Journal of Computational Physics 227 (2008) 9955–9983

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An immersed interface method for solving incompressible viscous flows with piecewise constant viscosity across a moving elastic membrane

Zhijun Tan[a], D.V. Le[a], Zhilin Li[b], K.M. Lim[a,c], B.C. Khoo[a,c,*]

[a] Singapore-MIT Alliance, 4 Engineering Drive 3, National University of Singapore, Singapore 117576, Singapore
[b] Center for Research in Scientific Computation and Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, USA
[c] Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore

ARTICLE INFO

Article history:
Received 27 January 2008
Received in revised form 18 August 2008
Accepted 19 August 2008
Available online 30 August 2008

Keywords:
Incompressible viscous flows
Piecewise constant viscosity
Singular force
Immersed interface method
Projection method
Front tracking method

ABSTRACT

This paper presents an implementation of the second-order accurate immersed interface method to simulate the motion of the flexible elastic membrane immersed in two viscous incompressible fluids with different viscosities, which further develops the work reported in Tan et al. [Z.-J. Tan, D.V. Le, K.M. Lim, B.C. Khoo, An Immersed Interface Method for the Incompressible Navier–Stokes Equations with Discontinuous Viscosity Across the Interface, submitted for publication] focussing mainly on the fixed interface problems. In this work, we introduce the velocity components at the membrane as two augmented unknown interface variables to decouple the originally coupled jump conditions for the velocity and pressure. Three forms of augmented equation are derived to determine the augmented variables to satisfy the continuous condition of the velocity. The velocity at the membrane, which determine the motion of the membrane, is then solved by the GMRES iterative method. The forces calculated from the configuration of the flexible elastic membrane and the augmented variables are interpolated using cubic splines and applied to the fluid through the jump conditions. The position of the flexible elastic membrane is updated implicitly using a quasi-Newton method (BFGS) within each time step. The Navier–Stokes equations are solved on a staggered Cartesian grid using a second order accurate projection method with the incorporation of spatial and temporal jump conditions. In addition, we also show that the inclusion of the temporal jump contributions has non-negligible effect on the simulation results when the grids are crossed by the membrane. Using the above method, we assess the effect of different viscosities on the flow solution and membrane motion.

1. Introduction

Many problems of fluid mechanics involve the interaction of a viscous incompressible fluid and a moving elastic membrane (called fluid-membrane interactions). One can consider the membrane as a part of the fluid which exerts forces to the surrounding viscous fluid, while also moving with the velocity of adjacent fluid particles. The mathematical formulation and numerical method for this type of problems was first introduced by Peskin in what we now commonly called the immersed boundary method to simulate the blood flow in the heart and through heart valves [25,30]; the method has been used for many other applications particularly in bio-fluid dynamics. Examples include the deformation of red blood cell in a shear

[*] Corresponding author. Address: Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore. Tel.: +65 65162889; fax: +65 67791459.

E-mail addresses: smatz@nus.edu.sg (Z. Tan), smaldv@nus.edu.sg (D.V. Le), zhilin@math.ncsu.edu (Z. Li), mpelimkm@nus.edu.sg (K.M. Lim), mpekbc@nus.edu.sg (B.C. Khoo).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.08.013

flow [5], swimming of organisms [6], platelet aggregation [7,8], cochlear dynamics [1], biofilm processes [4], wood pulp fiber dynamics [27], and so on. A summary of the development of the immersed boundary method can be found in [24].

In the immersed boundary method, the force densities are computed at the control points which are used to represent the boundaries, and then spread to the neighbouring Cartesian grid points by a discrete representation of the delta function. The Navier–Stokes equations with the forcing terms are then solved for the pressure and velocity at the Cartesian grid points. The resulting velocities are interpolated back to the control points using the same set of discrete delta functions. As such, the immersed boundary method has several attractive features: the method is simple to implement, it can handle complex geometries easily and it uses standard regular Cartesian grid Navier–Stokes solvers. However, since the immersed boundary method uses the discrete delta function approach, it smears out sharp interface to a thickness of order of the mesh width and it is only about first-order accurate for general problems.

Unlike the immersed boundary method with numerical smearing near the interface, the immersed interface method (IIM) 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. For fluid dynamics problems, in the case of a continuous viscosity, the singular force $\mathbf{f}$ along the immersed boundaries results in solution to the Navier–Stokes equations which may be non-smooth across the interface, i.e., there may be jumps in pressure and in the derivatives of both pressure and velocity at the interface. An essential ingredient of the IIM is the relation between the applied singular forces and the jumps in the solutions and their derivatives. The basic idea of the IIM is to discretize the Navier–Stokes equations on a uniform Cartesian grid and to account for the singular forces with the incorporation of the jumps in the solutions and their derivatives into the difference schemes. The IIM was originally proposed by LeVeque and Li [22] for solving elliptic equations, and later extended to Stokes flow with elastic boundaries or surface tension [21]. The immersed interface method was developed further for the Navier–Stokes equations in [19,20,15,33,23]. The IIM was also used in [3,16,26] for solving the two-dimensional streamfunction–vorticity equations on irregular domains. Very recently, Xu and Wang [34] 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 [11] or the book by Li and Ito [12] and the references therein.

It should be noted that the employment/applicability of the immersed interface method depends critically on whether the necessary jump conditions are available/implementable. With the same viscosity across the interface, the derivation of the jump conditions can be found in [15,19,21] for two-dimensional flows and in [10,32] for three-dimensional flows. In the case of a discontinuous viscosity, the jump conditions for the pressure and the velocity are coupled together, see (2.10)–(2.14), which makes it difficult to discretize the system accurately and renders the application of the immersed interface method challenging. Li et al. [14,17] have recently developed the augmented immersed interface method for incompressible 2D Stokes flows with discontinuous viscosity. In [14], the interface is represented by a cubic spline interpolation and tracked explicitly in a Lagrangian manner. In [17], the level set method is used to represent the interface. Very recently, an efficient numerical method for solving the two-fluid Stokes equations with a moving immersed boundary was presented by Layton [13]. The method uses integral equations to reduce the two-fluid Stokes problem to the single-fluid case, which can then be solved via the immersed interface method in the usual way. It should be noted that the above works are limited to the Stokes equation with discontinuous viscosity across the interface using immersed interface method. In [29], Tan et al. have developed a second order accurate immersed interface method for the Navier–Stokes equations with discontinuous viscosity across the interface, where the work is mainly concentrated on the fixed interface problems. They have provided the necessary jump conditions, in particular a new jump condition for the normal derivative of pressure in [29]. The main objective of this paper is to extend further the immersed interface method developed in [19,29] to the incompressible Navier–Stokes equations with piecewise constant viscosity across a moving elastic interface immersed in the fluid. To our knowledge, there seems to be no other work on implementing the immersed interface method for solving the incompressible Navier–Stokes flow with different viscosity across a moving elastic membrane.

Our immersed interface strategy is based on the approach proposed by Li et al. [14] with the introduction of two augmented variables that are defined only along the interface so that the jump conditions can be made decoupled. In the proposed method, the augmented interface variables are defined as the velocity components at the interface which determine the motion of the interface and imposed to satisfy the continuous condition of the velocity across the interface. The idea is similar to [19] where the singular forces at the rigid boundaries are determined to enforce the prescribed velocity condition. In this work, the augmented interface variables are determined by solving a small system of equations via the GMRES iterative 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 interpolated using cubic splines. The forces associated with the elastic membrane exerting on the fluid are computed from the configuration of the flexible membrane and are applied to the fluid through the jump conditions. The position of the flexible elastic membrane is updated implicitly within each time step. The Navier–Stokes equations are discretized on a staggered Cartesian grid by a second order accurate projection method for the pressure and velocity quantities via incorporating the spatial and temporal jump contributions. In the case of a discontinuous viscosity, the augmented velocity across the interface is discontinuous; this gives rise to jumps of the augmented velocity and temporal derivatives of the augmented velocity and hence contributed to the jump condition for the temporal discretizations of augmented velocity at a grid point when it is crossed by an immersed interface. It is shown that the inclusion of the temporal jump contributions has non-negligible effect on simulation results and numerical stability due to the discontinuity of the augmented velocity when the grids are crossed by the interface.

This paper is organized as follows. In Section 2, the model of incompressible Navier–Stokes equations with piecewise constant viscosity across a moving elastic membrane is described, and the corresponding decoupled jump conditions are also presented. The numerical algorithm and numerical implementation are presented in Sections 3 and 4, respectively. In Section 5, we present some numerical results. Some concluding remarks are made in Section 6.

2. Governing equations

Let Ω be a two-dimensional bounded domain that contains a material interface Γ. We shall consider the incompressible Navier–Stokes equations, written as

$$\rho(\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u}) + \nabla p = \nabla\cdot\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}}) + \mathbf{F}(\mathbf{x},t) + \mathbf{g}(\mathbf{x},t), \quad \mathbf{x}\in\Omega, \tag{2.1}$$

$$\nabla\cdot\mathbf{u} = 0, \quad \mathbf{x}\in\Omega, \tag{2.2}$$

with boundary conditions

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

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

$$\mathbf{F}(\mathbf{x},t) = \int_\Gamma \mathbf{f}(s,t)\delta(\mathbf{x}-\mathbf{X}(s,t))\mathrm{d}s. \tag{2.3}$$

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, $\mathbf{f} = (f_1,f_2)^{\mathrm{T}}$ is the force density, and $\delta(\cdot)$ is the Dirac delta function defined in the distribution sense. The body force term $\mathbf{g}$ may also have a finite jump across the interface Γ as well. The interface Γ separates the fluid into two regions Ω^+ and Ω^- with $\Omega = \Omega^+\cup\Gamma\cup\Omega^-$. We use Ω^+ to express the exterior region of the interface, and Ω^- is enclosed by the interface. We refer the readers to Fig. 1 for an illustration of the problem. The motion of the interface satisfies

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

The above equation means the interface moves at the same velocity as the local fluid; thus the velocity field is continuous across the interface (i.e., $[\mathbf{u}] = 0$). Eqs. (2.3) and (2.4) represent the interaction between the interface and the fluid.

In this model, we consider an immersed moving interface problems which involves an elastic membrane, where the force strength $\mathbf{f}$ exerted by elastic membrane on the fluid of the form is given by

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

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

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

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

$$\boldsymbol{\tau}(s,t) = \frac{\partial\mathbf{X}}{\partial s_0}\Big/\left|\frac{\partial\mathbf{X}}{\partial s_0}\right|.$$

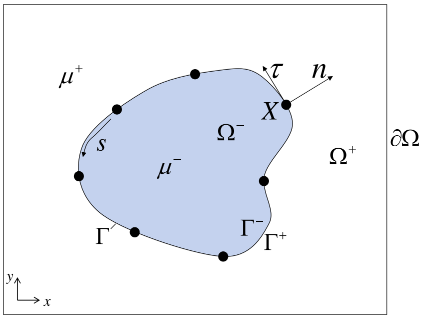


Fig. 1. A typical domain with a flexible elastic boundary represented by some Lagrangian control points. The domain Ω^+ and Ω^- are divided by a closed curve Γ across which the viscosity is a piecewise constant. We use $\mathbf{n}$ and $\boldsymbol{\tau}$ to denote the unit outward normal and tangential directions of the boundary, respectively.

Thus, the force density can be computed directly from the location $\mathbf{X}$ of the membrane Γ. Note that at the relaxed state, $\left|\frac{\partial \mathbf{X}}{\partial s_0}\right| = 1$ and tension vanishes. An equivalent form of Eq. (2.5) is

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

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

Throughout this paper, we shall assume that the fluid density ρ is constant over the whole domain. We also assume that the fluid has discontinuous viscosity and the viscosity μ is a piecewise constant across the interface as

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

where μ^+ and μ^- are two positive constants. This means the jump of viscosity across the interface is a constant, i.e., $[\mu] \equiv$ constant.

2.1. Decoupled jump conditions across the membrane

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

$$[q] = \lim_{\epsilon\to 0^+} q(\mathbf{X} + \epsilon\mathbf{n}) - \lim_{\epsilon\to 0^+} q(\mathbf{X} - \epsilon\mathbf{n}). \tag{2.9}$$

When the viscosity is discontinuous across the membrane Γ, the jump conditions for the pressure and the velocity are coupled together [29] and summarized as follows:

$$[p] = 2\left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]\cdot\mathbf{n} + \widehat{f}_1, \tag{2.10}$$

$$\left[\frac{\partial p}{\partial \mathbf{n}}\right] = [\mathbf{g}]\cdot\mathbf{n} + \frac{\partial \widehat{f}_2}{\partial \tau} + 2\frac{\partial^2[\mu\mathbf{u}]}{\partial \eta^2}\cdot\mathbf{n} - 2\kappa\left[\mu\frac{\partial \mathbf{u}}{\partial \tau}\right]\cdot\boldsymbol{\tau}, \tag{2.11}$$

$$\left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]\cdot\boldsymbol{\tau} + \left[\mu\frac{\partial \mathbf{u}}{\partial \tau}\right]\cdot\mathbf{n} + \widehat{f}_2 = 0, \tag{2.12}$$

$$\left[\mu\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right]\cdot\mathbf{n} + \left[\mu\frac{\partial \mathbf{u}}{\partial \tau}\right]\cdot\boldsymbol{\tau} = 0, \tag{2.13}$$

$$[\mu\nabla\cdot\mathbf{u}] = 0, \tag{2.14}$$

where $\widehat{f}_1$ and $\widehat{f}_2$ are the force density in the normal and tangential directions, denoting $\hat{\mathbf{f}} = (\widehat{f}_1, \widehat{f}_2)$. This coupling causes numerical difficulty in implementation. There are different ways to introduce augmented variables so that the jump conditions for the pressure and the velocity can be decoupled. For example, one could introduce $\left[\mu\frac{\partial \mathbf{u}}{\partial n}\right]$ or $[\mu\mathbf{u}]$ as the augmented variable. Different augmented variables may lead to different algorithms. For flexible elastic membrane problems, one important attention is on the motion of the membrane which is determined by the velocity at the membrane. Based on this consideration, we introduce the velocity components at the membrane interface as two augmented variables to obtain the decoupled jump conditions for use in the immersed interface method in this work. We denote (ξ, η) the local coordinates associated with the directions of $\mathbf{n}$ and $\boldsymbol{\tau}$, respectively, and define the augmented velocity field

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

Note the jump relation $\left[\frac{\partial \mathbf{u}}{\partial \tau}\right] = 0$ due to $[\mathbf{u}] = 0$, then we can rewrite the two jump conditions (2.12) and (2.13) as follows:

$$\left[\frac{\partial \tilde{u}}{\partial n}\right] = \left(\widehat{f}_2 + [\mu]\frac{\partial \mathbf{u}}{\partial \tau}\cdot\mathbf{n}\right)n_2 - [\mu]\frac{\partial \mathbf{u}}{\partial \tau}\cdot\boldsymbol{\tau}\tau_2, \tag{2.15}$$

$$\left[\frac{\partial \tilde{v}}{\partial n}\right] = -\left(\widehat{f}_2 + [\mu]\frac{\partial \mathbf{u}}{\partial \tau}\cdot\mathbf{n}\right)n_1 + [\mu]\frac{\partial \mathbf{u}}{\partial \tau}\cdot\boldsymbol{\tau}\tau_1. \tag{2.16}$$

