Journal of Computational Physics 353 (2018) 336–355

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A level-set method for two-phase flows with soluble surfactant

Jian-Jun Xu [a], Weidong Shi [a,b], Ming-Chih Lai [c]

[a] Chongqing Key Lab of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, 400714, China
[b] School of Mathematical and Computational Sciences, Xiangtan University, Xiangtan, Hunan, 411105, China
[c] Department of Applied Mathematics, National Chiao Tung University, Hsinchu 30010, Taiwan

ARTICLE INFO

Article history:
Received 5 May 2017
Received in revised form 7 October 2017
Accepted 14 October 2017
Available online 19 October 2017

Keywords:
Soluble surfactant
Multi-phase flow
Level-set method
Diffusive domain method
Surfactant mass conservation
Topological change

ABSTRACT

A level-set method is presented for solving two-phase flows with soluble surfactant. The Navier–Stokes equations are solved along with the bulk surfactant and the interfacial surfactant equations. In particular, the convection–diffusion equation for the bulk surfactant on the irregular moving domain is solved by using a level-set based diffusive-domain method. A conservation law for the total surfactant mass is derived, and a re-scaling procedure for the surfactant concentrations is proposed to compensate for the surfactant mass loss due to numerical diffusion. The whole numerical algorithm is easy for implementation. Several numerical simulations in 2D and 3D show the effects of surfactant solubility on drop dynamics under shear flow.

1. Introduction

Surfactant molecules consist of a hydrophilic head and a hydrophobic tail thus typically like to stay at the fluid interface. Surfactant can be adsorbed from the bulk fluid onto the interface and desorbed from the interface into the bulk fluid. Surfactant can have a significant effect on the fluid dynamics of two-phase flows by reducing the surface tension. The non-uniform distribution of surfactant concentration also gives rise to the surface tension gradient, i.e., the so-called Marangoni effect. Surfactants are widely used in many important scientific and engineering applications. For example, they can be used to manipulate drops and bubbles in micro-channels [32], to synthesize micro or sub-micron size mono-dispersed drops and bubbles in micro-fluidic applications [3], to enhance oil recovery [20], and to help to cure pulmonary disease [11].

Many numerical methods were developed for simulations of two-phase flows with insoluble surfactant, including the front tracking method, the volume-of-fluid method, the level-set method, etc. Simulating the effect of soluble surfactant is more challenging due to the surfactant transport between the interface and the bulk fluid. Also the bulk surfactant concentration equation is defined on an evolving domain, which poses another numerical difficulty for the Cartesian grid method. If a body-fitted method is used, mesh generation may be difficult and time-consuming in 3D. In addition, an important issue is to resolve the problem of the surfactant mass loss due to the numerical diffusion. Small numerical error in each time step could be accumulated and could lead to large error in long time simulation. In the case of insoluble surfactant, total surfactant mass is conserved since surfactant does not move away from the interface. One can simply

E-mail address: xujianjun@cigit.ac.cn (J.-J. Xu).

https://doi.org/10.1016/j.jcp.2017.10.019
0021-9991/

re-scale the surfactant concentration to enforce the mass conservation. For soluble surfactant case, total surfactant mass in the computational domain may not be conserved due to the outflow boundary conditions.

Several numerical methods were developed for computing two-phase/interfacial flows with soluble surfactant, including the front-tracking method [19,42,21,4,7,12,22], and the phase-field method [18,34]. The front-tracking method is accurate but may be difficult to handle interfacial topological changes. The phase-field method, capable of capturing the topological change via an appropriate free energy functional, has become quite popular. In the phase-field method, however, it may be difficult to choose an appropriate interface thickness and the corresponding mesh width to resolve the interfacial region. Generally speaking, the mesh width needs to be chosen much smaller than the interface thickness in order to achieve numerical stability in evolving the reaction-diffusion equations for the phase-field function [6]. Very recently, a level-set approach was proposed in [5] for simulating two-phase flows driven by soluble surfactant. The level-set method [28] captures the interface by solving Hamilton–Jacobi equation, including the auxiliary re-initialization equation [33]. The advantages of the level-set method are relatively easy implementation and the capability of handling interfacial topological changes.

Previously we studied the level-set methods for computing two phase/interfacial flows with insoluble surfactant in [38,40,41]. In this work, we present a level-set method for computing two-phase flows with soluble surfactant. The convection–diffusion equation for bulk surfactant concentration with inner Robin type boundary condition needs to be solved in evolving irregular domains. Recently, several Cartesian grid methods were proposed for solving PDEs on irregular domains, in which a level-set function was used for representing the irregular boundary. The immersed interface method [16] may achieve second-order accuracy by embedding the elliptic equation into a larger regular domain, and introducing an augmented variable for jump condition across the physical boundary to account for the boundary condition. Standard finite-difference approximations for the derivatives at the irregular grid points adjacent to the physical boundary must be modified by adding some correction terms which involve the curvature of the physical boundary, and the values and derivatives of the jumps. In [10], a simple finite difference method was proposed for solving elliptic problems with the Dirichlet boundary condition on irregular domain. In the method, the standard finite difference scheme at the irregular grid points is modified by introducing a ghost value, which is determined by extrapolation utilizing a level-set function. For PDEs on irregular domains with Robin boundary conditions, a finite volume method was developed in [25,26]. In that method, special integration techniques utilizing a level-set function are used in the finite volume discretization of the PDE at the grid cells intersecting the boundary. Numerical experiments demonstrated that the method achieves second-order accuracy for the diffusion equations, and first-order accuracy for the Stephan problem.

The diffusive-domain (DD) method for solving PDEs on irregular domains (see [15,14] and the references therein) has become popular in the community of phase-field method due to its simplicity. In this method, the irregular domain is embedded into a larger regular domain and the original PDE is reformulated by including the original boundary condition via the indication function of the physical domain. First-order convergence of the DD method was proven in [9]. The phase-field based DD method has been used in various applications, including cell biology [31], wave propagation in the heart [8], tissue engineering [13], micro-fluidics [2], and two-phase flows with soluble surfactant [34]. In this work we present a level-set based DD method for solving the bulk surfactant equation.

It appears that the issue of total surfactant mass conservation has not been well addressed in literature. The total surfactant mass may not be conserved if surfactant is transported out of the computational domain. In this work, a conservation law for the total surfactant mass is derived. We simply re-scale the bulk and interfacial surfactant concentrations to compensate for the surfactant mass loss caused by numerical diffusion.

The rest of paper is organized as follows. In Section 2, the complete governing equations of the interfacial flow with soluble surfactant are introduced and the non-dimensionalization of those equations are performed. We then derive a conservation law for the total surfactant mass and rewrite the bulk surfactant equation in level-set diffusive domain formulation. The numerical methods for solving the whole governing equations are described in Section 3 in details. Numerical results are shown in Section 4 while some concluding remarks are given in Section 5.

2. Mathematical formulation

We consider the dynamics of two immiscible incompressible fluids separated by the interface Γ and occupied by the drop Ω_1 and the bulk fluid Ω_2, respectively. In addition, there is a soluble surfactant existing along the interface Γ and in the bulk fluid Ω_2 but not in the drop Ω_1. The fluid interface $\Gamma = \partial\Omega_1$ is represented by the zero level set of a function ϕ as configured in Fig. 1. For simplicity, we assume that two fluids have the same density and viscosity, and the gravitational force is neglected. (A more general two-phase flow with unmatched density/viscosity can be treated by the method described in [40].) The governing equations consist of the usual Navier–Stokes in $\Omega = \Omega_1 \cup \Omega_2$, the interfacial surfactant concentration equation on Γ, and the bulk surfactant equation in Ω_2. The coupling between the fluid equations and surfactant equations comes from the surface tension which depends on the interfacial surfactant concentration. The mathematical formulation is described as follows.

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) + \nabla p = \mu\nabla^2\mathbf{u} + \mathbf{F} \quad \text{in} \quad \Omega, \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0, \quad \text{in} \quad \Omega, \tag{2}$$

$$\mathbf{u}|_{\partial\Omega} = \mathbf{u}_b, \tag{3}$$

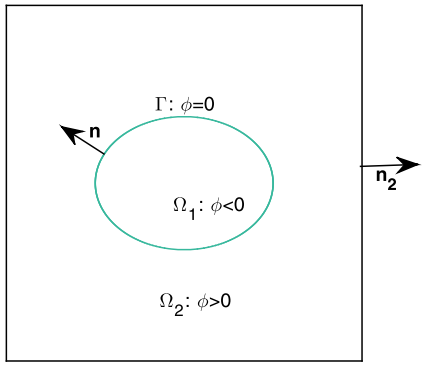


Fig. 1. A configuration of the two-phase flow with soluble surfactant: ϕ is the level-set function, the outward normal vector $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ points into the bulk fluid while $\mathbf{n}_2$ is the outward normal of the computational domain. The surfactant exists on the interface Γ and in the bulk fluid Ω_2 but not in the drop Ω_1.

where ρ is the fluid density, $\mathbf{u}$ the velocity field, p the pressure, μ the viscosity and $\mathbf{F}$ is the singular force term arising from the surface tension defined by

$$\mathbf{F} = (\nabla_s\sigma - \sigma\kappa\mathbf{n})\delta(\phi)|\nabla\phi|. \tag{4}$$

In above Eq. (4), σ is the surface tension, $\kappa = \nabla\cdot\mathbf{n}$ is the mean curvature, and δ is the Dirac delta function. Also, the surface gradient ∇_s is defined as $(I - \mathbf{n}\otimes\mathbf{n})\nabla$. As mentioned before, the surface tension σ is no longer a constant and depends on the interfacial surfactant concentration c via the Langmuir equation of state [27] as

$$\sigma(c) = \sigma_0 + RTc_\infty \ln(1 - \frac{c}{c_\infty}), \tag{5}$$

where σ_0 is the surface tension of a clean interface (no surfactant, $c = 0$), R and T are the constant of ideal gas and the absolute temperature, respectively, and c_∞ is the interfacial surfactant concentration at the maximum packing.

The interfacial surfactant equation can be written as (e.g., [37,34])

$$\frac{\partial c}{\partial t} + \mathbf{u}\cdot\nabla c - \mathbf{n}\cdot(\nabla\mathbf{u}\,\mathbf{n})c = D_s\nabla_s^2 c + S, \quad \text{on} \quad \Gamma, \tag{6}$$

where D_s is the surface diffusive coefficient and the surface Laplacian operator is defined by $\nabla_s^2 = \nabla_s\cdot\nabla_s$. The source term is $S = r_a C_s(c_\infty - c) - r_d c$, where C_s is the bulk surfactant concentration just adjacent to the interface, r_a and r_d are the adsorption and desorption coefficients, respectively. The interfacial surfactant equation (6) is coupled with the following bulk surfactant equation due to the solubility,

$$\frac{\partial C}{\partial t} + \mathbf{u}\cdot\nabla C = D_b\nabla^2 C, \quad \text{in} \quad \Omega_2, \tag{7}$$

$$D_b\frac{\partial C}{\partial \mathbf{n}}|_\Gamma = S, \quad \frac{\partial C}{\partial \mathbf{n}_2}|_{\partial\Omega} = 0. \tag{8}$$

The interface moves along with the fluid velocity leading to the convection equation of the level-set function

$$\frac{\partial\phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = 0. \tag{9}$$

In order to keep the level-set function as a signed distance function, the re-initialization technique must be adopted by solving the following Hamilton–Jacobi (HJ) equation [33] as

$$\begin{cases} \dfrac{\partial\phi}{\partial\tau} + S_1(\phi_0)(|\nabla\phi| - 1) = 0, \\ \phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}), \end{cases} \tag{10}$$

where τ is a pseudo-time, ϕ_0 is the level-set function obtained from solving Eq. (9), and $S_1(\cdot)$ is the sign function. Readers who are interested in the level-set method and its numerical implementation and applications can refer to the book [23].

