Level Set Methods for Two-Fluid Flows

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

1 Introduction

There are a wide variety of numerical methods for computing solutions to incompressible two-fluid flow problems. A literature survey will not be given here. These notes present an overview on level set methods for incompressible two-fluid flows. The basic method presented in these notes was developed in the PhD dissertation of Mark Sussman [1] [2]. Several improvements were developed in [3]. The method was extended to axisymmetric flows and compared with experimental results and boundary integral calculations in [4].

2 Basic Ideas

To fix ideas we shall consider interface problems in two dimensions where the interfaces are closed curves, denoted \( \Gamma_k \), which can be written as

\[
\Gamma_k = \{ X_k(\alpha, t) | 0 \leq \alpha \leq 1 \},
\]

where \( X_k(\alpha, t) \) is a point on the \( k \)th interface. The entire interface is

\[
\Gamma = \bigcup_k \Gamma_k.
\]

The velocity of the \( k \)th interface is denoted \( \mathbf{v}_k \). To find the time evolution of the interface, we solve

\[
\frac{dX_k}{dt} = \mathbf{v}_k.
\]

The difficulty with solving this set of equations is that interfaces can either merge or break-up. This means that one has to identify topology changes and then re-parameterize the interface. This is a nontrivial task. The level

---

1Lecture notes from a short course given at INRIA
set approach avoids this difficulty. We introduce smooth function, \( \phi(x, t) \), so that
\[
\Gamma = \{ x \mid \phi(x, t) = 0 \}
\]
Therefore we see that the zero level set of \( \phi \) is the interface. We take \( \phi > 0 \) inside the region surrounded by the curve and \( \phi < 0 \) outside. Next we consider a vector field, \( \mathbf{u} = \mathbf{u}(x, t) \), which has the property
\[
\mathbf{v}_k = \mathbf{u}(x, t) \big|_{x = \Gamma_k}.
\] (1)
We call \( \mathbf{u} \) the extension of \( \mathbf{v}_k \) off the front. We let the time evolution of \( \phi \) be given by
\[
\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0.
\] (2)
In view of Eq. (1) it is clear that if \( \phi(x, t) \) satisfies Eq. (2) the zero level set of \( \phi(x, t) \) marks the location of the interface. The level set formulation is then an Eulerian description of the interface motion.

This means that the interface problem can be solved on a fixed grid by using a finite difference scheme to solve Eq. (2). The explicit location is not necessary to solve Eq. (2). In order to find the location of the interface a contour plotting routine is used to find the zero level set of \( \phi(x, t) \). In this way the front is “captured” instead of tracked.

One difficulty with numerical implementation of the level set formulation presented here is that under the evolution of Eq. (2), \( \phi(x, t) \) may develop large gradients which will be difficult to handle numerically. To alleviate this difficulty we shall consider \( \phi(x, t) \) to be the signed distance from the interface. This means that \( |\nabla \phi| = 1 \). We can initialize \( \phi \) to be a distance function; however, under the evolution of Eq. (2) it will not remain a distance function and will need to be reinitialized after each time step. We were able to devise a simple efficient method to achieve this goal, which is as follows: let \( \phi_0(x, t) \) denote a level set function that is not a distance function, we would like to construct a \( \phi(x, t) \) which has the same zero level set of \( \phi_0(x, t) \) but \( \phi(x, t) \) is a distance function. This is achieved by solving the following partial differential equation:
\[
\frac{\partial \phi}{\partial \tau} + \mathbf{w} \cdot \nabla \phi = \text{sgn}(\phi_0) \quad \text{where} \quad \mathbf{w} = \text{sgn}(\phi_0) \frac{\nabla \phi}{|\nabla \phi|}.
\] (3)
Here \( \tau \) is an artificial time and \( \text{sgn}(x) = 1 \) if \( x > 0 \), \( \text{sgn}(x) = -1 \) if \( x < 0 \), and \( \text{sgn}(0)=0 \). It is evident that Eq. (3) is a nonlinear hyperbolic equation whose
characteristics point outward, in the normal direction, from the interface with speed one. Thus the level set function will be reinitialized near the interface first. In most applications we only need \( \phi(\mathbf{x}, t) \) to be a distance function near the interface. This method was developed by Sussman, Smereka and Osher[2]. The numerical implementation was improved by Sussman & Fatemi [11] to reduce mass loss.

The level set approach was first implemented numerically by Osher and Sethian[12] for motion by mean curvature (they did not keep the level set function a distance function).

