Available online at www.sciencedirect.com
ScienceDirect

Computers & Fluids 36 (2007) 622–635

computers & fluids

www.elsevier.com/locate/compfluid

An augmented approach for Stokes equations with a discontinuous viscosity and singular forces

Zhilin Li [a,*], Kazufumi Ito [a], Ming-Chih Lai [b]

[a] *Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA*
[b] *Department of Applied Mathematics, National Chiao Tung University, Hsinchu 30050, Taiwan*

Received 5 September 2005; received in revised form 4 February 2006; accepted 13 March 2006
Available online 27 June 2006

Abstract

For Stokes equations with a discontinuous viscosity across an arbitrary interface or/and singular forces along the interface, it is known that the pressure is discontinuous and the velocity is non-smooth. It has been shown that these discontinuities are coupled together, which makes it difficult to obtain accurate numerical solutions. In this paper, a new numerical method that decouples the jump conditions of the fluid variables through two augmented variables has been developed. The GMRES iterative method is used to solve the Schur complement system for the augmented variables that are only defined on the interface. The augmented approach also rescales the Stokes equations in such a way that a fast Poisson solver can be used in each iteration. Numerical tests using examples that have analytic solutions show that the new method has average second order accuracy for the velocity in the infinity norm. An example of a moving interface problem is also presented.

1. Introduction

The incompressible Stokes or Navier–Stokes equations with a discontinuous viscosity and singular forces arise from many important applications in fluid and bio-fluid mechanics. One particular example is Peskin's immersed boundary (IB) model that was introduced to simulate the blood flow in a human's heart [28,29,37]. The idea of IB formulation is to treat the complicated immersed boundary (such as a heart valve) as a force generator in the fluid domain, or mathematically, a Dirac delta function force distribution along the immersed boundary. In this paper, we consider the following two-dimensional stationary Stokes equations:

$$\nabla p = \nabla\cdot\mu\Big(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}}\Big) + \mathbf{g} + \int_\Gamma \mathbf{f}(s)\delta_2(\mathbf{x}-\mathbf{X}(s))\,\mathrm{d}s, \quad \mathbf{x}\in\Omega, \tag{1.1}$$

$$\nabla\cdot\mathbf{u} = \mathbf{0}, \quad \mathbf{x}\in\Omega, \tag{1.2}$$

where $\mathbf{u}=(u,v)$ is the velocity vector, p is the pressure, Γ is an arbitrary interface parameterized by the arc-length s, and $\mathbf{g}(\mathbf{x})$ is an external force. The force density $\mathbf{f}$ is only defined on the interface Γ[1] that separates two fluid regions Ω^- and Ω^+. The viscosity μ is assumed to be a piecewise constant

$$\mu = \begin{cases} \mu^+, & \text{if } \mathbf{x}\in\Omega^+, \\ \mu^-, & \text{if } \mathbf{x}\in\Omega^-. \end{cases} \tag{1.3}$$

We refer the readers to Fig. 1(a) for an illustration of the problem. The existence of the solution to the system (1.1)–(1.2) can be found in [35].

There are at least two difficulties in solving (1.1)–(1.2) 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 to distribute the singular force to nearby grid points. Such a discretization is typically first order accurate and will smooth out the

* Corresponding author.
E-mail addresses: zhilin@math.ncsu.edu (Z. Li), kito@math.ncsu.edu (K. Ito), mclai@math.nctu.edu.tw (M.-C. Lai).

[1] The singular source term $\int_\Gamma \mathbf{f}(s)\delta_2(\mathbf{x}-\mathbf{X}(s))\,\mathrm{d}s$ can also be written as $((\mathbf{f}\cdot\mathbf{n})\mathbf{n}+(\mathbf{f}\cdot\boldsymbol{\tau})\boldsymbol{\tau})\delta(\Gamma)$, or in the form of $((\mathbf{f}\cdot\mathbf{n})\mathbf{n}+(\mathbf{f}\cdot\boldsymbol{\tau})\boldsymbol{\tau})|\nabla\varphi|\delta(\varphi)$, where φ is a level set function whose zero level set represents the interface Γ.

0045-7930/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2006.03.003

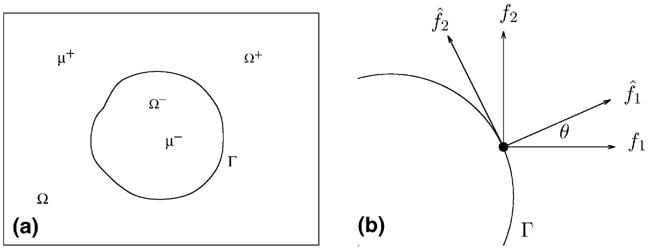

Fig. 1. (a) A diagram for the incompressible Stokes equations defined on a domain Ω with an interface Γ across which the viscosity μ is discontinuous. (b) Force density decomposition in which f_1 and f_2 are the force density in the x- and y-directions, while $\hat{f}_1$ and $\hat{f}_2$ are the force density in the normal and tangential directions.

solution. The second difficulty is how to deal with the discontinuous viscosity. A simple smoothing method may introduce large errors, see [21] for a one-dimensional example there.

In the case of a continuous viscosity with a Dirac delta function source distribution along an interface, various methods have been developed in the literature. We refer the readers to [4,5,9,18,26,37] 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, see (2.7)–(2.10), which makes it difficult to discretize the system accurately.

In this paper, we develop a new second-order sharp interface method that uses the exact jump conditions for the two two-phase Stokes equations (1.1)–(1.3) with a discontinuous viscosity. The idea is to introduce two augmented variables that are defined only along the interface so that the jump conditions can be decoupled and the immersed interface method (IIM) [17,19] can be applied. The GMRES iterative method is applied to the Schur complement system for the discrete augmented variables. Furthermore, our approach rescales the original problem and enables us to use a fast Poisson solver in the iterative process. Each GMRES iteration requires solving the rescaled Stokes equations, which can be done by calling a fast Poisson solver three times, and an interpolation scheme to evaluate the residual of the Schur complement. While augmented approaches have been developed for elliptic interface problems or problems defined on irregular domain problems in [3,8,10,20,23,24,40], and for dealing with the pressure boundary condition in [22], the augmented approach proposed in this paper provides a way to get a second order immersed interface method using a finite difference discretization to solve the Stokes equations (1.1)–(1.3) with a discontinuous viscosity.

The main focus of this paper is to test and implement our proposed augmented approach to the two-phase Stokes equations. Our implementation and results reported in this paper are based on the cubic spline representation of the interface [19]. In order to apply the method for problems on multi-connected domains and moving interface problems, the level set method is preferred, see, for example, [24,27,32]. The level set method using the augmented approach for the two-phase Stokes equations has yet to be developed.

A few other finite difference methods may be applicable for solving the Stokes equations (1.1)–(1.3) with a discontinuous viscosity. A simple approach is the smoothing method that treats the viscosity as a continuous function using

$$\mu(\mathbf{x}) = \mu^- + (\mu^+ - \mu^-)H_\epsilon(\varphi(\mathbf{x})), \tag{1.4}$$

where $\varphi(\mathbf{x})$ is a two-dimensional Lipschitz continuous function whose zero level set is the interface Γ, H_ϵ is a smoothed Heaviside function, see, for example, [33,34,36]. In this way, a standard Stokes solver on a rectangular region can be used. The disadvantage is that the solution is smeared in a neighborhood of the interface and it is usually first order accurate. Among sharp interface methods, that is, methods for which the computed solution satisfies the jump conditions across the interface either exactly or approximately, the fast Stokes solver on irregular domains based on an integral equation coupled with a finite difference discretization [2,6] may be applicable to the two-phase Stokes flow discussed here. In [25], an immersed interface method that requires three fast Poisson solvers within each iteration was developed assuming that the jump in the viscosity is small so that the jump conditions can be approximated with those for a continuous viscosity, that is, the coupling terms in (2.7)–(2.10) are ignored. Such approximation may not be valid anymore if the jump in the viscosity is large. The ghost fluid method developed in [14] for the two-phase Navier–Stokes equations is applicable to the two-phase Stokes equations with first order accuracy. It is also possible to generalize the finite volume methods developed for elliptic and Navier–Stokes equations on irregular domains in [1,12] to the two-phase Stokes flow.

Motivated by the augmented approach (fast IIM) [20] for elliptic interface problems with piecewise constant coefficients, Bube and Wiegmann developed the explicit immersed interface method (EJIIM) for elliptic interface problems in [38,40]. The EJIIM introduces unknown jumps in the solution and its up to second order derivatives along the interface. The GMRES method is then applied for the unknown jumps. The EJIIM has been applied to nonlinear 1D interface problems in [39] and two-dimensional elastic equations in shape design in conjunction with the level set method [31]. While we believe that the EJIIM should be applicable to the two-phase Stokes flow, it has not yet been done and the derivation and implementation are not trivial. Furthermore, the augmented system of equations using EJIIM will be larger than the augmented system in this paper because more unknowns will be introduced as augmented variables.

Our method is a new finite difference method that rescales the problem, has second order accuracy in the maximum norm, and takes advantage of a fast Poisson solver. These are a few merits about the new method that is competitive against other finite difference methods such as the one that uses smoothing technique and a discrete delta function.

While augmented methods have some similarities to boundary integral methods to find a source strength, the augmented methods have a few special features: (1) no Green function is needed, and therefore no need to evaluate singular integrals; (2) no need to set up the system of equations for the augmented variable explicitly; (3) applicable to general PDEs with or without source terms; (4) the process does not depend on boundary conditions. On the other hand, we have less information about the condition number of the Schur complement system and how to apply pre-conditioning techniques. A boundary integral method requires less computation compared with the augmented method if only the velocity on the interface is needed. The augmented method proposed here requires less computation compared with a boundary integral method if we need the velocity in the entire domain as in many applications.

The paper is organized as follows. In the next section, we discuss the jump conditions of the Stokes equations with a discontinuous viscosity along an interface. We will see how the jump conditions for the pressure and velocity are coupled together, and explain how to decouple the jump conditions by introducing two augmented variables and two augmented equations. In Section 3, we present the new algorithm in detail. We explain how the GMRES iterative method can be applied to the discrete augmented variables without explicitly forming the coefficient matrix. In Section 4, we present some numerical experiments using examples that have analytic solutions to check the convergence and the performance of the new method. An example of moving interface is also presented there.

