Sixth Mississippi State Conference on Differential Equations and Computational Simulations, *Electronic Journal of Differential Equations*, Conference 15 (2007), pp. 193–210.
ISSN: 1072-6691. URL: http://ejde.math.txstate.edu or http://ejde.math.unt.edu
ftp ejde.math.txstate.edu (login: ftp)

AN AUGMENTED IIM-LEVEL SET METHOD FOR STOKES EQUATIONS WITH DISCONTINUOUS VISCOSITY

ZHILIN LI, SHARON R. LUBKIN, XIAOHAI WAN

Abstract. A finite difference method is proposed for incompressible Stokes equations with discontinuous viscosity. The method combines the augmented immersed interface method with a level set representation of the interface on a uniform Cartesian grid. In the proposed method, augmented interface variables are introduced along the interface and be solved by the GMRES iterative method. The augmented approach allows fast Poisson solvers to be used and therefore is very efficient. Numerical examples including two moving interface problems are presented.

1. Introduction

In this paper, we present a numerical method for solving the stationary incompressible Stokes equations for two-phase Newtonian flows with an arbitrary interface. The governing equations have the following form

$$\nabla p = \nabla \cdot \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) + \mathbf{F}(\mathbf{x}) + \mathbf{g}(\mathbf{x}), \tag{1.1}$$

with the incompressibility condition

$$\nabla \cdot \mathbf{u} = 0, \tag{1.2}$$

where $\mathbf{u} = (u, v)^T$ is the fluid velocity, p is the fluid pressure, μ is the fluid viscosity, $\mathbf{x} = (x, y)$ is the Cartesian coordinate variable, $\mathbf{g} = (g_1, g_2)^T$ is a body force such as gravity, and $\mathbf{F}$ is a source which can have a Dirac delta function singularity,

$$\mathbf{F}(\mathbf{x}) = \int_\Gamma \mathbf{f}(s)\delta(\mathbf{x} - \mathbf{X}(s))ds \tag{1.3}$$

where Γ is a smooth interface in the solution domain with the arc-length parameter s, $\mathbf{f} = (f_1, f_2)^T$ is the source density, and $\delta(\cdot)$ is the Dirac delta function defined in the distribution sense. The body force term $\mathbf{g}$ may also have a finite jump across the interface Γ as well.

The interface Γ divides the solution domain Ω into two parts Ω^+ and Ω^- with $\Omega = \Omega^+ \cup \Omega^-$. If the interface is closed, we use Ω^+ to express the exterior domain of the interface.

2000 *Mathematics Subject Classification.* 65M06, 65M12, 76T05.
Key words and phrases. Incompressible Stokes equations; discontinuous viscosity; interface problem; immersed interface method; level set method; fast Poisson solver, GMRES.

Published February 28, 2007.

We assume that the viscosity coefficient μ is piecewise constant across the interface:

$$\mu(\mathbf{x}) = \begin{cases} \mu^+ & \text{if } \mathbf{x} \in \Omega^+, \\ \mu^- & \text{if } \mathbf{x} \in \Omega^-, \end{cases} \tag{1.4}$$

where μ^+ and μ^- are two positive constants.

In this paper, we propose a finite difference method coupled with a level set function representation of the interface on a uniform Cartesian grid to solve the problem. Finite element and boundary integral methods for this problem can be found in the literature. These methods may be more efficient if only the solution on the interface is solved. In our method, we solve for the velocity in the entire domain.

There are at least two difficulties in solving (1.1)-(1.4) numerically using finite difference methods. The first one is to deal with the singular force term. A simple way is to use Peskin's discrete delta function approach [24] to distribute the singular force to nearby grid points. Such a discretization is typically first order accurate and will smooth out the solution. The second difficulty is how to deal with the discontinuous viscosity. A simple smoothing method may introduce large errors; see [12] for a one-dimensional example.

In the case of a continuous viscosity with a Dirac delta function source distribution along an interface, various methods have been developed. We refer the readers to [2, 5, 10, 19, 32] for various methods and the references therein. The difficulty with a discontinuous viscosity is that the jump conditions for the pressure and velocity are coupled together (section 3), which makes it difficult to discretize the system accurately. Approximations of the jump conditions have been used to decouple the system [16] and have been productively used to simulate problems in biological fluid dynamics [18].

The augmented immersed interface method for incompressible 2D Stokes flows with discontinuous viscosity has been developed [15] with the interface represented by a cubic spline interpolation, a variation of the particle approach. It is easy with splines to obtain the surface derivatives of jumps of the material variables (pressure and velocity) since they can be expressed as the functions of the arc-length. However, the spline approach is difficult to use for multi-connected domains, moving interface problems with topological changes such as merging and splitting, and for three dimensional (3D) problems. In addition, the level set method usually has better stability. For a spline approach, re-parameterization, filtering, and re-griding may be needed at every or every other time steps. All these reasons favor a level set approach over a spline approach. Nevertheless, it is not trivial to compute surface derivatives of an interface quantity from the level set method since the interface is explicitly defined only at grid points.

2. The Stokes solver: Three Poisson equations approach

Instead of writing the singular force as a source term for the Stokes equations (1.1), we can solve the Stokes equations in the form

$$\nabla p = \nabla \cdot \mu^+ \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) + \mathbf{g}(\mathbf{x}), \quad \mathbf{x} \in \Omega^+ \tag{2.1}$$

and

$$\nabla p = \nabla \cdot \mu^- \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) + \mathbf{g}(\mathbf{x}), \quad \mathbf{x} \in \Omega^-. \tag{2.2}$$

We define viscosity ratio $\lambda = \mu^-/\mu^+$. If Γ is closed and $\lambda > 1$, then the fluid phase inside the interface is more viscous and is called a drop; if Γ is closed and $\lambda < 1$, then the fluid phase inside the interface is less viscous than the ambient fluid and is called a bubble (see for example [26]); if $\lambda = 1$ the fluids inside and outside have equal viscosity.

The solutions to (2.1)-(2.2) are coupled by the jump conditions which will be stated in Section 3. If we apply the divergence operator to both sides of the momentum equation (2.1, 2.2) and make use of the incompressibility condition(1.2) in each fluid, we get

$$\Delta p = \nabla \cdot \mathbf{g}, \tag{2.3}$$

where Δ is the Laplacian. This is a Poisson equation for the pressure p. Once p is solved from the equation above, we can rewrite the Stokes equation (2.1)-(2.2) in the form [1]

$$\Delta \tilde{u} = \frac{\partial p}{\partial x} - g_1, \tag{2.4}$$

$$\Delta \tilde{v} = \frac{\partial p}{\partial y} - g_2, \tag{2.5}$$

where

$$\tilde{u}(\mathbf{x}) = \begin{cases} \mu^+ u(\mathbf{x}) & \mathbf{x} \in \Omega^+ \\ \mu^- u(\mathbf{x}) & \mathbf{x} \in \Omega^-, \end{cases} \qquad \tilde{v}(\mathbf{x}) = \begin{cases} \mu^+ v(\mathbf{x}) & \mathbf{x} \in \Omega^+ \\ \mu^- v(\mathbf{x}) & \mathbf{x} \in \Omega^-. \end{cases} \tag{2.6}$$

We obtain $\mathbf{u} = (u, v)$ by solving $\tilde{\mathbf{u}} = (\tilde{u}, \tilde{v})$ from the Poisson equations above. The three decoupled Poisson equations (2.3)-(2.5) each can be solved efficiently using a fast Poisson solver if the solutions are smooth across the interface.