3 Navier-Stokes Equations for Two Fluids

To fix ideas we shall call one of the fluids a liquid and the other a gas. We assume that both fluids are governed by the incompressible Navier-Stokes equation; therefore,

\[
\frac{\rho}{\rho} \frac{Du}{Dt} = -\nabla p + 2\mu \nabla \cdot \mathbf{D} + \rho \mathbf{g}, \quad \nabla \cdot u = 0, \quad \mathbf{x} \in \text{the liquid},
\]

\[
\rho \frac{Du}{Dt} = -\nabla p + 2\mu \nabla \cdot \mathbf{D} + \rho g, \quad \nabla \cdot u = 0, \quad \mathbf{x} \in \text{the gas},
\]

where \( u \) is the velocity, \( p \) is the pressure, \( \rho \) is the density, and \( \mu \) is the viscosity of the fluid. The subscripts \( \ell \) and \( g \) denote the liquid and the gas phase, respectively. \( \nabla / \rho \) is the material derivative, \( \mathbf{D} \) is the rate of deformation tensor, and \( \mathbf{g} \) is the acceleration due to gravity. The boundary conditions at the interface, \( \Gamma \), between the phases are:

\[
(2\mu \nabla - 2\mu g \nabla) \cdot n = (p - p + \sigma \kappa) n \quad \text{and} \quad u = u_\ell, \quad x \in \Gamma,
\]

where \( n \) is the unit normal to the interface drawn outwards from the gas to the liquid, \( \kappa = \nabla \cdot n \) is the curvature of the interface, and \( \sigma \) is the coefficient of surface tension.

We will denote the domain containing the two fluids as \( \Omega \) and its boundary as \( \sigma \Omega \). Since the fluid cannot penetrate the boundary, we have

\[
\mathbf{u} \cdot n = 0 \quad \text{on} \quad \partial \Omega.
\]

Our main interest is the free boundary between the two fluids and not the boundary layer at the solid wall (\( \partial \Omega \)). To suppress the formation of vorticity
at the wall we use the free slip boundary condition
\[ \mathbf{n} \cdot \nabla \mathbf{u} = 0 \quad \text{on} \quad \partial \Omega. \] (7)

It is convenient to define the stress tensor
\[ \mathbf{\sigma} = -p\mathbf{I} + 2\mu\mathbf{D} \] (8)
where \( \mathbf{I} \) is the identity matrix. The equations of motion become
\[ \rho_l \frac{D\mathbf{u}_l}{Dt} = \nabla \cdot \mathbf{\sigma}_l + \rho_l \mathbf{g}, \quad \mathbf{x} \in \text{the liquid} \] (9)
\[ \rho_g \frac{D\mathbf{u}_g}{Dt} = \nabla \cdot \mathbf{\sigma}_g + \rho_g \mathbf{g}, \quad \mathbf{x} \in \text{the gas}, \] (10)
with the boundary condition
\[ (\mathbf{\sigma}_l - \mathbf{\sigma}_g) \cdot \mathbf{n} = \sigma_K \mathbf{n}. \] (11)

Let \( \Omega_l \) and \( \Omega_g \) denote an arbitrary portion of the liquid phase and gas phase. Let \( \Omega_p = \Omega_l \cup \Omega_g \). The portion of gas-liquid boundary contained in \( \Omega_p \) is denoted as \( \Gamma_p \). Next we integrate (9) over \( \Omega_l \) and (10) over \( \Omega_g \) to obtain
\[ \rho_l \int_{\Omega_l} \frac{D\mathbf{u}_l}{Dt} d\mathbf{x} = \int_{\partial \Omega_l} \mathbf{\sigma}_l \cdot d\mathbf{S} + \rho_l \int_{\Omega_l} \mathbf{g} d\mathbf{x} \] (12)
\[ \rho_g \int_{\Omega_g} \frac{D\mathbf{u}_g}{Dt} d\mathbf{x} = \int_{\partial \Omega_g} \mathbf{\sigma}_g \cdot d\mathbf{S} + \rho_g \int_{\Omega_g} \mathbf{g} d\mathbf{x}, \] (13)
where we have used the divergence theorem. We make the following definition:
\[ \mathbf{\sigma} = \begin{cases} \mathbf{\sigma}_l & \text{if } \mathbf{x} \in \text{the liquid} \\ \mathbf{\sigma}_g & \text{if } \mathbf{x} \in \text{the gas}. \end{cases} \] (14)
Adding the above equations and using our definition of \( \mathbf{\sigma} \) we find
\[ \rho_l \int_{\Omega_l} \frac{D\mathbf{u}_l}{Dt} d\mathbf{x} + \rho_g \int_{\Omega_g} \frac{D\mathbf{u}_g}{Dt} d\mathbf{x} \]
\[ = \int_{\partial \Omega_p} \mathbf{\sigma} \cdot d\mathbf{S} - \int_{\Gamma_p} (\mathbf{\sigma}_l - \mathbf{\sigma}_g) \cdot \mathbf{n} dS + \rho_l \int_{\Omega_l} \mathbf{g} d\mathbf{x} + \rho_g \int_{\Omega_g} \mathbf{g} d\mathbf{x}. \] (15)
Next we use the divergence theorem to write
\[ \int_{\partial \Omega_p} \mathbf{\sigma} \cdot d\mathbf{S} = \int_{\Omega_p} \nabla \cdot \mathbf{\sigma} d\mathbf{x}. \] (16)
This should be considered in the sense of distributions since $\sigma$ is not continuous at the gas-liquid interface. Combining, (11), (16), (15) we obtain

