

An Implementation of MAC Grid-Based IIM-Stokes Solver for Incompressible Two-Phase Flows

Zhijun Tan^{1,*}, K. M. Lim^{2,3} and B. C. Khoo^{2,3}

¹ *Guangdong Province Key Laboratory of Computational Science and School of Mathematics and Computational Science, Sun Yat-sen University, Guangzhou 510275, China.*

² *Singapore-MIT Alliance, 4 Engineering Drive 3, National University of Singapore, Singapore 117576, Singapore.*

³ *Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore.*

Received 16 October 2009; Accepted (in revised version) 22 February 2011

Available online 2 September 2011

Abstract. In this paper, a novel implementation of immersed interface method combined with Stokes solver on a MAC staggered grid for solving the steady two-fluid Stokes equations with interfaces. The velocity components along the interface are introduced as two augmented variables and the resulting augmented equation is then solved by the GMRES method. The augmented variables and/or the forces are related to the jumps in pressure and the jumps in the derivatives of both pressure and velocity, and are interpolated using cubic splines and are then applied to the fluid through the jump conditions. The Stokes equations are discretized on a staggered Cartesian grid via a second order finite difference method and solved by the conjugate gradient Uzawa-type method. The numerical results show that the overall scheme is second order accurate. The major advantages of the present IIM-Stokes solver are the efficiency and flexibility in terms of types of fluid flow and different boundary conditions. The proposed method avoids solution of the pressure Poisson equation, and comparisons are made to show the advantages of time savings by the present method. The generalized two-phase Stokes solver with correction terms has also been applied to incompressible two-phase Navier-Stokes flow.

AMS subject classifications: 65N06, 35R05, 65M12

Key words: Incompressible two-phase Stokes equations, two-phase Navier-Stokes equations, discontinuous viscosity, singular force, immersed interface method, Uzawa method, front tracking method.

*Corresponding author. *Email addresses:* tzhiij@mail.sysu.edu.cn (Z.-J. Tan), mpelimkm@nus.edu.sg (K. M. Lim), mpekbc@nus.edu.sg (B. C. Khoo)

1 Introduction

Many problems of fluid flows with interfaces between two different fluids have a broad range of natural, science, engineering, and physiological applications. A popular approach for solving such fluid problems on a Cartesian grid is Peskin's immersed boundary method (IBM) [28], which was originally developed to study the fluid dynamics of blood flow in a human heart [27], and was further developed and has been used in a wide variety of applications, particularly applied to biological problems where complex geometries and immersed elastic interfaces are present. Some examples include the deformation of red blood cell in a shear flow [7], swimming of organisms [10], platelet aggregation [11, 12], cochlear dynamics [2], biofilm processes [6], wood pulp fiber dynamics [33], and so on. A summary of the development of the immersed boundary method can be found in [28].

An alternative approach is the immersed interface method (IIM) which can capture the solution and its derivative jumps sharply and maintains second-order accuracy via incorporating the known jump conditions into the finite difference approximations near the interface. The IIM was originally proposed by LeVeque and Li [19] for solving elliptic equations, and was later extended to Stokes flow with elastic boundaries or surface tension [18]. The IIM was developed further for the Navier-Stokes equations in [16, 17, 23, 25, 38]. The IIM was also used in [4, 24, 30] for solving the two-dimensional streamfunction-vorticity equations on irregular domains. Xu and Wang [39] have extended the IIM to the 3D Navier-Stokes equation for simulating fluid-solid interaction. Other more applications on the IIM can be found in Li's recent review article [20] or the book by Li and Ito [21] and the references therein.

Recently, Li et al. [22] developed an augmented IIM for incompressible 2D Stokes flows with discontinuous viscosity. However, the method employed an explicit time stepping and a standard (not MAC) grid, so the time step is strictly restricted and the bi-periodic boundary condition also has to be assumed. A numerical method for solving the two-fluid Stokes equations with a moving immersed boundary was presented by Layton [15], which uses integral equations to reduce the two-fluid Stokes problem to the single-fluid case. In [35], Tan et al. developed an IIM for the Navier-Stokes equations with discontinuous viscosity across the interface based on the pressure increment projection method. For most biological flows, however, the Reynolds number is pretty low such that Stokes flow simulation is just simply more appropriate. The approach of [35] using Navier-Stokes equation directly to get the steady state solution for such flows is deemed generally impractical and much more expensive. Also using the method in [35] to approach the Stokes flow regime, needs to solve a pressure Poisson equation with discontinuous coefficient subject to Neumann boundary condition, which is a fairly large time-consuming step. On the other hand, to the best of our knowledge, no work on IIM for the steady two-phase Stokes flows involving moving interface with Dirichlet boundary conditions exists in the literature so far. The previously published work on IIM for two-phase Stokes flow with interface is based on solving three Poisson problems

on a regular non-staggered grid, which is seriously limited to the case of the bi-periodic boundary conditions [22]. Based on these considerations, in this study, the non-trivial implementation of IIM-Stokes solver is presented, where the fluid dynamics is described as a creeping flow using the two-phase Stokes equations. One of the advantages for such problems is that an efficient Stokes solver can be directly applied here based on the efficient preconditioners for saddle point systems and fast Poisson solver. The number of CG iterations and the number of GMRES iterations are shown to be independent of the mesh size, so the proposed IIM-Stokes solver is very efficient. Another advantage is that the proposed IIM-Stokes solver is flexible for different types of two-phase flow problems, which can be easily extended to solve for almost all other types of two-fluid flows (including the steady two-phase Navier-Stokes flows, unsteady two-phase Stokes flows and unsteady two-phase Navier-Stokes flows) with interfaces and singular forces. It needs only a simple extension from the present Stokes solver to the efficient generalized Stokes solver [9, 14, 31] in a fairly straightforward manner. So particular interest in this work is focused on steady two-phase Stokes flows with interfaces. In addition, the present IIM-Stokes solver can be used to solve the two-fluid problems subject to not only Dirichlet boundary conditions but also periodic boundary conditions. As such, the present method is very flexible for solving different types of two-phase flow problems in terms of different boundary conditions. The main objective of this paper is to develop an efficient and flexible IIM-Stokes solver to solve the two-phase Stokes equations. The implementation of the current IIM combined Stokes solver on a MAC grid for such two-phase flow problems is both new and non-trivial, which also avails the potential reader a choice of the associated numerical techniques on IIM combined with efficient fluid solvers for different applications. The present method has been applied to solve the incompressible two-phase Navier-Stokes equations with singular forces as a typical application.

The present IIM is based on the augmented strategy by introducing the velocity components along the interface as two augmented variables so that the jump conditions are decoupled. The augmented variables are determined to satisfy the continuous condition of the velocity across the interface and the augmented equations are solved by solving a small system of equations with the GMRES method. The jumps in pressure and velocity and the jumps in their derivatives are related to the augmented interface variables and/or the forces which are either prescribed for fixed interface problem or computed from the configuration of the moving interface and are next interpolated using cubic splines and then applied to the fluid through the jump conditions together with the augmented variables. The position of the moving interface is updated implicitly within each time step. The Stokes equations are discretized on a staggered Cartesian grid by a second order finite difference method for the pressure and velocity quantities via incorporating the spatial jump contributions and solved by the conjugate gradient Uzawa-type method. Fast solvers from the FISHPACK software library [1] are then used to solve the resulting discrete systems of the Poisson equations. In this work, the jump condition of the second-order spatial derivative for pressure has been incorporated. The numerical re-

sults show that second order accuracy for the velocity and nearly second order accuracy for the pressure are achieved.

The remaining part of the paper is organized as follows. In Section 2, the model of the steady incompressible two-fluid Stokes equations with interfaces is described, and the corresponding decoupled jump conditions are presented in Section 3. The numerical algorithm and numerical implementation are presented in Sections 4 and 5, respectively. In Section 6, we present some numerical results. Some concluding remarks are made in Section 7.

2 Two-phase Stokes equations

For simplicity, this paper first considers the steady two-phase Stokes flows with interfaces in 2D here. Let Ω be a two-dimensional bounded domain containing a material interface Γ , the steady incompressible Stokes equations formulated in the velocity-pressure variables are written as

$$\nabla p = \nabla \cdot \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (2.1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in \Omega, \quad (2.1b)$$

with boundary conditions

$$\mathbf{u}|_{\partial\Omega} = \mathbf{u}_b,$$

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, $\mathbf{G}(\mathbf{x}, t) = (g_1, g_2)^T$ (may be discontinuous) is an external forcing term such as gravity, and \mathbf{F} is a singular source which can have a Dirac delta function singularity,

$$\mathbf{F}(\mathbf{x}, t) = \int_{\Gamma} \mathbf{f}(s, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) ds. \quad (2.2)$$

Here $\mathbf{X}(s, t) = (X(s, t), Y(s, t))$ is the arc-length parametrization of the interface Γ , s is the arc-length, $\mathbf{f} = (f_1, f_2)^T$ is the force density, and $\delta(\cdot)$ is the Dirac delta function defined in the distribution sense. The interface Γ separates the fluid into two regions Ω^+ and Ω^- with $\Omega = \Omega^+ \cup \Gamma \cup \Omega^-$, where Ω^+ is used to express the exterior region of the interface, and Ω^- is enclosed by the interface. Eq. (2.1b) together with the Dirichlet boundary condition Eq. (2.2) leads to the compatibility condition that \mathbf{u}_b must satisfy:

$$\int_{\partial\Omega} \mathbf{u}_b \cdot \mathbf{n}_b dS = 0, \quad (2.3)$$

where \mathbf{n}_b is the outer unit normal to $\partial\Omega$. The reader is referred to Fig. 1 for an illustration of the problem. The motion of the moving interface satisfies

$$\frac{\partial}{\partial t} \mathbf{X}(s, t) = \mathbf{u}(\mathbf{X}, t) = \int_{\Omega} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) d\mathbf{x}. \quad (2.4)$$

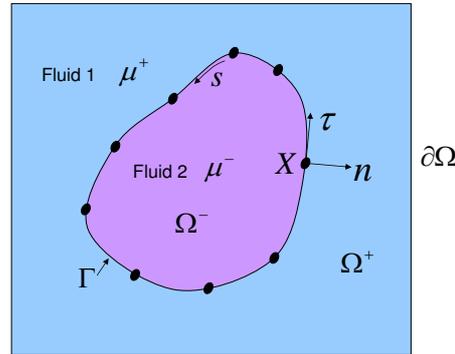


Figure 1: A typical domain with an interface represented by some Lagrangian control points. The domain Ω^+ and Ω^- are divided by a closed curve Γ across which the viscosity is discontinuous. Here \mathbf{n} and $\boldsymbol{\tau}$ are denoted as the unit outward normal and tangential directions of the boundary, respectively.

Eq. (2.2) and Eq. (2.4) represent the interaction between the interface and the fluid. In this model of moving interface, a fluid problem involving a closed elastic interface is considered, where the force strength \mathbf{f} exerted by elastic interface on the fluid has the form of

$$\mathbf{f}(s,t) = \frac{\partial}{\partial s} (T(s,t) \boldsymbol{\tau}(s,t)), \tag{2.5}$$

with the tension $T(s,t)$ given by

$$T(s,t) = T_0 \left(\left| \frac{\partial \mathbf{X}(s,t)}{\partial s_0} \right| - 1 \right). \tag{2.6}$$

Here, the tension coefficient T_0 is the stiffness constant which describes the elastic property of the elastic interface, and s_0 is a material parameter and equal to arc-length in the unstretched configuration of the interface. The vector tangential to Γ is given by $\boldsymbol{\tau}(s,t)$ in the form of

$$\boldsymbol{\tau}(s,t) = \frac{\partial \mathbf{X}}{\partial s} / \left| \frac{\partial \mathbf{X}}{\partial s} \right|.$$

Thus, the force density can be computed directly from the location \mathbf{X} of the interface Γ . An equivalent form of Eq. (2.5) can be written as

$$\mathbf{f}(s,t) = \left(\frac{\partial T}{\partial s} \right) \boldsymbol{\tau}(s,t) + T \kappa \mathbf{n}, \tag{2.7}$$

where κ is the curvature defined by $\partial \boldsymbol{\tau} / \partial s = \kappa \mathbf{n}$.