3. Jump conditions

It is well known that if there are singular interface forces, then the pressure p is discontinuous, and the velocity $\mathbf{u}$ is non-smooth across the interface. If the viscosity is discontinuous, that is $\lambda \neq 1$, then $\tilde{\mathbf{u}}$ is also discontinuous. Due to these discontinuities, the direct application of three Poisson solvers without modifications will not give the correct solution. It is critical to incorporate the jump conditions in the finite difference schemes so that the solutions in different phases can be coupled together correctly across the interface. The jump condition on the pressure p at point $\mathbf{x}^*$ on the interface Γ is defined as

$$[p](\mathbf{x}^*) = \Big(\lim_{\mathbf{x} \to (\mathbf{x}^*)^+} p(\mathbf{x}) \Big) - \Big(\lim_{\mathbf{x} \to (\mathbf{x}^*)^-} p(\mathbf{x}) \Big). \tag{3.1}$$

Other jump conditions are defined similarly. Jump conditions on each variable can be derived either from the physics of the problem itself or from the partial differential equations. The derivations of the jump conditions for the Stokes equation are described elsewhere [15]. Here we just list the results. Define

$$\mathbf{q} = (q_1, q_2)^T = ([\tilde{u}], [\tilde{v}])^T. \tag{3.2}$$

Then the solutions of the Stokes flow are the following three Poisson equations:

$$\begin{cases} \Delta p = \nabla \cdot \mathbf{g}, \\ [p] = \hat{f}_1 - 2\dfrac{\partial}{\partial \tau}\left(\mathbf{q} \cdot \tau\left(\mathbf{x}^*\right)\right), \\ \left[\dfrac{\partial p}{\partial \mathbf{n}}\right] = \dfrac{\partial \hat{f}_2}{\partial \tau} + 2\dfrac{\partial^2}{\partial \tau^2}\left(\mathbf{q} \cdot \mathbf{n}\left(\mathbf{x}^*\right)\right) - 4\kappa\dfrac{\partial}{\partial \tau}(\mathbf{q} \cdot \tau\left(\mathbf{x}^*\right)) + [\mathbf{g} \cdot \mathbf{n}], \end{cases} \tag{3.3}$$

$$\begin{cases} \Delta \tilde{u} = p_x - g_1, \\ [\tilde{u}] = q_1, \\ \left[\dfrac{\partial \tilde{u}}{\partial \mathbf{n}}\right] = \left(\hat{f}_2 + \dfrac{\partial}{\partial \tau}\left(\mathbf{q} \cdot \mathbf{n}\left(\mathbf{x}^*\right)\right)\right) \sin\theta - \left(\dfrac{\partial}{\partial \tau}\left(\mathbf{q} \cdot \tau\left(\mathbf{x}^*\right)\right)\right)\cos\theta, \end{cases} \tag{3.4}$$

$$\begin{cases} \Delta \tilde{v} = p_y - g_2, \\ [\tilde{v}] = q_2, \\ \left[\dfrac{\partial \tilde{v}}{\partial \mathbf{n}}\right] = -\left(\hat{f}_2 + \dfrac{\partial}{\partial \tau}\left(\mathbf{q} \cdot \mathbf{n}\left(\mathbf{x}^*\right)\right)\right) \cos\theta - \left(\dfrac{\partial}{\partial \tau}\left(\mathbf{q} \cdot \tau\left(\mathbf{x}^*\right)\right)\right)\sin\theta, \end{cases} \tag{3.5}$$

$$\left[\frac{\tilde{u}}{\mu}\right] = 0, \quad \left[\frac{\tilde{v}}{\mu}\right] = 0. \tag{3.6}$$

In the jump conditions above, $\mathbf{n} = (\cos\theta, \sin\theta)^T$ and $\tau = (-\sin\theta, \cos\theta)^T$ are unit outward normal and tangential directions respectively, θ is the angle between the x-axis and the normal directions, $\hat{f}_1 = \mathbf{f} \cdot \mathbf{n}$ and $\hat{f}_2 = \mathbf{f} \cdot \tau$ are the force components in the normal and tangential directions. κ is interface curvature. Unit circle is defined to have constant curvature -1. The vector $\mathbf{q}$ is called the augmented variable which is unknown unless we know the solution $(u, v)^T$. However, if we know $\mathbf{q}$, then the jump conditions for the pressure and the velocity are decoupled in terms of $\mathbf{q}$ and its surface derivatives. In this paper, we assume that we have a periodic boundary for all the variables.

The main idea of the augmented approach is to solve $\mathbf{q}$ such that $(\tilde{u}, \tilde{v})^T$ and p satisfy the Stokes equations, the jump conditions above, and the kinematic condition

$$[u] = [v] = 0. \tag{3.7}$$

Then $(u, v)^T$ and p must be the solution to the original problem.

The jump conditions above enable us to use the immersed interface method (IIM) to solve the three Poisson equations (2.3)-(2.5) if a guess $\mathbf{q}$ is given. The generalized minimum residual (GMRES) method [28] is used to solve for $\mathbf{q}$ so that the kinematic condition can be satisfied.

4. The numerical algorithm

4.1. **Level set representation of interfaces and related quantities.** We use the zero level set of a two dimensional function $\varphi(x, y)$ to represent the interface. That is, $\varphi(x, y)$ is such a function that

$$\Gamma = \{(x, y) : \varphi(x, y) = 0\}. \tag{4.1}$$

Generally $\varphi(x, y)$ is chosen as the signed distance function from the interface. We denote $\Omega^+ = \{(x, y), \mid \varphi(x, y) \geq 0\}$ and $\Omega^- = \{(x, y), \mid \varphi(x, y) < 0\}$. For simplicity, we assume that the domain Ω is a rectangle $[a,\ b] \times [c,\ d]$. The solution domain is discretized using a uniform Cartesian grid

$$\begin{aligned} x_i &= a + ih_x, \quad i = 0, 1, \dots, m, \quad h_x = \frac{b-a}{m}; \\ y_j &= c + jh_y, \quad j = 0, 1, \dots, n, \quad h_y = \frac{d-c}{n}. \end{aligned} \tag{4.2}$$

In the rest of the paper, we take $h_x = h_y = h$ to simplify the notation. The level set function $\varphi_{i,j} = \varphi(x_i, y_j)$ is defined at grid points. A grid point (x_i, y_j) is an *irregular inner grid point* if

$$\varphi_{i,j} < 0 \tag{4.3}$$

and any of the following four inequalities is true:

$$\begin{aligned} &\varphi_{i,j} \cdot \varphi_{i-1,j} \le 0, \quad \varphi_{i,j} \cdot \varphi_{i+1,j} \le 0, \\ &\varphi_{i,j} \cdot \varphi_{i,j-1} \le 0, \quad \varphi_{i,j} \cdot \varphi_{i,j+1} \le 0. \end{aligned} \tag{4.4}$$

Similarly, a grid point (x_i, y_j) is an *irregular outer grid point* if

$$\varphi_{i,j} \ge 0 \tag{4.5}$$

and any of the following four inequalities is true:

$$\begin{aligned} &\varphi_{i,j} \cdot \varphi_{i-1,j} < 0, \quad \varphi_{i,j} \cdot \varphi_{i+1,j} < 0, \\ &\varphi_{i,j} \cdot \varphi_{i,j-1} < 0, \quad \varphi_{i,j} \cdot \varphi_{i,j+1} < 0. \end{aligned} \tag{4.6}$$

For a given irregular grid point $\mathbf{x} = (x_i, y_j)$ near a sufficiently flat section of interface, there is a corresponding orthogonal projection point $\mathbf{x}^* = (x^*, y^*)$ on the interface satisfying

$$\mathbf{x}^* = \mathbf{x} + a\mathbf{n}(\mathbf{x}) \tag{4.7}$$

where a is a real unknown variable and $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$ is the unit outward normal direction at $\mathbf{x}$. See Fig. 1 for an illustration.

