ICES REPORT 18-20

September 2018

A coupled grid based particle method and implicit boundary integral method for two-phase flows with insoluble surfactant

by

S.-H. Hsu, J. Chu, M.-C. Lai and R. Tsai

The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: S.-H. Hsu, J. Chu, M.-C. Lai and R. Tsai, "A coupled grid based particle method and implicit boundary integral method for two-phase flows with insoluble surfactant," ICES REPORT 18-20, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, September 2018.

A coupled grid based particle and implicit boundary integral method for two-phase flows with insoluble surfactant

Shih-Hsuan Hsu [*] Jay Chu [†] Ming-Chih Lai [‡] Richard Tsai [§]

Abstract

We develop a coupled grid based particle and implicit boundary integral method for simulations of three-dimensional interfacial flows with the presence of insoluble surfactant. The grid based particle method (GBPM, Leung and Zhao [23]) tracks the interface by the projection of the neighboring Eulerian grid points and does not require stitching of parameterizations nor body fitted moving meshes. Using this GBPM to represent the interface, the surfactant equation defined on the interface is discretized naturally following a new volumetric constant-along-surface-normal extension approach (Chu and Tsai [4]). We first examine our scheme to solve the convection-diffusion equation for the problems with available analytical solutions. The numerical results demonstrate second-order accuracy of the scheme. We then perform a series of simulations for the interfacial flows with insoluble surfactant. The numerical results agree well with the ones obtained in theory, and are comparable with other numerical works in literature.

Keywords: interfacial flow; insoluble surfactant; closest point extensions; grid based particle method

1 Introduction

Surfactant is a chemical compound consisting of molecules with hydrophilic heads and hydrophobic tails. Surfactant adheres to the two-phase fluid interface and reduce the surface tension. It plays an important role in many industrial applications such as pharmaceutical, cosmetic, oil industries. The two-phase flow problems with surfactant draw a lot of attention, not only for the sake of its applications but also for the numerical point of interests. In

[*]Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 30010, Taiwan. E-mail: narisawa.am02g@g2.nctu.edu.tw

[†]Corresponding author. Department of Mathematics, National Tsing Hua University, 101 Section 2 Kuang-Fu Road, Hsinchu, Taiwan. E-mail: ccchu@math.nthu.edu.tw

[‡]Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 30010, Taiwan. E-mail: mclai@math.nctu.edu.tw

[§]Department of Mathematics and Institute for Computational Engineering and Sciences (ICES), The University of Texas at Austin, TX 78712, USA. E-mail: ytsai@ices.utexas.edu

order to predict the surfactant concentration on the fluid interface, it is necessary to solve a convection-diffusion equation defined on the evolving interface. It is a challenging task to accurately solve such type of problems that involve non-trivial moving interfaces embedded in three dimensions. In this work, we propose a numerical method to solve the three dimensional problem in an efficient and relatively simple fashion.

Solving PDEs on surfaces has been explored by many researchers for decades. Dziuk, Elliott et al. developed a series of work to solve PDEs on surfaces by finite element approaches [5, 6, 7]. The idea of the method is to approximate both surfaces and solutions on suitable finite element spaces and formulate the PDEs in the weak form. Error analysis and optimal convergence rate have been studied. We refer a review paper [8] for more detail. Other finite difference methods based on surface front-tracking triangular meshes to solve the surfactant equation in 3D have been developed in past years. Muradoglu and Tryggvason [18] used the front-tracking method. They first wrote the equation in the integral form on each front element and then converted the area integral of surface diffusion term into a line integral using the mathematical identity. The scheme is in spirit an explicit finite volume method to solve the surfactant convection-diffusion equation. De Jesus et. al. [19] used the implicit finite volume method developed in [21] to solve the surfactant equation and applied to the 3D two-phase flow with insoluble surfactant. Another front-tracking method uses the spherical harmonics expansion to represent the surface with adopting re-parametrization techniques to the surface meshes and corresponding surfactant concentration. Sorgentone and Tornberg [34] used this approach and coupled with the boundary integral method for Stokes flow to study surfactant-laden drop dynamics in 3D. While these front-tracking methods have the advantage of keeping the conservation of local surfactant mass, the Lagrangian grid restructuring or optimizing must be applied from time to time, and this increases the computational overhead. Recently, Seol et al. [16] have developed an equi-arclength parametrization technique to automatically control the Lagrangian meshes so that the total surfactant mass is conserved numerically. However, their scheme is limited to planar surfaces.

To avoid the difficulty of re-meshing process for surface representation, other embedding methods based on solving the equations on Eulerian domains have been developed. The main advantage of Eulerian embedding method to solve surface PDEs is no need to layout the Lagrangian mesh that moves and conforms to the moving interface. Consequently, there is no need to remesh nor discretize different versions of the surface PDEs on different patches and "glue" them together. A common approach for the Eulerian methods is to extend the surface quantity (surfactant) and equations defined on the interface to a narrowband around the surface and solve the extended PDEs in that narrowband by some simple and robust finite difference methods. The level set method proposed by Xu and Zhao [36] and the closest point method proposed by Ruuth and Merriman [30] belong to such category. Xu and his collaborators also applied their method to simulate the interfacial flows with insoluble surfactant [37] and soluble surfactant [38].

To track a moving surface, the implicit method such as level set method requires to solve the Hamilton-Jacobi equation in the entire Eulerian domain. Rather than using the implicit method, we here employ the grid based particle method (GBPM) proposed by Leung and Zhao [23] for surface representation. The GBPM is a numerical method to track an evolving surface which combines the advantages of the Lagrangian and Eulerian approaches. The surface is represented by meshless and non-parametrized Lagrangian markers explicitly us-

ing underlying Eulerian grid reference to avoid the re-meshing process for usual Lagrangian tracking method. The same as traditional Lagrangian method, the surface evolution is captured by solving an ordinary differential equation on each marker, so that it is relatively efficient for the same simulation time. In addition, the method can compute the projection point and geometric quantities accurately by local polynomial approximations.

Petras and Ruuth [29] coupled a modified grid based particle method and the closest point method to solve the convection-diffusion equation on moving surfaces. As with other closest point methods, their approach requires an extra "reinitialization" procedure at every time step. The procedure involves the closest point projection and interpolation to enforce the computed quantities being locally constant along surface normals. For systems with strong reaction terms and interfaces with high curvature relative to the underlying grid, the "non-surface intrinsic" reactions computed by the method could influence the values computed on the interface via the underlying diffusion. However, such nuances can be hard to observe if the width of the narrowband is thin and the frequency of the "reinitialization" sufficiently frequent.

Recently Chu and Tsai [4] propose a framework to extend surface PDEs into a tubular neighborhood in the embedding space. For a wide class of surface PDEs, the solutions of the properly extended versions are naturally constant-along-surface-normal without the need to enforce additional constraints. This saves some computation cost on interpolation and provides better stability for resulted system. In this paper, we shall refer to this framework as the IBIM framework. The term IBIM stands for Implicit Boundary Integral Method, which is a general approach to formulate volumetric extensions of boundary integrals, see [14, 15] in detail. For these above reasons, in this paper we develop a method using GBPM and IBIM to solve the convection-diffusion equation on an evolving surface and apply to simulate the three-dimensional interfacial flows with insoluble surfactant. The presented scheme enjoys the advantages of those two methods and is competitively accurate and efficient. Furthermore, this framework can be applied to electrohydrodynamic applications which require coupling with boundary integral solvers for electric potential computations.

The rest of the paper is organized as follows. Immediately in Section 2, we present the the governing equations that we solve numerically in this paper. The equations describe an immiscible two-phase fluid in which the interface between the two phases are dynamic and under the influence of surfactant. The numerical method includes two major coupled components, one for evolving the fluid as well as the interface and the other for evolving the surfactant concentration along the interface. These components are described in Section 3. We then propose a coupled GBPM-IBIM scheme to solve the convection-diffusion equation on an evolving surface and perform a series of numerical tests for accuracy in Section 4. In Section 5, we present some numerical results computed by the proposed method for a few three-dimensional test cases.