$$
\rho_l \int_{\Omega_l} \frac{Du_l}{Dt} \, dx + \rho_g \int_{\Omega_g} \frac{Du_g}{Dt} \, dx \\
= \int_{\Omega_p} \nabla \cdot \sigma \, dx - \int_{\Gamma_p} \sigma \kappa n \, dS + \rho_l \int_{\Omega_l} g \, dx + \rho_g \int_{\Omega_g} g \, dx. 
$$

(17)

Now we make two more definitions

$$
\mathbf{u} = \begin{cases} 
\mathbf{u}_l & \mathbf{x} \in \text{the liquid} \\
\mathbf{u}_g & \mathbf{x} \in \text{the gas},
\end{cases}
$$

and

$$
\rho = \begin{cases} 
\rho_l & \mathbf{x} \in \text{the liquid} \\
\rho_g & \mathbf{x} \in \text{the gas}.
\end{cases}
$$

Since $\mathbf{u}_l = \mathbf{u}_g$ at the boundary then $\mathbf{u}$ is a continuous function. Using this fact and the definitions of $\mathbf{u}$ and $\rho$, we can write

$$
\rho_l \int_{\Omega_l} \frac{Du_l}{Dt} \, dx + \rho_g \int_{\Omega_g} \frac{Du_g}{Dt} \, dx = \int_{\Omega_p} \rho \frac{Du}{Dt} \, dx.
$$

(18)

Combining (17) and (18), we obtain

$$
\int_{\Omega_p} \rho \frac{Du}{Dt} \, dx = \int_{\Omega_p} \nabla \cdot \sigma \, dx - \int_{\Gamma_p} \sigma \kappa n \, dS + \int_{\Omega_p} \rho g \, dx.
$$

(19)

The advantage of this form of the two-fluid Navier-Stokes equations is that the air-liquid boundary conditions are implicitly contained in the equation of motion. We shall exploit this fact when we deduce the level-set form of the Navier-Stokes equations.

4 Level Set Function

We introduce the level set function, $\phi$, the zero level set of $\phi$ is the gas-liquid interface:

$$
\Gamma = \{ \mathbf{x} | \phi(\mathbf{x}, t) = 0 \}.
$$
We also take $\phi < 0$ in the gas region and $\phi > 0$ in the liquid region. Hence, we have

$$
\phi(x, t) = \begin{cases} 
> 0 & \text{if } x \in \text{the liquid} \\
= 0 & \text{if } x \in \Gamma \\
< 0 & \text{if } x \in \text{the gas.}
\end{cases}
$$

(20)

The unit normal on the interface, drawn from the gas into the liquid, and the curvature of the interface can easily be expressed in terms of $\phi(x, t)$

$$
n = \frac{\nabla \phi}{|\nabla \phi|}_{\phi=0} \text{ and } \kappa = \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right)_{\phi=0}.
$$

(21)

Since the interface moves with the fluid particles, the evolution of $\phi$ is then given by

$$
\frac{\partial \phi}{\partial t} + u \cdot \nabla \phi = 0.
$$

(22)

Because the density and viscosity are constant in each fluid, they take on two different values depending on the sign of $\phi$; hence we write

$$
\rho(\phi) = \rho_2 + (\rho_\ell - \rho_2)H(\phi)
$$

(23) and

$$
\mu(\phi) = \mu_2 + (\mu_\ell - \mu_2)H(\phi),
$$

(24)

where $H(\phi)$ is the Heaviside function given by

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

Next we use the level set function to rewrite the Navier-Stokes equations. We begin with the following:

**Claim**

$$
\int_{\Omega_\rho} \kappa(\phi) \nabla \phi \delta(\phi) \, dx = \int_{\Gamma_\rho} n \kappa dS.
$$

(25)