Since $\varphi(\mathbf{x}^*) = 0$, we have

$$0 = \varphi(\mathbf{x} + a\mathbf{n}) \approx \varphi(\mathbf{x}) + |\nabla\varphi| a + \frac{1}{2}(\mathbf{n}^{\mathsf{T}}\mathbf{H}(\varphi)\mathbf{n})a^2 \tag{4.8}$$

where the Hessian matrix $\mathbf{H}$ is defined as

$$\mathbf{H}(\varphi) = \begin{pmatrix} \frac{\partial^2\varphi}{\partial x^2} & \frac{\partial^2\varphi}{\partial x\partial y} \\ \frac{\partial^2\varphi}{\partial y\partial x} & \frac{\partial^2\varphi}{\partial y^2} \end{pmatrix}. \tag{4.9}$$

The derivatives are approximated using the central finite difference schemes:

$$\frac{\partial\varphi}{\partial x} \approx \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2h}, \tag{4.10}$$

$$\frac{\partial\varphi}{\partial y} \approx \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2h}, \tag{4.11}$$

$$\frac{\partial^2\varphi}{\partial x^2} \approx \frac{\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}}{h^2}, \tag{4.12}$$

$$\frac{\partial^2\varphi}{\partial y^2} \approx \frac{\varphi_{i,j+1} - 2\varphi_{i,j} + \varphi_{i,j-1}}{h^2}, \tag{4.13}$$

$$\frac{\partial^2\varphi}{\partial x\partial y} = \frac{\partial^2\varphi}{\partial y\partial x} \approx \frac{\varphi_{i+1,j+1} + \varphi_{i-1,j-1} - \varphi_{i+1,j-1} - \varphi_{i-1,j+1}}{4h^2}. \tag{4.14}$$

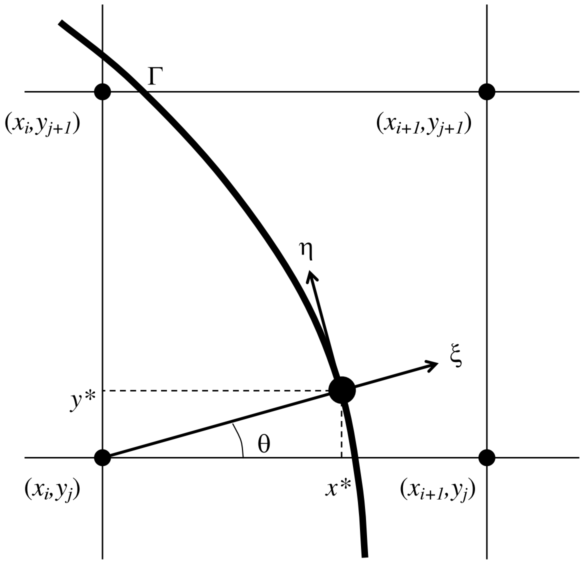

FIGURE 1. An irregular grid point $\mathbf{x} = (x_i, y_j)$ and its normal projection point $\mathbf{x}^* = (x^*, y^*)$ on the interface Γ. (ξ, η) is the local coordinate system at $\mathbf{x}^*$.

At each orthogonal projection $\mathbf{x}^*$, the geometrical information needed includes the curvature κ and the angle θ between the normal direction and the x-axis at $\mathbf{x}^*$. The curvature is defined as usual

$$\kappa = -\frac{\frac{\partial^2\varphi}{\partial x^2}(\frac{\partial\varphi}{\partial y})^2 - 2\frac{\partial^2\varphi}{\partial x\partial y}\frac{\partial\varphi}{\partial x}\frac{\partial\varphi}{\partial y} + \frac{\partial^2\varphi}{\partial y^2}(\frac{\partial\varphi}{\partial x})^2}{\Big((\frac{\partial\varphi}{\partial x})^2 + (\frac{\partial\varphi}{\partial y})^2\Big)^{\frac{3}{2}}}. \tag{4.15}$$

The unit normal direction is determined by

$$\cos\theta = \frac{\frac{\partial\varphi}{\partial x}}{\sqrt{(\frac{\partial\varphi}{\partial x})^2 + (\frac{\partial\varphi}{\partial y})^2}}, \quad \sin\theta = \frac{\frac{\partial\varphi}{\partial y}}{\sqrt{(\frac{\partial\varphi}{\partial x})^2 + (\frac{\partial\varphi}{\partial y})^2}}. \tag{4.16}$$

The derivatives at $\mathbf{x}^*$ can be approximated using the bilinear interpolation formula from the derivatives already calculated at grid points using (4.10)-(4.14) as following. Given an interface point $\mathbf{x}^*$ and a grid function $f(\cdot)$, we label the four grid points surrounding $\mathbf{x}^*$ as

$$(x_i, y_j), (x_{i+1}, y_j), (x_i, y_{j+1}), (x_{i+1}, y_{j+1}). \tag{4.17}$$

See Fig. 1 for an example. If we define

$$a = \frac{2(x^* - x_i)}{h} - 1, \quad b = \frac{2(y^* - y_j)}{h} - 1, \tag{4.18}$$

then we can use the bilinear interpolation scheme

$$\begin{aligned}&\big\{(1-a)(1-b)f_{i,j} + (1+a)(1-b)f_{i+1,j}\\ &+ (1-a)(1+b)f_{i,j+1} + (1+a)(1+b)f_{i+1,j+1}\big\}/4\end{aligned} \tag{4.19}$$

to approximate $f(\mathbf{x}^*)$. We can calculate κ, $\cos\theta$ and $\sin\theta$ at the orthogonal projection point $\mathbf{x}^*$ if we set $f = \kappa(\mathbf{x})$, $f = \cos\theta$, and $f = \sin\theta$ respectively using the interpolated formula above. Note that these quantities are poorly conditioned if $|\nabla\varphi| \sim 0$. However our choice of φ as the signed distance function avoids this difficulty.

4.2. **The immersed interface method for elliptic interface problems.** Given an intermediate approximation of $\mathbf{q}$, the solution to the Stokes equations can be obtained by solving the three Poisson equations (2.3)-(2.5) with known jump conditions in the solution and the flux:

$$\begin{gathered}(\Psi_x)_x + (\Psi_y)_y = R(x, y)\\ [\Psi] = w_1, \quad [\Psi_n] = w_2.\end{gathered} \tag{4.20}$$

using the immersed interface method [9, 13, 14]. The finite difference scheme using the immersed interface method can be simply written as

$$\frac{\Psi_{i-1,j} + \Psi_{i+1,j} + \Psi_{i,j-1} + \Psi_{i,j+1} - 4\Psi_{ij}}{h^2} = R_{ij} + C_{ij} \tag{4.21}$$

where $C_{ij} = 0$ at regular grid points. At an irregular grid point, C_{ij} depends on the interface curvature, first and second order surface derivatives of w_1 and w_2 (see [11] for its formulation and derivation). The immersed interface method not only gives a second order accurate solution, but also enables us to use widely available fast Poisson solvers to solve the discrete systems of equations. The fast Poisson solver, for example the one from Fishpack [23], only requires $O(N\log(N))$ operations, where N is the total number of grid points. Therefore the Stokes equations with given $\mathbf{q}$ can be solved in $O(3N\log(N))$ operations, which corresponds to the cost of one GMRES iteration.

4.3. **Computing an interface quantity and its tangential derivatives using the least squares interpolation scheme.** To determine the correction term C_{ij} we need to know the jumps $[\Psi] = w_1(s)$, $[\frac{\partial\Psi}{\partial\mathbf{n}}] = w_2(s)$ and the tangential derivatives w_1', w_1'' and w_2', where s is the arc-length parameterization of the interface [11]. The problem may be formulated thus: given the values of an interface function $\phi(s)$ at a set of interface points $s = s_1, \dots, s = s_n$, find the values of $\phi(s^*)$, $\phi'(s^*)$ and $\phi''(s^*)$. It is critical to choose an interpolation scheme for the immersed interface-level set method to work well.

We assume that the values of the interface function $\phi(s)$ are known at all the projection points from the inner irregular grid points and we want to interpolate $\phi(s)$ at the projection points from the outer irregular grid points, and $\phi'(s)$ and $\phi''(s)$ at the projection points from both inner and outer irregular grid points. See Fig. 2 for an illustration.

