UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

# A Hybrid Method for Moving Interface Problems with Application to the Hele-Shaw Flow

Zhilin Li
Hongkai Zhao
Stanley Osher

April 1996
CAM Report 96-9

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

# A HYBRID METHOD FOR MOVING INTERFACE PROBLEMS WITH APPLICATION TO THE HELE-SHAW FLOW *

ZHILIN LI †, HONGKAI ZHAO ‡, AND STANLEY OSHER §

**Abstract.** In this paper, a hybrid approach which combines the *immersed interface method* with the *level set approach* is presented. The fast version of the immersed interface method [Li, 1995], is used to solve the differential equations whose solutions and their derivatives may be discontinuous across the interfaces due to the discontinuity of the coefficients or/and singular sources along the interfaces. The moving interfaces then are updated using the newly developed fast level set formulation which involves computation only inside some small tubes containing the interfaces. Several key steps in the implementation are addressed in detail. This new approach is then applied to compute Hele-Shaw flow, an unstable flow involving two fluids with very different viscosity. Compared to the traditional boundary integral approaches, the method proposed in this paper is faster and simpler.

**Key words.** discontinuous coefficients, fast level set formulation, finite difference, fast immersed interface method, Hele-Shaw flow, level set approach, moving interface, re-initialization

**AMS subject classifications.** 65N06, 76H20, 76S05

## 1. Introduction.

Many physical problems involve interfaces where two different materials contact with each other. The interfaces between different materials usually move with time. Mathematically, interface problems lead to differential equations whose input data and solutions have discontinuities or non-smoothness across the interfaces. The motion of the interfaces is often governed by some non-linear differential equations.

* This work was supported by URI grant #N00014092-J-1890 from ARPA, NSF Grant DMS-9303404, and DOE Grant DE-FG06-93ER25181.

† Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095. (zhilin@math.ucla.edu).

‡ Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095. (hzhao@math.ucla.edu).

§ Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095. (sjo@math.ucla.edu).

As a model test problem, which itself is important to computational fluid dynamics world, we consider the Hele-Shaw flow in which a less viscous and immiscible fluid is injected into a more viscous flow. The interface between the two fluids is expanding with time. We would like to predict the location of the interface as a function of time. In order to do so, we have to find the pressure and the velocity fields which are determined from some partial differential equation. Across the interface, the pressure has a finite jump which is proportional to the surface tension and the curvature. Mathematically, this corresponds to a singular dipole source along the interface in the partial differential equation. Obviously, the viscosity, which determines the coefficient of the partial differential equation, is very different across the interface. Therefore to solve an interface problem, there are two major parts involved.

The first part is to solve the differential equations involving interfaces across which the discontinuities occur either in the coefficients or in the solution and its derivatives. The *immersed interface method* [10, 15] is an efficient approach for such problems. This new method can handle both discontinuous coefficients and singular sources. The main idea is to incorporate the known jump conditions in the solution and its derivatives, e.g., the flux across the interface, into the finite difference scheme, obtaining a modified scheme on a Cartesian grid for quite arbitrary interfaces. Numerical tests have shown that solutions obtained from this method are second order accurate at all points. This approach has also been applied to three-dimensional elliptic equations [18], parabolic equations[17, 19, 20], hyperbolic wave equations with discontinuous coefficients [12, 13], and incompressible Stokes flow problems with moving interfaces [11, 14].

As a brief example to show how the immersed interface method works, consider a Poisson equation on a solution domain $\Omega$

$$\Delta u = \int_\Gamma C(s)\,\delta(x - X(s))\,\delta(y - Y(s))\,ds, \qquad (x, y) \in \Omega, \tag{1.1}$$

with given boundary condition on $\partial\Omega$, where $(X(s), Y(s))$ is the arc length parameterization of the interface $\Gamma$. With the immersed interface method, the problem can be written as

$$\Delta u = 0, \qquad (x, y) \in \Omega - \Gamma, \tag{1.2}$$

$$[u] = 0; \qquad [u_n] = C(s), \tag{1.3}$$

where $[u]$ and $[u_n]$ are the jumps of the solution and its normal derivative across the interface $\Gamma$ respectively. The discrete form of (1.1) is simply the standard five-point scheme plus a correction term at irregular grid points where the interface cuts through the five-point stencil,

$$\frac{U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - 4\,U_{i,j}}{h^2} = f_{ij} + C_{ij}. \tag{1.4}$$

Thus a fast Poisson solver such as a fast Fourier transformation method (FFT), cyclic reduction, etc. [30], can be employed if the solution domain is a rectangle.

Solving an elliptic equation in the following form

$$\nabla\,(\beta(x,y)\nabla u) = f(x,y), \tag{1.5}$$

where the coefficient $\beta$ is discontinuous across the interface(s), is a major part in many computational fluid dynamics problems, see [3, 8, 29] etc. It is also one of the two governing equations in the simulation of Hele-Shaw flow, see Section 5. For the Hele-Shaw flow, the solution itself is discontinuous across the interface which corresponding to a dipole source along the interface. The equation (1.5) is only valid in the interior of the domain, but not on the interface. To solve equation (1.5), we can not use a fast Poisson solver with the original immersed interface even if $\beta$ is piecewise constant. Thus better iterative methods are sought to solve the linear system of equations resulting from the discretization of *the immersed interface method.* One such method is the multi-grid method developed by Adams [2]. The approach used here is based on a *preconditioning approach* [16], which will be briefly discussed in Section 2.

The second question in solving an interface problem is how to update the interface(s) to a certain, preferably second order, accuracy. Three possible numerical algorithms are, the "volume of fluid" technique, the marker particle approach, and the level set formulation which is the one used in this paper.

