Computers and Fluids 168 (2018) 201–215

Contents lists available at ScienceDirect

Computers and Fluids

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

A coupled immersed boundary and immersed interface method for interfacial flows with soluble surfactant

Wei-Fan Hu [a,*], Ming-Chih Lai [b], Chaouqi Misbah [c]

[a] Department of Applied Mathematics, National Chung Hsing University, 145, Xingda Road, Taichung City 402, Taiwan
[b] Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, Taiwan
[c] Université Grenoble Alpes, LIPHY, Grenoble F-38000, France

ARTICLE INFO

Article history:
Received 30 October 2017
Revised 25 January 2018
Accepted 6 April 2018
Available online 9 April 2018

Keywords:
Immersed interface method
Moving irregular domain
Soluble surfactant
Immersed boundary method
Navier–Stokes equations

ABSTRACT

In this paper, we develop a coupled immersed boundary (IB) and immersed interface method (IIM) to simulate the interfacial flow problems with soluble surfactant. That is, a coupled system of surface-bulk convection-diffusion equations must be solved not only on a moving interface but also in an evolving irregular domain. Based on the immersed interface framework, we first begin with the numerical development for the diffusion equation in a fixed irregular domain and then extend the scheme to solve the convection-diffusion equation in an evolving domain. The fluid motion governed by the incompressible Navier–Stokes equations is solved by using traditional immersed boundary method while the bulk surfactant is solved by the proposed immersed interface method. A series of numerical tests for the present scheme have been conducted to illustrate the accuracy and applicability of the method. We first perform the accuracy and efficiency tests for the present IIM solver. We then check the convergence of the surface and bulk surfactant and the fluid variables in the interfacial flow problems. We further run a series of numerical simulations for a suspended droplet under shear flow with presence of soluble surfactant to study the effects of the dimensionless Biot number and the bulk Peclet number on the droplet deformation in details.

1. Introduction

Many problems in biological, physical and material sciences involve solving partial differential equations (PDEs) in complex domains or deformable interfaces. It is known that solving PDEs in complex domains or deformable interfaces numerically is quite challenging especially when the interface (or the interior boundary of domains) is moving. Even in the case of only surface material, developing numerical methods for convection-diffusion equations on an evolving interface is still of major interest in scientific computing community.

The immersed interface method is a sharp interface method and has been successfully applied to problems arising from a wide range of area. For instance, LeVeque and Li proposed the first IIM to solve elliptic interface problems with discontinuous coefficients [15]; further applications such as fluid-structure problems [11,28], AC dielectrophoresis [8], droplet electrohydrodynamics [9], and vesicle electrohydrodynamics [10,13], just to name a few. Various versions of IIM have also been implemented such as immersed finite element method [5], particle-mesh methods [18], coupling immersed interface and level set method [29], etc. See the review in the book [16].

In this paper, we develop a coupled immersed boundary and immersed interface method to study the interfacial flow problems with soluble surfactant. That is, the fluid system governed by the incompressible Navier–Stokes equations is solved by using traditional IB method while the convection-diffusion equation for the bulk surfactant is solved by a sharp immersed interface method. Surfactant molecules typically consist of a hydrophilic head and a hydrophobic tail may adsorb and desorb between bulk fluids and the interface so that the interfacial tension can be reduced. Meanwhile, this non-uniform distribution of surfactant molecules along the interface produces an extra force (Marangoni force) in the tangential direction to affect the dynamics. The effect due to insoluble surfactant has been extensively investigated since past two decades. For the insoluble surfactant case we only need to solve the surface concentration equation on an evolving surface, while for the soluble case we have to consider a coupled system of surface-bulk equations in complex domains or deformable interfaces with the interaction through adsorption and desorption pro-

* Corresponding author.
E-mail addresses: wfhu@nchu.edu.tw (W.-F. Hu), mclai@math.nctu.edu.tw (M.-C. Lai), chaouqi.misbah@univ-grenoble-alpes.fr (C. Misbah).

https://doi.org/10.1016/j.compfluid.2018.04.013
0045-7930/

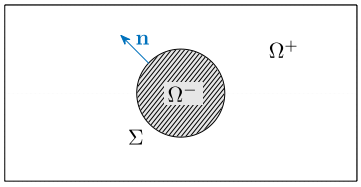

Fig. 1. The computational domain Ω consists of the physical domain Ω^+ and the extended domain Ω^- with the embedded interface Σ. The unit outward normal vector on Σ is defined by $\mathbf{n}$.

cesses. Therefore, tackling such problem via numerical methods is indeed essential and can be a quite challenging issue. The numerical simulation by using boundary integral method was proposed by Eggleton and Stebe [4], they assumed a simplified model that the bulk diffusivity dominates the bulk surfactant transportation (i.e., the bulk Peclet number is assumed to be zero) so that the effect of the constant bulk surfactant only appears in the source term of the surface surfactant equation. Booty and Siegel [2] investigated the interfacial fluid flow at large values of Peclet number via a numerical method that incorporates a singular perturbation analysis into a full numerical solution, whereas the methods proposed in [2,4] are limited to the flow in Stokes regime. For the Navier–Stokes flow, depending on the interface representation, Muradoglu and Tryggvason introduced a front-tracking method to study the interfacial flows with soluble surfactant in axisymmetric [20] and 3D multiphase flows [21]. Teigen et al. [26] proposed a diffuse-interface method for solving two-phase flows with soluble surfactant in both 2D and 3D spaces. Chen and Lai [3] developed a conservative scheme for the coupled system of surface-bulk concentration equation. Their extension work on the droplet bouncing and coalescence with soluble surfactant can be found in [23]. Recently, Xu et al. [30] proposed a level set method for two-phase flows with soluble surfactant. However, most of the aforementioned references employ the numerical methods using a smoothing technique so that the bulk concentration is regularized near the vicinity of interfaces which fail to capture the solution discontinuity across interfaces. Until recently, Khatri and Tornberg [12] used a two-dimensional segment projection method to represent the interface which is capable of capturing the jump discontinuity of the bulk concentration solution.

In this paper, we take advantage of IIM in which a uniform Cartesian grid is layout in a regular computational domain so that the spatial discretizations using finite difference for derivative operators can be performed in a straightforward manner except for the grid points near an interface. Firstly, we propose a simple and efficient method to solve the diffusion equation in an irregular domain (see the setup in Fig. 1). By extending the irregular domain to a regular one, the boundary condition of the solution on the interface is alternatively regarded as the jump condition which is incorporated in the immersed interface discretization. Although the idea of using IIM to solve the elliptic type of PDEs in irregular domains is not new as already described in [16], the treatment of jump conditions in the present scheme is different from the one in [16] where the augmented approach is adopted. We also provide both implicit backward Euler and Crank-Nicolson scheme and then check the convergence of the method. Secondly, we extend the scheme for solving the convection-diffusion equation in an evolving domain. A simple splitting method is adopted in which we first update the intermediate solution by a convection step and then solve the diffusion equation. The numerical discretization near the interface of the convection part can be tricky and here we propose a simple explicit method which follows the idea of method of characteristics to track the discrete solution accurately via the usage of the jump information. Again a convergence study for the solver is performed. Lastly, we apply the immersed interface solver to solve the coupled surface-bulk concentration equation with Navier–Stokes flow. The fluid system is solved by traditional immersed boundary method. This coupled solver shares the same spirit with our previous works for the simulations of droplet electrohydrodynamics [9] and vesicle electrohydrodynamics [10].

The paper is organized as follows. An efficient immersed interface method for solving the diffusion equation in an irregular domain is developed in Section 2. The extension of the numerical scheme for solving the convection-diffusion equation in a moving irregular domain is presented in Section 3. The new numerical algorithm that couples immersed boundary and immersed interface method for solving the interfacial problems with presence of soluble surfactant is outlined in Section 4. A series of numerical simulations to investigate the study of how the Biot number and bulk Peclet number affect the droplet dynamics in shear flow are presented. Some concluding remarks and future works are given in Section 5.

2. A simple version of immersed interface method for the diffusion equation in an irregular domain

In this section, we propose a numerical method for solving the diffusion equation in an irregular domain based on the framework of the immersed interface method. Consider a domain Ω in $\mathbb{R}^2$ and a simple closed interface Σ immersed in Ω which separates Ω into two domains; namely, Ω^+ (exterior to the interface) and Ω^- (interior to the interface) so that $\Omega = \Omega^+ \cup \Omega^-$, see the illustration of these domains in Fig. 1. Throughout this paper, we focus on solving equations in the exterior domain Ω^+ although our method can be applied to solve the equations in the interior domain Ω^- without any difficulty. The interface Σ is represented by a Lagrangian parametric form of $\mathbf{X}(s,t) = (X(s,t), Y(s,t))$, $0 \le s \le 2\pi$, where s is the parameter of the initial configuration of the interface. Here, we assume that the interface Σ is not evolving, and thus the domain Ω^+ (or Ω^-) is fixed.

Let us consider the following diffusion equation in Ω^+ as

$$\frac{\partial \phi}{\partial t} = \frac{1}{Pe}\Delta\phi \quad \text{in} \quad \Omega^+, \quad \phi|_{\partial\Omega} = \phi_b, \tag{1}$$

where Pe is the dimensionless Peclet number. Of course, the above equation should be accompanied with an initial condition $\phi(\mathbf{x}, 0)$ and one inner boundary condition on Σ; namely the Dirichlet ($\phi = \phi_D$), or Neumann ($\frac{\partial \phi}{\partial \mathbf{n}} = \phi_N$), or Robin type ($\frac{\partial \phi}{\partial \mathbf{n}} = \alpha\phi + g, \alpha > 0$), where $\mathbf{n}$ is the unit outward normal vector pointing from Ω^- to Ω^+ along the interface Σ. It is important to mention that, the type of boundary condition on Σ just results in a slight difference of the present numerical scheme as we can see from the numerical implementation later in next subsection.

The present immersed interface method for solving Eq. (1) is to set the solution being identical zero in the interior domain Ω^-, and replace the inner boundary condition on Σ by the jump condition so that the equation in irregular domain Ω^+ becomes an equation in the whole regular domain Ω as

$$\frac{\partial \phi}{\partial t} = \frac{1}{Pe}\Delta\phi \quad \text{in} \quad \Omega, \quad \phi|_{\partial\Omega} = \phi_b, \tag{2}$$

$$[\![\phi]\!] = \phi^+(s,t), \quad [\![\phi_n]\!] = \phi_n^+(s,t) \quad \text{on} \quad \Sigma, \tag{3}$$

where $\phi_n = \frac{\partial \phi}{\partial \mathbf{n}}$ stands for the normal derivative and the bracket $[\![\cdot]\!]$ denotes the jump discontinuity across the interface Σ of the quantity approaching from the Ω^+ side to the Ω^- side. One should notice that, either one of the above jump conditions is given for imposing Dirichlet or Neumann boundary condition on Σ; whereas for Robin boundary condition, both jumps are unknown but related

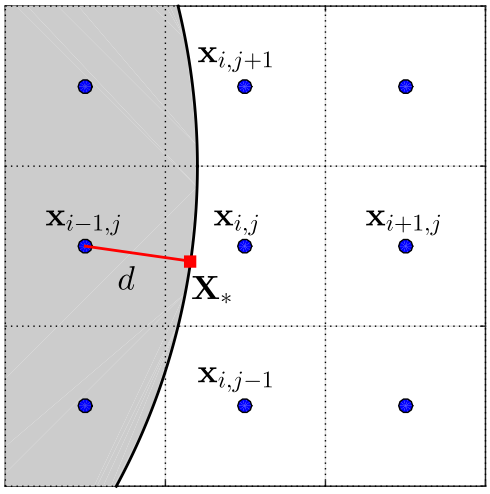

Fig. 2. The five-point Laplacian of the irregular point $\mathbf{x}_{i,j}$.

to each other. These two jump conditions will be incorporated in the numerical discretization for spatial derivatives. The remaining issue is how to solve the above diffusion Eq. (2) with the jump discontinuities (3) in Ω which we address as follows.

2.1. An immersed interface framework

To proceed, let us first layout a uniform Cartesian grid in the computational domain Ω with mesh width $h = \Delta x = \Delta y$ for simplicity. The grid point $\mathbf{x}_{i,j} = (x_i, y_j)$ is defined at the grid center where the discrete solutions $\phi_{i,j} = \phi(\mathbf{x}_{i,j})$ are located (here we temporarily omit the time variable). The interface is embedded and cuts through some grid cells so the solution is not smooth across the interface as we can see from the jump conditions in Eq. (3). We then classify the grid point as either a regular or irregular point. For a regular point, it means that the standard five-point Laplacian discretization (denoted by Δ_h) does not cut through the interface so the second-order local truncation error is achieved. On the other hand, at an irregular point, the five-point Laplacian cuts through the interface so the grid points used involve both inside and outside the interface. Since the solution and its derivatives have jumps across the interface, a correction for the Laplacian discretization is thus needed at the irregular point to maintain the desired accuracy. Thus, the discretization of Laplace operator at the grid point $\mathbf{x}_{i,j}$ can be generally written in the form of

$$\Delta_h \phi_{i,j} + \frac{C_{i,j}}{h^2}, \tag{4}$$

where $C_{i,j}$ is the correction term which is nonzero only if the grid point is irregular.

