Commun. Comput. Phys.
doi: 10.4208/cicp.241012.310513a

Vol. **15**, No. 2, pp. 451-469
February 2014

A Coupled Immersed Interface and Level Set Method for Three-Dimensional Interfacial Flows with Insoluble Surfactant

Jian-Jun Xu[1,4,*], Yunqing Huang[1,4], Ming-Chih Lai[2] and Zhilin Li[3,5]

[1] *School of Mathematical and Computational Sciences, Xiangtan University, Hunan 410005, China.*

[2] *Department of Applied Mathematics, National Chiao Tung University, Hsinchu 30050, Taiwan.*

[3] *Department of Mathematics, North Carolina State University, Raleigh, NC, 27695, USA.*

[4] *Hunan Key Lab for Computation and Simulation in Science and Engineering, Hunan 410005, China.*

[5] *School of Mathematical Sciences, Nanjing Normal University, Nanjing, China.*

Received 24 October 2012; Accepted (in revised version) 31 May 2013

Available online 10 September 2013

Abstract. In this paper, a numerical method is presented for simulating the 3D interfacial flows with insoluble surfactant. The numerical scheme consists of a 3D immersed interface method (IIM) for solving Stokes equations with jumps across the interface and a 3D level-set method for solving the surfactant convection-diffusion equation along a moving and deforming interface. The 3D IIM Poisson solver modifies the one in the literature by assuming that the jump conditions of the solution and the flux are implicitly given at the grid points in a small neighborhood of the interface. This assumption is convenient in conjunction with the level-set techniques. It allows standard Lagrangian interpolation for quantities at the projection points on the interface. The interface jump relations are re-derived accordingly. A novel rotational procedure is given to generate smooth local coordinate systems and make effective interpolation. Numerical examples demonstrate that the IIM Poisson solver and the Stokes solver achieve second-order accuracy. A 3D drop with insoluble surfactant under shear flow is investigated numerically by studying the influences of different physical parameters on the drop deformation.

AMS subject classifications: 76D07, 65M06

Key words: 3D immersed interface method, level-set method, insoluble surfactant, Stokes interfacial flow, drop deformation.

[*]Corresponding author. *Email addresses:* jjxu21@xtu.edu.cn (J.-J. Xu), huangyq@xtu.edu.cn (Y. Q. Huang), mclai@math.nctu.edu.tw (M.-C. Lai), zhilin@math.ncsu.edu (Z. Li)

©2014 Global-Science Press

1 Introduction

Surfactant is an organic amphiphilic compound which consists of a hydrophilic head and a hydrophobic tail. It tends to adhere to the fluid interface and reduces the surface tension. Surfactant plays an important role in many applications in the industries of food, cosmetics, oil, etc. Due to many important applications and the associated computational challenges for surfactant dynamics, several new numerical methods have been proposed recently based on different interface tracking/capturing techniques, including the arbitrary Lagrangian-Eulerian method [5, 32], the front-tracking or immersed boundary method [10,12–14,19], the volume of fluid (VOF) method [8], the level-set method [28,31], and the diffusive interface method [27], just to name a few.

The traditional (still popular) approach to deal with the jump conditions of flow variables and fluxes across the interface is the continuum surface force (CSF) approach. In the CSF approach, a smooth discrete δ function is used to regularize the Navier-Stokes/Stokes equations by distributing the singular forces into a small neighborhood of the interface. The CSF approach is based on Peskin's immersed boundary (IB) method, see e.g., [23]. The IB method was originally developed in [22] for computing the blood flow in humans' heart. Most of the previously mentioned methods belong to the CSF approach. In this approach, however, the physical jumps of the pressure, and the gradient of the velocity are smeared out in the numerical solution. Generally the CSF type methods can only achieve first order accuracy.

Motivated by improving the accuracy of the IB method, LeVeque and Li [15] proposed the immersed interface method (IIM), in which the numerical schemes at the grid points adjacent to the interface are redesigned to incorporate the jump conditions. Consequently the IIM captures the jumps in a sharp fashion. Numerical evidence and theoretical analysis have shown that the IIM can achieve second order accuracy, see e.g., [2, 4, 6, 17].

As for interfacial flows with surfactant, an approach was proposed in [28], which coupled the IIM as the flow solver and an Eulerian level-set method as the surfactant solver. This approach was applied to the simulation of surface phase separations in [18], and the surfactant-laden drop-drop interactions in [29]. All these works were in 2D.

Despite many works of 2D IIM in the literature(see e.g., [17]), there have been few works of 3D. In [4] a 3D solver of IIM for elliptic interface problems was developed. The formulation of [4] assumes that the jumps for the solution and the flux are explicitly given on the interface, though the interface is implicitly represented by a level-set function. This explicit jump assumption may be inconvenient when in conjunction with the level-set techniques. Some local interface reconstruction is needed when a non-standard surface least square interpolation is used to calculating the surface derivatives of the jumps. In [4] this 3D solver was applied to an inverse problem of shape identification, however, it has not been utilized to simulate a moving and deforming interface in a complex fluid yet.

Geared toward fully 3D simulation of two-phase flows with surfactant, a different version of the 3D IIM Poisson solver is presented in this paper. In this solver we assume that the jumps of the solution are implicitly given by functions defined at the grid points in a small neighborhood of the interface. The interface jump relations and the correction terms in the finite difference scheme in the original 3D IIM Poisson solver in [4] are modified accordingly to make the present 3D Poisson solver second-order accurate.

This 3D Poisson solver is then exploited in the Stokes flow solver. The global surfactant flow solver forms by coupling the Stokes flow solver and the 3D level-set surfactant solver in [30,31].

The rest of this paper is organized as follows. The mathematical formulation is given in Section 2. The numerical method is described in Section 3. Numerical results including accuracy check and simulations of single drop deformation are presented in Section 4. Some concluding remarks are given in Section 5.

2 Mathematical formulation

We consider a neutrally buoyant and contaminated (with insoluble surfactant) liquid drop of radius a suspended in an immiscible fluid where the drop/fluid interface is denoted by Σ. For simplicity, we consider that the density ρ and the viscosity μ of the drop equal to those of the ambient fluid. In the far-field, we assume that the fluid is under a steady flow $\mathbf{u}=\mathbf{u}_\infty$ in the case of shear flow with a shear rate $\dot{\gamma}$. The nondimensionalization procedure is the same as in [31], thus omitted here. We further assume that the Reynolds number Re is small, so that the inertia term can be neglected. Thus, the resulting equations of motion can be described as the dimensionless Stokes equations in a fluid domain Ω as follows.

$$\nabla p=\Delta\mathbf{u}, \tag{2.1}$$

$$\nabla\cdot\mathbf{u}=0. \tag{2.2}$$

Across the interface Σ, the velocity $\mathbf{u}$ must be continuous (no-slip boundary condition), and the normal component of stress must be balanced by the Laplace-Young condition

$$[-pI+(\nabla\mathbf{u}+\nabla\mathbf{u}^T)]\mathbf{n}=\frac{1}{Ca}(\sigma\kappa\mathbf{n}-\nabla_s\sigma), \tag{2.3}$$

where the jump $[\cdot]$ is defined as the difference of the limit of quantity from the outside of the drop and that of the inside. Here, Ca is the capillary number which measures the ratio of the viscous force to surface tension force. σ is the surface tension varied by the surfactant concentration, κ is the mean curvature of the interface, and $\mathbf{n}$ is the interface outward normal vector pointing into the ambient fluid.