Over the whole domain, the viscosity μ is assumed to be a piecewise constant across the interface and can be written as

$$\mu(\mathbf{x}) = \begin{cases} \mu^+, & \text{if } \mathbf{x} \in \Omega^+, \\ \mu^-, & \text{if } \mathbf{x} \in \Omega^-, \end{cases} \tag{2.8}$$

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

3 Jump conditions across the interface

Let $\mathbf{n} = (n_1, n_2)$ and $\boldsymbol{\tau} = (\tau_1, \tau_2)$ be the unit outward normal and tangential vectors of the interface, respectively. The jump across the interface Γ is denoted $[\cdot]$ and defined for an arbitrary function $q(\mathbf{X})$ along the interface at \mathbf{X} by

$$[q] := \lim_{\epsilon \rightarrow 0^+} q(\mathbf{X} + \epsilon \mathbf{n}) - \lim_{\epsilon \rightarrow 0^+} q(\mathbf{X} - \epsilon \mathbf{n}). \quad (3.1)$$

When the viscosity is discontinuous across the interface Γ , the jump conditions for the pressure and velocity are coupled together [35] and summarized as follows:

$$\left[\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \right] \cdot \boldsymbol{\tau} + \left[\mu \frac{\partial \mathbf{u}}{\partial \boldsymbol{\tau}} \right] \cdot \mathbf{n} + \hat{f}_2 = 0, \quad (3.2a)$$

$$\left[\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \right] \cdot \mathbf{n} + \left[\mu \frac{\partial \mathbf{u}}{\partial \boldsymbol{\tau}} \right] \cdot \boldsymbol{\tau} = 0, \quad (3.2b)$$

$$[\mu \nabla \cdot \mathbf{u}] = 0, \quad [\mathbf{u}] = \mathbf{0}, \quad (3.2c)$$

$$[p] = 2 \left[\mu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \right] \cdot \mathbf{n} + \hat{f}_1, \quad (3.2d)$$

$$\left[\frac{\partial p}{\partial \mathbf{n}} \right] = [\mathbf{G}] \cdot \mathbf{n} + [\mu \Delta \mathbf{u}] \cdot \mathbf{n}, \quad (3.2e)$$

where \hat{f}_1 and \hat{f}_2 are the force density in the normal and tangential directions, i.e., $\hat{f}_1 = \mathbf{f} \cdot \mathbf{n}$ and $\hat{f}_2 = \mathbf{f} \cdot \boldsymbol{\tau}$, denoting $\hat{\mathbf{f}} = (\hat{f}_1, \hat{f}_2)$. Furthermore, (ζ, η) denotes the local coordinates associated with the directions of \mathbf{n} and $\boldsymbol{\tau}$ at the interface, and the transformed velocity field is defined as

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

By introducing the velocity components at the interface as two augmented variables, i.e.,

$$\mathbf{q}(s) = (q_1(s), q_2(s)) = \mathbf{u}(X(s, t), Y(s, t)),$$

the applicable jump conditions across the interface for pressure can be decoupled from (3.2a)-(3.2e) as follows (see [35] for details):

$$[p] = \hat{f}_1 - 2[\mu] \frac{\partial \mathbf{q}}{\partial \eta} \cdot \boldsymbol{\tau}, \quad [p_\eta] = \frac{\partial \hat{f}_1}{\partial \eta} - 2[\mu] \left(\frac{\partial^2 \mathbf{q}}{\partial \eta^2} \cdot \boldsymbol{\tau} + \kappa \frac{\partial \mathbf{q}}{\partial \eta} \cdot \mathbf{n} \right), \quad (3.3a)$$

$$[p_\zeta] = [\mathbf{G}] \cdot \mathbf{n} + \frac{\partial \hat{f}_2}{\partial \eta} + 2[\mu] \frac{\partial^2 \mathbf{q}}{\partial \eta^2} \cdot \mathbf{n} - 2[\mu] \kappa \frac{\partial \mathbf{q}}{\partial \eta} \cdot \boldsymbol{\tau}, \quad (3.3b)$$

$$[p_{\eta\eta}] = \frac{d}{d\eta} [p_\eta] - \kappa [p_\zeta], \quad [p_{\zeta\eta}] = \frac{d}{d\eta} \left[\frac{\partial p}{\partial \mathbf{n}} \right] + \kappa [p_\eta], \quad (3.3c)$$

$$[p_{\zeta\zeta}] = [\nabla \cdot \mathbf{G}] - [p_{\eta\eta}]. \quad (3.3d)$$

The corresponding applicable jump conditions across the interface for the velocity can also be obtained as follows:

$$[\tilde{\mathbf{u}}] = [\mu] \mathbf{q}, \quad [\tilde{u}_\xi] = \left(\hat{f}_2 + [\mu] \frac{\partial \mathbf{q}}{\partial \eta} \cdot \mathbf{n} \right) n_2 - [\mu] \left(\frac{\partial \mathbf{q}}{\partial \eta} \cdot \boldsymbol{\tau} \right) \tau_2, \quad (3.4a)$$

$$[\tilde{\mathbf{u}}_\eta] = [\mu] \frac{\partial \mathbf{q}}{\partial \eta}, \quad [\tilde{v}_\xi] = - \left(\hat{f}_2 + [\mu] \frac{\partial \mathbf{q}}{\partial \eta} \cdot \mathbf{n} \right) n_1 + [\mu] \left(\frac{\partial \mathbf{q}}{\partial \eta} \cdot \boldsymbol{\tau} \right) \tau_1, \quad (3.4b)$$

$$[\tilde{\mathbf{u}}_{\eta\eta}] = [\mu] \frac{\partial^2 \mathbf{q}}{\partial \eta^2} - \kappa [\tilde{\mathbf{u}}_\xi], \quad [\tilde{\mathbf{u}}_{\xi\eta}] = \frac{d}{d\eta} \left[\frac{\partial \tilde{\mathbf{u}}}{\partial n} \right] + [\mu] \kappa \frac{\partial \mathbf{q}}{\partial \eta}, \quad (3.4c)$$

$$[\tilde{\mathbf{u}}_{\xi\xi}] = -[\tilde{\mathbf{u}}_{\eta\eta}] + [p_\xi] \mathbf{n} + [p_\eta] \boldsymbol{\tau} - [\mathbf{G}]. \quad (3.4d)$$

It is noted from expressions (3.3a)-(3.4d) that the values of the jumps of the first and second order derivatives of velocity and pressure can be obtained by a simple coordinate transformation:

$$[q_x] = [q_\xi] n_1 + [q_\eta] \tau_1, \quad [q_y] = [q_\xi] n_2 + [q_\eta] \tau_2, \quad (3.5a)$$

$$[q_{xx}] = [q_{\xi\xi}] n_1^2 + 2[q_{\xi\eta}] n_1 \tau_1 + [q_{\eta\eta}] \tau_1^2, \quad (3.5b)$$

$$[q_{yy}] = [q_{\xi\xi}] n_2^2 + 2[q_{\xi\eta}] n_2 \tau_2 + [q_{\eta\eta}] \tau_2^2, \quad q = \tilde{\mathbf{u}}, p. \quad (3.5c)$$

From (3.3a)-(3.5c), it is noted that, if the augmented variable \mathbf{q} and singular force \mathbf{f} are both known, then all the jump conditions, say $[p]$, $[p_x]$, $[p_y]$, $[p_{xx}]$, $[p_{yy}]$, $[\tilde{\mathbf{u}}]$, $[\tilde{\mathbf{u}}_x]$, $[\tilde{\mathbf{u}}_y]$, $[\tilde{\mathbf{u}}_{xx}]$, $[\tilde{\mathbf{u}}_{yy}]$, are also known. In this work, the singular force is either prescribed for the fixed interface or computed from the configuration of the moving interface. Therefore, with the above strategy, the jump conditions for the velocity, pressure and their derivatives are decoupled in the sense that all the jump conditions are only dependent on the unknown augmented variables. As such, once the augmented variables are determined, the immersed interface method for incompressible two-fluid Stokes flows with interfaces can be implemented as before (for example as in [16]).

4 Numerical algorithm

The numerical algorithm is based on working with $\tilde{\mathbf{u}} = \mu \mathbf{u}$ instead of \mathbf{u} for Stokes equations as in [22] in order to easily incorporate the decouple jump conditions and use the direct Stokes solver, which is based on the conjugate gradient Uzawa-type method for the discretization of the Stokes equations with special treatment at the grid points near the interface. The spatial discretization is carried out on a standard marker-and-cell (MAC) staggered grid similar to that found in Tau [36] with mesh size $h = \Delta x = \Delta y$. With the MAC mesh, the pressure field is defined at the cell center (i, j) , where $i \in \{1, 2, \dots, N_x\}$ and $j \in \{1, 2, \dots, N_y\}$. The transformed velocity components \tilde{u} and \tilde{v} are defined at the vertical and horizontal edges of a cell, respectively. The original velocity components u and v have the same locations as \tilde{u} and \tilde{v} , respectively. The pressure and the velocity components \tilde{u} and \tilde{v} are arranged as in Fig. 2. An advantage of such a staggered grid is that

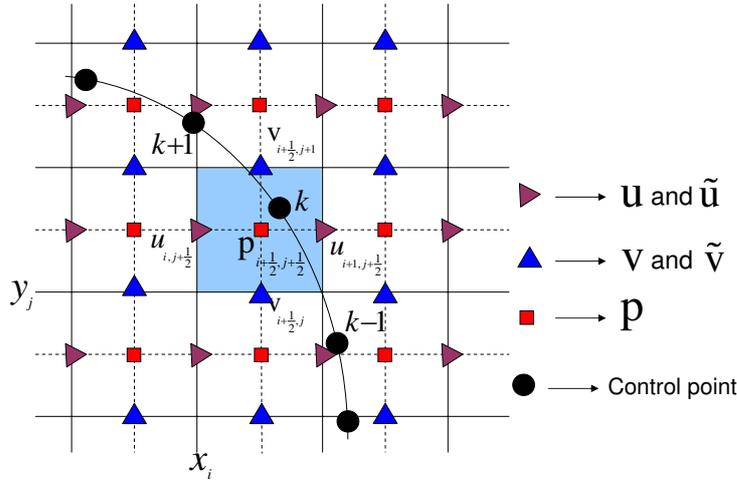


Figure 2: A diagram of the interface cutting through a staggered grid with a uniform mesh size h , where the velocity component u, \tilde{u} is at the left-right face of the cell and v, \tilde{v} is at the top-bottom face, and the pressure is at the cell center.

there is no need for pressure boundary conditions when dealing with the derivative of pressure since the pressure nodes are at the cell center.

4.1 Two-fluid Stokes solver involving correction terms

The discretization of Eqs. (2.1a)-(2.2) by second order MAC finite difference scheme leads to the following discrete saddle point system

$$\begin{aligned}
 & - \left(\frac{\tilde{u}_{i+1,j+\frac{1}{2}} - 2\tilde{u}_{i,j+\frac{1}{2}} + \tilde{u}_{i-1,j+\frac{1}{2}}}{h^2} + \frac{\tilde{u}_{i,j+\frac{3}{2}} - 2\tilde{u}_{i,j+\frac{1}{2}} + \tilde{u}_{i,j-\frac{1}{2}}}{h^2} \right) - C\{\Delta\tilde{u}\}_{i,j} \\
 & + \frac{p_{i+\frac{1}{2},j+\frac{1}{2}} - p_{i-\frac{1}{2},j+\frac{1}{2}}}{h} + C\{p_x\}_{i,j} = g_{i,j}^1, \tag{4.1a}
 \end{aligned}$$

$$\begin{aligned}
 & - \left(\frac{\tilde{v}_{i+\frac{3}{2},j} - 2\tilde{v}_{i+\frac{1}{2},j} + \tilde{v}_{i-\frac{1}{2},j}}{h^2} + \frac{\tilde{v}_{i+\frac{1}{2},j+1} - 2\tilde{v}_{i+\frac{1}{2},j} + \tilde{v}_{i+\frac{1}{2},j-1}}{h^2} \right) - C\{\Delta\tilde{v}\}_{i,j} \\
 & + \frac{p_{i+\frac{1}{2},j+\frac{1}{2}} - p_{i+\frac{1}{2},j-\frac{1}{2}}}{h} + C\{p_y\}_{i,j} = g_{i,j}^2, \tag{4.1b}
 \end{aligned}$$

$$\frac{\tilde{u}^{i+1,j+\frac{1}{2}} - \tilde{u}^{i,j+\frac{1}{2}}}{h} + \frac{\tilde{v}^{i+\frac{1}{2},j+1} - \tilde{v}^{i+\frac{1}{2},j}}{h} + C\{\nabla \cdot \tilde{\mathbf{u}}\}_{i,j} = 0. \tag{4.1c}$$