By $\kappa(\phi)$ we mean

$$
\kappa(\phi) = \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right).
$$
This claim follows from the level set form of $\mathbf{n}$, (21), and the relation
\[
\int g(x)\delta(f(x))\,dx = \frac{g(0)}{|f'(0)|}
\]
where $f(x)$ is a function which only vanishes at $x = 0$.
Combining (19), (23), and (25) we have
\[
\int_{\Omega_p} \left[ \rho(\phi) \frac{Du}{Dt} - \nabla \cdot \sigma \, dx + \kappa(\phi) \nabla \delta(\phi) - \rho(\phi)g \right] \, dx = 0.
\]
Since $\Omega_p$ is arbitrary we then have
\[
\rho(\phi) \frac{Du}{Dt} = \nabla \cdot \sigma - \kappa(\phi) \nabla \delta(\phi) + \rho(\phi)g.
\]
Next we substitute (8) and (24) into the above equation to obtain
\[
\rho(\phi) \frac{Du}{Dt} = -\nabla p + \nabla \cdot (2\mu(\phi)D) - \sigma \kappa(\phi) \delta(\phi) \nabla \phi + \rho(\phi)g. \tag{26}
\]

The Navier-Stokes equations for two-fluid flows was written in similar form by Unverdi & Tryggvason [6]. The fact that the surface tension can be written as a delta function concentrated at the interface has been used by Unverdi & Tryggvason [6] and Brackbill, Kothe, & Zemach [7]. The form of the surface tension we use here is due to Chang, Hou, Merriman, & Osher [8] but is similar to that used in [7]. The derivation of (26) presented here follows closely that found in [8].

### 4.1 Dimensionless form

It is useful to write (26) in dimensionless form using the following dimensionless variables:
\[
x = Lx', \quad u = Uu', \quad t = (L/U)t', \quad p = p'\rho_0U^2, \quad \rho = \rho\rho', \quad \mu = \mu\mu',
\]
where the primes denote dimensionless variables. If we substitute these variables into (26) and drop the primes, we have
\[
u_t + \nabla p = F, \tag{27}
\]
where

\[
F = -\mathbf{u} \cdot \nabla \mathbf{u} + \frac{z}{Fr} + \frac{1}{\rho(\phi)} \left( \frac{1}{Re} \nabla \cdot (2\mu(\phi)D) - \frac{1}{We} \kappa(\phi) \delta(\phi) \nabla \phi \right), \tag{28}
\]

and \( z \) is a unit vector in the \( z \)-direction. The density and viscosity are now

\[
\rho(\phi) = \lambda + (1 - \lambda) H(\phi) \quad \text{and} \quad \mu(\phi) = \eta + (1 - \eta) H(\phi), \tag{29}
\]

where \( \lambda = \rho_g/\rho_\ell \) is the density ratio and \( \eta = \mu_g/\mu_\ell \) is the viscosity ratio. The dimensionless groups used above are the Reynolds number,

\[ Re = \frac{\rho_\ell U L}{\mu_\ell}, \]

the Froude number,

\[ Fr = \frac{U^2}{gL}, \]

and the Weber number,

\[ We = \frac{\rho_\ell U^2}{\sigma}. \]

The level set form of the Navier-Stokes equation is given by (27), (28), and (29). The pressure is determined so that

\[ \nabla \cdot \mathbf{u} = 0 \tag{30} \]

This is achieved by a density weighted divergence-free projection operator described below. The level set function is updated from (22).

### 4.2 Weighted Divergence-Free Projection

Let \( w(\mathbf{x}) \) be a weight function and \( \mathbf{f}(\mathbf{x}) \) be an arbitrary vector field defined on \( \Omega \). Then the weighted divergence-free projection of \( \mathbf{f} \), denoted \( \mathbf{g} \), is defined as

\[
\mathbf{f} = \frac{1}{w} \nabla \theta + \mathbf{g},
\]

with \( \mathbf{g} \cdot \mathbf{n} = 0 \) on \( \partial \Omega \). Since \( \nabla \cdot \mathbf{g} = 0 \) then \( \theta \) must satisfy the following elliptic equation:

\[ \nabla \cdot \left( \frac{1}{w} \nabla \theta \right) = \nabla \cdot \mathbf{f}, \]
with
\[
\frac{\partial \theta}{\partial n} = f \cdot n \quad \text{on} \quad \partial \Omega.
\]
We shall denote the weighted projection by \( P_w \); therefore
\[
g = P_w(f).
\]
Note that
\[
P_w\left( \frac{1}{w} \nabla q \right) = 0,
\]
where \( q \) is any scalar field.