Next, we define the velocity components at the membrane interface $u(X(s,t), Y(s,t))$ and $v(X(s,t), Y(s,t))$ as two augmented interface variables; these are used directly to determine the motion of the membrane subsequently. Let $\mathbf{q}(s) = (q_1(s), q_2(s)) = \mathbf{u}(X(s,t), Y(s,t))$. As such, we have the following jump conditions

$$[p] = \widehat{f}_1 - 2[\mu]\frac{\partial \mathbf{q}}{\partial \eta}\cdot\boldsymbol{\tau}, \quad [p_\xi] = [\mathbf{g}]\cdot\mathbf{n} + \frac{\partial \widehat{f}_2}{\partial \eta} + 2[\mu]\frac{\partial^2 \mathbf{q}}{\partial \eta^2}\cdot\mathbf{n} - 2[\mu]\kappa\frac{\partial \mathbf{q}}{\partial \eta}\cdot\boldsymbol{\tau}, \tag{2.17}$$

$$[p_\eta] = \frac{\partial \widehat{f}_1}{\partial \eta} - 2[\mu]\left(\frac{\partial^2 \mathbf{q}}{\partial \eta^2}\cdot\boldsymbol{\tau} + \kappa\frac{\partial \mathbf{q}}{\partial \eta}\cdot\mathbf{n}\right), \tag{2.18}$$

$$[\tilde{\mathbf{u}}] = [\mu]\mathbf{q}, \quad [\tilde{u}_\xi] = \left(\widehat{f}_2 + [\mu]\frac{\partial \mathbf{q}}{\partial \eta}\cdot\mathbf{n}\right)n_2 - [\mu]\left(\frac{\partial \mathbf{q}}{\partial \eta}\cdot\boldsymbol{\tau}\right)\tau_2, \tag{2.19}$$

$$[\tilde{\mathbf{u}}_\eta] = [\mu]\frac{\partial \mathbf{q}}{\partial \eta}, \quad [\tilde{v}_\xi] = -\left(\widehat{f}_2 + [\mu]\frac{\partial \mathbf{q}}{\partial \eta}\cdot \mathbf{n}\right)n_1 + [\mu]\left(\frac{\partial \mathbf{q}}{\partial \eta}\cdot \boldsymbol{\tau}\right)\tau_1, \tag{2.20}$$

$$[\tilde{\mathbf{u}}_{\eta\eta}] = [\mu]\frac{\partial^2 \mathbf{q}}{\partial \eta^2} - \kappa[\tilde{\mathbf{u}}_\xi], \quad [\tilde{\mathbf{u}}_{\xi\eta}] = \frac{d}{d\eta}\left[\frac{\partial \tilde{\mathbf{u}}}{\partial n}\right] + [\mu]\kappa\frac{\partial \mathbf{q}}{\partial \eta}, \tag{2.21}$$

$$[\tilde{\mathbf{u}}_{\xi\xi}] = -[\tilde{\mathbf{u}}_{\eta\eta}] + [p_\xi]\mathbf{n} + [p_\eta]\boldsymbol{\tau} - [\mathbf{g}]. \tag{2.22}$$

In our computations, we approximate $\frac{d}{d\eta}\left[\frac{\partial \tilde{\mathbf{u}}}{\partial n}\right]$ by using a cubic spline interpolation from the known $\left[\frac{\partial \tilde{\mathbf{u}}}{\partial n}\right]$ and taking its derivative in this work. From (2.17)–(2.22), it is noted that, if $\mathbf{q}$ are known, then all the jump conditions, say $[p]$, $[p_\xi]$, $[p_\eta]$, $[\tilde{\mathbf{u}}]$, $[\tilde{\mathbf{u}}_\xi]$, $[\tilde{\mathbf{u}}_\eta]$, $[\tilde{\mathbf{u}}_{\eta\eta}]$, $[\tilde{\mathbf{u}}_{\xi\eta}]$, $[\tilde{\mathbf{u}}_{\xi\xi}]$, are known. We note that from expressions (2.17)–(2.22) the values of the jumps of the first and second derivatives of velocity and pressure taken with respect to the (x, y) coordinates are obtained by a simple coordinate transformation presented in [15]. For instance, we have

$$[\tilde{\mathbf{u}}_x] = [\tilde{\mathbf{u}}_\xi]n_1 + [\tilde{\mathbf{u}}_\eta]\tau_1, \quad [\tilde{\mathbf{u}}_y] = [\tilde{\mathbf{u}}_\xi]n_2 + [\tilde{\mathbf{u}}_\eta]\tau_2, \tag{2.23}$$

$$[\tilde{\mathbf{u}}_{xx}] = [\tilde{\mathbf{u}}_{\xi\xi}]n_1^2 + 2[\tilde{\mathbf{u}}_{\xi\eta}]n_1\tau_1 + [\tilde{\mathbf{u}}_{\eta\eta}]\tau_1^2, \tag{2.24}$$

$$[\tilde{\mathbf{u}}_{yy}] = [\tilde{\mathbf{u}}_{\xi\xi}]n_2^2 + 2[\tilde{\mathbf{u}}_{\xi\eta}]n_2\tau_2 + [\tilde{\mathbf{u}}_{\eta\eta}]\tau_2^2. \tag{2.25}$$

So, by the above strategy, the jump conditions for the velocity, the pressure and their derivatives are decoupled. As such, the immersed interface method for incompressible Navier–Stokes flows with discontinuous viscosity can be implemented as before (for example as in [19]).

Remark 2.1. If the viscosity μ is the same constant across the membrane, then (2.17)–(2.22) reduce to the jump conditions for the pressure, velocity and their derivatives as derived previously in [15,19,18] where the jump conditions for $\mathbf{u}$ and p are decoupled, i.e. the terms involving $[\mu]$ in Eqs. (2.17)–(2.22) are zero due to $[\mu] = 0$.

3. Numerical algorithm

Our numerical algorithm is based on the pressure-increment projection algorithm for the discretization of the Navier–Stokes equations with special treatment at the grid points near the interface [19]. The spatial discretization is carried out on a standard marker-and-cell (MAC) staggered grid similar to that found in Harlow and Welch [9]. We use a uniform MAC grid with mesh width $h = \Delta x = \Delta y$ in the computation. With the MAC mesh, the pressure field is defined at the cell center (i,j), where $i \in \{1, 2, \ldots, N_x\}$ and $j \in \{1, 2, \ldots, N_y\}$. The velocity fields u and v are defined at the vertical edges and horizontal edges of a cell, respectively. The pressure and the velocity components u and v are arranged as in Fig. 2.

In our numerical scheme, we use a set of control points $\mathbf{X}_k = (X_k, Y_k)$ to represent the immersed boundary, as marked by circles in Fig. 2, where $k \in \{1, 2, \ldots, N_b\}$. We calculate the geometric quantities and the singular force density and their derivatives at the control points on the interface Γ. We then use periodic cubic splines to interpolate these quantities at the intersection points between grid lines and interface Γ.

3.1. Projection method

The pressure-increment procedure for problems with immersed interfaces is analogous to the projection method presented in [2]. The discretization of the Navier–Stokes equations at those grid points near the interface needs to be modified to account for the jump conditions across the interface due to the presence of singular forces at the interface. Given the velocity $\mathbf{u}^n$, the augmented velocity $\tilde{\mathbf{u}}^n$, and the pressure $p^{n-1/2}$, we compute the augmented velocity $\tilde{\mathbf{u}}^{n+1}$, the velocity $\mathbf{u}^{n+1}$ and pressure $p^{n+1/2}$ at the next time step as follows:

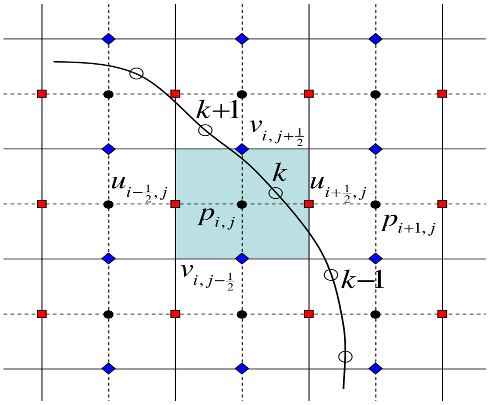


Fig. 2. The staggered grid crossed by an interface, with arrangements of the velocity components and the pressure.

Step 1: Compute an intermediate augmented velocity field $\tilde{\mathbf{u}}^*$ by solving

$$\frac{\tilde{\mathbf{u}}^* - \tilde{\mathbf{u}}^n}{\mu\Delta t} + \frac{1}{\mu}(\mathbf{u}\cdot\nabla\tilde{\mathbf{u}})^{n+\frac{1}{2}} = -\frac{1}{\rho}\nabla p^{n+\frac{1}{2}} + \frac{1}{2\rho}(\Delta_h\tilde{\mathbf{u}}^* + \Delta_h\tilde{\mathbf{u}}^n) + \frac{1}{\rho}\mathbf{g}^{n+\frac{1}{2}} + \mathbf{C}_1 + \gamma_1\frac{1}{\rho}[\Delta\tilde{\mathbf{u}}]_\chi, \quad \tilde{\mathbf{u}}^*|_{\partial\Omega} = \tilde{\mathbf{u}}_b^{n+1}, \tag{3.1}$$

where the advection term is extrapolated using the formula

$$(\mathbf{u}\cdot\nabla\tilde{\mathbf{u}})^{n+\frac{1}{2}} = \frac{3}{2}(\mathbf{u}\cdot\nabla_h\tilde{\mathbf{u}})^n - \frac{1}{2}(\mathbf{u}\cdot\nabla_h\tilde{\mathbf{u}})^{n-1} + \mathbf{C}_2 + \gamma_2\{\mathbf{u}\cdot[\nabla\tilde{\mathbf{u}}]\}_\chi, \tag{3.2}$$

and the pressure gradient is approximated simply as

$$\nabla p^{n+\frac{1}{2}} = G^{\mathrm{MAC}}p^{n-\frac{1}{2}} + \mathbf{C}_3 + \gamma_3[\nabla p]_\chi. \tag{3.3}$$

The above can be rewritten in the following Helmholtz equations form:

$$\lambda_0\tilde{\mathbf{u}}^* + \Delta_h\tilde{\mathbf{u}}^* = \mathbf{RHS}, \tag{3.4}$$

where **RHS** is the right-hand side which includes some correction terms, and $\lambda_0 = -\frac{2\rho}{\mu\Delta t}$. The term $[\cdot]_\chi$ in Eqs. (3.1)–(3.3) denotes a jump in time and is only nonzero when the interface crosses the grid point over the time interval considered. The coefficients γ_1, γ_2 and γ_3 correspond to the first order corrections in time. These terms can be computed as shown in [19].

Step 2: Compute a pressure update ϕ^{n+1} by solving the Poisson equation

$$D^{\mathrm{MAC}}(\mu G^{\mathrm{MAC}}\phi^{n+1}) = \frac{\rho}{\Delta t}D^{\mathrm{MAC}}\tilde{\mathbf{u}}^* + C_4, \tag{3.5}$$

with boundary condition

$$\mathbf{n}\cdot\nabla\phi^{n+1}|_{\partial\Omega} = 0.$$

Step 3: Once ϕ^{n+1} is obtained by solving Eq. (3.5), both the pressure and the augmented velocity field $\left(p^{n+\frac{1}{2}}, \tilde{\mathbf{u}}^{n+1}\right)$ are updated as

$$\tilde{\mathbf{u}}^{n+1} = \tilde{\mathbf{u}}^* - \frac{\Delta t\mu}{\rho}G^{\mathrm{MAC}}\phi^{n+1} + \mathbf{C}_5, \tag{3.6}$$

$$p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \phi^{n+1} - \frac{1}{2\rho}D^{\mathrm{MAC}}\tilde{\mathbf{u}}^* + C_6. \tag{3.7}$$

After obtaining the augmented velocity $\tilde{\mathbf{u}}^{n+1}$, the velocity $\mathbf{u}^{n+1}$ can be obtained by using $\mathbf{u}^{n+1} = \frac{\tilde{\mathbf{u}}^{n+1}}{\mu}$.

The coefficients $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$, C_4, $\mathbf{C}_5$ and C_6 are the spatial correction terms added to the finite different equations at the points near the interface to improve the accuracy of the local finite different approximations. These correction terms can be computed by using the generalized finite different formulas presented in [31] if we know the jumps in the solution and their derivatives. We will review how to compute the correction terms in the next section. In addition to the spatial correction terms, we also need to perform correction for the jump in time.In the above expressions, ∇_h and Δ_h are the standard central difference operators, G^{MAC} and D^{MAC} are the MAC gradient and divergence operators, respectively. These operators are defined as

$$\begin{aligned}
\nabla_h\tilde{\mathbf{u}}_{I,J} &= \left(\frac{\tilde{\mathbf{u}}_{I+1,J} - \tilde{\mathbf{u}}_{I-1,J}}{2h}, \frac{\tilde{\mathbf{u}}_{I,J+1} - \tilde{\mathbf{u}}_{I,J-1}}{2h}\right),\\
\Delta_h\tilde{\mathbf{u}}_{I,J} &= \frac{\tilde{\mathbf{u}}_{I+1,J} + \tilde{\mathbf{u}}_{I-1,J} + \tilde{\mathbf{u}}_{I,J+1} + \tilde{\mathbf{u}}_{I,J-1} - 4\tilde{\mathbf{u}}_{IJ}}{h^2},\\
G^{\mathrm{MAC}}_{ij} &= \left(\frac{p_{i+\frac{1}{2},j} - p_{i-\frac{1}{2},j}}{h}, \frac{p_{i,j+\frac{1}{2}} - p_{i,j-\frac{1}{2}}}{h}\right),\\
(D^{\mathrm{MAC}}\tilde{\mathbf{u}})_{i,j} &= \frac{\tilde{u}_{i+\frac{1}{2},j} - \tilde{u}_{i-\frac{1}{2},j}}{h} + \frac{\tilde{v}_{i,j+\frac{1}{2}} - \tilde{v}_{i,j-\frac{1}{2}}}{h},
\end{aligned} \tag{3.8}$$

where $\tilde{u}_{IJ}$ and $\tilde{v}_{IJ}$ correspond to $\tilde{u}_{i+\frac{1}{2},j}$ and $\tilde{v}_{i,j+\frac{1}{2}}$, respectively.

In our projection method, we need to solve, at each time step, two Helmholtz equations for $\tilde{\mathbf{u}}^*$ in (3.1) or (3.4) and one Poisson-like equation for ϕ^{n+1} in (3.5). Since the correction terms in (3.1) or (3.4) and (3.5) only affect the right-hand sides of the discrete systems for the Helmholtz and Poisson equations, there are many methods that can be used to solve these linear system, for example, the conjugate gradient method and multigrid method. In this study, we use the incomplete Cholesky preconditioned CG (ICCG) method to solve these equations in all our simulations.