Along the interface Σ, the insoluble surfactant is distributed and governed by the surface convection-diffusion equation as

$$f_t+\mathbf{u}\cdot\nabla f-\mathbf{n}\cdot(\nabla\mathbf{u}\cdot\mathbf{n})f=\frac{1}{Pe}\nabla_s^2 f, \tag{2.4}$$

where Pe is the surfactant Peclet number which measures the ratio of the effect of the surface convection to the surface diffusion, and $\nabla_s^2=\nabla_s\cdot\nabla_s$ is the surface Laplacian with the surface gradient $\nabla_s = (I - \mathbf{n}\otimes\mathbf{n})\nabla$. Larger Pe leads to less uniform surfactant concentration, higher f around drop tips, and larger drop deformations. The relation between the surface tension and surfactant concentration is described by Langmuir equation of state (EOS) as

$$\sigma(f) = 1 + E\ln(1 - xf), \tag{2.5}$$

where E is the surfactant elasticity measuring the sensitivity of the surface tension to the surfactant concentration, and x is the initial surfactant coverage. An increase in E increases the drop deformation. Also larger x means more reduction of the surface tension, and so larger drop deformation. For small surfactant coverage, Eq. (2.5) can be approximated by a linear EOS,

$$\sigma(f) = 1 - Exf. \tag{2.6}$$

Note that if the initial surfactant is uniformly distributed, then in dimensionless form $f(\mathbf{X},0) = 1$ on Σ_0.

A level-set function ϕ is used to capture the motion of the interface:

$$\frac{D\phi}{Dt} = \phi_t + \mathbf{u}\cdot\nabla\phi = 0. \tag{2.7}$$

One advantage of the level-set method [21,24] is that geometrical quantities can be easily calculated. Assuming that $\phi < 0$ inside the drop, then

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \quad \kappa = \nabla\cdot\mathbf{n}. \tag{2.8}$$

2.1 Jump conditions of the Stokes equations

In order to apply the immersed interface method in the solution of Stokes equations, we need to know the jump conditions for the pressure p, the velocity $\mathbf{u}$, and their normal derivatives. Let us denote the righthand side of normal component of stress jump in (2.3) by $\mathbf{F}$, then the equation becomes

$$[-pI + (\nabla\mathbf{u} + \nabla\mathbf{u}^T)]\mathbf{n} = \mathbf{F}. \tag{2.9}$$

Suppose $\boldsymbol{\eta}$ and $\boldsymbol{\tau}$ are two orthogonal tangential directions of the interface Σ. Then jump conditions derived in the literature (see, e.g., [7,11]) can be rewritten as following:

$$[p] = \mathbf{F}\cdot\mathbf{n}, \quad \left[\frac{\partial p}{\partial \mathbf{n}}\right] = \frac{\partial(\mathbf{F}\cdot\boldsymbol{\eta})}{\partial\boldsymbol{\eta}} + \frac{\partial(\mathbf{F}\cdot\boldsymbol{\tau})}{\partial\boldsymbol{\tau}}, \tag{2.10}$$

$$[\mathbf{u}] = 0, \quad \left[\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right] = -(\mathbf{n},\boldsymbol{\tau},\boldsymbol{\eta})\begin{pmatrix} 0 \\ \boldsymbol{\tau}^T \\ \boldsymbol{\eta}^T \end{pmatrix}\mathbf{F}. \tag{2.11}$$

The 3D Stokes equations can be rewritten as a system of four Poisson equations for the pressure and velocity. By taking the divergence on Eq. (2.1) and using the incompressibility constraint (2.2), we first obtain the pressure Poisson equation as

$$\Delta p = 0, \tag{2.12}$$

with the interface jump conditions $[p]$ and $[\frac{\partial p}{\partial \mathbf{n}}]$ given in Eq. (2.10). The external boundary condition is chosen as

$$\left.\frac{\partial p}{\partial \mathbf{n}}\right|_{\partial\Omega} = \Delta \mathbf{u}\cdot\mathbf{n}. \tag{2.13}$$

Once the pressure is found, the velocity can be solved by following Poisson equations

$$\Delta \mathbf{u} = \nabla p, \tag{2.14}$$

with the interface jump conditions $[\mathbf{u}]$ and $[\frac{\partial \mathbf{u}}{\partial \mathbf{n}}]$ given in Eq. (2.11). The external boundary condition of $\mathbf{u}$ is given by $\mathbf{u}|_{\partial\Omega} = \mathbf{u}_\infty$.

The righthand side of Eq. (2.14) also has a finite jump across the interface Σ. Notice that, $[\nabla p]\cdot\mathbf{n} = [\frac{\partial p}{\partial \mathbf{n}}]$, $[\nabla p]\cdot\boldsymbol{\tau} = [\frac{\partial p}{\partial \boldsymbol{\tau}}] = \frac{\partial [p]}{\partial \boldsymbol{\tau}}$, $[\nabla p]\cdot\boldsymbol{\eta} = [\frac{\partial p}{\partial \boldsymbol{\eta}}] = \frac{\partial [p]}{\partial \boldsymbol{\eta}}$, so we have

$$[\nabla p] = (\mathbf{n},\boldsymbol{\tau},\boldsymbol{\eta}) \begin{pmatrix} \left[\dfrac{\partial p}{\partial \mathbf{n}}\right] \\ \dfrac{\partial [p]}{\partial \boldsymbol{\tau}} \\ \dfrac{\partial [p]}{\partial \boldsymbol{\eta}} \end{pmatrix}. \tag{2.15}$$

For the Stokes flow with surfactant, the interfacial force $\mathbf{F}$ is given by Eq. (2.3), i.e., $\mathbf{F} = (\sigma\kappa\mathbf{n} - \nabla_s\sigma)/Ca$. Using (2.10) and (2.11), the jump conditions for the pressure and the velocity can be written as

$$[p] = -\frac{\sigma\kappa}{Ca}, \quad \left[\frac{\partial p}{\partial \mathbf{n}}\right] = \frac{\nabla_s^2\sigma}{Ca}, \quad [\mathbf{u}] = 0, \quad \left[\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right] = -\frac{\nabla_s\sigma}{Ca}. \tag{2.16}$$

3 Numerical method

3.1 A 3D IIM Poisson solver

The four Poisson equations for $\mathbf{u}$ and p together with jump conditions are solved by a new version of the 3D IIM Poisson solver originally developed in [4]. The IIM is second order accurate in maximum norm and is briefly described below.