Let us describe what the correction term is at the particular irregular point as depicted in Fig. 2. When we apply the five-point Laplacian to $\mathbf{x}_{i,j}$, the grid point $\mathbf{x}_{i-1,j}$ falls in different side of the interface so the correction of discretization comes only from the point $\mathbf{x}_{i-1,j}$. To derive the correction term, one needs to find the orthogonal projection of $\mathbf{x}_{i-1,j}$ at the interface (say $\mathbf{X}_*$ in Fig. 2), and then apply the Taylor's expansion along the normal direction at $\mathbf{X}_*$. The correction term thus becomes

$$C_{i,j} = [\![\phi]\!] + d[\![\phi_n]\!] + \frac{d^2}{2}\left([\![\Delta\phi]\!] - \kappa[\![\phi_n]\!] - \Delta_s[\![\phi]\!]\right)\Big|_{\mathbf{X}_*}, \tag{5}$$

where the value d is the signed distance between the grid point $\mathbf{x}_{i-1,j}$ and the orthogonal projection $\mathbf{X}_*$, κ is the local curvature of the interface, and Δ_s is the surface Laplace (or Laplace-Beltrami) operator defined by

$$\Delta_s = \frac{1}{|\mathbf{X}_s|}\frac{\partial}{\partial s}\left(\frac{1}{|\mathbf{X}_s|}\frac{\partial}{\partial s}\right). \tag{6}$$

Here, the subscript in $\mathbf{X}_s$ denotes the partial derivative of $\mathbf{X}$ with respect to s. It can be seen that the five-point Laplacian discretization in Eq. (4) gives the second-order accuracy at regular points but only first-order accuracy at irregular ones. We leave the computation of the interfacial derivatives such as local curvature and the surface Laplace operator to the next subsection. The more detailed derivation of the correction term in Eq. (5) can be found in [7,24].

2.2. Implementation details

Since we assume that the interface is closed, we can use the Fourier spectral discretization to represent the configuration $\mathbf{X}(s)$. We first choose a collection of discrete Lagrangian markers $\mathbf{X}_k = \mathbf{X}(s_k)$ with equally distributed interfacial coordinates $s_k = k\Delta s$, $k = 0, 1, \cdots, M$ with $s = 2\pi/M$. The interface thus can be represented in truncated Fourier series (in vector form) as

$$\mathbf{X}(s) = \sum_{\ell=-M/2}^{M/2-1} \widehat{\mathbf{X}_\ell}\, e^{i\ell s}, \tag{7}$$

where $\widehat{\mathbf{X}_\ell}$ are the Fourier coefficients for $\mathbf{X}(s)$, and can be computed very efficiently using the Fast Fourier Transform (FFT). Under this Fourier representation, the derivatives with respect to s can also be computed quite easily by using the pseudospectral method [25]. In this fashion, all geometrical quantities such as curvature κ and the surface Laplace operator Δ_s in Eq. (6) can be computed with spectral accuracy (in addition, the unit tangent vector can be computed by $\boldsymbol{\tau} = (\tau_1, \tau_2) = \mathbf{X}_s/|\mathbf{X}_s|$ and the normal vector is $\mathbf{n} = (\tau_2, -\tau_1)$). Since all the interfacial quantities are defined in Lagrangian manner so the jumps of $[\![\phi]\!]$, $[\![\phi_n]\!]$, $[\![\Delta\phi]\!]$, and $\Delta_s[\![\phi]\!]$ are all defined at $\mathbf{X}_k$. Thanks to the Fourier representation, those jumps and the curvature at the orthogonal projection points $\mathbf{X}_*$ in the correction term in Eq. (5) can be easily interpolated through the values at the Lagrangian markers $\mathbf{X}_k$.

Now we are ready to discretize the difference equation with the jump conditions in Eqs. (2) and (3) based on the aforementioned immersed interface framework. Here we use the Backward Euler (BE) and Crank–Nicolson (CN) method for the time integration. Denoting Φ, Ψ_0, Ψ_1, and Ψ_2 as the solution vectors formed by $\phi_{i,j}$, $[\![\Delta\phi]\!]_{\mathbf{X}_k}$, $[\![\phi]\!]_{\mathbf{X}_k}$, and $[\![\phi_n]\!]_{\mathbf{X}_k}$, respectively, the Backward Euler discretization for Eq. (2) is given by

$$\frac{\Phi^{n+1} - \Phi^n}{\Delta t} = \frac{1}{Pe}\left(\Delta_h \Phi^{n+1} + C_0\Psi_0^{n+1} + C_1\Psi_1^{n+1} + C_2\Psi_2^{n+1}\right), \tag{8}$$

where the superscript n stands for the time step $t_n = n\Delta t$ with the time step size Δt; C_0, C_1 and C_2 are the formal matrix operators involving the Fourier interpolation of those jumps in the correction term as described previously. Let us describe how to solve the above matrix equation in the following. Suppose that the Neumann boundary condition is imposed on Σ, it yields that Ψ_2^{n+1} is given but Ψ_1^{n+1} is treated as an unknown to be determined. In addition, unlike our previous works on solving Laplace equation for the electric potential ($[\![\Delta\phi]\!] = 0$) for the droplet and vesicle electrohydrodynamics in [9,10], the term $\Psi_0^{n+1} (= [\![\Delta\phi^{n+1}]\!]_{\mathbf{X}_k})$ is also an unknown here. We can approximate this term using the original differential equation by applying on the both sides of interface with Backward Euler discretization to obtain $(\Psi_1^{n+1} - \Psi_1^n)/\Delta t = \Psi_0^{n+1}/Pe$. It is interesting to see that, although the approximation for Ψ_0^{n+1} gives only first-order accuracy in time, the local truncation error for the five-point Laplacian operator remains first-order accuracy in space at irregular points as long as we take $\Delta t = O(h)$

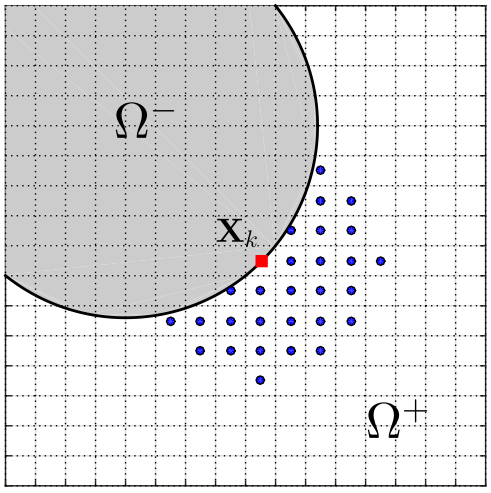

Fig. 3. A diagram showing least squares approximation for one-sided interpolation or normal derivative at $\mathbf{X}_k$.

(one can see from Eqs. (4) to (5)). Proceeding to rearrange the discretization in Eq. (8) gives the matrix equation

$$\left(\Delta_h - \frac{Pe}{\Delta t}I\right)\Phi^{n+1} + \left(\frac{Pe}{\Delta t}C_0 + C_1\right)\Psi_1^{n+1} = -\frac{Pe}{\Delta t}\Phi^n + \frac{Pe}{\Delta t}C_0\Psi_1^n - C_2\Psi_2^{n+1}. \tag{9}$$

On the other hand, for the Crank-Nicolson discretization of Eq. (2), we have $(\Psi_1^{n+1} - \Psi_1^n)/\Delta t = (\Psi_0^n + \Psi_0^{n+1})/2Pe$. Thus, the resultant matrix equation becomes

$$\left(\Delta_h - \frac{2Pe}{\Delta t}I\right)\Phi^{n+1} + \left(\frac{2Pe}{\Delta t}C_0 + C_1\right)\Psi_1^{n+1} = -\frac{2Pe}{\Delta t}\Phi^n - \Delta_h\Phi^n + C_0\left(\frac{2Pe}{\Delta t}\Psi_1^n + \Psi_0^n\right) - C_2\Psi_2^{n+1}. \tag{10}$$

One should notice that, the computation of $\Delta_h\Phi^n$ at irregular points on the righthand side of Eq. (10) indeed involves the correction term at previous step but we use the same notation for simplicity. At this stage, it becomes clear that we have to add one more matrix equation to approximate the augmented variable Ψ_1^{n+1}. However, as we can see from Eq. (3), a one-sided approximation for $\phi^+(\mathbf{X}_k, t_{n+1})$ is needed. To approximate that, we first construct a least squares cubic polynomial by using the values of $\phi_{i,j}^{n+1}$ at the grid points satisfying $\|\mathbf{x}_{i,j} - \mathbf{X}_k\| \le 4h$ in the domain Ω^+ as depicted in Fig. 3. In general, the number of chosen grid points is larger than the number of unknown coefficients in the least squares approximation so the cubic polynomial $P(x, y)$ can be obtained. For the interface point $\mathbf{X}_k$ in a high curvature region, we have to increase the spatial resolution to make sure that enough grid points fall into the same region so the least squares cubic polynomial is achievable. Once we have the polynomial $P(x, y)$, the one-sided interpolation $\phi^+(\mathbf{X}_k, t_{n+1})$ can be computed directly. This least squares cubic polynomial approach roughly has third-order accuracy to the approximation of the function itself. With this approximation, we can write the jump condition $[\![\phi]\!] = \phi^+$ in Eq. (3) in a matrix form by

$$B\Phi^{n+1} = \Psi_1^{n+1} \tag{11}$$

where B denotes the formal matrix arising from the above least squares cubic polynomial approximation.

Now, we couple either the matrix Eq. (9) (BE) or Eq. (10) (CN) with Eq. (11) into one linear system as

$$\begin{bmatrix} H & C \\ B & -I \end{bmatrix}\begin{bmatrix} \Phi^{n+1} \\ \Psi_1^{n+1} \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}. \tag{12}$$

In practice, we do not form the matrices C and B explicitly as we can see from the iterative procedure of our matrix solver below. To solve the above linear system, we first eliminate Φ^{n+1} from Eq. (12) by Schur complement technique to obtain a new linear system for Ψ_1^{n+1} as

$$\left(BH^{-1}C + I\right)\Psi_1^{n+1} = BH^{-1}F.$$

This is a $M \times M$ system for Ψ_1^{n+1} which is much smaller than the original one in Eq. (12). We then use the GMRES iterative method to solve the above linear system. Since the GMRES method only requires the matrix-vector multiplication, it is not necessary to construct the matrices H^{-1}, C, and B explicitly. Note that, the matrix H comes from the five-point Laplacian discretization so the inversion of H can be performed efficiently by applying the fast Poisson solver provided by Fishpack public software package [1].

In summary, the detailed numerical algorithm for solving the linear system Eq. (12) to find Φ^{n+1} and Ψ_1^{n+1} can be split into the following three steps.

Step 1. Apply one fast Poisson solver to obtain Φ^* in

$$H\Phi^* = F.$$

Step 2. Use GMRES iteration to solve Ψ_1^{n+1} in

$$\left(BH^{-1}C + I\right)\Psi_1^{n+1} = B\Phi^*.$$

Step 3. Apply one fast Poisson solver to solve Φ^{n+1} in

$$H\Phi^{n+1} = F - C\Psi_1^{n+1}.$$

During the GMRES iteration in Step 2, we set the stopping criterion as h^2. The overall computational cost for Step 1-3 in our present scheme can be evaluated in terms of the number of fast Poisson solver being applied. In the next subsection, we shall provide a more systematic numerical check on the accuracy and efficiency for our present numerical algorithm.

We should remark that, for the case of imposing Dirichlet or Robin boundary condition on the interface Σ, there is a slight difference of the BE and CN discretizations from Eqs. (9) to (10). For the Robin boundary condition case, one still needs to apply the one-sided interpolation for $\phi^+(\mathbf{X}_k)$, whereas for the Dirichlet case the approximation for $\phi_n^+(\mathbf{X}_k)$ is required and it can be simply done by taking normal derivative of the least squares polynomial $\nabla P(\mathbf{X}_k)\cdot\mathbf{n}(\mathbf{X}_k)$ (thus the derivative approximately has second-order accuracy). The resultant linear system from other types of boundary conditions on Σ can be represented and solved similarly as in Eq. (12).