We will concentrate on the two phase flow problems here although the similar techniques can be, and actually have been, applied to multi-phase problems. Suppose there is a closed interface in the solution domain $\Omega$. For the simulation of Hele-Shaw flow discussed in this paper, the interface separates the less viscous flow from more viscous one. Let $\Gamma(t)$ be the moving interface, $\Omega^-(t)$ and $\Omega^+(t)$ be the interior and exterior regions of the interface respectively. The moving interface $\Gamma(t)$ can be described

as the zero set of a function $\varphi(x,y,t)$, which is Lipschitz continuous, satisfying

$$\varphi(x,y,t) > 0 \quad \text{for} \quad (x,y) \in \Omega^-, \tag{1.6a}$$

$$\varphi(x,y,t) = 0 \quad \text{for} \quad (x,y) \in \Gamma, \tag{1.6b}$$

$$\varphi(x,y,t) < 0 \quad \text{for} \quad (x,y) \in \Omega^+. \tag{1.6c}$$

Therefore, by differentiating the level set $\varphi(x,y,t) = c$ with respect to time $t$, we can obtain the equation of motion of the level set,

$$\varphi_t + \vec{u} \cdot \nabla\varphi = 0. \tag{1.7}$$

This is referred to as a Hamilton-Jacobi level set equation. Initially, $\varphi(x,y,0)$ is often chosen as the signed normal distance from the interface which satisfies $|\nabla\varphi| = 1$. By solving the Hamilton-Jacobi equation, we can update the moving interface, the zero set of $\varphi(x,y,t) = 0$. However, while equation (1.7) will move the interface, the level set $\varphi = 0$ at the correct speed, $\varphi$ will no longer be a distance function after some time. Thus $\varphi$ can have a very steep or flat gradient especially when topological changes such as breaking and merging take place, or when the velocity field near the interface has some singularities. This difficulty can be avoided by a re-initialization process [28, 29] so that $\varphi$ will remain as the signed distance function to certain accuracy. The new level set function $\varphi$ is the steady state solution of the following equations,

$$\varphi_t + (|\nabla\varphi| - 1)\, H(\varphi) = 0, \tag{1.8}$$

where $H(\varphi)$ is any smooth monotone function of $\varphi$ with $H(0) = 0$. ENO schemes for Hamilton-Jacobi equations [23, 24] may be used to compute the signed distance quickly.

Another issue related to the level set approach is to reduce the computational cost. A natural way is to introduce a tube along the moving interface and just compute the level set function inside the tube, [1, 33]. This is also the approach used in this paper.

**2. The preconditioned immersed interface method for elliptic equations with piecewise constant coefficients.** Li [16] has proposed a new scheme for general elliptic interface problems which satisfies the maximum principle. Therefore it is easier to analyze. Moreover this new approach can be easily applied to time-dependent

problems. The idea is to precondition the differential equation before applying the immersed interface method. An intermediate unknown function, the jump in the normal derivative across the interface, is introduced. The discretization is equivalent to using a second order difference scheme for the regular grid points in the region, and a second order discretization for a Neumann-like interface condition. The maximum principle will be satisfied and the solution is second order accurate globally based on conventional analysis [22]. The idea is explained briefly by considering the following elliptic interface problem.

**Problem (I).**

$$\nabla\left(\beta(x,y)\nabla u\right) = f(x,y), \tag{2.9a}$$

$$\textit{Given BC} \quad \textit{on} \quad \partial\Omega, \tag{2.9b}$$

with specified jump conditions along the interface $\Gamma(s)$

$$[u] = w(s), \tag{2.10a}$$

$$[\beta u_n] = v(s). \tag{2.10b}$$

where $s$ is the arc-length of the interface.

With the original immersed interface method, we are able to derive a difference scheme for which the local truncation error is $O(h^2)$ away from the interface, and $O(h)$ near the interface, with a six-point stencil. However, if the jump in the coefficient $\beta$ is too large as in the case of the Hele-Shaw flow discussed in this paper, the difference scheme may lose the sign property required for the maximum principle because the scheme has to be modified to satisfy the flux condition (2.10b). On the other hand, if we can use the jump condition in the normal derivative $[u_n]$, which is unknown in the problem (I), then the modified difference scheme will preserve the sign property. Thus we consider the solution $u_g(x,y)$ of the following problem, which depends on a newly introduced function $g(s)$, the jump in the normal derivative.

**Problem (II).**

$$\begin{aligned} \Delta u + \frac{\nabla\beta^-}{\beta^-}\cdot\nabla u &= \frac{f}{\beta^+}, \quad \textit{if } x\in\Omega^+ \\ \Delta u + \frac{\nabla\beta^+}{\beta^+}\cdot\nabla u &= \frac{f}{\beta^-}, \quad \textit{if } x\in\Omega^-, \end{aligned} \tag{2.11a}$$

$$\textit{Given BC} \quad \textit{on} \quad \partial\Omega, \tag{2.11b}$$

with specified jump conditions

$$[u] = w(s), \tag{2.12a}$$

$$[u_n] = g(s). \tag{2.12b}$$

Notice that the jump conditions (2.12a) and (2.12b) depend on the singularities of the source term $f(x,y)$ along the interface, for example, if $f(x,y)$ contains the source distribution along the interface, then the source strength will contribute to the jump in the normal derivative across the interface. However, in the expression of (2.11a), we do not need information about $f(x,y)$ on the interface $\Gamma$, so there is no need to write $f(x,y)$ differently. Let the solution of Problem (I) be $u^*(x,y)$, and define

$$g^*(s) = [u_n^*](s) \tag{2.13}$$

along the interface $\Gamma$. Then $u^*(x,y)$ satisfies the elliptic equation (2.11a)-(2.11b) and the jump conditions (2.12a)-(2.12b) with $g(s) \equiv g^*(s)$. In other words, $u_{g^*}(x,y) \equiv u^*(x,y)$, and

$$[\beta \frac{\partial u_{g^*}}{\partial n}] = v(s) \tag{2.14}$$

is satisfied. Therefore, solving Problem (I) is equivalent to finding the corresponding $g^*(s)$ and then $u_{g^*}(x,y)$ in Problem (II). Notice that $g^*(s)$ is only defined along the interface, so it is one dimension lower than $u(x,y)$.