We consider the following Poisson equation

$$\Delta u = f, \quad \text{in } \Omega - \Sigma, \tag{3.1}$$

with jump condition

$$[u]=w(x,y,z)|_{\Sigma}, \quad \left[\frac{\partial u}{\partial \mathbf{n}}\right]=q(x,y,z)|_{\Sigma}, \quad [f]=g(x,y,z)|_{\Sigma}, \tag{3.2}$$

where in the discrete level, the jump functions w,q,g are in a small neighborhood (namely, Γ) of the interface Σ. The implicit jump assumption (3.2) is convenient in conjunction with the level-set techniques. In our Eulerian level-set approach for surfactant flows, the surfactant concentration is extended onto grid points in a small neighborhood Γ of the interface, so does the surface tension. Accordingly, the jumps for the velocity and the pressure in (2.16) are available on these grid points, since they depend on the surface tension and the level-set function.

For simplicity we assume that the grid lengths in all coordinate directions are the same, namely h. Let ϕ be a level-set function whose zero level set is the interface Σ. Assume $\phi<0$ inside the interface. Then we can divide grid points $\mathbf{x}_{ijk}(\equiv(x_i,y_j,z_k))$ into two groups as follows.

Denote

$$f_1\equiv\max\{\phi_{i,j,k},\phi_{i+1,j,k},\phi_{i-1,j,k},\phi_{i,j+1,k},\phi_{i,j-1,k},\phi_{i,j,k+1},\phi_{i,j,k-1}\},$$
$$f_2\equiv\min\{\phi_{i,j,k},\phi_{i+1,j,k},\phi_{i-1,j,k},\phi_{i,j+1,k},\phi_{i,j-1,k},\phi_{i,j,k+1},\phi_{i,j,k-1}\}.$$

If $f_1>0$ and $f_1f_2\leq 0$, then $\mathbf{x}_{ijk}$ is an irregular grid point. Otherwise it is a regular grid point.

If $\mathbf{x}_{ijk}$ is regular, then the standard 7-point discrete Laplacian difference scheme is used, and the local truncation error is $\mathcal{O}(h^2)$. If $\mathbf{x}_{ijk}$ is irregular, a correction term c_{ijk} is added,

$$\frac{u_{i+1,j,k}+u_{i-1,j,k}+u_{i,j+1,k}+u_{i,j-1,k}+u_{i,j,k+1}+u_{i,j,k-1}-6u_{ijk}}{h^2}=f_{ijk}+c_{ijk}. \tag{3.3}$$

c_{ijk} will be constructed so that the local truncation error is $\mathcal{O}(h)$. Roughly speaking, all irregular grids points form a set of co-dimension 1, global second-order accuracy of the numerical solution can be achieved in maximum norm, see e.g., [2,17]. The construction of c_{ijk} consists of the following steps:

- Find the approximate orthogonal projection point $\mathbf{x}^*=\mathbf{x}_{ijk}+\alpha\mathbf{n}$ on Σ by solving the following quadratic equation for α:

$$0=\phi(\mathbf{x}_{ijk}+\alpha\mathbf{n})\approx\phi(\mathbf{x}_{ijk})+\alpha|\nabla\phi|+\frac{\alpha^2}{2}\mathbf{n}^T\mathsf{He}(\phi)\mathbf{n}, \tag{3.4}$$

where $\mathbf{n}$ is the normal of the level set at $\mathbf{x}_{ijk}$, and $\mathsf{He}(\phi)$ is the Hessian matrix at $\mathbf{x}_{ijk}$:

$$\mathsf{He}(\phi)=\begin{pmatrix}\phi_{xx} & \phi_{xy} & \phi_{xz}\\ \phi_{yx} & \phi_{yy} & \phi_{yz}\\ \phi_{zx} & \phi_{zy} & \phi_{zz}\end{pmatrix}.$$

- Select a local coordinate system $\xi-\eta-\tau$ with the origin at $\mathbf{x}^*$.
- Derive the interface jump relations at $\mathbf{x}^*$ for $[u]$,$[u_\xi]$, $[u_\eta]$, $[u_\tau]$, $[u_{\xi\xi}]$, $[u_{\xi\eta}]$, $[u_{\xi\tau}]$, $[u_{\eta\tau}]$, $[u_{\eta\eta}]$,$[u_{\tau\tau}]$.
- Carry out the Taylor expansions for $u(\mathbf{x}_{ijk})$, $u(\mathbf{x}_{i+1,j,k})$, $u(\mathbf{x}_{i-1,j,k})$, $u(\mathbf{x}_{i,j+1,k})$, $u(\mathbf{x}_{i,j-1,k})$, $u(\mathbf{x}_{i,j,k+1})$, $u(\mathbf{x}_{i,j,k-1})$ at $\mathbf{x}^*$ up to second order derivatives in the local coordinates
- Use those interface jump relations to determine c_{ijk} so that the local truncation error of (3.3) is $\mathcal{O}(h)$.

Below we briefly describe the above steps, which follow the lines in [4] with some modifications.

1) The selection of the local coordinate systems at the projection points of the irregular grid points is crucial for the success of the IIM. Denote the set of all the grid points in Γ as Γ_h. The idea is to select coordinate directions $\boldsymbol{\xi}$,$\boldsymbol{\eta}$,$\boldsymbol{\tau}$ at each grid point in Γ_h, then use standard Lagrange quadratic interpolation to get the coordinate directions $\boldsymbol{\xi}^*$,$\boldsymbol{\eta}^*$,$\boldsymbol{\tau}^*$ at each projection point $\mathbf{x}^*$ on the interface Σ. In principle, $\boldsymbol{\xi}$,$\boldsymbol{\eta}$,$\boldsymbol{\tau}$ as grid functions in Γ_h should be smooth in order for the interpolation to be effective.

The normal directions can be easily computed via the level-set function as $\boldsymbol{\xi}=\mathbf{n}=\nabla\phi/|\nabla\phi|$ at all grid points in Γ_h.

The selection of two orthogonal tangential directions $\boldsymbol{\eta}$,$\boldsymbol{\tau}$ is a little more complicated. Arbitrarily choose $\mathbf{x}_0\in\Gamma_h$, let $d_1=\sqrt{\phi_x^2(\mathbf{x}_0)+\phi_y^2(\mathbf{x}_0)}$, and $d_2=\sqrt{\phi_x^2(\mathbf{x}_0)+\phi_z^2(\mathbf{x}_0)}$. Then the first tangential direction at $\mathbf{x}_0$ is obtained as following:

$$\boldsymbol{\eta}_0=\begin{cases}\dfrac{(\phi_y(\mathbf{x}_0),\,-\phi_x(\mathbf{x}_0),\,0)^T}{d_1}, & \text{if } d_1>d_2,\\[2mm] \dfrac{(\phi_z(\mathbf{x}_0),\,0,\,-\phi_x(\mathbf{x}_0))^T}{d_2}, & \text{else.}\end{cases}\tag{3.5}$$

The second tangential direction at $\mathbf{x}_0$ is simply $\boldsymbol{\tau}_0=\boldsymbol{\xi}_0\times\boldsymbol{\eta}_0$, where "$\times$" is the cross product operator.