2. Jump conditions for the Stokes equations with a discontinuous viscosity and singular forces

Referring to Fig. 1(a) and Eqs. (1.1)–(1.3), we assume the interface Γ is a smooth curve. We use $\mathbf{n}$ and $\boldsymbol{\tau}$ to represent the unit normal and tangential directions at a point (X, Y) on the interface. We use $\hat{f}_1(s) = \mathbf{f}(s)\cdot\mathbf{n}$ and $\hat{f}_2(s) = \mathbf{f}(s)\cdot\boldsymbol{\tau}$ to represent the force density in the normal and tangential directions. If Γ is smooth, then in a neighborhood of Γ, the distance function $d(\mathbf{x},\Gamma)$ is also a smooth function. The normal and tangential directions of Γ can be extended to the neighborhood, for example, $\mathbf{n} = \nabla d(\mathbf{x},\Gamma)/|\nabla d(\mathbf{x},\Gamma)|$. Therefore the quantities such as $\mathbf{u}\cdot\mathbf{n}$, $\frac{\partial \mathbf{u}}{\partial n}$, $\frac{\partial \mathbf{u}}{\partial \tau}$, etc., are well defined in the neighborhood of Γ.

The interface conditions for the two-phase Stokes equations with a discontinuous viscosity and a Dirac delta function singular force are derived in [11]. The idea is from the immersed interface method to express all the quantities in the local Cartesian coordinate at a point (X, Y) on the interface Γ as,

$$\left\{\begin{array}{l} \xi = (x - X)\cos\theta + (y - Y)\sin\theta, \\ \eta = -(x - X)\sin\theta + (y - Y)\cos\theta, \end{array}\right. \tag{2.5}$$

where θ is the angle between the x-axis and the normal direction at the point (X, Y), see Fig. 1(b) for an illustration. In the local coordinate system, the interface can parameterized by $\xi = \chi(\eta)$, $\eta = \eta$. Note that $\chi(0) = 0$, $\chi'(0) = 0$, $\chi'(0) = \kappa$, the curvature of the interface at the point (X, Y). The unit normal and tangential directions are

$$\begin{aligned} \mathbf{n} &= \left(\frac{1}{\sqrt{1+(\chi'(\eta))^2}}, -\frac{\chi'(\eta)}{\sqrt{1+(\chi'(\eta))^2}}\right), \\ \boldsymbol{\tau} &= \left(\frac{\chi'(\eta)}{\sqrt{1+(\chi'(\eta))^2}}, \frac{1}{\sqrt{1+(\chi'(\eta))^2}}\right). \end{aligned} \tag{2.6}$$

We also express the velocity components in the tangential and normal directions as $\hat{u} = \mathbf{u}\cdot\mathbf{n}$ and $\hat{v} = \mathbf{u}\cdot\boldsymbol{\tau}$.

The jump conditions at a fixed point (X, Y) (or $(0, 0)$ in the local coordinate system) are summarized in the following theorem.

Theorem 1. *Assume $\Gamma(s) \in C^2$, $\hat{f}_1(s) \in C^1$, and $\hat{f}_2(s) \in C^1$. Let p and $\mathbf{u}$ be the solution to the Stokes equations (1.1)–(1.2). We have the following jump conditions across the interface Γ at the fixed point (X, Y) in terms of the local Cartesian coordinate system*

$$[p] = 2\left[\mu\frac{\partial \hat{u}}{\partial \xi}\right] + \hat{f}_1, \tag{2.7}$$

$$\left[\frac{\partial p}{\partial \xi}\right] = [\mathbf{g}\cdot\mathbf{n}] + \frac{\partial \hat{f}_2}{\partial \eta} + 2\left[\mu\frac{\partial^2 \hat{u}}{\partial \eta^2}\right] - 4\kappa\left[\mu\frac{\partial \hat{v}}{\partial \eta}\right], \tag{2.8}$$

$$\left[\mu\frac{\partial \hat{v}}{\partial \xi}\right] + \left[\mu\frac{\partial \hat{u}}{\partial \eta}\right] + \hat{f}_2 = 0, \tag{2.9}$$

$$[\mu\nabla\cdot\mathbf{u}] = 0, \tag{2.10}$$

$$[\mathbf{u}] = 0, \tag{2.11}$$

where the jump $[\,\cdot\,]$ *of a quantity, for example,* $[p]$ *at a point* $\mathbf{X}$ *is defined as*

$$[p]|_{\mathbf{X}\in\Gamma} \overset{def}{=} \lim_{\mathbf{x}\to\mathbf{X},\mathbf{x}\in\Omega^+} p(\mathbf{x}) - \lim_{\mathbf{x}\to\mathbf{X},\mathbf{x}\in\Omega^-} p(\mathbf{x}). \tag{2.12}$$

Note that, although the jump condition $\left[\frac{\partial p}{\partial n}\right]$ is not needed to close the system mathematically, the condition is used in our numerical algorithm to obtain second-order discretization for the pressure. The jump condition is not arbitrary but determined from the governing equations and the force strength. This is similar to the discussion of the boundary condition of the pressure when a no-slip boundary condition is prescribed for the velocity.

We will omit the subscript $\mathbf{X}\in\Gamma$ in the rest of the paper for simplicity. The first jump condition (2.7) is the result of balancing force in the normal direction while the third one (2.9) is the result of balancing force in the tangential direction. The jump condition (2.10) is obtained from the incompressibility condition $\nabla\cdot\mathbf{u} = 0$. The jump condition for the normal derivative of the pressure (2.8) can be obtained by applying the divergence operator to the momentum equation (1.1),

$$\nabla p = \nabla \cdot \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}}) + \mathbf{g}$$

excluding the interface, see [11] for the details.

From the incompressibility condition, one can easily prove that $\left[\frac{\partial \hat{u}}{\partial \xi}\right] = \left[\frac{\partial \mathbf{u}}{\partial n} \cdot \mathbf{n}\right] = 0$. If μ is continuous, then $\left[\mu \frac{\partial \hat{u}}{\partial \xi}\right] = 0$ and $2\left[\mu \frac{\partial^2 \hat{u}}{\partial \eta^2}\right] - 4\kappa\left[\mu \frac{\partial \hat{v}}{\partial \eta}\right] = 0$, and the jump conditions for the pressure and the velocity are decoupled in (2.7)–(2.10). In this case, we recover the jump conditions derived and used in [16,18]. A second order accurate immersed interface method has been developed in [18,19] if the viscosity is continuous.

Note that using the invariance of the first order derivatives, the jump conditions (2.7) and (2.9) can also be written as

$$[p] = 2\left[\mu \frac{\partial \mathbf{u}}{\partial n} \cdot \mathbf{n}\right] + \hat{f}_1, \tag{2.13}$$

$$\left[\mu \frac{\partial \mathbf{u}}{\partial n} \cdot \boldsymbol{\tau}\right] + \left[\mu \frac{\partial \mathbf{u}}{\partial \tau} \cdot \mathbf{n}\right] + \hat{f}_2 = 0. \tag{2.14}$$

These jump conditions can be found in the literature.

To get a second order accurate algorithm based on the immersed interface method for the Stokes equations with a discontinuous viscosity, our strategy is to introduce two intermediate, or augmented variables, along the interface so that the jump conditions can be decoupled. We also need two augmented equations to close the system of the equations.

There are different ways to introduce augmented variables so that the jump conditions can be decoupled. For example, one could introduce $\left[\mu \frac{\partial \mathbf{u}}{\partial n}\right]$ as an intermediate unknown so that the jump conditions in the pressure and the velocity are de-coupled. Different augmented variables and equations will lead to different algorithms. Note that for a discontinuous viscosity, there are two different scales corresponding to the two-phase flow. Generally, if the viscosity ratio $\max\{\mu^-, \mu^+\}/\min\{\mu^-, \mu^+\}$ is large, then the velocity field with smaller viscosity has larger gradient than that with larger viscosity. Furthermore, with a constant viscosity and a periodic boundary condition, we can use a fast Poisson solver to solve the stationary Stokes equations, see [18,19]. Based on these two considerations, we introduce the jumps $[\mu u](s)$ and $[\mu v](s)$ along the interface as two augmented variables. The advantages and details can be seen in the rest of the paper.

Using this local coordinate system, we have $\frac{\partial}{\partial n} = \frac{\partial}{\partial \xi}$ and $\frac{\partial}{\partial \tau} = \frac{\partial}{\partial \eta}$. We can rewrite the last two jump conditions (2.9)–(2.10) in terms of the augmented variables $[\mu u](s)$ and $[\mu v](s)$ as follows.

Lemma 1. *Let p, u, and v be the solution to the Stokes equations (1.1)–(1.2). Define*

$$\tilde{u} = \mu u, \quad \tilde{v} = \mu v, \quad \tilde{\mathbf{u}} = (\tilde{u}, \tilde{v}). \tag{2.15}$$

Then we have the following jump relations for $\tilde{u}$ and $\tilde{v}$

$$\left[\frac{\partial \tilde{u}}{\partial n}\right] = \left(\hat{f}_2 + \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \mathbf{n}\right]\right) \sin\theta - \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \boldsymbol{\tau}\right] \cos\theta, \tag{2.16}$$

$$\left[\frac{\partial \tilde{v}}{\partial n}\right] = -\left(\hat{f}_2 + \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \mathbf{n}\right]\right) \cos\theta - \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \boldsymbol{\tau}\right] \sin\theta, \tag{2.17}$$

$$\left[\frac{\partial \tilde{\mathbf{u}}}{\partial n} \cdot \mathbf{n}\right] = -\left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \boldsymbol{\tau}\right]. \tag{2.18}$$

Proof. Note that $\mathbf{n} = (\cos\theta, \sin\theta)$ and $\tau = (-\sin\theta, \cos\theta)$. Re-write the incompressibility condition $[\mu \nabla \cdot \mathbf{u}] = 0$ in the local coordinates, we have

$$\left[\frac{\partial \tilde{u}}{\partial n}\right] \cos\theta - \left[\frac{\partial \tilde{u}}{\partial \tau}\right] \sin\theta + \left[\frac{\partial \tilde{v}}{\partial n}\right] \sin\theta + \left[\frac{\partial \tilde{v}}{\partial \tau}\right] \cos\theta = 0, \tag{2.19}$$

which is

$$\left[\frac{\partial \tilde{u}}{\partial n}\right] \cos\theta + \left[\frac{\partial \tilde{v}}{\partial n}\right] \sin\theta = -\left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \boldsymbol{\tau}\right]. \tag{2.20}$$

Re-write the interface relation (2.14) in the local coordinates, we have

$$-\left[\frac{\partial \tilde{u}}{\partial n}\right] \sin\theta + \left[\frac{\partial \tilde{v}}{\partial n}\right] \cos\theta = -\hat{f}_2 - \left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \mathbf{n}\right]. \tag{2.21}$$

From the two equalities (2.20) and (2.21) above, we solve $\left[\frac{\partial \tilde{u}}{\partial n}\right]$ and $\left[\frac{\partial \tilde{v}}{\partial n}\right]$ to get (2.16) and (2.17). The last equality is verified by substituting $\left[\frac{\partial \tilde{u}}{\partial n}\right]$ with (2.16), and $\left[\frac{\partial \tilde{v}}{\partial n}\right]$ with (2.17) in the following:

$$\left[\frac{\partial \tilde{\mathbf{u}}}{\partial n} \cdot \mathbf{n}\right] = \left[\frac{\partial \tilde{u}}{\partial n}\right] \cos\theta + \left[\frac{\partial \tilde{v}}{\partial n}\right] \sin\theta = -\left[\frac{\partial \tilde{\mathbf{u}}}{\partial \tau} \cdot \boldsymbol{\tau}\right]. \qquad \square$$

3. The numerical algorithm

Our numerical method is based on the following theorem.