2 Interfacial flows with insoluble surfactant

In this section, we consider an incompressible two-phase flow in a fixed three-dimensional domain $\Omega = \Omega_1 \cup \Omega_2$ where the interface Σ separates Ω_1 (interior) from Ω_2 (exterior) and is a closed evolving surface. The interface is contaminated by an insoluble surfactant, which

changes the surface tension accordingly. We further assume that the two fluids have matched density and viscosity. Using the non-dimensionalization process described in [13], the governing equations in dimensionless form lead to the following single-fluid formulation as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = \frac{1}{Re} \Delta \mathbf{u} + \frac{1}{ReCa} \mathbf{f}, \quad \text{in } \Omega \tag{1}$$

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

$$\mathbf{f} = \Big(\nabla_s \sigma - 2\kappa\sigma \mathbf{n} \Big) \delta\left(d\right), \quad \text{in } \Omega \tag{3}$$

$$\sigma = 1 - \beta\Gamma, \quad \text{on } \Sigma \tag{4}$$

$$\frac{D\Gamma}{Dt} + \left(\nabla_s \cdot \mathbf{u}\right)\Gamma = \frac{1}{Pe_s} \Delta_s \Gamma, \quad \text{on } \Sigma \tag{5}$$

$$\frac{\partial \mathbf{X}}{\partial t} = \mathbf{u}(\mathbf{X}, t), \quad \text{on } \Sigma. \tag{6}$$

Eqs. (1)-(2) are the Navier-Stokes equations in which $\mathbf{u}$ is the fluid velocity and p is the pressure. The force $\mathbf{f}$ in Eq. (1) is the interfacial force that arises from the surface tension σ, which consists of Marangoni force $\nabla_s \sigma$ and capillary force $2\kappa\sigma\mathbf{n}$ as shown in Eq. (3). Here, δ is the Dirac delta function and d is the signed distance function to the interface, κ is the mean curvature of Σ, with the convention of positive sign when the surface is convex. The surface tension relating to the surfactant concentration is described by the Langmuir equation of state [31]. We adopt the linear approximation Eq. (4) as in [17]. Equation (5) is the convection-diffusion equation that governs the surfactant concentration along the interface [32]. Equation (6) shows that the interface moves along with the local fluid velocity determined by the Navier-Stokes equations. To complete the system, the above equations (1)-(6) should be accompanied with suitable initial conditions.

There are four dimensionless numbers: the Reynolds number Re describing the ratio between the inertial force and the viscous force, the capillary number Ca describing the strength of the surface tension, the parameter β, $0 \leq \beta < 1$, measuring the sensitivity of surface tension change to the surfactant concentration, and the surface Peclet number Pe_s measuring the ratio of surfactant diffusion and convection effects along the interface.

3 Components of the proposed method

We first introduce some notations usef in the paper. Assume Σ to be a C^2-closed surface in $\mathbb{R}^3$. For any small $\epsilon > 0$, we define an ϵ-narrowband of Σ as

$$T_\epsilon := \{\mathbf{x} \in \mathbb{R}^3 : \min_{\mathbf{y} \in \Sigma} |\mathbf{x} - \mathbf{y}| < \epsilon\}. \tag{7}$$

The closest point mapping $P_\Sigma : T_\epsilon \mapsto \Sigma$ is then given by

$$P_\Sigma(\mathbf{x}) = \arg\min_{\mathbf{y} \in \Sigma} |\mathbf{x} - \mathbf{y}|. \tag{8}$$

The closest point map P_Σ is well-defined on T_ϵ, provided $\epsilon \in (0, \kappa_\infty^{-1})$, where κ_∞ is an upper bound of the curvatures of Σ.

3.1 The Grid Based Particle Method for tracking the interface

The Grid Based Particle Method (GBPM) [23] is a numerical method to track an evolving surface using Cartesian grids. The main idea is to represent the surface by meshless and non-parametrized Lagrangian particles and to compute the geometric information by local parametrization with suitable polynomial approximations. Every Lagrangian particle is in fact the closest point of some Cartesian grid node lying inside a thin narrowband around the surface. Given that the underlying Cartesian grid resolves the surface well, this approach provides a quasi-uniform point cloud sampling of the surface. The surface position is then updated by moving the particles according to some given velocity which this step is exactly as in the traditional Lagrangian method. After each time step, the surface is "resampled" in the sense that a new set of Lagrangian particles closest to the Cartesian grid nodes within the narrowband is constructed via interpolation of the old set of Lagrangian particles. In addition, GBPM provides a natural way to compute the geometric information accurately which is essential for IBIM. We will describe more details about IBIM in the following section.

Throughout this paper, we use the GBPM to track the surface motion and compute the corresponding geometric quantities, such as the principle curvatures, that are used in later computation. A brief numerical algorithm at each time step is outlined as follows.

1. **Initialization.** Collect all grid points $\mathbf{x}$ that are within T_ϵ as defined before. The surface Σ is represented by the closest points to surface of grid points in T_ϵ, that is, $\Sigma = \{P_\Sigma(\mathbf{x}) : \mathbf{x} \in T_\epsilon\}$.

2. **Evolving surface.** Evolve the surface by solving the advection equation at each Lagrangian particles in Σ.

3. **Resampling surface.** After the evolution, those Lagrangian particles are no longer to be closest points so we need to compute the new closest points to the surface for each grid points in T_ϵ by performing some local polynomial reconstruction.

4. **Updating the computational tube.** We first activate the grid points that are neighbors of the computational tube and find their corresponding closest points. Then we deactivate the grid points that have the distance to the surface larger than the given tube radius ϵ and obtain the new computational tube domain T_ϵ. As a result, this yields a new closest point representation of the surface Σ.

The original GBPM in [23] used the tube radius $\epsilon = 1.1\Delta x \sim 1.5\Delta x$ to track the surface motion. However, due to the bi-cubic interpolation used in the closest point based methods for solving surface PDEs (including the method used in this paper), one needs to use narrowbands with larger radius. In [30], Ruuth and Merriman gave an estimated tube radius for interpolating the degree p of Lagrange polynomial in d-dimensional space as

$$\epsilon = \sqrt{(d-1)\left(\frac{p+1}{2}\right)^2 + \left(1+\frac{p+1}{2}\right)^2}\Delta x.$$

For instance, the interpolation of degree 3 polynomial in three-dimensional space leads the tube radius of $\epsilon = 4.1\Delta x$ which is about two times larger than the one used in the standard GBPM.

In the resampling step, the inaccurate polynomial fitting occurs frequently when the grid point has a large distance from surface. In the standard GBPM, these grid points are simply removed from T_ϵ, i.e. grid point deactivation. Recently, Petras and Ruuth [29] proposed a modified GBPM using an osculating circle and sphere reconstruction to replace the local polynomial reconstruction when grid point deactivation occurs. However, this approach degrades the computing accuracy of the closest points and evaluations of geometric quantities.

In order to preserve the desired accuracy of the closest points and geometric quantities, we use the least squares polynomial fitting for local reconstruction of the surface throughout this work. Notice that, the inaccurate polynomial fitting occurs when the chosen particles in local reconstruction are insufficient. We can simply increase the number of candidates for choosing particles to solve the issue. Figure 1 illustrates the local reconstruction using two different regions of neighboring grid points. It shows by enlarging the chosen region, the local reconstruction is more accurate. In our numerical experiments, we use the region of $\epsilon + \Delta x$ for choosing particles in local reconstruction. We also modify the way to obtain geometric quantities. In the standard GBPM [23], the first and second fundamental forms are computed from local polynomials to approximate both the mean curvature and the Gaussian curvature at the closest point of the surface. Instead of computing fundamental forms, we evaluate the principal curvatures k_i first. Then the principle directions $\mathbf{t}_i$ can be obtained from the eigenvectors corresponding to the eigenvalue k_i of the shape operator which is given by the Weingarten equations.

3.2 The Implicit Boundary Integral Method for solving surface PDEs

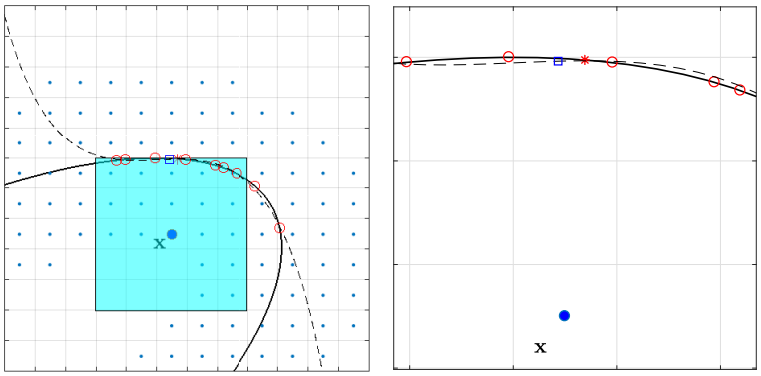
The Implicit Boundary Integral Method (IBIM) is a general framework for developing numerical methods for a class of integro-differential operators on non-parametrically defined surfaces or curves in $\mathbb{R}^n$, for instance implicit surfaces defined by level set method or closest point mapping, see e.g. [14, 15, 4]. The main idea is to extend the operators via the closest point map to the surfaces. By replacing surface gradient ∇_s and surface divergence $\nabla_s \cdot$ by Euclidean gradient and divergence with suitable tensor coefficients, the method admits solutions which are constant along the surface normals, if the initial data has the same property. This property provides an equivalence between solution computed in a thin narrowband around the surface and the solution to the intrinsic surface PDEs. The discretization of the extended PDEs can be done in Cartesian grids with finite difference or finite element schemes. We briefly illustrate the method in this subsection.