Note that the discretization of the Stokes equations at the grid points near the interface has been modified to account for the jump conditions across the interface due to the presence of singular forces at the interface. In Eqs. (4.1a)-(4.1c), $C\{\Delta\tilde{u}\}_{i,j}$, $C\{\Delta\tilde{v}\}_{i,j}$, $C\{p_x\}_{i,j}$, $C\{p_y\}_{i,j}$ and $C\{\nabla \cdot \tilde{\mathbf{u}}\}_{i,j}$ are the corresponding spatial correction terms, which are added to the finite difference equations and only non-zero at those points near the interface, to improve the accuracy of the local finite difference approximations. These corrections will

be evaluated later. In order to satisfy the discrete compatibility condition corresponding to (2.3) to thereby ensure the solvability of system Eqs. (4.1a)-(4.1c), a solvable perturbed system in a way similar to that in [18] via perturbing $C\{\nabla \cdot \tilde{\mathbf{u}}\}_{i,j}$ to $\hat{C}\{\nabla \cdot \tilde{\mathbf{u}}\}_{i,j}$ in Eq. (4.1c) is employed, where

$$\hat{C}\{\nabla \cdot \mathbf{u}\}_{i,j} = C\{\nabla \cdot \tilde{\mathbf{u}}\}_{i,j} - \bar{C}\{\nabla \cdot \tilde{\mathbf{u}}\}_{i,j}.$$

Here $\bar{C}\{\nabla \cdot \tilde{\mathbf{u}}\}_{i,j}$ is the mean value of the correction term $C\{\nabla \cdot \tilde{\mathbf{u}}\}_{i,j}$. The reader is referred to [18] for details. Let Δ_h , G^{MAC} and D^{MAC} be the standard central difference operator, the MAC gradient, and the divergence operators, respectively, then system (4.1a)-(4.1c) can be written as

$$-\Delta_h \tilde{\mathbf{u}} + G^{\text{MAC}} p = \mathbf{G} + \mathbf{C}_1, \tag{4.2a}$$

$$D^{\text{MAC}} \tilde{\mathbf{u}} = \mathbf{C}_2 - \hat{\mathbf{C}}_2, \tag{4.2b}$$

where the coefficients \mathbf{C}_1 and \mathbf{C}_2 are the spatial correction terms whose expressions will be given in the next subsection and $\hat{\mathbf{C}}_2$ is the perturbing term. Let $\mathbf{B}_1 = \mathbf{G}(\mathbf{x}) + \mathbf{C}_1$ and $\mathbf{B}_2 = \mathbf{C}_2 - \hat{\mathbf{C}}_2$, then (4.2a)-(4.2b) can be written in the matrix-vector form as

$$\begin{pmatrix} -\Delta_h & G^{\text{MAC}} \\ D^{\text{MAC}} & 0 \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{u}} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{pmatrix}. \tag{4.3}$$

There are some fast solvers for the solution of (4.3), such as the PCG method [8, 29], the PMINRES method [8, 29], the FFT-based method [5], and the multigrid method [8, 26, 29]. In this work, the fast solvers from FISHPACK [1] are utilized to incorporate the CG-Uzawa method. The Uzawa procedure for problems with immersed interfaces is analogous to the fast iterative method presented in [32, 36] and it consists of two steps:

Step 1 Solve $D^{\text{MAC}} \Delta_h^{-1} G^{\text{MAC}} p = \mathbf{B}_2 + D^{\text{MAC}} \Delta_h^{-1} \mathbf{B}_1$ for the pressure p .

Step 2 Solve $\Delta_h \tilde{\mathbf{u}} = \mathbf{B}_1 - G^{\text{MAC}} p$ for the transformed velocity $\tilde{\mathbf{u}}$.

Here, $D^{\text{MAC}} \Delta_h^{-1} G^{\text{MAC}}$ is the Schur complement of system (4.3). In Step 1, the system is solved by the conjugate gradient method (CG) in this work. In the CG method, each matrix-vector product with $D^{\text{MAC}} \Delta_h^{-1} G^{\text{MAC}}$ requires the inverse of Δ_h which corresponds to solving a Poisson equation. Several fast methods can be applied, such as the ICCG method, the FFT method and multigrid method. In the present work, the fast solvers from FISHPACK [1] are used. Once the pressure is obtained, the transformed velocity field $\tilde{\mathbf{u}}$ can be solved by the fast solvers from FISHPACK [1] via Step 2. The computational complexity for the fast Poisson solver from FISHPACK is $\mathcal{O}(M \log(M))$, where M is the number of interior grid points of the computational domain. The present CG method converges fast as discussed in [9, 32, 36] and the number of iterations in the CG method is small and almost independent of the mesh size which can be seen from the numerical examples in Section 6.

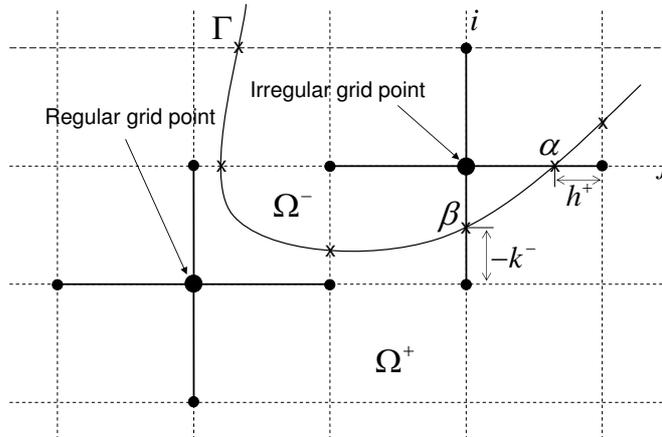


Figure 3: Interface and mesh geometry near the irregular grid point (i,j) , where the symbol "X" represents the intersection point between the interface and the grid lines.

4.2 Calculation of correction terms

Accordingly, the correction terms C_1 and C_2 are evaluated as follows:

$$C_1 = C\{\Delta\tilde{\mathbf{u}}\} - C\{\nabla p\}, \tag{4.4a}$$

$$C_2 = -C\{\nabla \cdot \tilde{\mathbf{u}}\}. \tag{4.4b}$$

To evaluate the correction term $C\{\Delta\tilde{\mathbf{u}}\}$ of (4.4a) at an irregular point (i,j) as depicted in Fig. 3, the jump conditions $[\tilde{\mathbf{u}}_x]$ and $[\tilde{\mathbf{u}}_{xx}]$ at the intersection point α of the interface with the grid lines, and $[\tilde{\mathbf{u}}_y]$ and $[\tilde{\mathbf{u}}_{yy}]$ at β of the interface with the grid lines, need to be computed. The correction term $C\{\Delta\tilde{\mathbf{u}}\}$ is calculated as follows:

$$C\{\Delta\tilde{\mathbf{u}}\}_{i,j} = -\frac{[\tilde{\mathbf{u}}] + h^+[\tilde{\mathbf{u}}_x]_\alpha + \frac{(h^+)^2}{2}[\tilde{\mathbf{u}}_{xx}]_\alpha}{h^2} - \frac{[\tilde{\mathbf{u}}] + k^-[\tilde{\mathbf{u}}_y]_\beta + \frac{(k^-)^2}{2}[\tilde{\mathbf{u}}_{yy}]_\beta}{h^2},$$

where $h^+ = x_{i+1} - x_\alpha$, $k^- = y_{j-1} - y_\beta$ and x_α and y_β are the x -coordinate of the intersection point α and the y -coordinate of the intersection point β as shown in Fig. 3, respectively. $\Delta\tilde{\mathbf{u}}$ is approximated at the irregular point (i,j) as

$$\Delta\tilde{\mathbf{u}}(i,j) = \Delta_h \tilde{\mathbf{u}}_{i,j} + C\{\Delta\tilde{\mathbf{u}}\}_{i,j} + \mathcal{O}(h).$$

Similarly, the other correction terms in (4.4a)-(4.4b) can be computed as follows

$$C\{\nabla \cdot \tilde{\mathbf{u}}\}_{i,j} = -\frac{[\tilde{u}] + h^+[\tilde{u}_x]_\alpha + \frac{(h^+)^2}{2}[\tilde{u}_{xx}]_\alpha}{h} + \frac{[\tilde{v}] + k^-[\tilde{v}_y]_\beta + \frac{(k^-)^2}{2}[\tilde{v}_{yy}]_\beta}{h},$$

$$C\{\nabla p\}_{i,j} = \left(-\frac{[p] + h^+[p_x]_\alpha + \frac{(h^+)^2}{2}[p_{xx}]_\alpha}{h}, \frac{[p] + k^- [p_y]_\beta + \frac{(k^-)^2}{2}[p_{yy}]_\beta}{h} \right).$$

4.3 Determination of the augmented variables

Assuming the augmented variable \mathbf{q} at \mathbf{X} is known, then the jump conditions for the velocity, pressure and their derivatives are known as a functional of this augmented variable. The transformed velocity field $\tilde{\mathbf{u}}$ at all the grid points can be then computed via the CG-Uzawa method as discussed in Section 4.1 with the incorporation of the jumps in the solutions and their derivatives into the difference schemes. The transformed velocity at the control points from the outside of the interface \mathbf{X} , $\tilde{\mathbf{U}}_k^+$, can be interpolated from the transformed velocity $\tilde{\mathbf{u}}^{n+1}$ at the grid points and can be written as

$$\tilde{\mathbf{U}}_k^+ = \tilde{\mathbf{U}}^+(\mathbf{X}) = \mathcal{B}^+(\tilde{\mathbf{u}}), \tag{4.5}$$

where \mathcal{B}^+ is the modified bilinear interpolation operator which includes the appropriate correction terms required to guarantee second order accuracy when the velocity is discontinuous, which can be found in Appendix A of [35]. Since the relationships between the singular forces and the jumps in the solution or its derivatives are linear and all the equations solved are linear in discrete form, Eq. (4.5) can be rewritten as

$$\tilde{\mathbf{U}}_k^+(\mathbf{q}) = \tilde{\mathbf{U}}_k^+(0) + \mathbf{A}^+ \mathbf{q}, \tag{4.6}$$

where $\tilde{\mathbf{U}}_k^{+,0}$ corresponds to the transformed velocity at the control points from the outside of the interface at \mathbf{X} obtained by solving Eqs. (2.1a)-(2.1b) with $\mathbf{q} = 0$. \mathbf{A}^+ is a $2N_b \times 2N_b$ matrix, where N_b is the number of control points. The vector $\mathbf{A}^+ \mathbf{q}$ is the transformed velocity at the control points from the outside of the interface $\mathbf{X}^{(k)}$ obtained by solving the following equations:

$$\nabla_h p_{\mathbf{q}} = \Delta_h \tilde{\mathbf{u}}_{\mathbf{q}} + \bar{\mathbf{C}}_1, \tag{4.7a}$$

$$\nabla_h \cdot \tilde{\mathbf{u}}_{\mathbf{q}} = \bar{\mathbf{C}}_2, \quad \tilde{\mathbf{u}}_{\mathbf{q}}|_{\partial\Omega} = 0, \tag{4.7b}$$

$$\mathbf{A}^+ \mathbf{q} = \mathcal{B}^+(\tilde{\mathbf{u}}_{\mathbf{q}}). \tag{4.7c}$$

Here, $\bar{\mathbf{C}}_1$ and $\bar{\mathbf{C}}_2$ are the correction terms which only take into account the contribution of \mathbf{q} at the interface without the contribution of $\hat{\mathbf{f}}$ to the jump conditions. Note that the relation $\tilde{\mathbf{U}}_k^+(\mathbf{q}) = \mu^+ \mathbf{q}$, when \mathbf{q} is satisfied exactly, then the augmented system Eq. (4.6) can be further written as

$$(\mathbf{A}^+ - \mu^+ I) \mathbf{q} = -\tilde{\mathbf{U}}_k^+(0). \tag{4.8}$$

Therefore, the augmented variable \mathbf{q} at control points is determined by solving Eq. (4.8).

Note that the matrix \mathbf{A}^+ depends on only the location of the interface. For fixed interface problem, the coefficient matrix \mathbf{A}^+ is a constant matrix at every time step, so the augmented system (4.8) can be solved by the LU method. For moving interface problem, the matrix \mathbf{A}^+ changes at each time step and have to be formed at every time step. To avoid generating \mathbf{A}^+ , the GMRES method is employed to solve the linear system (4.8) iteratively in this work. Each iteration of GMRES method requires a matrix-vector product which can be obtained by solving (4.7a)-(4.7c). In the present computations, only a few iterations are needed in the GMRES iteration, so the algorithm is effective.