2.1. Dimensionless equations

The density and the viscosity are made dimensionless by using ρ and μ as corresponding characteristics. The surface tension is re-scaled by σ_0. The other physical quantities are made dimensionless by using the characteristic length L (drop radius), the characteristic speed U, the characteristic interfacial concentration c_0, and the characteristic bulk surfactant concentration C_0. Specifically, we define

$$\sigma := \frac{\sigma}{\sigma_0}, \quad \mathbf{x} := \frac{\mathbf{x}}{L}, \quad c := \frac{c}{c_0}, \quad C := \frac{C}{C_0},$$
$$t := \frac{Ut}{L}, \quad \mathbf{u} := \frac{\mathbf{u}}{U}, \quad p := \frac{p}{\rho U^2}.$$

Therefore, the relevant dimensionless numbers are the Reynolds number Re, the capillary number Ca, the Biot number Bi, the adsorption number γ, the adsorption depth η, the surfactant coverage ζ, the surfactant elasticity E, the bulk Peclet number Pe, and the surface Peclet number Pe_s. These quantities are defined as follows.

$$Re = \frac{\rho UL}{\mu}, \quad Ca = \frac{\mu U}{\sigma_0}, \quad Bi = \frac{r_d L}{U},$$
$$\gamma = \frac{r_a C_0}{r_d}, \quad \eta = \frac{c_0}{LC_0}, \quad \zeta = \frac{c_0}{c_\infty},$$
$$E = \frac{RTc_\infty}{\sigma_0}, \quad Pe = \frac{LU}{D_b}, \quad Pe_s = \frac{LU}{D_s}.$$

For simplicity, we still use the same notations for the same variables after the above non-dimensionalization. Similar non-dimensionalization processes for the governing equations can be found in [34,7].

The dimensionless Navier–Stokes equations become

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = \frac{1}{Re}\nabla^2\mathbf{u} + \frac{1}{ReCa}\mathbf{F}, \quad \text{in} \quad \Omega, \tag{11}$$
$$\nabla\cdot\mathbf{u} = 0, \quad \text{in} \quad \Omega, \tag{12}$$
$$\mathbf{u}|_{\partial\Omega} = \mathbf{u}_b, \tag{13}$$

where the singular force $\mathbf{F}$ has the same form as Eq. (4).

The dimensionless Langmuir equation of state becomes

$$\sigma(c) = 1 + E\ln(1-\zeta c), \tag{14}$$

while the dimensionless interfacial surfactant equation becomes

$$\frac{\partial c}{\partial t} + \mathbf{u}\cdot\nabla c - \mathbf{n}\cdot(\nabla\mathbf{u}\,\mathbf{n})c = \frac{1}{Pe_s}\nabla_s^2 c + S, \quad \text{on} \quad \Gamma, \tag{15}$$

where the source term

$$S = Bi[\gamma C_s(\frac{1}{\zeta} - c) - c]. \tag{16}$$

Notice that, in practice, the interfacial surfactant concentration c is extended into a small neighborhood of the interface by solving the following HJ equation [43]

$$\frac{\partial c}{\partial \tau} + S_1(\phi)\mathbf{n}\cdot\nabla c = 0. \tag{17}$$

The dimensionless bulk surfactant equation now becomes

$$\frac{\partial C}{\partial t} + \mathbf{u}\cdot\nabla C = \frac{1}{Pe}\nabla^2 C, \quad \text{in} \quad \Omega_2, \tag{18}$$

$$\frac{1}{Pe}\frac{\partial C}{\partial \mathbf{n}}|_\Gamma = \eta S, \quad \frac{\partial C}{\partial \mathbf{n}_2}|_{\partial\Omega} = 0. \tag{19}$$

In order to calculate the bulk surfactant adjacent to the interface C_s in the source term S, the following extension equation must be solved to extrapolate the bulk surfactant concentration C to the outside of the bulk fluid domain (or inside of the drop domain) [1]

$$\frac{\partial C}{\partial \tau} + S_1(\phi - h)\mathbf{n}\cdot\nabla C = 0, \tag{20}$$

where h is the Eulerian mesh width used in numerical discretization.

2.2. Conservation law for total surfactant mass

In this paper, the bulk fluid domain Ω_2 evolves but with fixed outer boundary $\partial\Omega$. Thus the Reynolds transport theorem [36] in Ω_2 should be modified as in the following lemma.

Lemma 1. *Let $f(\mathbf{x},t)$ be any differentiable function on the evolving domain Ω_2, then*

$$\frac{d}{dt}\int_{\Omega_2} f(\mathbf{x},t)d\mathbf{x} = \int_{\Omega_2}\left(\frac{\partial f}{\partial t} + \nabla\cdot(f\mathbf{u})\right)d\mathbf{x} - \int_{\partial\Omega} f\mathbf{u}\cdot\mathbf{n}_2 ds. \tag{21}$$

Proof. Assume that f can be extended to the whole domain Ω, then

$$\begin{aligned}
\frac{d}{dt}\int_{\Omega_2(t)} f(\mathbf{x},t)d\mathbf{x} &= \frac{d}{dt}\left(\int_{\Omega} f d\mathbf{x} - \int_{\Omega_1} f d\mathbf{x}\right)\\
&= \int_{\Omega}\frac{\partial f}{\partial t}d\mathbf{x} - \int_{\Omega_1}\left(\frac{\partial f}{\partial t} + \nabla\cdot(f\mathbf{u})\right)d\mathbf{x}\\
&= \int_{\Omega_2}\left(\frac{\partial f}{\partial t} + \nabla\cdot(f\mathbf{u})\right)d\mathbf{x} - \int_{\Omega}\nabla\cdot(f\mathbf{u})d\mathbf{x}\\
&= \int_{\Omega_2}\left(\frac{\partial f}{\partial t} + \nabla\cdot(f\mathbf{u})\right)d\mathbf{x} - \int_{\partial\Omega} f\mathbf{u}\cdot\mathbf{n}_2 d\mathbf{x}.
\end{aligned}$$

In the above, the Reynolds transport theorem is applied to the integral in Ω_1 since it is moving with the fluid. □

Theorem 1. *Let the total surfactant mass in the bulk fluid Ω_2 and on the interface Γ be*

$$M(t) = \int_{\Omega_2} C\, d\mathbf{x} + \eta\int_{\Gamma} c\, ds. \tag{22}$$

Then the rate of change of $M(t)$ is

$$\frac{dM(t)}{dt} = -\int_{\partial\Omega}\mathbf{u}\cdot\mathbf{n}_2\, C\, ds. \tag{23}$$

Proof. For the bulk surfactant mass, we have

$$\begin{aligned}
\frac{d}{dt}\int_{\Omega_2} C\, d\mathbf{x} &= \int_{\Omega_2}\left(\frac{\partial C}{\partial t} + \mathbf{u}\cdot\nabla C\right)d\mathbf{x} - \int_{\partial\Omega}\mathbf{u}\cdot\mathbf{n}_2\, C\, ds \quad \text{(using Lemma 1 and incompressibility condition)}\\
&= \int_{\Omega_2}\frac{1}{Pe}\nabla^2 C\, d\mathbf{x} - \int_{\partial\Omega}\mathbf{u}\cdot\mathbf{n}_2\, C\, ds \quad \text{(by Eq. (18))}\\
&= \frac{1}{Pe}\int_{\partial\Omega}\frac{\partial C}{\partial\mathbf{n}_2}ds - \frac{1}{Pe}\int_{\Gamma}\frac{\partial C}{\partial\mathbf{n}}ds - \int_{\partial\Omega}\mathbf{u}\cdot\mathbf{n}_2\, C\, ds\\
&= -\eta\int_{\Gamma} S\, ds - \int_{\partial\Omega}\mathbf{u}\cdot\mathbf{n}_2\, C\, ds \quad \text{(by the boundary conditions in Eq. (19)).}
\end{aligned} \tag{24}$$

For the interfacial surfactant mass, we have

$$\frac{d}{dt}\int_{\Gamma} c\, ds = \int_{\Gamma}\left(\frac{1}{Pe_s}\nabla_s^2 c + S\right)ds = \int_{\Gamma} S\, ds \quad \text{(since } \Gamma \text{ is closed).} \tag{25}$$

Combining equations of (24) and (25), we obtain Eq. (23). □

2.3. Distributional bulk surfactant equation

A systematic derivation of the diffusive domain reformulation of PDEs on irregular domain was developed in [15]. For PDEs on an evolving domain, it is assumed that the whole domain is moving with the velocity; however, this is not the case for the present bulk surfactant equations (18)–(19) in which the domain Ω_2 has a fixed outer boundary $\partial\Omega$. To proceed, we introduce the indicator function of Ω_2 denoted by $H(\phi)$, where H is the Heaviside function, so that

$$\nabla H \cdot \mathbf{n} = \delta(\phi)|\nabla\phi|. \tag{26}$$

Based on the diffusive domain idea in [15,34], the bulk surfactant equation (18) with the boundary condition (19) can be reformulated into a distributional equation defined in the whole computation domain Ω as

$$\frac{\partial(HC)}{\partial t} + \mathbf{u}\cdot\nabla(HC) = \frac{1}{Pe}\nabla\cdot(H\nabla C) - \eta S\nabla H\cdot\mathbf{n}, \quad \text{in} \quad \Omega, \tag{27}$$

with the Neumann boundary condition $\frac{\partial C}{\partial \mathbf{n}_2}|_{\partial\Omega} = 0$. The reformulation for a more general convection–diffusion equation on an evolving domain will be given in Appendix.

3. Numerical algorithm

In practical calculations followed in [23], the Heaviside function H and the delta function δ are smoothed as the following

$$H_\epsilon(x) = \begin{cases} 0, & \text{if } x < -\epsilon, \\ \frac{1}{2}(1 + \frac{x}{\epsilon} + \frac{1}{\pi}\sin(\frac{\pi x}{\epsilon})), & \text{if } |x| \le \epsilon, \\ 1, & \text{if } x > \epsilon, \end{cases} \tag{28}$$

and

$$\delta_\epsilon(x) = \begin{cases} \frac{1}{2\epsilon}(1 + \cos(\frac{\pi x}{\epsilon})), & \text{if} \quad |x| \le \epsilon, \\ 0, \quad \text{otherwise.} \end{cases} \tag{29}$$

where we choose $\epsilon = 1.5h$ with h the mesh width. In the following, we shall present how to advance the solutions from time level n to $n+1$.

3.1. Solving the Navier–Stokes equations

The projection method is used for Navier–Stokes equations (11) and (12). First we solve the intermediate velocity $\mathbf{u}^*$ by the prediction step as

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \frac{3}{2}((\mathbf{u}\cdot\nabla)\mathbf{u})^n - \frac{1}{2}((\mathbf{u}\cdot\nabla)\mathbf{u})^{n-1} + \nabla p^{n-\frac{1}{2}} = \frac{1}{2Re}(\nabla^2\mathbf{u}^* + \nabla^2\mathbf{u}^n) + \frac{1}{ReCa}\mathbf{f}^{n+\frac{1}{2}}, \tag{30}$$

with Dirichlet boundary condition $\mathbf{u}^*|_{\partial\Omega} = \mathbf{u}_b$. Now we project the intermediate velocity $\mathbf{u}^*$ into the divergence free space by first solving Poisson equation for the auxiliary function ψ

$$\Delta\psi = \frac{\nabla\cdot\mathbf{u}^*}{\Delta t}, \qquad \frac{\partial\psi}{\partial\mathbf{n}_2}|_{\partial\Omega} = 0, \tag{31}$$

and then update the velocity by

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \nabla\psi\,\Delta t. \tag{32}$$

Lastly, we update the pressure gradient by

$$\nabla p^{n+\frac{1}{2}} = \nabla p^{n-\frac{1}{2}} + \nabla\psi. \tag{33}$$

In the above, standard centered difference schemes are used for the spatial derivatives discretization, except that the third-order WENO scheme is used for the nonlinear convection terms $((\mathbf{u}\cdot\nabla)\mathbf{u})^n$ and $((\mathbf{u}\cdot\nabla)\mathbf{u})^{n-1}$.