For simplicity, consider the special case where $\beta$ is piecewise constant. Now we have $\nabla\beta^- = 0$ and $\nabla\beta^+ = 0$. It is very easy to discretize Problem (II) using the immersed interface method. We have the standard discrete Laplacian operator for $\Delta u$ and the central difference scheme for $\nabla \cdot u$ plus some correction terms. The correction terms are the functions of the jump $[u]$, $[u_x]$, $[u_y]$, $[u_{xx}]$, $[u_{yy}]$, which are all known if we know $[u]$ and $[u_n]$ along the interface. The jump condition (2.10b) then is discretized using our new technique called *the weighted least squares interpolation* [16], which is very robust and second order accurate for interface problems. Thus the discrete form of our approach can be written as the following linear system,

$$\begin{bmatrix} A & B \\ E & D \end{bmatrix} \begin{bmatrix} U \\ G \end{bmatrix} = \begin{bmatrix} F \\ V \end{bmatrix}. \tag{2.15}$$

The solution $U$ and $G$ are the discrete form of the solution $u_{g^*}(x,y)$ and $g^*(s)$, the solution of Problem (II) which satisfies (2.14). Eliminating $U$ from (2.15) gives a linear system for $G$

$$\begin{aligned}(D - EA^{-1}B)G &= V - EA^{-1}F \\ &\stackrel{\text{def}}{=} \bar{V}.\end{aligned} \tag{2.16}$$

This is a much smaller linear system compared to the one for $U$. The coefficient matrix is the Schur complement of $D$ in (2.15). In practice, the matrices $A$, $B$, $\cdots$, and the vectors $\bar{V}$, $F$ are never explicitly formed. The matrix and vector are written above merely for theoretical purposes. Thus an iterative method, such as GMRES iteration [26], is preferred. With the preconditioner proposed in [16], the number of iterations for solving the Schur complement system of (2.16) is independent of the jump $[\beta]$ and the mesh size $h$. The details of the algorithms can be found in our recent work [16] for piecewise constant $\beta$ where we can take advantage of fast Poisson solvers. For variable coefficients, we plan to use the multi-grid method developed by Adams [2].

**3. Reconstruction of the interface from the level set function.** In the level set representation, the interface, which is the set of points $(x,y)$ satisfying $\varphi(x,y) = 0$, is not explicitly given[1]. Instead we only have information $\varphi(x_i, y_j)$ at each grid point. In order to use the preconditioned immersed interface method, we need to find a number of control points on the interface so that we can set up equations for the intermediate unknowns $[u_n]$. We now describe this reconstruction process.

First we can find a starting point on the interface. Take any grid point $(x_i, y_j)$ where we have $\varphi_{ij}\varphi_{i+1,j} \le 0$. We know that there is a point $(x^*, y_j)$ which is on the interface. The function $\varphi(x, y_j)$ can be approximated, say, by a quadratic function $q(x)$. The point $(x^*, y_j)$ can be found with second order accuracy by solving the quadratic equation $q(x) = 0$. Higher order accuracy can be achieved if a higher order approximating function is used.

Once we have a point on the interface, we can march along the interface in a fixed direction, say clockwise, and mark the control points. We now give an outline of the process to locate the next control point $(X_{k+1}, Y_{k+1})$ on the interface from a known

[1] Note: We will omit the time dependence of the level set fuction $\varphi(x,y,t)$ for simplicity if no confusion occurs.

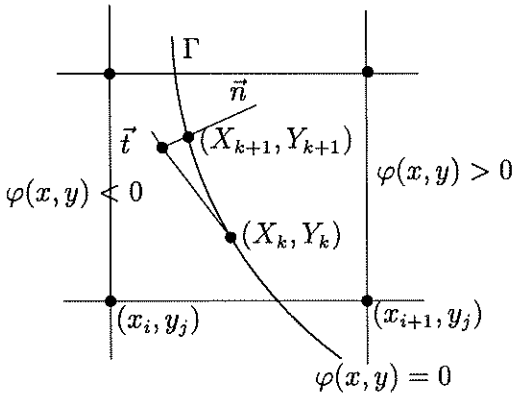


FIG. 1. *Finding the control point $\vec{X}_{k+1}$ from $\vec{X}_k$.*

point $(X_k, Y_k)$ on the interface. We will use the following notations

$$\vec{X} = (X, Y), \quad \vec{p} = \frac{\nabla \cdot \varphi}{||\nabla \cdot \varphi||}.$$

1. Find the tangential direction $\vec{t}_k$, $||\vec{t}_k|| = 1$.
2. Find the approximation of $\vec{X}_{k+1}$ along the tangential line:

$$\vec{X}^*_{k+1} = \vec{X}_k + s_k \vec{t}_k \tag{3.17}$$

where $s_k$ is the step size chosen to discretize the interface, we can choose $s_k$ as an increasing function of the smoothness of the interface.

3. Find the unit steepest ascent direction $\vec{p}$ at $\vec{X}^*_{k+1}$.
4. Locate the projection of $\vec{X}^*_{k+1}$ on the interface along the the direction $p$:

$$\vec{X}^*_{k+1} \longleftarrow \vec{X}^*_{k+1} + \alpha_k \vec{p} \tag{3.18}$$

where $\alpha_k$ is determined from the following quadratic equation:

$$\varphi\left(\vec{X}^*_{k+1}\right) + ||\nabla \cdot \varphi||\, \alpha + \frac{1}{2}\left(\vec{p}^T He(\varphi)\vec{p}\right)\ \alpha^2 = 0 \tag{3.19}$$

where $He(\varphi)$ is the Hessian matrix of $\varphi$

$$He(\varphi) = \begin{bmatrix} \varphi_{xx} & \varphi_{xy} \\ \varphi_{yx} & \varphi_{yy} \end{bmatrix} \tag{3.20}$$

calculated at $\vec{X}^*_{k+1}$.

5. Repeat step 3 and step 4 if necessary.

The key to the success of the algorithm above is to find accurate values of $\varphi$, $\varphi_x$, $\varphi_y$, $\varphi_{xx}$, $\varphi_{xy}$, and $\varphi_{yy}$ on a point lying off the grid. This is done through bi-linear interpolation. Note that at a grid point, this information can be calculated through central difference approximations unless there is a singularity of $\varphi$ in the neighborhood of that grid point.