Theorem 2. *Let p, u, and v be the solution to the Stokes equations (1.1)–(1.2). Let $q_1(s) = [\tilde{u}](s) = [\mu u](s)$, $q_2(s) = [\tilde{v}](s) = [\mu v](s)$, and $\mathbf{q}(s) = (q_1(s), q_2(s))$. Then $\tilde{u}$, $\tilde{v}$, p, $q_1(s)$, $q_2(s)$ are the solution of the following augmented system of partial differential equations*:

$$\begin{cases} \Delta p = \nabla \cdot \mathbf{g}, \\ [p] = \hat{f}_1 - 2\frac{\partial \mathbf{q}}{\partial \tau} \cdot \boldsymbol{\tau}, \quad \left[\frac{\partial p}{\partial n}\right] = [\mathbf{g} \cdot \mathbf{n}] + \frac{\partial \hat{f}_2}{\partial \tau} + w(\mathbf{q}), \end{cases} \tag{3.22}$$

$$\begin{cases} \Delta \tilde{u} = p_x - g_1, \\ [\tilde{u}] = q_1, \quad \left[\frac{\partial \tilde{u}}{\partial n}\right] = \left(\hat{f}_2 + \frac{\partial \mathbf{q}}{\partial \tau} \cdot \mathbf{n}\right) \sin\theta - \left(\frac{\partial \mathbf{q}}{\partial \tau} \cdot \boldsymbol{\tau}\right) \cos\theta, \end{cases} \tag{3.23}$$

$$\begin{cases} \Delta \tilde{v} = p_y - g_2, \\ [\tilde{v}] = q_2, \quad \left[\frac{\partial \tilde{v}}{\partial n}\right] = -\left(\hat{f}_2 + \frac{\partial \mathbf{q}}{\partial \tau} \cdot \mathbf{n}\right) \cos\theta - \left(\frac{\partial \mathbf{q}}{\partial \tau} \cdot \boldsymbol{\tau}\right) \sin\theta, \end{cases} \tag{3.24}$$

$$\left[\frac{\tilde{u}}{\mu}\right] = 0, \quad \left[\frac{\tilde{v}}{\mu}\right] = 0, \tag{3.25}$$

where

$$w(\mathbf{q}) = 2\frac{\partial^2 \hat{q}_1}{\partial \eta^2} - 2\kappa \frac{\partial \hat{q}_2}{\partial \eta}, \tag{3.26}$$

$\hat{q}_1 = \mathbf{q}\cdot\mathbf{n}$, *and* $\hat{q}_1 = \mathbf{q}\cdot\boldsymbol{\tau}$ *in the local Cartesian coordinate system.*

The proof of the theorem is straightforward from the Stokes equations (1.1)–(1.2), the jump conditions in Theorem 1 and Lemma 1. The periodic boundary condition is used here so that we are not introducing additional boundary condition for the pressure.

The existence and uniqueness of the solution to the system above is the same as the original incompressible Stokes equations (1.1)–(1.2). This is because if (u, v) and p are the solution to the original Stokes equations, then they are also the solution to the system in Theorem 2 according to the definition of (u, v), Theorem 1, and Lemma 1. On the other hand, if (u, v) and p are the solution to the system (3.22)–(3.25) above plus the periodic boundary condition, then they satisfy all the equations in (1.1)–(1.2) and the incompressibility condition. So they are also solution to the original problem.

Notice that if we know $\mathbf{q}$, then the jump conditions for the pressure are all known and we can solve the pressure independently. After the pressure is solved, we can solve the velocity from (3.23) and (3.24). The three equations with the given jump conditions can be solved using the immersed interface method [17–19] in which a fast solver is invoked with modified right hand sides at grid points near or on the interface. This observation is the basis of our new method. The compatibility condition for the two augmented variables are the two equations in (3.25) (which means that the velocity is continuous across the interface). It is also important to mention that the incompressibility condition is used to obtain the pressure Poisson equation of (3.22).

Once the augmented variables ($[\mu u]$ and $[\mu v]$) and the augmented equations (the two equations in (3.25)) are chosen, the success of the numerical algorithm depends on how efficiently we can solve the augmented variables. Note that the augmented approaches have been developed for elliptic interface problems with a piecewise constant coefficient [8,20], and the fast algorithms for Poisson and biharmonic equations on irregular domains [3,10,24,23].

We assume that the domain Ω is a rectangle: $[a, b]\times[c, d]$, and use a uniform Cartesian grid

$$x_i = a + ih_x, \quad i = 0, 1, \ldots, m, \qquad h_x = \frac{b-a}{M},$$
$$y_j = c + jh_y, \quad j = 0, 1, \ldots, n, \qquad h_y = \frac{d-c}{N}.$$

For simplicity of discussion, we also assume that $M = N$.

We first choose a set of interface points, sometimes also called the control points, $\{\mathbf{X}_k\} = \{(X_k, Y_k)\}$, $k = 1,2,\ldots, N_b$, on the interface.[2] The auxiliary variable $\mathbf{q}(s) = (q_1(s), q_2(s))$ is defined, and the augmented equations (3.25) are discretized, at $\{\mathbf{X}_k\}$. We use upper case letters such as P_{ij}, U_{ij}, V_{ij}, $\mathbf{Q}_k$, for the discrete approximations at grid points and at the interface points, respectively. We use the bold face upper case letters without subscripts to represent the vectors formed by those discrete components.

Given an initial guess of $\mathbf{Q}$ at the interface points, we can approximate its first and second order tangential derivatives $\frac{\partial \mathbf{q}}{\partial \tau}$ and $\frac{\partial^2 \mathbf{q}}{\partial \tau^2}$. Thus all the jump conditions in (3.22)–(3.24) are known and we can solve (3.22)–(3.24) using the immersed interface method [17,18]. Note that the jump conditions for the pressure and the velocity are decoupled if we know $\mathbf{Q}$. Since the solution depends on $\mathbf{Q}$, the solution can be written as $\mathbf{P}(\mathbf{Q})$, $\widetilde{\mathbf{U}}(\mathbf{Q})$ and $\widetilde{\mathbf{V}}(\mathbf{Q})$.

If the computed $\widetilde{\mathbf{U}}(\mathbf{Q})$ and $\widetilde{\mathbf{V}}(\mathbf{Q})$ satisfy the two equations in (3.25), then $\mathbf{P}(\mathbf{Q})$, $\widetilde{\mathbf{U}}(\mathbf{Q})/\mu$ and $\widetilde{\mathbf{V}}(\mathbf{Q})/\mu$ are an approximate solution to the original system (1.1)–(1.2). Otherwise, we use an accurate linear interpolation scheme to evaluate the residual of the two equations in (3.25) which will be explained in Section 3.2.

3.1. The discrete system of equations in the matrix–vector form

Given a discrete approximation of (q_1, q_2) at $\{\mathbf{X}_k\}$, we can solve the first three equations (3.22)–(3.24) using the immersed interface method [18] to get an approximate solution: the pressure $\mathbf{P}(\mathbf{Q})$, the scaled velocity $\widetilde{\mathbf{U}}(\mathbf{Q})$ and $\widetilde{\mathbf{V}}(\mathbf{Q})$. Generally the computed velocity $(\widetilde{\mathbf{U}}(\mathbf{Q}), \widetilde{\mathbf{V}}(\mathbf{Q}))$ do not satisfy the two augmented equations in (3.25), that is, $(\mathbf{U}, \mathbf{V}) = (\widetilde{\mathbf{U}}/\mu, \widetilde{\mathbf{V}}/\mu)$ may not be continuous across the interface.

Let us assemble the discrete solution $\{P_{ij}\}$, $\{U_{ij}\}$, and $\{V_{ij}\}$ together as a big vector $\widetilde{\mathcal{U}}$ whose dimension is $3MN$. We denote also the vector of the discrete values of (q_1, q_2) at the interface points $\{\mathbf{X}_k\}$ by $\mathbf{Q}$ whose dimension is $2N_b$. Then the discrete solution of (3.22)–(3.24) given $\mathbf{Q}$ can be written as

$$A\widetilde{\mathcal{U}} + B\mathbf{Q} = \mathbf{F}_1 \tag{3.27}$$

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

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

$$[\mathbf{U}(\mathbf{Q})]|_{\Gamma} = ([\widetilde{\mathbf{U}}/\mu], [\widetilde{\mathbf{U}}/\mu])^{\mathrm{T}} = S\widetilde{\mathcal{U}} + E\mathbf{Q} - \mathbf{F}_2, \tag{3.28}$$

[2] In a front tracking method, $\{\mathbf{X}_k\}$ is a set of points that represents the interface, see [19], for examples in which a cubic spline is used. In a level set method, the interface points can be chosen as the orthogonal projections of irregular grid points on the interface, see [8,20], for example.

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 our algorithm. We want 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 (3.27) and (3.28) together we get

$$\begin{bmatrix} A & B \\ S & E \end{bmatrix} \begin{bmatrix} \widetilde{\mathcal{U}} \\ \mathbf{Q} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{bmatrix}. \tag{3.29}$$

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

$$(E - SA^{-1}B)\mathbf{Q} = \mathbf{F}_2 - SA^{-1}\mathbf{F}_1 = \overline{\mathbf{F}}. \tag{3.30}$$

After we solve (3.30) for $\mathbf{Q}$, we can get $\widetilde{\mathcal{U}}$ easily. Because the dimension of $\mathbf{Q}$ is much smaller than that of $\widetilde{\mathcal{U}}$, we expect to get a reasonably fast algorithm for the two-phase Stokes equations.

In implementation, we use the GMRES [30] iterative method to solve (3.30). The GMRES method only requires the matrix–vector multiplication, and thus the Schur complement, maybe a dense matrix, of $\mathbf{Q}$ does not have to be explicitly constructed. We explain below how to evaluate the right hand side $\overline{\mathbf{F}}$ of the Schur complement, and how to evaluate the matrix–vector multiplication needed in the GMRES iteration.