3.2. Solving the equations for the level-set convection, the re-initialization, the extensions and the local level-set technique

The level-set convection equation (9), the re-initialization equation (10), the extension equations (17) and (20) are all solved by the classical Eulerian method, i.e., third-order WENO scheme for the spatial discretization and third-order TVD RK scheme for the time marching (see, e.g., [40]).

Also the local level-set technique developed in [29] is used. More precisely, those equations (9), (10), (17) and (20) are solved in a small tube $\{|\phi| \leq w\}$ containing the interface with different bandwidth w as $9h, 12h, 9h$, respectively. Here h is the mesh width. The number of the pseudo-time iterations is chosen as 12. The interfacial concentration equation (15) is solved in a tube with bandwidth $8h$.

3.3. Solving the convection–diffusion equation for bulk surfactant concentration

A modified Crank–Nicolson scheme is used to discretize Eq. (27) as

$$\frac{(HC)^{n+1}-(HC)^n}{\Delta t}+\frac{3}{2}\left(\mathbf{u}\cdot\nabla(HC)+\eta S\left(\nabla H\cdot\mathbf{n}\right)\right)^n - \frac{1}{2}\left(\mathbf{u}\cdot\nabla(HC)+\eta S\left(\nabla H\cdot\mathbf{n}\right)\right)^{n-1}=\frac{1}{2Pe}\left((\nabla\cdot(H\nabla C))^{n+1}+(\nabla\cdot(H\nabla C))^n\right). \tag{34}$$

Standard centered difference schemes are used for the spatial discretization, except that the third-order WENO scheme is used for the convection term $\mathbf{u}\cdot\nabla(HC)$. In practice, the coefficient H appearing in the diffusion term is made non-degenerate by adding a small positive number 10^{-6} as in [15,7]. The resultant algebraic equation is solved by an algebraic multigrid solver.

3.4. Solving the convection–diffusion equation for interfacial surfactant concentration

We solve the interfacial surfactant equation (15) using the following Eulerian scheme developed earlier in [37]

$$\frac{c^{n+1}-c^n}{\Delta t}=\frac{1}{2Pe_s}\left(\nabla^2c^{n+1}+\nabla^2c^n\right)+\frac{3}{2}A^n-\frac{1}{2}A^{n-1}, \tag{35}$$

where

$$A=-\frac{1}{Pe_s}\left(\mathbf{n}\cdot(\nabla\nabla c)\mathbf{n}+\kappa\mathbf{n}\cdot\nabla c\right)-\mathbf{u}\cdot\nabla c+c\mathbf{n}\cdot(\nabla\mathbf{u})\mathbf{n}+S. \tag{36}$$

Again, the spatial derivatives are discretized using the standard central difference schemes, except the convection term $\mathbf{u}\cdot\nabla c$ for which the third-order WENO scheme is used. Note that, Eq. (35) is solved only in a small neighborhood of the interface. An alternative for solving (15) is to use the semi-Lagrangian scheme in [30]. The semi-Lagrangian method can be more stable and more convenient for local mesh refinement.

3.5. Area/volume conservation and total surfactant mass conservation

Theoretically speaking, the drop area/volume must be conserved in an incompressible flow. However, it is known that using the level-set method to simulate two-phase flow problems might encounter non-neglected area/volume loss. There have been several procedures proposed for improving the area/volume conservation in level-set formulation, see e.g., [23, 38,35,17]. In this work, we implement the numerical technique used in [38] to improve the area/volume conservation if it is needed.

From Theorem 1 in Section 2.2, the total surfactant mass is not conserved mathematically during the time evolution unless $\int_{\partial\Omega}\mathbf{u}\cdot\mathbf{n}_2Cds=0$. Typically numerical diffusion leads to artificial surfactant mass loss as can be seen in numerical test in Section 4.3. Although the error is small at each time step, it can be accumulated and leads to inaccurate computational result eventually. Here, we re-scale the bulk and interfacial surfactant concentrations to compensate for the numerical diffusion. More precisely speaking, at each time step we multiply both the interfacial and the bulk concentrations by a re-scaling parameter α which is determined by the surfactant conservation law Eq. (23). The re-scaling procedure is as follows. At each time step t^n, after solving corresponding equations (15) and (18), we obtain both the interfacial surfactant concentration c^n and the bulk surfactant concentration C^n, respectively. Then we compute the numerical total surfactant mass

$$\tilde{M}(t^n)=\int_{\Omega_2}C^nd\mathbf{x}+\eta\int_\Gamma c^nds=\int_\Omega H(\phi^n)C^nd\mathbf{x}+\eta\int_\Omega c^n\delta(\phi^n)|\nabla\phi^n|d\mathbf{x}. \tag{37}$$

Table 1
Errors and convergence rates at $T = 0.5$, $\Delta t = h/2$.

h	e_∞	Rate	e_2	Rate
0.04	9.58E−2		1.68E−1	
0.02	5.22E−2	0.88	7.60E−2	1.14
0.01	2.73E−2	0.94	3.62E−2	1.07
0.005	1.40E−2	0.96	1.77E−2	1.03

However, according to the conservation law of Theorem 1, the actual total mass should be

$$M(t^n) = M(t^{n-1}) - \int_{t^{n-1}}^{t^n} \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n}_2 C ds dt, \tag{38}$$

where the time integral is calculated by the trapezoidal rule. Note that, we have $M(t^0) = \tilde{M}(t^0)$. Therefore, we can compute the re-scaling number α by

$$\alpha = \frac{M(t^n)}{\tilde{M}(t^n)}, \tag{39}$$

and then reset the surfactant concentrations $C^n := \alpha C^n$, and $c^n := \alpha c^n$, so that the conservation law is numerically enforced.

4. Numerical results

In this section, we first check the accuracy of the level-set based diffusive domain (DD) method for the convection–diffusion equation on irregular domains. We then check the accuracy of the whole numerical algorithm described in previous section for the two-phase flows with soluble surfactant problem, and show the efficiency of the re-scaling procedure for surfactant mass conservation. Through various numerical simulations, we show the effect of surfactant solubility on fluid dynamics, and the effects of different dimensionless parameters. Unless specified otherwise, we consider the problem of a drop in the bulk fluid under shear flow; i.e., the velocity boundary condition is set to be $\mathbf{u}_b = (y, 0)^T$ in 2D or $\mathbf{u}_b = (z, 0, 0)^T$ in 3D.

4.1. Accuracy check for the DD method

We first present the numerical accuracy check for the level-set based diffusive-domain method for solving the convection–diffusion equation on irregular domains.

Example 1. Let us first consider a two-dimensional computational domain $\Omega = (-2, 2)^2$ in which an elliptical domain $\Omega_1 = \{\phi < 0\}$ represented by the level-set function $\phi(\mathbf{x}) = \sqrt{x^2 + (y/1.5)^2} - 1$ initially is moving with constant velocity $\mathbf{u} = (1, 0)^T$. The convection–diffusion equation is defined on the exterior domain $\Omega_2 = \Omega \setminus \Omega_1$ as

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = \nabla^2 C + f, \tag{40}$$

with Robin boundary condition $\frac{\partial C}{\partial \mathbf{n}} = (C - g)$ on the interface $\Gamma = \partial\Omega_1$. Here we choose the analytic solution $C(x, y, t) = e^{-t} \sin x \sin y$ so that the source terms f and g can be computed accordingly. Using the diffusive-domain formulation developed in the Appendix, Eq. (40) can be reformulated as

$$\frac{\partial (HC)}{\partial t} + \mathbf{u} \cdot \nabla (HC) = \nabla \cdot (H \nabla C) + Hf - (C - g) \nabla H \cdot \mathbf{n}, \quad \text{in} \quad \Omega. \tag{41}$$

The outer boundary condition is given as the exact Neumann boundary condition.

Although the domain Ω_1 is just shifting, we still solve the level-set equation and the re-initialization equation by using the third-order WENO scheme for spatial discretization and the third-order TVD RK scheme for time stepping. Table 1 shows the mesh refinement results in both L_∞ and L_2 norms at time $T = 0.5$. Here h is the spatial mesh width while the time step size is chosen as $\Delta t = h/2$. One can immediately see that the present DD method has clean first-order accuracy in both norms.

Example 2. We now consider a three-dimensional domain $\Omega = (-2, 2)^3$ in which an irregular domain $\Omega_1 = \{\phi < 0\}$ moves with constant velocity $\mathbf{u} = (1, 0, 0)^T$. The initial interface is represented by the level-set function $\phi = \min(\sqrt{(x - 0.7 - t)^2 + y^2 + z^2} - 0.5, \sqrt{(x + 0.7 - t)^2 + y^2 + z^2} - 0.5)$. As in Example 1, we solve the same equation (40) in the domain $\Omega_2 = \Omega \setminus \Omega_1$. Here we choose the exact solution $C(x, y, z, t) = e^{-t} \cos(\pi x) \cos(\pi y) \cos(\pi z)$. The inner and

Table 2
Errors and convergence rates at $T = 0.4$, $\Delta t = h/2$.

h	e_∞	Rate	e_2	Rate
0.16	1.17E−1		1.45E0	
0.08	7.47E−2	0.65	5.78E−1	1.33
0.04	3.43E−2	1.12	2.55E−1	1.18
0.02	1.77E−2	0.95	1.19E−1	1.01

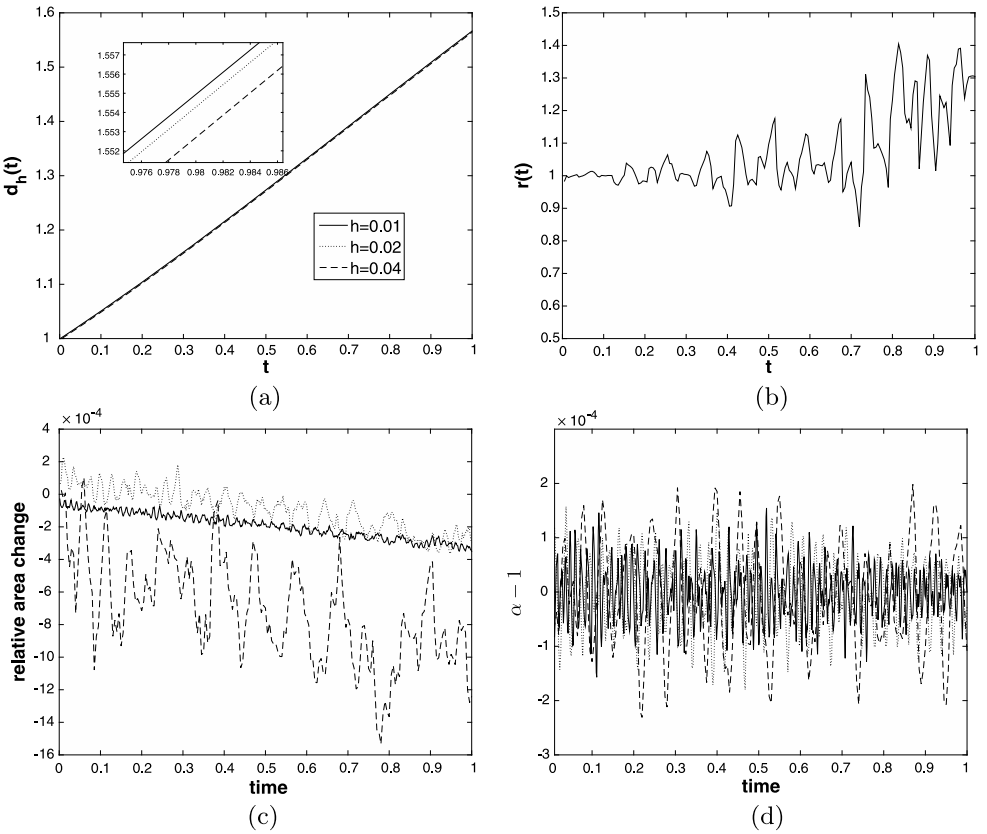


Fig. 2. (a) the time evolutionary plots of the maximum distance from the interface to the drop center $d_h(t)$; (b) the time evolutionary plot of the estimated convergence rate $r(t)$; (c) the time evolutionary plots of relative area loss of the drop; (d) the time evolutionary plots of the re-scaling parameter $\alpha - 1$.

the outer boundary conditions are of the same Robin and Neumann type, respectively, as in Example 1. Table 2 shows the mesh refinement results computed up to time $T = 0.4$ with the time step $\Delta t = h/2$. Again, the present DD method achieves first-order accuracy in both L_∞ and L_2 norms.