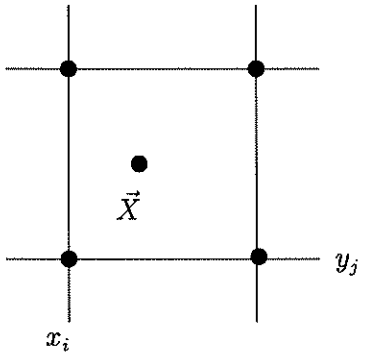


FIG. 2. *A diagram for the bi-linear interpolation.*

Given any point $(x, y)$, we can find the square which contains the point $(x, y)$ but no other grid points except the four vertices, say, $(x_i, y_j)$, $(x_{i+1}, y_j)$, $(x_i, y_{j+1})$, and $(x_{i+1}, y_{j+1})$. Let $G_{ij}$ be a grid function which approximate $\varphi$, $\varphi_x$ etc. Then we can use the following bi-linear interpolation to get an approximation to $G(x, y)$

$$G(x,y) = \frac{1}{4} \sum_{k=0,l=0}^{1} G_{i+k,j+l}\, \bar{x}_k\, \bar{y}_l \tag{3.21}$$

where

$$\begin{aligned} \bar{x}_k &= 1 + (2k-1)\left(\frac{2(x - x_{i+k})}{h} - 1\right), \\ \bar{y}_l &= 1 + (2l-1)\left(\frac{2(y - y_{j+l})}{h} - 1\right). \end{aligned}$$

With the approach described in the outline, we can get the set of control points $(X_k, Y_k)$ representing the interface with second order accuracy. To use the precondintioned immersed interface method we still need to compute the normals, tangents, and curvatures. We might obtain these information by interpolating those control points to get an expression of the interface, for example, a spline interpolation, and then differentiating the interpolant. This is not very accurate and is quite complicated to implement. The better way to express the derivative information is to use the level set

information again. For example, the unit normal, unit tangent, and the curvature at a point can be expressed as

$$\vec{n} = -\frac{\nabla\cdot\varphi}{||\nabla\cdot\varphi||} = -\left[\frac{\varphi_x}{\sqrt{\varphi_x^2+\varphi_y^2}}, \frac{\varphi_y}{\sqrt{\varphi_x^2+\varphi_y^2}}\right]^T, \tag{3.22}$$

$$\vec{t} = \left[-\frac{\varphi_y}{\sqrt{\varphi_x^2+\varphi_y^2}}, \frac{\varphi_x}{\sqrt{\varphi_x^2+\varphi_y^2}}\right]^T, \tag{3.23}$$

$$\kappa = -\frac{\varphi_{xx}\varphi_y^2 - 2\varphi_{xy}\varphi_x\varphi_y + \varphi_{yy}\varphi_x^2}{\left(\varphi_x^2+\varphi_y^2\right)^{2/3}} \tag{3.24}$$

They can be expressed in terms of the values $\varphi_{ij}$ of the level set function at the grid points, using bi-linear interpolation.

**4. Updating the interface by the localized level set method.** When the traditional level set method is used to capture the moving interface in real physical problems, some difficulties may occur.

- It is difficult and expensive to extend the normal velocity field, which is often only physically meaningful at the interface, to the whole domain. This is true especially when global quantities are used to determine the normal velocity at the interface or when the normal velocity field is singular, both of which occur in our Hele-Shaw example.
- In the classical level set method, we need to calculate $\varphi$ at all grid points, which involves an extra unnecessary order of magnitude of calculations.

Both these difficulties can be very well addressed by the fast localization technique introduced in [33] and details can be found there. The whole computation for the level set method is now only done in a very narrow tube around the moving interface. The size of the tube is fixed and can be just a few grid cells wide. The whole process is very simple and intuitive mathematically. It is composed of the following three steps at each time level (without loss of generality, here we suppose that the initial level set function is a signed distance normal to the interface)

1. Update the level set function in a tube of width $\epsilon_1 > 0$ by the evolution PDE for the level set function.

$$\phi_t + \vec{u}\cdot\bigtriangledown\phi = 0.$$

2. Construct a new tube of width $\epsilon_2 > \epsilon_1$ around the new interface (zero level set of the updated level set function) by correctly adding and deleting grid points

following the moving direction of the interface.

3. Reinitialize the level set function in the new $\epsilon_2$ tube to be a signed distance function by interspersing iterations of the following nonlinear partial differential equation, see [29, 33].

$$\phi_\tau + sign(\phi)(|\bigtriangledown \phi| - 1) = 0 \tag{4.25}$$

with the evolution procedure.

This localization technique involves an upwind scheme which requires only one boundary condition. However, with the tube approach, that boundary of the old tube is no longer needed. Thus an explicit tube boundary conditions, which can be very difficult to prescribe and affect the computation, is avoided. Also any discontinuities at the edge of the tube do not affect the computation in the tube, see [33].

**5. Hele-Shaw flow simulation.** In 1958, Saffman and Taylor [27] performed experiments replacing a viscous fluid from between two closely spaced, parallel plates with a less viscous fluid. The shape of the interface is well known to exhibit a fingering phenomenon.

$$\vec{u} = -\beta \nabla p, \tag{5.26}$$

$$\nabla \cdot \vec{u} = \phi, \tag{5.27}$$

with $\beta = b^2/(12\mu)$, where $b$ is the gap width and $\mu$ is the viscosity, which is very different inside and outside the interface separating the two fluids. The source term $\phi$ is the result of the injection of the less viscous fluid into the Hele-Shaw cell,

$$\phi = \begin{cases} \phi_0 \left(1 + \cos(r\pi/r_0)\right) & \text{if } r \le r_0 \\ 0 & \text{if } r > r_0, \end{cases} \tag{5.28}$$

where $r = \sqrt{x^2 + y^2}$. The total injection rate is

$$\bar{\phi} = \int \phi(x, y) dx dy = \phi_0 \left(\pi - \frac{4}{\pi}\right) r_0^2. \tag{5.29}$$

The jump conditions across the interface are

$$[p] = \tau\kappa, \qquad \text{the Laplace-Young Condition,} \tag{5.30}$$

$$[\beta\, p_n] = 0, \qquad \text{the kinematic interface condition,} \tag{5.31}$$

where $\tau$ is the surface tension and $\kappa$ is the curvature of the interface.

Simulation of Hele-Shaw flow has attracted a lot of attention and serves as a benchmark problem for numerical algorithms to compute unstable fronts [4, 5, 6, 9, 21, 25, 31, 32]. This is an ideal test model for our proposed algorithm since (5.26) can be written as

$$\nabla (\beta \nabla p) = -\phi. \tag{5.32}$$

We have developed a second order method for (5.32) with more general jump conditions than (5.30) already [16]. We also have the information of $p_n^{\pm}$ with our updated immersed interface method discussed in Section 2.