For any function f defined on Σ, we define its normal extension $\overline{f}$ on T_ϵ by $\overline{f}(\mathbf{x}) = f(P_\Sigma(\mathbf{x}))$ for any $\mathbf{x} \in T_\epsilon$. It can be shown that for any $f \in H^1(\Sigma)$, we have

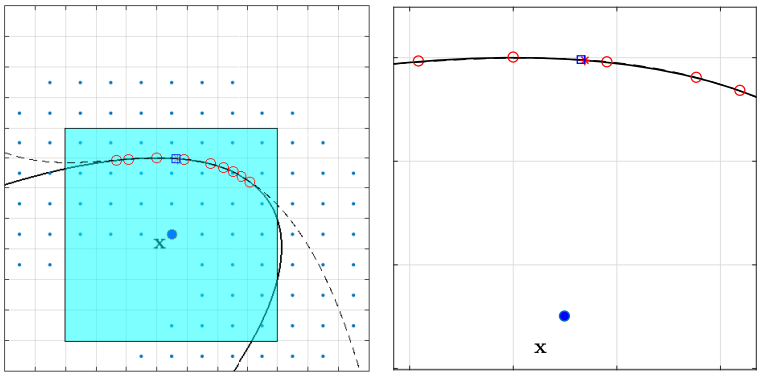
$$\nabla_s f(P_\Sigma(\mathbf{x})) = A \nabla \overline{f}(\mathbf{x}), \quad \text{for all } \mathbf{x} \in T_\epsilon, \tag{9}$$

where $A(\mathbf{x}; \mu) = A_0(\mathbf{x}) + \mu A_1(\mathbf{x})$. Here, $A_0(\mathbf{x})$ is the scaling tensor, $A_1(\mathbf{x})$ is an extra term to make the matrix tensor A non-degenerate, and μ is any real number. Two matrix tensors are calculated by

$$\begin{aligned} A_0(\mathbf{x}) :=& \sigma_1^{-1} \mathbf{t}_1 \otimes \mathbf{t}_1 + \sigma_2^{-1} \mathbf{t}_2 \otimes \mathbf{t}_2, \\ A_1(\mathbf{x}) :=& \mathbf{n} \otimes \mathbf{n}, \end{aligned}$$



(a)



(b)

Figure 1: The local cubic polynomial reconstruction (dashed curve $--$) for the surface (solid curve $-$) using Lagrangian markers (red circles $\circ$) inside the square region centered at $\mathbf{x}$. $(*)$ and $(\square)$ are the closest points to $\mathbf{x}$ on the surface and reconstruction surface, respectively. The right column are the zoom-ins of Figure (a) and (b), respectively. Figure (a) and (b) use different sizes of regions for reconstruction. This comparison shows that the accuracy of local reconstruction can be improved by enlarging the interpolation region.

where $\mathbf{t}_1$, $\mathbf{t}_2$ are the two orthonormal tangent vectors corresponding to the directions that yield the principle curvatures of Σ, $\mathbf{n}$ is the unit normal vector of Σ, σ_1 and σ_2 are two largest singular values of DP_Σ, where $DP_\Sigma(\mathbf{x})$ is the Jacobian matrix of $P_\Sigma(\mathbf{x})$ [15].

Similarly, we can extend the surface divergence for a vector field $\mathbf{F}$ defined on Σ to T_ϵ by

$$(\nabla_s \cdot \mathbf{F})(P_\Sigma(\mathbf{x})) = J^{-1}\nabla \cdot (JA\overline{\mathbf{F}}(\mathbf{x})), \qquad \text{for all } \mathbf{x} \in T_\epsilon, \tag{10}$$

where J is the Jacobian that comes from the change of variables by the closest point mapping and can be computed by $J = \sigma_1\sigma_2$. In the present GBPM surface representation, we can simply compute those singular values by σ_i by $\sigma_i(\mathbf{x}) = 1 + d(\mathbf{x})k_i(P_\Sigma(\mathbf{x}))$ directly, where $d(\mathbf{x})$ is the signed distance between $\mathbf{x}$ and $P_\Sigma(\mathbf{x})$, and $k_i(P_\Sigma(\mathbf{x}))$ is the corresponding principal curvature at the surface. In particular, the surface Laplacian of f can be extended to T_ϵ by simply substituting the surface gradient of f defined in Eq. (9) into

$$\Delta_s f(P_\Sigma(\mathbf{x})) = J^{-1}\nabla \cdot (JA^2\nabla\overline{f}(\mathbf{x})), \qquad \text{for all } \mathbf{x} \in T_\epsilon.$$

To deal with Neumann boundary condition on ∂T_ϵ, a local interpolation strategy is used to close the system. We project the ghost grid points outside of T_ϵ into it along the normal direction and interpolate the projected position by nearby inner grid points. It can be shown the closure is second-order accurate and stable for parabolic equations. See [4] for detail.

3.3 The coupled scheme for interfacial flows

In the following, we describe our proposed numerical scheme for solving Eqs. (1)-(6). We assume the computational domain to be a rectangular domain $\Omega = [a, b] \times [c, d] \times [e, f]$. Within this domain, a uniform lattice grid with mesh width h is employed. The velocity components u, v and w are defined at usual staggered MAC grid [11] as

$$(x_{i-1/2}, y_j, z_k) = (a + (i-1)h, c + (j-1/2)h, e + (k-1/2)h),$$
$$(x_i, y_{j-1/2}, z_k) = (a + (i-1/2)h, c + (j-1)h, e + (k-1/2)h),$$
$$(x_i, y_j, z_{k-1/2}) = (a + (i-1/2)h, c + (j-1/2)h, e + (k-1)h).$$

The pressure p is defined at the cell center labelled as

$$(x_i, y_j, z_k) = (a + (i-1/2)h, c + (j-1/2)h, e + (k-1/2)h).$$

Notice that, the Cartesian grid points used by GBPM to represent the interface and by IBIM for solving surfactant concentration are all chosen at cell center points $\mathbf{x} = (x_i, y_j, z_k)$.

Let Δt be the time step size, and n be the time step index. At the beginning of each time step n, the fluid velocity $\mathbf{u}^n$, the interface position $\mathbf{X}^n$, and the surfactant concentration Γ^n are all given. More precisely speaking, in the GBPM framework, the interface position Σ^n are those projection points of the grid points $\mathbf{x} = (x_i, y_j, z_k)$ within the narrowband T_ϵ so that the signed distance function d and the mean curvature κ of the interface can be easily obtained. The time advancement of one time step can be summarized as follows.

1. Compute the surface tension and interfacial force

   $$\sigma^n = 1 - \beta\Gamma^n, \quad \mathbf{f}^n = \Big(A\nabla\sigma^n - 2\kappa^n\sigma^n\mathbf{n}^n\Big)\delta_h(d^n),$$

   directly at each grid point $\mathbf{x}$ in the neighborhood of interface Σ^n. Notice that, all the surface geometric quantities in the above equation are evaluated at corresponding closest point $\mathbf{X} = P_\Sigma(\mathbf{x})$. Since the force $\mathbf{f}^n$ is defined at the cell center point $\mathbf{x}$, we need to further interpolate the value at the corresponding staggered grid point used in the following Navier-Stokes solver.