4.2. A convergence study for the global numerical algorithm

In this subsection, we perform a convergence study for the global numerical algorithm proposed in previous section. Let us put a drop with radius $L = 1$ at the center of the computational domain $\Omega = (-5, 5) \times (-2, 2)$ under shear flow ($\mathbf{u}_b = (y, 0)^T$) initially. The initial bulk surfactant concentration and the interfacial surfactant concentration are both set to be uniformly as $C(\mathbf{x}, 0) \equiv 1$ and $c(s, 0) \equiv 1$. The dimensionless parameters are chosen as $Re = 1$, $Ca = 0.5$, $E = 0.2$, $\zeta = 0.3$, $Bi = 0.1$, $Pe = Pe_s = 10$, $\gamma = 0.1$, $\eta = 0.1$.

Under the shear flow, the drop is stretched along the flow direction. Due to the lack of exact solution, we choose the maximum distance from the interface to the drop center as our measuring quantity. The computations are run up to $T = 1$ with three different mesh widths. The time step is chosen as $\Delta t = h/8$. Let us denote by $d_h(t)$ the maximum distance from the interface to the drop center at time t using the mesh width h, then the estimated rate of convergence $r(t)$ can be computed by the following formula

$$r(t) = \log_2 \frac{|d_{4h}(t) - d_{2h}(t)|}{|d_{2h}(t) - d_h(t)|}. \tag{42}$$

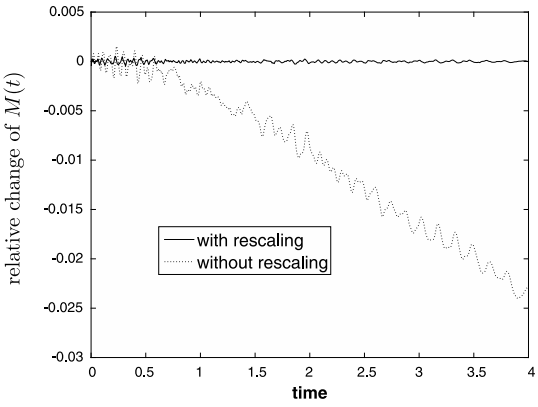


Fig. 3. Comparison of relative change of total surfactant mass with and without the re-scaling procedure.

Fig. 2(a) shows the time evolutionary plots of the maximum distance from the interface to the drop center $d_h(t)$ with three different mesh widths $h = 0.01, 0.02, 0.04$. As expected, the drop stretches so the maximum distance increases as time evolves. In addition, one can see the convergence behavior as we zoom in the plots locally. Fig. 2(b) shows the evolutionary plot of $r(t)$. One can see the estimated convergence rate for the maximum distance oscillates around one so the overall method is roughly first-order accurate. Suppose that the error has the form $|d_h(t) - d(t)| = \beta h^r$, where $d(t)$ is the exact maximum distance. The oscillation may indicate that coefficient β is not a constant but depends on h and/or t. We also show how the relative area of the drop $\frac{A(t)-A(0)}{A(0)}$ changes as time evolves in Fig. 2(c), here $A(t)$ is the area of the drop region at time t. One can see that the relative area loss for the cases of $h = 0.01, 0.02$ are within 0.04% and certainly the finer mesh preserves the drop area better. Instead of plotting the re-scaling parameter α for the surfactant mass conservation, we plot the difference of α from one. Fig. 2(d) shows the time evolutionary plots of $\alpha - 1$ for the cases of three different meshes. Although it is hard to distinguish in details, one can see that the magnitudes of $\alpha - 1$ are all within 3×10^{-4} meaning that only very small correction for the bulk and interfacial surfactant concentrations are needed at each time step in order to preserve the total surfactant mass numerically. Therefore, our method preserves the drop area and the total surfactant mass pretty well.

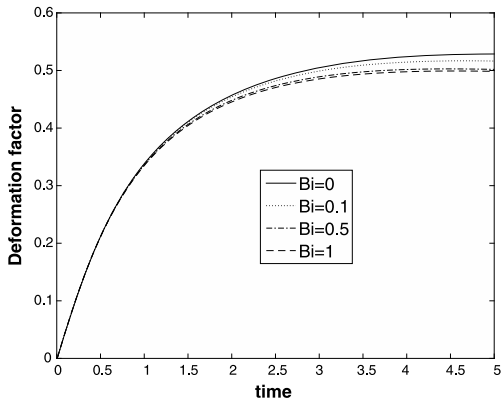
4.3. Effect of re-scaling surfactant procedure

In this test, we investigate the effect of re-scaling bulk and interfacial surfactant as described in Subsection 3.5. Again, we put a drop with radius $L = 1$ at the center of the computational domain $\Omega = (-5, 5) \times (-2, 2)$ under shear flow initially. The initial interfacial surfactant concentration is set to be $c(s, 0) \equiv 1$ while the bulk surfactant concentration is $C(\mathbf{x}, 0) \equiv 1$ only in the region of $0 \le \phi \le 0.5$ and zero elsewhere. The other parameters are all the same as in previous subsection except the absorption depth $\eta = 1$. The mesh width is $h = 0.04$ and the time step size is $\Delta t = h/8$. Fig. 3 shows the comparison of relative change of total surfactant mass $\frac{M(t)-M(0)}{M(0)}$. One can immediately see that with re-scaling, the total surfactant mass conserves well while without the re-scaling, the surfactant mass will lose gradually. The plot of the magnitude of re-scaling parameter $\alpha - 1$ is similar to those in previous subsection so we omit here.

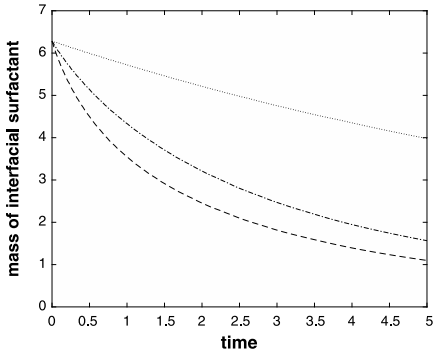
4.4. Effect of varied Biot number

In this test, we study the desorption effect of the interfacial surfactant by varying the Biot number. The problem setup and dimensionless parameters are almost the same as in Subsection 4.3 except a slightly small capillary number $Ca = 0.3$ is used since we like to measure the drop steady shape quantitatively. To see the desorption effect, we choose the initial interfacial surfactant concentration to be $c(s, 0) \equiv 1$ while the bulk surfactant concentration is $C(\mathbf{x}, 0) \equiv 0.1$ only in the region of $0 \le \phi \le 0.5$ and zero elsewhere. The computations are run up to time $T = 5$.

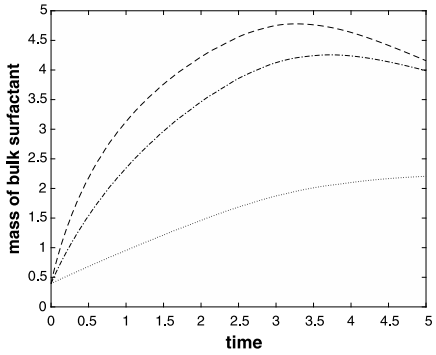
The shape of drop deformation can be measured by the deformation factor defined as $D = \frac{L-B}{L+B}$, where L and B are the maximum and minimum distance from the interface to drop center, respectively. Fig. 4 shows the time plots of the drop deformation factor, the interfacial surfactant mass, and the bulk surfactant mass for varied Biot number. Here, $Bi = 0$ corresponds to the case of insoluble surfactant (no absorption or desorption occur). One can see that as Biot number increases, the desorption of the surfactant from the interface to the bulk increases as well which decreases the interfacial surfactant mass more significantly as we can see from Fig. 4(b). Correspondingly, in Fig. 4(c), the increase of the bulk surfactant mass becomes more significantly as Biot number increases at initial stage. However, after some time, the bulk surfactant mass begins to decrease due to the fact that the surfactant is transported out of the computational domain. Since the drop is stretched, the deformation factor increases as time evolves, and the smaller Biot number is, the larger deformation will be, as shown in Fig. 4(a). Fig. 5 shows the detailed interfacial quantities at time $T = 5$; namely (a) interfacial surfactant concentration, (b) surface tension, (c) Marangoni force, and (d) capillary force, for different Biot numbers. As you can see



(a) Deformation factor



(b) The interfacial surfactant mass



(c) The bulk surfactant mass

Fig. 4. The time evolutionary plots for varied Biot number. (a) drop deformation factor; (b) the interfacial surfactant mass; (c) the bulk surfactant mass. $Bi = 0$ (solid line), $Bi = 0.1$ (dotted line), $Bi = 0.5$ (dash-dotted line), $Bi = 1$ (dashed line).

from Fig. 5(a), the interfacial surfactant concentration decreases as Bi decreases which confirms the interfacial surfactant mass behavior shown in Fig. 4(b). Meanwhile, as Bi increases, the surface tension gradient becomes smaller, and thus less change of Marangoni force which results in less drop deformation. The capillary force remains largely unchanged since it is dominated by the interface curvature.

4.5. Effects of varied adsorption number γ and adsorption depth η

In this test, we study the effects of varied adsorption number γ and adsorption depth η. We use the same problem setup and dimensionless parameters as in previous Subsection 4.4 except with fixed $Bi = 0.1$ and varied $\gamma = 0.1, 0.3, 1$ and $\eta = 0.1, 0.5, 1$, respectively. Moreover we consider an unsteady case, so we choose $Ca = 0.5$. Fig. 6 shows the time evolutionary plots of interfacial surfactant mass (top), bulk surfactant mass (middle) and drop deformation factor (bottom). The computations are run up to time $T = 6$. Since the interfacial concentration is higher than the bulk concentration initially, the interfacial desorption always occurs as you can see the interfacial surfactant mass decreases as time evolves. For varied adsorption number case, the higher γ has less decrease in the interfacial surfactant mass and less increase in the bulk surfactant mass (up to some time) as you can see from Fig. 6(a) and (c) which is not surprising from the physical point of view. On the other hand, for the varied adsorption depth case, the higher η has more decrease in the interfacial surfactant mass and more increase in the bulk surfactant mass (up to some time) as you can see from Fig. 6(b) and (d). Nevertheless, those variations on γ and η affect slightly on the drop deformation factor as shown in Fig. 6(e) and (f), respectively.

