, Figure 8.6 with parameters Re=110 and We=.27 shows how the presence of surface tension in the injection flow causes the pinch-off of the vapor. Following pinch off, the separated bubble retains the more round shape.

These numerical results qualitatively agree with the experimental results and give us some hope that this method can be applied to assist the design of a simple, effective low gravity heat exchanging device.

CHAPTER 7

Application to Core Annular Flows

An interesting physical problem which seems ideally suited to solution using level sets recently came to our attention. In the transport of heavy crude oil in pipelines, water is injected into the pipe and it naturally migrates to the walls of the pipe leaving the oil in the center. The water then acts as a lubricant. The pumping pressure then must balance the shear stress of the water on the wall of the pipe, not the shear stress of the oil, which would be much greater due its larger viscosity. Thus, significant savings can be realized. This process is reviewed in [JBC97]

It is then of interest to find parameter values where this core annular flow (CAF) is stable. The stable steady state is characterized by a wavy interface between the oil and water. Since the viscosity of the oil is much larger than that of water $\mu_{oil}/\mu_{water} \approx 10^5$, the fluid velocity in the oil is nearly a constant throughout. Previous numerical simulations [BKJ96] have iterated on the shape of the interface until force balance is reached. At each step of the iteration the fluid velocities, pressures and stresses must be computed. The procedure is very computationally expensive.

We propose to solve the problem in a periodic domain by starting with a reasonable interfacial shape and pressure jump, and solve the Navier-Stokes equations forward in time until steady state is reached. The interface will be the zero level set of a periodic function. There is nothing exotic about the equations

except for the drastic jump in viscosity across the interface.

We treat this problem as flow between infinite parallel plates. To further simplify the problem, we only consider flows which are symmetric about the center line of the pipe. We truncate the domain to a rectangular shape with the length in the direction of the flow twice the size as the other dimension. On the pipe wall we assign no slip boundary conditions, and on the line of symmetry (the center of the pipe) we impose symmetry boundary conditions. We assume periodicity in the variable parallel to the flow. While this two dimensional problem is not ideal, the actual flows are fully three dimensional and we are not yet capable of solving such flows. If we modify (2.19) for CAF,

$$\begin{aligned} \mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} &= \mathbf{F} + \frac{1}{\rho}(-\nabla P + \frac{1}{Re}\nabla\cdot(2\mu\mathbf{D}) + \frac{1}{We}\kappa\delta(d)\mathbf{n}) \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned} \tag{7.1}$$

where $\mathbf{F}$ contains both the gravity and imposed pressure gradient. We rewrite (7.1)

$$\begin{aligned} \mathbf{u}_t &= \mathcal{L}(\mathbf{u}) - \frac{1}{\rho}\nabla P + \frac{1}{\rho Re}\nabla\cdot(2\mu\mathbf{D}) \\ \nabla\cdot\mathbf{u} &= 0 \end{aligned} \tag{7.2}$$

The major computational difficulty we encountered is the stiffness due to the large difference in viscosity between the fluids. We first implemented a simple splitting method that has shown some success. An implicit treatment of the viscous terms has also been implemented. These methods are described below.

7.1 Splitting Method

We discretize in time using explicit Euler, leaving out the viscous term in the projection method

$$\begin{aligned} \frac{\mathbf{u}^*_{i,j} - \mathbf{u}^n_{i,j}}{dt} &= \mathbf{L}(\mathbf{u}^n_{i,j}) \\ \frac{\mathbf{u}^{n+1}_{i,j} - \mathbf{u}^*_{i,j}}{dt} &= -\frac{1}{\rho}\nabla P \end{aligned} \tag{7.3}$$

After taking one time step of the above method, we take $\mathbf{u}^{n+1}$ as initial data for

$$\mathbf{u}_t = \frac{1}{\rho}\nabla \cdot (2\mu \mathbf{D}) \tag{7.4}$$

and advance this equation using explicit Euler from t^n to t^{n+1}. Essentially, we take one large time step of (7.3), then many small time steps of (7.4). This turns out to be much faster than the non-split explicit method, since using an ENO method to compute the convective derivative and computing the projection are computationally expensive, while solving the heat equation is relatively fast.