It clear then, we can eliminate the pressure from (27) by applying the weighted divergence-free projection operator. Therefore we have
\[
\mathbf{u}_t = P_{\mu(\phi)}(\mathbf{F}).
\tag{31}
\]
In the form of the Navier-Stokes equation above, we must take the divergence-free projection projection of a Dirac delta function. This will present some numerical difficulties; to alleviate this problem we note that
\[
\nabla H(\phi) = \delta(\phi) \nabla(\phi).
\]
Therefore it follows
\[
\kappa(\phi) \delta(\phi) \nabla(\phi) = \nabla (\kappa(\phi) H(\phi)) - H(\phi) \nabla \kappa(\phi);
\]
hence (31) can be rewritten as
\[
\mathbf{u}_t = P_{\rho(\phi)}(\bar{\mathbf{F}})
\]
where
\[
\bar{\mathbf{F}} = -\mathbf{u} \cdot \nabla \mathbf{u} + \frac{z}{Fr} + \frac{1}{\rho(\phi)} \left( \frac{1}{Re} \nabla \cdot (2\mu(\phi) \mathbf{D}) + \frac{1}{We} H(\phi) \nabla \kappa(\phi) \right),
\tag{32}
\]

4.3 Stream Function

In our computations we consider only two dimensional flows or axisymmetric flows; therefore \( \mathbf{x} = (x, y)^T \). In the axisymmetric case \( x \) is the radial coordinate and \( y \) is vertical coordinate. Let \( \nabla_c = (\partial_x, \partial_y)^T \) and \( \nabla_c^\perp = (\partial_x, -\partial_y)^T \). The restriction to these flows offers some simplification
as the velocity, \( \mathbf{u} = (u, v)^T \), can be deduced by a stream function. Thus it follows
\[
\mathbf{u} = \frac{1}{x^m} \nabla^+ \psi
\]
(33)
where \( m = 0 \) for two dimensional flows and \( m = 1 \) for axisymmetric flows. If we substitute (33) into the \( \mathbf{u}_t \) term of (27), multiply by \( \rho(\phi) \), and apply \( \nabla^+ \) to both sides, the pressure term is eliminated. This gives the following variable coefficient Poisson equation for \( \psi_t \) on \( \Omega \):
\[
\nabla_c \cdot \left( \frac{\rho(\phi)}{x^m} \nabla_c \psi_t \right) = \nabla^+ \cdot (\rho(\phi) \mathbf{F}),
\]
(34)
with boundary conditions
\[
\psi_t = 0 \quad \text{for} \quad \mathbf{x} \in \partial \Omega.
\]
(35)
These boundary conditions follow since the normal velocity of \( \mathbf{u} \) must vanish on the boundary. In our numerical computations we will take
\[
\Omega = \{(x, y)|0 \leq x \leq W \text{ and } 0 \leq y \leq H\},
\]
where \( W \) and \( H \) are the width and height of the domain.

The axisymmetric Navier-Stokes equations for two incompressible immiscible fluids in level set form are given by Equations (22), (29), (33), and (34).

5 Thick Interfaces

In order to solve (34) numerically we must modify it slightly due to the sharp changes in \( \rho \) across the front and also because of the numerical difficulties presented by the Heaviside function contained in \( \mathbf{F} \). To alleviate these problems we shall give the interface a fixed thickness that is proportional to the spatial mesh size. This allows us to replace \( \rho(\phi) \) by a smoothed density, \( \rho_\varepsilon(\phi) \), which is given by
\[
\rho_\varepsilon(\phi) = \lambda + (1 - \lambda) H_\varepsilon(\phi),
\]
(36)
with
\[
H_\varepsilon(\phi) = \begin{cases} 
0 & \text{if } \phi < -\varepsilon \\
\frac{1}{2} [1 + \frac{\phi}{\varepsilon} + \frac{1}{\pi} \sin(\pi \phi / \varepsilon)] & \text{if } |\phi| \leq \varepsilon \\
1 & \text{if } \phi > \varepsilon.
\end{cases}
\]
(37)
The smoothed or mollified delta function is

$$\delta_\varepsilon(\phi) = \frac{dH_\varepsilon}{d\phi}. \quad (38)$$

It is clear from (37) that the thickness of the interface is approximately

$$\frac{2\varepsilon}{|\nabla \phi|}. \quad (39)$$

In our algorithm the front will have a uniform thickness, consequently we must have $|\nabla \phi| = 1$ when $|\phi| \leq \varepsilon$. A function that satisfies

$$|\nabla d| = 1 \quad \text{for} \quad \mathbf{x} \in \Omega \quad \text{with} \quad d = 0 \quad \text{for} \quad \mathbf{x} \in \Gamma \quad (40)$$

is called a distance function. This is because $d$ is the signed normal distance to the interface, $\Gamma$.