The least squares interpolation scheme for approximating $\phi(s^*)$ (or $\phi'(s^*)$ and $\phi''(s^*)$) can be written as

$$\sum_i \gamma_i \phi(s_i), \tag{4.22}$$

where γ_i's are the coefficients that need to be determined from the Taylor expansion at the interpolation point s^*

$$\phi(s_i) = \phi(s^*) + \phi'(s^*)(s_i - s^*) + \frac{1}{2}\phi''(s^*)(s_i - s^*)^2 + \dots. \tag{4.23}$$

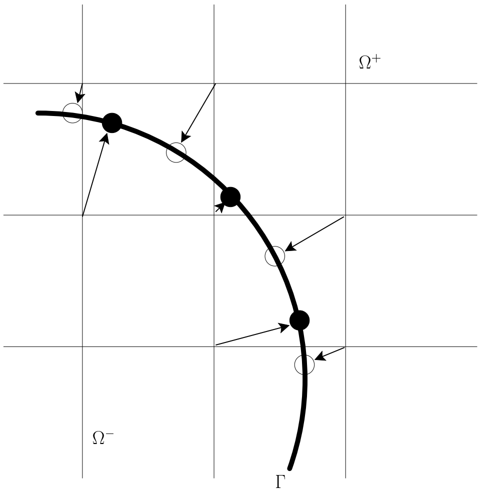

FIGURE 2. On the interface Γ, the values of the interface function $\phi(s)$ at the projection points from the inner irregular grid points (dark circles) can be used to interpolate both the values of $\phi(s)$ at the projection points from the outer irregular grid points (white circles) and the values of $\phi'(s)$ and $\phi''(s)$ at all the projection points.

We define

$$a_1 = \sum_i \gamma_i, \quad a_2 = \sum_i (s_i - s^*)\gamma_i \quad a_3 = \sum_i \frac{1}{2}(s_i - s^*)^2\gamma_i, \tag{4.24}$$

where $(s_i - s^*)$ is the signed arc length from the interface point at $s = s_i$ to the point at $s = s^*$(see [17] for its formula using the Hermite cubic spline approximation method). Substituting (4.23) into (4.22), we obtain the coefficients γ_i from setting

$$a_1 = 1, \quad a_2 = 0, \quad a_3 = 0. \tag{4.25}$$

If we choose more than three interface points for the interpolation, then the linear system of equations (4.25) is under-determined, that is, there are an infinite number of solutions. We use the singular value decomposition (SVD) method to solve the system of equations (4.25) to get the unique solution that has the least 2-norm among all feasible solutions.

For a given $s = s^*$, the interpolation points are chosen as a few orthogonal projections of inner grid points ($\varphi_{i,j} < 0$) that are the first few closest to the point $(X(s^*), Y(s^*))$. The criterion is the arc length distance such that $|s - s^*| \le \alpha$, where α is a parameter. We choose $\alpha = 6.5h$.

4.4. **Augmented immersed interface method for the Stokes equations.** The augmented unknown $\mathbf{q}$ needs to be solved only at orthogonal projection points from the inner irregular grid points because we can use interface interpolation to get the values at outer projection points.

Given a discrete approximation of (q_1, q_2) at $\{\mathbf{X}_k\}$, we can solve the first three equations (3.3)-(3.5) to get an approximate solution: the pressure $\mathbf{P}(\mathbf{Q})$, the scaled velocity $\tilde{\mathbf{U}}(\mathbf{Q})$ and $\tilde{\mathbf{V}}(\mathbf{Q})$. Generally the computed velocity $(\tilde{\mathbf{U}}(\mathbf{Q}), \tilde{\mathbf{V}}(\mathbf{Q}))$ does not satisfy the two augmented equations in (3.6), that is, $(\mathbf{U}, \mathbf{V}) = (\tilde{\mathbf{U}}/\mu, \tilde{\mathbf{V}}/\mu)$ may not be continuous across the interface.

Let us concatenate the discrete solutions $\{P_{ij}\}$, $\{U_{ij}\}$, and $\{V_{ij}\}$ as a vector $\tilde{\mathcal{U}}$ whose dimension is $O(3MN)$, where M and N are the number of grid lines in the x and y direction respectively. We denote also the vector of the discrete values of (q_1, q_2) at the control points $\{\mathbf{X}_k\}$ (consisted of projection points only from the inner irregular grid points) by $\mathbf{Q}$ whose dimension is $O(2N_b)$. Then the discrete solution of (3.3)-(3.5) given $\mathbf{Q}$ can be written as

$$A\tilde{\mathcal{U}} + B\mathbf{Q} = \mathbf{F}_1 \tag{4.26}$$

for some vector $\mathbf{F}_1$ and sparse matrices A and B. It requires solving three Poisson equations with different source terms and jump conditions to get $\tilde{\mathcal{U}}$.

Once we know the solution $\tilde{\mathcal{U}}$ given $\mathbf{Q}$, we can use $(\tilde{\mathbf{U}}, \tilde{\mathbf{V}})$ and the jump conditions $[\tilde{\mathbf{U}}_{\mathbf{n}}]$ and $[\tilde{\mathbf{V}}_{\mathbf{n}}]$ which also depend on $\mathbf{Q}$, to get $[\mathbf{U}(\mathbf{Q})] = [\tilde{\mathbf{U}}(\mathbf{Q})/\mu]$ and $[\mathbf{V}(\mathbf{Q})] = [\tilde{\mathbf{V}}(\mathbf{Q})/\mu]$ at those control points $\{\mathbf{X}_k\}$, $1 \le k \le N_b$. If both $\|\,[\mathbf{U}(\mathbf{Q})]\,\|$ and $\|\,[\mathbf{V}(\mathbf{Q})]\,\|$ are smaller than a given tolerance, then the method has already converged and the set of $\mathbf{Q}$, $\tilde{\mathbf{U}}/\mu$, $\tilde{\mathbf{V}}/\mu$ is an approximate solution. The interpolation scheme to get $[\tilde{\mathbf{U}}(\mathbf{Q})/\mu]$ and $[\tilde{\mathbf{V}}(\mathbf{Q})/\mu]$, which will be explained in the next sub-section, depends on $\tilde{\mathcal{U}}, \mathbf{Q}$ *linearly*. Therefore we can write

$$\Big([\tilde{\mathbf{U}}/\mu],\ [\tilde{\mathbf{V}}/\mu]\Big)^T = S\tilde{\mathcal{U}} + E\mathbf{Q} - \mathbf{F}_2, \tag{4.27}$$

where S and E are two sparse matrices, and $\mathbf{F}_2$ is a vector. The matrices depend on the interpolation scheme but do not need to be actually constructed in the algorithm. We need to choose such a vector $\mathbf{Q}$ that the continuity condition for the velocity is satisfied along the interface Γ. If we put the two matrix-vector equations (4.26) and (4.27) together we get

$$\begin{bmatrix} A & B \\ S & E \end{bmatrix} \begin{bmatrix} \tilde{\mathcal{U}} \\ \mathbf{Q} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{bmatrix}. \tag{4.28}$$

Note that $\mathbf{Q}$ is defined only on a set of points $\{\mathbf{X}_k\}$ on the interface while $\tilde{\mathcal{U}}$ is defined at all grid points. The Schur complement for $\mathbf{Q}$ is

$$(E - SA^{-1}B)\mathbf{Q} = \mathbf{F}_2 - SA^{-1}\mathbf{F}_1 = \bar{\mathbf{F}}. \tag{4.29}$$

If we can solve the system above to get $\mathbf{Q}$, then we can get $\tilde{\mathcal{U}}$ easily. Because the dimension of $\mathbf{Q}$ is much smaller than that of $\tilde{\mathcal{U}}$, we expect to get a reasonably fast algorithm for the two-phase Stokes equations if we can solve (4.29) efficiently.