Using this method allows us to compute the CAF flow with viscosity ratios of 100:1 in a reasonable amount of time. Figures 8.12, are 8.13 are plots of time evolution of this method for horizontal flows. These flows were computed in a non symmetric domain, The parameters for this example are Re=450, We=2000, Fr=.001, with a non dimensional pressure jump of 10. These plots show the formation of the wavy interface similar to those seen in experiments and shown in analytic solutions. However, the very weak surface tension allows the interesting behavior seen on the top interface.

7.2 Implicit Method

Although the splitting method showed some success, it is too slow to be used for larger jumps in viscosity. A fully implicit method was then developed to solve

this problem. We again restrict ourselves to first order accuracy and discretize as follows. We first solve for $\mathbf{u}^*$ in

$$\frac{\mathbf{u}^*_{i,j} - \mathbf{u}^n_{i,j}}{dt} = \mathbf{L}(\mathbf{u}^n_{i,j}) + \frac{1}{\rho Re}\nabla \cdot (2\mu \mathbf{D}(\mathbf{u}^*)) \tag{7.5}$$

Then we compute $\mathbf{u}^{n+1}$ from

$$\frac{\mathbf{u}^{n+1}_{i,j} - \mathbf{u}^*_{i,j}}{dt} = -\frac{1}{\rho}\nabla P$$

Using the pressure Poisson equation method. Equation (7.5) is a coupled system of equations because of the presence of the $\mu(x, y)$ term. We discretize (7.5) using centered differences. The resulting linear system is non-symmetric, but positive definite. A preconditioned GMRES routine is used to solve the system.

We have applied this method to a vertical core flow. Our parameters for this problem are Re=450, We=200, Fr=.1, and the imposed pressure jump which forces the flow vertically, has a non dimensional value of 100. Figures 8.14, 8.15, 8.16, and 8.17 show the numerical solutions. The number of grid points in each dimension 40. The density is constant across the interface, as it is in the physical problem, and the viscosity in the core is 1000 times higher than that in the lubricating layer. The implicit method shows better qualitative results, the waves in the flow have shapes very similar to those seen experimentally, and computed in [BKJ96].

CHAPTER 8

Figures for the Dissertation

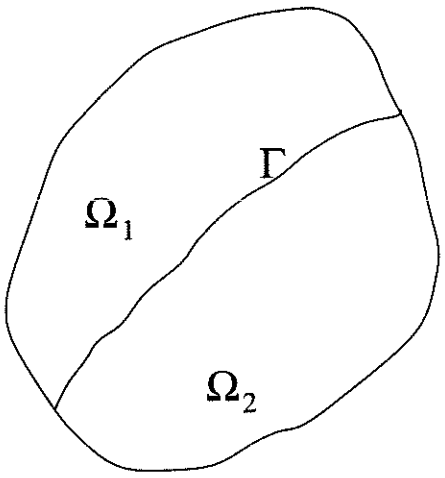

Figure 8.1: Differential volume element straddling interface.