2. Solve the Navier-Stokes equations using the second-order accurate projection method proposed in [10] to update the new velocity $\mathbf{u}^{n+1}$.

   $$\frac{3\mathbf{u}^* - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\Delta t} + 2\left(\mathbf{u}^n \cdot \nabla_h\right)\mathbf{u}^n - \left(\mathbf{u}^{n-1} \cdot \nabla_h\right)\mathbf{u}^{n-1} + \nabla_h p^n$$
   $$= \frac{1}{Re}\Delta_h\mathbf{u}^* + \frac{\mathbf{f}^n}{Re\,Ca}, \quad \mathbf{u}^*|_{\partial\Omega} = \mathbf{u}_b,$$
   $$\Delta_h p^* = \frac{3}{2\Delta t}\nabla_h \cdot \mathbf{u}^*, \quad \frac{\partial p^*}{\partial \mathbf{n}}|_{\partial\Omega} = 0,$$
   $$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{2\Delta t}{3}\nabla_h p^*, \quad \nabla_h p^{n+1} = \nabla_h p^* + \nabla_h p^n - \frac{2\Delta t}{3Re}\Delta_h(\nabla_h p^*).$$

3. Move the interface by GBPM and solve the surfactant concentration equation by IBIM with Strang splitting technique [35] as

$$\Gamma^{n+1} = \mathcal{A}_{\Delta t/2}\mathcal{B}^{n+1/2}\mathcal{A}_{\Delta t/2}\Gamma^n, \tag{11}$$

where $\mathcal{A}$ represents the simple transport operator and $\mathcal{B}$ represents the discrete convection-diffusion operator for the surfactant concentration. The complete numerical details are given in the following Section 4.

4 Approximation of the convection-diffusion equation on an evolving surface

In this section, we elaborate the numerical details in Eq. (11) and introduce a coupled GBPM and IBIM to solve the convection-diffusion equation on an evolving surface. Here, we assume the closed surface $\Sigma(t)$ can be immersed in a fixed three-dimensional domain Ω. It is deformable and moving with a given velocity $\mathbf{u}$ (or obtained from the Navier-Stokes equations in previous section) as

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = \mathbf{u}(\mathbf{X}, t) \quad \text{on } \Sigma(t). \tag{12}$$

Along this evolving surface $\Sigma(t)$, the surfactant concentration follows the convection-diffusion equation

$$\frac{D\Gamma}{Dt} + (\nabla_s \cdot \mathbf{u})\,\Gamma = \frac{1}{Pe_s}\Delta_s\Gamma \quad \text{on } \Sigma(t), \tag{13}$$

where $\frac{D}{Dt}$ is material derivative, ∇_s and $\Delta_s = \nabla_s \cdot \nabla_s$ are the surface gradient and surface Laplacian defined as before.

Since the surface is tracked explicitly by the closest points in the GBPM setting, we denote Σ^n the set of closest points used for surface representation at the time level $n\Delta t$, and the function Γ defined at Σ^n denoted by Γ^n. Instead of solving the convection-diffusion equation (13) directly, we adopt the Strang splitting technique [35] to solve the equation. We first write the equation into the simple transport equation

$$\frac{D\Gamma}{Dt} = 0, \tag{14}$$

and the convection-diffusion equation

$$\frac{\partial\Gamma}{\partial t} + (\nabla_s \cdot \mathbf{u})\,\Gamma = \frac{1}{Pe_s}\Delta_s\Gamma. \tag{15}$$

The Strang splitting consists of the following three steps:

Step 1: Solve Eq. (14) by GBPM with one half of time step.

First, we update the new surface position $\Sigma^{n+1/2}$ and obtain the value of surface function $\hat{\Gamma}^n = \mathcal{A}_{\Delta t/2}\Gamma^n$ at $\Sigma^{n+1/2}$ by GBPM which involves the surface moving, re-sampling and interpolation presented in Section 3.1.

Step 2: Solve Eq. (15) by IBIM with one time step.

We decompose the velocity into the tangential and normal components as $\mathbf{u} = \mathbf{u}_t + \mathbf{u}_n$. Then, the surface divergence of velocity can be rewritten as

$$\nabla_s \cdot \mathbf{u} = \nabla_s \cdot \mathbf{u}_t + (\mathbf{u} \cdot \mathbf{n})\kappa, \tag{16}$$

where $\kappa = \nabla_s \cdot \mathbf{n}$ is the mean curvature of the surface. Applying the IBIM to Eq. (15), we obtain the embedding equation in T_ϵ

$$\frac{\partial \Gamma}{\partial t} + \left(J^{-1}\nabla \cdot (JA\mathbf{u}_t) + (\mathbf{u} \cdot \mathbf{n})\kappa\right)\Gamma = \frac{1}{Pe_s} J^{-1}\nabla \cdot (JA^2 \nabla \Gamma).$$

Then we use Crank-Nicolson scheme for the time integration and second-order central difference for the spatial discretization to update the surface function $\hat{\Gamma}^{n+1}$ as

$$\begin{aligned}\frac{\hat{\Gamma}_{ij}^{n+1} - \Gamma_{ij}^{n}}{\Delta t} &+ \frac{1}{2}\left(J^{-1}\nabla_h \cdot (JA\mathbf{u}_t) + (\mathbf{u} \cdot \mathbf{n})\kappa\right)(\hat{\Gamma}_{ij}^{n+1} + \Gamma_{ij}^{n}) \\ &= \frac{1}{2Pe_s} J^{-1}\nabla_h \cdot (JA^2 \nabla_h (\hat{\Gamma}_{ij}^{n+1} + \Gamma_{ij}^{n})).\end{aligned}$$

In short, the above scheme can be written as $\hat{\Gamma}^{n+1} = \mathcal{B}^{n+1/2}\Gamma^n$.

Step 3: Solve Eq. (14) by GBPM with one half of time step.

We repeat the procedure in Step 1 by using $\Sigma^{n+1/2}$ and $\hat{\Gamma}^n$ as the initial conditions and update the new surface position Σ^{n+1} and function Γ^{n+1} by $\Gamma^{n+1} = \mathcal{A}_{\Delta t/2}\hat{\Gamma}^{n+1}$.

4.1 Numerical results

In order to validate the proposed numerical scheme, we perform a series of tests to check if our algorithm is correct for benchmark problems. Throughout this section, we choose a uniform Cartesian mesh $h = \Delta x = \Delta y = \Delta z$ and compute the corresponding L_∞, L_1 and L_2 errors using the following formulas

$$\|e_h\|_\infty = \max_{\mathbf{x}_i \in T_\epsilon} e_h(P_\Sigma(\mathbf{x}_i)), \tag{17}$$

$$\|e_h\|_1 = \int_{T_\epsilon} e_h\left(P_\Sigma(\mathbf{x})\right)\delta_h\left(d\left(\mathbf{x}\right)\right)J(\mathbf{x})\mathrm{d}\mathbf{x}, \tag{18}$$

$$\|e_h\|_2 = \left(\int_{T_\epsilon} e_h^2\left(P_\Sigma(\mathbf{x})\right)\delta_h\left(d\left(\mathbf{x}\right)\right)J(\mathbf{x})\mathrm{d}\mathbf{x}\right)^{\frac{1}{2}}, \tag{19}$$

where the error $e_h(P_\Sigma(\mathbf{x})) = |\Gamma(P_\Sigma(\mathbf{x})) - \Gamma_h(P_\Sigma(\mathbf{x}))|$ is defined on those grid points within the narrowband T_ϵ. Here, $d(\mathbf{x})$ is the signed distance function obtained from GBPM, and $\delta_h(x)$ is the discrete delta function defined by

$$\delta_h(x) = \begin{cases} \frac{1}{4h} + \frac{1}{4h}\cos\frac{\pi x}{2h} & \text{if } -2h < x < 2h \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

Table 1: The errors and their convergent rates for an expending circle case at $t = 0.5$.

h	$\|e_h\|_\infty$	rate	$\|e_h\|_1$	rate	$\|e_h\|_2$	rate
1/10	2.46×10^{-4}	–	3.42×10^{-3}	–	7.94×10^{-4}	–
1/20	5.98×10^{-5}	2.04	8.39×10^{-4}	2.03	1.95×10^{-4}	2.03
1/40	1.48×10^{-5}	2.02	2.06×10^{-4}	2.02	4.81×10^{-5}	2.02
1/80	3.69×10^{-6}	2.00	5.13×10^{-5}	2.01	1.19×10^{-5}	2.00
1/160	9.26×10^{-7}	2.00	1.28×10^{-5}	2.00	3.00×10^{-6}	2.00

Note that, the integrals in Eq. (18)-(19) are approximated by the trapezoidal rule on Cartesian grid points within T_ϵ. The details for computing integrals over a curve or surface via closest point mapping can also be found in [15].

In the following, we consider the convection-diffusion equation (13) on the cases of 2D curves and 3D surfaces where the motions of curves and surfaces are known in priori. For simplicity, we set the surface Peclet number $Pe_s = 1$.