To determine the boundary condition on the pressure, we assume the interface is far enough away from the boundary that the flow at the boundary agrees with the radial outflow that would arise from the source term in a uniform fluid, *i.e.*,

$$p(x, y) = p_0 - \frac{\phi_0}{2\pi\beta} \log r \tag{5.33}$$

is specified for $(x, y)$ on the boundary, where $p_0$ is some arbitrary constant.

**5.1. Computing the velocity field near the interface.** When we solve the Hamilton-Jacobi equation to update the level set $\varphi$, we need to compute the velocity field from the pressure. The velocity field can be given either in component form in the $x$- and $y$- directions or $\vec{u} \cdot \vec{n}$, the the normal velocity of the level set. With the preconditioned immersed interface method, we have the normal derivatives $p_n^+$ and $p_n^-$ at control points $(X_k, Y_k)$ once the pressure $p$ is computed. We also know the normal velocity at each control point, which is

$$u_n = -\beta p_n. \tag{5.34}$$

Note that $\beta p_n$ is continuous, so it does not matter which side the quantity is taken from. Thus we decided to find the component of the normal velocity at those grid points near the interface. We use different methods to calculate the normal velocity inside and outside the Hele-Shaw bubble.

**5.1.1. Interpolating the normal velocity outside the Hele-Shaw cell.** At a regular grid point, which the standard five point stencil does not cut through, the

normal velocity can be computed from

$$u_n = -\beta p_n = -\beta \nabla p \cdot \vec{n} \tag{5.35}$$

where $n$ is the unit normal direction, and $\nabla p$ can be computed from the standard central difference scheme.

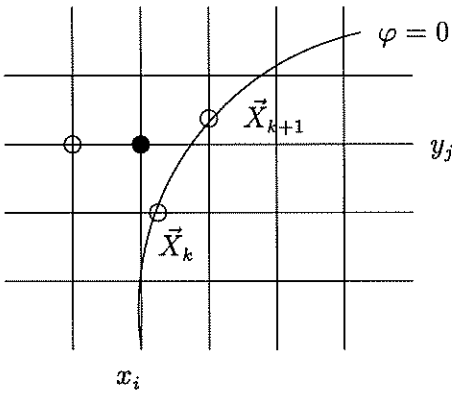


FIG. 3. *Diagram of an irregular grid point and the choice of three points for interpolating the normal velocity.*

At an irregular grid point, say, $(x_i, y_j)$ in Fig 3, care has to be taken to compute the $u_n$. If the grid point happens to be a control point $(X_k, Y_k)$, then we know the normal velocity from (5.34) already.

If the irregular grid point is not a control point, the following approach is used to find the normal velocity of the level set function at this grid point.

1. Find the two closest control points $\vec{X}_k$ and $\vec{X}_{k+1}$ from the irregular grid point $(x_i, y_j)$. We know the normal velocity at these two control points since we have solved for the pressure using the immersed interface method.
2. Find a regular grid point, say $(x_{i_0}, y_{j_0})$, which is close to $(x_i, y_j)$, $\vec{X}_k$, and $\vec{X}_{k+1}$ from the same side of the interface as the irregular grid point. The normal velocity at this point can be calculated using the central difference scheme from (5.35).
3. Interpolate the normal velocity at the three points to get an approximation at the irregular grid point $(x_i, y_j)$:

$$u_n(x_i, y_j) = \alpha_1 u_n(x_{i_0}, y_{j_0}) + \alpha_2 u_n(X_k, Y_k) + \alpha_3 u_n(X_{k+1}, Y_{k+1}) \tag{5.36}$$

where $\alpha_1$, $\alpha_2$, and $\alpha_3$ is the solution of the following linear system:

$$\begin{aligned} \alpha_1 + \alpha_2 + \alpha_3 &= 1 \\ \alpha_1 (x_i - x_{i_0}) + \alpha_2 (x_i - X_k) + \alpha_3 (x_i - X_{k+1}) &= 0 \\ \alpha_1 (y_j - y_{j_0}) + \alpha_2 (y_j - Y_k) + \alpha_3 (y_j - Y_{k+1}) &= 0 \end{aligned} \tag{5.37}$$

This is a second order interpolation formula.

**5.1.2. Extend the velocity inside the Hele-Shaw cell.** The algorithm described above for computing the normal velocity is successful for the irregular grid points outside of the Hele-Shaw cell, but does not work well for those grid points inside the Hele-Shaw cell. This can be explained roughly according to the following: Compared to the solution domain, the size of the Hele-Shaw cell is very small, especially at the beginning. The potential corresponding to a single point source or a small mass of sources is

$$p(r) \sim \frac{\log(r)}{2\pi}, \qquad r = \sqrt{x^2 + y^2}$$

assuming that the source center is at the origin. So the normal velocity $\beta^- p_n$ inside the Hele-Shaw bubble is nearly singular. On the other hand, from the non-flux condition

$$\beta^+ p_n^+ - \beta^- p_n^- = 0,$$

we also conclude that $|p_n^-|$ is very small compared to $p_n^+$ since $\beta^- \gg \beta^+$. Thus the normal derivative $p_n$ changes very rapidly inside the Hele-Shaw bubble which makes the computation very difficult even at regular grid points.

The solution to this difficulty is to extend the normal velocity from the information on the interface, which is known once we have solved for the pressure. A simple extension is

$$u_n(\vec{x}) = \begin{cases} \dfrac{u_n(\vec{x}_p)}{2} \left(1 + \cos(\dfrac{\pi\, d_{\vec{x},\vec{x}_p}}{\alpha h})\right), & \text{if } d_{\vec{x},\vec{x}_p} \le \alpha \\ 0 & \text{otherwise} \end{cases} \tag{5.38}$$

where $\vec{x}_p$ is the projection of $\vec{x}$ on the interface, $d_{\vec{x},\vec{x}_p}$ is the distance between $\vec{x}$ and $\vec{x}_p$. Usually $\alpha$ is about $4\,h \sim 6\,h$. Note that the Hele-Shaw bubble is expanding outwards and we only update the level set function inside the tube containing the interface. So the error from extending the normal velocity has only a small effect on the overall accuracy.