For any other grid points $\mathbf{x}_1\in\Gamma_h$, the normal direction $\boldsymbol{\xi}_1$ is available already. Let θ be the angle between $\boldsymbol{\xi}_0$ and $\boldsymbol{\xi}_1$. Form $\mathbf{v}\equiv(v_1,v_2,v_3)=(\boldsymbol{\xi}_0\times\boldsymbol{\xi}_1)/|\boldsymbol{\xi}_0\times\boldsymbol{\xi}_1|$. Then the first tangential direction $\boldsymbol{\eta}_1$ at $\mathbf{x}_1$ can be obtained by rotating $\boldsymbol{\eta}_0$ with angle θ about $\mathbf{v}$-axis. Denote $\boldsymbol{\eta}_0\equiv(u_1,u_2,u_3)$, we have the following formula for the first tangential direction $\boldsymbol{\eta}_1$ at $\mathbf{x}_1$ (e.g., [20]):

$$\boldsymbol{\eta}_1\equiv\begin{pmatrix}w_1\\w_2\\w_3\end{pmatrix}=\begin{pmatrix}v_1\boldsymbol{\eta}_0\cdot\mathbf{v}(1-\cos\theta)+u_1\cos\theta+(v_2u_3-v_3u_2)\sin\theta\\ v_2\boldsymbol{\eta}_0\cdot\mathbf{v}(1-\cos\theta)+u_2\cos\theta+(v_3u_1-v_1u_3)\sin\theta\\ v_3\boldsymbol{\eta}_0\cdot\mathbf{v}(1-\cos\theta)+u_3\cos\theta+(v_1u_2-v_2u_1)\sin\theta\end{pmatrix}.\tag{3.6}$$

Again the second tangential direction is obtained by the cross product: $\boldsymbol{\tau}_1=\boldsymbol{\xi}_1\times\boldsymbol{\eta}_1$. The resulting directions $\boldsymbol{\xi}$,$\boldsymbol{\eta}$,$\boldsymbol{\tau}$ as grid functions in Γ_h are smooth due to the rigid rotations, which is good for interpolation. Notice that in [4], the coordinate direction vectors at the involved grid points were all calculated by using formula (3.5). In that way some discontinuity occurs when $d_1=d_2$, which could lead to inaccurate interpolation.

2) Assume that under the local coordinate system $\xi-\eta-\tau$ (with origin at the projection point $\mathbf{x}^*$), the interface has a local parametric representation $\xi=\chi(\eta,\tau)$, then we can derive the following interface jump relations according to (3.2):

$$[u]=w,\quad [u_\xi]=q,\quad [u_\eta]=w_\eta,\quad [u_\tau]=w_\tau, \tag{3.7a}$$

$$[u_{\eta\eta}]=-q\chi_{\eta\eta}+w_\xi\chi_{\eta\eta}+w_{\eta\eta},\quad [u_{\tau\tau}]=-q\chi_{\tau\tau}+w_\xi\chi_{\tau\tau}+w_{\tau\tau}, \tag{3.7b}$$

$$[u_{\eta\tau}]=-q\chi_{\eta\tau}+w_\xi\chi_{\eta\tau}+w_{\eta\tau}, \tag{3.7c}$$

$$[u_{\xi\eta}]=w_\eta\chi_{\eta\eta}+w_\tau\chi_{\eta\tau}+q_\eta,\quad [u_{\xi\tau}]=w_\tau\chi_{\tau\tau}+w_\eta\chi_{\eta\tau}+q_\tau, \tag{3.7d}$$

$$[u_{\xi\xi}]=q(\chi_{\eta\eta}+\chi_{\tau\tau})+g-(w_\xi\chi_{\eta\eta}+w_{\eta\eta})-(w_\xi\chi_{\tau\tau}+w_{\tau\tau}), \tag{3.7e}$$

where $\chi_{\eta\eta},\chi_{\eta\tau},\chi_{\tau\tau}$ are the principal curvatures of the interface. Since $\phi(\chi(\eta,\tau),\eta,\tau)=0$, these principal curvatures can be written via the level-set function as following:

$$\chi_{\eta\eta}=-\frac{\phi_{\eta\eta}}{\phi_\xi},\quad \chi_{\eta\tau}=-\frac{\phi_{\eta\tau}}{\phi_\xi},\quad \chi_{\tau\tau}=-\frac{\phi_{\tau\tau}}{\phi_\xi}. \tag{3.8}$$

3) The correction term c_{ijk} in [4] is also modified according to (3.7). The details are quite tedious and omitted here.

All the interface quantities involved in c_{ijk}, such as the surface derivatives of jumps and the principal curvatures in (3.7), are obtained by using standard Lagrange quadratic interpolation from the corresponding grid functions in Γ_h to the projection point $\mathbf{x}^*$s. The corresponding grid functions are computed using the standard central finite difference schemes.

3.2 Sketch of the global algorithm

In this subsection, we describe the algorithm for computing the global system of the Stokes flow with surfactant.

Given $\mathbf{u}^n,\mathbf{u}^{n-1},p^n,\phi^n,\phi^{n-1},f^n,f^{n-1}$, our algorithm to compute the quantities at the $(n+1)$-th time level consists of the following steps:

Step 1. The four Poisson equations for the pressure and velocity with known jump conditions are solved using the modified IIM Poisson solver described in previous subsection. For the pressure, we have

$$\Delta p^{n+1}=0, \tag{3.9}$$

with the jump conditions:

$$[p^{n+1}]=-\left(\frac{1}{Ca}\sigma\kappa\right)^n,\quad \left[\frac{\partial p^{n+1}}{\partial \mathbf{n}}\right]=\left(\frac{1}{Ca}\nabla_s^2\sigma\right)^n, \tag{3.10}$$

and the Neumann boundary conditions on $\partial\Omega$:

$$\frac{\partial p^{n+1}}{\partial \mathbf{n}} = (\Delta \mathbf{u}\cdot\mathbf{n})^n. \tag{3.11}$$

The velocity $\mathbf{u}^{n+1}$ is then obtained by solving

$$\Delta \mathbf{u}^{n+1} = \nabla p^{n+1}, \tag{3.12}$$

together with jump conditions:

$$[\mathbf{u}^{n+1}] = 0, \quad \left[\frac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}}\right] = -\left(\frac{1}{Ca}\nabla_s \sigma\right)^n, \tag{3.13}$$

and

$$\left[\nabla p^{n+1}\right] = (\mathbf{n},\boldsymbol{\tau},\boldsymbol{\eta})^n \begin{pmatrix} \left[\frac{\partial p}{\partial \mathbf{n}}\right] \\ \frac{\partial [p]}{\partial \boldsymbol{\tau}} \\ \frac{\partial [p]}{\partial \boldsymbol{\eta}} \end{pmatrix}^n, \tag{3.14}$$

and the Dirichlet boundary condition on $\partial\Omega$:

$$\mathbf{u}^{n+1} = \mathbf{u}_\infty, \quad \text{on } \partial\Omega. \tag{3.15}$$

Caution needs to be taken in calculating ∇p at an irregular grid point $\mathbf{x}_{ijk}$ due to the jump discontinuity. If $\mathbf{x}_{ijk}$ and $\mathbf{x}_{i-1,j,k}$ (or $\mathbf{x}_{i+1,j,k}$) are on the same side of the interface, the standard one-sided first order forward (or backward) finite difference scheme is used. Otherwise the jump conditions at the projection point $\mathbf{x}^* = (x^*,y^*,z^*)$ is used to construct a first order approximation of p_x as follows.