3.1.1. Evaluation of the right hand side of the Schur complement

First we set $\mathbf{Q} = \mathbf{0}$ and solve the de-coupled system (3.22)–(3.24), or (3.27) in the discrete form, to get $\widetilde{\mathcal{U}}(\mathbf{0})$ which is $A^{-1}\mathbf{F}_1$ from (3.27). From the interpolation scheme (3.28), we also have

$$[\mathbf{U}(\mathbf{0})]|_\Gamma = S\widetilde{\mathcal{U}}(\mathbf{0}) + E\mathbf{0} - \mathbf{F}_2 = S\widetilde{\mathcal{U}}(\mathbf{0}) - \mathbf{F}_2. \tag{3.31}$$

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

$$\begin{aligned} R(\mathbf{0}) &= (E - SA^{-1}B)\mathbf{0} - \overline{\mathbf{F}} = -\overline{\mathbf{F}} = -(\mathbf{F}_2 - SA^{-1}\mathbf{F}_1) \\ &= -\mathbf{F}_2 + S\widetilde{\mathcal{U}}(0) = [\mathbf{U}(\mathbf{0})]|_\Gamma \end{aligned} \tag{3.32}$$

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

3.1.2. 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 (3.22)–(3.24), or (3.27) in the discrete form, to get $\widetilde{\mathcal{U}}(\mathbf{Q})$.

Step 2: Interpolate $\widetilde{\mathcal{U}}(\mathbf{Q})$ using (3.28) to get $[\mathbf{U}(\mathbf{Q})]|_\Gamma$. Then the matrix–vector multiplication is

$$(E - SA^{-1}B)\mathbf{Q} = [\mathbf{U}(\mathbf{Q})]|_\Gamma - [\mathbf{U}(\mathbf{0})]|_\Gamma. \tag{3.33}$$

This is because

$$\begin{aligned} (E - SA^{-1}B)\mathbf{Q} &= E\mathbf{Q} - SA^{-1}B\mathbf{Q} \\ &= E\mathbf{Q} - S(A^{-1}\mathbf{F}_1 - \widetilde{\mathcal{U}}(\mathbf{Q})) \quad (\text{from } (3.27)), \\ &= E\mathbf{Q} + S\widetilde{\mathcal{U}}(\mathbf{Q}) - \mathbf{F}_2 + \mathbf{F}_2 - SA^{-1}\mathbf{F}_1 \\ &= [\mathbf{U}(\mathbf{Q})]|_\Gamma + \overline{\mathbf{F}} \quad (\text{from } (3.28)), \\ &= [\mathbf{U}(\mathbf{Q})]|_\Gamma - [\mathbf{U}(\mathbf{0})]|_\Gamma \quad (\text{from } (3.32)). \end{aligned}$$

Now we can see that a matrix–vector multiplication is equivalent to solving the coupled system (3.22)–(3.24), or (3.27) in the discrete form, to get $\widetilde{\mathcal{U}}$, and using an interpolation scheme (3.28) to get $[\mathbf{U}(\mathbf{Q})]|_\Gamma$ at the interface 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 the GMRES or other iterative methods.

3.2. The least squares interpolation scheme to compute the residual

The interpolation scheme (3.28) to evaluate $[\widetilde{\mathbf{U}}/\mu]$ and $[\widetilde{\mathbf{V}}/\mu]$ is crucial to the efficiency of the method. This is because the interpolation coefficients will affect the entries of the Schur complement, so its condition numbers. To reduce the number of iterations, it is important to couple the solutions on both sides of the interface using the jump conditions. Although the least squares interpolation scheme is not a new idea anymore, the details vary with problems. We explain the least squares interpolation scheme for computing (3.28) to see why we have the second matrix–vector equation in expression (3.29).

Given an approximation to the augmented variable $\mathbf{Q}$, we can solve the pressure, and then the velocity $(\mathbf{U}, \mathbf{V}) = (\widetilde{\mathbf{U}}/\mu, \widetilde{\mathbf{V}}/\mu)$ from (3.22)–(3.24). Since

$$\left[\frac{\widetilde{\mathbf{U}}}{\mu}\right] = \frac{\widetilde{\mathbf{U}}^+}{\mu^+} - \frac{\widetilde{\mathbf{U}}^-}{\mu^-},$$

we need to evaluate $\{\widetilde{U}^+\}$ and $\{\widetilde{U}^-\}$ at all interface points to get the vector $[\widetilde{\mathbf{U}}/\mu]$. To explain the idea, however, we just need to explain the interpolation scheme for $\widetilde{U}^-(\mathbf{X})$ at a point $\mathbf{X}$ on the interface. The interpolation scheme can be written as

$$\widetilde{U}^-(\mathbf{X}) = \sum_{k=0}^{k_s-1} \gamma_k \widetilde{U}_{i^*+i_k, j^*+j_k} + C, \tag{3.34}$$

where k_s is the number of grid points involved in the interpolation scheme, (x_{i^*}, y_{j^*}) is the closest grid point to $\mathbf{X}$, and C is a correction term. We should point it out that a one-sided interpolation scheme works poorly in the sense that the convergence speed is slow for the GMRES iteration. Below we discuss how to determine the coefficients γ_k and the correction term C using the information from both sides of the interface. Note that γ_k and C depend on $\mathbf{X}$, but for simplicity of the notation, we have omitted the dependency.

We use an un-determined coefficients method to determine the coefficients γ_k by minimizing the truncation error of (3.34) when $\widetilde{U}_{i^*+i_k,j^*+j_k}$ is substituted by the exact solution $\tilde{u}(x_{i^*+i_k}, y_{j^*+j_k})$. Using the local coordinates system centered at the point $\mathbf{X}$, see (2.5), and denoting the local coordinates of $(x_{i^*+i_k}, y_{j^*+j_k})$ as (ξ_k, η_k), we have the following from the Taylor expansion:

$$\begin{aligned}\tilde{u}(x_{i^*+k}, y_{j^*+k}) &= \tilde{u}(\xi_k, \eta_k)\\ &= \tilde{u}^{\pm} + \xi_k \tilde{u}^{\pm}_{\xi} + \eta_k \tilde{u}^{\pm}_{\eta} + \tfrac{1}{2}\xi_k^2 \tilde{u}^{\pm}_{\xi\xi}\\ &\quad + \xi_k \eta_k \tilde{u}^{\pm}_{\xi\eta} + \tfrac{1}{2}\eta_k^2 \tilde{u}^{\pm}_{\eta\eta} + \mathrm{O}(h^3),\end{aligned} \tag{3.35}$$

where the + or − sign is chosen depending on whether (ξ_k, η_k) lies on the + or − side of Γ, $\tilde{u}^{\pm}$, $\tilde{u}^{\pm}_{\xi}, \ldots, \tilde{u}^{\pm}_{\eta\eta}$ are evaluated at the local coordinates $(0,0)$, or $\mathbf{X} = (X, Y)$ in the original coordinates system (Fig. 2). Note that we should have used something like $\tilde{u}(X, Y) = \bar{\tilde{u}}(0,0)$ to distinguish the two coordinate systems. However, we omit the bars and use the same notation $\tilde{u}(X, Y) = \tilde{u}(0,0)$ for simplicity.

We carry out this expansion at all the grid points used in the interpolation scheme and plug (3.35) into (3.34). After collecting and arranging terms, we can write

$$\begin{aligned}\tilde{u}^-(\mathbf{X}) \approx{}& a_1\tilde{u}^- + a_2\tilde{u}^+ + a_3\tilde{u}^-_{\xi} + a_4\tilde{u}^+_{\xi} + a_5\tilde{u}^-_{\eta} + a_6\tilde{u}^+_{\eta}\\ &+ a_7\tilde{u}^-_{\xi\xi} + a_8\tilde{u}^+_{\xi\xi} + a_9\tilde{u}^-_{\eta\eta} + a_{10}\tilde{u}^+_{\eta\eta} + a_{11}\tilde{u}^-_{\xi\eta}\\ &+ a_{12}\tilde{u}^-_{\xi\eta} + C,\end{aligned} \tag{3.36}$$

where the a_i's are given by

$$\begin{aligned}
&a_1 = \sum_{k\in K^-} \gamma_k, \quad a_2 = \sum_{k\in K^+} \gamma_k,\\
&a_3 = \sum_{k\in K^-} \xi_k\gamma_k, \quad a_4 = \sum_{k\in K^+} \xi_k\gamma_k,\\
&a_5 = \sum_{k\in K^-} \eta_k\gamma_k, \quad a_6 = \sum_{k\in K^+} \eta_k\gamma_k,\\
&a_7 = \frac{1}{2}\sum_{k\in K^-} \xi_k^2, \quad \gamma_k a_8 = \frac{1}{2}\sum_{k\in K^+} \xi_k^2\gamma_k\\
&a_9 = \frac{1}{2}\sum_{k\in K^-} \eta_k^2\gamma_k, \quad a_{10} = \frac{1}{2}\sum_{k\in K^+} \eta_k^2\gamma_k,\\
&a_{11} = \sum_{k\in K^-} \xi_k\eta_k\gamma_k, \quad a_{12} = \sum_{k\in K^+} \xi_k\eta_k\gamma_k.
\end{aligned} \tag{3.37}$$

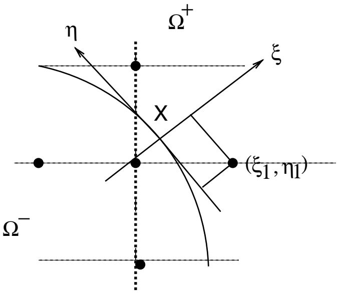

Fig. 2. A diagram of the local coordinates used in the interpolation (3.35).