4.6. Effect of varied bulk Peclet number Pe

In this test, we study the effect of varied bulk Peclet number by fixing other parameters used in Subsection 4.2. In order to have better resolution for the case of high Peclet number, we choose the mesh width $h = 0.02$ and the time step size $\Delta t = h/16$. To see the bulk surfactant diffusion effect, we choose the initial interfacial surfactant concentration to be $c(s, 0) \equiv 0.1$ while the bulk surfactant concentration is $C(\mathbf{x}, 0) \equiv 1$ only in the region of $0 \le \phi \le 0.5$ and zero elsewhere. The computations are run up to time $T = 8$.

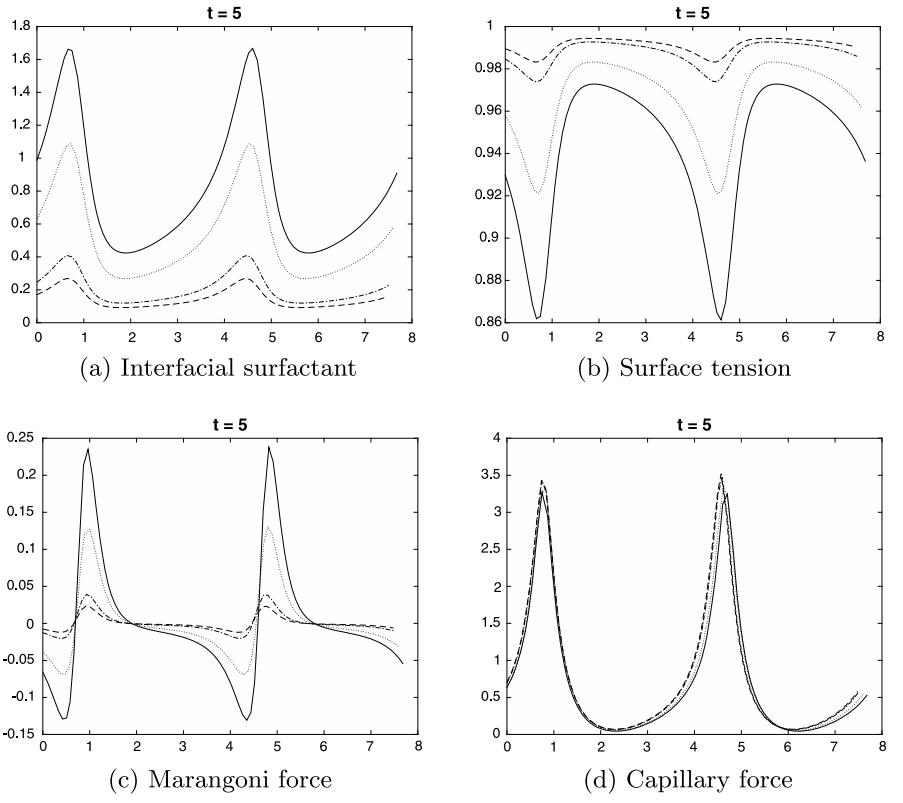


Fig. 5. Interfacial quantities versus arc-length at $T = 5$. (a) interfacial surfactant concentration; (b) surface tension; (c) Marangoni force; (d) capillary force. $Bi = 0$ (solid line), $Bi = 0.1$ (dotted line), $Bi = 0.5$ (dash-dotted line), $Bi = 1$ (dashed line).

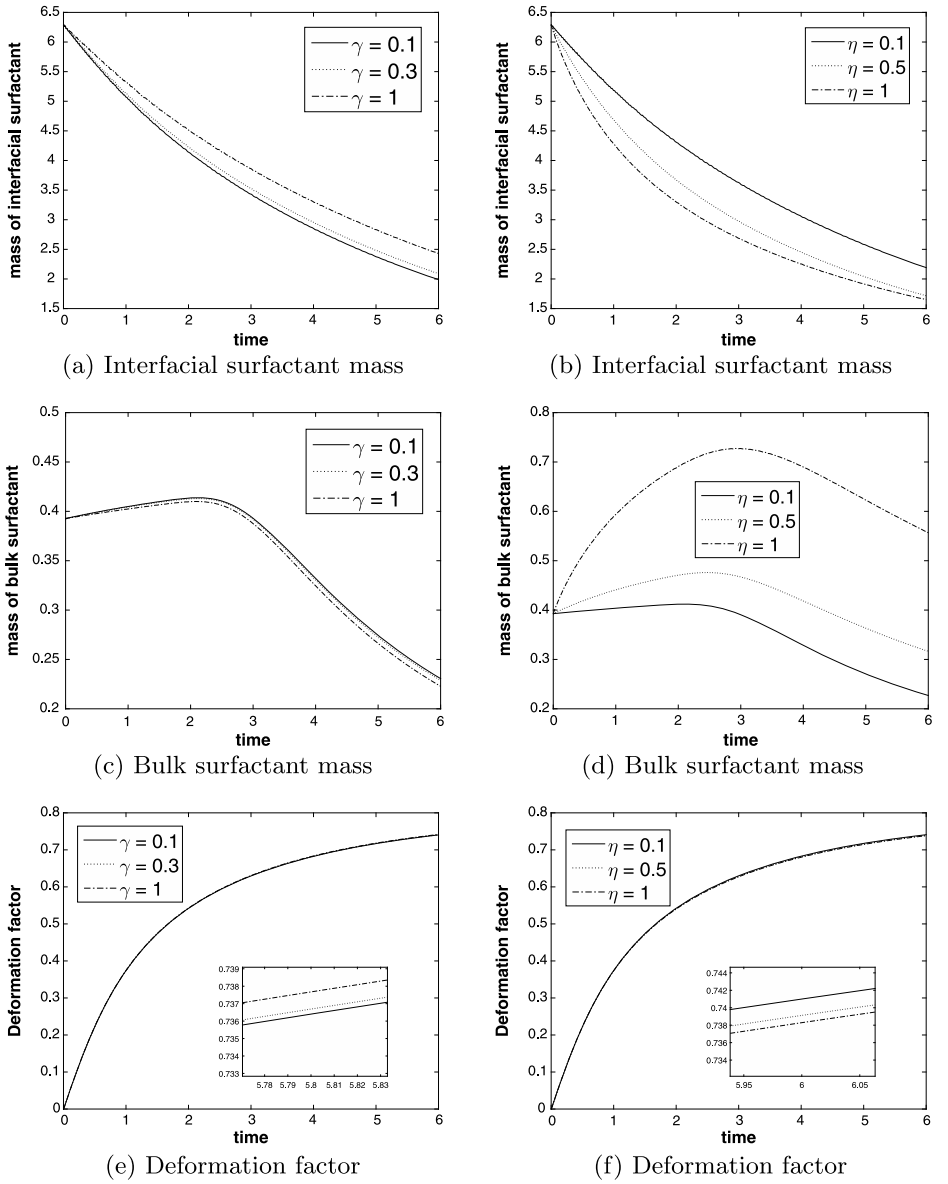
Drop deformation for different bulk Peclet number is shown in Fig. 7(a). It seems to be insensitive to Pe, but from the locally zooming plot, it increases slightly as Pe increases. Fig. 7(b) and (c) show the Marangoni force and the source term S at time $T = 8$, respectively. One can clearly see that as Pe increases, the magnitudes of the Marangoni force and source term increases as well. However, the patterns of S are similar for $Pe = 0.1, 1, 10$ but is quite different for the case of $Pe = 1000$. In the latter case, the source term reaches a positive maximum at one side of each drop tip and then abruptly jumps to a negative minimum at the adjacent side of the tip which means the surfactant is desorbed from the interface to bulk on one side and meanwhile it is absorbed from the bulk to interface on the other side.

Fig. 8 shows the corresponding contour plots of bulk surfactant concentration at $T = 8$ for different Peclet number. With small $Pe = 0.1$, the bulk surfactant spreads quickly and almost uniformly into the whole bulk domain. For large $Pe = 1000$, a thin layer of bulk surfactant concentration develops near the interface as observed in Fig. 8. In this case, surfactant is transported away from the drop in drop-tip direction. For even larger Pe, it is infeasible to resolve this boundary layer by mesh refinement. Singular perturbation analysis should be included in the numerical algorithm, see [4].

4.7. Interaction of two drops in 3D

We now are ready to perform three-dimensional simulations. In the first test, we study the drops interaction under shear flow and see the effect of surfactant solubility. We consider two identical drops with unit radius centered at $(1.4, 0.0, -0.8)$ and $(-1.4, 0.0, 0.8)$ in computational domain $\Omega = (-5, 5) \times (-2, 2) \times (-3, 3)$. So the initial two level-set functions are $\phi_1 = \sqrt{(x-1.4)^2 + y^2 + (z+0.8)^2} - 1$ and $\phi_2 = \sqrt{(x+1.4)^2 + y^2 + (z-0.8)^2} - 1$, respectively. If the drops are covered with soluble surfactant, the initial interfacial surfactant concentration is set to be $c(s, 0) \equiv 1$ while the initial bulk concentration is $C(\mathbf{x}, 0) \equiv 0.5$ in the region $\{0 \le \phi_1 \le 0.5\} \cup \{0 \le \phi_2 \le 0.5\}$ and is zero elsewhere. The dimensionless parameters are $Re = 0.4$, $Ca = 0.2$, $\gamma = 0.1$, $\eta = 0.01$, $E = 0.2$, $\zeta = 0.3$, $Pe_s = Pe = 10$. We perform numerical simulations for different Biot number $Bi = 0, 0.01, 0.1, 1$ to see the effect of surfactant solubility. For comparison purpose, we also run the simulation of the clean interface which has no surfactant at all. We choose the mesh width $h = 0.04$ and the time step size $\Delta t = h/16$, and run the simulations up to time $T = 5$.

It is known that the surfactant enhances drops bouncing and prevents coalescence as shown in [39,24]. Under the shear flow, two drops will first approach each other, then pass around, and finally depart away. Fig. 9 shows the time evolutionary plots of the minimum distance between two drops. One can see that the distances between two drops with



(a) Interfacial surfactant mass (b) Interfacial surfactant mass

(c) Bulk surfactant mass (d) Bulk surfactant mass

(e) Deformation factor (f) Deformation factor

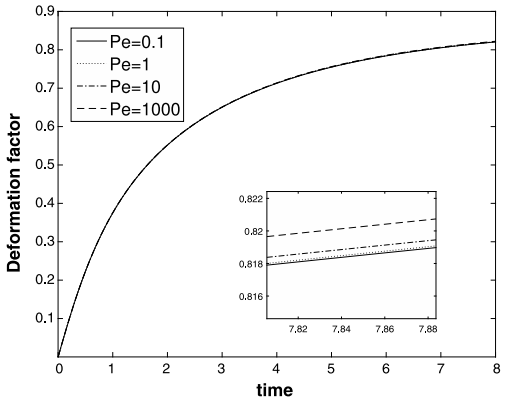
Fig. 6. The time evolutionary plots of interfacial surfactant mass (top), bulk surfactant mass (middle) and drop deformation factor (bottom) for varied γ (left column) and varied η (right column).

surfactant are always larger than the one of without surfactant which shows the fact that surfactant prevents drops coalescence. When the two drops are getting close, the Marangoni force creates an inflow near the contact region, which helps to separate those drops, see, e.g., [39] as well. As the Biot number increases, the minimum distance decreases due to the fact that with larger Bi, more interfacial surfactant is desorbed into the bulk fluid leading to weaker Marangoni force.

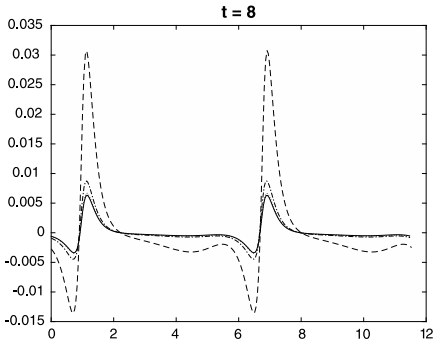
The drop morphology together with the interfacial concentration for the case of $Bi = 0.01$ at different times are shown in Fig. 10, the corresponding bulk surfactant concentration on xz-plane is presented in Fig. 11.