If a level set function is equal to a distance function it then follows from (37) that the thickness of the interface is $2\varepsilon$. In our numerical calculations we shall take $\varepsilon = \alpha \Delta x$ where $\Delta x$ is the grid size. Therefore the interface will reduce in thickness as we refine our mesh.

It seems ideal then to choose the level set function to be a distance function. It is clear that we can choose $\phi(\mathbf{x}, 0)$ to be a distance function; however, under the evolution of (22) it will not necessarily remain as such. We must then be able to solve the following problem: given a level set function, $\phi(\mathbf{x})$, reinitialize it to be a distance function without changing its zero level set. This can be achieved by solving the following partial differential equation:

$$\frac{\partial d}{\partial \tau} = \text{sign}(\phi)(1 - |\nabla d|) \quad (41)$$

with initial conditions

$$d(\mathbf{x}, 0) = \phi(\mathbf{x}),$$

where

$$\text{sign}(\phi) = \begin{cases} -1 & \text{if } \phi < 0 \\ 0 & \text{if } \phi = 0 \\ 1 & \text{if } \phi > 0 \end{cases} \quad (42)$$

and $\tau$ is an artificial time. The steady solutions of (41) are distance functions. Furthermore, since $\text{sign}(0) = 0$, $d(\mathbf{x}, \tau)$ has the same zero level set as $\phi(\mathbf{x})$. Therefore we simply solve (41) to steady state and then replace $\phi(\mathbf{x})$ by
$d(x, \tau_{\text{steady}})$. It is clear from (37) that we need only $\phi$ to be a distance function close to the front. We have then reached “steady state” when

$$|\nabla d| = 1 \quad \text{for} \quad |d| \leq \varepsilon.$$ 

A nice feature of using this procedure is that the level set function is first reinitialized near the front. To see this we rewrite (41) as

$$d_r + \mathbf{w} \cdot \nabla d = \text{sign}(\phi),$$  

(43)

where

$$\mathbf{w} = \text{sign}(\phi) \frac{\nabla d}{|\nabla d|}.$$ 

It is evident that (43) is a nonlinear hyperbolic equation with the characteristic velocities pointing outwards from the interface in the direction of the normal. Thus $d$ will be first reinitialized to $|\nabla d| = 1$ near the interface. Since we need the level set function only to be a distance function near the interface, it is not necessary to solve (43) to steady state over the whole domain. This indicates only a fixed number of iterations are necessary in order to ensure that the level set function is a distance function near the interface. For example, if the iteration stepsize is $\Delta \tau$ and the total interfacial thickness is $2\varepsilon$, we can stop the iteration process after no more than $\varepsilon / \Delta \tau$ time steps. In practice we find that we need only two or three iterations as we are already close to the distance function.

6 Numerical Procedure

The details of the numerical procedure can be found in Paper I. The modifications for the axisymmetric geometry are straightforward and can be found in Sussman [1].

6.1 Numerical Algorithm

Step 1. Initialize $\phi(x, 0)$ to be the signed normal distance to the front. Initialize $\mathbf{u}(x, 0)$ to be the initial divergence-free velocity (identically zero in our case).

Step 2. Compute $\mathbf{F}$ by replacing $\delta(\phi)$ with $\delta_\varepsilon(\phi)$ and $H(\phi)$ with $H_\varepsilon(\phi)$ in (28).
Step 3. Solve the Poisson equation \((34)\) for \(\psi_i\) and compute a new \(u_i\). Use this to update \(u\); denote the updated \(u\) as \(u^{n+1}\).

Step 4. Update the location of the interface by solving \((22)\) for one time step using \(u^{n+1}\). Denote the updated value of \(\phi\) by \(\phi^{n+\frac{1}{2}}\).

Step 5. Reinitialize \(\phi^{n+\frac{1}{2}}\) by solving
\[
\phi_x = S_\varepsilon(\phi^{n+\frac{1}{2}})(1 - |\nabla\phi|) \quad \text{with} \quad \phi(x, 0) = \phi^{n+\frac{1}{2}}(x)
\]
to "steady state". Here \(S_\varepsilon\) is a smoothed version of \((42)\). We denote this solution by \(\phi^{n+1}\). By steady state we mean that \(|\nabla\phi| = 1 + O(\Delta x)\) for \(|\phi| < \varepsilon\) (as noted previously \(\varepsilon = \alpha \Delta x\) and typically \(\alpha\) is 2).

Step 6. We have now advanced one time step. The zero level set of \(\phi^{n+1}\) gives the new interface position and it is a distance function close to the front. One then proceeds back to step 2.

### 6.2 Numerical Implementation

We shall describe the implementation for two dimensional flows. The velocity field and the level set function are defined at cell centers \((i, j)\) where \(i\) and \(j\) are integers. The stream function is defined at cell corners \((i + \frac{1}{2}, j + \frac{1}{2})\); we shall use square cells of size \(h\).