3.2. Correction terms calculation

In this section, we will illustrate how to evaluate the above-mentioned correction terms $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$, $\mathbf{C}_4$, $\mathbf{C}_5$ and $\mathbf{C}_6$. One of the basic components for determining the correction terms is the generalized finite difference formulas. Here, we briefly review the generalized finite difference formulas and show two particular generalized finite different formulas for demonstration. Assume that the interface cuts a grid line between two grid points at $x = \alpha$, $x_i \leqslant \alpha < x_{i+1}$, $x_i \in \Omega^-$, $x_{i+1} \in \Omega^+$, where Ω^- and Ω^+ denote the region inside and outside the interface, respectively. Then, the following approximations hold for a piecewise twice differentiable function $w(x)$:

$$w_x(x_i) = \frac{w_{i+1} - w_{i-1}}{2h} - \frac{1}{2h}\sum_{m=0}^{2}\frac{(h^+)^m}{m!}[w^{(m)}]_\alpha + O(h^2), \tag{3.9a}$$

$$w_{xx}(x_i) = \frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} - \frac{1}{h^2}\sum_{m=0}^{2}\frac{(h^+)^m}{m!}[w^{(m)}]_\alpha + O(h), \tag{3.9b}$$

where $w^{(m)}$ denotes the mth derivative of w, $w_i = w(x_i)$, $h^+ = x_{i+1} - \alpha$, $h^- = x_i - \alpha$ and h is the mesh width in x-direction. The jump in w and its derivatives are defined as

$$[w^{(m)}]_\alpha = \lim_{x\to\alpha, x\in\Omega^+} w^{(m)}(x) - \lim_{x\to\alpha, x\in\Omega^-} w^{(m)}(x); \tag{3.10}$$

in short, $[\cdot] = [\cdot]_\alpha$, and $w^{(0)} = w$. Note that if the interface cuts a grid line between two grid points $x_i \in \Omega^+$ and $x_{i+1} \in \Omega^-$, these expressions need to be modified by changing the sign of the second terms on the respective right-hand sides. Expressions involving two or more interface crossings could also be derived, we refer the readers to [31] for details. From Eqs. (3.9a) and (3.9b) the correction terms for $w_x(x_i)$ and $w_{xx}(x_i)$ can be defined as

$$C\{w_x(x_i)\} = -\frac{1}{2h}\sum_{m=0}^{2}\frac{(h^+)^m}{m!}[w^{(m)}] + O(h^2), \quad C\{w_{xx}(x_i)\} = -\frac{1}{h^2}\sum_{m=0}^{2}\frac{(h^+)^m}{m!}[w^{(m)}].$$

Thus, the finite difference approximations for the derivatives of a function w at some grid points near the interface include the standard central difference terms plus the additional correction terms. Accordingly, the correction terms $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$, C_4, $\mathbf{C}_5$ and C_6 are evaluated as follows:

$$\mathbf{C}_1 = \frac{-C\{\tilde{\mathbf{u}}_t\}}{\mu} + \frac{1}{2\rho}(C\{\Delta\tilde{\mathbf{u}}^*\} + C\{\Delta\tilde{\mathbf{u}}^n\}), \tag{3.11a}$$

$$\mathbf{C}_2 = \frac{3}{2}C\{(\mathbf{u}\cdot\nabla\tilde{\mathbf{u}})^n\} - \frac{1}{2}C\{(\mathbf{u}\cdot\nabla\tilde{\mathbf{u}})^{n-1}\}, \tag{3.11b}$$

$$\mathbf{C}_3 = C\{\nabla p^{n-\frac{1}{2}}\}, \tag{3.11c}$$

$$C_4 = \rho\frac{C\{\nabla\cdot\tilde{\mathbf{u}}^*\}}{\Delta t} - C\{\nabla(\mu\nabla p^{n+\frac{1}{2}})\} + C\{\nabla(\mu\nabla p^{n-\frac{1}{2}})\}, \tag{3.11d}$$

$$\mathbf{C}_5 = -\frac{\Delta t\mu}{\rho}(C\{\nabla p^{n+\frac{1}{2}}\} - C\{\nabla p^{n-\frac{1}{2}}\}), \tag{3.11e}$$

$$C_6 = -\frac{1}{2\rho}C\{\nabla\cdot\tilde{\mathbf{u}}^*\}. \tag{3.11f}$$

We note that all the correction terms are evaluated at least to first order accuracy. This is sufficient to guarantee second order accuracy globally since our numerical scheme is of second order away from the boundary and only the points near the boundary are treated with a first order scheme. The first term on the right-hand side of (3.11a) is the correction term in time and is only nonzero at the grid points crossed by the interface between time level n and time level $n+1$. This is because the time derivative of the augmented velocity will not be discontinuous in time at any Cartesian grid point crossed by the interface during the time step, which can be accounted for by including additional terms in the Crank–Nicolson algorithm to correct for these jumps.

Assume that as the interface crosses a grid point (i,j) at time χ, $t^n \leqslant \chi \leqslant t^{n+1}$, the correction term $C\{\tilde{\mathbf{u}}_t\}$ in (3.11a) for $\tilde{\mathbf{u}}_t$ at this point is given by

$$C\{\tilde{\mathbf{u}}_t\} = \begin{cases} -\frac{1}{\Delta t}([\tilde{\mathbf{u}}]_\chi + (t^n - \chi)[\tilde{\mathbf{u}}_t]_\chi), & \text{if } t^n \leqslant \chi \leqslant t^{n+\frac{1}{2}}, \\ -\frac{1}{\Delta t}([\tilde{\mathbf{u}}]_\chi + (t^{n+1} - \chi)[\tilde{\mathbf{u}}_t]_\chi), & \text{if } t^{n+\frac{1}{2}} \leqslant \chi \leqslant t^{n+1}. \end{cases} \tag{3.12}$$

The above temporal jump condition of the augmented velocity $[\tilde{\mathbf{u}}_t]$ can be made related to the corresponding spatial jump condition as in [19,33]. In the case of a discontinuous viscosity, noting that the velocity $\mathbf{u}$ is continuous but the augmented velocity $\tilde{\mathbf{u}}$ is discontinuous across the interface, we have $[\tilde{\mathbf{u}}] = [\mu]\mathbf{q}$. By taking time derivative for both sides of $[\tilde{\mathbf{u}}] = [\mu]\mathbf{q}$, we can then obtain the jump condition of the first-order temporal derivative of the augmented velocity

$$[\tilde{\mathbf{u}}_t] = [\mu]\frac{d\mathbf{q}}{dt} - [\mathbf{u}\cdot\nabla\tilde{\mathbf{u}}] = [\mu]\frac{d\mathbf{q}}{dt} - \mathbf{u}\cdot[\nabla\tilde{\mathbf{u}}] = \pm[\tilde{\mathbf{u}}_t]_\chi, \tag{3.13}$$

where the sign depends on the motion of the interface. In particular, we use a plus sign when the grid point moves from the inside of the interface to the outside of the interface, i.e. from Ω^- to Ω^+, and a minus sign when the grid point moves from the outside of the interface to the inside of the interface, i.e. Ω^+ to Ω^-. In (3.11a), (3.11d) and (3.11f), we use the jump conditions for $\tilde{\mathbf{u}}^{n+1}$ to approximate the jump conditions for $\tilde{\mathbf{u}}^*$ as we expect that $\tilde{\mathbf{u}}^*$ is a good approximation for $\tilde{\mathbf{u}}^{n+1}$. To evaluate the correction term $C\{\Delta\tilde{\mathbf{u}}^*\}$ of (3.11a) at a point (I,J) as depicted in Fig. 3, we need to compute $[\tilde{\mathbf{u}}^*_x]$, $[\tilde{\mathbf{u}}^*_{xx}]$ at the intersection point α and $[\tilde{\mathbf{u}}^*_y]$, $[\tilde{\mathbf{u}}^*_{yy}]$ at β using the force strength at time level $n+1$. The correction term $C\{\Delta\tilde{\mathbf{u}}^*\}$ is calculated as follows:

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

and $\Delta\tilde{\mathbf{u}}^*$ is approximated at the point (I,J) as

$$\Delta\tilde{\mathbf{u}}^*(I,J) = \Delta_h\tilde{\mathbf{u}}^*_{I,J} + C\{\Delta\tilde{\mathbf{u}}^*\}_{I,J} + O(h).$$

Similarly, we can compute for the other correction terms in (3.11b), (3.11c), (3.11d), (3.11e), and (3.11f).

4. Numerical implementation

From the previous section, we can see that the solution can be advanced for one time step from t^n to t^{n+1} if the correction terms in the projection steps have been computed. Those corrections incorporate the jump conditions which depend on the augmented variable $\mathbf{q}$ or/and the force $\mathbf{f}$ at the interface, see Eqs. (2.17)–(2.22). At the moment, we first assume that the augmented variable $\mathbf{q}$ in the jump conditions is known, and its evaluation is presented at the end of this section, see Section 4.1. In the following, we shall describe a basic implementation of the proposed algorithm for the incompressible viscous flows with an immersed elastic membrane. Before computing the jump conditions, the force strength at the intersection points needs to be calculated. To do so, we first compute the force strength at the control points using expression (2.5). And then, based on the force strength at the control points, a new cubic spline is determined to approximate the force strength along the interface. The reason for introducing a new cubic spline is that the jump conditions discussed in the previous section depend on further derivatives of the force along the interface. With this new cubic spline, the force strength and its derivatives can be calculated at the intersection points.

Having the jump conditions for pressure and velocity with known force and the augmented variables, we then apply the projection method with the correction terms to advance for the velocity and pressure from t^n to t^{n+1}. The location of the interface is also updated based on the surrounding fluid velocity. To decrease the strict limit of very small time steps with explicit method and increase the stability of the current method, the updated location of the flexible elastic membrane is advanced in time in an implicit manner, according to

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \frac{1}{2}\Delta t\Big(\mathbf{u}^n(\mathbf{X}^n) + \mathbf{u}^{n+1}(\mathbf{X}^{n+1})\Big). \tag{4.1}$$

The new positions of the control points $\mathbf{X}^{n+1}$ are determined by solving a nonlinear system of equations

$$Q(\mathbf{X}^{n+1}) = 0, \tag{4.2}$$

where

$$Q(\mathbf{X}) = \mathbf{X} - \mathbf{X}^n - \frac{1}{2}\Delta t\big(\mathbf{u}^n(\mathbf{X}^n) + \mathbf{u}^{n+1}(\mathbf{X})\big).$$

The BFGS method [28] which is a quasi-Newton method is employed to solve the nonlinear system of Eq. (4.1) iteratively to calculate the location of the flexible interface. For more details on the immersed interface method for flexible interfaces, see [18–20]. In each iteration of the BFGS method, we need to solve the system of Eq. (4.17) for the augmented variable $\mathbf{q}$ at the interface to satisfy the continuous condition of the velocity $\mathbf{u}$. This is necessary because the velocity field and pressure field are updated at every iterations of the BFGS method. In the numerical tests, it takes only a few iterations for the BFGS method.

Since the viscous term is treated implicitly and the convection term is approximated explicitly, the time step is chosen to satisfy the CFL condition

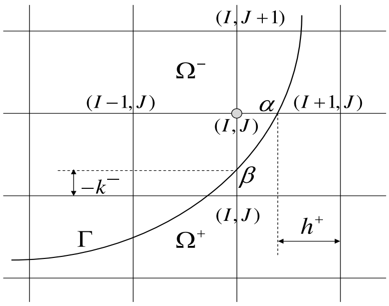


Fig. 3. Interface and mesh geometry near the grid point (I,J) which corresponds to $(i+\frac{1}{2},j)$ for u and $(i,j+\frac{1}{2})$ for v shown in Fig. 2, respectively.

$$\max_{i,j}\left(\left|\frac{u_{i+\frac{1}{2},j}}{h}\right|,\left|\frac{v_{i,j+\frac{1}{2}}}{h}\right|\right)\Delta t \leqslant 1.$$

In summary, given the location of the control points, $\mathbf{X^n}$, the velocity, $\mathbf{u}^n$, the augmented velocity, $\tilde{\mathbf{u}}^n$ and the pressure $p^{n-\frac{1}{2}}$, the algorithm for computing the augmented velocity $\tilde{\mathbf{u}}^{n+1}$ that satisfies the continuous condition of the velocity $\mathbf{u}$ at the interface, pressure $p^{n+\frac{1}{2}}$ and the location of the control points $\mathbf{X}^{n+1}$ can be described as follows:

Algorithm

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 at the flexible interface using expression (2.5).
- Compute the augmented variable $\mathbf{q}$ at the flexible interface to satisfy the continuous condition of the velocity $\mathbf{u}$. That is, calculate the right hand side vector of (4.17). Then solve for the small system of Eq. (4.17) using GMRES method to obtain the augmented variable $\mathbf{q}$ at the interface.

Step 3:

- Employ the projection method as described in Section 3.1 to update the augmented velocity field $\tilde{\mathbf{u}}^{n+1}$ and the pressure. The updated velocity field $\mathbf{u}^{n+1}$ can be obtained directly from the augmented velocity field $\tilde{\mathbf{u}}^{n+1}$. This step involves computing the appropriate correction terms for the spatial and temporal derivatives as described in Section 3.2.
- Compute the velocity $\mathbf{u}^{n+1}(\mathbf{X}^{(k)})$ at $\mathbf{X}^{(k)}$ based on the interpolated augmented 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 augmented 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 B_{k+1}^{n+1} [28]. Set $k = k + 1$ and go to **Step 2**.

4.1. Determination of q at control points in Step 2 of algorithm

Now we start to evaluate the augmented variable $\mathbf{q}$ in **Step 2** of above algorithm in the concerned iteration step of BFGS at time level (t^n, t^{n+1}). The first thing to be done here is to derive the corresponding augmented equations. Assuming the velocity (i.e. $\mathbf{q}$) of the membrane interface at $\mathbf{X}^{(k)}$ is known for this concerned iteration step of BFGS, we then know the jump conditions for the velocity, pressure and their derivatives at this iteration step. Then the augmented velocity field $\tilde{\mathbf{u}}^{n+1}$ at all the grid points can be computed via the projection method as discussed in Section 3.1 with the incorporation of the jumps in the solutions and their derivatives into the difference schemes as mentioned before. The augmented velocity at the control points from the outside of the interface $\mathbf{X}^{(k)}$, $\tilde{\mathbf{U}}_k^+$, can be interpolated from the augmented velocity $\tilde{\mathbf{u}}^{n+1}$ at the grid points. Thus, we can write as

$$\tilde{\mathbf{U}}_k^+ = \tilde{\mathbf{U}}^+(\mathbf{X}^{(k)}) = \mathscr{B}^+(\tilde{\mathbf{u}}^{n+1}), \tag{4.3}$$

where $\mathscr{B}^+$ 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.