4.4 Numerical implementation

In this section, the implementation of the proposed algorithm is described. For the fixed interface, the algorithm for finding $\tilde{\mathbf{u}}$, \mathbf{u} , p and the augmented variable \mathbf{q} can be summarized as follows:

Algorithm I: fixed interface

- Step 1 Compute the right hand side of (4.8) by calculating $\tilde{\mathbf{U}}_k^+(0)$.
- Set $\mathbf{q}=0$ and solve (2.1a)-(2.1b) for the velocity at all the grid points.
 - Interpolate the transformed velocity at the control points \mathbf{X}_k from the outside of the interface as in (4.5).
- Step 2 Compute the augmented variable \mathbf{q} by solving (4.8) using the LU method or the GMRES method.
- Step 3 Compute $\tilde{\mathbf{u}}$ and p using the Stokes solver as described in Section 4.1. The velocity field \mathbf{u} can be directly obtained from the transformed velocity field $\tilde{\mathbf{u}}$.

For moving interface problem, the position of the moving interface is updated implicitly within each time step [35] and the BFGS method [34] is employed to solve the resulting nonlinear system as in [16]. Given the location of the control points \mathbf{X}^n , the transformed velocity $\tilde{\mathbf{u}}^n$, the algorithm for computing the transformed velocity $\tilde{\mathbf{u}}^{n+1}$ that satisfies the continuous condition of the velocity at the interface, pressure p^{n+1} and the location of the control points \mathbf{X}^{n+1} can be described as follows:

Algorithm II: moving interface

- Step 1 Set $k:=0$, make an initial guess for \mathbf{X}^{n+1} , i.e., $\mathbf{X}^{(0)}$ as $\mathbf{X}^{(0)} = 2\mathbf{X}^n - \mathbf{X}^{n-1}$ and set the inverse Jacobian $\mathbf{B}_0^{n+1} = \mathbf{B}_k^n$. At the first time step, the inverse Jacobian is initialized to the identity matrix \mathbf{I} .
- Step 2
- Compute the force strength along the moving interface using expression (2.5).
 - Compute the augmented variable \mathbf{q} at the moving interface to satisfy the continuous condition of the velocity \mathbf{u} . That is, calculate the right hand side vector of (4.8). Then solve for the small system of Eq. (4.8) using the GMRES method to obtain the augmented variable \mathbf{q} along the interface.
- Step 3
- Compute $\tilde{\mathbf{u}}^{n+1}$ and p^{n+1} using the Stokes solver. This step involves computing the appropriate correction terms as described in Section 4.1, the velocity field \mathbf{u}^{n+1} can be also obtained from the transformed velocity field $\tilde{\mathbf{u}}^{n+1}$.
 - Compute the velocity $\mathbf{u}^{n+1}(\mathbf{X}^{(k)})$ at $\mathbf{X}^{(k)}$ based on the interpolated transformed velocity $\tilde{\mathbf{U}}_k^+(\mathbf{X}^{(k)})$ at the control points from the outside of the interface $\mathbf{X}^{(k)}$, which is interpolated from the transformed velocity $\tilde{\mathbf{u}}^{n+1}$ at the surrounding grid points.
- Step 4
- Evaluate $Q(\mathbf{X}^{(k)})$.
 - If $\|Q^{(k)}\| < \varepsilon$ then $\mathbf{X}^{n+1} = \mathbf{X}^{(k)}$ and stop the iteration. Otherwise, update $\mathbf{X}^{(k+1)}$ and the inverse Jacobian matrix \mathbf{B}_{k+1}^{n+1} [34]. Set $k=k+1$ and go to Step 2.

5 Application to two-phase Navier-Stokes equations

The previous method can be extended to other types of fluid flow (including the unsteady two-phase Stokes flow, steady two-phase Navier-Stokes flow and unsteady two-phase Navier-Stokes flow) involving interfaces based on the generalized Stokes solver. Without loss of generality, only the unsteady two-phase Navier-Stokes flow is considered in this section. The two-phase Stokes equations (2.1a)-(2.2) describe fluid only for very low Reynolds number. In modeling flows with higher Reynolds numbers, Eqs. (2.1a)-(2.2) are replaced by the following two-phase Navier-Stokes equations

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \nabla \cdot \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) + \mathbf{F}(\mathbf{x}, t) + \mathbf{G}(\mathbf{x}, t), \quad \mathbf{x} \in \Omega, \quad (5.1a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in \Omega. \quad (5.1b)$$

The initial condition and the Dirichlet boundary conditions for the velocity fields are given by $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0$ and $\mathbf{u}_{\partial\Omega} = \mathbf{u}_b$, respectively. With a semi-implicit temporal discretization by using the Crank-Nicolson scheme for the viscous terms and the Adams-Bashforth scheme for the convective terms, then the time integration from t^n to t^{n+1} can be written as:

$$\frac{\tilde{\mathbf{u}}^{n+1} - \tilde{\mathbf{u}}^n}{\mu \Delta t} + \frac{1}{\mu} (\mathbf{u} \cdot \nabla \tilde{\mathbf{u}})^{n+\frac{1}{2}} + \nabla p^{n+\frac{1}{2}} = \frac{1}{2} (\Delta \tilde{\mathbf{u}}^{n+1} + \Delta \tilde{\mathbf{u}}^n) + \mathbf{G}^{n+\frac{1}{2}} - \frac{C\{\tilde{\mathbf{u}}_t\}}{\mu}, \quad \mathbf{x} \in \Omega, \quad (5.2a)$$

$$\nabla \cdot \tilde{\mathbf{u}}^{n+1} = 0, \quad \mathbf{x} \in \Omega, \quad (5.2b)$$

where $(\mathbf{u} \cdot \nabla \tilde{\mathbf{u}})^{n+1/2}$ is approximated by,

$$(\mathbf{u} \cdot \nabla \tilde{\mathbf{u}})^{n+\frac{1}{2}} = \frac{3}{2} (\mathbf{u} \cdot \nabla \tilde{\mathbf{u}})^n - \frac{1}{2} (\mathbf{u} \cdot \nabla \tilde{\mathbf{u}})^{n-1}. \quad (5.3)$$

Next put the known terms in Eq. (5.2a) to the right side and denote in the form of

$$\mathcal{G} = \frac{1}{\mu \Delta t} \tilde{\mathbf{u}}^n + \frac{1}{2} \Delta \tilde{\mathbf{u}}^n - \frac{1}{\mu} (\mathbf{u} \cdot \nabla \tilde{\mathbf{u}})^{n+\frac{1}{2}} + \mathbf{G}^{n+\frac{1}{2}} - \frac{C\{\tilde{\mathbf{u}}_t\}}{\mu},$$

where $C\{\tilde{\mathbf{u}}_t\}$ is the correction term for the discretization of $\tilde{\mathbf{u}}_t$ and is only nonzero at a particular grid point which the interface crosses over the time interval $[t^n, t^{n+1}]$.

Given the transformed velocity $\tilde{\mathbf{u}}^n$, the transformed velocity $\tilde{\mathbf{u}}^{n+1}$ and pressure $p^{n+1/2}$ at the next time step can be computed by solving Eqs. (5.2a)-(5.2b), which is equivalent to find the solution $(\tilde{\mathbf{u}}^{n+1}, p^{n+1/2})$ from the following generalized Stokes interface problem with slightly modified jump conditions similar to those in (3.3a)-(3.4d):

$$\chi \tilde{\mathbf{u}} - \nu \Delta \tilde{\mathbf{u}} + \nabla p = \mathcal{G}, \quad \mathbf{x} \in \Omega, \quad (5.4a)$$

$$\nabla \cdot \tilde{\mathbf{u}} = 0, \quad \mathbf{x} \in \Omega, \quad (5.4b)$$

where $\chi = 1/(\mu\Delta t)$ and $\nu = 1/2$. Similar to the case of Steady two-phase Stokes equations (when $\chi = 0$ and $\nu = 1$), the discretization of Eqs. (5.4a)-(5.4b) by second order MAC finite difference scheme leads to the following discrete saddle point system:

$$\begin{pmatrix} \chi\mathbf{I} - \nu\Delta_h & G^{\text{MAC}} \\ D^{\text{MAC}} & 0 \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{u}} \\ p \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{B}}_1 \\ \tilde{B}_2 \end{pmatrix}, \tag{5.5}$$

where $\tilde{\mathbf{B}}_1$ and \tilde{B}_2 are the right hand terms which involve the appropriate correction terms similar to those in Section 4.1. The above system Eq. (5.5) can be efficiently solved by the preconditioned conjugate gradient Uzawa-type method with the preconditioners proposed by Cahouet and Chabard [3]. Based on the generalized Stokes solver for system of Eq. (5.5), the similar IIM algorithm as in Section 4.4 can be implemented for two-phase Navier-Stokes equation with interfaces.

6 Numerical experiments

In this section, several numerical experiments are carried out to demonstrate the capabilities of the proposed method. All the simulations are done on a Laptop with 2.00GHz processor.

Example 6.1 (An example with an exact solution under Stokes flow). In the first example taken from [22], the numerical test is performed to check the accuracy of the algorithm. The provided exact velocity and pressure are given by

$$u = \begin{cases} \frac{y}{4}, & x^2 + y^2 < 1, \\ \frac{y}{4}(x^2 + y^2), & x^2 + y^2 \geq 1, \end{cases} \tag{6.1a}$$

$$v = \begin{cases} -\frac{x}{4}(1 - x^2), & x^2 + y^2 < 1, \\ -\frac{xy^2}{4}, & x^2 + y^2 \geq 1, \end{cases} \tag{6.1b}$$

$$\tilde{p} = \begin{cases} \left(-\frac{3}{4}x^3 + \frac{3}{8}x\right)y, & x^2 + y^2 < 1, \\ 0, & x^2 + y^2 \geq 1, \end{cases} \tag{6.1c}$$

$$p = \tilde{p} - \text{mean}(\tilde{p}), \tag{6.1d}$$

where $\text{mean}(\tilde{p})$ is the average of \tilde{p} . The external force term $\mathbf{g} = (g_1, g_2)^T$ is derived directly from the exact solution as follows

$$g_1 = \begin{cases} \left(-\frac{9}{4}x^2 + \frac{3}{8}\right)y, & x^2 + y^2 < 1, \\ -2\mu^+y, & x^2 + y^2 \geq 1, \end{cases} \tag{6.2a}$$

$$g_2 = \begin{cases} -\frac{3}{4}x^3 + \frac{3}{8}x - \frac{3\mu^-}{2}x, & x^2 + y^2 < 1, \\ \frac{\mu^+}{2}x, & x^2 + y^2 \geq 1. \end{cases} \tag{6.2b}$$

And the singular force terms in the normal direction and tangential direction are

$$\hat{f}_1 = \left(\frac{3}{4} \cos^3 \theta - \frac{3}{8} \cos \theta \right) \sin \theta - \frac{3}{2} [\mu] \cos^3 \theta \sin \theta, \tag{6.3a}$$

$$\hat{f}_2 = \frac{1}{2} \mu^+ + \frac{3}{4} [\mu] \cos^2 \theta (1 - 2 \cos^2 \theta) \tag{6.3b}$$

calculated from (3.2a) and (3.2d), respectively, where θ is the angle between the x -axis and the normal direction at the point of the interface.

The simulation is performed with a 64×64 grid on a computational domain of $[-2, 2] \times [-2, 2]$. The interface is presented by 64 control points. Fig. 4 shows the plots of the x -component of the transformed velocity field, the x -component of the velocity field and the pressure profile with $\mu^+ = 0.1$ and $\mu^- = 1$. It is observed that the discontinuities in the transformed velocity and the pressure are well captured.

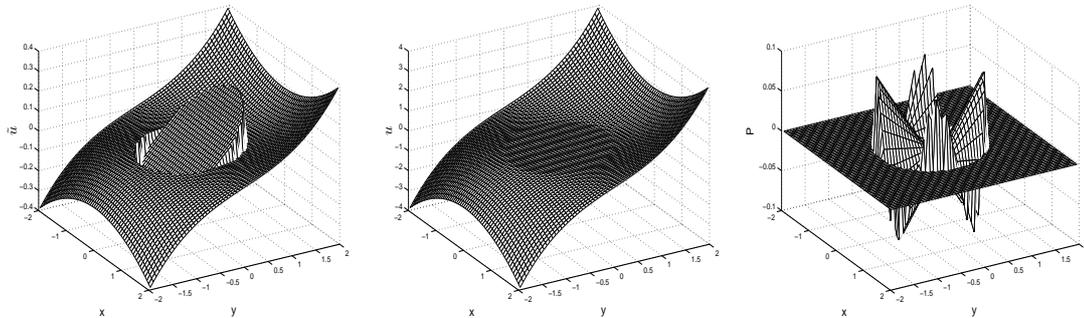


Figure 4: The x -component of the transformed velocity field $\tilde{\mathbf{u}}$ (left), the x -component of the velocity field \mathbf{u} (middle) and the pressure profile (right) with $\mu^+ = 0.1$ and $\mu^- = 1$.