2.3. Numerical accuracy and efficiency study

In this subsection, we demonstrate the accuracy and efficiency test of the present IIM scheme for solving the diffusion equation developed in previous subsection. We construct an analytical solution for Eq. (1) with a three-leaved interface as $\mathbf{X}(s) = (r(s)\cos s, r(s)\sin s)$ with the radius function $r(s) = 0.1(4 + \cos 3s)$ as

$$\phi(x, y, t) = 1 + 0.5\exp\left(\frac{-2\pi^2 t}{Pe}\right)\cos(\pi x)\cos(\pi y), \quad \mathbf{x} \in \Omega^+. \tag{13}$$

Here, we test three cases by imposing different inner boundary conditions as Dirichlet, Neumann, and Robin type boundary condition on Σ. The computational domain is set by $\Omega = [-1, 1] \times [-1, 1]$ and the Peclet number is chosen as $Pe = 10$. We set N to be the grid size and thus the mesh size is $h = 2/N$. The time step for the BE and CN discretization of Eq. (9) or (10) is chosen as the same of mesh width as $\Delta t = h$ and the simulation is run up to

Table 1
Mesh refinement results for accuracy and efficiency for the BE (upper half-panel) and CN (lower half-panel) discretization. The numerical solution is denoted by ϕ_h and exact solution by ϕ_e.

N	Dirichlet			Neumann			Robin		
	$\|\phi_h - \phi_e\|_\infty$	Rate	Iter.	$\|\phi_h - \phi_e\|_\infty$	Rate	Iter.	$\|\phi_h - \phi_e\|_\infty$	Rate	Iter.
128	7.627E − 04	–	3.25	9.051E − 04	–	2.10	7.935E − 04	–	2.04
256	3.789E − 04	1.01	3.42	4.489E − 04	1.01	2.10	3.902E − 04	1.02	2.02
512	1.889E − 04	1.00	3.50	2.233E − 04	1.01	2.09	1.935E − 04	1.01	2.01
1024	9.423E − 05	1.00	3.75	1.113E − 04	1.00	2.23	9.633E − 05	1.01	2.19
128	7.121E − 05	–	3.10	3.825E − 05	–	2.17	2.369E − 05	–	2.04
256	1.720E − 05	2.05	3.47	5.571E − 06	2.78	2.17	5.958E − 06	1.99	2.02
512	4.648E − 06	1.89	3.16	1.395E − 06	2.00	2.18	1.490E − 06	2.00	2.01
1024	1.145E − 06	2.02	3.43	3.460E − 07	2.01	2.24	3.722E − 07	2.00	2.06

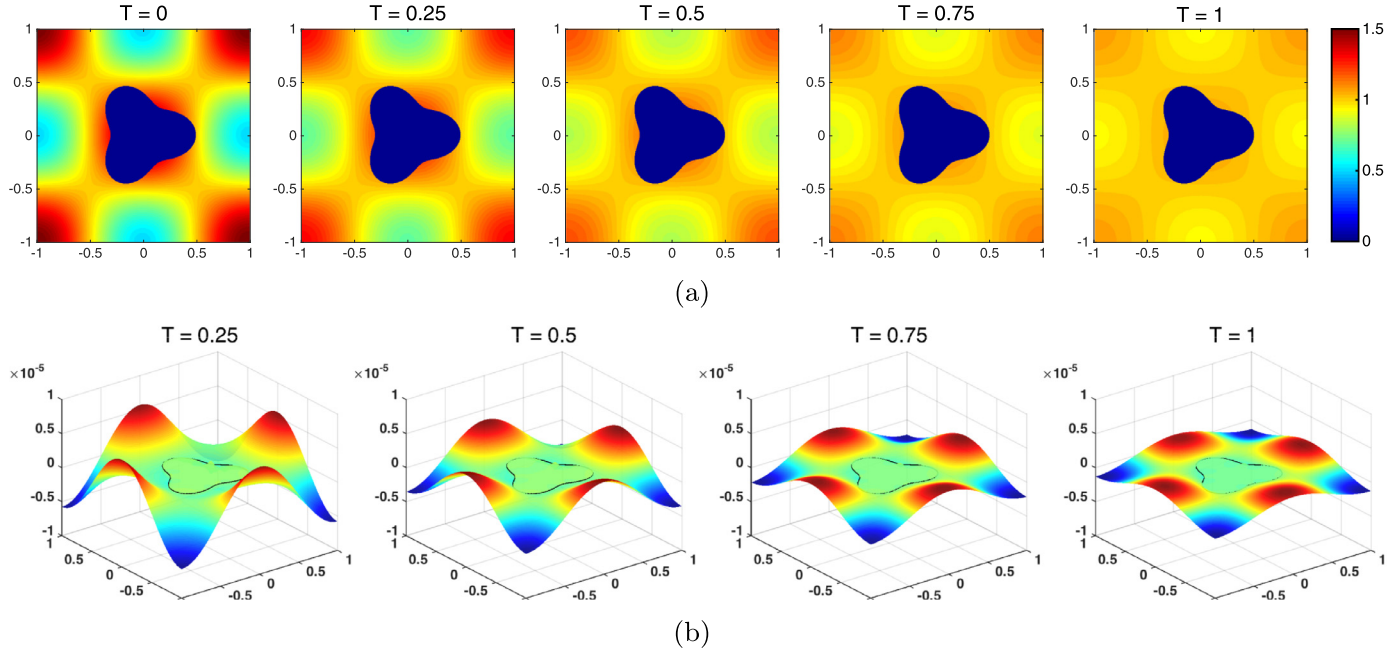

Fig. 4. (a) The snapshots of the numerical solution at $T = 0$, 0.25, 0.5, 0.75 and 1. The numerical solution is subject to Dirichlet boundary condition on the outer boundary $\partial\Omega$ and Robin boundary condition on the inner boundary Σ. (b) The snapshots of the numerical error at $T = 0.25$, 0.5, 0.75 and 1.

the terminal time $T = 1$. We present the rate of convergence for the maximum error between the numerical solution ϕ_h and the exact solution ϕ_e in Table 1. As expected, the first- and second-order convergence are achieved for BE and CN scheme, respectively. As for the computational complexity, one can see that the average number of GMRES iterations in the time interval [0,1] is just less than 4 and becomes steady even as the grid number doubles. Moreover, we plot the snapshots of numerical solution (obtained by CN discretization with Robin boundary condition on Σ with $N = 512$) and snapshots of their errors $\phi_e - \phi_h$ at different times in Fig. 4. One can see that the present method captures the jump discontinuity across the interface accurately and the largest numerical error mainly results from the computational domain Ω^+ rather than the extended domain Ω^-. In fact, for each case we tested here, the numerical errors at grid points in the extended domain Ω^- are consistent with the spatial discretization of magnitude $O(h^2)$.

3. An immersed interface method for the convection-diffusion equation in a moving irregular domain

With successful development of IIM for diffusion equation in a fixed irregular domain, in this section, we extend our method to the following convection-diffusion equation in a *moving* irregular domain $\Omega^+(t)$ as

$$\frac{\partial \phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = \frac{1}{Pe}\Delta\phi \quad \text{in} \quad \Omega^+, \quad \phi|_{\partial\Omega} = \phi_b, \tag{14}$$

where $\mathbf{u}$ is a given velocity or is obtained from solving Navier–Stokes equations (see in next section). As before, besides of the boundary condition on $\partial\Omega$, one of inner boundary conditions on $\Sigma(t)$ (Dirichlet ($\phi = \phi_D$), Neumann ($\frac{\partial\phi}{\partial\mathbf{n}} = \phi_N$), or Robin type ($\frac{\partial\phi}{\partial\mathbf{n}} = \alpha\phi + g, \alpha > 0$)) must be given. Again, assuming the zero solution in the interior domain $\Omega^-(t)$ and regarding the inner boundary condition as jump conditions across the interface $\Sigma(t)$, we can rewrite the convection-diffusion Eq. (14) in the fixed regular domain $\Omega = \Omega^+(t) \cup \Omega^-(t)$ (referring the setup of those domains in Fig. 1) as

$$\frac{\partial \phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = \frac{1}{Pe}\Delta\phi \quad \text{in} \quad \Omega, \quad \phi|_{\partial\Omega} = \phi_b, \tag{15}$$

$$[\![\phi]\!] = \phi^+(s,t), \quad [\![\phi_n]\!] = \phi_n^+(s,t) \quad \text{on} \quad \Sigma(t). \tag{16}$$

Certainly, Eq. (15) is subject to an initial condition in $\Omega^+(0)$. We remind the readers that the above jump conditions are required for the calculation of the correction terms whereas either one or both of them can be unknown (depending on the type of the boundary condition on Σ).

With presence of the applied velocity filed $\mathbf{u}$, the interface Σ is assumed to move with the local velocity by

$$\frac{\partial \mathbf{X}}{\partial t}(s,t) = \mathbf{u}(\mathbf{X}(s,t),t). \tag{17}$$

Unlike in previous section where the irregular domain Ω^+ is fixed, here due to the fact that the interface $\Sigma(t)$ is moving according to Eq. (17) (thus $\Omega^+(t)$ and $\Omega^-(t)$ are both evolving, but Ω remains as a fixed domain), we have to carefully track the discrete solutions in the neighborhood of the interface. Therefore, instead of discretizing Eqs. (15) and (16) directly, here, we propose a simple operator splitting scheme. That is, we first solve the intermediate solution for the convection step, and then update the solution by solving the diffusion part. The full numerical algorithm is described in the following subsections.

3.1. An explicit immersed interface method for the convection equation

In this subsection, we focus on developing a numerical scheme based on IIM for solving the intermediate solution of the convection equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = 0 \qquad \text{in} \quad \Omega. \tag{18}$$

Since the interface $\Sigma(t)$ moves with the velocity field $\mathbf{u}$, capturing the jump discontinuity behavior of the discrete solutions in the vicinity of $\Sigma(t)$ becomes the major task. Eq. (18) is discretized by the second-order Lax–Wendroff scheme (9-point stencil) [14] as

$$\begin{aligned}
\phi_{i,j}^{n+1} &= \phi_{i,j}^{n} \\
&+ \frac{1}{2}\alpha_{i,j}^{n+1/2}(1+\alpha_{i-1/2,j}^{n})\phi_{i-1,j}^{n} - \frac{1}{2}\alpha_{i,j}^{n+1/2}(\alpha_{i-1/2,j}^{n}+\alpha_{i+1/2,j}^{n})\phi_{i,j}^{n} \\
&+ \frac{1}{2}\alpha_{i,j}^{n+1/2}(-1+\alpha_{i+1/2,j}^{n})\phi_{i+1,j}^{n} + \frac{1}{2}\beta_{i,j}^{n+1/2}(1+\beta_{i,j-1/2}^{n})\phi_{i,j-1}^{n} \\
&- \frac{1}{2}\beta_{i,j}^{n+1/2}(\beta_{i,j-1/2}^{n}+\beta_{i,j+1/2}^{n})\phi_{i,j}^{n} + \frac{1}{2}\beta_{i,j}^{n+1/2}(-1+\beta_{i,j+1/2}^{n})\phi_{i,j+1}^{n} \\
&+ \frac{1}{8}\alpha_{i,j}^{n+1/2}\left[(\beta_{i+1/2,j}^{n}-\beta_{i-1/2,j}^{n})\phi_{i,j+1}^{n} - (\beta_{i+1/2,j}^{n}-\beta_{i-1/2,j}^{n})\phi_{i,j-1}^{n}\right] \\
&+ \frac{1}{8}\beta_{i,j}^{n+1/2}\left[(\alpha_{i,j+1/2}^{n}-\alpha_{i,j-1/2}^{n})\phi_{i+1,j}^{n} - (\alpha_{i,j+1/2}^{n}-\alpha_{i,j-1/2}^{n})\phi_{i-1,j}^{n}\right] \\
&+ \frac{1}{8}\left(\alpha_{i,j}^{n+1/2}\beta_{i+1/2,j}^{n} + \alpha_{i,j+1/2}^{n}\beta_{i,j}^{n+1/2}\right)\phi_{i+1,j+1}^{n} \\
&- \frac{1}{8}\left(\alpha_{i,j}^{n+1/2}\beta_{i+1/2,j}^{n} + \alpha_{i,j-1/2}^{n}\beta_{i,j}^{n+1/2}\right)\phi_{i+1,j-1}^{n} \\
&- \frac{1}{8}\left(\alpha_{i,j}^{n+1/2}\beta_{i-1/2,j}^{n} + \alpha_{i,j+1/2}^{n}\beta_{i,j}^{n+1/2}\right)\phi_{i-1,j+1}^{n} \\
&+ \frac{1}{8}\left(\alpha_{i,j}^{n+1/2}\beta_{i-1/2,j}^{n} + \alpha_{i,j-1/2}^{n}\beta_{i,j}^{n+1/2}\right)\phi_{i-1,j-1}^{n},
\end{aligned} \tag{19}$$

where $\alpha_{i,j}^{n} = u_{i,j}^{n}\Delta t/\Delta x$ and $\beta_{i,j}^{n} = v_{i,j}^{n}\Delta t/\Delta y$. We assume that all the coefficients on the righthand side of the above discretization are given for a prescribed velocity field. Notice that the velocity component u and v appearing in those coefficients are not only evaluated at the cell center $\mathbf{x}_{i,j}$ but also the cell-normal edges $\mathbf{x}_{i\pm1/2,\,j\pm1/2}$.

As before, the grid point is identified as either a regular or irregular point. For a regular point, we mean that all the grid points used on the righthand side of Eq. (19) falls into the same side of the interface; whereas an irregular point involves using the grid points from both inside and outside the interface. It is clear that we need to modify Eq. (19) at irregular points to achieve the desired accuracy.