Example 1: A 2D expanding circle

Although the description of the present scheme is based on 3D formulation, the method can be applied to 2D in a simpler manner. As the first test, we follow the example given in [9] by considering the interface as a unit circle with velocity field $\mathbf{u} = 5\mathbf{n}$. Therefore, the interface is an expanding circle and can be represented by $r(t) = 1 + 5t$. The function

$$\Gamma(x, y, t) = e^{\frac{4}{5r(t)}} \frac{xy}{r^3(t)}$$

satisfies the exact solution of Eq. (13) with initial condition $\Gamma(x, y, 0)$ on the unit circle. The solutions are computed up to time $t = 0.5$ with time step size $\Delta t = h/5$. The numerical errors and their convergence rates for different mesh widths are shown in Table 1. One can see that the numerical results show second-order convergence clearly.

Example 2: A 2D unit circle under simple shear flow

In this test, we consider a unit circle under simple shear flow $\mathbf{u} = (y, 0)$. Thus, the interface evolves to a slanted ellipse of the form

$$\Sigma(t) = \Big\{\mathbf{x}(\theta, t) = \big(\cos(\theta) + t\sin(\theta), \sin(\theta)\big),\ 0 \le \theta \le 2\pi\Big\}.$$

Along the interface $\Sigma(t)$, we set the function $\Gamma(\theta, t) = e^{-t}\cos(\theta)$ as the exact solution of Eq. (13) with an extra source function g defined by $g = \frac{D\Gamma}{Dt} + (\nabla_s \cdot \mathbf{u})\,\Gamma - \Delta_s\Gamma$. Again, the initial interface is a unit circle with initial condition $\Gamma(\theta, 0)$. The solutions are computed up to time $t = 1$ with time step size $\Delta t = h/2$. We set the largest mesh width $h = 1/40$ to make sure that the closest point mapping is well-defined in T_ϵ during computations. The numerical errors and their convergence rates are shown in Table 2 which shows the clean second-order convergence as well.

Table 2: The errors and their convergent rates for the convection-diffusion equation under the simple shear flow at $t = 1$.

h	$\|e_h\|_\infty$	rate	$\|e_h\|_1$	rate	$\|e_h\|_2$	rate
1/40	2.02×10^{-4}	–	5.14×10^{-4}	–	2.32×10^{-4}	–
1/80	4.61×10^{-5}	2.13	1.23×10^{-4}	2.06	5.51×10^{-5}	2.07
1/160	1.15×10^{-5}	2.01	3.05×10^{-5}	2.01	1.36×10^{-5}	2.02
1/320	2.86×10^{-6}	2.01	7.61×10^{-6}	2.00	3.40×10^{-6}	2.00
1/640	7.16×10^{-7}	2.00	1.90×10^{-6}	2.00	8.50×10^{-7}	2.00

Table 3: The errors and their convergent rates for the convection-diffusion equation on the expending sphere at $t = 0.5$.

h	$\|e_h\|_\infty$	rate	$\|e_h\|_1$	rate	$\|e_h\|_2$	rate
1/10	6.25×10^{-4}	–	1.45×10^{-2}	–	2.30×10^{-3}	–
1/20	1.54×10^{-4}	2.02	3.56×10^{-3}	2.02	5.74×10^{-4}	2.00
1/40	3.83×10^{-5}	2.01	8.84×10^{-4}	2.01	1.42×10^{-4}	2.01
1/80	9.63×10^{-6}	1.99	2.21×10^{-4}	2.00	3.57×10^{-5}	2.00

Example 3: A 3D expanding sphere

In this example, we consider the case of an expanding sphere with the velocity $\mathbf{u} = 2\mathbf{n}$. Thus, the expanding sphere can be written in terms of radius function $r(t) = 1 + 2t$. Along the surface, we define the function

$$\Gamma(x, y, z, t) = e^{1/r(t)} \frac{z}{r^2(t)},$$

which is the exact solution of Eq. (13) with the initial condition $\Gamma(x, y, z, 0)$ on the unit sphere. The solutions are computed up to the time $t = 0.5$ with time step size $\Delta t = h/2$. The numerical errors and their convergence rates are shown in Table 3 which again show clean second-order convergence for different norms too.

Example 4: An oscillating ellipsoid

In this test, we consider the case of an oscillating ellipsoid developed in [6],

$$\Sigma(t) = \left\{ \mathbf{x} = (x, y, z) : \frac{x^2}{a^2(t)} + y^2 + z^2 = 1 \right\},$$

where $a(t) = \sqrt{1 + \sin(t)/4}$, and associated velocity field $\mathbf{u} = \left(\frac{a'(t)}{a(t)} x, 0, 0 \right)$. More generally, the oscillating ellipsoid has the parametric form

$$\mathbf{x}(\theta, \phi, t) = \Big(a(t) \sin(\theta) \cos(\phi), \sin(\theta) \sin(\phi), \cos(\theta) \Big),$$

Table 4: The errors and their convergent rates for the convection-diffusion equation on the oscillating ellipsoid at $t = 4$.

h	$\|e_h\|_\infty$	rate	$\|e_h\|_1$	rate	$\|e_h\|_2$	rate
1/10	1.35×10^{-4}	–	5.29×10^{-4}	–	1.61×10^{-4}	–
1/20	2.62×10^{-5}	2.37	1.30×10^{-4}	2.02	4.14×10^{-5}	1.96
1/40	6.12×10^{-6}	2.10	3.20×10^{-5}	2.03	1.03×10^{-5}	2.01
1/80	1.45×10^{-6}	2.08	7.92×10^{-6}	2.02	2.55×10^{-6}	2.01

where $0 \le \theta < \pi$ and $0 \le \phi < 2\pi$. The exact solution is defined by $\Gamma(\theta, \phi, t) = a(t)\sin^2(\theta)\cos(\phi)\sin(\phi)$ on the interface $\Sigma(t)$ and an extra source term $g = \frac{D\Gamma}{Dt} + (\nabla_s \cdot \mathbf{u})\,\Gamma - \Delta_s\Gamma$ is added so that the surface function Γ satisfies Eq. (13). The solutions are computed up to the time $t = 4$ with time step size $\Delta t = h$. The numerical errors and their convergence rates are shown in Table 4 which confirms the desired second-order convergence again.

5 Numerical results on interfacial flows

5.1 An ellipsoid droplet in quiescent flow

As a first test, we consider an ellipsoid droplet with the interface $\Sigma(0) = \{x^2 + y^2/(1.6)^2 + z^2/(0.4)^2 = 1\}$ immersed in a quiescent flow initially. The computational domain $\Omega = [-4, 4] \times [-4, 4] \times [-4, 4]$ with mesh size $h = 1/32$, and the time step size $\Delta t = h/4$. The initial surfactant concentration is set to be $\Gamma_0 = 1$ along the interface. The dimensionless numbers are chosen as $Re = 0.1, Ca = 0.1, \beta = 0.6$, and $Pe_s = 1$. The computations are up to time $t = 4$. Since the surfactant is insoluble, the total surfactant mass must be conserved along the interface. Meanwhile, the fluid is incompressible, so the total volume of the droplet must be conserved theoretically as well. The total surfactant mass $M(t)$ and total volume of the droplet $V(t)$ can be computed by

$$M(t) = \int_\Omega \Gamma\left(P_\Sigma(\mathbf{x})\right)\delta_h\left(d\left(\mathbf{x}\right)\right)J(\mathbf{x})\mathrm{d}\mathbf{x}, \quad V(t) = \int_\Omega \left(1 - H_h\left(d\left(\mathbf{x}\right)\right)\right)J(\mathbf{x})\mathrm{d}\mathbf{x},$$

where the smoothing delta function δ_h is defined in Eq. (20) so the corresponding smoothing Heaviside function is

$$H_h(x) = \begin{cases} 0, & \text{if } x \le -2h \\ \frac{x}{4h} + \frac{1}{2\pi}\sin\frac{\pi x}{2h}, & \text{if } -2h < x < 2h \\ 1, & \text{if } x \ge 2h. \end{cases}$$

Fig. 2(a) shows the relative error of total surfactant mass $E_M = \frac{|M(t)-M_0|}{M_0}$ while Fig. 2(b) shows the relative error of total droplet volume $E_V = \frac{|V(t)-V_0|}{V_0}$. Here, M_0 and V_0 are the initial total surfactant mass and total droplet volume, respectively. One can immediately see that the relative error of total surfactant mass is less than 0.1% and the one of total droplet volume is less than 0.05% up to time $t = 4$. This confirms that our present GBPM-IBIM

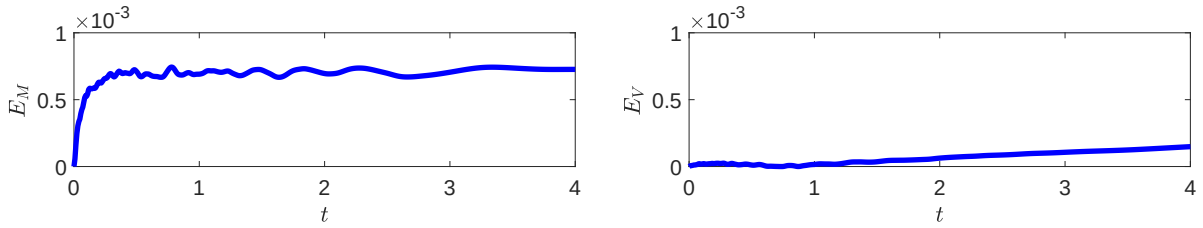


Figure 2: Relative errors of (a) total surfactant mass and (b) droplet volume.

method preserves the total surfactant mass and droplet volume quite well without performing any re-scalings during the computation.