4.8. Droplet breakup in 3D

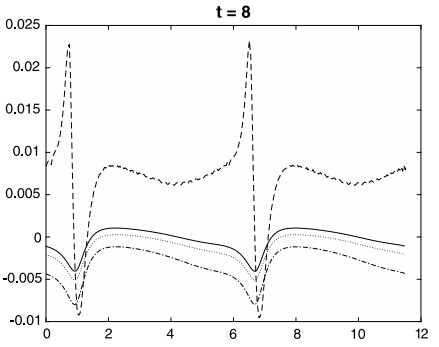
In this simulation, we study the drop breakup behavior under shear flow. Here, we compare three different cases; namely, a clean drop (without surfactant), a drop with insoluble surfactant ($Bi = 0$) and with soluble surfactant case ($Bi = 0.1$). We put the drop with unit radius at the center of the computational domain $\Omega = (-7, 7) \times (-2, 2) \times (-2, 2)$ initially so the initial level-set function $\phi = \sqrt{x^2 + y^2 + z^2} - 1$. If the drop is covered with soluble surfactant, the initial interfacial surfactant concentration is set to be $c(s, 0) \equiv 1$ while the initial bulk concentration is $C(\mathbf{x}, 0) \equiv 0.5$ in the region of $\{0 \leq \phi \leq 0.5\}$ and



(a) Deformation factor



(b) Marangoni force



(c) Source term S

Fig. 7. The plots for varied bulk Peclet number. (a) drop deformation factor; (b) Marangoni force; (c) the source term S. $Pe = 0.1$ (solid line), $Pe = 1$ (dotted line), $Pe = 10$ (dash-dotted line), $Pe = 1000$ (dashed line).

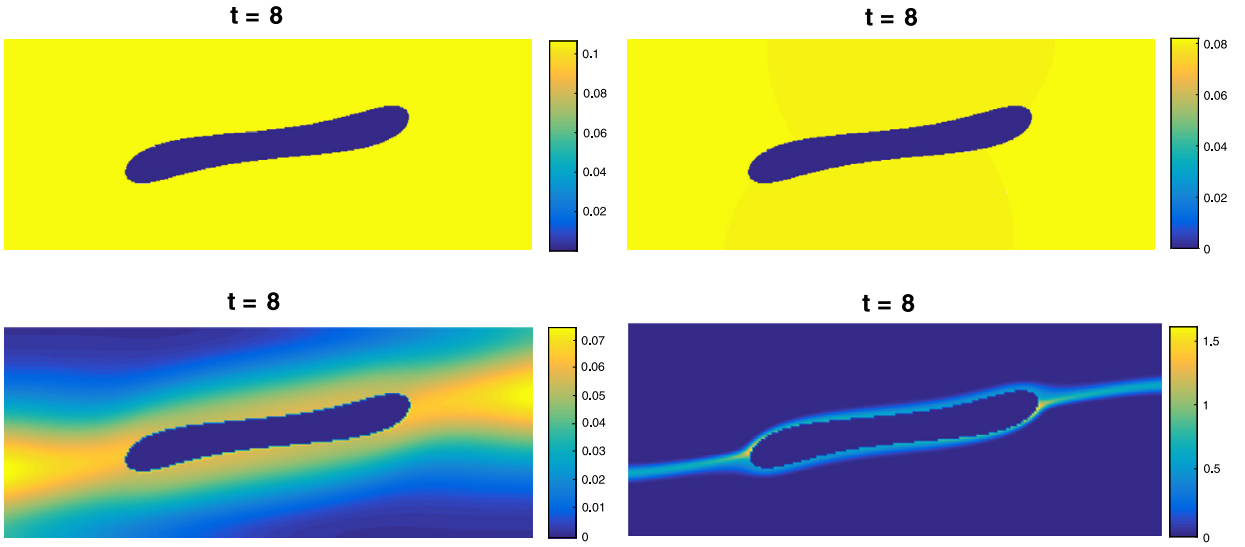


Fig. 8. Contour plot of bulk surfactant concentration at $T = 8$. $Pe = 0.1$ (top left), $Pe = 1$ (top right), $Pe = 10$ (bottom left), $Pe = 1000$ (bottom right). Thin bulk surfactant boundary layer forms along the drop sides for $Pe = 1000$. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

is zero elsewhere. The dimensionless parameters are $Ca = 0.4$, $Re = 0.4$, $E = 0.2$, $\zeta = 0.3$, $Pe_s = 10$, $Pe = 10$, $\gamma = 0.8$, $\eta = 0.01$. We choose the mesh width $h = 0.04$ and the time step size $\Delta t = h/12$, and run the simulations up to time $T = 27$.

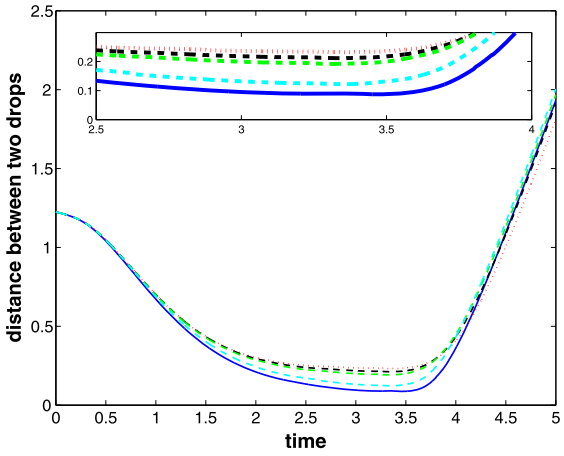


Fig. 9. The time evolutionary plots of the minimum distance between two drops under shear flow. $Bi = 0$ (insoluble surfactant, red dotted), $Bi = 0.01$ (black dashed), $Bi = 0.1$ (green dashed), $Bi = 1$ (cyan dashed), without surfactant (blue solid). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

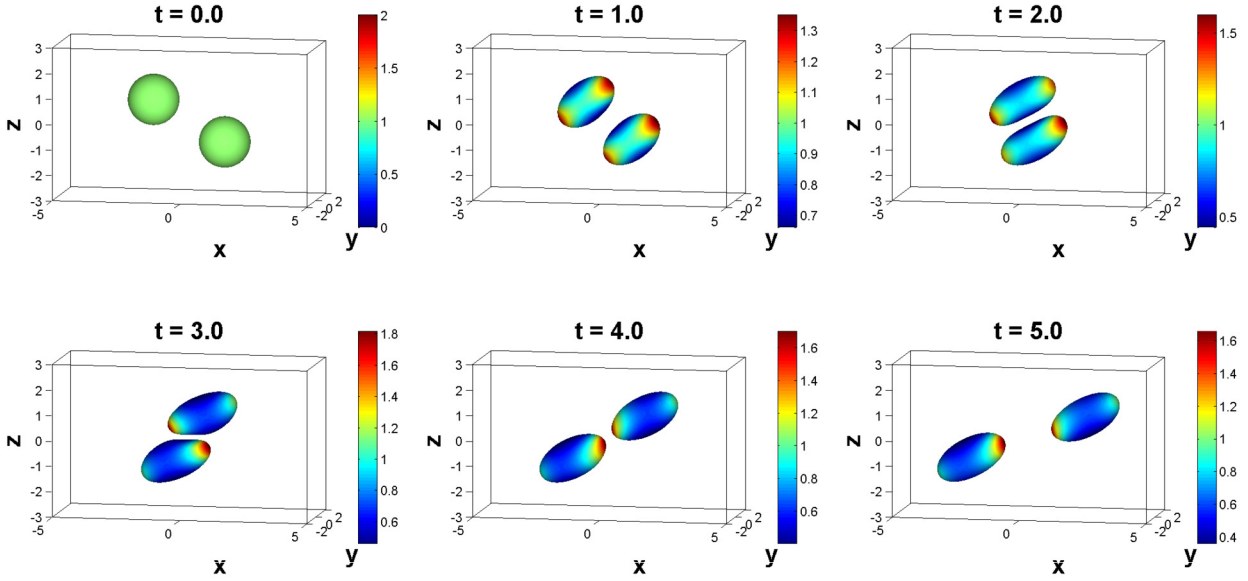


Fig. 10. Morphology of drops (with soluble surfactant, $Bi = 0.01$) together with interfacial surfactant concentration at different times. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

Due to the shear stress, the drop is elongated. Theoretically, there is a critical capillary number Ca^* beyond which the drop eventually breaks up, and under which the drop reaches a steady state. Simulation results for the clean drop and drop with soluble surfactant are given in Fig. 12 and Fig. 13, respectively. For the soluble surfactant case, the dynamics of the bulk concentration is shown in Fig. 14. After some time of elongation, the drop gradually evolves into three main bodies connected with thin necks. Then the necks become thinner and thinner until the drop breaks up.

It is clearly observed that the surfactant-laden drop breaks up at earlier time than the clean case. This is because the non-uniform distribution of surfactant leads to larger deformation which thins the drop more than the clean case. Also the drop with soluble surfactant breaks up later than drop with insoluble surfactant (not shown here) due to the desorption of surfactant from the interface to the bulk which reduces the effect of surfactant. These simulations demonstrate the capability of the level-set method in handling topological changes. Similar results were presented in [34] for the phase-field method where the velocity periodic boundary conditions are used.

5. Conclusion

A level-set method for computing two-phase flows with soluble surfactant is presented in this paper. Firstly, we develop a level-set based diffusive domain method for solving the bulk surfactant equation on complex domains with appropriate

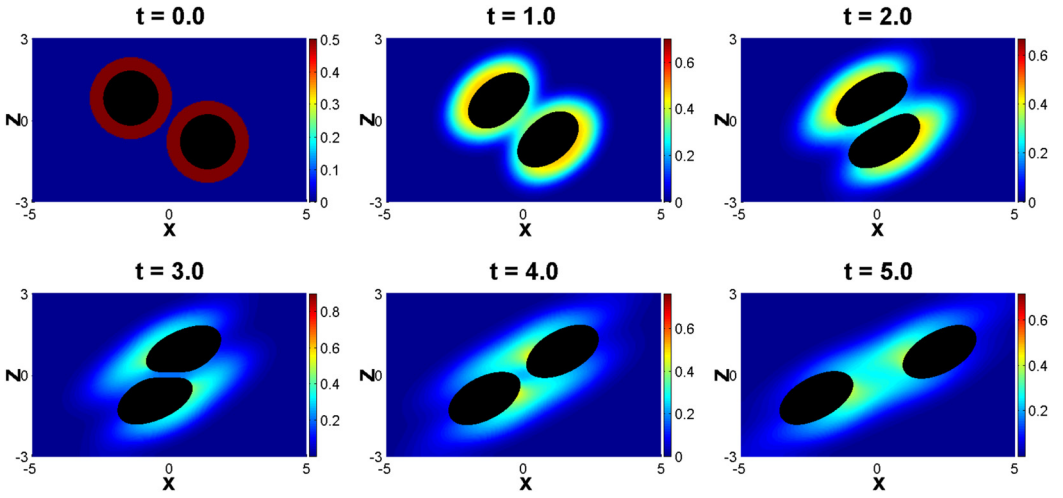


Fig. 11. Bulk surfactant concentration on *xz*-plane at different times. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