We illustrate how to find the correction term via the two cases shown in Fig. 5. As a first case depicted in Fig. 5(a), the interface moves form $\Sigma(t_n)$ to $\Sigma(t_{n+1})$. For a certain small time step Δt, Eq. (19) can be regarded as an interpolation for some grid value lying in the region $[x_{i-1},x_{i+1}]\times[y_{j-1},y_{j+1}]$ at the previous time step t_n, and this grid point flows along some characteristic line to $\mathbf{x}_{i,j}$ at time step t_{n+1}. This is exactly the concept of method of characteristics. Since the grid point $\mathbf{x}_{i,j}$ stays inside the interface, one needs to correct the grid values which come from the outside of the interface ($\Omega^+(t_n)$). For instance,

$$\phi_{i+1,j-1}^{n} \overset{\text{correct}}{=\!=} \phi_{i+1,j-1}^{n} + C_{i+1,j-1}^{n},$$

in which the above correction term follows the same computation as in Eq. (5). Of course, one also has to correct other grid values $\phi_{i+1,j}^{n}$, $\phi_{i,j+1}^{n}$, and $\phi_{i+1,j+1}^{n}$ as depicted in Fig. 5(a). For the second case shown in Fig. 5(b), as the interface moves, the grid point $\mathbf{x}_{i,j}$ falls into the different sides of the interface (from $\Omega^+(t_n)$ to $\Omega^-(t_{n+1})$). In this case, the grid value $\phi_{i,j}^{n+1}$ comes from some grid value lying inside the interface at time step t_n. Therefore, one should correct those grid values which originating from $\Omega^+(t_n)$ side (say $\phi_{i+1,j-1}^{n}$, $\phi_{i,j}^{n}$, $\phi_{i+1,j}^{n}$, $\phi_{i-1,j+1}^{n}$, $\phi_{i,j+1}^{n}$, and $\phi_{i+1,j+1}^{n}$ in Fig. 5(b)).

From these two cases, we can conclude a rule for the calculation of correction terms: if the grid point $\mathbf{x}_{i,j}$ remains on the same side of the interface when the interface is moving, one needs to correct the grid value(s) which is on the different side of $\mathbf{x}_{i,j}$ at the previous time step t_n. On the other hand, if the grid point $\mathbf{x}_{i,j}$ crosses the interface at different time steps, one have to correct the grid value(s) which lies in the different side of $\mathbf{x}_{i,j}$ at t_{n+1}. As a consequence, the Lax–Wendroff scheme in Eq. (19) can be generally modified by a matrix form of

$$\Phi^{n+1} = A\Phi^{n} + \widetilde{C}_0\Psi_0^{n} + \widetilde{C}_1\Psi_1^{n} + \widetilde{C}_2\Psi_2^{n},$$

where A denotes the coefficient matrix constructed by the right-hand side of Eq. (19) and $\widetilde{C}_0\Psi_0^{n} + \widetilde{C}_1\Psi_1^{n} + \widetilde{C}_2\Psi_2^{n}$ are the correction terms. One should notice that the correction matrix $\widetilde{C}_0$, $\widetilde{C}_1$, and $\widetilde{C}_2$ are different from those matrices C_0, C_1, and C_2 in previous section since they contain correction terms multiplying those coefficients in Eq. (19). As one can see from the above equation, the solution is explicitly updated by the solution at previous time step, which can be done very efficiently.

3.2. Full numerical algorithm

Once we know how to update the intermediate solution for the convection part, now we are ready to solve Eq. (15) with Eq. (16) via the splitting technique as follows. Given the interface position, the solution and jumps at the time step $n\Delta t$, $\mathbf{X}^n$, Φ^n, Ψ_0^n, Ψ_1^n, and Ψ_2^n, we update Φ^{n+1} by using the following two steps:

Step 1. Advance the interface to the position $\mathbf{X}^{n+1}$ and then apply the modified Lax–Wendroff scheme to obtain the intermediate solution Φ^* due to the convection part

$$\Phi^* = A\Phi^{n} + \widetilde{C}_0\Psi_0^{n} + \widetilde{C}_1\Psi_1^{n} + \widetilde{C}_2\Psi_2^{n}.$$

Step 2. Update the solution Φ^{n+1} by solving the diffusion part using the backward Euler scheme

$$\frac{\Phi^{n+1}-\Phi^*}{\Delta t} = \frac{1}{Pe}\left(\Delta_h\Phi^{n+1} + C_0\Psi_0^{n+1} + C_1\Psi_1^{n+1} + C_2\Psi_2^{n+1}\right)$$

and update the discrete jump conditions Ψ_0^{n+1}, Ψ_1^{n+1}, and Ψ_2^{n+1}.

In Step 1, we not only obtain the intermediate solution Φ^* but also the jump condition for Φ^* (say Ψ_1^*) on the interface $\Sigma(t_{n+1})$ as $\Psi_1^* = \Psi_1^n$. This is due to the fact that solution on the interface also flows along some characteristic line so that $[\![\phi^*]\!] = \phi^+(s,t_n)$. In Step 2, Ψ_0^{n+1} is thus approximated by using $(\Psi_1^{n+1}-\Psi_1^*)/\Delta t =$

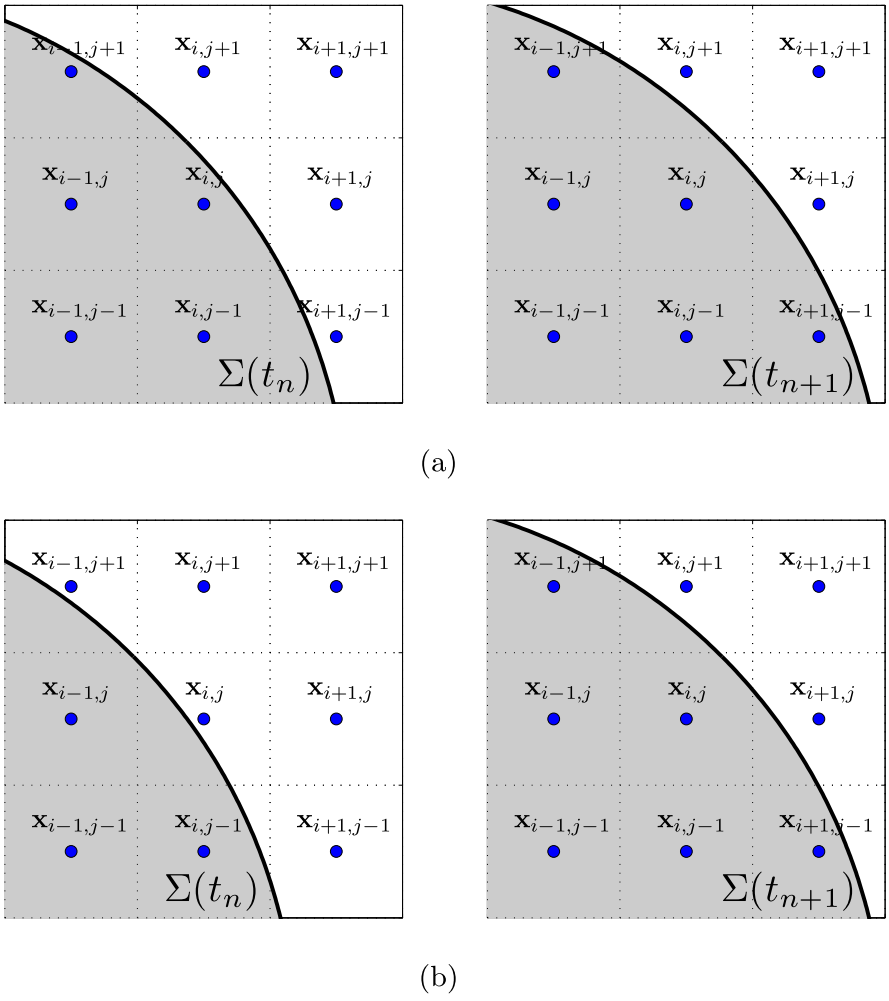

Fig. 5. Two cases for the interface moves from $\Sigma(t_n)$ (left figure) to $\Sigma(t_{n+1})$ (right figure).

Ψ_0^{n+1}/Pe. Thus, solving for Φ^{n+1} is nothing but following the diffusion equation solver developed in previous section. The above splitting procedure is expected to have first-order convergence due to the backward Euler method in Step 2. Of course, one can enhance the overall accuracy of the present method by employing a higher order scheme such as Strang splitting [12], however, in this paper we aim to implement the numerical solver in a simple manner. Next, we shall carry out a numerical test for the convergence and efficiency of our proposed numerical method.

3.3. Numerical accuracy and efficiency study

In this subsection, we perform a numerical test for the convergence and efficiency of the present method for Eq. (14) in the exterior of a moving interface in the computational domain $\Omega = [-2, 2] \times [-2, 2]$. Here, we choose the initial interface as a unit circle centered at the origin and apply the shear flow $\mathbf{u} = (y, 0)$ as our prescribed velocity so the evolving interface $\Sigma(t)$ can be explicitly described as $\mathbf{X}(s, t) = (\cos(s) + t \sin(s), \sin(s))$ by solving Eq. (17). In this moving irregular domain, we can construct an analytical solution for Eq. (14) as

$$\phi(x, y, t) = 1 + 0.5 \exp\left(\frac{-(t + t^3/3)}{Pe}\right) \sin(x - yt), \quad \mathbf{x} \in \Omega^+. \tag{20}$$

Again, three cases are tested by imposing different inner boundary conditions as Dirichlet, Neumann, and Robin type boundary condition on Σ. In these cases, the solution is subject to the Dirichlet boundary condition on $\partial\Omega$.

We choose the Peclet number $Pe = 1$ with different grid sizes $N = 128, 256, 512, 1024$. and the time step size $\Delta t = h/2$. Table 2 shows the maximum errors for the numerical solution at the terminal time $T = 1$. One can see that the present numerical result performs clean first-order convergence for the numerical solution ϕ in each case. The average number of GMRES iterations in Step 2 is just within two even as the grid number doubles. We further show the snapshots for the numerical solution (obtained by using inner Robin boundary condition on Σ with the grid size $N = 512$) at different times in Fig. 6.

4. Interfacial flows with soluble surfactant

In this section, we apply the IIM solver developed in previous section to simulate interfacial flow problems with soluble surfactant. Consider an incompressible Navier–Stokes flow consisting of two-phase fluids in a fixed rectangular domain Ω with an immersed droplet interface Σ represented by the parametric form $\mathbf{X}(s, t)$ with the parameter $s \in [0, 2\pi]$ (see Fig. 1). It is assumed that the surfactant exists on the interface as a monolayer and is adsorbed from or desorbed to the bulk fluid in Ω^+; that is, the surfactant is soluble in the exterior bulk but not in the interior droplet. The interface is contaminated by the surfactant so that

Table 2
Mesh refinement results for the numerical solution at the terminal time $T = 1$.

N	Dirichlet			Neumann			Robin		
	$\|\phi_h - \phi_e\|_\infty$	Rate	Iter.	$\|\phi_h - \phi_e\|_\infty$	Rate	Iter.	$\|\phi_h - \phi_e\|_\infty$	Rate	Iter.
128	3.988E − 04	–	1.14	9.038E − 04	–	1.03	6.173E − 04	–	1.03
256	1.996E − 04	0.99	1.14	4.072E − 04	1.15	1.67	2.989E − 04	1.04	1.67
512	9.880E − 05	1.01	1.84	2.024E − 04	1.00	1.85	1.486E − 04	1.00	1.85
1024	4.927E − 05	1.00	1.93	1.009E − 04	1.00	1.93	7.410E − 05	1.00	1.93