In summary, the equations that need to be solved in order to calculate $\tilde{\mathbf{u}}^{n+1}$ and $\tilde{\mathbf{U}}_k^+$, can be written symbolically as

$$\text{Eq. (3.4)} \quad \rightarrow \quad \mathbf{H}\tilde{\mathbf{u}}^* = \mathbf{C} + \mathbf{B_1 f} + \mathbf{A_1 q}, \tag{4.4}$$

$$\text{Eq. (3.5)} \quad \rightarrow \quad \mathbf{L}\boldsymbol{\phi}^{n+1} = \mathbf{D}\tilde{\mathbf{u}}^* + \mathbf{B_2 f} + \mathbf{A_2 q}, \tag{4.5}$$

$$\text{Eq. (3.6)} \quad \rightarrow \quad \tilde{\mathbf{u}}^{n+1} = \tilde{\mathbf{u}}^* - \mathbf{G}\boldsymbol{\phi}^{n+1} + \mathbf{B_3 f} + \mathbf{A_3 q}, \tag{4.6}$$

$$\text{Eq. (4.3)} \quad \rightarrow \quad \tilde{\mathbf{U}}_k^+ = \mathbf{M}\tilde{\mathbf{u}}^{n+1} + \mathbf{B_4 f} + \mathbf{A_4 q}, \tag{4.7}$$

where

$$\mathbf{C} = -\frac{2\rho}{\mu\Delta t}\tilde{\mathbf{u}}^n - \Delta_h\tilde{\mathbf{u}}^n + 2\left[\frac{\rho}{2\mu}(3(\mathbf{u}\cdot\nabla_h\tilde{\mathbf{u}})^n - (\mathbf{u}\cdot\nabla_h\tilde{\mathbf{u}})^{n-1}) + G^{\mathrm{MAC}}p^{n-\frac{1}{2}} - \mathbf{g}^{n+\frac{1}{2}}\right],$$

which can be computed based on the solutions at the previous time step. The operators $\mathbf{H}$, $\mathbf{L}$, $\mathbf{D}$ and $\mathbf{G}$ correspond to $\Delta_h - \frac{2\rho}{\mu\Delta t}$, $D^{\mathrm{MAC}}(\mu G^{\mathrm{MAC}})$, $\frac{\rho}{\Delta t}D^{\mathrm{MAC}}$ and $\frac{\Delta t\mu}{\rho}G^{\mathrm{MAC}}$, respectively. $\mathbf{M}$ is the modified bilinear interpolation operator. $\mathbf{B}_1$–$\mathbf{B}_4$ are some operators that map the forces involved in the jump conditions on the control points to the resulting finite difference approximation by incorporating the correction terms; similarly for the operators $\mathbf{A}_1$–$\mathbf{A}_4$. Eliminating $\tilde{\mathbf{u}}^*$, ϕ^{n+1} and $\tilde{\mathbf{u}}^{n+1}$ from Eqs. (4.4)–(4.7), we can compute the augmented velocity $\tilde{\mathbf{U}}_k^+$ at the control points from the outside of the interface $\mathbf{X}^{(k)}$ as follows:

$$\widetilde{\mathbf{U}}_k^+ = \mathbf{M}(\mathbf{I} - \mathbf{G}\mathbf{L}^{-1}\mathbf{D})\mathbf{H}^{-1}\mathbf{C} + (\mathbf{M}(\mathbf{H}^{-1}\mathbf{B}_1 - \mathbf{G}\mathbf{L}^{-1}\mathbf{D}\mathbf{H}^{-1}\mathbf{B}_1 - \mathbf{G}\mathbf{L}^{-1}\mathbf{B}_2 + \mathbf{B}_3) + \mathbf{B}_4)\mathbf{f} \\ + (\mathbf{M}(\mathbf{H}^{-1}\mathbf{A}_1 - \mathbf{G}\mathbf{L}^{-1}\mathbf{D}\mathbf{H}^{-1}\mathbf{A}_1 - \mathbf{G}\mathbf{L}^{-1}\mathbf{A}_2 + \mathbf{A}_3) + \mathbf{A}_4)\mathbf{q}, \tag{4.8}$$

where $\mathbf{I}$ is an identity matrix. For simplicity, we can re-write (4.8) as

$$\widetilde{\mathbf{U}}_k^+ = \widetilde{\mathbf{U}}_k^{+,0} + \mathbf{A}^+\mathbf{q}, \tag{4.9}$$

where $\widetilde{\mathbf{U}}_k^{+,0}$ corresponds to the augmented velocity at the control points from the outside of the interface at $\mathbf{X}^{(k)}$ obtained by solving Eqs. (3.4), (3.5) and (3.6) with $\mathbf{q} = 0$, given $\mathbf{u}^n$, $\tilde{\mathbf{u}}^n$ and $p^{n-1/2}$. $\mathbf{A}^+$ 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 augmented velocity at the control points from the outside of the interface $\mathbf{X}^{(k)}$ obtained by solving the following equations:

$$\frac{\tilde{\mathbf{u}}_{\mathbf{q}}^*}{\Delta t} = \frac{1}{2\rho}\Delta_h \tilde{\mathbf{u}}_{\mathbf{q}}^* + \overline{\mathbf{C}_1}, \quad \tilde{\mathbf{u}}_{\mathbf{q}}^*|_{\partial\Omega} = 0, \tag{4.10}$$

$$D^{\mathrm{MAC}}(\mu G^{\mathrm{MAC}}\phi_{\mathbf{q}}^{n+1}) = \frac{\rho}{\Delta t}D^{\mathrm{MAC}}\tilde{\mathbf{u}}_{\mathbf{q}}^* + \overline{C_2}, \quad \mathbf{n}\cdot\nabla\phi_{\mathbf{q}}^{n+1}|_{\partial\Omega} = 0, \tag{4.11}$$

$$\tilde{\mathbf{u}}_{\mathbf{q}}^{n+1} = \tilde{\mathbf{u}}_{\mathbf{q}}^* - \frac{\Delta t \mu}{\rho}G^{\mathrm{MAC}}\phi_{\mathbf{q}}^{n+1} + \overline{\mathbf{C}_3}, \tag{4.12}$$

$$\mathbf{A}^+\mathbf{q} = \mathscr{B}^+(\mathbf{u}_{\mathbf{q}}^{n+1}), \tag{4.13}$$

with $\mathbf{q}$ being the velocity $\mathbf{u}(\mathbf{X}^{(k)})$ at the immersed interface $\mathbf{X}^{(k)}$. Here, $\bar{\mathbf{C}}_1$, $\overline{C_1}$ and $\overline{\mathbf{C}_3}$ are the correction terms which only take into account the contribution of $\mathbf{q}$ at the interface to the jump conditions without the contribution of $\hat{\mathbf{f}}$.

Similarly, we also have

$$\widetilde{\mathbf{U}}_k^- = \widetilde{\mathbf{U}}_k^{-,0} + \mathbf{A}^-\mathbf{q}, \tag{4.14}$$

where $\widetilde{\mathbf{U}}_k^-$ is the augmented velocity at the control points from the inside of the interface $\mathbf{X}^{(k)}$ which is interpolated from the velocity $\tilde{\mathbf{u}}^{n+1}$ at the grid points, and $\widetilde{\mathbf{U}}_k^{-,0}$ is simply the augmented velocity at the control points from the inside of the interface $\mathbf{X}^{(k)}$ obtained by solving Eqs. (3.4)–(3.6) with $\mathbf{q} = 0$. $\mathbf{A}^-$ is also a $2N_b \times 2N_b$ matrix, and the vector $\mathbf{A}^-\mathbf{q}$ corresponds to the augmented velocity at the control points from the inside of the interface $\mathbf{X}^{(k)}$. Since the velocity $\mathbf{u}$ is continuous, i.e., $[\mathbf{U}_k] = 0$, then we have

$$0 = \mathbf{U}_k^+ - \mathbf{U}_k^- = \frac{\widetilde{\mathbf{U}}_k^+}{\mu^+} - \frac{\widetilde{\mathbf{U}}_k^-}{\mu^-}. \tag{4.15}$$

From Eqs. (4.9), (4.14) and (4.15), therefore, the augmented variable $\mathbf{q}$ at the control points can be determined by solving

$$\mathbf{A}\mathbf{q} = -\left(\frac{\widetilde{\mathbf{U}}_k^{+,0}}{\mu^+} - \frac{\widetilde{\mathbf{U}}_k^{-,0}}{\mu^-}\right), \tag{4.16}$$

where $\mathbf{A} = \frac{\mathbf{A}^+}{\mu^+} - \frac{\mathbf{A}^-}{\mu^-}$.

There are also other ways to compute the augmented variable $\mathbf{q}$. For example, if q is the exact solution, we have relations $\widetilde{\mathbf{U}}_k^+ = \mu^+\mathbf{q}$ and $\widetilde{\mathbf{U}}_k^- = \mu^-\mathbf{q}$ from $\mathbf{u}^+ = \mathbf{u}^-$ (which means the velocity $\mathbf{u}$ is continuous across the interface). From Eqs. (4.9) and (4.14), we can then get

$$(\mathbf{A}^+ - \mu^+ I)\mathbf{q} = -\widetilde{\mathbf{U}}_k^{+,0}, \tag{4.17}$$

and

$$(\mathbf{A}^- - \mu^- I)\mathbf{q} = -\widetilde{\mathbf{U}}_k^{-,0}, \tag{4.18}$$

respectively. Therefore, the augmented variable $\mathbf{q}$ at control points can be also determined by solving Eqs. (4.17) or (4.18).

Remark 4.1. The augmented variable q at the interface can be determined by solving any one form of Eqs. (4.16)–(4.18) for most cases except for the one special case of the same viscosity. For this special case, we know that $\mathbf{A}^+ = \mathbf{A}^- = \mathbf{A} = 0$. Thus, Eq. (4.16) is not valid, and we have to determine the augmented variable q by solving Eq. (4.17) or Eq. (4.18). For this case, Eqs. (4.17) and (4.18) are reduced to $\mathbf{q} = -\frac{1}{\mu^+}\widetilde{\mathbf{U}}_k^{+,0}$ and $\mathbf{q} = -\frac{1}{\mu^-}\widetilde{\mathbf{U}}_k^{-,0}$, respectively. Note that, in Eq. (4.16), we have to compute two terms on the right hand side. Based on these two considerations, in our computations, the augmented variable q is determined by solving Eq. (4.17) unless it is stated otherwise.

Note that the matrix $\mathbf{A}^+$ depends on the location of the interface and the time step Δt. If the interface is fixed, we will have the same matrix $\mathbf{A}^+$ at every time step if we use the same Δt throughout. In this work, a moving interface is involved, the matrix $\mathbf{A}^+$ changes at each time step and have to be formed at every time step. To avoid generating $\mathbf{A}^+$, we employ GMRES method and solve the linear system (4.17) iteratively. Each iteration of GMRES method requires a matrix-vector product which can be obtained by solving (4.10), (4.11), (4.12). In each matrix-vector product, we have to solve two Helmholtz Eq. (4.10) and a Poisson Eq. (4.11). In our computations, only a few iterations are needed in the GMRES iteration, so the algorithm is effective.

5. Numerical examples

In this section, numerical experiments will be carried out to demonstrate the accuracy and the effectiveness of our algorithm proposed in this work. Throughout this section, we take $\rho = 1$ in all simulations. The ratio of the viscosity is defined as $\lambda = \frac{\mu^+}{\mu^-}$ unless it is stated otherwise.

Example 5.1. Circular flow

In this first example selected, there is provided exact solution, whereby we can compare and ascertain the numerical accuracy of our assembled code based on our proposed immersed interface method for different viscosity across the interface. The exact velocity and the pressure are given by

$$u(x,y,t) = \begin{cases} (1-e^{-t})\left(\frac{y}{r} - 2y\right), & r > \frac{1}{2}, \\ 0, & r \leqslant \frac{1}{2}, \end{cases} \tag{5.1}$$

$$v(x,y,t) = \begin{cases} (1-e^{-t})\left(-\frac{x}{r} + 2x\right), & r > \frac{1}{2}, \\ 0, & r \leqslant \frac{1}{2}, \end{cases} \tag{5.2}$$

$$p(x,y,t) = \begin{cases} \sin(\pi x)\sin(\pi y), & r > \frac{1}{2}, \\ 0, & r \leqslant \frac{1}{2}, \end{cases} \tag{5.3}$$

where $r = \sqrt{x^2+y^2}$. The interface is the circle with the radius $\frac{1}{2}$ and the solution domain is $[-1,1] \times [-1,1]$. It is easy to verify that the velocity satisfies the incompressibility constraint, and it is continuous but has a finite jump in the normal derivative across the interface [15]. The pressure is discontinuous for this example. The external force term $\mathbf{g}$ is computed to satisfy the Navier–Stokes equation (2.1). The force density components in the normal and tangential directions are calculated from (2.10) and (2.12), respectively.

The time step is taken as $\Delta t = \Delta x/8$. We perform the grid refinement analysis for $\lambda = 10$ with $\mu^+ = 1$ and $\mu^+ = 0.1$ at $t = 2$. The errors in the velocity, $E_{\mathbf{u}}$ and the errors in the pressure, E_{p} are measured in the maximum norm at all grid points. The results in Table 1 show that the velocity is second order accurate and the pressure is nearly second order accurate.

Example 5.2. As an extension, this example shows that our method can handle the multi-phase flow problems. The computational domain is $[-1,1] \times [-1,1]$. The initial velocity and pressure are taken to be zero. The interfaces are two circles with radius $r_1 = 0.4$ and $r_2 = 0.7$, respectively. The viscosity is μ_1 outside the outer interface, the viscosity is μ_2 between the outer interface and inner interface, and the viscosity is μ_3 inside the inner interface. Along the outer circle, the normal force

Table 1
Grid refinement analysis for Example 5.1

N	N_b	$\Vert E_{\mathbf{u}}\Vert_\infty$	Order	$\Vert E_p\Vert_\infty$	Order
32	20	1.9154E−03	–	7.5161E−03	–
64	40	4.4982E−04	2.09	2.1107E−03	1.83
128	80	1.1014E−04	2.03	6.1036E−04	1.79
256	160	2.8114E−05	1.97	1.6814E−04	1.86

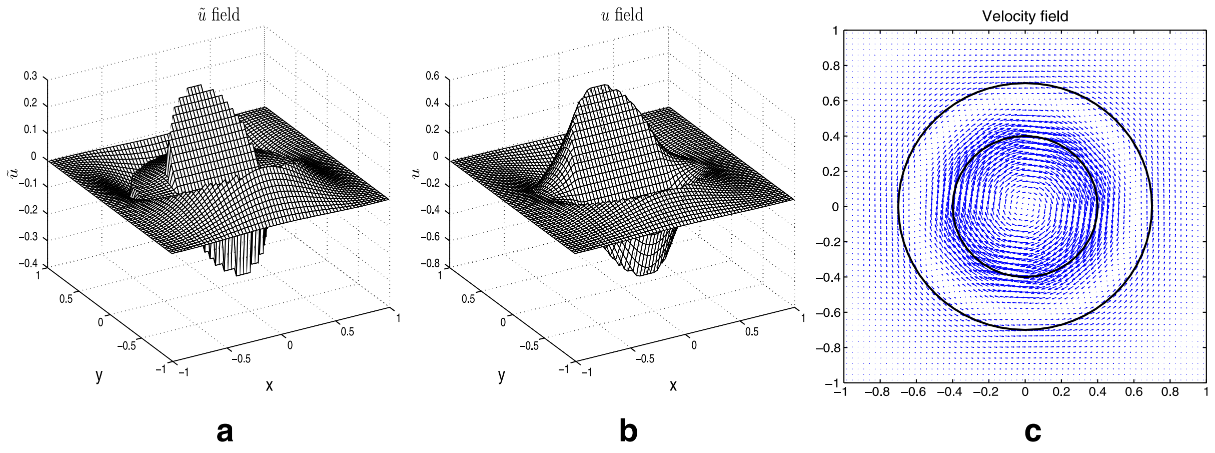


Fig. 4. (a) The x-component of the augmented velocity field $\tilde{\mathbf{u}}$, (b) the x-component of velocity field $\mathbf{u}$, and (c) the velocity field $\mathbf{u}$ at $t = 10$, with $\mu_1 = 1.0$, $\mu_2 = 0.1$, $\mu_3 = 0.5$, $\hat{f}_1^O = 0$, $\hat{f}_2^O = 1.0$, $\hat{f}_1^I = 0$, and $\hat{f}_2^I = -0.5$.