#### 6.2.1 Projection

We use the discrete projection operator developed by Bell, Colella, and Glaz [9]. This produces velocity which is discretely divergence-free and uses a compact stencil (unlike Chorin’s projection operator). We define the discrete derivatives:
\[
G_x(\psi) = \frac{1}{2h} \left( \psi_{i+\frac{1}{2},j+\frac{1}{2}} - \psi_{i-\frac{1}{2},j+\frac{1}{2}} + \psi_{i+\frac{1}{2},j-\frac{1}{2}} - \psi_{i-\frac{1}{2},j-\frac{1}{2}} \right)
\]
\[
G_y(\psi) = \frac{1}{2h} \left( \psi_{i+\frac{1}{2},j+\frac{1}{2}} - \psi_{i+\frac{1}{2},j-\frac{1}{2}} + \psi_{i-\frac{1}{2},j+\frac{1}{2}} - \psi_{i-\frac{1}{2},j-\frac{1}{2}} \right),
\]

the discrete 2D curl,
\[
G^\perp(\psi) = (G_y(\psi), -G_x(\psi))^T,
\]

13
and the discrete gradient

\[ \mathbf{G}(\psi) = (G_x(\psi), G_y(\psi)). \]

The discrete form of (34) is

\[ \mathbf{G} \cdot (\rho \mathbf{G}(\psi_t)) = \mathbf{G}^\perp \cdot (\rho \mathbf{F}). \]

This is solved iteratively with a preconditioned conjugate gradient scheme. This gives us \( \psi_t \) at the cell corners, and then integrated in time to find the values of \( \psi_{i, j+1/2} \). Using a discrete form of (33) we can find velocities at the cell faces

\[ \bar{u}_{i, j+1/2} = \frac{1}{\Delta t} \left( \psi_{i+1/2, j+1/2} - \psi_{i+1/2, j-1/2} \right) \]

and

\[ \bar{v}_{i, j+1/2} = \frac{1}{\Delta t} \left( \psi_{i+1/2, j+1/2} - \psi_{i-1/2, j-1/2} \right). \]

This is discretely divergence-free

\[ \bar{u}_{i, j+1/2} - \bar{u}_{i-1/2, j} + \bar{v}_{i, j+1/2} - \bar{v}_{i, j-1/2} = 0. \]
We find velocities at the cell centers by

\[ u_{ij} = \frac{1}{2} (\bar{u}_{i+\frac{1}{2},j} + \bar{u}_{i-\frac{1}{2},j}) \]

\[ v_{ij} = \frac{1}{2} (\bar{v}_{i,j+\frac{1}{2}} + \bar{v}_{i,j-\frac{1}{2}}) \]

### 6.2.2 Convection Terms

We compute convection terms using upwind methods with ENO-type interpolants to achieve high order. The discrete form we use is

\[ \mathbf{u} \cdot \nabla f = u_{ij} \frac{f_{i+\frac{1}{2},j} - f_{i-\frac{1}{2},j}}{h} + v_{ij} \frac{f_{i,j+\frac{1}{2}} - f_{i,j-\frac{1}{2}}}{h}, \]

where \( f_{i+\frac{1}{2},j} \) and \( f_{i,j+\frac{1}{2}} \) are computed using a variant of ENO procedures developed by Shu & Osher\[10\].

**ENO interpolation.** We show how the 2nd order ENO interpolants in the \( x \)-direction are made. We start with the following definitions:

\[ k = \begin{cases} i & \text{if } \bar{u}_{i+\frac{1}{2},j} > 0 \\ i+1 & \text{otherwise} \end{cases} \]  \hspace{1cm} (44)

\[ c = \begin{cases} a & \text{if } |a| < |b| \\ b & \text{otherwise} \end{cases} \]  \hspace{1cm} (45)

where \( a = (f_{k,j} - f_{k-1,j})/h \) and \( b = (f_{k+1,j} - f_{k,j})/h \). The ENO interpolant is then

\[ f_{i+\frac{1}{2},j} = f_{k,j} + \begin{cases} ch/2 & \text{if } \bar{u}_{i+\frac{1}{2},j} > 0 \\ -ch/2 & \text{otherwise} \end{cases} \]  \hspace{1cm} (46)

### 6.2.3 Reinitialization Step

As pointed out previously, the reinitialization step consisted of solving a nonlinear hyperbolic equation. This is also done with an ENO scheme analogous to the one outlined above (see \[3\] for more details).

### 6.2.4 Surface Tension and Viscous Terms

All the terms associated with curvature and the rate of deformation tensor, \( D \) are computed with center differences.
6.2.5 Time Discretization

We use second order Runge-Kutta to advance in time.