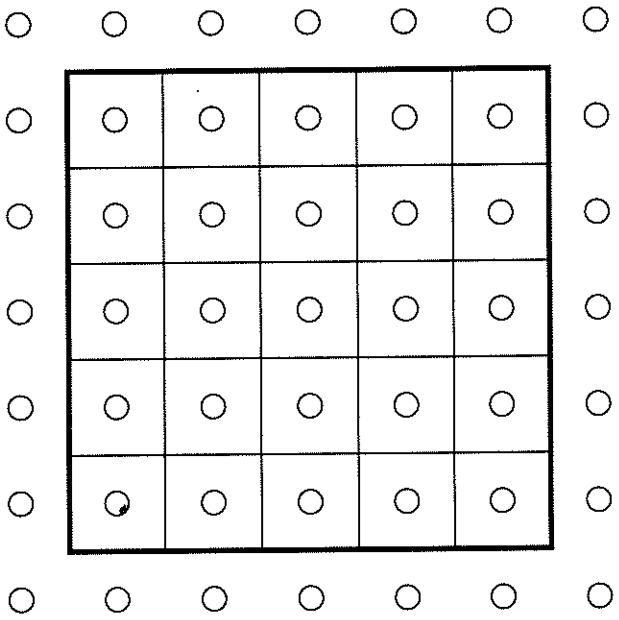

Figure 8.2: Computational grid with one layer of Ghost Points.

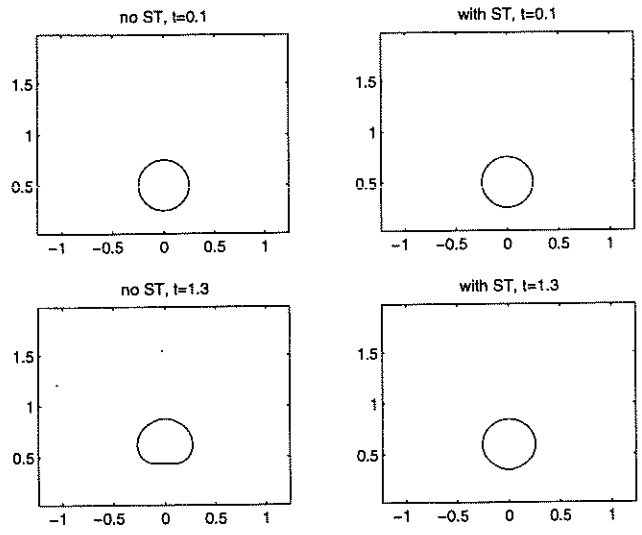

Figure 8.3: Rising bubble with small surface tension.

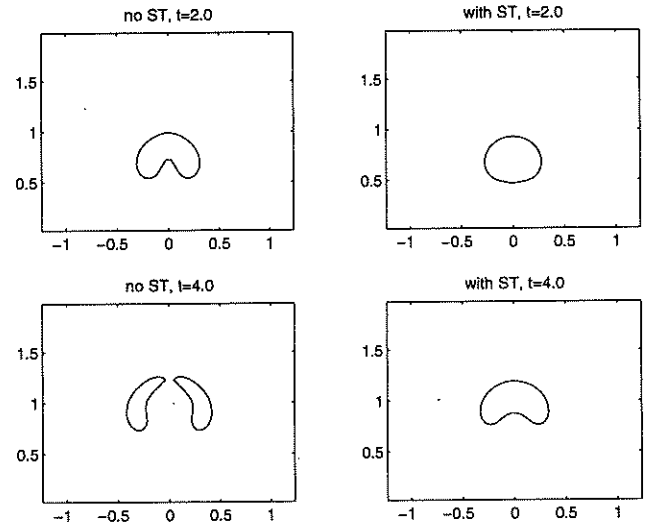

Figure 8.4: Rising bubble with large surface tension.

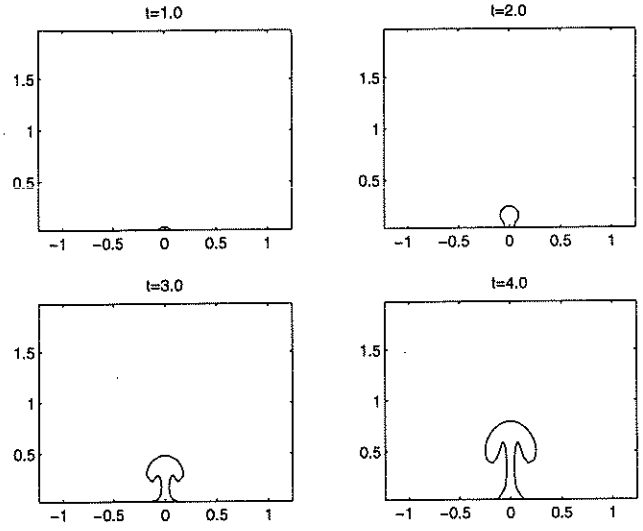

Figure 8.5: Injection with no surface tension.