**5.2. Re-initialization and filtering.** Initially, the level set function $\varphi(x, y)$ is often chosen as the signed normal distance from the interface which satisfies $|\nabla\varphi| = 1$. However, while equation (1.7) will move the interface (the level set $\varphi = 0$) at the correct speed, $\varphi$ will no longer be a distance function after some time. As we mentioned in the introduction section, a re-initialization process is often necessary to keep $\varphi$ as the signed distance function to a certain accuracy.

An interesting and important effect of the re-initialization procedure is to stabilize the problem nicely. Since the nature of the Hele-Shaw flow is very unstable, a small perturbation or numerical noise might be amplified and then kill the computation. In tracking approaches such as [9], a linear filtering process is needed to eliminate the high frequency modes in the solution. In our method, this is automatically done through the re-initialization step. A minimum dissipation, high order, essentially non-oscillatory (ENO) scheme is used to preserve the sharp interface (corners and cusps) while the noise is successfully eliminated. In order to minimize the effect of the dissipation we do not do re-initialization at the interface at every time step. Instead we do the following.

After some time steps, for example, every 10−20 iterations, we do our re-initialization using equation (4.25) at the grid points $(i, j)$ in the $\epsilon_2$ tube if $|\phi(i, j)| > h$, where $h$ is the grid size. Thus the interface is neither moved nor smoothed and the level set function $\varphi$ is replaced by an approximation to the signed distance function in the new $\epsilon_2$ tube.

We also do our re-initialization at points in the $\epsilon_2$ tube after a long time period, for example, every 100 iterations. Now the noise at the interface is eliminated by the numerical dissipation.

We find the re-initialization step is very crucial in our problem. A proper choice of the re-initialization procedure improves our numerical results.

We also note that the level set procedure itself imposed a "topological" regularization on unstable problems, see [7, 8]

**5.3. A benchmark problem.** We can construct an exact solution for an axisymmetric case to check if our method works properly.

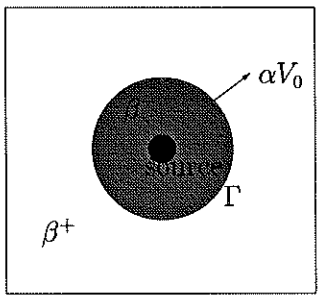


FIG. 4. An expanding interface with a constant speed $\alpha V_0$.

$$p(r) = \begin{cases} \frac{V_0}{\beta^-}\left(\frac{2r^3}{3\alpha^2} - \frac{3r^2}{2\alpha}\right) + C_1 & \text{if } 0 \le r \le \alpha \\ -\frac{V_0\,\alpha}{\beta^-}\log(r) + C_0 & \text{if } \alpha < r \le r_\Gamma \\ -\frac{V_0\,\alpha}{\beta^+}\log(r) & \text{if } r_\Gamma \le r, \end{cases} \tag{5.39}$$

where $r_\Gamma = \sqrt{2\alpha V_0 t + r_0^2}$, $r_0$ is the initial interface. Note: $[p_r] = 0$ at $r = \alpha$. $C_0$ is chosen as

$$C_0 = \frac{\tau}{r_\Gamma} + V_0\alpha\log(r_\Gamma)\left(\frac{1}{\beta^-} - \frac{1}{\beta^+}\right) \tag{5.40}$$

so that $[p]_{r_\Gamma} = \tau\kappa = -\tau/r_\Gamma$. Similarly $C_1$ is chosen such that $[p]_\alpha = 0$,

$$C_1 = C_0 - \frac{V_0\alpha\log(\alpha)}{\beta^-} + \frac{5V_0\alpha}{6\beta^-}. \tag{5.41}$$

The velocity is determined from (5.26). In polar coordinates, we have

$$u_r = -\beta\frac{\partial p}{\partial r} = \begin{cases} -V_0\left(\frac{2r^2}{\alpha^2} - \frac{3r}{\alpha}\right) & \text{if } 0 \le r \le \alpha, \\ \frac{\alpha V_0}{r} & \text{if } \alpha \le r \end{cases} \tag{5.42}$$

$$u_\theta = \frac{1}{r}\frac{\partial p}{\partial \theta} = 0. \tag{5.43}$$

We can get the Poisson equation for the pressure $p$

$$\begin{aligned} \nabla\left(\beta(x,y)\nabla p\right) &= \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial p}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 p}{\partial\theta^2} \\ &= \begin{cases} V_0\left(\frac{6r}{\alpha^2} - \frac{6r}{\alpha}\right) & \text{if } 0 \le r \le \alpha \\ 0 & \text{if } \alpha < r \end{cases} \\ &= -\nabla\cdot\vec{u} \end{aligned} \tag{5.44}$$

TABLE 1

*Grid refinement analysis in the infinity norm, $t_0 = 0$, $t_{out} = 0.1$*

| $n$ | $E_p$ | *rate* |
|---|---|---|
| 40 | $2.4230 \times 10^{-3}$ | |
| 80 | $7.8189 \times 10^{-4}$ | 1.6318 |
| 160 | $1.4923 \times 10^{-4}$ | 2.3894 |
| 320 | $3.7564 \times 10^{-5}$ | 1.9901 |

Our numerical computations show that, for a short period of time, we can obtain second order accuracy for the pressure $p$, the interface location $r_\Gamma$, and the velocity vector $(u, v)$. Table 1 shows the result of the grid refinement analysis for the pressure for a fixed time $t_{out} = 0.1$, where

$$E_p = \max_{ij} |p(x_i, y_j, t_{out}) - P_{ij}^n| ,$$

$p(x_i, y_j, t_{out})$ is the exact solution at time $t_{out}$, $P_{ij}^n$ is the computed solution at that time. Second order accuracy can be observed. For a longer time computation of this unstable flow, the truncation and round-off errors will have much more of an effect on the computation. See the next section for more details.

**5.4. Numerical experiments of Hele-Shaw flow.** [2] We have done a number of numerical experiments with different initial interfaces, viscosities, and surface tensions. All the results seem to agree with the theoretical analysis and numerical results in the literature. Since Hele-Shaw flow is unstable, for long time computation, the results do not converge to a unique solution due to the round-off and discretization errors. This consistent with experiments in which each one will give a different shape after some time. However this should not invalidate our simulations because we still can predict roughly the shape and the location of interface as time evolves. Moreover for a short time period, the solution does converge and the computational result is independent of the grid. The crucial parameter which affects the stability is the surface tension. The smaller surface tension, the more unstable is the Hele-Shaw flow. Below we present two typical examples.