The result of the convergence rate analysis is shown in Table 1 and Table 2, which indicates that the velocity is second order accurate, and the pressure is nearly second order accurate. The sixth column and the seventh column show the number of average CG iterations in the Stokes solver and the number of GMRES iterations in the augmented system, which indicates that a limited number of iterations are needed and the number of iterations is almost independent of the mesh size. The CPU time in seconds is listed in the last column, which shows that the present method is efficient.

Table 1: Grid refinement analysis for Example 6.1 with $\mu^+ = 1$ and $\mu^- = 0.1$.

N	$\ E_{\mathbf{u}}\ _{\infty}$	Order	$\ E_p\ _{\infty}$	Order	N_{CG}	N_{GMRES}	CPU(s)
32	1.5714E-02	–	1.8547E-02	–	10	8	0.39
64	3.8476E-03	2.03	5.1587E-03	1.85	10	8	1.15
128	9.7853E-04	1.98	1.3662E-03	1.92	11	7	3.95
256	2.4679E-04	1.99	3.6514E-04	1.90	12	7	20.79

Table 2: Grid refinement analysis for Example 6.1 with $\mu^+ = 0.1$ and $\mu^- = 1$.

N	$\ E_{\mathbf{u}}\ _{\infty}$	Order	$\ E_p\ _{\infty}$	Order	N_{CG}	N_{GMRES}	CPU(s)
32	5.8091E-03	–	2.5077E-02	–	9	9	0.36
64	1.5286E-03	1.93	6.6728E-03	1.91	10	9	1.45
128	3.5029E-04	2.13	1.8554E-03	1.85	11	9	5.17
256	8.9237E-05	1.97	5.0759E-04	1.87	12	9	22.52

Example 6.2 (Rotational flow under Stokes flow). In the second example, a fixed interface problem with no known solution under Stokes flow is considered. The computational domain is $[-1,1] \times [-1,1]$. The involved interface is a circle with radius $r = 1/2$, which is placed at the center of the domain. The no-slip boundary condition for the velocity is taken. The force is only taken along the tangential direction and the normal force is set as $\hat{f}_1 = 0$. In the cases below, the tangential force is taken as $\hat{f}_2 = 0.1$ unless it is stated otherwise.

In the computations, a 64×64 grid is used and 48 control points are set to represent the interface. For the first case, the viscosities outside and inside the interface are taken as $\mu^+ = 1$ and $\mu^- = 0.1$, respectively, where the viscosity outside the interface is larger than that inside the interface. The solution is presented in Fig. 5, and Fig. 5 (left) and Fig. 5 (middle) show the x -component of the transformed velocity $\tilde{\mathbf{u}}$ and the x -component of the velocity \mathbf{u} with the viscosity ratio $\lambda = 10$, respectively. From Fig. 5 (middle) it can be observed that the velocity u is continuous but not smooth, as expected. Due to discontinuous viscosity across the interface, the transformed velocity $\tilde{\mathbf{u}}$ is discontinuous, however, the discontinuity is well captured sharply by the proposed algorithm, which can be observed in Fig. 5 (left). The velocity field is presented in Fig. 5 (right), which corresponds to a rigid body motion inside the interface.

Next, the second case is considered, where the viscosity outside the interface is less than that inside the interface. Compared with the previous first one, the viscosities outside and inside the interface are just exchanged. That is, $\mu^+ = 0.1$ and $\mu^- = 1$ with $\lambda = 0.1$ are taken for the second case. The corresponding plots for the velocity are presented in Fig. 6. From Fig. 5 and Fig. 6, it can be seen that the respective velocity fields indicate some differences for the solution. A grid refinement analysis is carried out, using a referenced grid of 512×512 , to determine the order of the convergence of the algorithm. The results in Table 3 indicate that the velocity is second order accurate and pressure is nearly second order accurate. The number of CG iterations and the number GMRES are independent of the mesh size and shown in the sixth column and seventh column in the table. The last column shows the CPU time required. It is noted that the required CPU time using the current IIM-Stokes solver is 0.76s and 0.71s on a 64×64 grid for two cases of different viscosity ratios, respectively, compared to the computational cost of 218.85s and 226.51s by using the method in [35] directly to get the steady state solution. From the comparison, it is clearly seen that the computational cost is fairly low using the present method to solve steady two-phase Stokes flows.

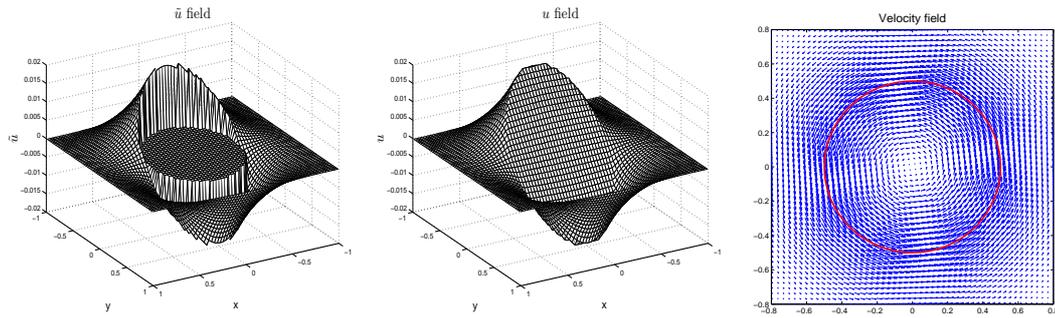


Figure 5: For Example 6.2 on rotational flow, first case. The x -component of the transformed velocity field $\tilde{\mathbf{u}}$ (left), the x -component of the velocity field \mathbf{u} (middle) and the velocity field \mathbf{u} (right), with $\hat{f}_1=0, \hat{f}_2=0.1, \mu^+=1, \mu^-=0.1$ and $\lambda=10$.

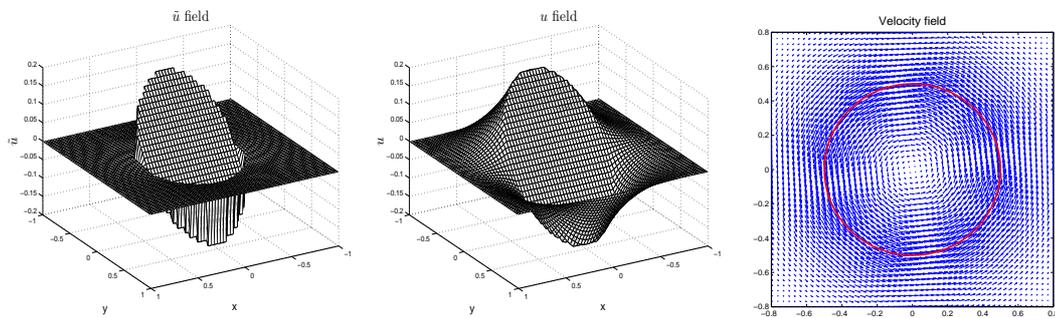


Figure 6: For Example 6.2 on rotational flow, second case. The x -component of the transformed velocity field $\tilde{\mathbf{u}}$ (left), the x -component of the velocity field \mathbf{u} (middle) and the velocity field \mathbf{u} (right), with $\hat{f}_1=0, \hat{f}_2=0.1, \mu^+=0.1, \mu^-=1$ and $\lambda=0.1$.

Table 3: Grid refinement analysis for Example 6.2.

N	$\ E_{\mathbf{u}}\ _{\infty}$	Order	$\ E_p\ _{\infty}$	Order	N_{CG}	N_{GMRES}	CPU(s)
(a) $\mu^+=1, \mu^-=0.1$							
64	1.5476E-05	–	7.1017E-04	–	8	5	0.76
128	3.8411E-06	2.01	2.0536E-04	1.79	8	5	2.74
256	9.0184E-07	2.09	5.5408E-05	1.89	8	5	12.15
(b) $\mu^+=0.1, \mu^-=1$							
64	1.9673E-05	–	8.4538E-04	–	8	6	0.71
128	4.4178E-06	2.15	2.2495E-04	1.91	8	6	3.39
256	1.0955E-06	2.01	6.1968E-05	1.86	8	6	14.18

Example 6.3 (Ellipse-shaped membrane under Stokes flow). In the second example, a moving interface problem which involves an ellipse-shaped membrane [17, 18, 37] is considered under Stokes flow. The initial elliptical membrane (the solid line in Fig. 7, labeled "Initial") has the semi-major axis $a=0.75$ and semi-minor axis $b=0.5$. The computational domain is $[-1, 1] \times [-1, 1]$ and the ellipse is located at the center of the domain. The un-

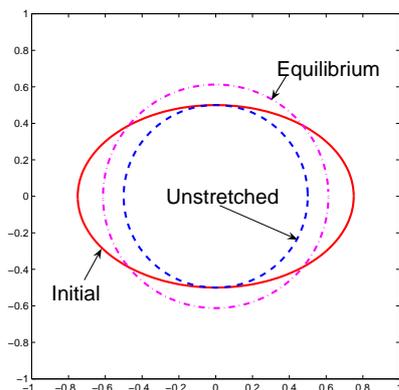


Figure 7: The interface configurations at different states in a square domain.

stretched state of membrane (the dashed line in Fig. 7, labeled "Unstretched") is a circle with radius $r_0 = 0.5$. The tension coefficient T_0 is set as 1 in this example.

Due to the restoring force, the ellipse will relax to a circle (the dashdotted line in Fig. 7, labeled "Equilibrium") with radius $r_e = \sqrt{ab} \approx 0.61237$, which is larger than the unstretched interface but has the same area as the initial ellipse because of the incompressibility of the enclosed fluid. In the computations, the homogeneous Dirichlet boundary condition is applied, i.e., $\mathbf{u}|_{\partial\Omega} = \mathbf{0}$ unless it is stated otherwise. In this example, a 64×64 grid is employed, and 64 control points are used to represent the interface. In the simulations, the velocity and pressure at time $t = 0$ based on the initial elliptical interface are first computed before the interface has moved.

For the case of $\lambda = 0.1$ with $\mu^+ = 0.1$ and $\mu^- = 1$, where the viscosity outside the interface is less than that inside the interface. The x -component of the transformed velocity field $\tilde{\mathbf{u}}$ and the x -component of velocity field \mathbf{u} at $t = 0$ are presented in Fig. 8 (left) and

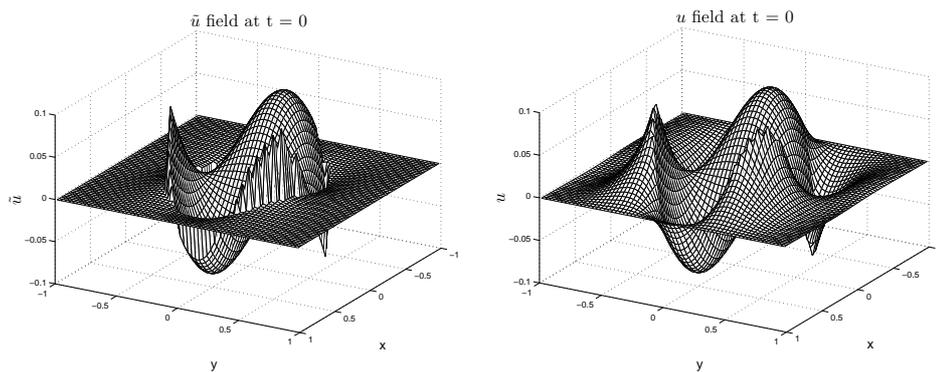


Figure 8: The x -component of the transformed velocity field $\tilde{\mathbf{u}}$ (left) and the x -component of velocity field \mathbf{u} (right) at $t = 0$ with $T_0 = 1$, $\mu^+ = 0.1$, $\mu^- = 1$ and $\lambda = 0.1$.