6.3 Improvements

The algorithm described above is presented in [3] and contains many improvements of [2]. In addition, we typically use third-order ENO methods. Finally, we use an improvement in the reinitialization step which was developed by Sussman & Fatemi [11] in order to increase its accuracy.

We will now briefly discuss the modification to the reinitialization step. We interpret the term, \( \text{sign}(\phi) \), in equation (41) as a “constraint” used both to prevent the interface from moving and also to implicitly prescribe boundary conditions at the interface. For discretization purposes we wish to enforce another constraint: the volume filled by each fluid must stay constant throughout the reinitialization. For each cell, \( \Omega_{ij} \), we define volume as

\[
V_{ij}^n = \int_{\Omega_{ij}} H(\phi^n)rdrdz,
\]

where \( H \) is the Heaviside function described by equation (4) and \( \phi^n \) is the value of \( \phi \) at \( t_n \), the “time” after the \( n \)th iteration in the reinitialization step.

Because volume should not change, we should have \( V_{ij}^n = V_{ij}^0 \). Nevertheless, if the reinitialization step slightly changes the location of the zero level set, we then have, for \( \tau_n - \tau_0 = O(\Delta x) \),

\[
V_{ij}^n - V_{ij}^0 \approx (\tau_n - \tau_0) \int_{\Omega_{ij}} \frac{dH_z(\phi^0)}{d\tau} rdrdz
\]

\[
= \int_{\Omega_{ij}} \frac{dH_z(\phi^0)}{d\phi} (\phi^n - \phi^0)rdrdz,
\]

where

\[
\frac{dH_z}{d\phi} = \begin{cases} 
0 & \text{if } |\phi| > \varepsilon \\
\frac{1}{\varepsilon^2} \left[ \frac{1}{\varepsilon} + \frac{1}{\varepsilon^2} \cos(\pi \phi/\varepsilon) \right] & \text{if } |\phi| \leq \varepsilon.
\end{cases}
\]

In order to minimize volume variation, we project the current values of the level set function, denoted as \( \tilde{\phi}_{ij}^n \), onto new values, denoted as \( \phi_{ij}^n \), which satisfy

\[
\int_{\Omega_{ij}} \frac{dH_z(\phi^0)}{d\phi} (\phi^n - \phi^0)rdrdz = 0.
\]
If (50) is satisfied then by (48) the volume change will be very small. To implement this projection we assume \( \phi_{ij}^n \) has the form,

\[
\phi_{ij}^n = \bar{\phi}_{ij}^n + \lambda_{ij}(\tau_n - \tau_0) \frac{dH(\phi^0)}{d\phi},
\]

where \( \lambda_{ij} \) is assumed constant in \( \Omega_{ij} \). After substituting (51) into (50), we have

\[
\lambda_{ij} = \frac{-\int_{\Omega_{ij}} \frac{dH(\phi^0)}{d\phi} \left(\bar{\phi}_{ij}^n - \phi^0\right) \left(\frac{\tau_n - \tau_0}{2}\right) rdrdz}{\int_{\Omega_{ij}} \left(\frac{dH(\phi^0)}{d\phi}\right)^2 rdrdz}.
\]

Equation (52) is discretized in each cell by using a nine-point stencil to perform the integration. Since \( \lambda_{ij} \) is assumed constant in each cell, the above equation can be solved both explicitly and quickly. The projection step given by (51) is applied after each reinitialization step. In our work we have found that the above constraint helps \( \phi \) converge to a distance function while still maintaining the original zero level set.

### 6.3.1 Far-field boundary conditions

Many experimental situations occur in large domains. Due to limited computing resources it is not possible to compute in domains of this size. In certain situations, the velocity field decays to zero sufficiently fast in space so that it is still possible to compare with experiments even when the computational domain is small. There are, however, situations where the effects of a small computational domain make comparison with experiments difficult. In these cases it is expedient to modify the scheme near the boundary to produce “far-field” boundary conditions.

We still enforce zero normal velocity at the boundary, but assume that the cells, \( \Omega_{ij} \), which touch the wall, have dimensions of \( M \times \Delta x \) as opposed to \( \Delta x \times \Delta x \). The difference formulas in the advection and projection steps are then modified accordingly. The constant, \( M \), can be specified to be as large as necessary, as long as the tangential velocity does not change much in the direction normal to the boundary. For a gas bubble that attains a steady state the expected steady rise speed matches experimental findings within 5% when our far-field boundary conditions are used. Without the
far-field boundary conditions, wall effects would cause the steady rise speed to be 14% slower than the expected value.

References