We choose i_0 such that

$$|x^* - x_{i_0}| = \min\{|x^* - x_{i-1}|, |x^* - x_{i+1}|\}, \tag{3.16}$$

and then use the following formula to compute p_x,

$$(p_x)_{ijk} = \begin{cases} \dfrac{p_{ijk} - p_{i_0jk}}{x_i - x_{i_0}} + \dfrac{[p] + [p_x](x_{i_0} - x^*) + [p_y](y_j - y^*) + [p_z](z_k - z^*)}{x_i - x_{i_0}}, & \text{if } \phi_{ijk} \le 0, \\ \dfrac{p_{ijk} - p_{i_0jk}}{x_i - x_{i_0}} - \dfrac{[p] + [p_x](x_{i_0} - x^*) + [p_y](y_j - y^*) + [p_z](z_k - z^*)}{x_i - x_{i_0}}, & \text{otherwise.} \end{cases} \tag{3.17}$$

The p_y, p_z are approximated using the similar approach.

Step 2. After the velocity $\mathbf{u}^{n+1}$ is computed, it is used to evolve the level-set function ϕ according to (2.7). The level-set function is re-initialized by solving the following Hamilton-Jacobi equation:

$$\begin{cases} \phi_\tau + S(\phi_0)(|\nabla\phi| - 1) = 0, \\ \phi(\mathbf{x},0) = \phi_0(\mathbf{x}), \end{cases} \tag{3.18}$$

where ϕ_0 is the level-set function before the re-initialization, τ is a pseudo-time and $S(x)$ is the sign function of x defined as

$$S(x)=\begin{cases} -1, & \text{if } x<0, \\ 0, & \text{if } x=0, \\ 1, & \text{if } x>0. \end{cases} \tag{3.19}$$

Step 3. The surfactant concentration is evolved by a level-set Eulerian approach in [30,31].

$$\begin{aligned} \frac{f^{n+1}-f^n}{\Delta t}=&\frac{1}{2Pe}\left(\nabla^2 f^{n+1}+\nabla^2 f^n\right) \\ &+\frac{3}{2}\left[-\frac{1}{Pe}\left(\kappa\frac{\partial f}{\partial \mathbf{n}}+\frac{\partial^2 f}{\partial \mathbf{n}^2}\right)-\mathbf{u}\cdot\nabla f+\mathbf{n}\cdot(\nabla\mathbf{un})f\right]^n \\ &-\frac{1}{2}\left[-\frac{1}{Pe}\left(\kappa\frac{\partial f}{\partial \mathbf{n}}+\frac{\partial^2 f}{\partial \mathbf{n}^2}\right)-\mathbf{u}\cdot\nabla f+\mathbf{n}\cdot(\nabla\mathbf{un})f\right]^{n-1}. \end{aligned} \tag{3.20}$$

The quantity f^{n+1} is extended to a small neighborhood of the interface by solving the following Hamilton-Jacobi equation:

$$\begin{cases} f_\tau+S(\phi)\mathbf{n}\cdot\nabla f=0, \\ f(\mathbf{x},0)=f^{n+1}(\mathbf{x}), \end{cases} \tag{3.21}$$

where $S(\phi)$ is the same sign function of ϕ as in the re-initialization process. The extended f is reset to be f^{n+1}, which is then used for updating σ.

We use the standard third-order upwinding WENO [25] and TVD RK [9] schemes for the space and time discretizations for the level-set convection equation, its re-initialization process, and the extension of the surfactant concentration, respectively. We use the standard central finite difference schemes to discretize the diffusion term and the WENO scheme for the convection term for (3.20). Also the local level-set technique is used, that is, the evolutions of ϕ,f are done in small tubes around the interface. A stability analysis of (3.20) is given in [30].

In the discrete level, the velocity obtained by the IIM is only approximately divergence free. This causes a small error in the volume conservation in each time step. The error can be accumulated and can lead to inaccurate results for long time computation. As done in 2D case in [28], a small correction $\alpha\mathbf{n}$ is added to the velocity of the interface at each time step. α is chosen such that

$$\int_{\Omega_1}\nabla\cdot(\mathbf{u}+\alpha\mathbf{n})d\Omega=\int_\Sigma(\mathbf{u}+\alpha\mathbf{n})\cdot\mathbf{n}ds=0, \tag{3.22}$$

where Ω_1 is the droplet domain, $\mathbf{u}$ is the discrete velocity field obtained by the IIM. Thus

$$\alpha=\frac{-\int_\Sigma\mathbf{u}\cdot\mathbf{n}ds}{\int_\Sigma ds}=\frac{-\int_\Omega\mathbf{u}\cdot\mathbf{n}\delta(\phi)d\mathbf{x}}{\int_\Omega\delta(\phi)d\mathbf{x}}. \tag{3.23}$$

The modified velocity $\tilde{\mathbf{u}}=\mathbf{u}+\alpha\mathbf{n}$ is then used in the evolution of the level-set function. We also need to re-scale the surfactant concentration at each time step by multiplying a factor β to enforce the surfactant conservation. Suppose initially total surfactant mass is m_0. Then β is determined by the following equation:

$$\int_\Sigma \beta f ds = m_0, \tag{3.24}$$

thus

$$\beta=\frac{m_0}{\int_\Sigma f ds}=\frac{m_0}{\int_\Omega f\delta(\phi)d\mathbf{x}}. \tag{3.25}$$

The above integrals are computed by using the trapezoidal rule and a discrete δ-function. The modified surfactant concentration is $\tilde{f}=\beta f$, and it is used for the evolution in next time step. These correction techniques are also frequently used in the other methods such as the boundary integral method.

4 Numerical results

4.1 Accuracy check of the 3D IIM Poisson solver

We consider the Poisson equation (3.1), $\Omega=[-1,1]^3$, Σ is the zero level-set of $\phi=r-0.5$, $r=\sqrt{x^2+y^2+z^2}$. Consider two different expressions of the jump functions.

Case 1. The jump functions are given as

$$w=e^{-r^2}-r^2+1, \quad q=-2r(e^{-r^2}+1), \quad g=(-6+4r^2)e^{-r^2}-6.$$

Case 2. The jump functions are given by the formulas in case 1 but with fixed $r=0.5$.

Notice that the jump functions are only needed in a small tube around the interface with a width of a few grid steps required by the Lagrangian interpolation. The corresponding jump functions in two cases are equal only at the interface. This illustrates the flexibility in obtaining such jump functions, following the spirit of the level-set extension technique.

The exact solution is given by

$$u=\begin{cases} e^{-r^2}, & \text{if } \phi>0, \\ r^2-1, & \text{if } \phi\leq 0, \end{cases}$$

The right hand side f is determined accordingly. A mesh refinement result is shown in Table 1. In both cases, the second order accuracy is achieved.