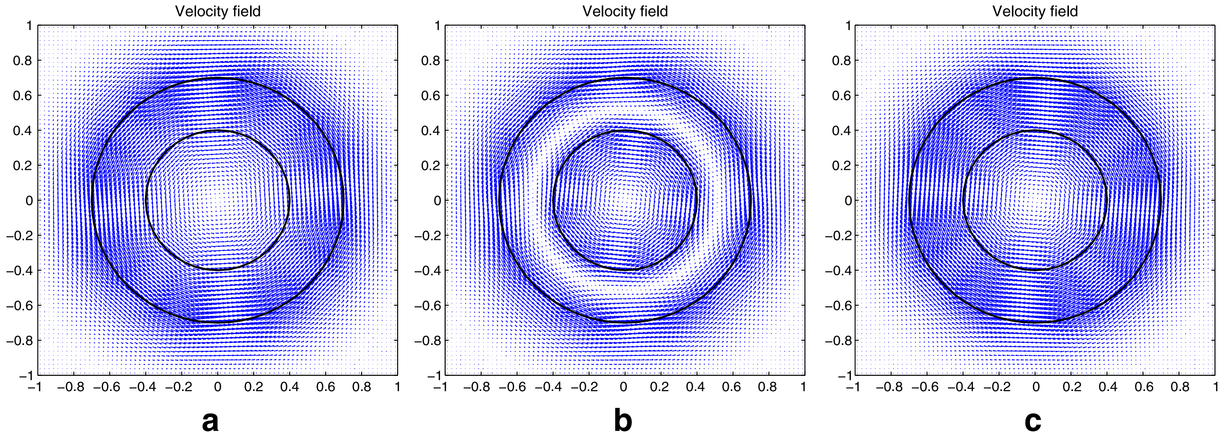


Fig. 5. The velocity field $\mathbf{u}$ at $t = 10$ with $\mu_1 = 0.01$, $\mu_2 = 1.0$, $\mu_3 = 0.1$, $\hat{f}_1^O = 0$, $\hat{f}_2^O = 0.1$, and $\hat{f}_1^I = 0$. (a) $\hat{f}_2^I = -0.2$, (a) $\hat{f}_2^I = -0.3$, and (c) $\hat{f}_2^I = -0.4$.

and tangential force are taken as $\hat{f}_1^O = 0$ and $\hat{f}_2^O = 1.0$, respectively, whereas, along the inner one, the normal force and tangential force are taken as $\hat{f}_1^I = 0$ and $\hat{f}_2^I = -0.5$, respectively. We use a 64×64 grid and set 40 control points and 64 control points on the inner interface and outer interface, respectively. We show the x-component of the augmented velocity $\tilde{\mathbf{u}}$ and the x-component of the velocity $\mathbf{u}$ at $t = 10$ with $\mu_1 = 1.0$, $\mu_2 = 0.1$, and $\mu_3 = 0.5$ in Fig. 4. The velocity field is provided in Fig. 4c. It is clear that the sharp profile of the augmented velocity is well captured. The motion of the steady state is a clockwise rotation along the inner interface and an anti-clockwise rotation along the outer interface. In Fig. 5, we also show the velocity fields at $t = 10$ but with $\mu_1 = 0.01$, $\mu_2 = 1.0$, and $\mu_3 = 0.1$ and different tangential forces along the inner interface and fixed tangential forces along outer interface. In Fig. 5a, the motion of the steady state is a simple anti-clockwise rotation along the two interfaces. In Fig. 5c, the motion of the steady state is a simple clockwise rotation along the two interfaces. In Fig. 5b, the motion of the steady state is a clockwise rotation along the inner interface and an anti-clockwise rotation along the outer interface.

Example 5.3. *Elastic membrane*

In this example, we consider a flexible interface problem which involves an elastic membrane, which is used by Tu and Peskin [30] to test their immersed boundary method, by LeVeque and Li [21] to test their immersed interface method for Stokes flows with same viscosity, and by Lee and LeVeque [20] to test their immersed interface method for Navier–Stokes equations with same viscosity. The initial state of membrane (the solid line in Fig. 6, labeled “Initial”) is an ellipse with the semi-major and semi-minor axes $a = 0.75$, $b = 0.5$, respectively. The computational domain is $[1.5, 1.5] \times [1.5, 1.5]$ and

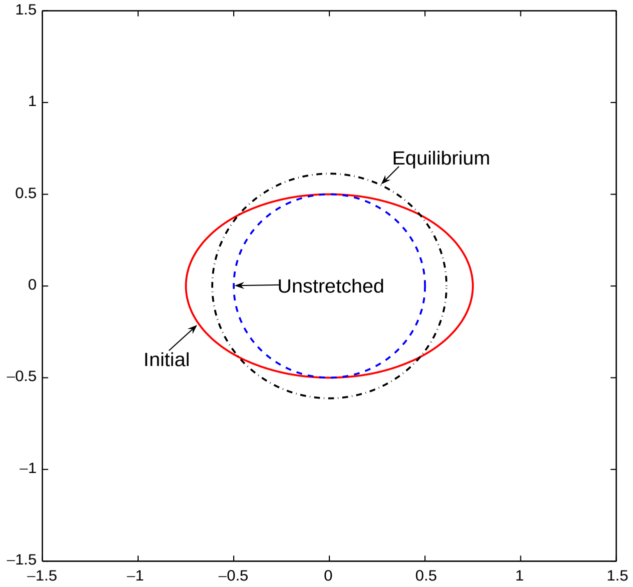


Fig. 6. Initial, unstretched and equilibrium positions of the elastic membrane.

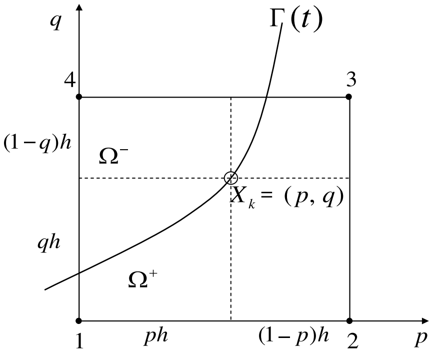


Fig. 7. A diagram of the interpolation of velocity from grid to control point.

the ellipse is located at the center of the computational domain, The unstretched state of membrane (the dashed line in Fig. 6, labeled "Unstretched") is a circle with radius $r_0 = 0.5$. The tension coefficient T_0 is set to 10 in this example unless it is stated otherwise, compared to the tension coefficient taken in the literatures [20,13,14], the problem of this example in this work is more stiffer.

Due to the restoring force, the ellipse will converge to a circle (the dash-dot line in Fig. 6, 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. So the interface is still stretched at the equilibrium state, and the nonzero interface force is balanced by a negative jump $[p]$. In the equilibrium state, the fluid velocity $\mathbf{u}$ is zero everywhere and the pressure p has two different constant values inside and outside the interface. We start our simulation by setting the initial velocity and pressure fields to zero. In our computations, we take $T_0 = 10$ and $\mathbf{u}|_{\partial\Omega} = \mathbf{0}$. In this example, we perform the simulations with a 64×64 grid and 64 control points to represent the interface for all the cases unless otherwise stated.

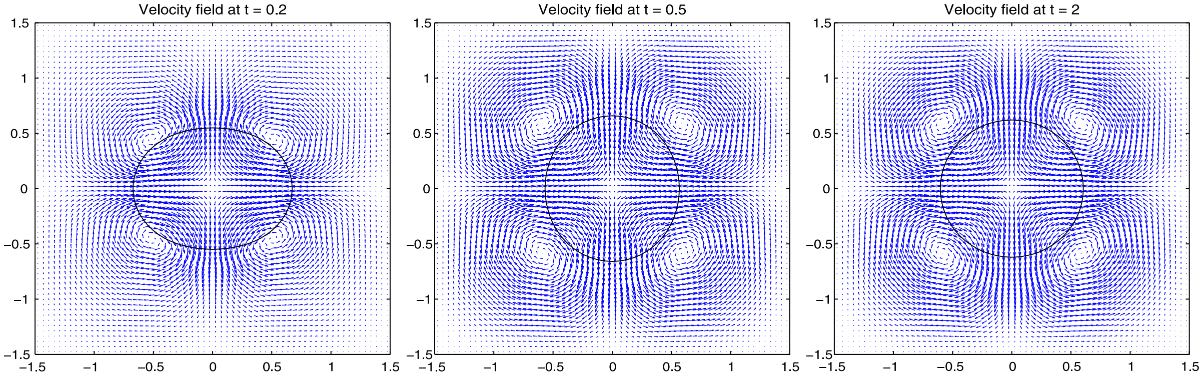
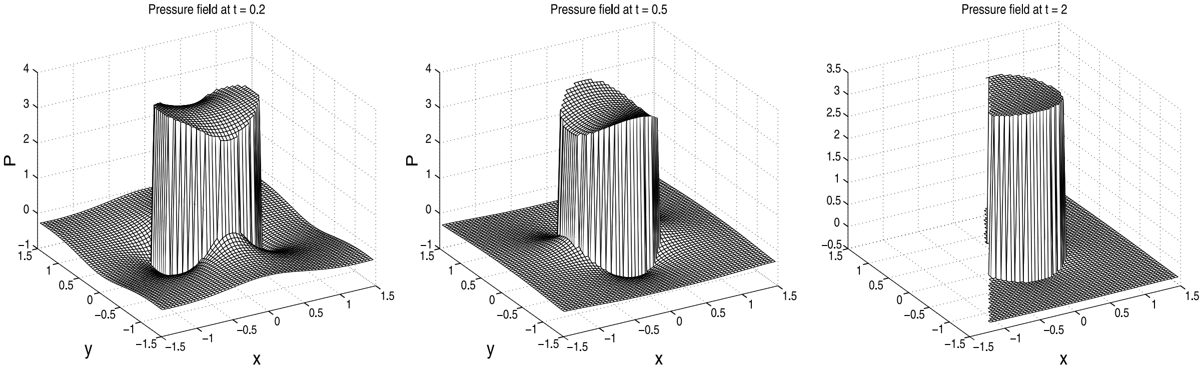


Fig. 8. The velocity field $\mathbf{u}$ at different times with $\mu^+ = \mu^- = 0.1$ (left: $t = 0.2$, middle: $t = 0.5$, right: $t = 2$).



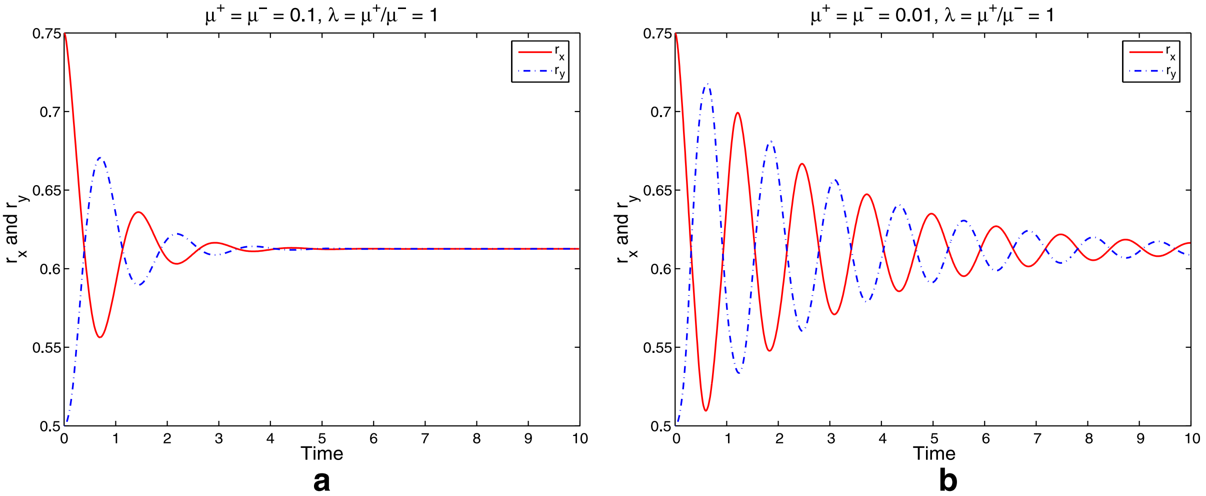
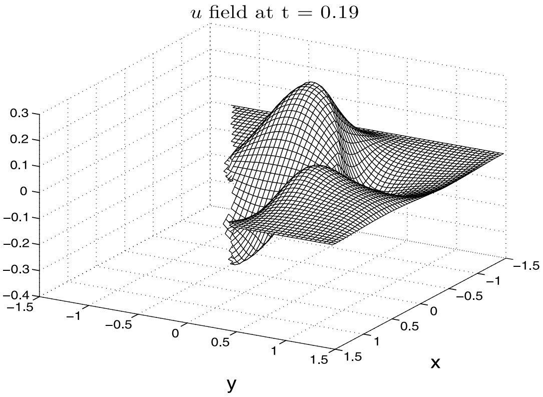


Fig. 10. The evolution of r_x and r_y with: (a) $\mu^+ = \mu^- = 0.1$ and (b) $\mu^+ = \mu^- = 0.01$.

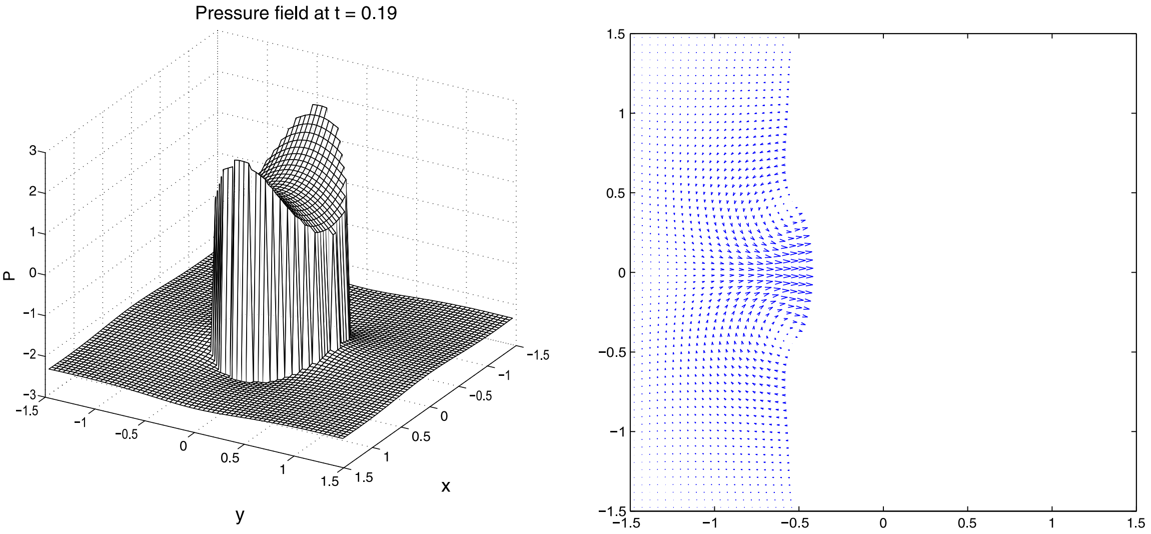
The different viscosity coefficients have different effects on interface motion and solution. The effect depends on the viscosity jump $[\mu]$ and on the viscosity ratio λ, noting that the relations $[\mu] = (\lambda - 1)\mu^- = (1 - 1/\lambda)\mu^+$. We first consider a special simple case where the viscosities are same inside and outside the interface, i.e., $\mu^+ = \mu^-$. In Figs. 8 and 9, we show the veloc-

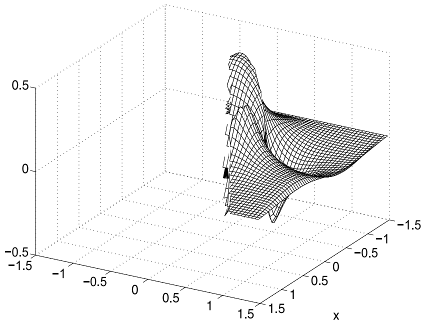


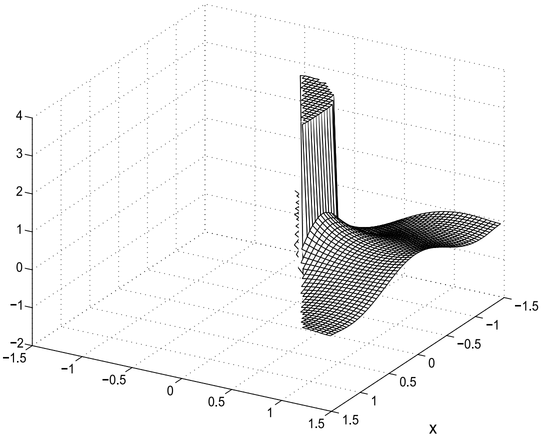
ity fields and the pressure fields at $t = 0.2$, $t = 0.5$ and $t = 2$. The evolutions of the semi-major and semi-minor axes with $\mu^+ = \mu^- = 0.1$ and $\mu^+ = \mu^- = 0.01$ are shown in Fig. 10a and b, respectively. The interface oscillates as it settles down to the equilibrium state. In Fig. 10b, since the fluid viscosity is lower, the elastic membrane takes a longer time to oscillate before settling down to the equilibrium state. All results are in good agreements with those obtained by [19], and are comparable to those in [20].