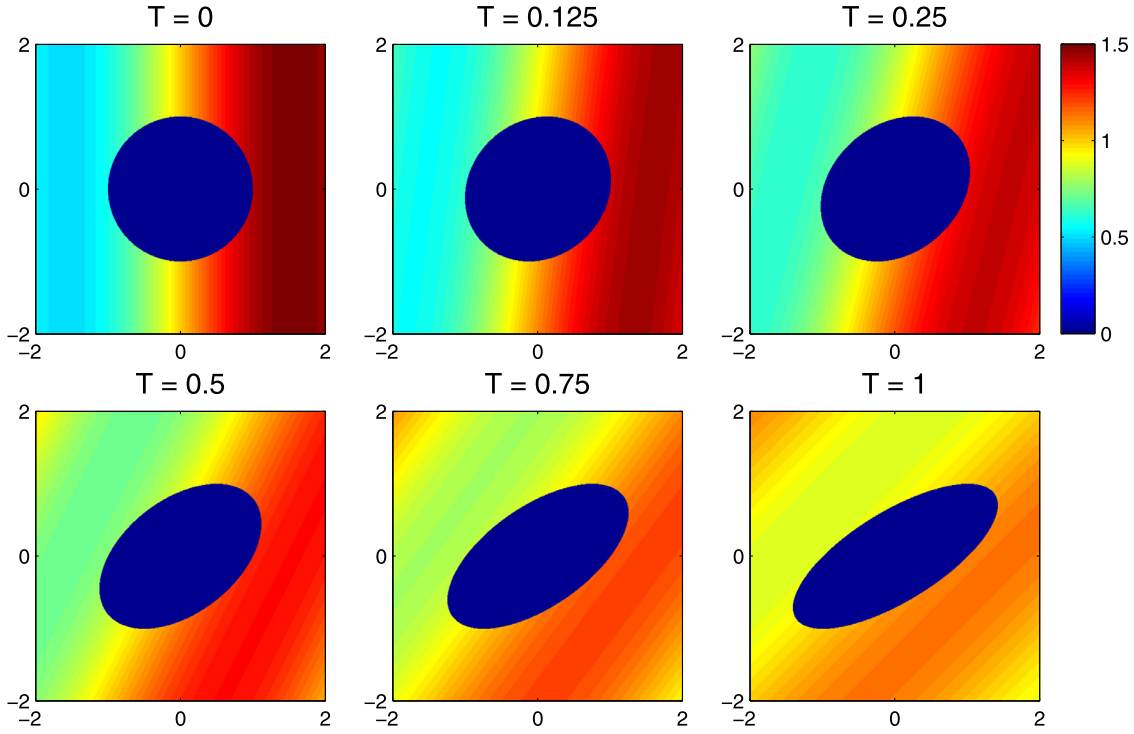

Fig. 6. The snapshots of the numerical solution at $T = 0, 0.125, 0.25, 0.5, 0.75$ and 1. The numerical solution is subject to Dirichlet boundary condition on the outer boundary $\partial\Omega$ and Robin boundary condition on the inner boundary $\Sigma(t)$.

the distribution of the surfactant changes the surface tension σ accordingly. Therefore, we have to introduce two surfactant concentrations in the system; namely, the surface concentration $\Gamma(s, t)$ along the interface Σ, and the bulk concentration $\phi(x, y, t)$ in the region Ω^+.

In order to formulate the fluid part of the problem using the IB approach, we simply treat the interface as an immersed boundary that exerts force on the fluids and moves with local fluid velocity. For simplicity, we assume equal viscosity μ and density ρ for both fluids, and neglect the gravity. Certainly, the present Navier–Stokes solver can be replaced by the one with different density and viscosity ratios. In this paper, we concentrate on the numerical investigation of droplet dynamics under shear flow in the presence of soluble surfactant effects. As in [4], we scale all physical variables by the associated characteristic scales as follows:

$$\mathbf{x}^* = \frac{\mathbf{x}}{R}, \quad t^* = t\dot{\gamma}, \quad \mathbf{u}^* = \frac{\mathbf{u}}{R\dot{\gamma}}, \quad p^* = \frac{p}{\rho\dot{\gamma}^2R^2},$$
$$\sigma^* = \frac{\sigma}{\sigma_e}, \quad \Gamma^* = \frac{\Gamma}{\Gamma_e}, \quad \phi^* = \frac{\phi}{\phi_e},$$

where R is the initial drop radius, $\dot{\gamma}$ is the shear rate, and σ_e, Γ_e, ϕ_e are the surface tension, surface surfactant concentration, bulk concentration at equilibrium respectively. Notice that, the normalization of the surface tension σ enables us to investigate the effect due to nonuniformly distributed surface tension (one shall see the relation between the surface tension and surfactant later). After some careful calculations, the non-dimensional Navier–Stokes flow in the usual IB formulation can be written as (dropping * in all notations)

$$\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\Delta\mathbf{u} + \frac{1}{ReCa}\int_\Sigma \mathbf{F}(s,t)\delta(\mathbf{x} - \mathbf{X}(s,t))|\mathbf{X}_s|\,\mathrm{d}s \quad \text{in } \Omega, \tag{21}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega, \tag{22}$$

$$\frac{\partial\mathbf{X}}{\partial t}(s,t) = \mathbf{U}(s,t) = \int_\Omega \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}(s,t))\,\mathrm{d}\mathbf{x} \quad \text{on } \Sigma, \tag{23}$$

$$\mathbf{F}(s,t) = \frac{\partial}{\partial s}(\sigma(s,t)\boldsymbol{\tau}(s,t)) \quad \text{on } \Sigma, \tag{24}$$

where $\mathbf{u} = (u, v)$ is the fluid velocity and p is the pressure. The dimensionless numbers are the Reynolds number $Re = \rho R^2\dot{\gamma}/\mu$ describing the ratio between the inertial force and the viscous force, and the Capillary number $Ca = \mu R\dot{\gamma}/\sigma_e$ describing the strength of the viscous force and the surface tension. The presence of surfactant will reduce the surface tension of the interface following the Langmuir equation of state [22]

$$\sigma = 1 + El\ln\left(\frac{1 - \chi\Gamma}{1 - \chi}\right). \tag{25}$$

In the above equation, $El = \bar{R}T\Gamma_\infty/\sigma_e$ is the elasticity number measuring the sensitivity of the surface tension to the surfactant concentration and $\chi = \Gamma_e/\Gamma_\infty$ is the fraction of the surfactant coverage, where $\bar{R}$ is the universal gas constant, T is the temperature, and Γ_∞ is the maximum surfactant packing. By taking the adsorption and desorption of bulk surfactant into account, along the above normalization scales, the system for the dimensionless surface-bulk surfactant concentration equations is

$$\frac{\partial \phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = \frac{1}{Pe}\Delta\phi \quad \text{in } \Omega^+, \quad \frac{1}{Pe}\frac{\partial \phi}{\partial \mathbf{n}}\Big|_\Sigma = Bi\, h_{ad}\big[\phi^+(1+K_{ad}-K_{ad}\Gamma)-\Gamma\big], \tag{26}$$

$$\frac{\partial \Gamma}{\partial t} + (\nabla_s\cdot\mathbf{U})\Gamma = \frac{1}{Pe_s}\Delta_s\Gamma + Bi\big[\phi^+(1+K_{ad}-K_{ad}\Gamma)-\Gamma\big] \text{ on } \Sigma, \tag{27}$$

where ϕ^+ denotes the bulk concentration adjacent to the interface and the term surface divergence of the velocity field $\nabla_s\cdot\mathbf{U}$ is calculated as $\mathbf{U}_s\cdot\boldsymbol{\tau}/|\mathbf{X}_s|$. Notice that, here the surface concentration $\Gamma(s,t)$ is defined in the Lagrangian manner, so $\frac{\partial\Gamma}{\partial t}$ represents the material derivative of Γ. In general, the flux of the bulk concentration is controlled by the adsorption-desorption mechanism between the surface and bulk surfactant [4,20]. As one can see from the inner boundary condition on Σ in Eq. (26), the first coupled term on the righthand side describes the mass adsorption to the droplet interface while the second term denotes the mass desorption to the bulk fluid. As before, the convection-diffusion Eq. (26) for the bulk concentration is reformulated in the fixed regular domain Ω incorporating with the jump discontinuity ($[\![\phi]\!]$ and $[\![\phi_n]\!]$) across the interface Σ, which can be solved efficiently by the present immersed interface method.

With the following physical characteristic parameters, the desorption coefficient α, the adsorption coefficient β, the bulk diffusivity D_b, and the surface diffusivity D_s, the above dimensionless numbers are defined as: the surface Peclet number, $Pe_s = R^2\dot{\gamma}/D_s$ is the surface convective flux to the surface diffusion; the Biot number, $Bi = \alpha/\dot{\gamma}$ is the ratio of desorption to surface convective time; the adsorption number, $K_{ad} = \beta\phi_e/\alpha$ is the adsorption to desorption rate; the bulk Peclet number, $Pe = R^2\dot{\gamma}/D_b$ is the convective flux to the bulk diffusion; the adsorption depth, $h_{ad} = \Gamma_e/\phi_e R$ is the depth beneath the interface diluted by surfactant adsorption. Here we have an immediate qualitative behavior of the effect of Biot number: when $Bi \ll 1$, the source term in Eq. (27) is quite small, hence the profile of the surface surfactant is nearly similar to that obtained in the insoluble case. When $Bi \gg 1$, the absorption from the bulk surfactant is overwhelming, thereby the surface surfactant should remain close to its equilibrium profile. The quantitative behavior of Biot number shall be investigated in details in numerical experiments later. It is interesting to see that, with some suitable boundary conditions for ϕ, one can show that the total amount of the bulk and surface surfactant is conserved as

$$\frac{\mathrm{d}}{\mathrm{d}t}M(t) = \frac{\mathrm{d}}{\mathrm{d}t}\left(h_{ad}\int_\Omega \phi\ \mathrm{d}\mathbf{x} + \int_\Sigma \Gamma|\mathbf{X}_s|\ \mathrm{d}s\right) = 0.$$

This conservation law for the total surfactant mass will be checked later in our numerical results.

4.1. Numerical algorithm

Here we describe the numerical scheme to simulate the entire system of Eqs. (21)–(27). As our droplet and vesicle work in [9,10], the idea is to solve the fluid equations by the usual immersed boundary method and the bulk surfactant by the immersed interface method proposed in Section 3. We consider the computational domain as a rectangle $\Omega = [a,b]\times[c,d]$. Within this domain, a uniform Cartesian grid with mesh width h in both x and y directions is employed. The fluid variables are defined on the standard staggered marker-and-cell (MAC) manner [6]. That is, the velocity component u and v are defined at the cell vertical and horizontal edges $(x_{i-1/2}, y_j) = (a+(i-1)h, c+(j-1/2)h)$ and $(x_i, y_{j-1/2}) = (a+(i-1/2)h, c+(j-1)h)$ respectively, while the pressure p and the bulk concentration ϕ are both defined at the cell center $(x_i, y_j) = (a+(i-1/2)h, c+(j-1/2)h)$. As mentioned before, for the droplet interface, we use the spectral collection points $s_k = k\Delta s,\ k = 0, 1, \cdots, M$ with $s = 2\pi/M$ to represent the Lagrangian markers $\mathbf{X}_k = \mathbf{X}(s_k)$ so that any spatial derivatives can be performed spectrally accurate by using Fast Fourier Transform (FFT).

The time-stepping for the overall system can be described as follows. At the beginning of each time step $n\Delta t$, the fluid velocity $\mathbf{u}^n$, the droplet configuration $\mathbf{X}^n$, the surface concentration Γ^n, and the bulk concentration ϕ^n must be given. The entire system for Eqs. (21)–(27) is solved by a hybrid immersed boundary and immersed interface method, that is, the fluid system is solved by the traditional IB method and the bulk surfactant is solved by the IIM introduced in Section 3. The detailed numerical algorithm is given as follows.

1. Given the interface markers $\mathbf{X}^n$, we first compute surface tension σ^n by using Eq. (25) and the unit tangent vector $\boldsymbol{\tau}^n$, then compute the interfacial tension force $\mathbf{F}^n$ in Eq. (24). The above terms involving spatial derivatives with respect to s can be computed with spectral accuracy.
2. Distribute the interfacial surface tension force from the Lagrangian markers to the fluid grid points by using the discrete delta function as in traditional IB method.
3. Solve the Navier–Stokes equations by the pressure-increment projection method to obtain the new velocity $\mathbf{u}^{n+1}$. This procedure involves solving two modified Helmholtz-type equations for the intermediate velocity and one Poisson equation for the pressure-increment which again can be efficiently done by applying the fast direct solver provided by Fishpack.
4. Interpolate the new velocity on the fluid grid points to the marker points and then move the markers to new positions $\mathbf{X}^{n+1}$ as in Eq. (23). Here, we employ an equi-arclength parametrization technique developed in [17] so that the Lagrangian markers can be equally distributed.
5. Solve the surface surfactant concentration in Eq. (27) by using the backward Euler scheme developed in [17,23].
6. Solve the bulk surfactant concentration in Eq. (26).

Here, the detailed numerical implementation of Steps 2–4 are quite standard in IB method and can be found in any related literatures while Step 6 is described in Section 3. Notice that in Step 6, the coefficients used in Eq. (19) do not coincide with the fluid solutions for u and v at the staggered grid, a simple interpolation is thus implemented.