In implementation, the GMRES iterative method [28] is used to solve (4.29). The GMRES method only requires matrix-vector multiplication. We explain below how to evaluate the right hand side $\bar{\mathbf{F}}$ of the Schur complement, and how to evaluate the matrix-vector multiplication needed by the GMRES iteration. We can see why we do not need to form the coefficient matrix $E - SA^{-1}B$ explicitly.

4.4.1. *Evaluation of the right hand side of the Schur complement.* First we set $\mathbf{Q} = \mathbf{0}$ and solve the de-coupled system (3.3)-(3.5), or (4.26) in the discrete form, to get $\tilde{\mathcal{U}}(\mathbf{0})$ which is $A^{-1}\mathbf{F}_1$ from (4.26). From the interpolation (4.27), we also have

$$\big([\mathbf{U}(\mathbf{0})], [\mathbf{V}(\mathbf{0})]\big)^T = S\tilde{\mathcal{U}}(\mathbf{0}) + E\,\mathbf{0} - \mathbf{F}_2 = S\tilde{\mathcal{U}}(\mathbf{0}) - \mathbf{F}_2.$$

Note that the residual of the Schur complement for $\mathbf{Q} = \mathbf{0}$ is

$$\begin{aligned} R(\mathbf{0}) &= (E - SA^{-1}B)\,\mathbf{0} - \bar{\mathbf{F}} = -\bar{\mathbf{F}} \\ &= -\left(\mathbf{F}_2 - SA^{-1}\mathbf{F}_1\right) = -\mathbf{F}_2 + S\tilde{\mathcal{U}}(\mathbf{0}) \\ &= \left([\mathbf{U}(\mathbf{0})]\,, [\mathbf{V}(\mathbf{0})]\right)^T, \end{aligned} \tag{4.30}$$

which gives the right hand side of the Schur complement system with an opposite sign.

4.4.2. *Evaluation of the matrix-vector multiplication.* The matrix-vector multiplication of the Schur complement system given $\mathbf{Q}$ is obtained from the following two steps:

- **Step 1:** Solve the coupled system (3.3)-(3.5), or (4.26) in the discrete form, to get $\tilde{\mathcal{U}}(\mathbf{Q})$.
- **Step 2:** Interpolate $\tilde{\mathcal{U}}(\mathbf{Q})$ using (4.27) to get $\left([\mathbf{U}(\mathbf{0})]\,, [\mathbf{V}(\mathbf{0})]\right)^T$. Then the matrix vector multiplication is

$$\left(E - SA^{-1}B\right)\mathbf{Q} = \left([\mathbf{U}(\mathbf{Q})]\,, [\mathbf{V}(\mathbf{Q})]\right)^T - \left([\mathbf{U}(\mathbf{0})]\,, [\mathbf{V}(\mathbf{0})]\right)^T. \tag{4.31}$$

This is because

$$\begin{aligned} \left(E - SA^{-1}B\right)\mathbf{Q} &= E\mathbf{Q} - SA^{-1}B\mathbf{Q} \\ &= E\mathbf{Q} - S\left(A^{-1}\mathbf{F}_1 - \tilde{\mathcal{U}}(\mathbf{Q})\right) \quad \text{(from (4.26))}\,, \\ &= E\mathbf{Q} + S\tilde{\mathcal{U}}(\mathbf{Q}) - \mathbf{F}_2 + \mathbf{F}_2 - SA^{-1}\mathbf{F}_1 \\ &= \left([\mathbf{U}(\mathbf{Q})], [\mathbf{V}(\mathbf{Q})]\right)^T + \bar{\mathbf{F}} \quad \text{(from (4.27))}, \\ &= \left([\mathbf{U}(\mathbf{Q})], [\mathbf{V}(\mathbf{Q})]\right)^T - \left([\mathbf{U}(\mathbf{0})], [\mathbf{V}(\mathbf{0})]\right)^T, \quad \text{(from (4.30))}. \end{aligned}$$

Now we can see that a matrix-vector multiplication is equivalent to solving the coupled system (3.3)-(3.5), or (4.26) in the discrete form, to get $\tilde{\mathcal{U}}$, and using an interpolation scheme (4.27) to get $([\mathbf{U}], [\mathbf{V}])^T$ at the control points.

Since we know the right hand side of the linear system of equations and the matrix-vector multiplication of the coefficient matrix, it is straightforward to use GMRES or other iterative methods.

In summary, the system of equations for $\mathbf{Q}$ can be written as

$$G\mathbf{Q} = \mathbf{b}, \tag{4.32}$$

where $\mathbf{b} = -\begin{pmatrix} [\mathbf{U}(\mathbf{0})] \\ [\mathbf{V}(\mathbf{0})] \end{pmatrix}$. The coefficient matrix G does not need to be formed explicitly. Given a guess of $\mathbf{Q}^k$, the matrix vector multiplication that is needed for the GMRES iterative method is simply

$$G\mathbf{Q^k} = \begin{pmatrix} [\mathbf{U}(\mathbf{Q}^k)] \\ [\mathbf{V}(\mathbf{Q}^k)] \end{pmatrix} - \begin{pmatrix} [\mathbf{U}(\mathbf{0})] \\ [\mathbf{V}(\mathbf{0})] \end{pmatrix}. \tag{4.33}$$

4.5. **Computing the residual vector using the least squares interpolation.** The matrix-vector multiplication (4.33) only involves the velocity jumps at the inner projection points. However, the velocity solved from the Poisson equations (2.3, 2.4, 2.5) is defined only at grid points. We use a least squares interpolation scheme again to get the velocity at the interface points. The interpolation scheme is critical for the accuracy and convergence speed of the GMRES method and thus is very important for the new method.

For an inner projection point $\mathbf{x}^* = (x^*, y^*)$ and its neighboring grid points $\{\mathbf{x} = (x_m, y_n) | m, n \in \mathbb{N}\}$, we can find

$$(\tilde{u}(\mathbf{x}^*))^+ = \lim_{\mathbf{x} \to (\mathbf{x}^*)^+} \tilde{u}(\mathbf{x}) \tag{4.34}$$

using the following interpolation

$$\sum_{m,n} \beta_{m,n} \tilde{u}(x_m, y_n) + C \tag{4.35}$$

where the coefficients $\beta_{m,n}$ and the correction term C are to be determined. It is more convenient to use the local coordinate system in the tangential and normal directions (Fig. 1) defined as

$$\begin{aligned} \xi_m &= (x_m - x^*) \cos\theta + (y_n - y^*) \sin\theta, \\ \eta_n &= -(x_m - x^*) \sin\theta + (y_n - y^*) \cos\theta. \end{aligned} \tag{4.36}$$

The Taylor expansion of $\tilde{u}(x_m, y_n)$ at $(\xi, \eta) = (0, 0)$ is

$$\begin{aligned} &\tilde{u}(\xi_m, \eta_n) \\ &= \tilde{u}^\pm + \frac{\partial \tilde{u}^\pm}{\partial \xi} \xi_m + \frac{\partial \tilde{u}^\pm}{\partial \eta} \eta_n + \frac{1}{2} \frac{\partial^2 \tilde{u}^\pm}{\partial \xi^2} \xi_m^2 + \frac{1}{2} \frac{\partial^2 \tilde{u}^\pm}{\partial \eta^2} \eta_n^2 + \frac{\partial^2 \tilde{u}^\pm}{\partial \xi \partial \eta} \xi_m \eta_n + \dots . \end{aligned} \tag{4.37}$$