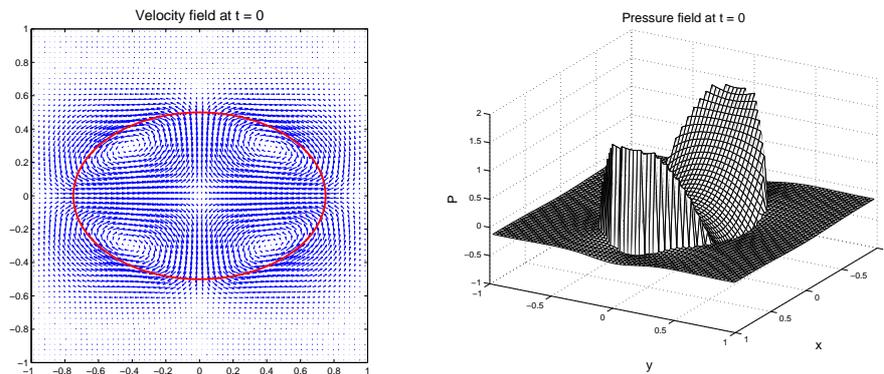


Figure 9: The velocity field \mathbf{u} (left) and the pressure distribution at $t=0$ with $T_0=1$, $\mu^+=0.1$, $\mu^-=1$ and $\lambda=0.1$.

Fig. 8 (right), respectively. The corresponding velocity field \mathbf{u} and pressure profile are presented in Fig. 9 (left) and Fig. 9 (right), respectively. As expected, it can be observed from these figures that the transformed velocity \tilde{u} and pressure p are discontinuous across the interface while the velocity u is continuous but not smooth. Fig. 10 shows this more clearly with the plots of cross section of \tilde{u} -component and u -component along the line $y = -0.39$ and the plot of cross section of p along the line $y = -0.015$ at $t = 0$. The discontinuities in the transformed velocity, the derivative of the velocity and pressure are very sharply captured across the interface as shown in Fig. 10 (left), Fig. 10 (middle) and Fig. 10 (right), respectively.

For the case of $\lambda = 10$ with $\mu^+ = 1$ and $\mu^- = 0.1$, where the viscosity outside the interface is larger than that inside the interface, the x -component of the transformed velocity field \tilde{u} , the x -component of velocity field \mathbf{u} and pressure profile at $t = 0$ are plotted in Fig. 11 (left), Fig. 11 (middle) and Fig. 11 (right), respectively. Again, the sharp jump in the transformed velocity, the derivative of the velocity and the pressure are well re-

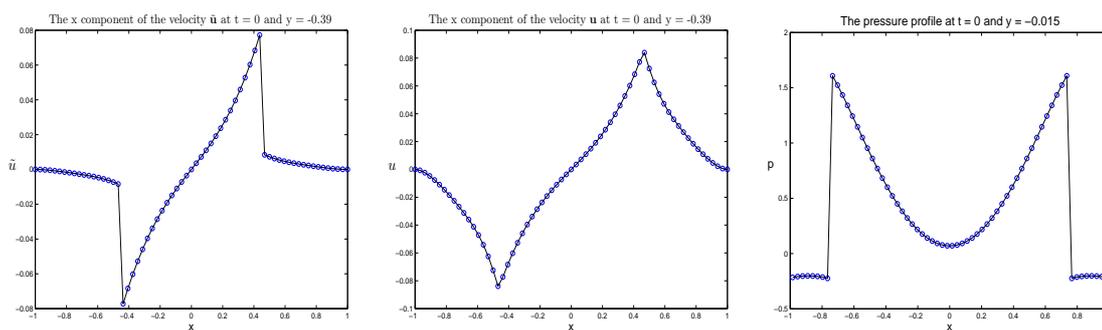


Figure 10: For Example 6.3. A slice of the \tilde{u} -component transformed velocity (left), u -component velocity (middle) along $y = -0.39$ and the pressure profile (right) along $y = -0.015$ at $t=0$ with $T_0=1$, $\mu^+=0.1$, $\mu^-=1$ and $\lambda=0.1$.

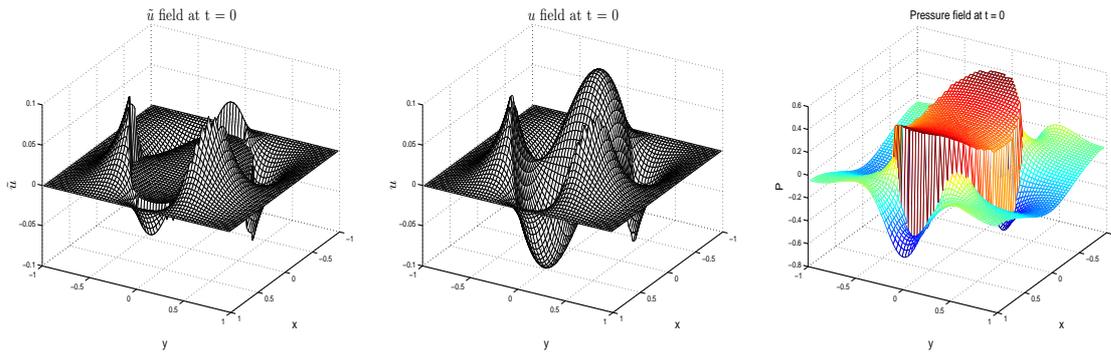


Figure 11: The x -component of the transformed velocity field $\tilde{\mathbf{u}}$ (left) and the x -component of velocity field \mathbf{u} (middle) and pressure profile (right) at $t=0$ with $T_0=1$, $\mu^+=1$, $\mu^-=0.1$ and $\lambda=10$, where the Dirichlet boundary condition is used.

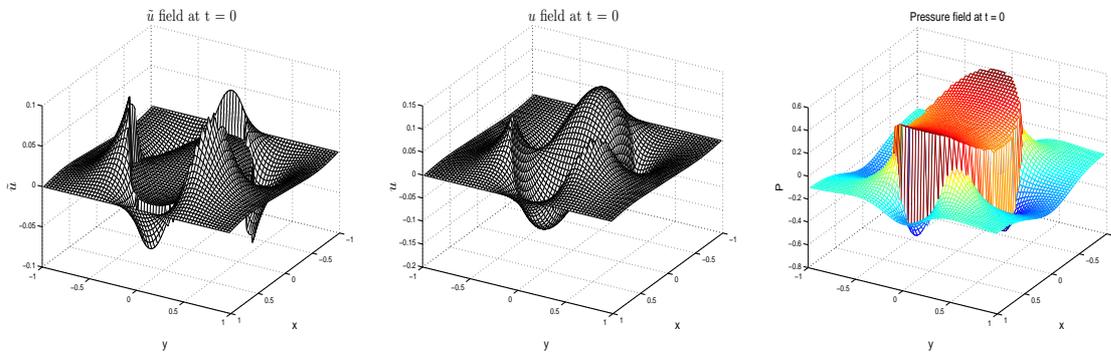


Figure 12: The x -component of the transformed velocity field $\tilde{\mathbf{u}}$ (left), the x -component of velocity field \mathbf{u} (middle) and pressure profile (right) at $t=0$ with $T_0=1$, $\mu^+=1$, $\mu^-=0.1$ and $\lambda=10$, where the periodic boundary condition is used.

solved. The present method can also directly be used to solve the flow problem with the periodic boundary condition. The corresponding solutions are presented in Fig. 12 for the same case, which can be compared to Fig. 11 where the Dirichlet boundary condition is employed. Some differences can be found due to the use of the different boundary condition from these two figures. It demonstrates that the present method is flexible for different boundary conditions.

The convergence analysis for the flow field is performed. Since the analytic solution is not available, the error in velocity and pressure are measured via using a reference solution which is obtained on a fine 512×512 grid. In Table 4, the convergence rate analysis at $t=0$ is shown, and the expected second order accuracy for the velocity and near second order accuracy for the pressure are achieved. Again it can be seen that the number of CG iterations and the number of GMRES iterations remains almost the same with grid refinement. The required CPU time is shown in the last column of the table, which shows the efficiency of the present IIM-Stokes solver. The CPU time performed on a 64×64 grid

Table 4: Grid refinement analysis for Example 6.3.

N	$\ E_{\mathbf{u}}\ _{\infty}$	Order	$\ E_p\ _{\infty}$	Order	N_{CG}	N_{GMRES}	CPU(s)
(a) $\mu^+ = 1, \mu^- = 0.1$							
64	1.3070E-03	–	5.4706E-03	–	11	8	1.21
128	3.2273E-04	2.02	1.4966E-03	1.87	12	7	4.54
256	7.3159E-05	2.14	3.9275E-04	1.93	13	7	21.70
(b) $\mu^+ = 0.1, \mu^- = 1$							
64	2.2359E-03	–	5.6951E-03	–	11	9	1.48
128	5.3925E-04	2.05	1.5154E-03	1.91	12	8	5.33
256	1.0277E-04	2.39	4.0887E-04	1.89	13	8	23.43

up to $t = 0.01$ by the current method is 8.32s and 9.34s for corresponding cases, respectively, compared to the computational cost of 138.69s and 139.17s by the method in [35] using Navier-Stokes equation and letting the low Reynolds number become very low so as to approach the Stokes flow regime. The comparison shows clearly the advantage of time savings for solving the Stokes solution in the present work vs-a-vs the Navier-Stokes equations in [35].

For the purpose of the comparison of viscosity effects on interface motion over longer times, the evolutions of the semi-major and semi-minor axes versus time for four cases are plotted in Fig. 13. It can be observed that, under the elastic force and viscosity, the elliptical interfaces relax gradually to the circular shape without oscillations until the equilibrium is reached for all the cases. At $t = 150$, the maximum amplitude between semi-major axis and r_e is $8.1185e-5$, and the maximum amplitude between semi-minor axis and r_e is $3.9432e-5$ among them. The final numerical equilibrium is in good agreement with the true equilibrium. It is also noted that, time taken to reach the equilibrium state is different due to different viscosities. For the case of same viscosities $\mu^+ = 0.1$ and $\mu^- = 0.1$, the viscosities taken are the smallest outside and inside the interface, and the interface takes the shortest time to converge to the equilibrium state as seen from the dotted line in Fig. 13. However, for the case of same viscosities with $\mu^+ = 1$ and $\mu^- = 1$, the viscosities taken are the largest outside and inside the interface, and the interface takes the longest time to converge to the equilibrium state as seen from the solid line in Fig. 13. For the other two cases of different viscosities outside and inside the interface, the times taken are between above-mentioned two cases of same viscosities. With larger viscosity outside the interface, i.e., $\mu^+ = 1$ and $\mu^- = 0.1$, the interface takes a relatively larger time to relax to the equilibrium state than the other one as seen from the dashed line and dash-dot line in Fig. 13.

In Fig. 14, the plots of the absolute value of the numerical divergence of velocity field, $|\text{div}_h \mathbf{u}|$, versus time up to $t = 150$ with different viscosities are given. It shows that, the divergence of the velocity field on the discrete level is small and the maximum discrete divergence of velocity field among the four cases reaches an accuracy of $4.2106e-5$. Moreover, after a later time the discrete divergence of the velocity field is getting very close to zero for all the cases.

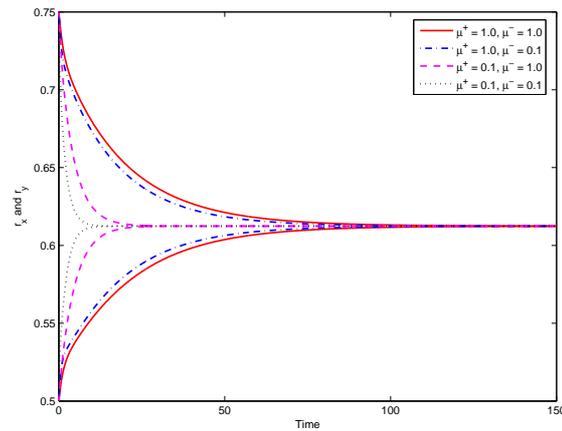


Figure 13: For Example 6.3. The evolution of r_x and r_y with $T_0=1$ and different viscosities. The interface relaxes gradually to the equilibrium state without oscillations.

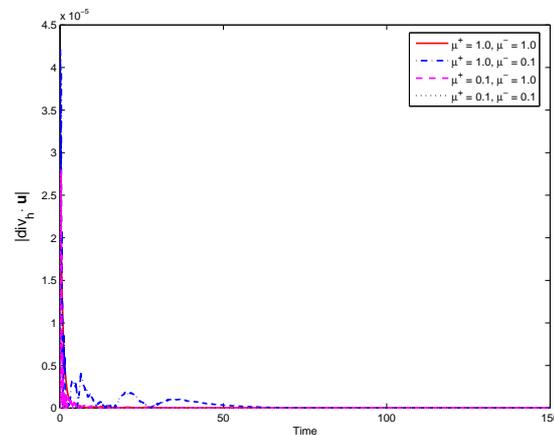


Figure 14: For Example 6.3. The absolute value of the numerical divergence of velocity field, $|\text{div}_h \mathbf{u}|$, versus time with $T_0=1$ and different viscosities.