4.2. Numerical setup

Throughout the following numerical experiments, we initially put a unit circular drop $\mathbf{X}(s,0) = (\cos s, \sin s),\ 0 \le s \le 2\pi$ at the center of the computational domain $\Omega = [-8,8]\times[-2,2]$. The initial velocity field is set as the shear flow profile $\mathbf{u} = (y, 0)$. In each simulation, the periodic boundary condition and the no-flux Neumann boundary condition are imposed on the bulk surfactant at $x = \pm 8$ and $y = \pm 2$, respectively. The mesh width is $h = 16/N$ with the grid size $N = 1024$ and the time step size is chosen as $\Delta t = h/8$. We choose the number of Lagrangian marker size M so that $\Delta s = 2\pi/M < h$. Practically, the average number of GMRES iterations in each step is just within 2 steps in most cases. In each

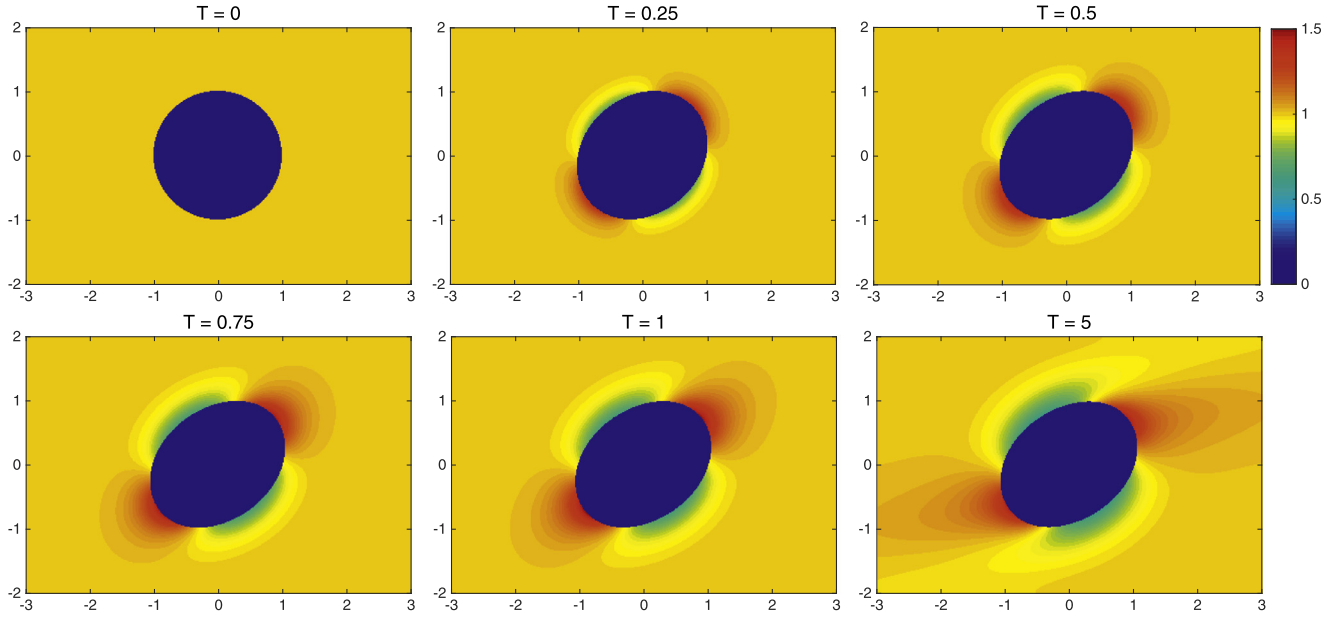

Fig. 7. The snapshots of the bulk surfactant ϕ at $T = 0$, 0.25, 0.5, 0.75, 1 and 5.

case, we choose the Reynolds number $Re = 0.1$, the elasticity number $El = 0.2$, the surfactant coverage $\chi = 0.9$, the surface Peclet number $Pe_s = 1$, the adsorption depth $h_{ad} = 1$, and the adsorption number $K_{ad} = 9$. The initial surface and bulk surfactant concentration are all set to be uniform one in the domains they are defined.

In the following subsections, we perform a series of numerical simulations for the interfacial flows with soluble surfactant. We first validate the present numerical method by running a simulation comparing to other results in literatures and check the convergence for our present numerical algorithm. Then, we investigate the effect of the Biot number Bi and the bulk Peclet number Pe to the overall system.

4.3. Droplet dynamics under shear flow with soluble surfactant

In this case we choose the capillary number $Ca = 0.1$, the bulk Peclet number $Pe = 10$, and the Biot number $Bi = 10$. Fig. 7 shows the snapshots of numerical results for the bulk surfactant ϕ at different times $T = 0$, 0.25, 0.5, 0.75, 1, and $T = 5$. One can see that, as the flow is driven by the shear stresses, the droplet is elongated and the surface surfactant is swept to the both drop tips. Hence, the lower surface surfactant concentration at the drop sides leads to the adsorption from the bulk surfactant through the mass flux, while the higher surface surfactant at the drop tips results in the desorption to the bulk. As a consequence, the bulk surfactant concentration decreases near the drop sides and increases near the both tips as we can see from Fig. 7. The system reaches equilibrium around $T = 5$ and the concentration profile of the surface surfactant along the parameter s is given in Fig. 8. These results are in a good agreement with those simulations in [12,26].

Moreover, we perform a convergence study of the numerical algorithm proposed in Section 4.1. Table 3 shows the errors for the fluid variables u and v, the interface position $\mathbf{X}$, the surface surfactant Γ, the bulk surfactant ϕ, and the relative error of the total mass at $T = 2$. The numerical result shows that first-order accuracy is obtained roughly for the fluid variables as in traditional IB method; the surfactant concentrations Γ and ϕ attain around first-order accuracy too. Again, despite the present method does not guarantee the perfect conservation of the total surfactant mass numerically, the relative error of the total mass still behaves convergent and the magnitude ranges from $0.02\% - 0.08\%$ when the mesh is refined.

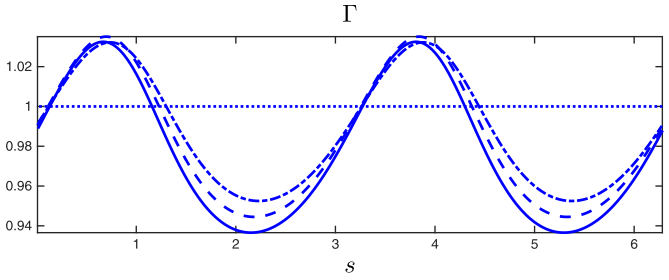

Fig. 8. The evolutional plots for the surface surfactant profile Γ at $T = 0$ (dotted line), 0.5 (dash-dotted line), 1 (dash line), and 5 (solid line).

Table 3
Mesh refinement results for the fluid component u and v, the interface position $\mathbf{X}$, the surface surfactant Γ, the bulk surfactant ϕ, and the relative error of the total mass at $T = 2$.

N	$\|u_{2N} - u_N\|_\infty$	Rate	$\|v_{2N} - v_N\|_\infty$	Rate	$\|\mathbf{X}_{2N} - \mathbf{X}_N\|_\infty$	Rate
128	2.402E − 02	–	1.566E − 02	–	4.802E − 02	–
256	1.086E − 02	1.14	1.047E − 02	0.58	2.120E − 02	1.18
512	6.362E − 03	0.77	5.464E − 03	0.94	1.018E − 02	1.06
1024	3.384E − 03	0.91	2.752E − 03	0.99	5.018E − 03	1.02
N	$\|\Gamma_{2N} - \Gamma_N\|_\infty$	Rate	$\|\phi_{2N} - \phi_N\|_\infty$	Rate	$\frac{\|M(2) - M(0)\|}{M(0)}$	Rate
128	9.581E − 03	–	9.941E − 02	–	7.840E − 04	–
256	4.767E − 03	1.01	5.303E − 02	0.90	5.960E − 04	0.39
512	2.429E − 03	0.97	2.429E − 02	1.12	3.135E − 04	0.92
1024	1.197E − 03	1.02	1.236E − 02	0.97	2.049E − 04	0.61

4.4. Effect of Biot number Bi

In this test, we study the effect of the Biot number Bi on droplet deformations. With moderate capillary numbers Ca, the droplet under shear flow reaches an equilibrium state as a nearly inclined ellipse shape. To quantify the droplet shape, we use the deformation number D_f by $D_f = L_d - B_d / L_d + B_d$, where L_d and B_d are the corresponding lengths of the major and minor axes of the droplet.

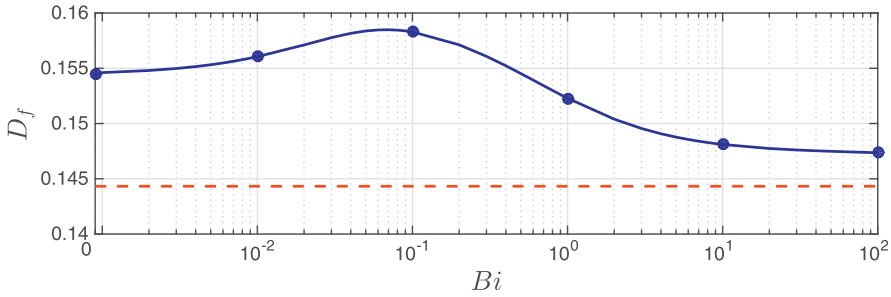

Fig. 9. The Biot number Bi versus the droplet deformation number D_f. The red dashed line shows the deformation for the clean drop case just for comparison purpose (here the values of Biot number are meaningless). $Ca = 0.1$.

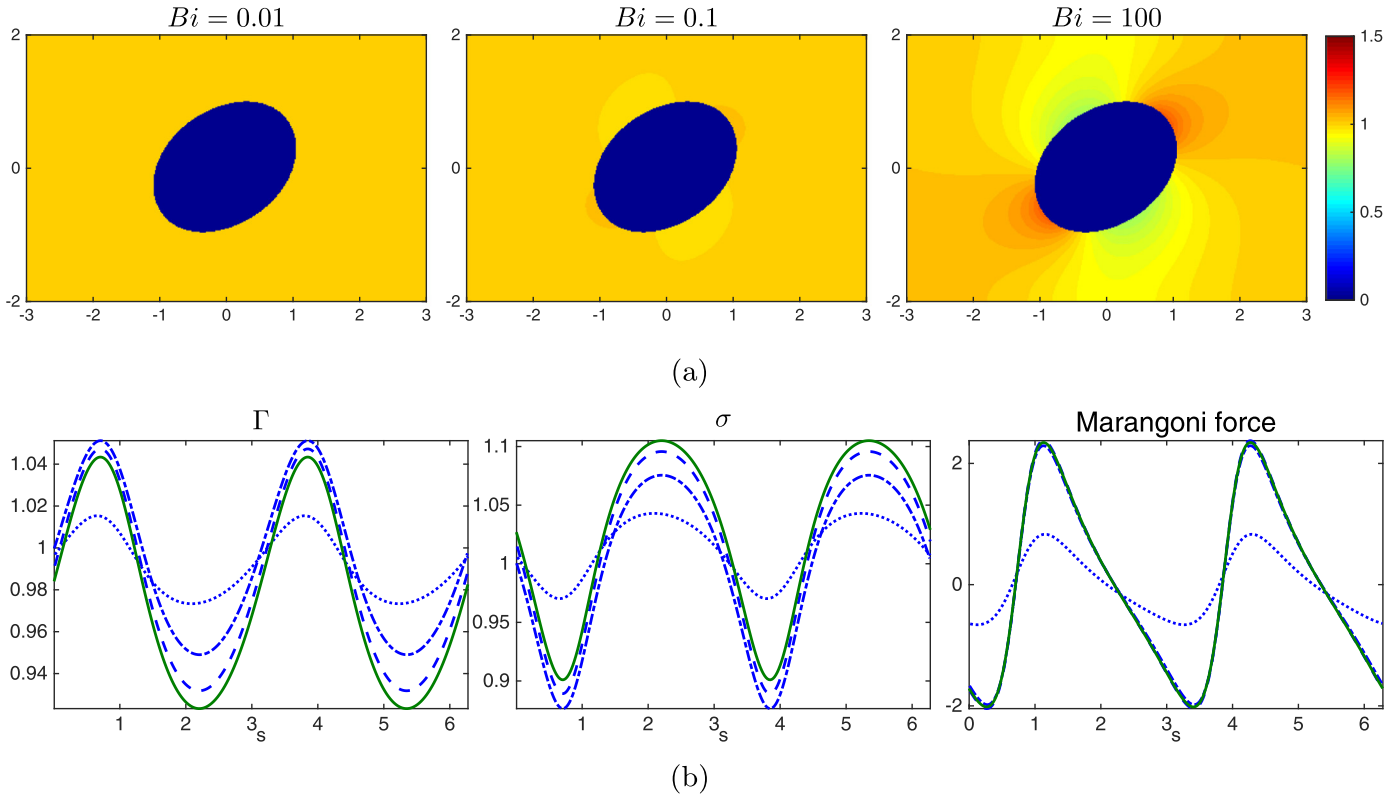

Fig. 10. (a) The concentration distribution of the bulk surfactant at equilibrium for $Bi = 0.01$, 0.1, and 100. (b) The interfacial profiles for the surface surfactant Γ, the surface tension σ, and the Marangoni force at equilibrium for $Bi = 0$ (solid line), 0.01 (dashed line), 0.1 (dash-dotted line), and 100 (dotted line). $Ca = 0.1$.