In order to see the influence of surfactant on the interfacial behaviors, it is natural to compare the droplet dynamics between the clean interface (no surfactant) by choosing $\beta = 0$ and the one with surfactant. Fig. 3 shows the time evolutionary plots of the droplet at different times for the cases of clean interface (top, $\beta = 0$) and the one with surfactant (bottom, $\beta = 0.6$). Since the surfactant reduces the surface tension, the retreating force of the contaminated droplet is significantly weaker than the one of without surfactant. As a result, the contaminated interface lags the drop oscillatory motion than the clean one; see the comparisons at $t = 0.25, 0.5, 1.5$. Nevertheless, at final time $t = 4$, both drops evolve to the same spherical equilibrium shape eventually. This can also be confirmed by checking the time evolutionary surface area as shown in Fig. 4 (a), where the steady surface area is $A_s = 4\pi r_s^2 = 4\pi(\frac{3V_0}{4\pi})^{2/3}$. One can also see in Fig. 4 (b) that the minimum and maximum values of the surfactant concentration converge to the equilibrium value $\Gamma_s = A_0/A_s$, where A_0 is the initial surface area of the droplet.

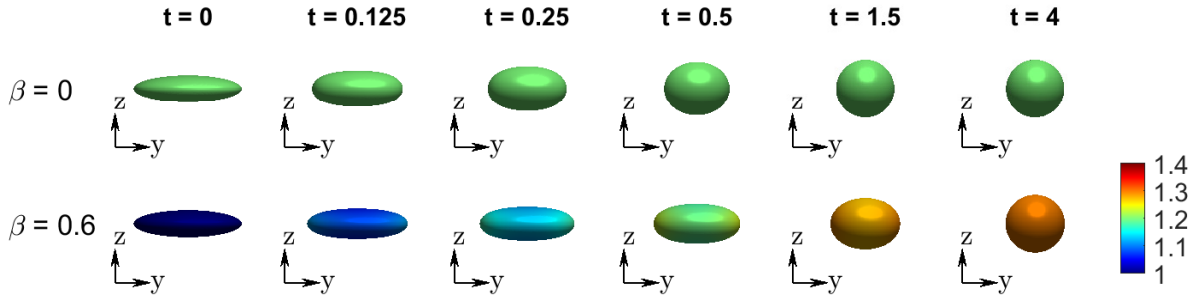


Figure 3: The time evolutionary plots of droplet in a quiescent flow.

5.2 Effect of capillary number

In this test, we study the effect of capillary number on the droplet dynamics under simple shear flow. We put a unit spherical droplet in the center of the computational domain $\Omega = [-4, 4] \times [-8, 8] \times [-4, 4]$ initially and apply the uniform boundary condition $\mathbf{u} = (0, z, 0)$ at $z = \pm 8$. The mesh width is $h = 1/16$, and the time step size is $\Delta t = h/4$. The initial surface concentration is chosen to be uniformly distributed $\Gamma_0 = 1$. The dimensionless numbers

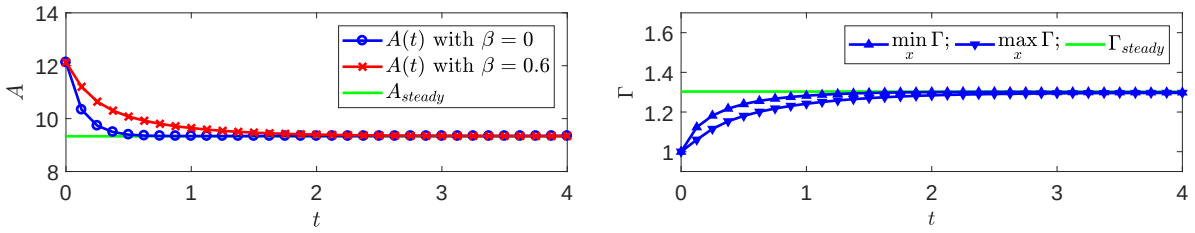


Figure 4: (a) The time evolutionary surface area plots for both clean and contaminated interfaces. (b) The time evolutionary plots of minimum and maximum surfactant concentrations along the contaminated interface.

are chosen as $Re = 1, \beta = 0.1, Pe_s = 1$ but we vary the capillary number $Ca = 0.2, 0.3$ and 0.4. It is known from previous numerical studies such as [24, 17] that as the capillary number increases, the droplet is elongated more so the deformation becomes larger. One can measure the deformation by the factor $D = \frac{L-B}{L+B}$, where L and B are the lengths of major and minor axis of the droplet, respectively. Fig. 5 shows the droplet morphology together with surfactant concentration at different times for three different capillary numbers. One can see that for smaller capillary number $Ca = 0.2, 0.3$, the droplet tends to be in equilibrium shape after $t = 6$ while the one with $Ca = 0.4$ elongate more even after $t = 12$. These droplet behaviors have been confirmed by the time evolutionary plots of the deformation factor as shown in Fig. 6.

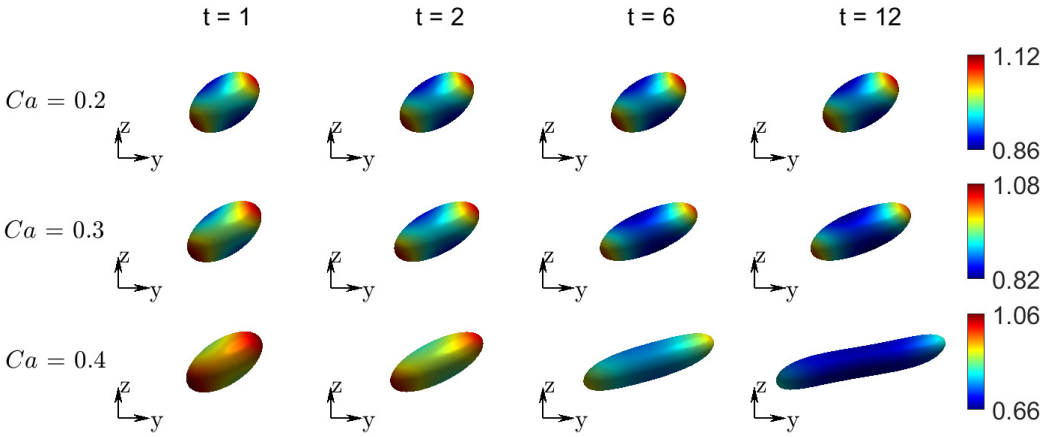


Figure 5: The droplet morphology together with surfactant concentration at different times for $Ca = 0.2, 0.3, 0.4$.

5.3 Comparison with small deformation theory

As shown in previous test, we know that the droplet under shear flow tends to become an equilibrium shape when the capillary number is small. Therefore, we can compare the present numerical results with the ones obtained from the small deformation theory [33, 24] in which

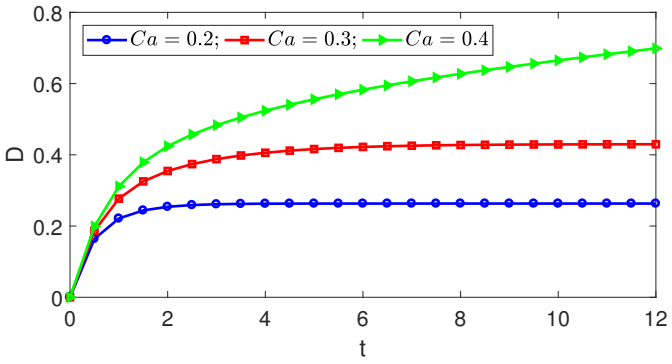


Figure 6: The time evolutionary plots of the deformation factor for $Ca = 0.2, 0.3, 0.4$.

the drop deformation is assumed to be small (smaller Ca) and the surfactant concentration is assumed to be nearly uniform. In such case, the deformation factor is given by

$$D = \frac{5}{8}\frac{35+4\epsilon}{20+2\epsilon}\frac{Ca}{1-\beta}, \tag{21}$$

where $\epsilon = \frac{Pe_s\beta}{Ca}$.