Note that $\tilde{u}^+ = \tilde{u}^- + q_1$ and $\left[\frac{\partial\tilde{u}}{\partial n}\right]$ is known from (2.16). From [18,19], we also have the following interface relations:[3]

$$\begin{aligned}
\tilde{u}^+_{\eta} &= \tilde{u}^-_{\eta} + \frac{\mathrm{d}q_1}{\mathrm{d}\eta},\\
\tilde{u}^+_{\xi\xi} &= \tilde{u}^-_{\xi\xi} + \kappa\left[\frac{\partial\tilde{u}}{\partial n}\right] - \frac{\mathrm{d}^2 q_1}{\mathrm{d}\eta^2} + [p_x - g_1],\\
\tilde{u}^+_{\eta\eta} &= \tilde{u}^-_{\eta\eta} - \kappa\left[\frac{\partial\tilde{u}}{\partial n}\right] + \frac{\mathrm{d}^2 q_1}{\mathrm{d}\eta^2},\\
\tilde{u}^+_{\xi\eta} &= \tilde{u}^-_{\xi\eta} + \kappa\frac{\mathrm{d}q_1}{\mathrm{d}\eta} + \frac{\mathrm{d}}{\mathrm{d}\eta}\left[\frac{\partial\tilde{u}}{\partial n}\right],
\end{aligned} \tag{3.38}$$

where κ is the curvature of the interface at $\mathbf{X}$. Therefore we can express all the quantities from + side in (3.36) in terms of those from − side and the known quantities. Thus (3.34), when $\widetilde{U}_{i^*+i_k,j^*+j_k}$ is substituted by the exact solution $\tilde{u}(x_{i^*+i_k}, y_{j^*+j_k})$, can be written as

$$\begin{aligned}
\tilde{u}^-(\mathbf{X}) &\approx \sum_k \gamma_k \tilde{u}(x_{i^*+i_k}, y_{j^*+j_k}) + C\\
&= a_1\tilde{u}^- + a_2\tilde{u}^+ + a_3\tilde{u}^-_{\xi} + a_4\tilde{u}^+_{\xi} + a_5\tilde{u}^-_{\eta} + a_6\tilde{u}^+_{\eta} + a_7\tilde{u}^-_{\xi\xi}\\
&\quad + a_8\tilde{u}^+_{\xi\xi} + a_9\tilde{u}^-_{\eta\eta} + a_{10}\tilde{u}^+_{\eta\eta} + a_{11}\tilde{u}^-_{\xi\eta} + a_{12}\tilde{u}^+_{\xi\eta} + C\\
&= (a_1 + a_2)\tilde{u}^- + (a_3 + a_4)\tilde{u}^-_{\xi} + (a_5 + a_6)\tilde{u}^-_{\eta}\\
&\quad + (a_7 + a_8)\tilde{u}^-_{\xi\xi} + (a_9 + a_{10})\tilde{u}^-_{\eta\eta} + (a_{11} + a_{12})\tilde{u}^-_{\xi\eta}\\
&\quad + a_2[\tilde{u}] + a_4[\tilde{u}_{\xi}] + a_6[\tilde{u}_{\eta}] + a_8[\tilde{u}_{\xi\xi}]\\
&\quad + a_{10}[\tilde{u}_{\xi\eta}] + a_{12}[\tilde{u}_{\xi\eta}] + C.
\end{aligned}$$

To minimize the local truncation error, we should set the following linear system of equations for the coefficients γ_k by matching the terms of $\tilde{u}^-, \tilde{u}^-_{\xi}, \ldots, \tilde{u}^-_{\xi\eta}$:

$$\begin{aligned}
&a_1 + a_2 = 1, \quad a_3 + a_4 = 0,\\
&a_5 + a_6 = 0, \quad a_7 + a_8 = 0,\\
&a_9 + a_{10} = 0, \quad a_{11} + a_{12} = 0.
\end{aligned} \tag{3.39}$$

The system of equations for $\{\gamma_k\}$ is independent of jumps which means we can calculate $\{\gamma_k\}$ outside of the GMRES iteration. Once we have the coefficients, the correction term is

$$\begin{aligned}
C &= -(a_2[\tilde{u}] + a_4[\tilde{u}_{\xi}] + a_6[\tilde{u}_{\eta}] + a_8[\tilde{u}_{\xi\xi}] + a_{10}[\tilde{u}_{\xi\eta}] + a_{12}[\tilde{u}_{\xi\eta}])\\
&= -a_2 q_1 - a_4\left[\frac{\partial\tilde{u}}{\partial n}\right] - a_6\frac{\mathrm{d}q_1}{\mathrm{d}\eta}\\
&\quad - a_8\left(\left[\frac{\partial\tilde{u}}{\partial n}\right]\kappa - \frac{\mathrm{d}^2[\tilde{u}]}{\mathrm{d}\eta^2} + [p_x - g_1]\right)\\
&\quad - a_{10}\left(\frac{\mathrm{d}^2 q_1}{\mathrm{d}\eta^2} - \left[\frac{\partial\tilde{u}}{\partial n}\right]\kappa\right) - a_{12}\left(\frac{\mathrm{d}[\tilde{u}]}{\mathrm{d}\eta}\kappa + \frac{\mathrm{d}}{\mathrm{d}\eta}\left[\frac{\partial\tilde{u}}{\partial n}\right]\right).
\end{aligned} \tag{3.40}$$

We choose a neighborhood of $\{\mathbf{X}\}$ that contains more than six different grid points so that we have an

[3] Note that q_1 now is a quantity defined only on the interface, and we can only take its derivatives along the interface which is called surface derivative in the literature.

under-determined system. In our numerical tests, we choose $k_s = 12$, that is, we selected 12 closest grid points to $\mathbf{X} = (X, Y)$ as the interpolation stencil. We use the singular value decomposition (SVD) to find the least squares solution, which also has the least $l-2$ norm among all the solutions. In this way, the magnitude of the coefficients γ_k is controlled and balanced. The least squares interpolation plays an important role in the stability of the algorithm.

The only trade-off of the least squares interpolation is that we have to solve an under-determined system of equations. However, the size of the linear system is small and the coefficients can be pre-determined before the GMRES iteration. The extra time needed in dealing with the interface is usually less that 5% of the total CPU time and the percentage decreases as the mesh size (h) decreases.

Remark. By setting $a_1 + a_2 = 0$ and $a_3 + a_4 = 1$ while keeping other equations unchanged in (3.39), we can easily get the normal derivative of the solution u. This is the method that we used in Section 4 for the convergence rate analysis of the computed normal derivative of the velocity.

There are also a few issues about how to evaluate p_x and p_y at all grid points before we can solve the velocity. Taking p_x as an example, if (x_{i-1}, y_j), (x_i, y_j), (x_{i+1}, y_j) are on the same side of the interface, we can approximate p_x by the standard central difference,

$$(p_x)_{ij} \approx \frac{1}{2h}(P_{i+1,j} - P_{i-1,j}).$$

Otherwise, we use

$$(p_x)_{ij} \approx \begin{cases} \frac{1}{h}(P_{i,j} - P_{i-1,j}) \\ \quad \text{if } (x_{i-1}, y_j) \text{ and } (x_i, y_j) \text{ are on the same side,} \\ \frac{1}{h}(P_{i+1,j} - P_{i,j}) \\ \quad \text{if } (x_i, y_j) \text{ and } (x_{i+1}, y_j) \text{ are on the same side,} \\ \dfrac{P_{ij} - P_{lj} \mp [p] \mp [p_x](x_l - X^*_{ij})}{(x_i - x_l)} \\ \quad \text{otherwise.} \end{cases}$$

In the last case the interface cuts through both between x_{i-1} and x_i, and x_i and x_{i+1}. Let (X^*_{ij}, y_j) be one of the two intersections between the interface and the grid line $y = y_j$. The subscript $l = i-1$ or $l = i+1$ is chosen such that $|x_l - X^*_{ij}|$ is the smaller one. The sign in the expression above depends on which side of the interface the point (i,j) is on, and $[p_x]$ (or $[p_y]$ in the y-direction) are calculated at (X^*_{ij}, y_j) from

$$[p_x] = \left[\frac{\partial p}{\partial n}\right]\cos\theta - \left[\frac{\partial p}{\partial \tau}\right]\sin\theta,$$

$$[p_y] = \left[\frac{\partial p}{\partial n}\right]\sin\theta + \left[\frac{\partial p}{\partial \tau}\right]\cos\theta.$$

The analysis of such approximations to p_x (or p_y) is discussed in [19].

4. Numerical experiments and analysis

In this section, we present several numerical tests to check the order of accuracy and efficiency of the augmented method proposed in this paper. In the first three examples, we choose the interface as a unit circle and we know the analytic solution under different situations. It is important to mention that the reason for choosing the circular interface is just for the sake of constructing the exact solutions that satisfy Eqs. (1.1)–(1.3) and the jump conditions (2.7)–(2.11). The circular interface is not a restriction of our method. In fact, we will also show the results for more complicated interfaces in this section.

In all the numerical tests, the interface points are chosen in such a way that the interface mesh size Δs has the same order of magnitude as the Cartesian mesh size h. In other words, the ratio $\Delta s/h$ is a constant. When we reduce the grid mesh h by half during a grid refinement analysis process, we also reduce the interface mesh size Δs by half.

How many interface points that are needed to represent the interface depends on the complexity of the interface. The detailed analysis about the effects of the number of interface points can be found in [20]. The surface derivatives needed in the algorithms are computed using the cubic spline generated by the interface points $\{\mathbf{X}_k\}$, see [19]. The main cost of the algorithm is from the fast Poisson solver that requires $O(N^2 \log N)$ operations, where N is the number of grid lines in the x- and y-directions. Therefore, each GMRES iteration requires about $O(3N^2 \log N)$ operations. All the simulations are done with double precision.

Most of the computations are done on workstations or a Notebook PC's within a few seconds to a few minutes for stationary Stokes problems. The tolerance of GMRES iteration is set as $\epsilon = 10^{-6}$. The GMRES is set to re-started after N_b iterations, where N_b is the number of interface points where $\mathbf{Q}$ is defined. In almost all the cases, the GMRES converges in much fewer steps than N_b. The initial guess for the GMRES iteration is chosen as zero vector unless $\mathbf{0}$ is the exact solution, in which case a random vector is used. Throughout this section, the computational domain is $\Omega = [-2,2] \times [-2,2]$ unless it is stated otherwise.

Example 4.1. We start with a simple example where the velocity is smooth and the pressure is discontinuous across the interface. The exact velocity and the pressure are given by

$$u = y(x^2 + y^2 - 1), \quad (x,y) \in \Omega, \tag{4.41}$$

$$v = -x(x^2 + y^2 - 1), \quad (x,y) \in \Omega, \tag{4.42}$$

$$p = \begin{cases} 1, & \text{if } x^2 + y^2 \leqslant 1, \\ 0, & \text{if } x^2 + y^2 > 1. \end{cases} \tag{4.43}$$

The interface is the unit circle. The viscosity is

$$\mu = \begin{cases} 1, & \text{if } x^2 + y^2 \leqslant 1, \\ \frac{1}{2}, & \text{if } x^2 + y^2 > 1. \end{cases} \tag{4.44}$$

The external forcing term $\mathbf{g}$ is given by

$$g_1 = \begin{cases} -8y, & \text{if } x^2+y^2 \leqslant 1, \\ -4y, & \text{if } x^2+y^2 > 1, \end{cases} \tag{4.45}$$

$$g_2 = \begin{cases} 8x, & \text{if } x^2+y^2 \leqslant 1, \\ 4x, & \text{if } x^2+y^2 > 1, \end{cases} \tag{4.46}$$

which has a finite jump across the interface. The normal and tangential force density are

$$\hat{f}_1 = [p] - 2\left[\mu\frac{\partial \mathbf{u}}{\partial n}\cdot\mathbf{n}\right] = -1, \tag{4.47}$$

$$\hat{f}_2 = -\left[\mu\frac{\partial \mathbf{u}}{\partial n}\cdot\boldsymbol{\tau}\right] - \left[\mu\frac{\partial \mathbf{u}}{\partial \tau}\cdot\mathbf{n}\right] = -1, \tag{4.48}$$

calculated from (2.7) and (2.9), respectively.