Table 1: A mesh refinement study for the 3D Poisson solver, where u_h is the numerical solution.

	Case 1		Case 2	
h	$\|u-u_h\|_\infty$	rate	$\|u-u_h\|_\infty$	rate
0.05	$7.02E-4$		$5.47E-4$	
0.025	$1.72E-4$	2.29	$1.40E-4$	1.95
0.0125	$4.27E-5$	2.01	$3.49E-5$	1.88

4.2 Accuracy check of the 3D IIM Stokes solver

We consider the Stokes system (2.1)-(2.2) with an interface in which the velocity and pressure are exactly given by

$$\begin{cases} u=y^3+z^3,\ v=x^3+z^3,\ w=x^3+y^3, & \text{if } \phi\le 0,\\ u=2(y^3+z^3),\ v=2(x^3+z^3),\ w=2(x^3+y^3), & \text{otherwise,}\end{cases} \tag{4.1}$$

and

$$p=\begin{cases} 6(xy+yz+xz), & \text{if } \phi\le 0,\\ 12(xy+yz+xz), & \text{otherwise,}\end{cases} \tag{4.2}$$

where $\phi=\sqrt{x^2+y^2+z^2}-0.5$. Thus, the interface is a spherical surface with radius 0.5. The computational domain is a unit cube $\Omega=(-1,1)^3$ with a uniform mesh h is used in all three coordinates. The surface force $\mathbf{F}$ and the jump functions for $[p]$, $[\frac{\partial p}{\partial \mathbf{n}}]$, $[\frac{\partial \mathbf{u}}{\partial \mathbf{n}}]$ can be derived analytically from the exact solution. The exact Dirichlet and Neumann boundary conditions are used for the velocity and pressure at $\partial\Omega$, respectively.

In Table 2, we show a grid refinement analysis in maximum norm. The results indicate second-order accuracy for both the pressure and the velocity components. The CSF methods, such as the immersed boundary method (e.g., [3]), can only achieve first-order accuracy.

Table 2: A mesh refinement study for the 3D Stokes solver.

h	$\|p_h-p\|_\infty$	rate	$\|u-u_h\|_\infty+\|v-v_h\|_\infty+\|w-w_h\|_\infty$	rate
1/15	$6.67E-4$		$3.27E-3$	
1/30	$1.44E-4$	2.21	$8.35E-4$	1.97
1/60	$3.36E-5$	2.10	$2.04E-4$	2.03

4.3 Influences of parameters on drop deformation

In the rest of the simulations, we exam the influences of parameters on a single drop deformation. The surfactant elasticity E and the surfactant coverage x have similar effects on surface tension, the effect of E is omitted here. We consider a simple shear with $\mathbf{u}_\infty=$

$(y,0,0)^T$. The computational domain is $\Omega=[-1.5,1.5]\times[-1.2,1.2]\times[-1.2,1.2]$, the mesh size is $150\times120\times120$. Initially the spherical drop is centered at the origin with radius being 0.5. The initial surfactant concentration is $f=1$. The nonlinear EOS is used. The deformation of the drop can be characterized by the quantity $D(t)=(L(t)-B(t))/(L(t)+B(t))$, where t is time, $L(t),B(t)$ are respectively the longest and the shortest axes of the evolutionary drop.

4.3.1 Influence of the capillary number

We study the surfactant laden drop with varying capillary numbers, $Ca=0.15,0.3,0.5$, with other parameters fixed as $x=0.3$, $Pe=10$, $E=0.2$. An increase in Ca reduces the surface tension force, and thus increases the drop deformation, as shown in Fig. 1. There is a critical capillary number $\tilde{Ca}$ beyond which the drop keeps stretching and eventually breaks up [26]. The drop reaches a steady state when $Ca=0.15,0.3$. There is no steady state when $Ca=0.5$, in which case the computation stops when a drop tip reaches the boundary of the computational domain.

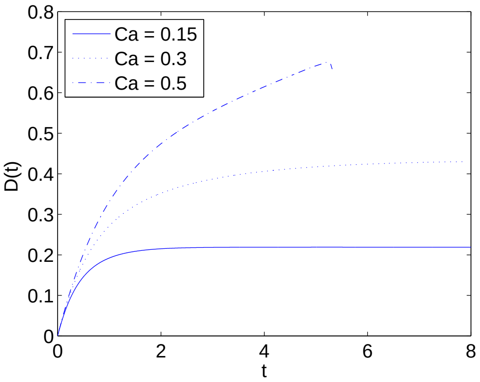

Figure 1: Plot of $D(t)$ versus time t. $D(t)$ is increasing in Ca. The steady state is reached for small $Ca(=0.15,0.3)$, but there is no steady state for large $Ca(=0.5)$.

The correction parameters α and β for the conservations of the volume and surfactant mass together with the relative volume loss are plotted in Fig. 2 for the case $Ca=0.3$. The results are quite acceptable.

The jumps of the flow variables are captured sharply in our method. For example, Fig. 3 shows the pressure distribution on the cross section of the $xy-$plane at $t=8$. There is a clear jump in the pressure across the interface.

4.3.2 Influence of the surfactant coverage

Now we study the effect of the surfacant coverage x with varying values 0,0.3,0.7,0.9. Note that $x=0$ corresponds to clean case. We consider two cases $Ca=0.15$, in which the drop reaches the steady state, and $Ca=0.5$ in which there is no steady state. The other

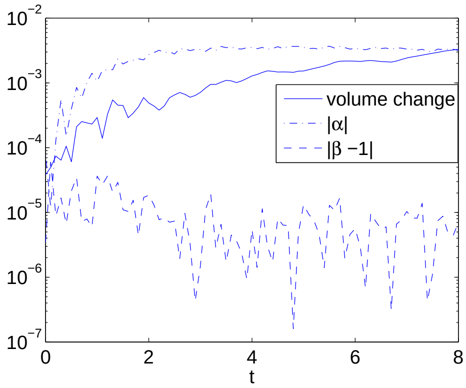

Figure 2: Relative volume loss during the time evolution is less than 5×10^{-3}, so is the absolute value of the parameter α for the velocity correction. The absolute value of the parameter β for the surfactant mass conservations is within a distance of 5×10^{-5} from 1.