Next we consider the case where the viscosity outside the interface is less than that inside the interface. We first consider relatively larger viscosity and take the case for $\lambda = 0.1$ with $\mu^+ = 0.1$ and $\mu^- = 1.0$. In Fig. 11, we plot the x-component of velocity field $\mathbf{u}$ (left) and x-component of the augmented velocity field $\tilde{\mathbf{u}}$ (right) at $t = 0.19$ and $t = 1.01$. The pressure field and the velocity field at $t = 0.19$ are presented in Fig. 12. As expected, from these figures, we can see the augmented velocity $\tilde{u}$ and the pressure p are discontinuous across the interface while the velocity u is continuous but not smooth. Fig. 13 shows this more clearly with the plot of cross sections of u, $\tilde{u}$ and p at $t = 0.19$ along the line $y = -0.352$. In Fig. 13b and c, the discontinuities in the augmented velocity $\tilde{u}$ and the pressure are captured very sharply by our algorithm, respectively. Next we keep the viscosity ratio $\lambda = 0.1$ fixed, but decrease the viscosity by taking $\mu^+ = 0.01$ and $\mu^- = 0.1$. The computed velocity field and the pressure field are shown in Figs. 14 and 15.

We also consider another two cases where the viscosity outside the interface is larger than that inside the interface. Compared with the previous two cases, we exchange the viscosities outside and inside the interface for each case. For the case of

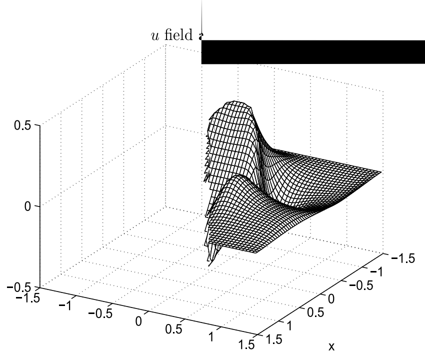



0.5
0
−0.5
−1.5
−1
−0.5
0
0.5
1
1.5
−1.5
−1
−0.5
0
0.5
1
1.5
x



$\lambda = 10$ with $\mu^+ = 1.0$ and $\mu^+ = 0.1$, the *x*-component of the velocity field $\mathbf{u}$ and the *x*-component of the augmented velocity field $\tilde{\mathbf{u}}$ are presented in Fig. 16, and the velocity fields $\mathbf{u}$ at $t = 0.19$ and $t = 1.99$ are presented in Fig. 17. In Fig. 18, we also show the corresponding pressure fields at $t = 0.19$ and $t = 1.99$. For the case of $\lambda = 10$ with $\mu^+ = 0.1$ and $\mu^+ = 0.01$, Fig. 19 shows the velocity field $\mathbf{u}$ and the *x*-component of the velocity field $\mathbf{u}$ and the *x*-component of the augmented velocity field $\tilde{\mathbf{u}}$. The corresponding pressure fields at $t = 0.12$ and $t = 1.48$ are presented in Fig. 20. From these figures, we can see the effect of different viscosities jump and viscosity ratios on the solution and the motion of the interface. In Fig. 21, we have the comparison of the evolving interfaces with different viscosity ratios λ at $t = 1.41$. It is clear from Fig. 21 that with more viscosity the fluid moves slower.

We run the simulation with time step $\Delta t = 1 \times 10^{-4}$ and different resolutions to check the spatial convergence rate. Since the analytic solution is not available, we measure the error in velocity and pressure using a reference solution that is obtained on the finest 512×512 grid. In Table 2, we show the spatial convergence rate analysis at $t = 0.5$, and the expected second order accuracy for the velocity and near second order accuracy for the pressure are observed. We also fix $t = 1.0$



and the spatial resolution with a 128 × 128 grid and 96 control points representing the interface, but use different time steps to check the temporal convergence rate. We compute a reference solution using a very small time step $\Delta t = 5 \times 10^{-5}$. The temporal convergence rate analysis at $t = 0.5$ is shown in Table 3, which indicates that near second-order temporal convergence rate is achieved.

For the purpose of the further comparison of viscosity effects on interface motion over longer times, in Figs. 22–25, we plot the evolutions of the semi-major and semi-minor axes for different cases. With our method, a numerical equilibrium is reached that agrees very well with the true equilibrium, and this equilibrium is then maintained. Particularly, at $t = 10$, the error between semi-major axis and r_e is only $2.7463e - 6$, and the error between semi-minor axis and r_e is only $2.6516e - 6$. In Fig. 22a and b, the viscosities outside and inside the interface are reverse. With the viscosities both outside and inside the interface relatively larger as compared to Fig. 23a and b, we can observe that the interfaces relax gradually to the equilibrium state without oscillations in Fig. 22a while with a small oscillation in Fig. 22b. In Fig. 23a and b, we decrease the viscosities both inside and outside the interface as compared Fig. 22a and b. We can see that the former elastic membranes takes a longer time to oscillate with a few cycles before setting down to the equilibrium state unlike the latter; there are also more oscillations in Fig. 23b than in Fig. 23a. We also show in Fig. 24 the evolutions of the semi-major and semi-minor axes for $\lambda = 0.1$, $\lambda = 1$ and $\lambda = 10$ with fixed $\mu^{+} = 0.1$. It is observed that with larger viscosity ratio the fluids (inside and outside the elastic membrane) move faster. In Fig. 25, the evolutions of the semi-major and semi-minor axes are presented for $\lambda = 0.1$, $\lambda = 1$ and $\lambda = 10$ with fixed $\mu^{-} = 0.1$; the larger the viscosity ratio the fluids move slower.

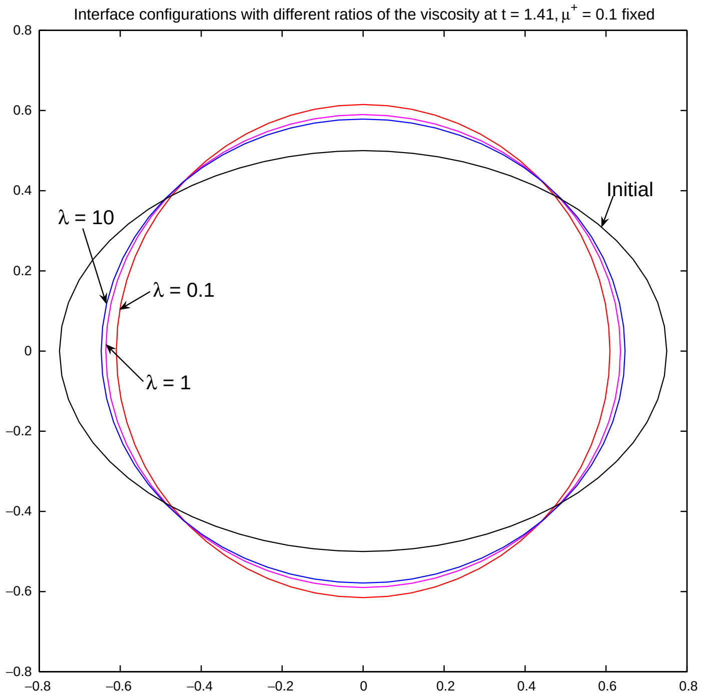


Fig. 21. Comparison of viscosity effects on interface configuration for $\lambda = 0.1$, $\lambda = 1$ and $\lambda = 10$ at $t = 1.41$. The elastic membrane with $\lambda = 0.1$ converges to equilibrium fastest.

Table 2
Spatial convergence analysis for Example 5.3 with $\mu^+ = 1.0$ and $\mu^- = 0.1$

N	N_b	$\|E_{\mathbf{u}}\|_\infty$	Order	$\|E_p\|_\infty$	Order
32	32	1.3115E−02	–	6.5419E−02	–
64	64	2.6444E−03	2.31	2.0558E−02	1.67
128	128	6.9879E−04	1.92	6.1544E−03	1.74
256	256	1.6527E−04	2.08	1.7552E−03	1.81

Table 3
Temporal convergence analysis for Example 5.3 with $\mu^+ = 1.0$ and $\mu^- = 0.1$

Δt	$\|E_{\mathbf{u}}\|_\infty$	Order	$\|E_p\|_\infty$	Order
4×10^{-4}	1.3896E−04	–	1.7642E−03	–
2×10^{-4}	3.8016E−05	1.87	4.7599E−04	1.89
1×10^{-4}	1.0619E−05	1.84	1.3860E−04	1.78

In all the simulations, the conservation of the areas is preserved very well. For example, in Fig. 26a, we present the plots of the absolute error in area versus time up to $t = 10$ for $\lambda = 0.1$ with $\mu^+ = 0.1$ and $\mu^+ = 0.01$. In this figure, the maximum absolute error in area is $3.7756e - 5$ with $\mu^+ = 0.1$ and indicates fairly little leakage of about 0.0032% for this case, while the maximum absolute error in area is $2.3472e - 4$ with $\mu^+ = 0.01$, which is somewhat larger due to lower viscosities outside and inside the interface and indicates a leakage of about 0.02%. The plots of the absolute error in area versus time up to $t = 10$ for $\lambda = 0.1$, $\lambda = 1$ and $\lambda = 10$ with fixed $\mu^+ = 0.1$ are also presented in Fig. 26b. From this figure, we can observe that the larger the viscosity ratio is, the relatively larger the absolute error of the area becomes. Since the area enclosed by elastic membrane should be exactly conserved, we can use the area based on the computed position of the membrane as another measure of the accuracy of the method. Fig. 27 shows that the area is conserved with second-order accuracy using our method.

Next we want to investigate the effect of the temporal jump contributions on the simulation results. The augmented velocity and its temporal derivative at some grid points in space have jumps in terms of time at the instant when the inter-

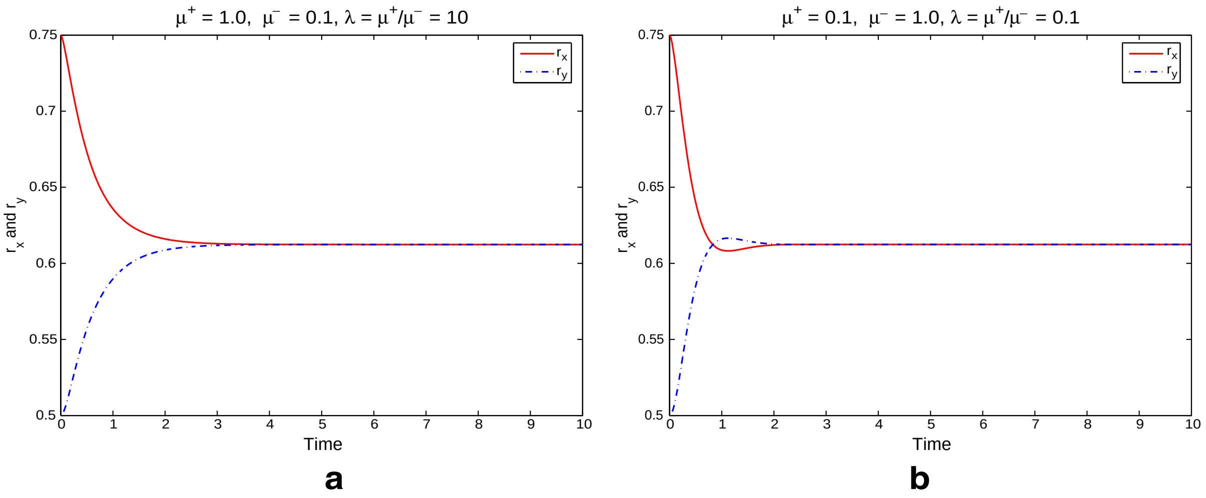


Fig. 22. The evolution of r_x and r_y with: (a) $\mu^+ = 1.0$, $\mu^- = 0.1$, and $\lambda = 10$ and (b) $\mu^+ = 0.1$, $\mu^- = 1.0$, and $\lambda = 0.1$.

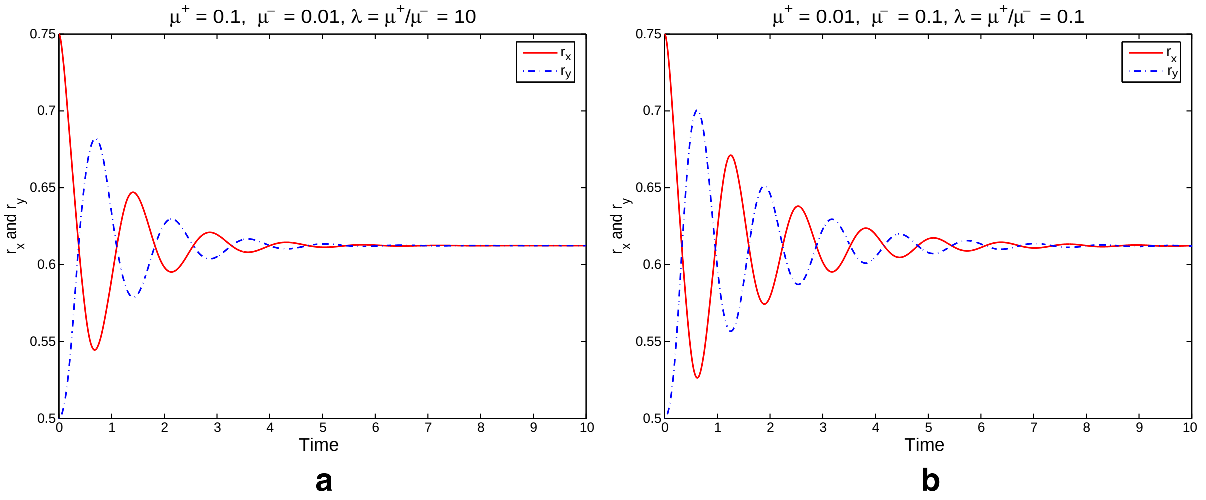


Fig. 23. The evolution of r_x and r_y with (a) $\mu^+ = 0.1$, $\mu^- = 0.01$, and $\lambda = 10$ and (b) $\mu^+ = 0.01$, $\mu^- = 0.1$, and $\lambda = 0.1$.