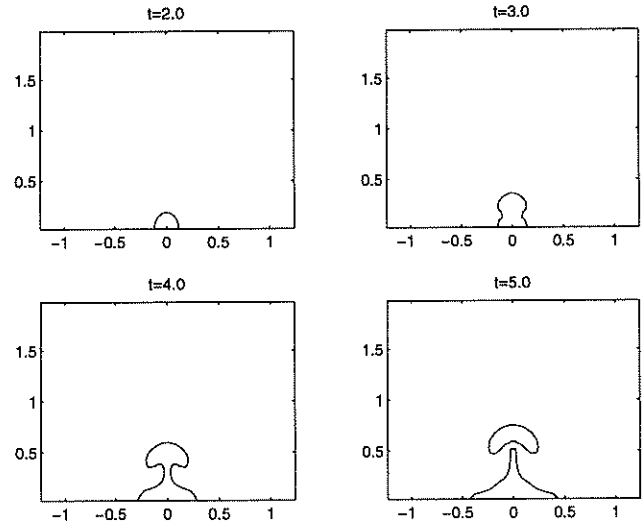

Figure 8.6: Injection with surface tension.

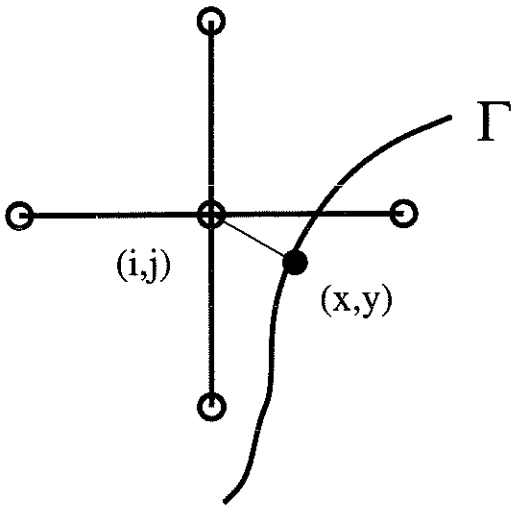

Figure 8.7: irregular grid point

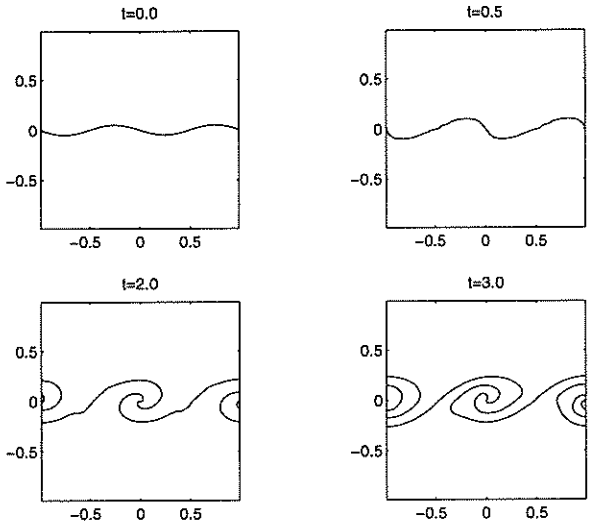

Figure 8.8: IIM vortex sheet problem

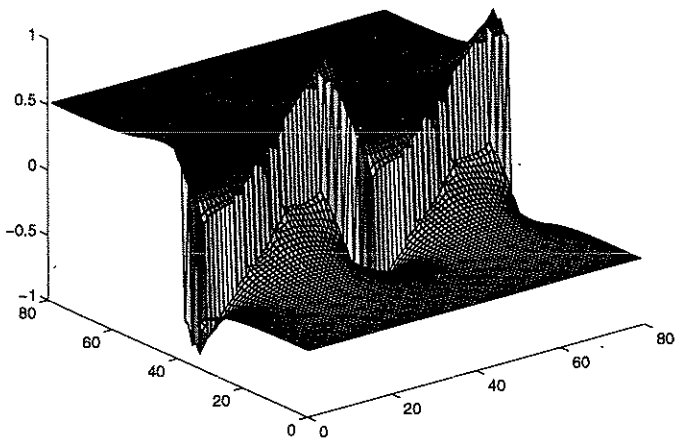

Figure 8.9: u velocity at t=0.5

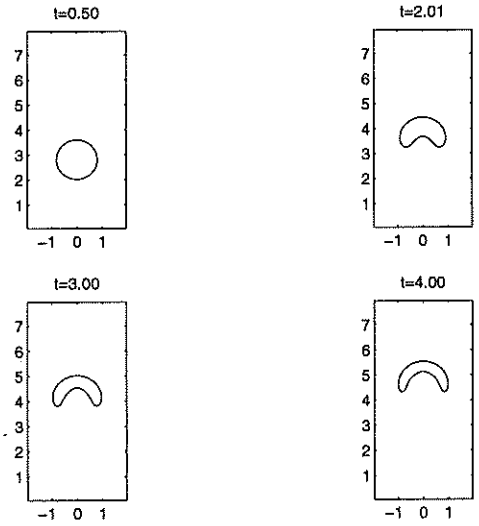

Figure 8.10: Rising axisymmetric bubble

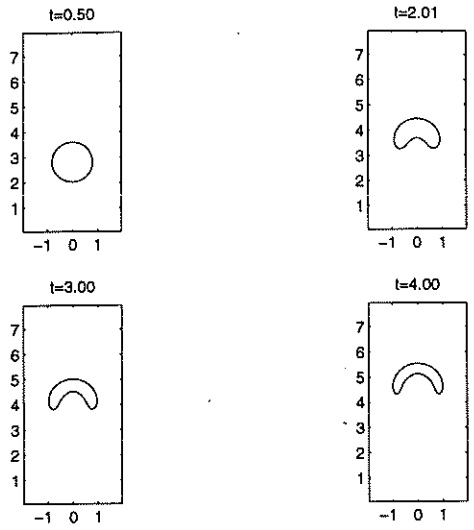

Figure 8.11: Falling droplet evolution

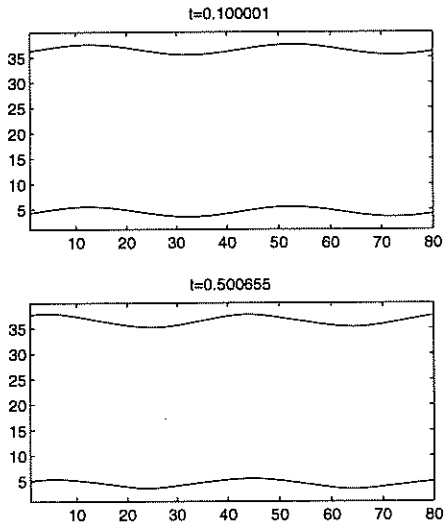

Figure 8.12: Initial data for CAF flow

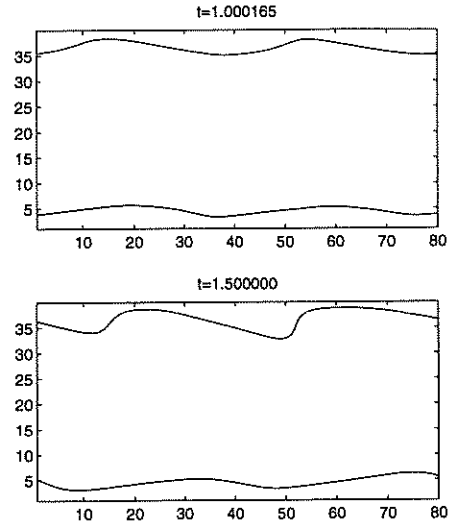

Figure 8.13: CAF flow at t=1.0

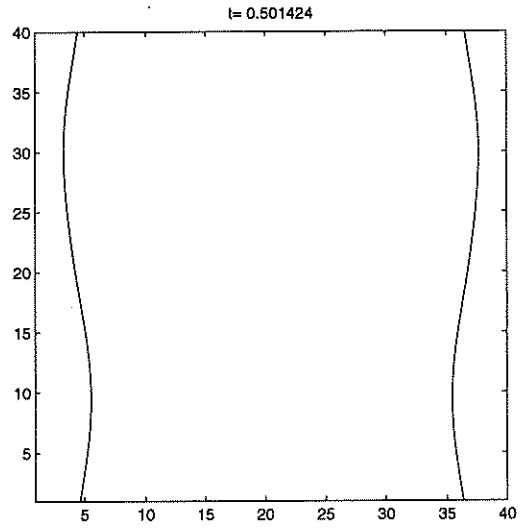

Figure 8.14: implicit-CAF flow at t=0.5

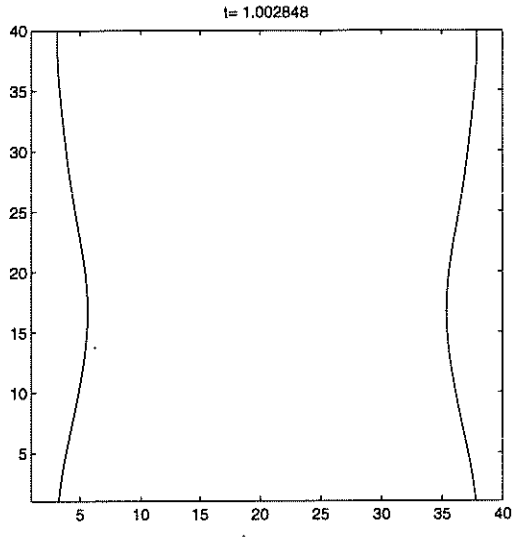

Figure 8.15: implicit-CAF upflow at t=1.0

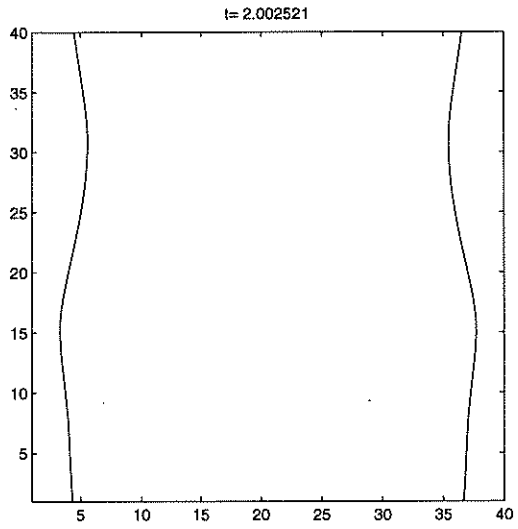

Figure 8.16: implicit-CAF upflow at t=2.0

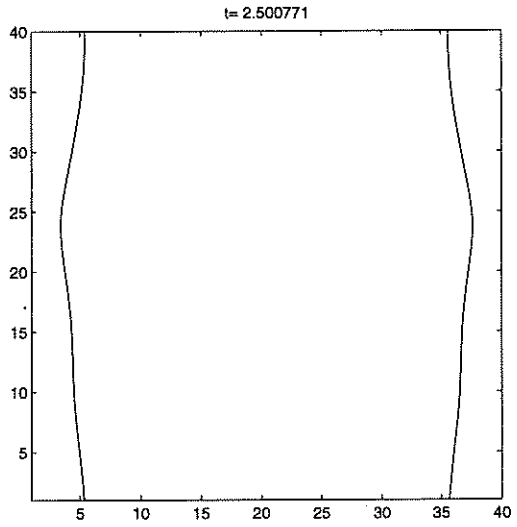