The $\pm$ sign is determined according to whether $\mathbf{x} \in \Omega^\pm$. Let us define

$$\begin{aligned}
a_1^+ &= \sum_{(x_m,y_n)\in\Omega^+} \beta_{m,n}, & a_1^- &= \sum_{(x_m,y_n)\in\Omega^-} \beta_{m,n}, \\
a_2^+ &= \sum_{(x_m,y_n)\in\Omega^+} \beta_{m,n}\xi_m, & a_2^- &= \sum_{(x_m,y_n)\in\Omega^-} \beta_{m,n}\xi_m, \\
a_3^+ &= \sum_{(x_m,y_n)\in\Omega^+} \beta_{m,n}\eta_n, & a_3^- &= \sum_{(x_m,y_n)\in\Omega^-} \beta_{m,n}\eta_n, \\
a_4^+ &= \sum_{(x_m,y_n)\in\Omega^+} \beta_{m,n}\frac{1}{2}\xi_m^2, & a_4^- &= \sum_{(x_m,y_n)\in\Omega^-} \beta_{m,n}\frac{1}{2}\xi_m^2, \\
a_5^+ &= \sum_{(x_m,y_n)\in\Omega^+} \beta_{m,n}\frac{1}{2}\eta_n^2, & a_5^- &= \sum_{(x_m,y_n)\in\Omega^-} \beta_{m,n}\frac{1}{2}\eta_n^2, \\
a_6^+ &= \sum_{(x_m,y_n)\in\Omega^+} \beta_{m,n}\xi_m\eta_n, & a_6^- &= \sum_{(x_m,y_n)\in\Omega^-} \beta_{m,n}\xi_m\eta_n.
\end{aligned} \tag{4.38}$$

Substituting (4.37) into (4.35), the coefficients $\beta_{m,n}$ of the interpolation scheme for $\tilde{u}^+$ are seen to be the solution of the following linear system of equations

$$\begin{aligned} a_1^+ + a_1^- &= 1, \quad a_2^+ + a_2^- = 0, \quad a_3^+ + a_3^- = 0, \\ a_4^+ + a_4^- &= 0, \quad a_5^+ + a_5^- = 0, \quad a_6^+ + a_6^- = 0. \end{aligned} \tag{4.39}$$

The correction term is

$$C = a_1^-[\tilde{u}] + a_2^-\big[\frac{\partial \tilde{u}}{\partial \xi}\big] + a_3^-\big[\frac{\partial \tilde{u}}{\partial \eta}\big] + a_4^-\big[\frac{\partial^2 \tilde{u}}{\partial \xi^2}\big] + a_5^-\big[\frac{\partial^2 \tilde{u}}{\partial \eta^2}\big] + a_6^-\big[\frac{\partial^2 \tilde{u}}{\partial \xi \partial \eta}\big]. \tag{4.40}$$

Again, we choose more than six grid points for the interpolation to get an under-determined system of equations. The linear system of equations is solved by the

SVD method. The jump conditions in the correction term can be represented by the jump in the solution and the flux (see [15]).

Similarly, we can get an interpolation scheme for $\tilde{u}^-$. Once we have $\tilde{u}^\pm$ at all orthogonal projections of inner irregular grid points, it is also straightforward to get $\begin{pmatrix} [\mathbf{U}] \\ [\mathbf{V}] \end{pmatrix}$ which is needed in the matrix-vector multiplication.

To get the interpolation grid points for a given inner projection point $\mathbf{x}^* = (x^*, y^*)$, we choose at least 6 closest grid points to (x^*, y^*) as the interpolation stencil so that we can have a third order accurate interpolation scheme. To avoid excess points we use $|\,(\xi_m, \eta_n)\,| \le 2.5h$.

4.6. **The level set method for moving interfaces.** We use the level set method first introduced by Osher & Sethian [22] to solve the moving interface problems modeled by the incompressible Stokes equations. The level set function is evolved according to the level set function (a Hamilton-Jacobi equation)

$$\frac{\partial \varphi}{\partial t} + V_n |\nabla \varphi| = 0 \tag{4.41}$$

where $V_n = \mathbf{u} \cdot \mathbf{n}$ is the component of the velocity in the level sets normal direction and $\varphi = \varphi(\mathbf{x}, t)$. The level set function φ is often chosen as the signed distance function which satisfies the Eikonal equation

$$|\nabla \varphi| = 1 \tag{4.42}$$

at least in a neighborhood of the interface. To ensure that the level set function φ remains the signed distance function, we solve the following re-initialization equation at each time step after interface advection until $|\nabla \varphi| = 1 + \mathrm{O}(h^2)$:

$$\begin{aligned} \varphi(\mathbf{x}, t = 0) &= \varphi_0, \\ \frac{\partial \varphi}{\partial t} &= \operatorname{sign}(\varphi_0)(1 - |\nabla \varphi|). \end{aligned} \tag{4.43}$$

Numerical methods [21, 30] for Hamilton-Jacobi (HJ) equations can be used to solve the level set equations (4.41)-(4.43). In our implementation, we use the explicit forward Euler scheme for the time derivative and the third order WENO (Weighted Essentially Non-Oscillatory) scheme for the spatial derivatives.

The standard level set methods do not preserve volume well. Sussman et. al. [31] incorporated the volume conservation condition into the re-initialization equation (4.43) and a hybrid particle level set approach has also been proposed [3]. We use a simple approach described in [7] to preserve the area.

5. Numerical examples

In this section we present one example with fixed interface, one with a moving interface and one with four moving interfaces. All the computations are done on a P4 2.80GHz DELL desktop PC with 512MB memory. The computational domain is $\Omega = [-2, 2] \times [-2, 2]$. We choose to fix $\mu^+ = 1.0$, so $\lambda = \frac{\mu^-}{\mu^+} = \mu^-$.

5.1. **A general fixed interface test problem.** We first present some results to a general fixed interface test problem. The interface is chosen to be a unit circle

$$\varphi(x, y) = \sqrt{x^2 + y^2} - 1.0. \tag{5.1}$$

The initial guess of GMRES for the augmented variable $\mathbf{Q}$ takes zero values. The non-physical exact test solutions of $(\mathbf{u}, p)$ are constructed as [15]

$$
\begin{aligned}
u &= \begin{cases} \frac{y}{4} & \text{if } x^2+y^2<1, \\ \frac{y}{4}(x^2+y^2) & \text{if } x^2+y^2 \geq 1; \end{cases} \\
v &= \begin{cases} -\frac{x}{4}(1-x^2) & \text{if } x^2+y^2<1, \\ -\frac{xy^2}{4} & \text{if } x^2+y^2 \geq 1; \end{cases} \\
p &= \begin{cases} (-\frac{3}{4}x^3+\frac{3}{8}x)y & \text{if } x^2+y^2<1, \\ 0 & \text{if } x^2+y^2 \geq 1. \end{cases}
\end{aligned} \tag{5.2}
$$

The body force term $\mathbf{g} = (g_1, g_2)^T$ is chosen to be

$$
\begin{aligned}
g_1 &= \begin{cases} \left(-\frac{9}{4}x^2+\frac{3}{8}\right)y & \text{if } x^2+y^2<1, \\ -2\mu^+ y & \text{if } x^2+y^2 \geq 1; \end{cases} \\
g_2 &= \begin{cases} -\frac{3}{4}x^3+\frac{3}{8}x-\frac{3\mu^-}{2}x & \text{if } x^2+y^2<1, \\ \frac{\mu^+}{2}x & \text{if } x^2+y^2 \geq 1. \end{cases}
\end{aligned} \tag{5.3}
$$

And the singular interface force terms are

$$
\begin{aligned}
\hat{f}_1 &= (\frac{3}{4}\cos^3\theta - \frac{3}{8}\cos\theta)\sin\theta - \frac{3}{2}\cos^3\theta\sin\theta[\mu], \\
\hat{f}_2 &= \frac{1}{2}\mu^+ + \frac{3}{4}[\mu]\cos^2\theta(1-2\cos^2\theta).
\end{aligned} \tag{5.4}
$$

To accommodate the exact solutions, the jump condition for $\frac{\partial p}{\partial n}$ is modified by adding a correction term, which does not change the difficulty of the problem. Note that this is a very general fixed interface problem with non-homogeneous jumps and tangential derivatives in all quantities.