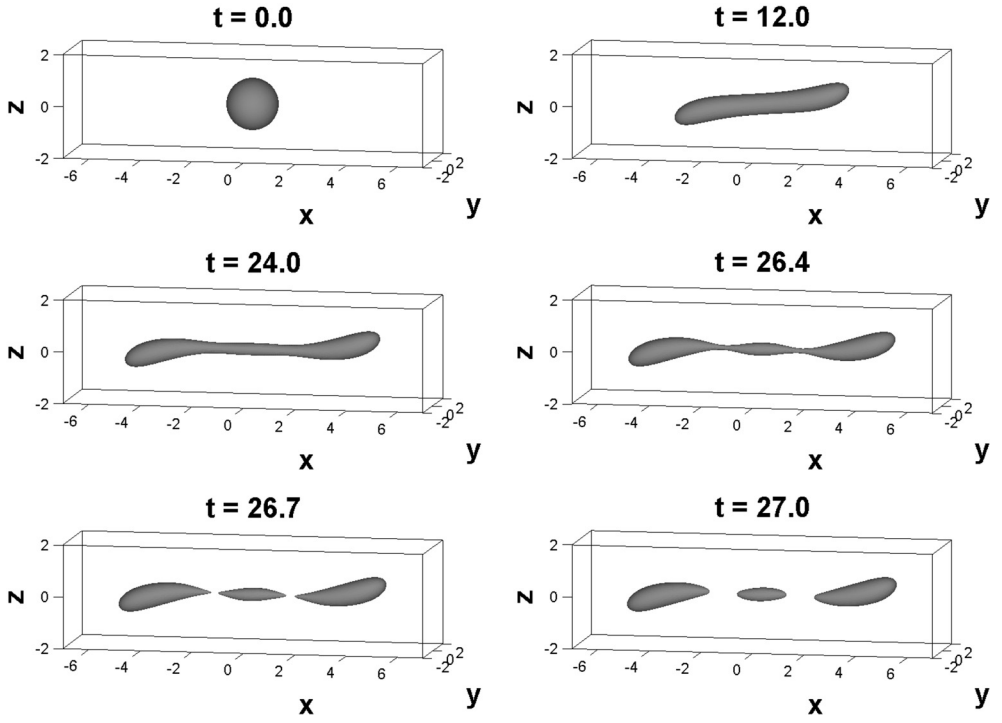


Fig. 12. Morphology of the clean drop at different times.

handling the adsorption/desorption of surfactant between the interface and bulk fluid. Conservation law for total surfactant mass is derived and an efficient re-scaling procedure for the bulk/interfacial surfactant concentrations is proposed to compensate for the surfactant mass loss due to the numerical diffusion. The whole numerical algorithm is easy to implement and has been tested for several numerical simulations in two- and three-dimensional spaces. In particular, numerical simulations show the surfactant solubility can have significant effects on the drop dynamics under shear flow. The present level-set method is capable of performing 3D drop breakup simulations. We shall use the present code to study more realistic applications in the future work.

Acknowledgements

We would like to thank the referee for insightful comments. This work was partially supported by national key scientific and technological project of China (No. 2014ZX07104-006) and national natural science fund of China (No. 91430213).

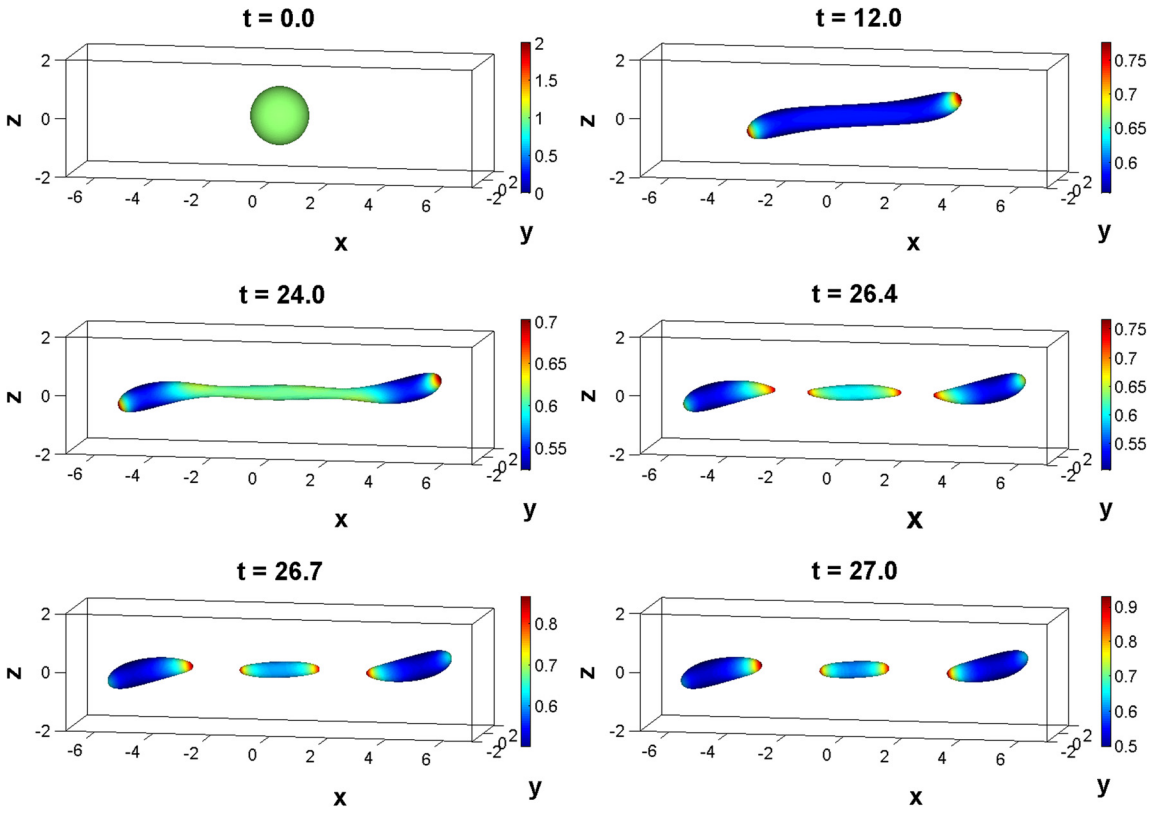


Fig. 13. Morphology of the drop (with soluble surfactant) together with interfacial surfactant concentration at different times. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

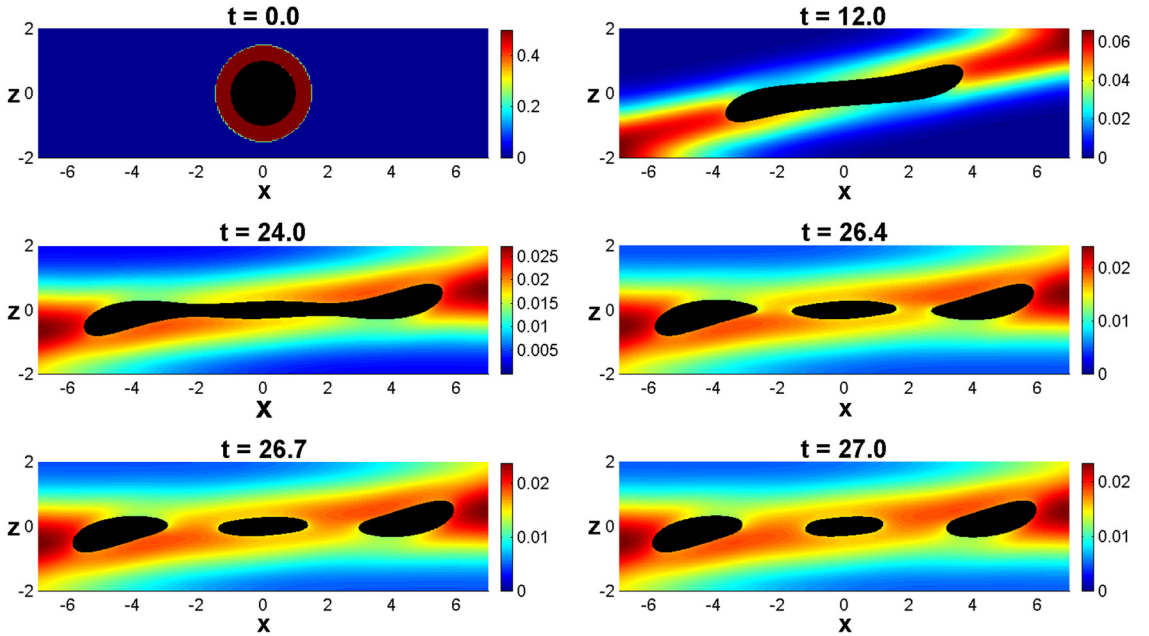


Fig. 14. Bulk surfactant concentration on *xz*-plane at the corresponding times for the soluble surfactant case. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

Appendix

In this appendix, we modify the derivation of the diffusive domain formulation for solving PDEs in complex domains in [15] by taking into account that the outer boundary is fixed and replacing the phase-field function by the level-set function.

We consider the following general convection–diffusion equation on an evolving domain as

$$\frac{\partial C}{\partial t} + \mathbf{u}\cdot\nabla C = \nabla\cdot(A\nabla C) + f, \quad \text{in} \quad \Omega_2(t), \tag{43}$$

where A is a positive definite matrix, and $\Omega_2 = \Omega\setminus\Omega_1$ as illustrated in Fig. 1. The interface Γ and the domain Ω_1 is moving with the prescribed velocity $\mathbf{u}$. The inner boundary condition (on Γ) for above equation is a Robin type as

$$A\nabla C\cdot\mathbf{n}|_\Gamma = \beta(C-g), \qquad \beta>0 \tag{44}$$

while the outer boundary condition on $\partial\Omega$ is arbitrary.

Assume that f and g are extended so that they are defined on Ω. Let $\psi(\mathbf{x},t)$ be a test function. Multiplying the test function on Eq. (43) and integrating in the domain Ω_2, we obtain

$$\int_{\Omega_2}\psi\left(\frac{\partial C}{\partial t}+\mathbf{u}\cdot\nabla C\right)d\mathbf{x} = \int_{\Omega_2}\psi\left(\nabla\cdot(A\nabla C)+f\right)d\mathbf{x}. \tag{45}$$

The first term becomes

$$\begin{aligned}\int_{\Omega_2}\psi\frac{\partial C}{\partial t}d\mathbf{x} &= \int_{\Omega_2}\left(\frac{\partial(\psi C)}{\partial t} - C\frac{\partial\psi}{\partial t}\right)d\mathbf{x}\\ &= \frac{d}{dt}\int_{\Omega_2}\psi C d\mathbf{x} - \int_{\Omega_2}\left(\nabla\cdot(\psi C\mathbf{u}) + C\frac{\partial\psi}{\partial t}\right)d\mathbf{x} + \int_{\partial\Omega}\psi C\mathbf{u}\cdot\mathbf{n}_2 ds\\ &= \frac{d}{dt}\int_{\Omega_2}\psi C d\mathbf{x} + \int_{\Gamma}\psi C\mathbf{u}\cdot\mathbf{n} ds - \int_{\Omega_2} C\frac{\partial\psi}{\partial t}d\mathbf{x},\end{aligned} \tag{46}$$

where Lemma 1 and the divergence theorem are used in above equalities.

The third term is

$$\begin{aligned}\int_{\Omega_2}\psi\nabla\cdot(A\nabla C)d\mathbf{x} &= \int_{\Omega_2}\left(\nabla\cdot(\psi A\nabla C) - \nabla\psi\cdot(A\nabla C)\right)d\mathbf{x}\\ &= -\int_{\Gamma}\psi A\nabla C\cdot\mathbf{n} ds + \int_{\partial\Omega}\psi A\nabla C\cdot\mathbf{n}_2 ds - \int_{\Omega_2}\nabla\psi\cdot(A\nabla C)d\mathbf{x}\\ &= -\int_{\Gamma}\psi A\nabla C\cdot\mathbf{n} ds + \int_{\partial\Omega}\psi A\nabla C\cdot\mathbf{n}_2 ds - \int_{\Omega}\nabla\psi\cdot(HA\nabla C)d\mathbf{x}\\ &= -\int_{\Gamma}\psi A\nabla C\cdot\mathbf{n} ds + \int_{\partial\Omega}\psi A\nabla C\cdot\mathbf{n}_2 ds - \int_{\partial\Omega}\psi A\nabla C\cdot\mathbf{n}_2 ds\\ &\quad + \int_{\Omega}\psi\nabla\cdot(HA\nabla C)d\mathbf{x} = -\int_{\Gamma}\psi A\nabla C\mathbf{n} ds + \int_{\Omega}\psi\nabla\cdot(HA\nabla C),\end{aligned} \tag{47}$$

where H is the Heaviside function, so it is the indicator function of Ω_2.