The area conservation in the simulation is also found to be kept well. In Fig. 15, the plots of the area conservation error versus time up to $t = 150$ for the four cases are presented. In the figure, the maximum area conservation error is $1.4166e-004$ and it indicates only a small area loss of 0.01202% in percentage.

Finally, the iteration in GMRES is briefly discussed for this problem. In Fig. 16 (left), the number of total GMRES iterations versus time for the case with $\mu^+ = 1.0$, $\mu^- = 0.1$ and $\lambda = 10$ is presented. The number of total GMRES iterations versus time with $\mu^+ = 0.1$, $\mu^- = 1.0$, and $\lambda = 0.1$ is presented in Fig. 16 (right). It can be seen that the number of total GMRES iterations is not very large for the present cases. In these plots, the number of total GMRES iterations means the sum of the number of GMRES iterations at each BFGS iteration within one time step.

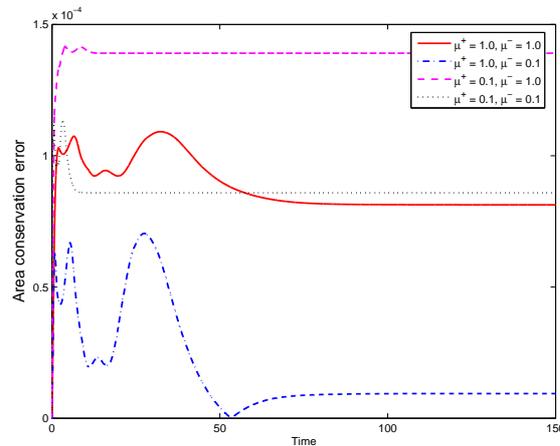


Figure 15: For Example 6.3. Plot of the absolute error in area versus time with $T_0=1$ and different viscosities.

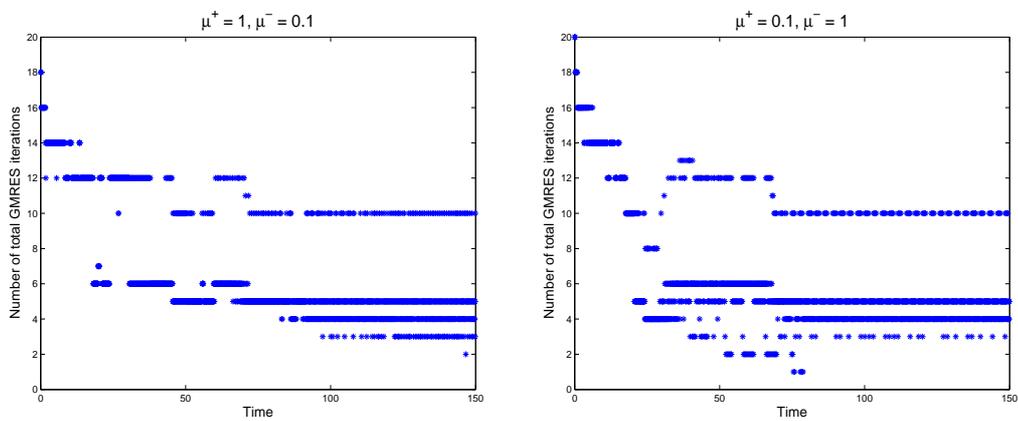


Figure 16: For Example 6.3. Number of total GMRES iterations with $T_0=1$ and $\Delta t=h/8$ (left) $\mu^+=1$ and $\mu^-=0.1$ (right) $\mu^+=0.1$ and $\mu^-=1$.

Example 6.4 (Flower-shaped membrane under Stokes flow). This example is to show that the present method can handle flows with a more complicated initial interface like the configuration found in the literature [18] for a stiff problem. Similar to the above example, the initial interface is now stretched to a flower shape describing in polar coordinate as is given by $r(\theta) = 0.8+0.3\sin(9\theta)$ in polar coordinates. The unstretched interface is a circle with the radius $r_0 = 0.3$. Both are shown in Fig. 17 (solid line and dashdotted line, respectively). In the computations, a 128×128 grid on a computational domain of $[-1.5,1.5] \times [-1.5,1.5]$ is employed, and 160 control points are used to represent the interface. The tension coefficient T_0 is set as 0.5.

In Fig. 17, the configurations of the interface at $t = 0$, $t = 0.2$, $t = 0.4$ and $t = 1$ are plotted with $\mu^+=1.0$ and $\mu^-=0.1$. At $t=1$, the interface is almost in the equilibrium state

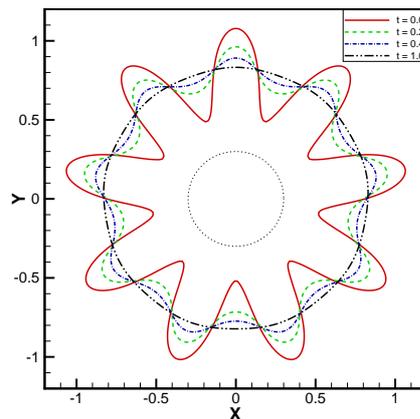


Figure 17: For Example 6.3. The configurations of the interface at $t=0$ (solid line), $t=0.2$ (dashed line), $t=0.4$ (dashdotted line) and $t=1.0$ (dashed line), with $\mu^+ = 1.0, \mu^- = 0.1$ and $T_0 = 0.5$. The innermost dotted circle shows the unstretched interface.

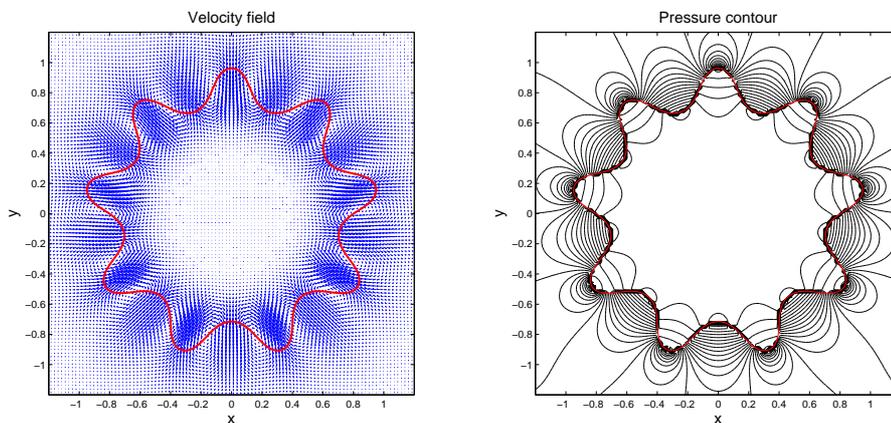


Figure 18: For Example 6.3. The velocity field (left) and pressure contours (right) at $t=0.2$ with $\mu^+ = 1.0, \mu^- = 0.1$ and $T_0 = 0.5$.

depicted as a circle in Fig. 17 (dashdotted line). The corresponding velocity field and pressure contour at $t=0.2$ are plotted in Fig. 18. From Fig. 18 (right) it is clear that the present method can capture the highly localized discontinuous profile for the pressure.

Example 6.5 (Elastic membrane under Navier-Stokes flow). This example is to show the current IIM combined with the generalized Stokes solver is flexible in terms of types of fluid flows, which is applied to solve for the incompressible two-phase Navier-Stokes equation with moving interface. Similar to the above Example 6.3, the initial interface is stretched to an ellipse-shaped membrane. The Navier-Stokes flow in the square domain of $[-1.5, 1.5] \times [-1.5, 1.5]$ is considered and the ellipse is located at the center of the domain. The semi-major and semi-minor axes of ellipse are $a = 0.75, b = 0.5$, respectively. The radius of unstretched circle-shaped membrane is $r_0 = 0.5$. The tension coefficient T_0 is

set as 10 in this example. The flow is initially at rest with $u_0 = v_0 = 0$. Under the effect of the elastic force and viscosity, the membrane finally converge to a circular shape. The simulations are conducted on a 64×64 grid and 64 control points are used to represent the interface.

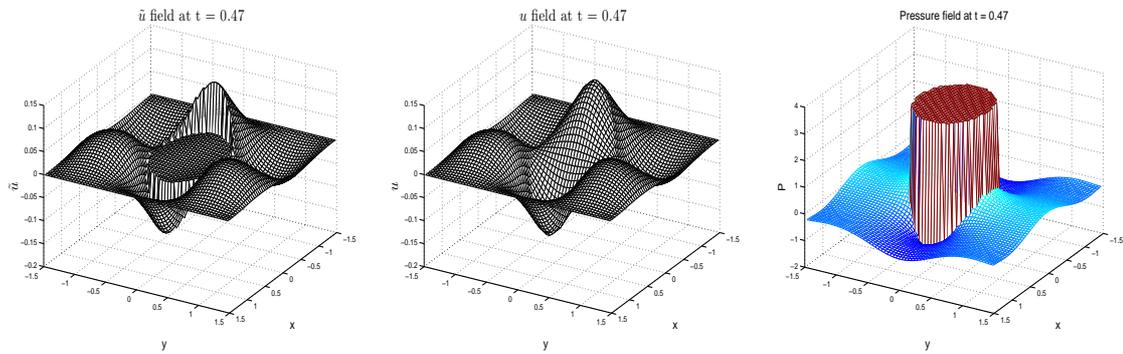


Figure 19: The x -component of the transformed velocity field $\tilde{\mathbf{u}}$ (left), the x -component of velocity field \mathbf{u} (middle) and pressure profile (right) at $t=0.47$ with $T_0=10$, $\mu^+=1$, $\mu^-=0.1$ and $\lambda=10$.

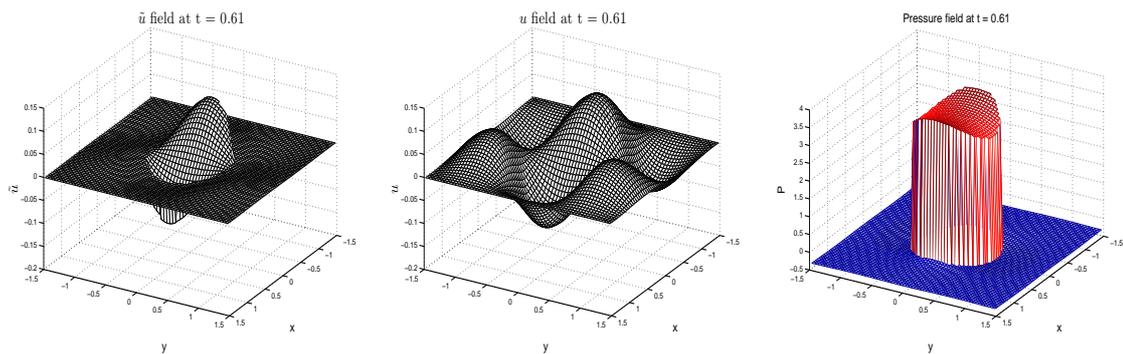


Figure 20: The x -component of the transformed velocity field $\tilde{\mathbf{u}}$ (left), the x -component of velocity field \mathbf{u} (middle) and pressure profile (right) at $t=0.61$ with $T_0=10$, $\mu^+=0.1$, $\mu^-=1$ and $\lambda=0.1$.

The x -component of the transformed velocity field, the x -component of velocity field and the pressure profile are presented in Fig. 19 and Fig. 20 for the case with $\lambda = 10$ ($\mu^+ = 1$ and $\mu^- = 0.1$) and the case with $\lambda = 0.1$ ($\mu^+ = 0.1$ and $\mu^- = 1.0$), respectively. It can be observed that the sharp jump in the transformed velocity field and pressure is well captured by the present algorithm.