We use linear regression analysis to find the convergence order of all quantities. For this purpose, we choose $m = n = 32 + 16k(k = 0, \dots, 54)$ and run the convergence analysis for $\lambda = 0.001, 0.1, 10$ and 1000. Fig. 3 is the grid refinement analysis of the test problem. The error is the $\mathcal{L}^\infty$ error at all grid points. The slope of the linear regression line is the order of the convergence. It is clear from these results that the algorithm is second order accurate in velocity $\mathbf{u} = (u, v)^T$, as expected, and almost second order accurate in pressure p. For this problem, it also seems that with larger λ, the numerical solution for u (or v) tends to be more accurate than the solution for p.

Fig. 4 shows the algorithm efficiency analysis. The number of GMRES iterations seems to be independent of grid size for fixed λ, and increases slowly as λ increases. Most simulations finish within two minutes.

5.2. **A moving interface test problem.** We present a moving interface test problem where the surface tension is the only interface force, i.e. the normal force $\hat{f}_1 = \alpha\kappa$ where $\alpha > 0$ and κ is the curvature as defined before; we set the tangential force $\hat{f}_2 = 0.0$. We choose $\alpha = 1.0$ and $m = n = 64$ in our simulations. It is known that starting from an arbitrary interface shape without any body forces, i.e. $\mathbf{g} = \mathbf{0}$, the interface will evolve to the equilibrium circle shape [10]. The initial interface is

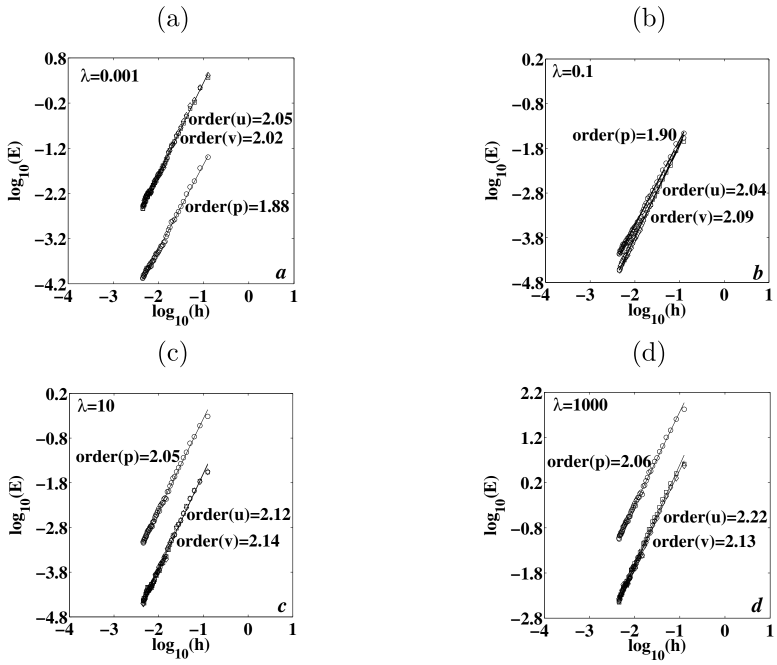

FIGURE 3. Convergence analysis in the $\mathcal{L}^\infty$ norm for four cases in log-log scale. The slope of the linear regression line is the convergence rate which is close to number 2.0 for all cases. (a) $\mu^- = 0.001$ and $\mu^+ = 1.0$. (b) $\mu^- = 0.1$ and $\mu^+ = 1.0$. (c) $\mu^- = 10$ and $\mu^+ = 1.0$. (d) $\mu^- = 1000$ and $\mu^+ = 1.0$. Linear regressions $R^2 > 0.98$ for all cases.

chosen to be an ellipse

$$\varphi(x,y) = \sqrt{\frac{(x+y)^2}{2a^2} + \frac{(-x+y)^2}{2b^2}} - 1.0 \tag{5.5}$$

where $a = 2.0$ and $b = 0.5$. We use periodic boundary conditions for both $\mathbf{u}$ and p. See Fig. 5 for an example of the interface evolution. Also see Fig. 6 for a comparison of the evolving interfaces with different λ's at a fixed time. It is clear from Fig. 6 that the more viscous fluid moves slower. For all the moving interface simulations, the area conservation algorithm [7] preserves the initial area at each time step exactly.

5.3. **A moving multiple-interface problem.** We present another moving interface test problem with a multiply-connected interface consisting initially of four ellipses. As in the other test problems, we use periodic boundary conditions on the edges of the simulation rectangle. The two fluids have equal viscosity, the surface tension is given as $\hat{f}_1 = \alpha\kappa$ where $\alpha = 0.1$, and we set $\hat{f}_2 = 0$. The initial interfaces are easily constructed using the union of individual level set functions [21, 30]. As the four symmetrically placed ellipses relax to circles, we observe preservation of the symmetries in the flow. Equilibrium is reached when the pressure is uniformly distributed.

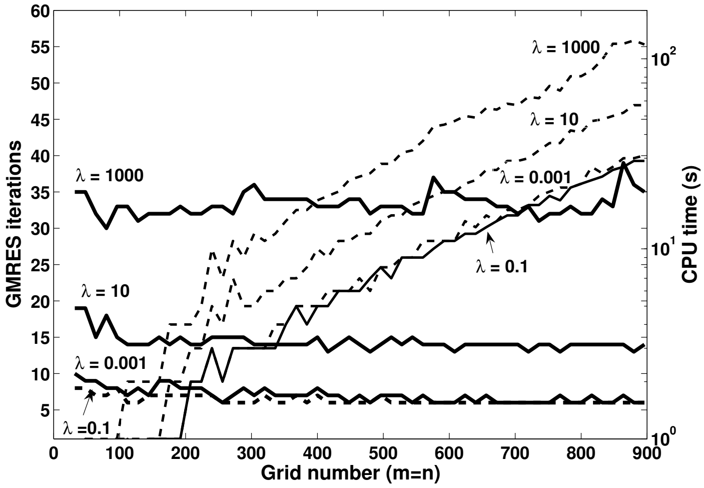

FIGURE 4. Algorithm efficiency analysis for 32 by 32 through 896 by 896 grids. Left axis represents number of GMRES iterations and right axis represents computational CPU cost with unit second using log scale. GMRES iterations (thick curves) remain relatively constant with grid size, and CPU time (thin curves) $\sim$ grid number^2.

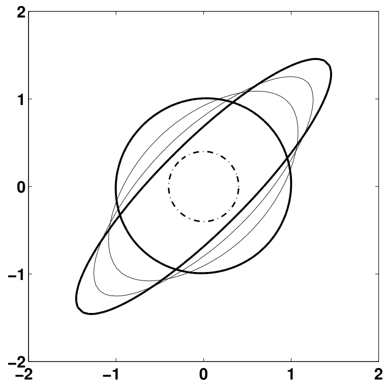

FIGURE 5. A moving interface simulation with $\lambda = 0.5$, at time $t = 0, 0.86, 1.90$ and 6.75. Starting from the ellipse, the interface evolves to a circle. The inner reference circle (dash curve) helps to show the symmetry of the equilibrium interface. The final area inside the interface is $3.14 \approx \pi$ so our algorithm preserves area well.

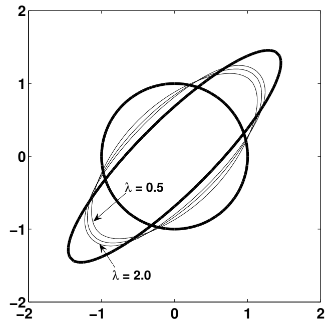

FIGURE 6. Comparison of viscosity effects on interface motion. Three curves (thin curves) at time $t = 1.5$, for $\lambda = 0.5, 1.0, 2.0$. $\lambda = 0.5$ approaches equilibrium (thick curve) fastest.