In Table 1, we show the result of the convergence rate analysis. We use the Dirichlet boundary condition for the velocity $\mathbf{u}$, and the Neumann boundary condition $\frac{\partial p}{\partial n} = 0$ for the pressure when we solve the three Poisson equations (3.22)–(3.24). Similarly, the Dirichlet boundary condition for the velocity and Neumann boundary condition for the pressure are used in other examples when the exact solution is not periodic in this section. In [7,13], the correct boundary conditions for the pressure has been derived. The Neumann boundary conditions for the pressure is coupled with the velocity. Since the augmented approach proposed in this paper is an iterative method, we can use the updated pressure and velocity to approximate the boundary condition as proposed in [7,13]. The fast Poisson solver that we use if from the Fishpack which allows Dirichlet, Neumann, and periodic boundary conditions along each side of a rectangular domain. The errors in Table 1 are measured in the maximum norm at all grid points, for example,

$$E_{\mathbf{u}}(N) = \frac{1}{2}\left(\max_{0\leqslant i,j\leqslant N}|U_{ij} - u(x_i,y_j)| + \max_{0\leqslant i,j\leqslant N}|V_{ij} - v(x_i,y_j)|\right), \tag{4.49}$$

where $u(x_i,y_j)$ is the exact solution at (x_i,y_j) while U_{ij} is the approximate solution and so on. In all the tables in this section, N is the number of grid lines used in both x- and y-directions. The ratio

$$p\text{-order} = \frac{\log(E_p(N)/E_p(2N))}{\log 2}, \tag{4.50}$$

is an indication of the order of accuracy for the pressure. On average, we observe second order convergence for all the quantities. The last column is the number of iterations (No.) of the GMRES method. We can see that the number of iterations remains roughly the same as we halve the mesh size h.

The errors usually do not decrease monotonically as we refine the grid, see [20]. To be more precise, we use linear regression analysis to find the approximate order of accuracy. In Fig. 3, we show the error plot on a log–log scale for the pressure and the velocity versus the grid spacing $h = h_x = h_y$, which shows that, from the slopes, the average convergence rate of the pressure and the velocity are 1.9187 and 1.9416, respectively. The mesh size varies from $N = 100$ to $N = 320$ according to $N = 100 + 5k$, $k = 0, 1, \ldots, 44$. The order of accuracy for the normal velocity $\partial u/\partial n$ is 2.1928 from the linear regression analysis.

Example 4.2. In the first example, while the pressure is discontinuous, the velocity is smooth and vanishes at the interface. Thus the exact solution to the augmented variable $\mathbf{q} = [\mu\mathbf{u}]$ is zero which makes the problem easier to compute. In this example, we keep all the quantities u, v, p in (4.41)–(4.48) inside the unit circle unchanged, but set both pressure and the velocity outside the circle to be zero. Therefore, the periodic boundary conditions are satisfied.

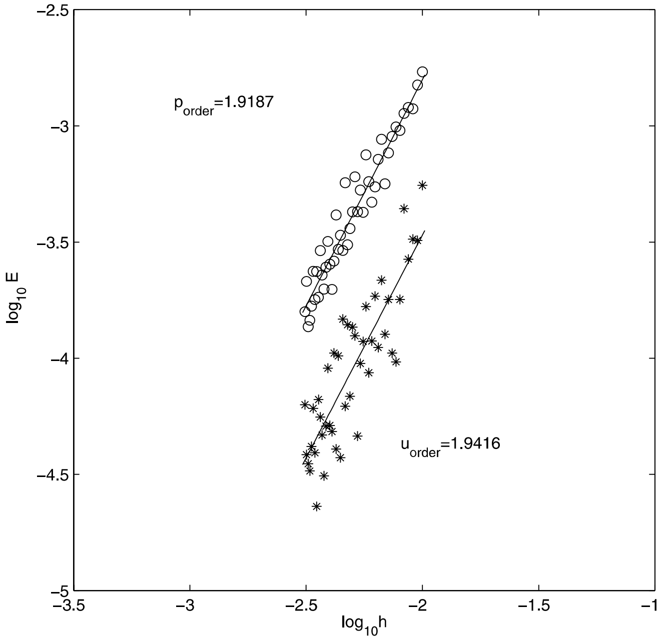

Fig. 3. Linear regression analysis of the convergence order in log–log scale for the pressure and the velocity for Example 4.1. The order of accuracy for the pressure and the velocity are 1.9187 and 1.9416, respectively.

Table 1
Numerical results and convergence analysis for Example 4.1

N	E_p	p-order	$E_{\mathbf{u}}$	$\mathbf{u}$-order	$E_{\partial\mathbf{u}/\partial n}$	$\frac{\partial\mathbf{u}}{\partial n}$-order	No.
32	8.2573×10^{-3}		6.5931×10^{-3}		2.4605×10^{-2}		8
64	3.0540×10^{-3}	1.4353	1.7372×10^{-3}	1.9242	6.0768×10^{-3}	2.0176	9
128	9.4747×10^{-4}	1.6883	3.9504×10^{-4}	2.1367	1.6713×10^{-3}	1.8623	8
256	2.6866×10^{-4}	1.8183	8.2274×10^{-5}	2.2635	4.1920×10^{-4}	1.9953	7
512	7.4314×10^{-5}	1.8541	2.5053×10^{-5}	1.7155	1.0445×10^{-4}	2.0048	7

Table 2
Numerical results and convergence analysis for Example 4.2

N	E_p	p-order	$E_{\mathbf{u}}$	$\mathbf{u}$-order	$E_{\partial\mathbf{u}/\partial n}$	$\frac{\partial\mathbf{u}}{\partial n}$-order	No.
32	8.4430×10^{-3}		3.4549×10^{-3}		2.8308×10^{-2}		9
64	2.8405×10^{-3}	1.5716	8.8800×10^{-4}	1.9600	6.1798×10^{-3}	2.1956	10
128	8.0952×10^{-4}	1.8110	2.2666×10^{-4}	1.9700	1.8260×10^{-3}	1.7589	11
256	2.5417×10^{-4}	1.6713	4.7693×10^{-5}	2.2487	5.3612×10^{-4}	1.7681	12
512	1.4086×10^{-5}	2.1296	1.4086×10^{-5}	1.7595	1.2538×10^{-4}	2.0962	13

The velocity is non-smooth and the jump in the normal velocity is not a constant along the interface. Now the normal force density component is still $\hat{f}_1 = -1$, but the tangential force density component is

$$\hat{f}_2 = -\left[\mu\frac{\partial\mathbf{u}}{\partial n}\cdot\boldsymbol{\tau}\right] - \left[\mu\frac{\partial\mathbf{u}}{\partial\tau}\cdot\mathbf{n}\right] = -2. \tag{4.51}$$

The external force is also adjusted to $\mathbf{g} = \mathbf{0}$ outside of the circle. Thus, there is a finite jump in $\mathbf{g}$ across the interface as well. In Table 2, we show the result of the convergence analysis. Once again we observe second order accuracy on average. Only a few iterations are needed in the GMRES iteration and the number changes little as N increases. The results of the linear regression analysis for the pressure and the velocity are given in Fig. 4 which confirms average second order accuracy for both the pressure and the velocity.

Example 4.3. In previous examples, the force density components are constants, and $[\mu\frac{\partial^2\hat{u}}{\partial\tau^2}] = 0$. In this example, we construct the exact solutions in such a way that all the jumps and their derivatives along the interface are nonconstant functions. The exact velocity and the pressure are given by

$$u = \begin{cases} \dfrac{y}{4}, & \text{if } x^2+y^2 \leqslant 1, \\ \dfrac{y}{4}(x^2+y^2), & \text{if } x^2+y^2 > 1, \end{cases} \tag{4.52}$$

$$v = \begin{cases} -\dfrac{x}{4}(1-x^2), & \text{if } x^2+y^2 \leqslant 1, \\ -\dfrac{xy^2}{4}, & \text{if } x^2+y^2 > 1, \end{cases} \tag{4.53}$$

$$p = \begin{cases} \left(-\dfrac{3}{4}x^3+\dfrac{3}{8}x\right)y, & \text{if } x^2+y^2 \leqslant 1, \\ 0, & \text{if } x^2+y^2 > 1. \end{cases} \tag{4.54}$$

The external forcing term $\mathbf{g}$ is

$$g_1 = \begin{cases} \left(-\dfrac{9}{4}x^2+\dfrac{3}{8}\right)y, & \text{if } x^2+y^2 \leqslant 1, \\ -2\mu^+ y & \text{if } x^2+y^2 > 1, \end{cases} \tag{4.55}$$

$$g_2 = \begin{cases} -\dfrac{3}{4}x^3+\dfrac{3}{8}x-\dfrac{3\mu^-}{2}x, & \text{if } x^2+y^2 \leqslant 1, \\ \dfrac{\mu^+}{2}x, & \text{if } x^2+y^2 > 1, \end{cases} \tag{4.56}$$

which is discontinuous across the interface. The force density components in the normal and tangential directions are

$$\hat{f}_1 = \left(\frac{3}{4}\cos^3\theta - \frac{3}{8}\cos\theta\right)\sin\theta - \frac{3}{2}[\mu]\cos^3\theta\sin\theta, \tag{4.57}$$

$$\hat{f}_2 = \frac{1}{2}\mu^+ + \frac{3}{4}[\mu]\cos^2\theta(1-2\cos^2\theta), \tag{4.58}$$

respectively. All the jump conditions (2.7)–(2.10) are satisfied. We use the exact Dirichlet boundary condition for the velocity and the homogeneous Neumann boundary condition for the pressure.