In this simulation, the computational domain setup is the same as the previous test while we choose smaller Reynolds number $Re = 0.01$ and Peclet number $Pe_s = 0.1$. Fig. 7 (a) shows the deformation factor D versus the capillary number for the present simulations and the theoretical results obtained from the relation (21). One can immediately see that the present results agree with the theoretical ones pretty well for the cases of $\beta = 0$ (clean interface) and $\beta = 0.5$ (contaminated interface with surfactant) when the capillary number is below 0.2. In addition, one can confirm again that given the same capillary number, the deformation for the case of with surfactant $\beta = 0.5$ is larger than the one of without surfactant.

When the droplet is in equilibrium, the shape does not change and the major axis will align with flow direction and forms an inclination angle θ (the angle between the major axis and the shear flow direction). The flow inside the droplet is rotational. In [3], for the case of the droplet without surfactant, the authors derived the relation between the angle and the capillary number as

$$\theta = \frac{\pi}{4} + \frac{35}{32}Ca. \tag{22}$$

Fig. 7 (b) shows the inclinational angle θ versus the capillary number Ca for the present simulations and the theoretical results obtained from Eq. (22). Again, one can see that our numerical results are in a good agreement with the theoretical ones for the case of $\beta = 0$. Meanwhile, from this figure, one can also see that at given capillary number, the inclinational angle for the case of with surfactant $\beta = 0.5$ is greater than the one of without surfactant $\beta = 0$.

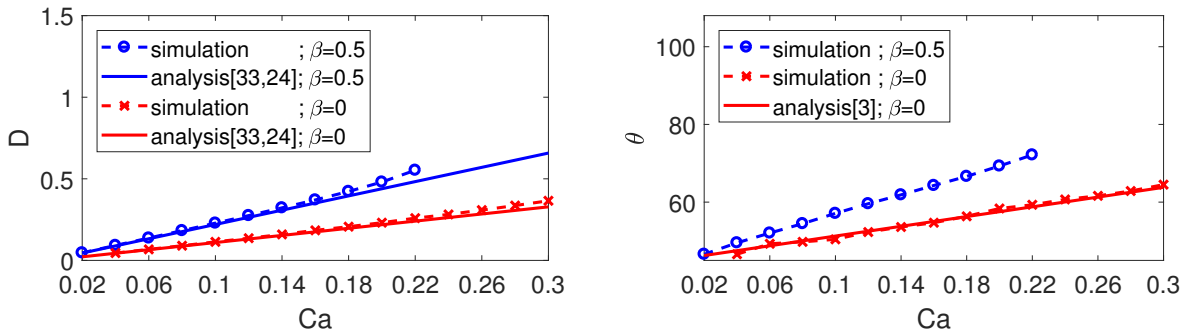


Figure 7: (a) The deformation factor versus the capillary number for the present simulations and theoretical results in Eq. (21). (b) The inclinational angle versus the capillary number for the present simulations and theoretical results in Eq. (22).

5.4 Effect of elasticity number

In this test, we study the effect of elasticity number β on the droplet under shear flow. The computational setup is same as the second test except we fix the dimensionless numbers $Re = 1, Ca = 0.2, Pe_s = 1$ and vary the elasticity number $\beta = 0, 0.3, 0.5$. From the equation of state Eq. (4), we can see that as β increases, the surface tension reduces more which results larger deformation. This is exactly the case as we can see from Fig. 8 in which the droplet morphology together with surfactant concentration at different times for $\beta = 0, 0.3, 0.5$ are plotted. One can see that for smaller elasticity number $\beta = 0, 0.3$, the droplet tends to be in equilibrium shape after sometime while the one with $\beta = 0.5$ still elongate even after $t = 12$. These droplet behaviors have been confirmed again for the time evolutionary plots of the deformation factor as shown in Fig. 9.

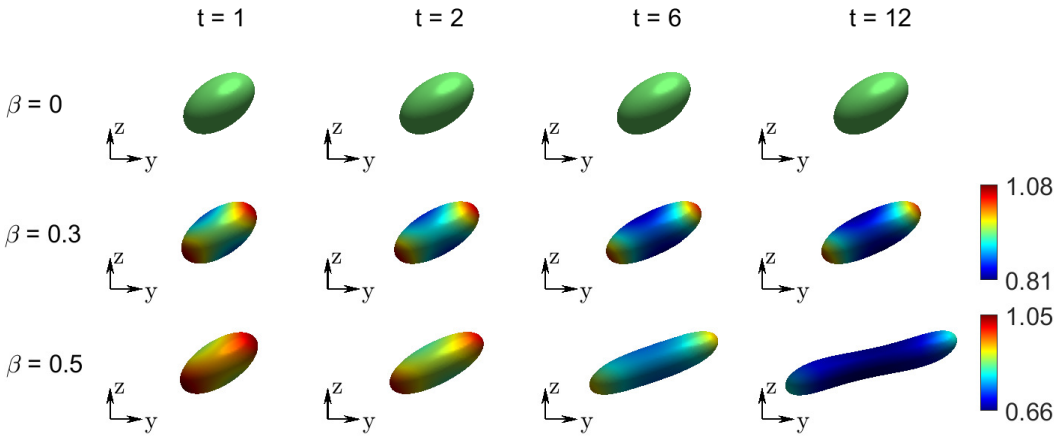


Figure 8: The droplet morphology together with surfactant concentration at different times for $\beta = 0, 0.3, 0.5$.

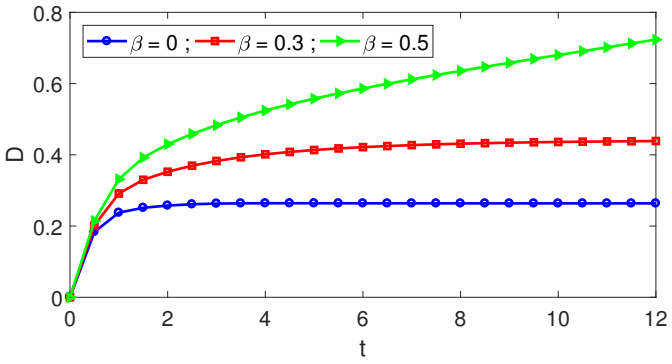


Figure 9: The time evolutionary plots of the deformation factor for $\beta = 0, 0.3, 0.5$.

5.5 Effect of Peclet number

As a last test, we study the effect of surface Peclet number Pe_s on the droplet under shear flow. The computational setup is same as the second test except we fix the dimensionless numbers $Re = 1, Ca = 0.2, \beta = 0.28$ and vary the Peclet number $Pe_s = 0.1, 1, 10$. As mentioned earlier, the Peclet number represents the relative importance between surfactant convection and diffusion in the droplet interface dynamics, so a larger Peclet number indicates the convection is more dominated. Since the initial surfactant concentration is uniformly $\Gamma_0 = 1$ along the interface, the larger Peclet number makes the surfactant distribution more profound as shown in Fig. 10 where the value range of surfactant concentration becomes larger as Pe_s increases. This results in larger droplet deformation as the Peclet number increases which is confirmed in Fig. 11.

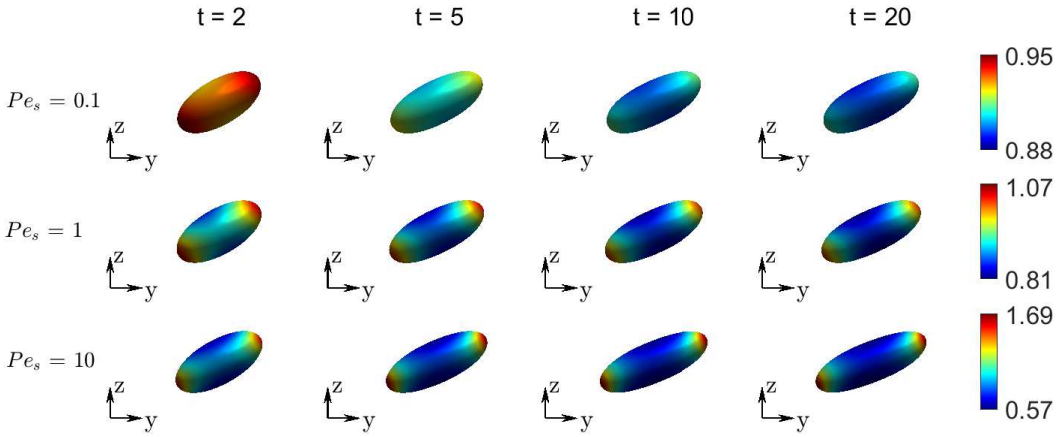


Figure 10: The droplet morphology together with surfactant concentration at different times for $Pe_s = 0.1, 1, 10$.