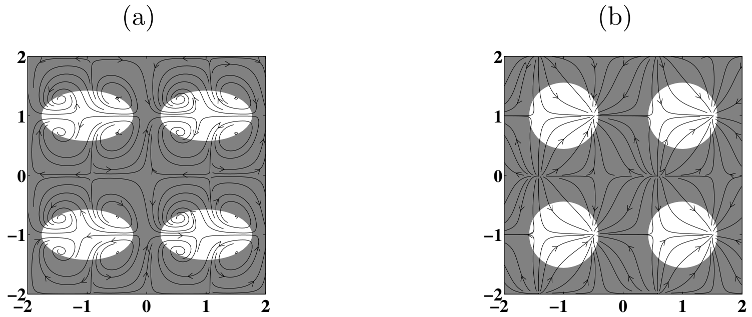

FIGURE 7. A moving interface simulation starting with four elliptic interfaces (white boundary). Streamlines are drawn and symmetry of streamlines are observed as expected. We choose $\lambda = 1$ here. (a) $t = 5$. (b) $t = 50$.

DISCUSSION

In this paper we proposed a new numerical method for Stokes equations with interfaces, by combining the augmented interface method with the level set method. Our method handles moving interfaces and multiply connected interfaces more easily than a spline approach, and could more easily be extended to 3D. Our algorithm is shown to be 2nd order accurate in both velocity and pressure, and conserves volume well. Our method should prove useful for interface problems in fluid flow and biological models.

One of the remaining open questions is how to use a preconditioning technique for the Schur complement system since the coefficient matrix for the Schur complement system is not explicitly formed. Even the simplest diagonal preconditioning technique needs to use the fast Stokes solver N_b times which would cost more than that of the entire solution process for the testing examples in this paper. On the other hand, as discussed in [13], a good interpolation scheme for the residual that takes advantages of the interface/jump conditions can reduce the number of iteration significantly. Therefore, it is more practical to tune-in the interpolation scheme as a preconditioner than to use preconditioning techniques that require structures of the coefficient matrix. Fortunately, for all the test examples in this paper, the number of GMRES iterations is modest (fewer than 55).

Acknowledgments. This work was partially supported by grants: 0201094 from NSF/NIH, 43751-MA from ARO, and DMS-0412654 from NSF. The first author would like to thank Prof. John Bishir for helpful discussions and proofreading the paper.

References

[1] A. J. Chorin, *Numerical solution of the Navier-Stokes equations*, Math. Comput. 22:745-762, 1968.

[2] R. Cortez, *The Method of Regularized Stokeslets*, sisc, 23: 1204-1225, 2001.

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

[4] L. J. Fauci and A. McDonald, *Sperm motility in the presence of boundaries*, B. Math. Biol., 57:679-699, 1994.

[5] A. L. Fogelson and C. S. Peskin; *Numerical solution of the three dimensional Stokes equations in the presence of suspended particles*, Proc. SIAM Conf. Multi-phase Flow, SIAM. June 1986.

[6] E. Givelberg and J. Bunn, *A comprehensive three-dimensional model of the cochlea*, J. Comput. Phys., 191:377-391, 2003.

[7] S. Groß, V. Reichelt and A. Reusken, *A finite element based level set method for two-phase incompressible flows*, IGPM-Report 243, RWTH Aachen, 2004.

[8] S. Jadhav, C. D. Eggleton and K. Konstantopoulos, *A 3-D computational model predicts that cell deformation affects selectin-mediated leukocyte rolling*, Biophys. J., 88:96-104, 2005.

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

[10] R. J. LeVeque and Z. Li, *Immersed interface methods for Stokes flow with elastic boundaries or surface tension*, SIAM J. Sci. Comput., 18:709-735, 1997.

[11] Z. Li, *The immersed interface method - a numerical approach for partial differential equations with interfaces*, PhD thesis, University of Washington, 1994.

[12] Z. Li, *An overview of the immersed interface method and its applications*, Taiwanese J. Mathematics, 7:1-49, 2003.

[13] Z. Li, *A Fast Iterative Algorithm for Elliptic Interface Problems*, SIAM J. Numer. Anal., 35:230-254, 1998.

[14] Z. Li and K. Ito, *Maximum Principle Preserving Schemes for Interface Problems with Discontinuous Coefficients*, SIAM J. Sci. Comput., 23:1225-1242, 2001.

[15] Z. Li and K. Ito and M-C. Lai; *An augmented approach for Stokes equations with a discontinuous viscosity and singular forces*, NCSU-CRSC Tech. Report: CRSC-TR04-23, North Carolina State University, 2004.

[16] Z. Li and S. R. Lubkin, *Numerical Analysis of Interfacial 2D Stokes Flow with Discontinuous Viscosity and Variable Surface Tension*, Numerical Methods in Fluids, 37:525-540, 2001.

[17] Z. Li, H. Zhao and H. Gao, *A numerical study of electro-migration voiding by evolving level set functions on a fixed Cartesian grid*, J. Comput. Phys., 152:281-304, 1999.

[18] S. R. Lubkin and Z. Li, *Force and Deformation on Branching Rudiments: Cleaving Between Hypotheses*, Biomechanics and Modeling in Mechanobiology, 1(1):5-16, 2002.

[19] A. Mayo and C. S. Peskin, *An implicit numerical method for fluid dynamics problems with immersed elastic boundaries*, Contemp. Math., 141:261-277. 1991.

[20] D. M. McQueen and C. S. Peskin, *Heart simulation by an immersed boundary method with formal second-order accuracy and reduced numerical viscosity*, In Mechanics for a New Millennium, Proceedings of the International Conference on Theoretical and Applied Mechanics (H. Aref and J. W. Phillips, eds.), Kluwer Academic Publishers, 2001.

[21] S. J. Osher and R. P. Fedkiw, *Level set methods and dynamic implicit surfaces*, Springer-Verlag, 2002.

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

[23] P. Swarztrauber and R. Sweet, *Efficient FORTRAN subprograms for the solution of elliptic partial differential equations*, NCAR Technical Note-TN/IA-109, 1975.

[24] C. S. Peskin, *The immersed boundary method*, Acta Numerica, 1-39, 2002.

[25] C. S. Peskin and D. M. McQueen, *Fluid dynamics of the heart and its valves*, In Case Studies in Methematical Modeling: Ecology, Physiology, and Cell Biology (H. G. Othmer, F. R. Adler, M. A. Lewis and J. C. Dallon, eds.), Prentice-Hall, Englewood Cliffs, NJ, 309-337, 1996.

[26] C. Pozrikidis eds., *Modeling and simulation of capsules and biological cells*, Chapman & Hall, London, 2003.

[27] Y. Saad and M. Schultz, *GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems*, SIAM J. Sci. Statist. Comput., 7:856-869, 1986.

[28] Y. Saad, *GMRES: A Generalized minimal residual algorithm for solving nonsymmetric linear systems*, sissc, 7:856-869, 1986.

[29] J. A. Sethian and P. Smereka, *Level set methods for fluid interfaces*, Annu. Rev. Fluid Mech., 35:341-72, 2003.

[30] J. A. Sethian, *Level Set Methods and Fast Marching methods*, Cambridge University Press, nd edition, 1999.

[31] M. Sussman and E. Fatemi, *An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow*, SIAM J. Sci. Comput., 20:1165-1191, 1999.

[32] C. Tu and C. S. Peskin, *Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods*, sissc, 13:1361-1376, 1992.

[33] X. Wang and W. K. Liu, *Extended immersed boundary method using FEM and RKPM*, Comput. Methods Appl. Mech. Eng., 193:1305-1321, 2004.

[34] A. Wiegmann and K. P. Bube, *The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions*, SIAM J. Numer. Anal., 37:827-862, 2000.

[35] L. Zhang, A. Gerstenberger, X. Wang and W. K. Liu, *Immersed finite element method*, Comput. Methods Appl. Mech. Engrg., 193:2051-2067, 2004.

Zhilin Li
Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA
E-mail address: zhilin@math.ncsu.edu

Sharon R. Lubkin
Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA
E-mail address: lubkin@math.ncsu.edu

Xiaohai Wan
Biomathematics Graduate Program & Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8203, USA
E-mail address: xwan@unity.ncsu.edu