In Table 3, we show the convergence rate analysis for different jump in μ. We scale the problem such that $\max\{\mu^-,\mu^+\} = 1$, and test our results for $\mu^-/\mu^+ = 10$, 10^{-3}, and 10^3. While the accuracy does depend on μ^-/μ^+, the average convergence rates are about the same (second order accurate). Note that, there are two very different scales for the problems in Table 3(b) and (c). The number of iterations seems to be dependent on the ratio μ^-/μ^+ but not on the mesh size N. Note that the number of iterations of the GMRES method in Table 3(c) is many more than that in Table 3(b). One intuitive explanation is the following. When $\mu^+ \ll \mu^-$, the solution inside the circle is much smoother than that outside. In other words, it is more difficult to resolve the solution outside than that inside. Since there are more grid points outside of the interface, there are

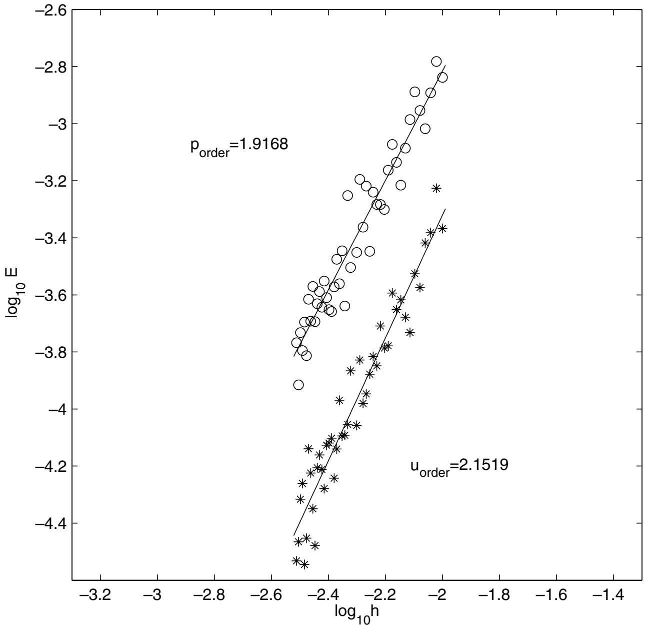

Fig. 4. Linear regression analysis for Example 4.2. The average convergence order for the pressure and the velocity are $p_{\text{order}} = 1.9168$, $u_{\text{order}} = 2.1519$.

Table 3
Numerical results and convergence analysis for Example 4.3

N	E_p	p-order	$E_{\mathbf{u}}$	$\mathbf{u}$-order	$E_{\partial\mathbf{u}/\partial n}$	$\frac{\partial\mathbf{u}}{\partial n}$-order	No.
(a) $\mu^- = 1,\ \mu^+ = 0.1$							
32	6.8928×10^{-2}		4.6299×10^{-2}		2.0170×10^{-2}		17
64	5.6851×10^{-3}	3.5998	3.4079×10^{-3}	3.7640	3.8114×10^{-3}	2.4038	17
128	2.2966×10^{-3}	1.3077	1.2068×10^{-3}	1.4977	8.7465×10^{-4}	2.1235	17
256	5.4715×10^{-4}	2.0695	2.6908×10^{-4}	2.1651	2.2514×10^{-4}	1.9579	14
512	1.5365×10^{-4}	1.8323	6.4921×10^{-5}	2.0513	5.6411×10^{-5}	1.9968	14
(b) $\mu^- = 0.001,\ \mu^+ = 1$							
32	1.3803×10^{-2}		8.1811×10^{-1}		13.323		12
64	4.1261×10^{-3}	1.7421	2.2177×10^{-1}	2.0017	2.8062	2.2371	13
128	1.0414×10^{-3}	1.9863	6.2257×10^{-2}	1.8328	7.7665×10^{-1}	1.8533	11
256	3.5892×10^{-4}	1.5368	1.4046×10^{-2}	2.1481	1.8162×10^{-1}	2.0969	11
512	7.0865×10^{-5}	2.3405	2.8175×10^{-3}	2.3177	4.7867×10^{-2}	1.9293	9
(c) $\mu^- = 1,\ \mu^+ = 0.001$							
32	0.6950		42.0260		1.4587		36
64	1.4356×10^{-2}	2.0017	9.4294×10^{-1}	5.4780	2.6627×10^{-1}	2.4537	30
128	6.5307×10^{-3}	1.1363	3.1469×10^{-1}	1.5832	5.9376×10^{-2}	2.1649	31
256	1.1757×10^{-3}	2.4737	4.6464×10^{-2}	2.7598	1.4749×10^{-2}	2.0093	36
512	3.0160×10^{-4}	1.8260	1.1697×10^{-3}	1.9900	3.4355×10^{-3}	2.1020	35

The CPU time for (c) are 1.5527, 2.8925, 6.1523, 17.275, and 44.248 s, respectively, for the 5 different grids.

more iterations needed in this case. If we increase the radius of the interface from $r = 1$ to $r = 1.5$, then the number of the GMRES iterations is down to 22 from 36 in Table 3(c) for $N = 32$. This is consistent with our explanation above. Another explanation is that: In the absence of information from the outer region, the inner solution would be arbitrary up to an additive constant. That constant is determined by consistency with the outer solution and takes some extra effort by the solver to pin down. This problem does not arise for $\mu^+ \gg \mu^-$ because in this case the outer solution is (relatively) small so that it is not difficult to approach the correct constant. In any case, the issue is less sensitive because of the scaling.

Example 4.4. In the examples above, the interface is chosen as a circle so that we can construct exact solutions to compare with the computed solution. In this example, we test our method for a complex geometry. The interface is given by

$$\rho = 0.35 + 0.1\sin(6\theta), \quad 0 \leqslant \theta \leqslant 2\pi, \tag{4.59}$$

in polar coordinates in the rectangular domain $[-1,1]\times[-1,1]$, see Fig. 5. The curvature of the interface varies both in the magnitude and the sign, and the source term is $\mathbf{g} = \mathbf{0}$. The force density is $\hat{f}_1 = 0.1\kappa$ and $\hat{f}_2 = -0.1$, respectively, where κ is the curvature. Periodic boundary conditions are used for all the variables. Since the analytic solution is not available, we compare the computed solution with the solution that is obtained from the finest grid, 512 by 512. In Table 4, we show the convergence rate analysis for the pressure and the velocity. We also listed the number of iterations, the CPU time in seconds, and the ratio of two consecutive errors. For a second order method, the ratio is between 4 and 5. The justification of such an analysis is given in [19]. The interface points are initially chosen from (4.59) with equally spaced $\Delta\theta$. Then they are redistributed with equally spaced arc-length by the cubic spline package developed in [19].

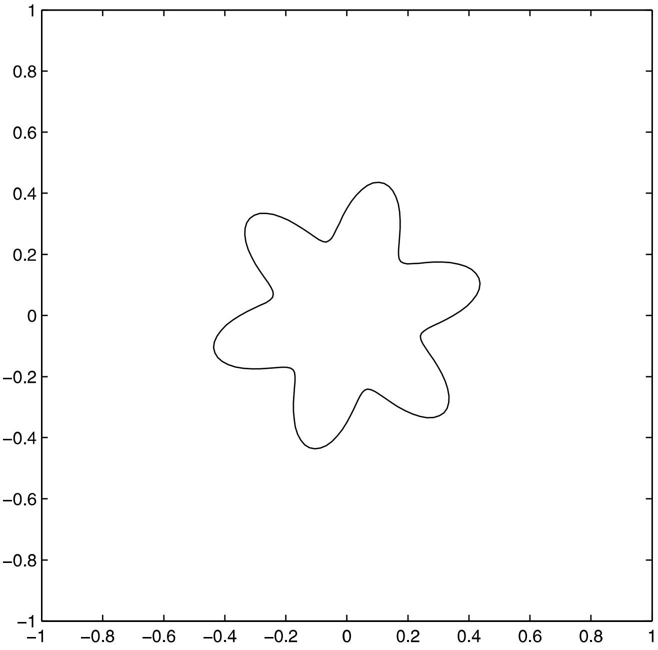

Fig. 5. The interface for Example 4.4.

Table 4
Numerical results and convergence analysis for Example 4.4 with $\mu^- = 1$ and $\mu^+ = 0.01$

N	E_p	p-ratio	$E_{\mathbf{u}}$	$\mathbf{u}$-ratio	CPU (s)	No.
32	16.149		6.6068×10^{-1}		3.5957	48
64	2.8549	5.6566	9.4029×10^{-2}	7.0263	8.9805	41
128	6.9682×10^{-1}	4.0970	2.5461×10^{-2}	3.6931	21.986	39
256	1.4587×10^{-4}	4.7770	3.7740×10^{-3}	6.7464	59.186	35

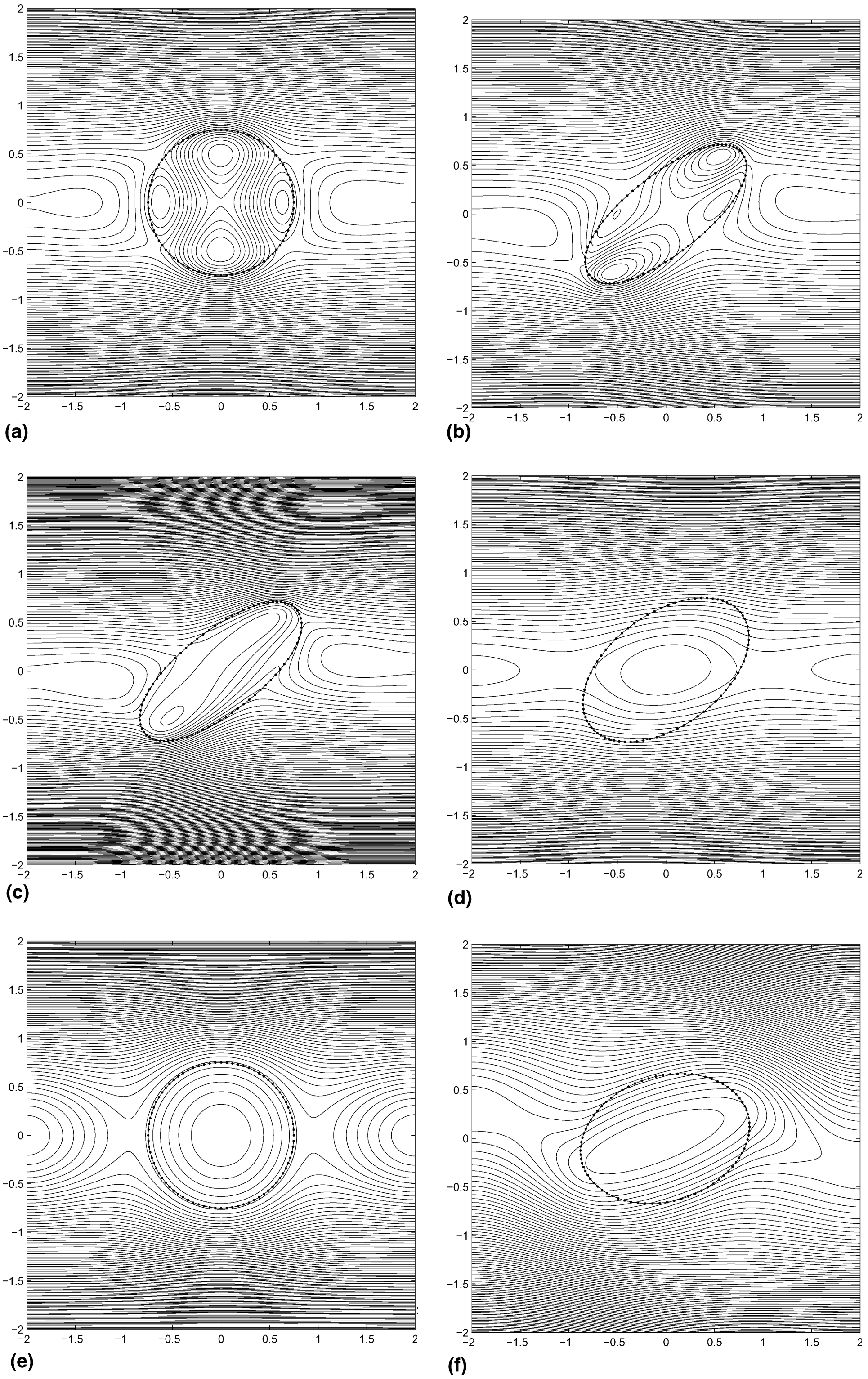

Fig. 6. Contour plot of stream-function of the shear flow with various viscosity ratio λ and Ca. In all of cases $Ca = 1.3898$, $\gamma = 0.4605$ except for (d) in which $Ca = 0.4695$, $\gamma = 1.4605$. (a) $\lambda = 0.04$ and $t = 0$. (b) $\lambda = 0.04$ and $t = 0.75$. (c) $\lambda = 0.08$ and $t = 0.75$. (d) $\lambda = 1$ and $t = 0.75$. (e) $\lambda = 6.4$ and $t = 0$. (f) $\lambda = 6.4$ and $t = 0.75$. The drop is the solid-dot line in the plots.