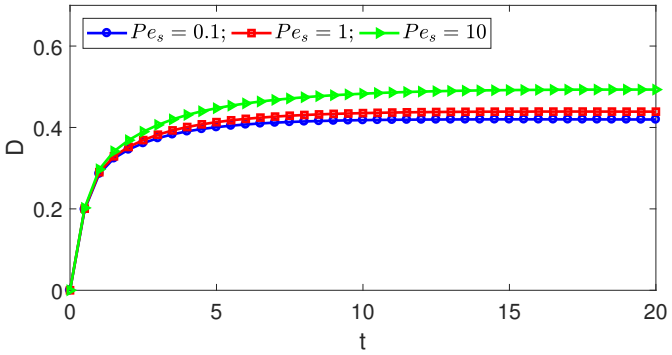


Figure 11: The The time evolutionary plots of the deformation factor for $Pe_s = 0.1, 1, 10$.

6 Conclusion

In this paper, we develop a coupled grid based particle and implicit boundary integral method for simulations of three-dimensional interfacial flows with the presence of insoluble surfactant. We use splitting scheme to handle the moving interface as well as the surfactant density that is being convected along. We compared our numerical simulations to some problems for which the analytical solutions are available. The comparison shows that the proposed method obtains the second-order convergence rate as grid refines. For complicated problems, such as a droplet in shear flow, our simulation results highly coincide with theoretical behavior. We plan to generalize the proposed method and code to study droplet dynamics in the presence of surfactant under a DC electric field. As we mentioned in the introduction, our proposed method can be used easily to compute electric potentials involving dielectric droplets in electrohydrodynamic applications.

Acknowledgment

The work of M.-C. Lai is supported in part by Ministry of Science and Technology of Taiwan under research grant MOST-107-2115-M-009-016-MY3 and NCTS. J. Chu's is partially supported by Ministry of Science and Technology of Taiwan under research grant MOST-106-2115-M-007 -002. R. Tsai's research is partially supported by National Science Foundation grant DMS-1720171. Tsai also thanks National Center for Theoretical Sciences (NCTS) for hosting this collaboration.

References

[1] J. Adams, P. Swarztrauber, R. Sweet, Fishpack - a package of Fortran subprograms for the solution of separable elliptic partial differential equations, Available at, http://www.netlib.org/fishpack, 1980

[2] H.D. Ceniceros, The effects of surfactants on the formation and evolution of capillary waves, Phys. Fluids. 15 (2003) 245-256.

[3] C.E. Chaffey, H. Brenner, A second-order theory for shear deformation of drops, J. Colloid Interface Sci. 24 (1967) 258-269.

[4] J. Chu, R. Tsai, Volumetric variational principles for a class of partial differential equations defined on surfaces and curves, Res. Math. Sci. 5 (2018).

[5] A. Demlow, G. Dziuk, An adaptive finite element method for the Laplace-Beltrami operator on implicitly defined surfaces, SIAM J. Numer. Anal. 45 (2007) 421-442.

[6] G. Dziuk, C.M. Elliott, Finite elements on evolving surfaces, IMA J. Numer. Anal. 27 (2007) 262-292.

[7] G. Dziuk, C.M. Elliott, A fully discrete evolving surface finite element method, SIAM J. Numer. Anal. 50 (2012) 2677-2694.

[8] G. Dziuk, C.M. Elliott, Finite element methods for surface PDEs, Acta Numer. 22 (2013) 289-396.

[9] C.M. Elliott, B. Stinner, V. Styles, R. Welford, Numerical computation of advection and diffusion on evolving diffuse interfaces, IMA J. Numer. Anal. 31 (2011) 786-812.

[10] J.L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Eng. 195 (2006) 6011-6045.

[11] F.H. Harlow, J.E. Welch, Francis H. Harlow J. Eddie Welch, Phys. Fluids. 8 (1965) 2182-2189.

[12] H. Huang, M.-C. Lai, H.-C. Tseng, A parametric derivation of the surfactant transport equation along a deforming fluid interface, Front. Appl. Comput. Math. - Dedic. to Daljit Singh Ahluwalia His 75th Birthd. - Proc. 2008 Conf. FACM'08. (2008) 198-205.

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

[14] C. Kublik, N.M. Tanushev, R. Tsai, An implicit interface boundary integral method for Poisson's equation on arbitrary domains, J. Comput. Phys. 247 (2013) 279-311.

[15] C. Kublik, R. Tsai, Integration over curves and surfaces defined by the closest point mapping, Res. Math. Sci. 3 (2016) 3.

[16] Y. Seol, S.-H. Hsu, M.-C. Lai, An immersed boundary method for simulating interfacial flows with insoluble surfactant in three dimensions, Commun. Comput. Phys. 23 (2018) 640-664.

[17] M.-C. Lai, Y.H. Tseng, H. Huang, An immersed boundary method for interfacial flows with insoluble surfactant, J. Comput. Phys. 227 (2008) 7279-7293.

[18] M. Muradoglu, G. Tryggvason, Simulations of soluble surfactants in 3D multiphase flow, J. Comput. Phys. 274 (2014) 737-757.

[19] W. C. de Jesus, A. M. Roma, M. R. Pivello, M. M. Villar, A. da Silveira-Neto, A 3D front-tracking approach for simulation of a two-phase fluid with insoluble surfactant, J. Comput. Phys. 281 (2015) 403-420.

[20] L.D. Landau, E.M. Lifshitz, Fluid Mechanics, Image Rochester NY. 6 (1987) 539.

[21] M. Lenz, S. Nemadjieu, M. Rumpf A convergent finite volume scheme for diffusion on evolving surfaces SIAM J. Numer. Anal., 49 (2011) 15-37.

[22] S. Leung, J. Lowengrub, H. Zhao, A grid based particle method for solving partial differential equations on evolving surfaces and modeling high order geometrical motion, J. Comput. Phys. 230 (2011) 2540-2561.

[23] S. Leung, H. Zhao, A grid based particle method for moving interface problems, J. Comput. Phys. 228 (2009) 2993-3024.

[24] X. Li, C. Pozrikidis, The effect of surfactants on drop deformation and on the rheology of dilute emulsions in Stokes flow, J. Fluid Mech. 341 (1997) 165-194.

[25] J. Liang, H. Zhao, Solving partial differential equations on point clouds, SIAM J. Sci. Comput. 35 (2013), A1461-A1486.

[26] C.B. Macdonald, S.J. Ruuth, The implicit closest point method for the numerical solution of partial differential equations on surfaces, SIAM J. Sci. Comput. 31 (2010) 4330-4350.

[27] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, 2004.

[28] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479-517.

[29] A. Petras, S.J. Ruuth, PDEs on moving surfaces via the closest point method and a modified grid based particle method, J. Comput. Phys. 312 (2016) 139-156.

[30] S.J. Ruuth, B. Merriman, A simple embedding method for solving partial differential equations on surfaces, J. Comput. Phys. 227 (2008) 1943-1961.

[31] K.J. Stebe, R. Balasubramaniam, Marangoni effects on drop deformation in an extensional flow: The role of surfactant physical chemistry. I. Insoluble surfactants, Phys. Fluids. 8 (1996).

[32] H. A. Stone, A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface, Phys. Fluids A Fluid Dyn. 2 (1990) 111-112.

[33] H. A. Stone, L.G. Leal, The effects of surfactants on drop deformation and breakup, J. Fluid Mech. 220 (1990) 161.

[34] C. Sorgentone, A.-K. Tornberg, A highly accurate boundary integral equation method for surfatant-laden drops in 3D, J. Comput. Phys. 360 (2018) 167-191.

[35] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal., 5 (1968) 506-517.

[36] J. Xu, H.K. Zhao, An eulerian formulation for solving partial differential equations along a moving interface, J. Sci. Comput. 19 (2003) 573-594.

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

[38] J. Xu, W. Shi, M.-C. Lai, A level-set method for two-phase flows with soluble surfactant, J. Comput. Phys. 353 (2018) 336-355.

[39] X. Yang, X. Zhang, Z. Li, G.W. He, A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations, J. Comput. Phys. 228 (2009) 7821-7836.