In the first case, we fix the bulk Peclet number $Pe = 1$ and the capillary number $Ca = 0.1$, but vary the Biot number from $Bi = 0$ to $Bi = 100$. The case $Bi = 0$ represents the insoluble surfactant case for which we do not need to solve the bulk surfactant equation. We further compare the results with the case of clean interface (fixed surface tension $\sigma = 1$ without solving any surface and bulk surfactant equations). The result of the deformation as a function of the Biot number is given in Fig. 9 in which the red dashed line shows the deformation for the clean drop case (just for comparison purpose, here the values of Biot number are meaningless). The bulk surfactant distributions, the interfacial profiles for Γ, σ, and Marangoni force σ_s/Ca corresponding to $Bi = 0$, 0.01, 0.1, 100 are shown in Fig. 10. For the insoluble case $Bi = 0$, the surface surfactant Γ is swept toward the drop tips until the Marangoni force develops to resist further accumulation in the region of the drop tips. The accumulation of surfactant results in a lower surface tension at the tips, which requires a larger curvature force to balance the normal capillary force. Thus, the nonuniform distribution of Γ leads to the so-called *tip stretching* for the contaminated droplet which is more highly deformed than the clean droplet. Upon an increase of the Biot number from 0 to 0.01, one can see that the Γ profile is slightly different from the insoluble case due to the less mass transfer (small Bi). While the adsorption-desorption mechanism on the bulk surfactant yields a small increase on Γ on the entire droplet interface, thereby the tip stretching becomes more pronounced and the more deformed equilibrium droplet is obtained. A similar behavior is found at $Bi = 0.1$. Increasing further the Biot number up to $Bi = 100$, the more mass transfer yields a decline in the surface surfactant gradient and thus the Γ profile becomes more uniformly distributed as its equilibrium value as the clean interface (and so does the surface tension and the Marangoni force). As a result, the equilibrium shape of the droplet is less deformed as compared to the case with moderate values Bi. This nonmonotonic variation of the deformation number D_f is also found in the case of imposing a spherical droplet under extensional flow at high coverage surfactant $\chi = 0.99$ in axisymmetric coordinates in [4].

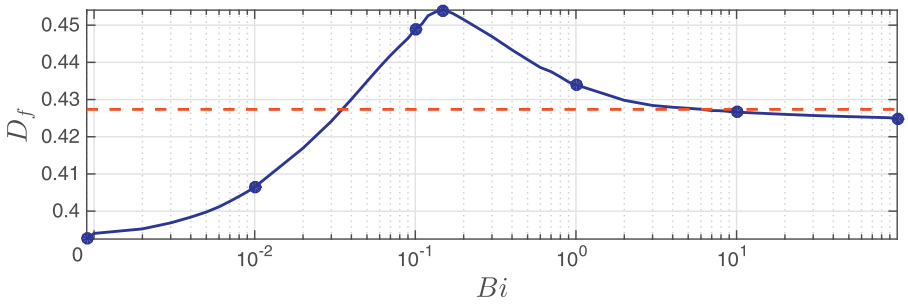

Fig. 11. The Biot number Bi versus the droplet deformation number D_f. The red dashed line shows the deformation for the clean drop case just for comparison purpose (here the values of Biot number are meaningless). $Ca = 0.3$.

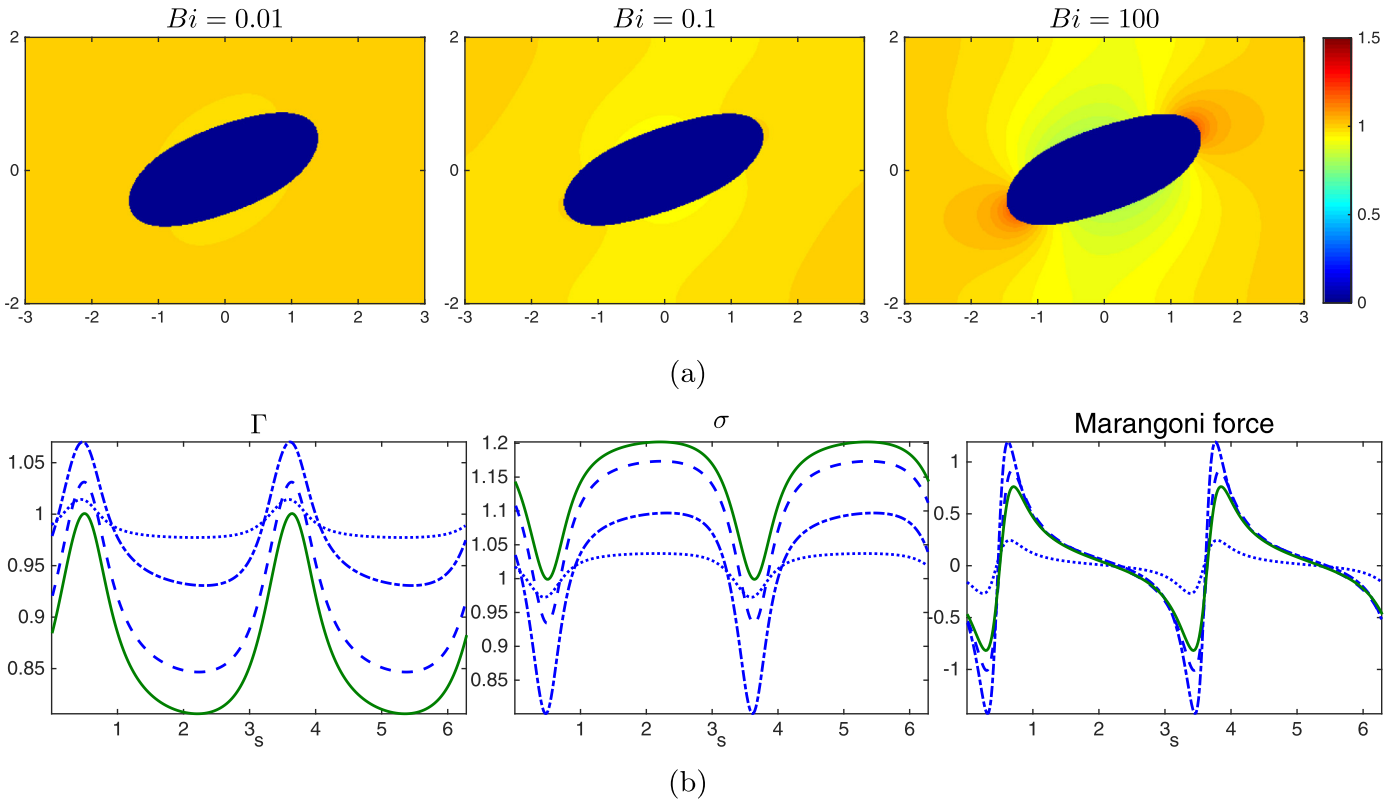

Fig. 12. (a) The concentration distribution of the bulk surfactant at equilibrium for $Bi = 0.01$, 0.1, and 100. (b) The interfacial profiles for the surface surfactant Γ, the surface tension σ, and the Marangoni force at equilibrium for $Bi = 0$ (solid line), 0.01 (dashed line), 0.1 (dash-dotted line), and 100 (dotted line). $Ca = 0.3$.

In the second case, we follow the same setup as in the first case except choosing a larger capillary number $Ca = 0.3$. Fig. 11 shows the deformation number versus the Biot number and Fig. 12 gives the equilibrium bulk surfactant concentration, the surface surfactant, the surface tension, and the Marangoni force corresponding to the results of $Bi = 0, 0.01, 0.1$, and 100. For the insoluble surfactant case ($Bi = 0$), again the accumulation of Γ is formed on the tips, whereas the strong elongation of the droplet dilutes the density of surface surfactant in average, which results in a larger surface tension resisting further deformation. Due to the *dilution* of the surface surfactant, the smaller deformation number comparing to the clean case is obtained. As increasing Bi from 0 to 0.01, we see again that the bulk surfactant is not only supplied on the sides but also at the tips. This leads to a weaker surface tension and stronger Marangoni force as compared to the case with $Bi = 0$, and thus equilibrium shape with a larger deformation is obtained. As increasing to moderate $Bi = 0.1$, the droplet deformation is more pronounced than the one in $Bi = 0.01$ and again the similar tendency is observed as in $Bi = 0.01$ case. When increasing the Biot number up to $Bi = 100$, Γ becomes uniform due to rapid mass transfer and the Marangoni force is strongly diminished yielding a result similar to the clean interface case.

4.5. *Effect of bulk Peclet number* Pe

Next we turn our attention to the study of the effect of bulk Peclect number Pe on the droplet deformation. In this test, we fix the Biot number by $Bi = 100$ and vary the bulk Peclet number from $Pe = 0.1$ to $Pe = 100$.

We first run the cases with $Ca = 0.1$. Fig. 13 shows the deformation D_f as a function of Pe, and Fig. 14 gives the bulk surfactant concentration, interfacial profiles for Γ, σ, and Marangoni force at equilibrium for the selected $Pe = 0.1$, 1, and 10. As one can see, for small bulk Peclet number $Pe = 0.1$, the strong bulk diffusivity redistributes the bulk surfactant concentration near its equilibrium, which supplies Γ to become uniformly distributed as its initial concentration. Upon increasing Pe, the bulk surfactant transfer becomes slower which leads to the depletion of surface surfactant

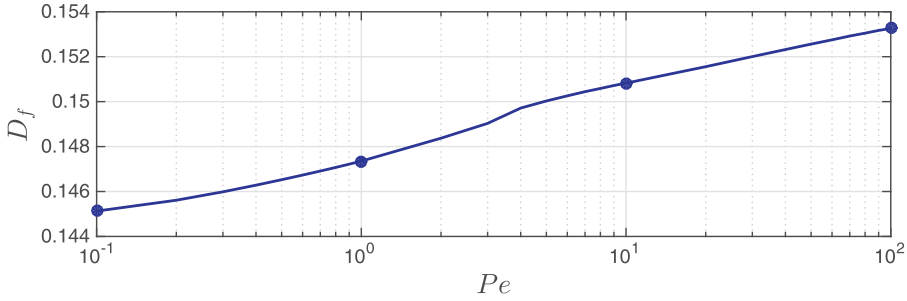

Fig. 13. The bulk Peclet number Pe versus the droplet deformation number D_f. $Ca = 0.1$.

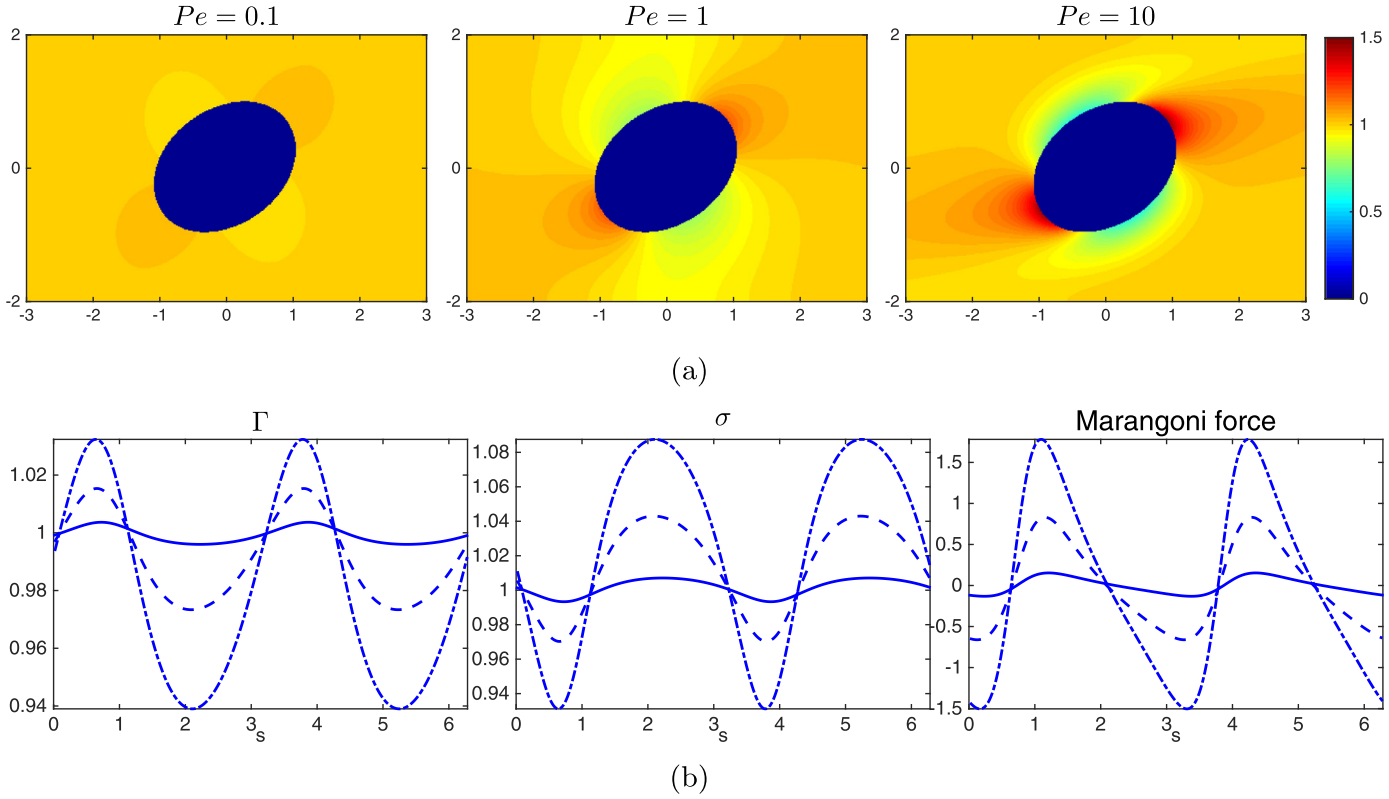

Fig. 14. (a) The concentration distribution of the bulk surfactant at equilibrium for $Pe = 0.1$, 1, and 10. (b) The interfacial profiles for the surface surfactant Γ, the surface tension σ, and the Marangoni force at equilibrium for $Pe = 0.1$ (solid line), 1 (dashed line), and 10 (dash-dotted line). $Ca = 0.1$.