[2] A short movie of the animation is available upon the request.

**Example 1.** The initial interface in the polar coordinates is:

$$\rho = 1 + 0.1\left(\sin(2\theta) + \cos(3\theta)\right), \qquad 0 \le \theta \le 2\pi,$$

(a) (b)

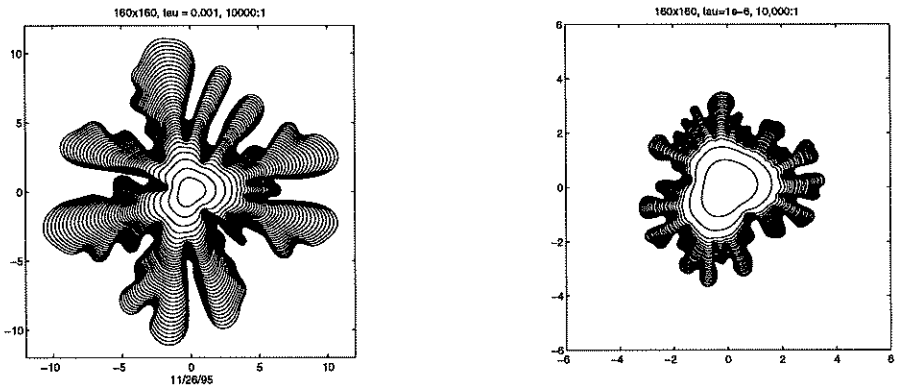


FIG. 5. *Expanding Hele-Shaw bubbles with different surface tension:* $(a)$ : $\tau = 0.001$; $(b)$ : $\tau = 10^{-6}$.

Fig 5 shows the expansion of a Hele-Shaw bubble with different surface tension at equally spaced time increments $\Delta t$. The viscosity ration is $10,000 : 1$. The simulation displays much of the behavior that has become known to the numerical analysts working on this subject. As we decrease the surface tension, the flow become more unstable and more tip splitting occurs as shown in Fig 5 (b). Note that as the interface becomes more complicated, we need to have finer grid to get higher resolution of the interface. That is the reason why we did not go as far in Fig 5 (b) as we did in Fig 5 (a). The envelop of the interface at late stages will approach a circle. For more detailed physical explanations of Hele-Shaw flow, see [4, 5, 6, 9, 21, 25, 31]. Typically, if we use a grid of 160 by 160, it takes less than one day to get the result shown in Fig 5 (a) on a IBM RS6000 machine using double precision computation. This is much faster than boundary integral methods. The main savings comes from the fast method for solving the elliptic interface problems.

**Example 2.** The initial interface in the polar coordinates now is:

$$\rho = 1 + 0.3\sin(3\theta), \quad 0 \le \theta \le 2\pi$$

So it is symmetric with $y$ axis.

Fig 6 shows computational results with two different surface tensions. With large surface tension, Fig 6 (a), the symmetry remains for quite a long time while Fig 6 (b) with smaller surface tension loses the symmetry at very early time.

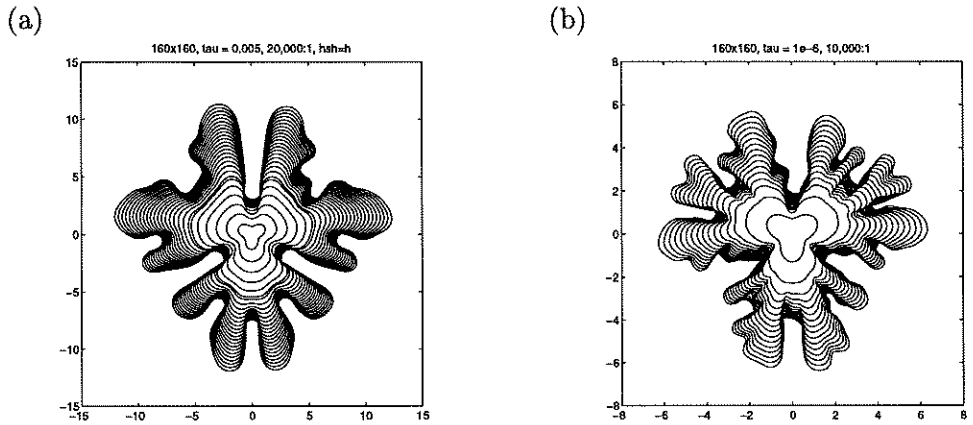


FIG. 6. *Expanding Hele-Shaw bubbles with different surface tension from a y-symmetric initial bubble:* $(a): \tau = 0.001$; $(b): \tau = 10^{-6}$.

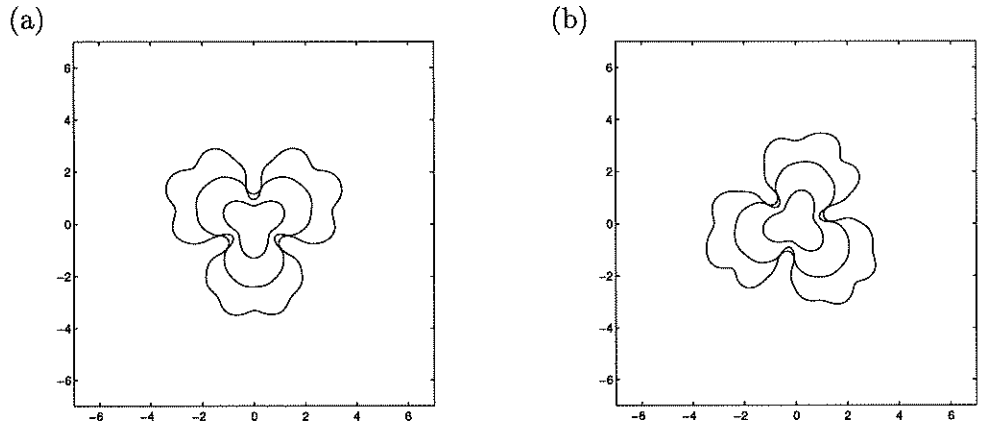


FIG. 7. *Comparison the results after the rotation of the interface. The innermost is the initial interface. The middle one is the interface after 1500 iterations. The outermost one is the interface after 2500 iteration and we begin to see a little difference between* $(a)$ *and* $(b)$.