Figure 8.17: implicit-CAF upflow at t=2.5

Appendices to the Dissertation

APPENDIX A

Level Set Facts

The following facts can be used to simplify the divergence of the surface tension term above.

Outward unit normal to interface

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \tag{A.1}$$

Normal derivatives of ϕ

$$\frac{\partial\phi}{\partial n} = \nabla\phi \cdot \mathbf{n} = |\nabla\phi| \tag{A.2}$$

$$\frac{\partial^2\phi}{\partial n^2} = \left(\frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla\right)(|\nabla\phi|) \tag{A.3}$$

Curvature

$$\kappa = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right) \tag{A.4}$$

Combination

$$\Delta\phi = |\nabla\phi|\kappa + \frac{\partial^2\phi}{\partial n^2} \tag{A.5}$$

A useful fact pointed out by Dr. Barry Merriman, is most easily seen by regarding the interface as locally circular. This formula is only true in two dimensions, but a formula valid in three dimensions does exist.

$$\nabla\kappa \cdot \mathbf{n} = -\kappa^2 \tag{A.6}$$

APPENDIX B

Axisymmetric Flow Equations

Solving these problems in cylindrical coordinate geometry requires some modification of the equations of motion and the form of the projection.

By axisymmetric flow, we mean flows in which there is no flow in the θ direction, and all θ derivatives are identically zero. So there are only two variables, r-the radial direction, and z. We define the fluid velocity by the vector $\mathbf{u} = (u, w)$ where $u(r, z)$ is the radial component of velocity and $w(r, z)$ is the component in the z direction. In what follows, we denote differentiation by r with a subscripted r and differentiation by z with a subscripted z. The incompressibility condition $\nabla \cdot \mathbf{u} = 0$ translates to

$$\frac{1}{r}(ru)_r + w_z = 0 \tag{B.1}$$

The convective derivative $\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u}$ is written:

$$u_t + uu_r + wu_z$$

$$w_t + uw_r + ww_z$$

The divergence of the stress tensor $\sigma = -P\mathbf{I} + 2\mu\mathbf{D}$ is transformed by computing the various elements of $\mathbf{D}$ and applying the divergence to the result. The

deformation tensor $\mathbf{D}$ has the following elements

$$\mathbf{D}_{rr} = u_r$$
$$\mathbf{D}_{rz} = \mathbf{D}_{zr} = \frac{1}{2}(u_z + w_r)$$
$$\mathbf{D}_{zz} = w_{zz}$$

The first component of $(\nabla \cdot \sigma)/\rho$ is

$$-\frac{1}{\rho}\frac{\partial P}{\partial r} + \frac{1}{\rho}\left\{2(\mu u_r)_r + (\mu(w_r + u_z))_z + \frac{2\mu}{r}u_r - \frac{2\mu}{r^2}u\right\}$$

which simplifies to

$$-\frac{1}{\rho}\frac{\partial P}{\partial r} + \frac{1}{\rho}\left\{2(\mu u_r)_r + (\mu(w_r + u_z))_z + 2\mu\left(\frac{u}{r}\right)_r\right\}$$

The second component is

$$-\frac{1}{\rho}\frac{\partial P}{\partial z} + \frac{1}{\rho}\left\{\frac{1}{r}(r\mu(u_z + w_r))_r + 2(\mu w_z)_z\right\}$$