Fig. 21 shows the evolutions of the semi-major and semi-minor axes for $\lambda = 10$, $\lambda = 1$ and $\lambda = 0.1$ with $\mu^+ = 0.1$ fixed, and reference to the same viscosities of $\mu^+ = \mu^- = 1$ as shown in the dotted line. It is observed that, the membrane with these three cases of different viscosities oscillates and finally converge to a circular shape and the effect from different viscosities results in a different oscillation number and a decay of the different

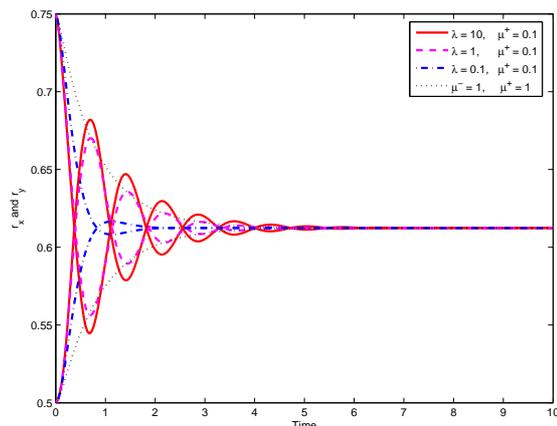


Figure 21: The evolution of r_x and r_y with fixed $\mu^+ = 0.1$, for $\lambda = 10$, $\lambda = 1$ and $\lambda = 0.1$.

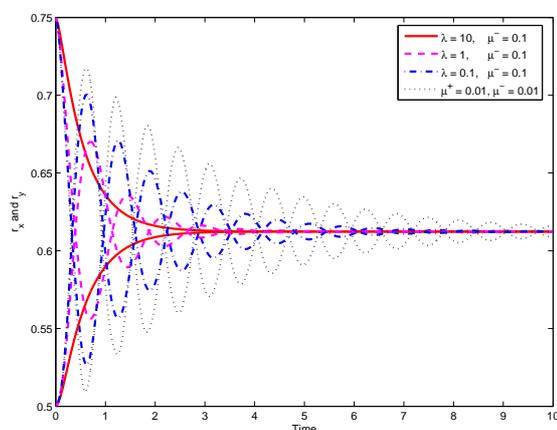


Figure 22: The evolution of r_x and r_y with fixed $\mu^- = 0.1$, for $\lambda = 10$, $\lambda = 1$ and $\lambda = 0.1$.

oscillation amplitude before the equilibrium is reached. For the referred case with the same viscosities of $\mu^+ = \mu^- = 1$, the membrane relax gradually to the equilibrium state without oscillations. With larger viscosity ratio the fluids (inside and outside the interface) move faster. In Fig. 22, the evolutions of the semi-major and semi-minor axes are presented for $\lambda = 10$, $\lambda = 1$ and $\lambda = 0.1$ with $\mu^- = 0.1$ fixed, the larger the viscosity ratio the fluids move slower. For the case of $\lambda = 10$ with $\mu^+ = 1$ and $\mu^- = 0.1$, the membrane relax gradually to the equilibrium state without oscillations, however, for the referred case with the same viscosities of $\mu^+ = \mu^- = 0.01$, the membrane takes the longest time to oscillate with most cycles before setting down to the equilibrium state. It is clearly demonstrated from Fig. 21 and Fig. 22 that with more viscosity the fluid moves slower. It is also interesting to compare Figs. 21 and 21 broadly with Fig. 13, the latter under Stokes flow dynamics. It is clear under Stokes flow, the time taken to reach equilibrium is far longer and there is the absence of oscillations.

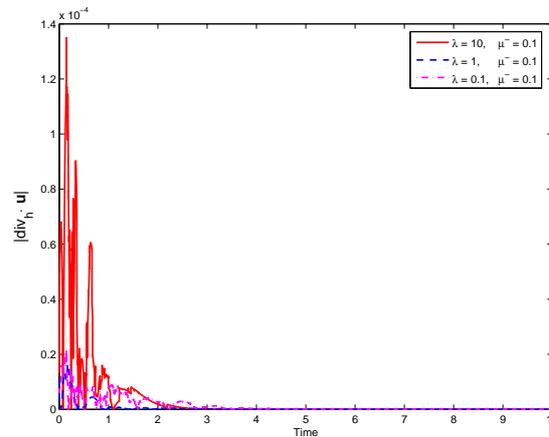


Figure 23: For Example 6.4. The absolute value of the numerical divergence of velocity field, $|\text{div}_h \mathbf{u}|$, versus time with $T_0=10$ and different viscosities.

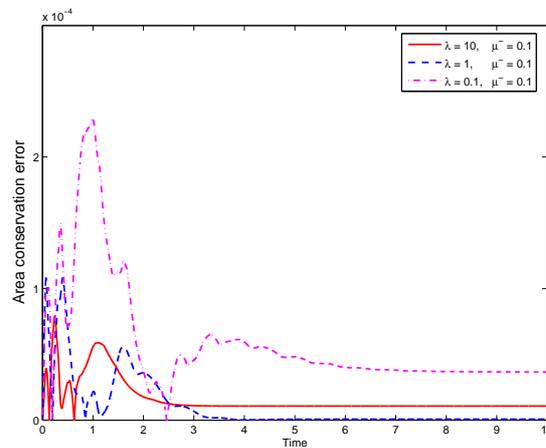


Figure 24: For Example 6.4. Plot of the absolute error in area versus time with $T_0=10$ and different viscosities.

Fig. 23 shows the plot of the absolute value of the numerical divergence of velocity field, $|\text{div}_h \mathbf{u}|$, versus time up to $t=150$ for fixed $\mu^- = 0.1$ with different viscosity ratios. In Fig. 24, the corresponding plots of the area conservation error versus time up to $t=150$ are presented. It indicates that the divergence-free condition on the discrete level and area conservation are both satisfied well with the present method.

7 Concluding remarks

In this paper, a novel implementation of MAC grid-based IIM combined with Stokes solver for solving incompressible two-phase flow with the interface is presented. The components of the transformed velocity are introduced as two augmented variables to

satisfy the continuous velocity condition across the interface. The augmented variables are solved by the GMRES method. The Stokes equations are discretized on a MAC grid via a second order finite difference method by incorporating the jump conditions and then solved by the conjugate gradient Uzawa-type method. The present method is flexible for different types of two-phase flow problems and different boundary conditions. The numerical results show that the proposed algorithm can achieve second order accuracy for the velocity and nearly second order accuracy for the pressure. The proposed method is efficient, and the advantages of time and computing resources savings for solving Stokes equation in the present work vis-a-vis the Navier-Stokes are shown. The present IIM combined with generalized two-phase Stokes solver with correction terms has also been applied to solve for incompressible two-phase Navier-Stokes flow with interfaces. Current method is limited to cases where the topology of the interface exists changes. Future works include the extension to 3D.

Acknowledgments

The authors would like to thank the referees for the valuable suggestions on the revision of the manuscript. The research of the first author was partially supported by Guangdong Provincial Government of China through the "Computational Science Innovative Research Team" program and the Sun Yat-sen University "Hundred Talents Program" (34000-3181201), and the National Natural Science Foundation of China (No. 11101446).

References

- [1] J. Adams, P. Swarztrauber and R. Sweet, FISHPACK: Efficient FORTRAN subprograms for the solution of separable elliptic partial differential equations, 1999, Available on the web at <http://www.scd.ucar.edu/css/software/fishpack/>.
- [2] R. P. Beyer, A computational model of the cochlea using the immersed boundary method, *J. Comput. Phys.*, 98 (1992), 145–162.
- [3] J. Cahouet and J.-P. Chabard, Some fast 3d finite element solvers for the generalized Stokes problem, *Int. J. Numer. Methods Fluids*, 8 (1988), 869–895.
- [4] D. Calhoun, A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions, *J. Comput. Phys.*, 176 (2002), 231–275.
- [5] B. Christoph, Domain imbedding methods for the Stokes equations, *Numer. Math.*, 57 (1990), 435–451.
- [6] R. Dillon, L. Fauci, A. Fogelson and D. Gaver, Modeling biofilm processes using the immersed boundary method, *J. Comput. Phys.*, 129 (1996), 57–73.
- [7] C. D. Eggleton and A. S. Popel, Large deformation of red blood cell ghosts in a simple shear flow, *Phys. Fluids*, 10 (1998), 1834–1845.
- [8] H. C. Elman, Multigrid and Krylov subspace methods for the discrete Stokes equations, *Int. J. Numer. Methods Fluids*, 227 (1996), 755–770.
- [9] H. C. Elman, Preconditioners for saddle point problems arising in computational fluid dynamics, *Appl. Numer. Math.*, 43 (2002), 75–89.

- [10] L. J. Fauci and C. S. Peskin, A computational model of aquatic animal locomotion, *J. Comput. Phys.*, (1988), 85–108.
- [11] A. L. Fogelson, A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting, *J. Comput. Phys.*, 1 (1984), 111–134.
- [12] A. L. Fogelson, Continuum models of platelet aggregation: formulation and mechanical properties, *SIAM J. Appl. Math.*, 52 (1992), 1089–1110.
- [13] F. H. Harlow and J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids*, 8 (1965), 2182–2189.
- [14] G. M. Kobelkov and M. A. Olshanskii, Effective preconditioning of Uzawa type schemes for a generalized Stokes problem, *Numer. Math.*, 86 (2000), 443–470.
- [15] A. T. Layton, An efficient numerical method for the two-fluid Stokes equations with a moving immersed boundary, *Comput. Methods Appl. Mech. Eng.*, 197 (2008), 2147–2155.
- [16] D. V. Le, B. C. Khoo and J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, *J. Comput. Phys.*, 220 (2006), 109–138.
- [17] L. Lee and R. J. LeVeque, An immersed interface method for incompressible Navier-Stokes equations, *SIAM J. Sci. Comput.*, 25 (2003), 832–856.
- [18] R. J. LeVeque and Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, *SIAM J. Sci. Comput.*, 18 (1997), 709–735.
- [19] R. J. LeVeque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.*, 31 (1994), 1019–1044.
- [20] Z. Li, An overview of the immersed interface method and its applications, *Taiwan J. Math.*, 7 (2003), 1–49.
- [21] Z. Li and K. Ito, The immersed interface method-numerical solutions of PDEs involving interfaces and irregular domains, *SIAM Frontiers Appl. Math.*, 33 (2006), ISBN: 0-89971-609-8.
- [22] Z. Li, K. Ito and M-C. Lai, An augmented approach for Stokes equations with a discontinuous viscosity and singular forces, *Comput. Fluids*, 36 (2007), 622–635.
- [23] Z. Li and M. C. Lai, The immersed interface method for the Navier-Stokes equations with singular forces, *J. Comput. Phys.*, 171 (2001), 822–842.
- [24] Z. Li and C. Wang, A fast finite difference method for solving Navier-Stokes equations on irregular domains, *Commun. Math. Sci.*, 1 (2003), 180–196.
- [25] M. N. Linnick and H. F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, *J. Comput. Phys.*, 204 (2005), 157–192.
- [26] C. W. Oosterlee and F. J. G. Lorenz, Multigrid methods for the Stokes system, *Comput. Sci. Eng.*, 8 (2006), 34–43.
- [27] C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.*, 25 (1977), 220–252.
- [28] C. S. Peskin, The immersed boundary method, *Acta Numerica*, 11 (2002), 479–517.
- [29] J. Peters, V. Reichelt and A. Reusken, Fast iterative solvers for discrete Stokes equations, *SIAM J. Sci. Comput.*, 27 (2005), 646–666.
- [30] D. Russell and Z. J. Wang, A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, *J. Comput. Phys.*, 191 (2003), 177–205.
- [31] V. Sarin and A. Sameh, An efficient iterative method for the generalized Stokes problem, *SIAM J. Sci. Comput.*, 19 (1998), 206–226.
- [32] D. Shin and J. C. Strikwerda, Fast solvers for finite difference approximations for the Stokes and Navier-Stokes equations, *J. Australian Math. Soc.*, 38 (1996), 274–290.
- [33] J. M. Stockie and S. I. Green, Simulating the motion of flexible pulp fibres using the im-

- mersed boundary method, *J. Comput. Phys.*, 147 (1998), 147–165.
- [34] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, 3rd ed., Springer-Verlag, 2002.
- [35] Z.-J. Tan, D. V. Le, Z. Li, K. M. Lim and B. C. Khoo, An immersed interface method for solving incompressible viscous flows with piecewise constant viscosity across a moving elastic membrane, *J. Comput. Phys.*, 227 (2008), 9955–9983.
- [36] E. Y. Tau, Numerical solution of the steady Stokes equations, *J. Comput. Phys.*, 99 (1992), 190–195.
- [37] C. Tu and C. S. Peskin, Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods, *SIAM J. Sci. Statist. Comput.*, 13 (1992), 1361–1376.
- [38] S. Xu and Z. J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, *J. Comput. Phys.*, 216 (2006), 454–493.
- [39] S. Xu and Z. J. Wang, A 3D immersed interface method for fluid-solid interaction, *Comput. Methods Appl. Mech. Eng.*, 197 (2008), 2068–2086.