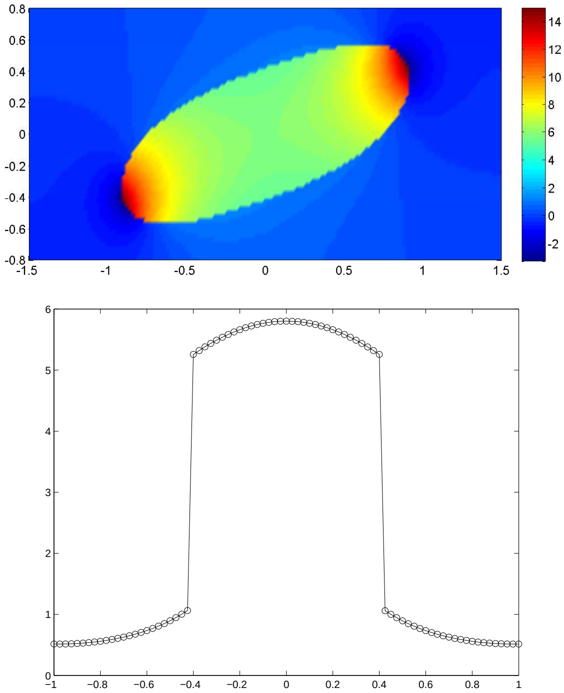

Figure 3: The pressure distributions on xy-plane (top), and along the y-axis (bottom) at time $t=8$. The jump $[p]$ is enforced sharply in our method. The parameters in the simulation are $Ca=0.3$, $x=0.3$, $E=0.2$, $Pe=10$.

parameters are fixed as $E=0.2$ and $Pe=10$. An increase in x reduces the surface tension, and thus increases the drop deformation, as shown in Fig. 4.

The drop morphologies together with surfactant concentrations at the steady states are plotted in Fig. 5, when $Ca=0.15$. Clean case is also included for comparison purpose.

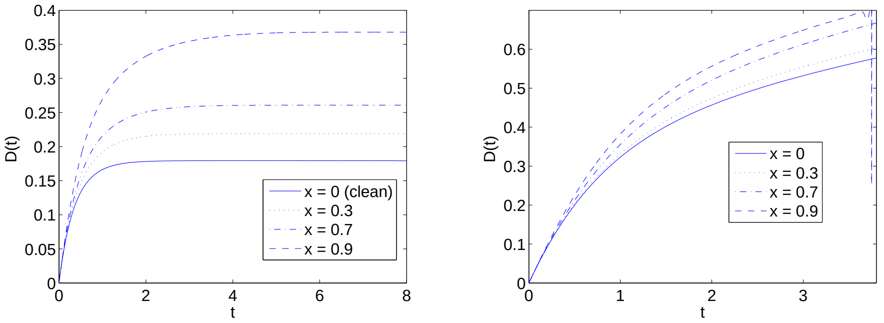

Figure 4: Plots of $D(t)$ versus time t for different values of x. $D(t)$ is increasing in x, as shown in the left plot with $Ca=0.15$, and in the right plot with $Ca=0.5$.

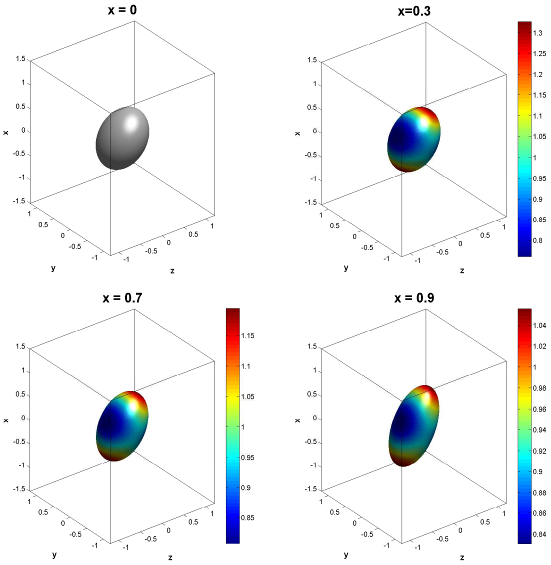

Figure 5: Drop morphologies together with the contour plots for surfactant concentration at the steady states ($t=8$) for different values of x, $Ca=0.15$.

The transient states of the drop morphologies together with surfactant concentration are plotted in Fig. 6, when $Ca=0.5$. Clean case is included for comparison purpose. It is observed that surfactant is swept to the drop tips, and as the drop deforms its concentration gets diluted. With high surfactant coverage, the drop deformation is much faster than the clean case.

4.4 Influence of the Peclet number

Lastly we study the surfactant-laden drop with varying Peclet numbers $Pe=0.1,10,100$. The other parameters are fixed as $Ca=0.3$, $E=0.2$, $x=0.3$. Larger Pe leads to less diffusion,

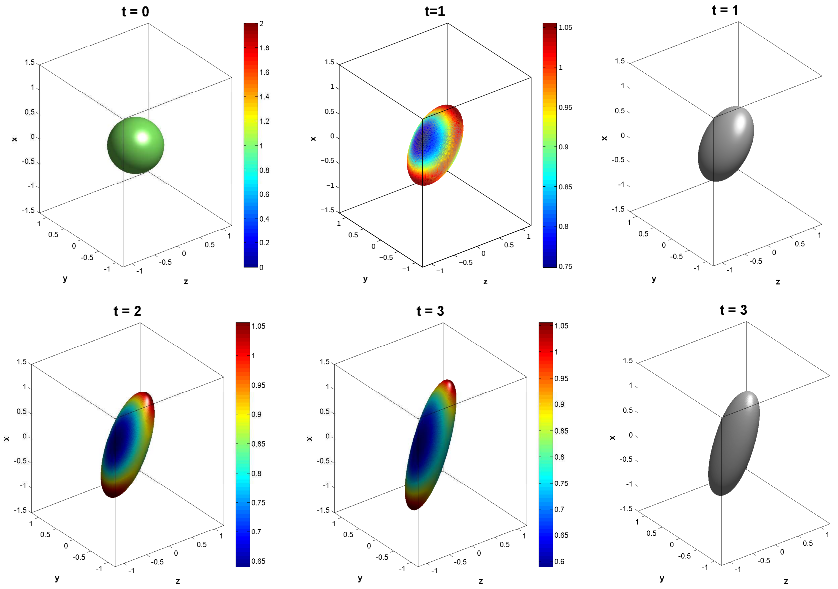

Figure 6: Transient states of drop morphologies and contour plots for surfactant concentration (left two columns, $Ca=0.5$, $x=0.9$) and the morphologies of clean drop (right column, $Ca=0.5$) at different times.

so more surfactant is accumulated at the drop tips, leading to larger drop deformation as shown in Fig. 7.

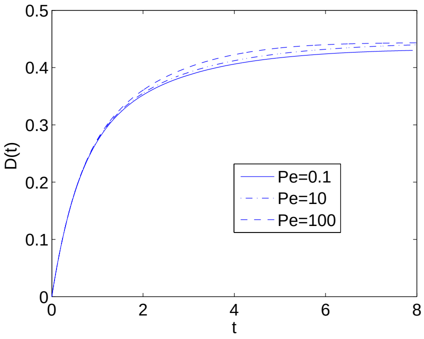

Figure 7: Plot of $D(t)$ versus time t. $D(t)$ is increasing in Pe.

These simulation results are qualitatively in good agreement the previous results in the literature, see e.g., [16,26].