Fig 7 shows a test in which we use the same parameters: viscosity ratio $10,000:1$, surface tension $\tau = 0.001$, but we rotate the interface $45^o$ degree counter-clockwise in Fig 7 (b). For a short time of period, that is after 1500 iterations, the results are almost identical if we rotate the result in Fig 7 (b) back. For a long period of time, that is after 2500 iterations, we can see some difference between the computation results. By the time, the error of the discretization and round-off are amplified to a significant level.

However, we do see some grid effects in the computation. The main source of such effect is from the Poisson solver. The standard five-point stencil is grid dependent. Such an effect is amplified for long time computation because the nature of the instability of the flow. The solution to this problem is to use a nine-point stencil or solve the

same Poisson problem with different grid orientations and then take the average of the solutions. We have tried the second approach which has effectively reduced the grid effect to a certain degree. However, since this is an unstable flow problem, it may not worth the effort to obtain a very accurate solution.

**6. Conclusions.** In this paper, we present a new numerical method which combines the immersed interface method with the level set formulation for moving interface problems. This method is second order accurate unless the interface develops singularities. Numerical experiments for Hele-Shaw flow shows the efficiency of this method. We believe that we are the first to use a finite difference method with a fixed grid to successfully simulate the Hele-Shaw flow.

We would like to thank Thomas Y. Hou and Randall J. LeVeque for very beneficial discussions about the Hele-Shaw flow and the treatment of the boundary condition. We would also like to thank Randall J. LeVeque for reading, correcting and commenting on this paper.

REFERENCES

[1] D. Adalsteinsson and J. Sethian. A fast level set method for propagating interfaces. Lawrence Berkeley Lab Report LBL-34893, 1993.

[2] L. M. Adams. A multigrid algorithm for immersed interface problems. Proceedings of Copper Mountain Multigrid Conference, to appear, 1995.

[3] Y.C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing method for incompressible fluid flows. *J. Comput. Phys.*, to appear.

[4] W. Dai, L. Kadanoff, and S. Zhou. Interface dynamics and the motion of complex singularities. *Phys. Rev.*, A 43:6672, 1991.

[5] W. Dai and M. J. Shelley. A numerical study of the effect of surface tension and noise on an expanding Hele-Shaw bubble. *Phys. Fluids.*, A 5 (9):2131–2146, 1993.

[6] A. J. Degregoria and L. W. Schwartz. A boundary-integral method for two-phase displacement in Hele-Shaw cells. *J. Fluid Mech.*, 164:383–400, 1986.

[7] E. Harabetian and S. Osher. Regulization of ill-posed problems via level set approach. *Submitted to SIAM J. Appl. Math.*, 1995.

[8] E. Harabetian, S. Osher, and C.W. Shu. An Eulerian approach for vortex motion. *J. Comput. Phys.*, to appear.

[9] T. Y. Hou, J. S. Lowengrub, and M. J. Shelley. Removing the stiffness from interfacial flows with surface tension. *J. Comput. Phys.*, 114:312–338, 1994.

[10] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Num. Anal.*, 31:1019–1044, 1994.

[11] R. J. LeVeque and Z. Li. Simulation of bubbles in creeping flow using the immersed interface method. in Proc. sixth international symposium on computational fluid dynamics, pp 688-693, 1995.

[12] R. J. LeVeque and C. Zhang. Immersed interface methods for wave equations with discontinuous coefficients. To appear, 1994.

[13] R. J. LeVeque and C. Zhang. Finite difference methods for wave equations with discontinuous coefficients. in Proc. 1995 ASCE Conference, S. Sture, ed., to appear, 1995.

[14] R.J. LeVeque and Z. Li. Immersed interface method for Stokes flow with elastic boundaries or surface tension. Tech. Report #95-01, University of Washington, *SIAM J. Sci. Stat. Compt.*, in press, 1995.

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

[16] Z. Li. A fast iterative algorithm for elliptic interface problems. UCLA CAM report #95-40, 1995.

[17] Z. Li. Immersed interface method for moving interface problems. Numerical Algorithms, to appear, 1995.

[18] Z. Li. A note on immersed interface methods for three dimensional elliptic equations. *Computers and Mathematics with Applications*, 31:9–17, 1996.

[19] Z. Li and A. Mayo. ADI methods for heat equations with discontinuties along an arbitrary interface. In *Proc. Symp. Appl. Math.* W. Gautschi, editor, volume 48, pages 311–315. AMS, 1993.

[20] A. Mayo. On the rapid evaluation of heat potentials on general regions. IBM Technical report 14305.

[21] E. Meiburg and G. Homsy. Nonlinear unstable viscous fingers in Hele-Shaw flows. ii. numerical simulation. *Phys. Fluids.*, 31 (3):429–439, 1988.

[22] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations.* Cambridge press, 1995.

[23] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

[24] S. Osher and C.-W. Shu. High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Num. Anal.*, 28:907–922, 1991.

[25] D. A. Reinelt. Interface conditions for two-phase displacement in Hele-Shaw cells. *J. Fluid Mech.*, 183:219–234, 1987.

[26] Y. Saad. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[27] P.G. Saffman and G. I. Taylor. The penetration of a fluid into a porous medium or Hele-Shaw cell containing a more viscous liquid. *Proc. R. Soc. Lond. A*, 245:312–329, 1958.

[28] M. Sussman. Ph.D thesis. UCLA CAM report #94-13, 1994.

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

[30] Paul N. Swarztrauber. Fast Poisson solver. In *Studies in Numerical Analysis,* G. H. Golub, editor, volume 24, pages 319–370. MAA, 1984.

[31] G. Tryggvason and H. Aref. Numerical experiments on Hele-Shaw flow with a sharp interface. *J. Fluid Mech.*, 136:1–20, 1983.

[32] N. Whitaker. Some numerical methods for the Hele-Shaw equations. *J. Comput. Phys.*, 111:81–88, 1994.

[33] H.-K. Zhao, J.-P. Shao, S. Osher, T. Chan, and B. Merriman. Motion of multiple junctions and interfaces in 3-d and applications to domain decomposition. preprint, 1995.