Integrating with respect to time in Eq. (45), we have

$$\int_0^T\int_{\Omega_2}\psi\left(\frac{\partial C}{\partial t}+\mathbf{u}\cdot\nabla C\right)d\mathbf{x}dt = \int_0^T\int_{\Omega_2}\psi\left(\nabla\cdot(A\nabla C)+f\right)d\mathbf{x}dt \tag{48}$$

Integrating the time in last term of Eq. (46), we obtain

$$\int_0^T \int_{\Omega_2} C\frac{\partial \psi}{\partial t} d\mathbf{x}dt = \int_0^T \int_{\Omega} HC\frac{\partial \psi}{\partial t} d\mathbf{x}dt = \int_0^T \int_{\Omega} \left(\frac{\partial (H\psi C)}{\partial t} - \psi \frac{\partial (HC)}{\partial t} \right) d\mathbf{x}dt$$

$$= \int_{\Omega_2(T)} C\psi d\mathbf{x} - \int_{\Omega_2(0)} C\psi d\mathbf{x} - \int_0^T \int_{\Omega} \psi \frac{\partial (HC)}{\partial t} d\mathbf{x}dt. \tag{49}$$

Plugging Eqs. (46), (47) and (49) into Eq. (48) and after some algebraic manipulation, we have

$$\int_0^T \int_{\Gamma} \psi C\mathbf{u} \cdot \mathbf{n} ds dt + \int_0^T \int_{\Omega} \psi \frac{\partial (HC)}{\partial t} d\mathbf{x}dt + \int_0^T \int_{\Omega} \psi \mathbf{u} \cdot \nabla C d\mathbf{x}dt$$

$$= -\int_0^T \int_{\Gamma} \psi A\nabla C \cdot \mathbf{n} ds dt + \int_0^T \int_{\Omega} \psi \nabla \cdot (HA\nabla C) d\mathbf{x}dt + \int_0^T \int_{\Omega} H\psi f d\mathbf{x}dt. \tag{50}$$

Now, we rewrite the integrals on Γ as integrals in Ω using the δ function as

$$\int_0^T \int_{\Gamma} \psi C\mathbf{u} \cdot \mathbf{n} \delta(\phi) |\nabla \phi| d\mathbf{x}dt + \int_0^T \int_{\Omega} \psi \frac{\partial (HC)}{\partial t} d\mathbf{x}dt + \int_0^T \int_{\Omega} \psi \mathbf{u} \cdot \nabla C d\mathbf{x}dt$$

$$= -\int_0^T \int_{\Gamma} \psi A\nabla C \cdot \mathbf{n} \delta(\phi) |\nabla \phi| d\mathbf{x}dt + \int_0^T \int_{\Omega} \psi \nabla \cdot (HA\nabla C) d\mathbf{x}dt + \int_0^T \int_{\Omega} H\psi f d\mathbf{x}dt. \tag{51}$$

Since the test function ψ is arbitrary, the above equation becomes

$$\frac{\partial (HC)}{\partial t} + H(\mathbf{u} \cdot \nabla C) + C(\mathbf{u} \cdot \mathbf{n})\delta(\phi)|\nabla \phi| = \nabla \cdot (HA\nabla C) + Hf - (A\nabla C \cdot \mathbf{n})\delta(\phi)|\nabla \phi|, \quad \text{in} \quad \Omega. \tag{52}$$

Using the fact $\nabla H \cdot \mathbf{n} = \delta(\phi)|\nabla \phi|$ and combining the second and third terms, we can formally write the equation as

$$\frac{\partial (HC)}{\partial t} + \mathbf{u} \cdot \nabla (HC) = \nabla \cdot (HA\nabla C) + Hf - (A\nabla C \cdot \mathbf{n})\nabla H \cdot \mathbf{n}, \quad \text{in} \quad \Omega. \tag{53}$$

Together with the outer boundary condition, Eq. (53) forms the diffusive-domain formulation of the original equation (43) in Ω_2.

References

[1] T.D. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (2003) 349–355.
[2] S. Aland, J. Lowengrub, A. Voigt, Two-phase flow in complex geometries: a diffuse domain approach, Comput. Model. Eng. Sci. 57 (2010) 77–108.
[3] S.L. Anna, H.C. Meyer, Microscale tip streaming in a microfluidic flow focusing device, Phys. Fluids 18 (2006) 463–470.
[4] M.R. Booty, M. Siegel, A hybrid numerical method for interfacial fluid flow with soluble surfactant, J. Comput. Phys. 229 (2010) 3864–3883.
[5] C.C. Langavant, A. Guittet, M. Theillard, F. Temprano-Coleto, F. Gibou, Level-set simulations of soluble surfactant driven flows, J. Comput. Phys. 348 (2017) 271–297.
[6] S. Chen, B. Merriman, S. Osher, P. Smereka, A simple level set method for solving Stephan problems, J. Comput. Phys. 135 (1997) 8–29.
[7] K.-Y. Chen, M.-C. Lai, A conservative scheme for solving coupled surface-bulk convection–diffusion equations with an application to interfacial flows with soluble surfactant, J. Comput. Phys. 257 (2014) 1–18.
[8] F. Fenton, E. Cherry, A. Karma, W. Rappel, Modeling wave propagation in realistic heart geometries using the phase-field method, Chaos 15 (2005) 1039–1052.
[9] S. Franz, R. Gärtner, H. Roos, A. Vogit, A note on the convergence analysis of a diffusive-domain approach, Comput. Methods Appl. Math. 12 (2012) 153–167.
[10] F. Gibou, R. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (2002) 205–227.
[11] J. Goerke, Pulmonary surfactant: functions and molecular composition, Biochim. Biophys. Acta 1408 (1998) 79–89.
[12] S. Khatri, A.-K. Tornberg, An embedded boundary method for soluble surfactant with interface tracking for two-phase flows, J. Comput. Phys. 256 (2014) 768–790.
[13] C. Landsberg, F. Stenger, A. Deutsch, M. Gerlinsky, A. Rösen-Wolff, A. Voigt, Chemotaxis of mesenchymal stem cells within 3D biomimetic scaffolds – a modeling approach, J. Biomech. 44 (2011) 359–364.
[14] K.Y. Lervag, J. Lowengrub, Analysis of the diffusive-domain method for solving PDEs in complex geometries, Commun. Math. Sci. 13 (2015) 1473–1500.
[15] X. Li, J. Lowengrub, A. Ratz, A. Voigt, Solving PDEs in complex geometries: a diffuse domain approach, Commun. Math. Sci. 7 (2009) 81–107.
[16] Z. Li, K. Ito, The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains, SIAM Front. Appl. Math. 33 (2006).
[17] Z. Li, S. Peng, Adaptive mesh refinement techniques for the immersed interface method applied to flow problems, Comput. Struct. 122 (2013) 249–258.
[18] H. Liu, Y. Zhang, Phase-field modeling droplet dynamics with soluble surfactant, J. Comput. Phys. 229 (2010) 9166–9187.

[19] W. Millken, L.G. Leal, The influence of surfactant on the deformation and breakup of a viscous drop: the effect of surfactant solubility, J. Colloid Interface Sci. 166 (1994) 275–285.
[20] N.R. Morrow, G. Mason, Recovery of oil by spontaneous inhibition, Curr. Opin. Colloid Interface Sci. 6 (2001) 321–337.
[21] M. Muradoglu, G. Tryggvason, A front-tracking method for computation of interfacial flows with soluble surfactant, J. Comput. Phys. 274 (2008) 2238–2262.
[22] M. Muradoglu, G. Tryggvason, Simulations of soluble surfactants in 3D multiphase flow, J. Comput. Phys. 274 (2014) 737–757.
[23] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Applied Mathematical Sciences, vol. 153, Springer, 2003.
[24] K.-L. Pan, Y.-H. Tseng, J.-C. Chen, K.-L. Huang, C.-H. Wang, M.-C. Lai, Controlling droplet bouncing and coalescence with surfactant, J. Fluid Mech. 799 (2016) 603–636.
[25] J. Papac, F. Gibou, C. Ratsch, Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions, J. Comput. Phys. 229 (2010) 875–889.
[26] J. Papac, A. Helgadottir, C. Ratsch, F. Gibou, A level set approach for diffusion and Stefan-type problems with Robin boundary conditions on quadtree/octree adaptive Cartesian grids, J. Comput. Phys. 223 (2013) 241–261.
[27] Y. Pawar, K.J. Stebe, Marangoni effects on drop deformation in an extensional flow: the role of surfactant physical chemistry, I: insoluble surfactants, Phys. Fluids 8 (1996) 1738–1751.
[28] S. Osher, J. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[29] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, M. Kang, A PDE-based fast local level set method, J. Comput. Phys. 155 (1999) 410–438.
[30] W. Shi, J. Xu, S. Shu, An adaptive semi-Lagrangian level-set method for convection–diffusion equations on evolving interfaces, Adv. Appl. Math. Mech. 9 (2017) 1364–1382.
[31] D. Shao, W. Rappel, H. Levine, Computational models for cell morphodynamics, Phys. Rev. Lett. 105 (2010) 108104.
[32] H.A. Stone, A.D. Stroock, A. Ajdari, Engineering flows in small devices: microfluidics toward lab-on-a-chip, Annu. Rev. Fluid Mech. 36 (2004) 381–411.
[33] M. Sussman, P. Smereka, S. Osher, A level-set approach for computing solutions of incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[34] K.E. Teigen, P. Song, J. Lowengrub, A. Voigt, A diffusive-interface method for two-phase flows with soluble surfactant, J. Comput. Phys. 230 (2011) 375–393.
[35] Y. Wang, S. Simakhina, M. Sussman, A hybrid level set-volume constraint method for incompressible two-phase flow, J. Comput. Phys. 231 (2012) 6407–6438.
[36] P. Wesseling, Principles of Computational Fluid Dynamics, Springer-Verlag, 2001.
[37] J. Xu, H. Zhao, An Eulerian formulation for solving partial differential equations along a moving interface, J. Sci. Comput. 19 (2003) 573–594.
[38] J. Xu, Z. Li, J. Lowengrub, H. Zhao, A level set method for interracial flows with surfactant, J. Comput. Phys. 212 (2006) 590–616.
[39] J. Xu, Z. Li, J. Lowengrub, H. Zhao, Numerical study of surfactant-laden drop-drop interactions, Commun. Comput. Phys. 10 (2011) 453–473.
[40] J. Xu, Y. Yang, J. Lowengrub, A level-set continuum method for two-phase flows with insoluble surfactant, J. Comput. Phys. 231 (2012) 5897–5909.
[41] J. Xu, W. Ren, A level-set method for two-phase flows with moving contact line and insoluble surfactant, J. Comput. Phys. 263 (2014) 71–90.
[42] J. Zhang, D.M. Eckmann, P.S. Ayyaswamy, A front tracking method for a deformable intravascular bubble in a tube with soluble surfactant transport, J. Comput. Phys. 214 (2006) 366–396.
[43] H. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion, J. Comput. Phys. 127 (1996) 179–195.