Compared with previous examples, the number of GMRES iterations is larger. This is not surprising since the curvature of the interface is large. Nevertheless, the expected second order accuracy on average is still observed.

Example 4.5. As a final example, we show the motion and deformation of a liquid drop in simple shear flow in two space dimensions. We refer the readers to [15] and the references therein for detailed description of the problem and physical explanations.

The set-up of the problem is as follows. A circular viscous drop with viscosity μ_2 is placed in a shear flow with viscosity μ_1. The velocity of the shear flow without drop is $\mathbf{u} = (Gy, 0)$, where G is the shear rate. The computational domain is $\Omega = [-2,2] \times [-2,2]$. The Dirichlet boundary condition $\mathbf{u}_{\partial\Omega} = (Gy, 0)$ is used along $\partial\Omega$; and Neumann boundary condition $\partial p/\partial n = 0$ is used for the pressure along $\partial\Omega$. The capillary number is defined as $Ca = Gr_0\mu_1/\gamma$, where r_0 is the radius of the circular drop, γ is the surface tension. The source strength is $\hat{f}_1 = \mathbf{f} \cdot \mathbf{n} = \gamma\kappa$, where κ is the curvature. The ratio of the viscosity is $\lambda = \mu_2/\mu_1$. A front tracking method is used to evolve the interface. The crucial parameters are the capillary number Ca and the viscosity ratio λ. The long-term behavior of the interface has different configurations depending on the values of Ca and λ. Typical long-term configurations include internal circulations and an asymptotic equilibrium state as explained below.

In Fig. 6 we present a sequence of contour plots of the stream function of the velocity for the two-phase Stokes flow in comparison with the results in [15]. In our simulation, the initial drop is a circle with $r_0 = 0.75$ and the shear rate is $G = 0.9143$. The parameters are chosen to mimic the set-up in [15]. We try to re-produce the results there except now it is in two space dimensions. Our results agree with those presented in [15] qualitatively. A quantitative comparison is not available. The stream-function is obtained by solving the Poisson equation $\Delta\psi = v_x - u_y$ with the boundary condition determined from the shear flow. When λ is small, we observe two regions of recirculating fluid within the drop. Generally, for smaller λ, we observe more deformation but larger angle (less tilting) between the longer axis of the drop and the x-axis. As λ gets larger, we see less deformation but smaller angle (more tilting) between the longer axis of the drop and the x-axis. All these agree with the results and analysis discussed in [15]. Note that for various parameters, internal circulations do not always occur. For instance, for circular drop at $t = 0$ with $\lambda = 1$, the solution of the pressure is piecewise constant and the solution of the velocity is simply the shear flow itself: $\mathbf{u} = (Gy, 0)$. While we do observe internal circulation for $Ca = 0.4695$ and $\gamma = 1.4605$ as presented in Fig. 6(d), this is not true for $Ca = 1.3898$ and $\gamma = 0.4605$. Also at $t = 0.75$, for all the simulations in Fig. 6 we have observed that the drop has reached its steady state.

5. Conclusion

In this paper, a new second order accurate finite difference method has been developed for incompressible stationary Stokes equations with a discontinuous viscosity in which the jump conditions for the pressure and the velocity are coupled together. The idea is to introduce two augmented variables that are only defined along the interface so that the jump conditions can be decoupled. The GMRES iterative method then is used to solve the Schur complement system for the augmented variables. The main cost in one step of the GMRES iteration is three calls to a fast Poisson solver. Numerical examples demonstrate the efficiency and accuracy of the method. The idea should be applicable to other interface problems with coupled jump conditions. One of the remaining open questions is how to use a preconditioning technique for the Schur complement system.

Acknowledgments

The first author is partially supported by NSF grants DMS-0201094 and DMS-0412654. The first and second author are supported by an ARO under grant number 43751-MA. The third author is partially supported by the National Science Council of Taiwan under research grant NSC-92-2115-M-009-012. We would like to thank Drs. X.-B. Lin and S. Lubkin of North Carolina State University for useful discussions. We would like also to thank the referees for their useful suggestions.

References

[1] Almgren A, Bell J, Colella P, Marthaler T. A Cartesian grid projection method for the incompressible Euler equations in complex geometries. SIAM J Sci Comput 1997;18:1289–309.

[2] Biros G, Ying L, Denis Z. An embedded boundary integral solver for the Stokes equations. J Comput Phys 2004;196:591–626.

[3] Chen G, Li Z, Lin P. A fast finite difference method for biharmonic equations on irregular domains. CRSC-TR04-09, North Carolina State University; 2004.

[4] Cortez R. The method of regularized Stokeslets. SIAM J Sci Comput 2001;23:1204–25.

[5] Fogelson AL, Peskin CS. Numerical solution of the three dimensional Stokes equations in the presence of suspended particles. In: Proceedings of the SIAM conference on multi-phase flow. SIAM, June 1986.

[6] Greengard L, Kropinski MC, Mayo A. Integral equation methods for Stokes flow and isotropic elasticity in the plane. J Comput Phys 1996; 125:403–14.

[7] Guermond JL, Shen J. A new class of truly consistent splitting schemes for incompressible flows. J Comput Phys 2003;192:262–76.

[8] Hou T, Li Z, Osher S, Zhao H. A hybrid method for moving interface problems with application to the Hele–Shaw flow. J Comput Phys 1997;134:236–52.

[9] Hou TY, Stredie VG, Wu TY. A 3D numerical method for studying vortex formation behind a moving plate. Commun Comput Phys 2006;1:207–28.

[10] Hunter J, Li Z, Zhao H. Autophobic spreading of drops. J Comput Phys 2002;183:335–66.

[11] Ito K, Li Z. Interface relations for Stokes equations with discontinuous viscosity and singular sources. Appl Math Lett 2006;19(3): 229–34.

[12] Johansen H, Colella P. A Cartesian grid embedded boundary method for Poisson equations on irregular domains. J Comput Phys 1998;147: 60–85.

[13] Johnston H, Liu J. Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term. J Comput Phys 2004;188:221–59.

[14] Kang M, Fedkiw R, Liu X. A boundary condition capturing method for multiphase incompressible flow. J Sci Comput 2000;15:323–60.

[15] Kenedy MR, Pozrikidis C, Skalak R. Motion and deformation of liquid drops and the rheology of dilute emulsions in shear flow. Comput Fluids 1994;23:251–78.

[16] Lai M-C, Li Z. A remark on jump conditions for the three-dimensional Navier–Stokes equations involving an immersed moving membrane. Appl Math Lett 2001;14:149–54.

[17] LeVeque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J Numer Anal 1994;31:1019–44.

[18] LeVeque RJ, Li Z. Immersed interface method for Stokes flow with elastic boundaries or surface tension. SIAM J Sci Comput 1997;18: 709–35.

[19] Li Z. The Immersed interface method — a numerical approach for partial differential equations with interfaces. PhD thesis, University of Washington; 1994.

[20] Li Z. A fast iterative algorithm for elliptic interface problems. SIAM J Numer Anal 1998;35:230–54.

[21] Li Z. An overview of the immersed interface method and its applications. Taiwan J Math 2003;7:1–49.

[22] Li Z, Wan X, Ito K, Lubkin S. An augmented pressure boundary condition for a Stokes flow with a non-slip boundary condition. Commun Comput Phys 2006;1(5):874–85.

[23] Li Z, Wang C. A fast finite difference method for solving Navier–Stokes equations on irregular domains. J Commu Math Sci 2003;1: 180–96.

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

[25] Lubkin SR, Li Z. Force and deformation on branching rudiments: cleaving between hypotheses. Biomech Model Mechanobiol 2002;1: 5–16.

[26] Mayo A, Peskin CS. An implicit numerical method for fluid dynamics problems with immersed elastic boundaries. Contemp Math 1991;141: 261–77.

[27] Osher S, Fedkiw R. Level set methods and dynamic implicit surfaces. New York: Springer; 2002.

[28] Peskin CS. Numerical analysis of blood flow in the heart. J Comput Phys 1977;25:220–52.

[29] Peskin CS. Lectures on mathematical aspects of physiology. Lect Appl Math 1981;19:69–107.

[30] Saad Y. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 1986; 7:856–69.

[31] Sethian J, Wiegmann A. Structural boundary design via level set and immersed interface methods. J Comput Phys 2000;163:489–528.

[32] Sethian JA. Level set methods and fast marching methods. 2nd ed. Cambridge University Press; 1999.

[33] Sussman M, Almgren A, Bell JB, Colella P, H Howell L, Welcome ML. An adaptive level set approach for incompressible two-phase flows. J Comput Phys 1999;148:81–124.

[34] Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. J Comput Phys 1994;114: 146–59.

[35] Teman R. Navier–Stokes equations. New York: North-Holland; 1977.

[36] Tornberg A-K, Engquist B. Segment projection method for interface tracking. Commu Pure App Math 2003;56:47–79.

[37] Tu C, Peskin CS. Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods. SIAM J Sci Stat Comput 1992;13:1361–76.

[38] Wiegmann A. The explicit jump immersed interface method and interface problems for differential equations. PhD thesis, University of Washington; 1998.

[39] Wiegmann A, Bube K. The immersed interface method for nonlinear differential equations with discontinuous coefficients and singular sources. SIAM J Numer Anal 1998;35:177–200.

[40] Wiegmann A, Bube K. The explicit jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions. SIAM J Numer Anal 2000;37:177–200.