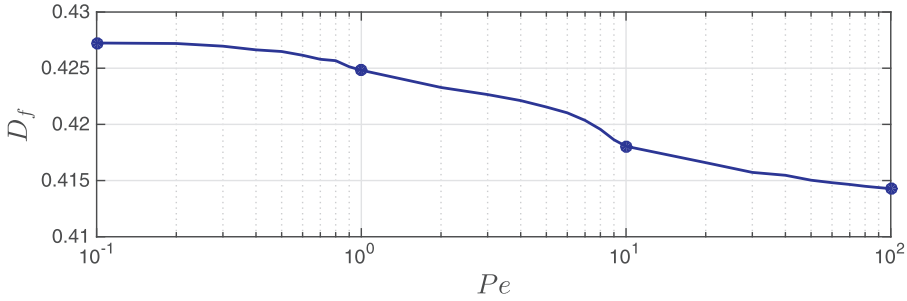

Fig. 15. The bulk Peclet number Pe versus the droplet deformation number D_f. $Ca = 0.3$.

on the drop sides but the accumulation on the tips. The resulting surface gradient on Γ yields the stronger Marangoni force. Despite the increase of surface tension on the drop sides, the reduction of the surface tension on the both tips leads to a larger deformation comparing with the small Pe cases.

We also investigate the cases of $Ca = 0.3$. The results are shown in Figs. 15 and 16. As expected, the drop deformation is larger than the previous case of $Ca = 0.1$ since the larger capillary number results smaller interfacial forces as seen from Eq. (21). However, unlike the case of $Ca = 0.1$, the present deformation number decreases as Pe number increases, whereas those interfacial profiles for Γ, σ, and Marangoni force seem to be similar to the case of $Ca = 0.1$. We attribute this to the surface surfactant distribution on the strongly elongated droplet. Since the arclength of the droplet

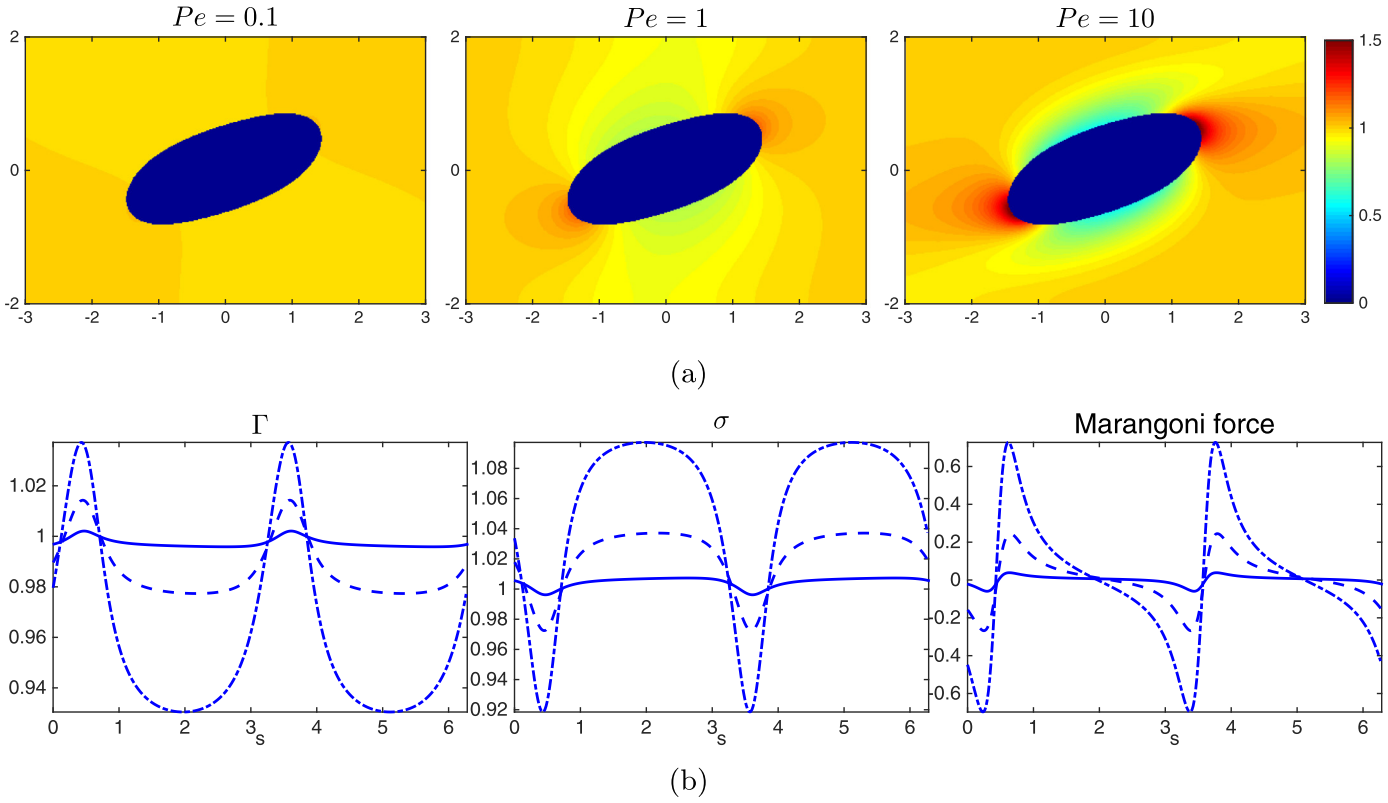

Fig. 16. (a) The concentration distribution of the bulk surfactant at equilibrium for $Pe = 0.1$, 1, and 10. (b) The interfacial profiles for the surface surfactant Γ, the surface tension σ, and the Marangoni force at equilibrium for $Pe = 0.1$ (solid line), 1 (dashed line), and 10 (dash-dotted line). $Ca = 0.3$.

is significantly increased (relative to the case $Ca = 0.1$), the surface tension is increased in average due to the larger region on the droplet sides (whereas the reduction of the surface tension is just in a narrow tip region). As a result, the droplet deformation becomes smaller as Pe increases. We have also tested the above two cases with smaller Biot number. However, the results are similar so we omit here.

5. Conclusions

In this paper, we have developed an immersed interface method to solve the diffusion equation in a fixed irregular domain and the convection-diffusion equation in a moving irregular domain. The major idea is to extend the computational domain to a regular domain and replace the irregular boundary condition by the interface jump conditions so the solution can be computed accurately and efficiently. We have checked the convergence and the efficiency of the proposed methods via some numerical experiments and the expected accuracy is obtained.

Moreover, we have extended the present immersed interface solver to study the interfacial problems with soluble surfactant. The bulk surfactant is governed by a convection-diffusion equation with the adsorption-desorption interaction with the surface surfactant which can be solved efficiently by the present method. The fluid equations with interfacial forces are solved by the traditional IB method. To carefully validate our method, we perform convergency study on the fluid and interfacial variables. We then conduct a series of simulations to investigate the effect of Biot number and bulk Peclet number on the droplet deformation. As for future works, we plan to extend our present methodology to the 3D axisymmetric case or even fully 3D simulations. Moreover, we intend to apply the present scheme to simulate the phoretic self-propulsion swimmer problem as discussed in [19,27].

Acknowledgments

W.-F. Hu acknowledges support by Ministry of Science and Technology, Taiwan under research grant MOST-105-2115-M-005-008-MY2. M.-C. Lai acknowledges support from MOST-104-2115-M-009-014-MY3 and NCTS. C. Misbah acknowledges support from CNES (Centre National d'Etudes Spatiales). All the authors acknowledge the French-Taiwanese ORCHID cooperation grant and MoST for a support allowing initiation of this project.

References

[1] Adams J, Swarztrauber P, Sweet R. Fishpack- a package of Fortran subprograms for the solution of separable elliptic partial differential equations. 1980. Available at http://www.netlib.org/fishpack.
[2] Booty MR, Siegel M. A hybrid numerical method for interfacial fluid flow with soluble surfactant. J Comput Phys 2010;229:3864–83.
[3] Chen KY, Lai MC. A conservative scheme for solving coupled surface-bulk convection-diffusion equations with an application to interfacial flows with soluble surfactant. J Comput Phys 2014;257:1–18.
[4] Eggleton CD, Stebe KJ. An adsorption-desorption-controlled surfactant on a deforming droplet. J Colloid Interface Sci 1998;208:68–80.
[5] Guo H, Yang X. Gradient recovery for elliptic interface problem: II. Immersed finite element methods. J Comput Phys 2017;338:606–19.
[6] Harlow FH, Welsh JE. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. Phys Fluids 1965;8:2181–9.
[7] Lai MC, Tseng HC. A simple implementation of the immersed interface methods for stokes flows with singular forces. Comput Fluids 2008;37:99–106.
[8] Hossan MR, Dillon R, Dutta P. Hybrid immersed interface-immersed boundary methods for AC dielectrophoresis. J Comput Phys 2014;270:640–59.
[9] Hu WF, Lai MC, Young YN. A hybrid immersed boundary and immersed interface method for electrohydrodynamic simulations. J Comput Phys 2015;282:47–61.
[10] Hu WF, Lai MC, Seol Y, Young YN. Vesicle electrohydrodynamic simulations by coupling immersed boundary and immersed interface method. J Comput Phys 2016;317:66–81.
[11] Ito K, Lai MC, Li Z. A well-conditioned augmented system for solving Navier-Stokes equations in irregular domains. J Comput Phys 2009;228:2616–28.
[12] Khatri S, Tornberg AK. An embedded boundary method for soluble surfactants with interface tracking for two-phase flows. J Comput Phys 2014;256:768–90.

[13] Kolahdouz EM, Salac D. Electrohydrodynamics of three-dimensional vesicles: a numerical approach. SIAM J Sci Comput 2015;37:B473–94.
[14] LeVeque RJ. Numerical methods for conservation laws. Springer; 1992.
[15] LeVeque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J Numer Anal 1994;31(4):1019–44.
[16] Li Z, Ito K. The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains. SIAM; 2006.
[17] Lai MC, Tseng YH, Huang H. Numerical simulation of moving contact lines with surfactant by immersed boundary method. Commun Comput Phys 2010;8:735–57.
[18] Marichal Y, Chatelain P, Winckelmans G. Immersed interface interpolation schemes for particle-mesh methods. J Comput Phys 2016;326:947–72.
[19] Michelin S, Lauga E. Phoretic self-propulsion at finite Peclet numbers. J Fluid Mech 2014(747):572–604.
[20] Muradoglu M, Tryggvason G. A front-tracking method for computation of interfacial flows with soluble surfactants. J Comput Phys 2008;227:2238–62.
[21] Muradoglu M, Tryggvason G. Simulations of soluble surfactants in 3d multiphase flow. J Comput Phys 2014;274:737–57.
[22] Pawar Y, Stebe KJ. Marangoni effects on drop deformation in an extensional flow: the role of surfactant physical chemistry, I. Insoluble surfactants. Phys Fluids 1996;8:1738–51.
[23] Pan KL, Tseng YH, Chen JC, Huang KL, Wang CH, Lai MC. Controlling droplet bouncing and coalescence with surfactant. J Fluid Mech 2016;799:603–36.
[24] Russel D, Wang ZJ. A cartesian grid method for modeling multiple moving objects in 2d incompressible viscous flow. J Comput Phys 2003;191:177–205.
[25] Trefethen LN. Spectral Methods in MATLAB. Philadelphia: SIAM; 2000.
[26] Teigen KE, Song P, Lowengrub J, Voigt A. A diffuse-interface method for two-phase flows with soluble surfactants. J Comput Phys 2011;230:375–93.
[27] Yariv E, Michelin S. Phoretic self-propulsion at large Peclet numbers. J Fluid Mech 2015(768):R1–R12.
[28] Xu S. A boundary condition capturing immersed interface method for 3d rigid objects in a flow. J Comput Phys 2011;230:7176–90.
[29] Xu JJ, Huang Y, Lai MC, Li Z. A coupled immersed interface and level set method for three-dimensional interfacial flows with insoluble surfactant. Commun Comput Phys 2014;15(2):451–69.
[30] Xu JJ, Shi W, Lai MC. A level-set method for two-phase flows with soluble surfactant. J Comput Phys 2018;353:336–55.