5 Conclusions

A method has been presented for computing interfacial flows with insoluble surfactant in 3D. The method couples a 3D IIM Stokes flow solver and a 3D level-set method for surfactant convection-diffusion equation along an evolving surface. The IIM maintains the jump conditions of flow variables in a sharp fashion, which in principle is more accurate than the CSF type methods. The IIM Poisson solver modifies the original one in [4] under the assumption that the jumps are implicitly captured by grid functions in a small neighborhood of the interface. This assumption is well-suited to the level set techniques. It allows standard Lagrange quadratic interpolation in calculating surface quantities, instead of the surface least square interpolation in [4], leading to relatively easier implementation. A novel rotational procedure was proposed in selecting smooth local coordinate systems, which is good for the interpolation. Numerical examples show that the modified 3D IIM Poisson solver and Stokes solver are indeed second-order accurate. Numerical simulations on single drop deformation are qualitatively in agreement with the previous studies in the literature. Our current method is restricted to Stokes interfacial flow with the same viscosity inside and outside the drop. For Stokes or Navier-Stokes flows with discontinuous viscosity and/or density, explicit jumps of flow variables are not available. The augmented approach in 2D in the literature (e.g., [17]) can be extended to 3D, which is currently under investigation. Due to the limit of computer storage, our current simulations are limited to single drop deformation. Adaptive mesh refinement and/or parallel method in 3D should be developed in order to simulate more complicated flows with surfactant, for example, multi-drop interactions and drop break-up.

Acknowledgments

J. Xu acknowledges partial supports by Hunan Provincial Education Department (10C1264), Xiangtan Univ. (10QDZ45), and Hunan NSFC (10JJ70). Y. Huang was supported in part by NSFC key project 11031006. M.-C. Lai was supported in part by National Science Council of Taiwan under grant NSC98-2115-M-009-014-MY3 and NCTS. Z. Li was supported in part by the US ARO grant 550694-MA, the AFSOR grant FA9550-09-1-0520, the US NSF grant DMS-0911434, the NIH grant 096195-01, and CNSF-11071123.

References

[1] J. Adams, P. Swarztrauber, and R. Sweet, Fishpack: Efficient Fortran subprograms for the solution of separable elliptic partial differential equations, `http://www.netlib.org/fishpack/`.

[2] J. T. Beale, and A. T. Layton, On the accuracy of finite differencec methods for elliptic problems with interfaces, Commun. Appl. Math. Comput. Sci., 1(2006), 91-119.

[3] K.Y. Chen, K.-A. Feng, Y. Kim and M.-C. Lai, A note on pressure accuracy in immersed boundary method for Stokes flow, J. Comput. Phys., 230(2011), 4377-4383.

[4] S. Deng, K. Ito, and Z. Li, Three dimensional elliptic solvers for interface problems and applications, J. Comput. Phys. , 184(2003), 215-243.

[5] S. Ganesan and L. Tobiska, A coupled arbitrary Lagrangian-Eulerian and Lagrangian method for computation of free surface flows with insoluble surfactants, J. Comput. Phys., 228(2009), 2859-2873.

[6] H. Huang, and Z. Li, Convergence analysis of the immersed interface method, IMA J. Numer. Anal., 19(1999), 583-608.

[7] K. Ito, Z. Li, and X. Wan, Pressure jump conditions for Stokes equations with discontinuous viscosity in 2D and 3D, Methods and Appl. Anal., 19(2006), 229-234.

[8] A.J. James and J. Lowengrub, A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactant, J. Comput. Phys., 201(2004), 685-722.

[9] G.-S. Jiang and D. Peng, Weighted ENO schemes for Hamilton-jacobi equations, SIAM J. Sci. Comput., 21(2000), 2126.

[10] S.M. Khatri and A.K. Tornberg, A numerical method for two-phase flows with insoluble surfactants, Comput. Fluids, 49(2011), 150-165.

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

[12] M.-C. Lai, Y.-H. Tseng and H. Huang, An immersed boundary method for interfacial flows with insoluble surfactant, J. Comput. Phys., 227(2008), 7279.

[13] M.-C. Lai, Y.-H. Tseng and H. Huang, Numerical simulation of moving contact lines with surfactant by immersed boundary method, Communications in Computational Physics, 8(2010), 735-757.

[14] M.-C. Lai, C.-Y. Huang and Y.-M. Huang, Simulating the axisymmetric interfacial flows with insoluble surfactant by immersed boundary method, International Journal of Numerical Analysis and Modeling, 8(2011), 105-117.

[15] LeVeque R. J., and Li Z. The immersed interface method for elliptic equation with discontinuous coefficients and singular sources, SIMA J. Numer. Anal., 31(1994), 1019-1044

[16] X. Li and C. Pozrikidis, Effect of surfactants on drop deformation and on the rheology of dilute emulsion in Stokes flow, J. Fluid Mech., 385(1999), 79-99.

[17] Z. Li, and K. Ito, The immersed interface method: Numerical for PDEs involving interfaces and irregular domains, SIAM frontiers in applied mathematics, SIAM, Philadelphia, USA, 2006.

[18] J. Lowengrub, J. Xu, and A. Voigt, Surface phase separation and flow in a simple model of multicomponent drops and vesicles, Fluid Dyn. Mater. Proc., 3(2007), 1-19.

[19] M. Muradoglu and G. Tryggvason, A front-tracking method for the computation of interfacial flows with soluble surfactants, J. Comput. Phys., 227(2008), 2238.

[20] G. Murry, Rotation about an arbitrary axis in 3 dimensions,
`http://inside.mines.edu/∼gmurray/ArbitraryAxisRotation/`

[21] S. Osher and R.P. Fedkiw, Level set methods: An overview and some recent results, J. Comput. Phys., 169(2001), 463.

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

[23] C. Peskin, The immersed boundary methods, Acta Numerica,11(2002), 479-517.

[24] J.A. Sethian and P. Smereka, Level set methods for fluid interfaces, Ann. Rev. Fluid Mech., 35(2003), 341.

[25] C.-W. Shu, Total-variation-diminishing time discretization, SIAM J. Sci. Stat. Comput., 9(1988), 1073.

[26] H.A. Stone and L.G. Leal, The effects of surfactants on drop deformation and breakup, J.

Fluid Mech., 220(1990), 161-186.

[27] K.E. Teigen, P. Song, J. Lowengrub and A. Voigt, A diffusive-interface method for two-phase flows with soluble surfactants, J. Comput. Phys., 230(2011), 375-393.

[28] J. Xu, Z. Li, J. Lowengrub, and H. Zhao, A level set method for solving interfactial flows with surfactant, J. Comput. Phys., 212(2006), 590-616.

[29] J. Xu, Z. Li, J. Lowengrub, H. Zhao, Numerical study of surfactant-laden drop-drop interactions, Commun. Comput. Phys., 10(2011), 453-473.

[30] J. Xu, H.-Z. Yuan and Y.-Q. Huang, A three dimensional level-set method for solving convection-diffusion equations along moving interfaces (in Chinese). Sci Sin Math, 42(2012), 445-454

[31] J. Xu, Y. Yang, and J. Lowengrub. A level-set continuum method for two-phase flows with insoluble surfactants, J. Comput. Phys., 231(2012), 5897-5909.

[32] X. Yang and A. J. James, An arbitrary Lagrangian-Eulerian (ALE) method for interfacial flows with insoluble surfactants, Fluid Dyn. Mater. Proc., 3(2007), 65-96.