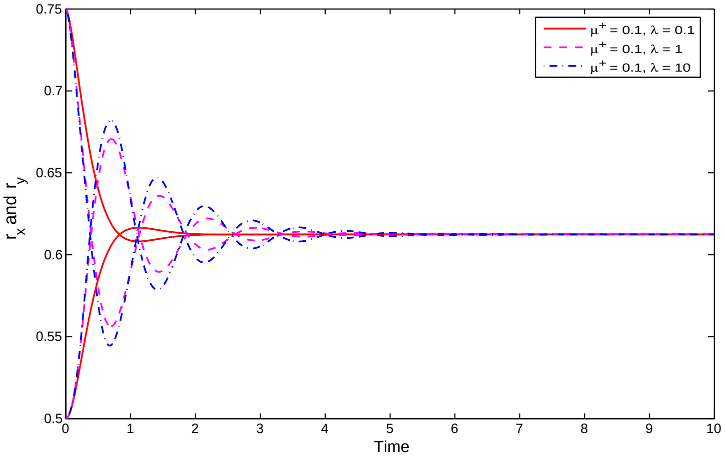
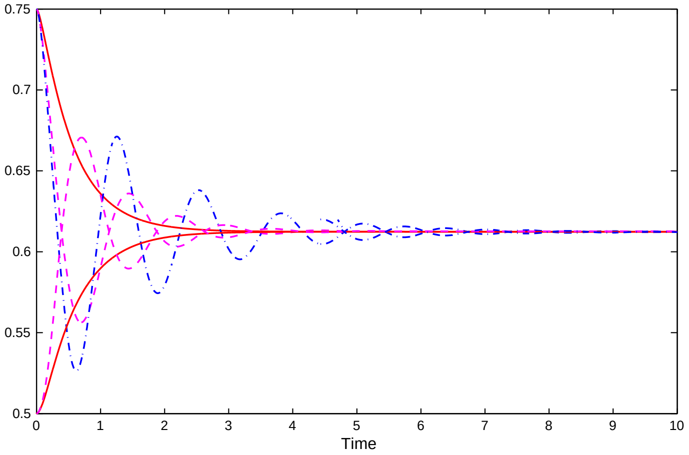
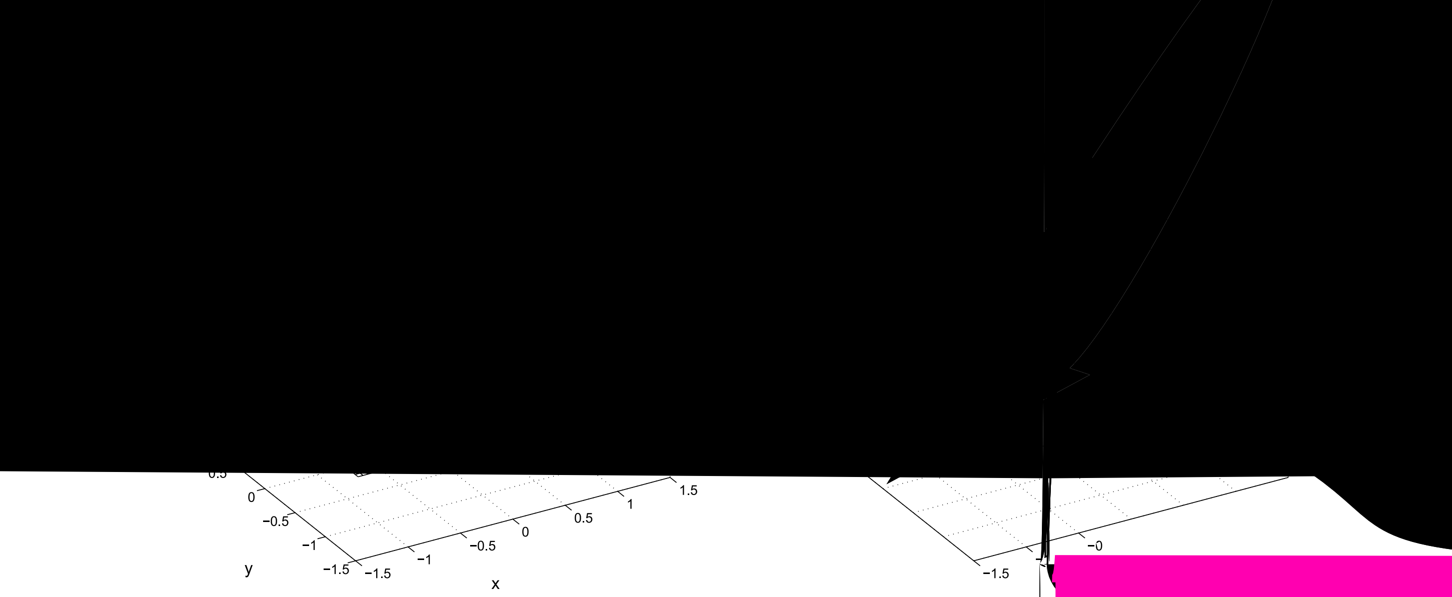


Fig. 24. The evolution of r_x and r_y with fixed $\mu^+ = 0.1$, for $\lambda = 0.1$, $\lambda = 1$, and $\lambda = 10$.


0.75
0.7
0.65
0.6
0.55
0.5
0
1
2
3
4
5
6
7
8
9
10
Time



in Fig. 31. In this case, due to relatively smaller viscosities outside and inside the interface, the t
causes numerical instability during the iterations without the temporal jump contributions a
end. It is shown that the inclusion of the temporal jump contributions has non-negligible effe
when the grids are crossed by the interface across which there are relatively larger differing
regions.

Next, we also want to briefly discuss about the iterations in BFGS and GMRES for ela
Fig. 32a and in Fig. 32b, we present the plots of the number of BFGS iterations and the numb
versus time, respectively, with $\mu^+ = 0.1$, $\mu^- = 1.0$, and $\lambda = 0.1$. From Fig. 32a, we can see th
tions say about $1 \sim 4$ to converge with a tolerance of 10^{-10} for the BFGS. The number o
Fig. 32b is fewer than 20 for this case. In Fig. 33a and in Fig. 33b, we also give the plots of th
and the number of total GMRES iterations versus time, respectively, with $\mu^+ = 0.01$, $\mu^- = 0$.

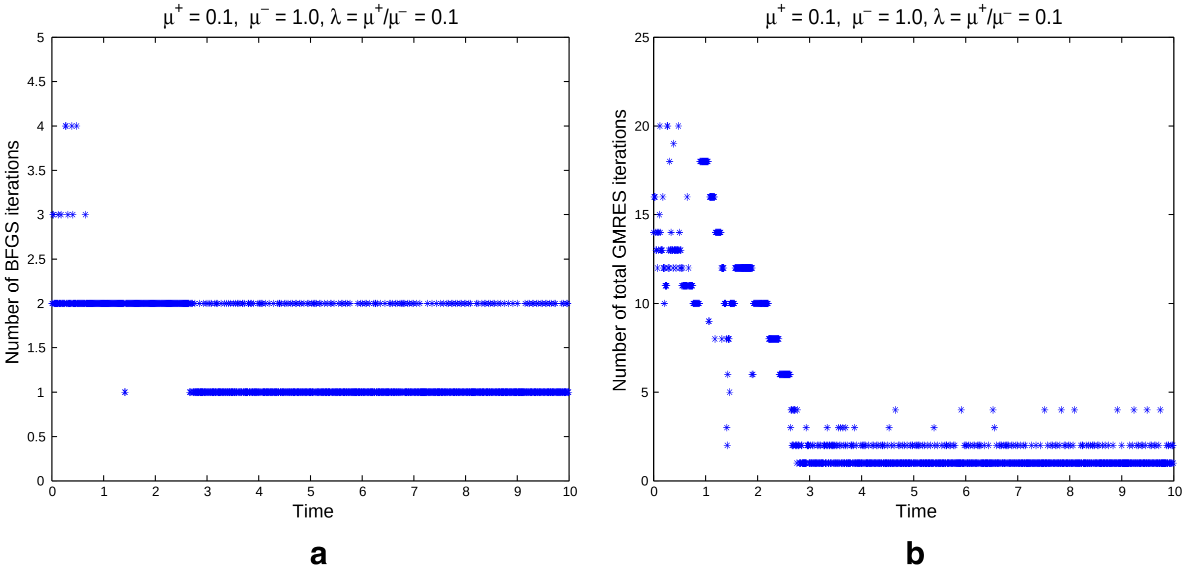
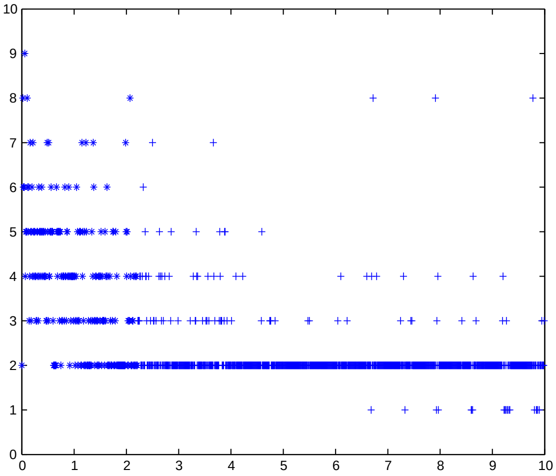
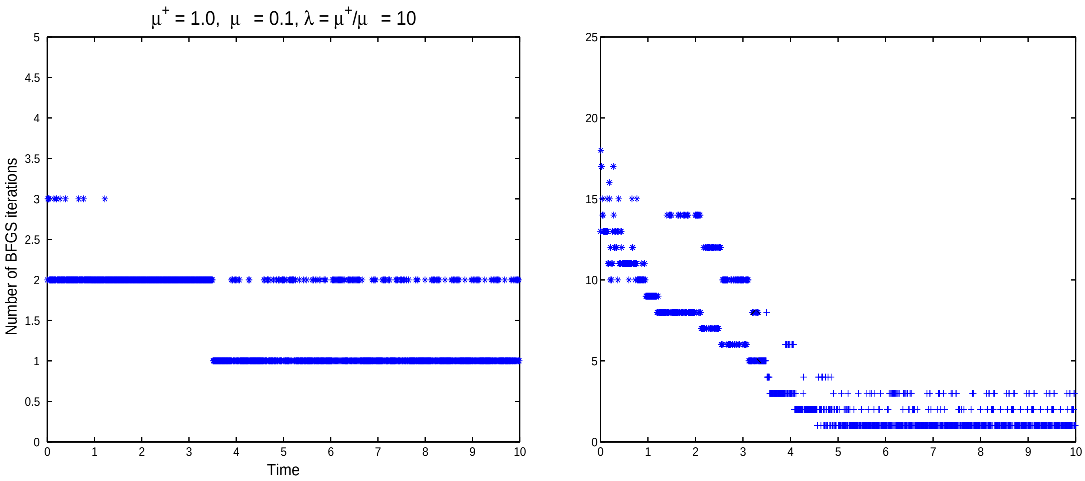


Fig. 32. (a) Number of BFGS iterations, and (b) Number of total GMRES iterations versus time with $\mu^+ = 0.1$, $\mu^- = 1.0$, and $\lambda = 0.1$.



those in Fig. 32a and in Fig. 32b, the number of BFGS iterations and the number of total GMRES iterations are slightly increased although the viscosity ratios are the same, which is not surprising due to the lower viscosities taken for both the outside and inside of the interface. We also show the plots of the number of BFGS iterations and the number of total GMRES iterations versus time in Fig. 34a and b, respectively, with $\mu^+ = 1.0$, $\mu^- = 0.1$, and $\lambda = 1$. These can be compared to those in Fig. 33a and in Fig. 33b, which show that the number of BFGS iterations and the number of total GMRES iterations decrease when one increases the ratio of viscosity with fixed μ^-. On the other hand, the number of BFGS iterations and the number of total GMRES iterations increase when one increases the ratio of viscosity with fixed μ^+; results not shown here. When the viscosity ratio becomes larger or smaller, it requires more iterations. However, in the simulations of the present viscosity ratios applicable for many problems in biological flows, the iteration times for the BFGS is not large and the number of GMRES iterations is still modest for all the present cases. Finally, we have also checked the iteration times with very larger tension coefficients $T_0 = 50$. One can observed from Fig. 35 that the iteration times increase for this stiffer case.



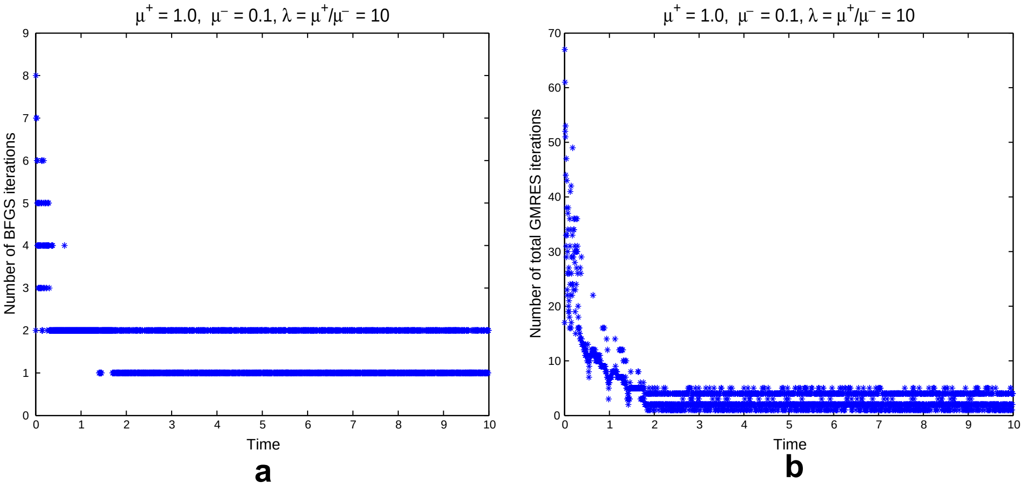


Fig. 35. (a) Number of BFGS iterations, and (b) Number of total GMRES iterations versus time with $\mu^+ = 1.0$, $\mu^- = 0.1$, $\lambda = 10$, and $T_0 = 50$.

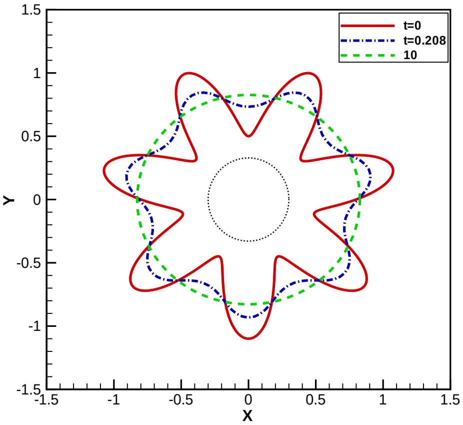
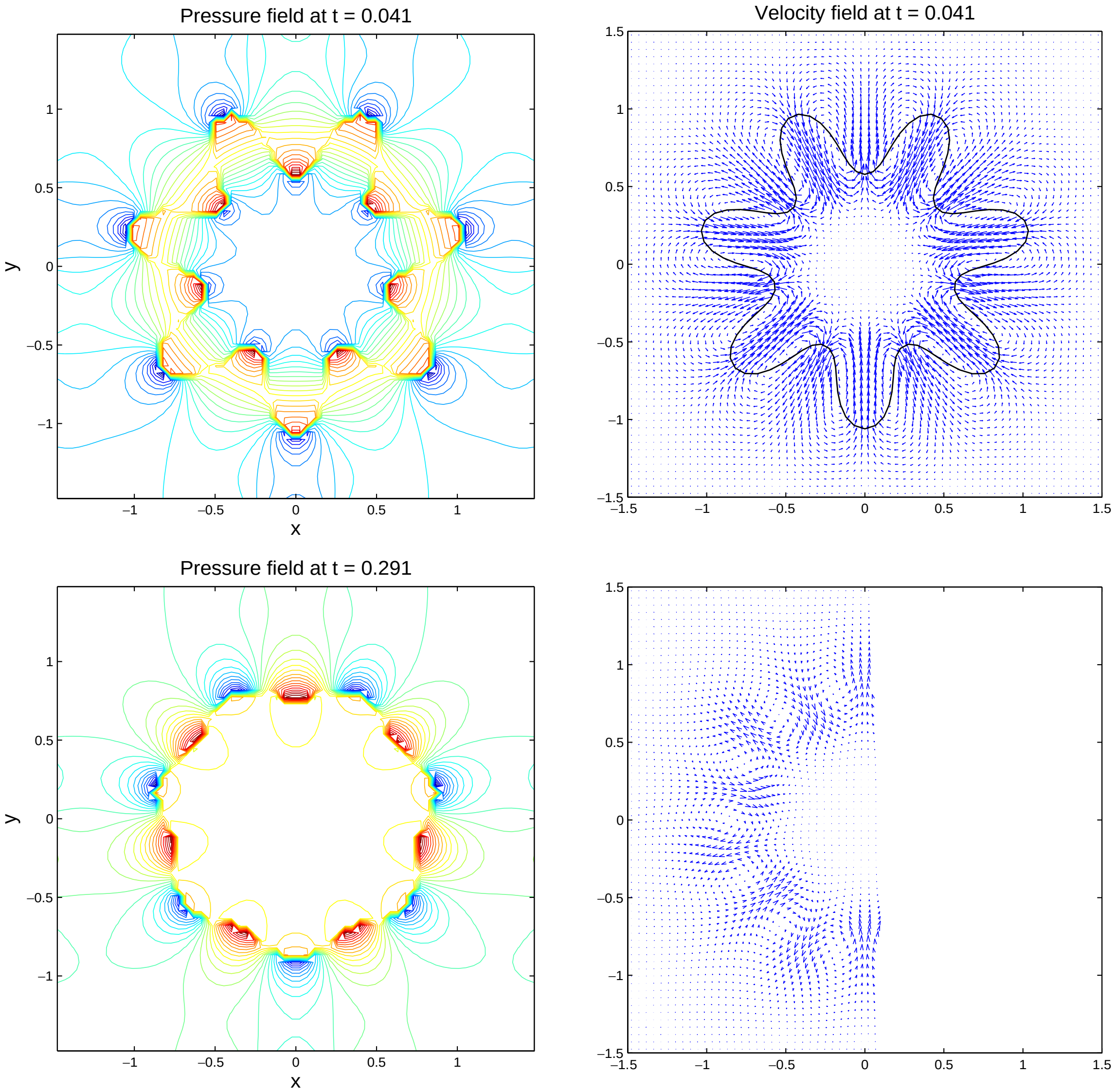


Fig. 36. The location of the interface at $t = 0$ (solid line), $t = 0.208$ (dashdoted line) and $t = 10$ (dashed line), with $\mu^+ = 1.0$, $\mu^- = 0.2$, and $\lambda = 5$. The innermost dotted circle shows the unstretched interface.