The Poisson equation we solve to compute the pressure (2.22), becomes

$$\frac{1}{r}\left(\frac{r}{\rho}P_r\right)_r + \left(\frac{1}{\rho}P_z\right)_z = \nabla \cdot \mathcal{L}(\mathbf{u}) \tag{B.2}$$

To make the equation symmetric, we multiply by r to get

$$\left(\frac{r}{\rho}P_r\right)_r + \left(\frac{r}{\rho}P_z\right)_z = r\nabla \cdot \mathcal{L}(\mathbf{u}) \tag{B.3}$$

We can solve B.2 using a GMRES solver, or B.3 using a preconditioned conjugate gradient method similar to that in 3.4. All the examples here have been computed using GMRES on B.2.

References

[BKJ96] R. Bai, K. Kelkar, and D.D. Joseph. "Direct simulation of interfacial waves in a high viscosity ratio and axisymmetric core annular flow." *Journal of Fluid Mechanics*, **327**:1–34, 1996.

[BKZ] J. U. Brackbill, D. B. Kothe, and C. Zemach. "A Continuum Method for Modeling Surface Tension." Theoretical Division, Los Alamos National Laboratory.

[CHM96] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. "A Level Set Formulation of Eulerian Interface Capturing Methods for Incompressible Fluid Flows." *Journal of Computational Physics*, **124**:449–464, 1996.

[CM93] Alxandre J. Chorin and Jerrold E. Marsden. *A Mathematical Introduction to Fluid Mechanics.* Springer-Verlag, 1993.

[Fed] Dr. Ron Fedkiw. "Personal Communication.".

[HK] William D. Henshaw and Heinz-Otto Kreiss. "Analysis of a Difference Approximation for the Incompressible Navier-Stokes Equations." Unpublished.

[HLO97] Thomas Y. Hou, Zhilin Li, Stanley Osher, and Hongkai Zhou. "A Hybrid Method for Moving Interface Problems with Application to the Hele-Shaw Flow." Technical report, UCLA, 1997.

[HOS95] Eduard Harabetian, Stanley Osher, and Chi-Wang Shu. "An Eulerian Approach for Vortex Motion Using a Level Set Regularization Procedure." Technical report, UCLA, 1995. CAM Report 95-30.

[JBC97] D.D. Joseph, R. Bai, K.P. Chen, and Y.Y. Renardy. "Core Annular Flows." *Annual Review of Fluid Mechanics*, **29**:65–90, 1997.

[Li94] Zhilin Li. *The Immersed Interface Method - A Numerical Approach for Partial Differential Equations with Interfaces.* PhD thesis, University of Washington, 1994.

[LL87] L. D. Landau and E. M. Lifshitz. *Fluid Mechanics.* Pergamon Press, 1987.

[LR80] John Gregg Lewis and Ronald G. Rehm. "The Numerical Solution of a Nonseparable Elliptic Partial Differential Equation by Preconditioned

Conjugate Gradients." *Journal of Research of the National Bureau of Standards*, **85**:367–390, 1980.

[MMR97] B. Miller, B. Merriman, K. Rafique, S. Osher, N. Noushkam, A. Spletzer, and T.H.K. Frederking. "Vapor-Liquid interfacial dynamics and related liquid nitrogen boiling on perforated plates." In *Space Cryogenics Workshop, Eugene, OR, Aug. 4-5, 1997*, 1997. to appear.

[SO88] Chi-Wang Shu and Stanley Osher. "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes." *Journal of Computational Physics*, **77**:439, 1988.

[SO89] C. W. Shu and S. Osher. "Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes, II." *Journal of Computational Physics*, **83**:32–78, 1989.

[Sus94] Mark Sussman. *A Level Set Approach for Computing Solutions to Incompressible Two Phase Flows.* PhD thesis, University of California, Los Angeles, 1994.