Example 5.4. Flower-configuration interface

In this final example, we consider the flow with a more complicated interfaces such as the flower configuration found in the literature [21], and employ to further test our method for a stiff problem. The initial interface is given in polar coordinates by $r(\theta) = 0.8 + 0.3\sin(7\theta)$. The unstretched interface is the circle with the radius $r_0 = 0.3$. The problem is very stiff because the curvature of the interface is large. The tension coefficient T_0 is set to 1, which is already much lar[illegible] those typically taken in the literature [15,33,13] for a complex interface, thus the problem arising [illegible] viscosities outside and inside the interface are taken to be $\mu^+ = 1.0$ and $\mu^- = 0.2$, respectively. [illegible] $[1.5, 1.5] \times [1.5, 1.5]$. We perform this simulation with a 64 × 64 grid and 120 control points [illegible]mpared to the previous example involving a relaxing ellipse, we take a relatively smaller [illegible] 36, we plot the interface at $t = 0$, $t = 0.208$ and $t = 10$. At $t = 10$, the interface is almost [illegible]cle. The pressure contour and the velocity field at $t = 0.041$ and $t = 0.291$ are plotted in [illegible]at our method can capture the highly localized discontinuous profile for the pressure. [illegible] is only about 0.09%, thus the conservation of the area of the flower configuration is [illegible]e. Compared with the other previous examples, the number of total GMRES iterations [illegible]er (see Fig. 38), which is not surprising since the curvature of the interface is much

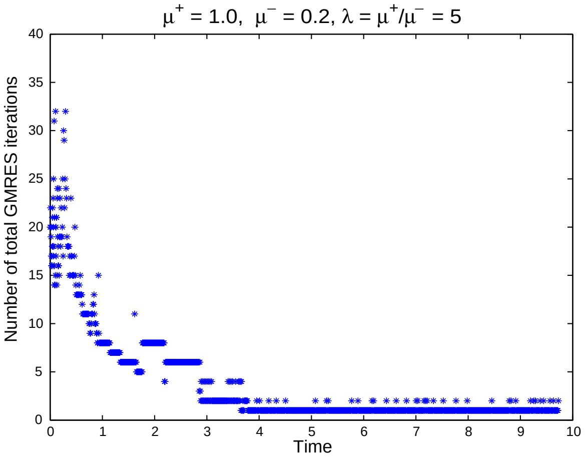


Fig. 38. Number of total GMRES iterations versus time with $\mu^+ = 1.0$, $\mu^- = 0.2$, and $\lambda = 5$.

6. Concluding remarks

In this paper, an immersed interface algorithm is further developed for solving the incompressible Navier–Stokes equations with piecewise constant viscosity across a flexible elastic membrane where the jump conditions for the pressure and the velocity are coupled together, which is a logical and further development of the previous work reported in [29] focussing mainly on the fixed interface problems. The idea of current work is to introduce the velocity components as two augmented variables that are only defined along the interface so that the jump conditions can be decoupled. The augmented interface variables are then solved by the GMRES iterative method. The new location of moving elastic membrane is advanced implicitly using BFGS iteration method within each time step, and the augmented variable along the interface is determined to satisfy the continuous velocity condition at the interface in each iteration of BFGS by solving a small and dense linear system of the augmented equation. Using the proposed method, we assess the effect of different viscosities on the flow solution and membrane motion.

In the present method, for very high viscosity ratio(say 1:1000 or 1000:1 or larger), the method suffers from the stability problems and the iteration fails to converge for the moving elastic membrane over a long time computation. One possible reason is the increased condition number of the matrix associated with the augmented equation and the system solved by GMRES becomes increasingly ill-conditioned. It is reiterated that the present method determines the augmented variables to satisfy the continuous condition at the interface in each iteration of BFGS. Alternatively, one may calculates the augmented variables from the location of interface at the end of time step. This approach results in a coupled nonlinear system for the augmented equation and the interface moving equation. A more sophisticated nonlinear solver should be convergent for the increasingly extreme condition of higher viscosity ratio. Such a solver, however, is likely to lead to high computational costs. So ensuring the stability of the present or improved method under the hallmark of efficiency and simplicity of the algorithm for higher viscosity ratio is still challenging, which is our further work. It is fortunate that the present viscosity ratio concerned and achieved in this work is sufficient for our immediate application of many realistic problems found in biological flow, including simulating the motion of the deformable cell in a complex micro-channel geometry. For our further future work, we would like to extend our method to solve for problems with discontinuous density across the interface. As such, we have to work on decoupling the jump conditions and the density discontinuity.

Appendix A. The modified bilinear interpolation

Once we have obtained the augmented velocity field $\tilde{\mathbf{u}}$ on the grid, we need to compute the augmented velocity $\widetilde{\mathbf{U}}_k^-$ and $\widetilde{\mathbf{U}}_k^+$ at the control points, which are interpolated from the inside and outside of the interface, respectively. In this appendix, we present the modified bilinear interpolation formulas to compute the limits on the augmented velocity $\tilde{\mathbf{u}}$ from each side of the interface. The modified bilinear interpolation formulas need to include the appropriate correction terms to guarantee second order accuracy when the velocity is discontinuous. For example, suppose we want to interpolate the augmented velocity $\widetilde{\mathbf{U}}_k^-$ at the control points $\mathbf{X}_k$ from the inside of the interface using the augmented velocity $\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_4$ at the four neighboring grid points, $\mathbf{x}_1, \ldots, \mathbf{x}_4$, respectively, as illustrated in Fig. 7, then the interpolation scheme can be written as

$$\widetilde{\mathbf{U}}_k^+ = (1-p)(1-q)\tilde{\mathbf{u}}_1 + \mathbf{C}^{(1)} + p(1-q)\tilde{\mathbf{u}}_2 + \mathbf{C}^{(2)} + pq\tilde{\mathbf{u}}_3 + \mathbf{C}^{(3)} + (1-p)q\tilde{\mathbf{u}}_4\mathbf{C}^{(4)}, \tag{6.1}$$

where $\mathbf{C}^{(1)},\ldots,\mathbf{C}^{(4)}$ are correction terms, $p = \frac{X_k - x_1}{h}$, $q = \frac{Y_k - y_1}{h}$, and h is the grid size. The correction terms can be derived using Taylor series expansion and have the following forms:

$$\mathbf{C}^{(1)} = \begin{cases} 0, & \mathbf{x}_1 \in \Omega^+, \\ (1-p)(1-q)([\tilde{\mathbf{u}}] + h(-p[\tilde{\mathbf{u}}_x] - q[\tilde{\mathbf{u}}_y])), & \mathbf{x}_1 \in \Omega^-, \end{cases} \tag{6.2}$$

$$\mathbf{C}^{(2)} = \begin{cases} 0, & \mathbf{x}_2 \in \Omega^+, \\ p(1-q)([\tilde{\mathbf{u}}] + h((1-p)[\tilde{\mathbf{u}}_x] - q[\tilde{\mathbf{u}}_y])), & \mathbf{x}_2 \in \Omega^-, \end{cases} \tag{6.3}$$

$$\mathbf{C}^{(3)} = \begin{cases} 0, & \mathbf{x}_3 \in \Omega^+, \\ pq([\tilde{\mathbf{u}}] + h((1-p)[\tilde{\mathbf{u}}_x] + (1-q)[\tilde{\mathbf{u}}_y])), & \mathbf{x}_3 \in \Omega^-, \end{cases} \tag{6.4}$$

$$\mathbf{C}^{(4)} = \begin{cases} 0, & \mathbf{x}_4 \in \Omega^+, \\ (1-p)q([\tilde{\mathbf{u}}] + h(-p[\tilde{\mathbf{u}}_x] + (1-q)[\tilde{\mathbf{u}}_y])), & \mathbf{x}_4 \in \Omega^-, \end{cases} \tag{6.5}$$

Similarly, we can get a modified bilinear interpolation scheme for $\widetilde{\mathbf{U}}_k^-$ as follows:

$$\widetilde{\mathbf{U}}_k^- = (1-p)(1-q)\tilde{\mathbf{u}}_1 - \mathbf{C}^1 + p(1-q)\tilde{\mathbf{u}}_2 - \mathbf{C}^2 + pq\tilde{\mathbf{u}}_3 - \mathbf{C}^3 + (1-p)q\tilde{\mathbf{u}}_4 - \mathbf{C}^4. \tag{6.6}$$

where $\mathbf{C}^1,\ldots,\mathbf{C}^4$ are corrections and have the following forms:

$$\mathbf{C}^{1} = \begin{cases} (1-p)(1-q)([\tilde{\mathbf{u}}] + h(-p[\tilde{\mathbf{u}}_x] - q[\tilde{\mathbf{u}}_y])), & \mathbf{x}_1 \in \Omega^+, \\ 0, & \mathbf{x}_1 \in \Omega^-, \end{cases} \tag{6.7}$$

$$\mathbf{C}^{2} = \begin{cases} p(1-q)([\tilde{\mathbf{u}}] + h((1-p)[\tilde{\mathbf{u}}_x] - q[\tilde{\mathbf{u}}_y])), & \mathbf{x}_2 \in \Omega^+, \\ 0, & \mathbf{x}_2 \in \Omega^-, \end{cases} \tag{6.8}$$

$$\mathbf{C}^{3} = \begin{cases} pq([\tilde{\mathbf{u}}] + h((1-p)[\tilde{\mathbf{u}}_x] + (1-q)[\tilde{\mathbf{u}}_y])), & \mathbf{x}_3 \in \Omega^+, \\ 0, & \mathbf{x}_3 \in \Omega^-, \end{cases} \tag{6.9}$$

$$\mathbf{C}^{4} = \begin{cases} (1-p)q([\tilde{\mathbf{u}}] + h(-p[\tilde{\mathbf{u}}_x] + (1-q)[\tilde{\mathbf{u}}_y])), & \mathbf{x}_4 \in \Omega^+, \\ 0, & \mathbf{x}_4 \in \Omega^-. \end{cases} \tag{6.10}$$

References

[1] R.P. Beyer, A computational model of the cochlea using the immersed boundary method, J. Comput. Phys. 98 (1992) 145–162.

[2] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464–499.

[3] D. Calhoun, A Cartesian grid method for solving the two-dimensional streamfunction–vorticity equations in irregular regions, J. Comput. Phys. 176 (2002) 231–275.

[4] R. Dillon, L. Fauci, A. Fogelson, D. Gaver, Modeling biofilm processes using the immersed boundary method, J. Comput. Phys. 129 (1996) 57–73.

[5] C.D. Eggleton, A.S. Popel, Large deformation of red blood cell ghosts in a simple shear flow, Phys. Fluids 10 (1998) 1834–1845.

[6] L.J. Fauci, C.S. Peskin, A computational model of aquatic animal locomotion, J. Comput. Phys. (1988) 85–108.

[7] 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.

[8] A.L. Fogelson, Continuum models of platelet aggregation: formulation and mechanical properties, SIAM J. Appl. Math. 52 (1992) 1089–1110.

[9] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.

[10] M.-C. Lai, Z. Li, A remark on jump conditions for the three-dimensional Navier–Stokes equations involving an immersed moving membrane, Appl. Math. Lett. 14 (2001) 149–154.

[11] Z. Li, An overview of the immersed interface method and its applications, Taiwanese J. Math. 7 (2003) 1–49.

[12] Z. Li, K. Ito, The immersed interface method-numerical solutions of PDEs involving interfaces and irregular domains. SIAM Frontiers in Applied Mathematics 2006, 33, ISBN: 0-89971-609-8.

[13] A.T. Layton, An efficient numerical method for the two-fluid Stokes equations with a moving immersed boundary, Comput. Meth. Appl. Mech. Eng. 197 (2008) 2147–2155.

[14] Z. Li, K. Ito, M.-C. Lai, An augmented approach for Stokes equations with a discontinuous viscosity and singular forces, Comput. Fluids 36 (2007) 622–635.

[15] Z. Li, M.-C. Lai, The immersed interface method for the Navier–Stokes equations with singular forces, J. Comput. Phys. 171 (2001) 822–842.

[16] Z. Li, C. Wang, A fast finite difference method for solving Navier–Stokes equations on irregular domains, Commun. Math. Sci. 1 (2003) 180–196.

[17] Z. Li, S.R. Lubkin, X. Wan, An augmented IIM-level set method for Stokes equations with a discontinuous viscosity, Electron. J. Different. Equat. 15 (2007) 193–210.

[18] D.V. Le, An immersed interface method for solving viscous incompressible flows involving rigid and flexible boundaries, Ph.D thesis, Singapore-MIT Alliance, June 2005.

[19] D.V. Le, B.C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, J. Comput. Phys. 220 (2006) 109–138.

[20] L. Lee, R.J. LeVeque, An immersed interface method for incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 25 (2003) 832–856.

[21] R.J. LeVeque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (1997) 709–735.

[22] R.J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.

[23] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (2005) 157–192.

[24] C.S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.

[25] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.

[26] D. Russell, Z.J. Wang, A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, J. Comput. Phys. 191 (2003) 177–205.
[27] J.M. Stockie, S.I. Green, Simulating the motion of flexible pulp fibres using the immersed boundary method, J. Comput. Phys. 147 (1998) 147–165.
[28] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, third ed., Springer-Verlag, 2002.
[29] Z.-J. Tan, D.V. Le, K.M. Lim, B.C. Khoo, An Immersed Interface Method for the Incompressible Navier–Stokes Equations with Discontinuous Viscosity Across the Interface, SIAM J. Sci. Comput., submitted for publication.
[30] C. Tu, C.S. Peskin, Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods, SIAM J. Sci. Stat. Comput. 13 (1992) 1361–1376.
[31] A. Wiegmann, K.P. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (2000) 827–862.
[32] S. Xu, Z.J. Wang, Systematic derivation of jump conditions for the immersed interface method in three-dimensional flow simulation, SIAM J. Sci. Comput. 27 (2006) 1948–1980.
[33] S. Xu, Z.J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, J. Comput. Phys. 216 (2006) 454–493.
[34] S. Xu, Z.J. Wang, A 3D immersed interface method for fluid–solid interaction, Comput. Meth. Appl. Mech. Eng. 197 (2008) 2068–2086.