An explicit Eulerian method for multiphase flow with contact line dynamics and insoluble surfactant

Ludvig af Klinteberg[1,*], Dag Lindbo[1], and Anna-Karin Tornberg[1]

[1] *Numerical Analysis/Linné Flow Centre, Royal Inst. of Tech. (KTH), 100 44 Stockholm, Sweden*

Abstract

The flow behavior of many multiphase flow applications is greatly influenced by wetting properties and the presence of surfactants. We present a numerical method for two-phase flow with insoluble surfactants and contact line dynamics in two dimensions. The method is based on decomposing the interface between two fluids into segments, which are explicitly represented on a local Eulerian grid. It provides a natural framework for treating the surfactant concentration equation, which is solved locally on each segment. An accurate numerical method for the coupled interface/surfactant system is given. The system is coupled to the Navier-Stokes equations through the Immersed Boundary Method, and we discuss the issue of force regularization in wetting problems, when the interface touches the boundary of the domain. We use the method to illustrate how the presence of surfactants influences the behavior of free and wetting drops.

Keywords: Multiphase flow; Insoluble surfactant; Marangoni force; Moving contact line; Immersed boundary method

1 Introduction

Microfluidics has become a very active area of research in fluid mechanics. In this area flows at very small scales are considered, and at these scales the effects of surface physics become increasingly important due to the large surface to volume ratio. The systems involved include effects from wetting, surfactants and other phenomena, and a rapid expansion of research objectives in computational mechanics has taken place to encompass modelling of such systems.

This interest in microfluidics is to a large extent motivated by the development of microfluidic devices. Here, the ability to create and manipulate small bubbles, or vesicles, is essential. A well known example is the so-called "Lab-on-a-chip" – a broad term for integrated microfluidic devices that aim to perform advanced chemical synthesis or analysis on a single chip. Experimental investigations include work by Anna and collaborators [3, 4] on microfluidic topics.

One might expect that the *multi-physics* modelling required for microfluidic applications would mostly be a fairly straight-forward matter of coupling existing solvers for the constituent

[*] To whom correspondence should be addressed. Email: ludvigak@kth.se

physical systems. This is however not the case, since these systems present many specific challenges such as fluid-fluid interfaces, strong capillary forces, interface-solid contact lines, non-Newtonian stress models and additional physical considerations, such as heating and the presence of surfactants. Several computational studies have been performed that take some of these features into account.

In this article we will discuss various computational aspects of microfluidic simulations. We discuss *interface tracking* from the point of view of domain decomposition and argue that it is natural to think of on-interface modelling of e.g. surfactants in terms of calculus on manifolds. Additionally, we shall bring modelling of wetting and contact line dynamics to fore, and consider a system involving both contact line dynamics and surfactants. First, though, we survey the diverse areas that we are concerned with.

1.1 Surfactants

Flow problems involving surfactants have captured the attention of many physicists for decades. A basic observation is that the presence of surfactants changes the surface tension in free-surface flows, and hence, among other things, the way drops form, break up and coalesce.

An example of huge practical importance is detergents, though physical interest in surfactants is much broader. We already mentioned that many microfluidic applications involve manipulation of drops or bubbles on small scales. If this is to be done with high precision, one must naturally understand the effect the presence of surfactants has on these operations, and if it can be used to improve them. Furthermore, there appears to be genuine interest from the medical community in the role that surfactants play in vascular disorders, such as decompression sickness, see e.g. [10, 74].

It comes as no surprise, then, that there is large interest in computational microflows and simulation of flows involving surfactants. It is not uncommon that 2D models are quite sound in this context, though there are certainly many applications where a full 3D model is required to accurately describe the flow. Regardless of dimensionality, the main computational tasks are: (*i*) flow modelling (solving e.g. Stokes or Navier-Stokes equations), (*ii*) interface dynamics (i.e. representing and evolving dynamic interfaces), and (*iii*) surfactant modelling (i.e. how surfactant concentrations on the interface, and possibly in the bulk fluid, interacts with the rest of the system).

The first two tasks, collectively denoted multi-phase flow methods, have been studied computationally in a remarkable volume of work, as summarized in a recent review by Wörner [68]. Prominent surveys for each of the main families of methods are (Level-set methods) Sethian & Smereka [55], (Front-tracking) Tryggvason et al. [66], (Volume fraction methods) Scardovelli & Zaleski [54] and (Phase field methods) Anderson et al. [2]. The reader is assumed a certain familiarity with these methods.

Before the third task (surfactants) is considered, it is appropriate to note that the established interface-tracking methods, coupled to flow equations, pose numerical challenges which must be carefully considered. One issue has to do with time-stability, and originates from the coupling of flow equations to interface dynamics via the surface tension force, which is proportional to the interface curvature. This proportionality necessitates differentiation of the interface, and magnifies errors in models such as the Continuum Surface Force (CSF) model [9], which underpins level-set, front tracking and volume-fraction methods. Much effort has been devoted to addressing this issue and successful solution strategies now exist, see for example Ceniceros & Fisher [12] and Sussman & Ohta [60].

Another thing to point out is that interface representation imposes natural limitations. For instance, in level-set methods, the location of the interface is not known (as the method is implicit), and in front-tracking the interface is represented in a point-wise sense. These are two extremes that will pose distinct challenges to further modelling. There have been many attempts to marry the strengths of two methods (or eliminating notable drawbacks) in hybrid methods, such as the work by Sussman & Puckett [61, 59], Aulisa et al. [5], Enright & Fedkiw [22], and Gaudlitz & Adams [24]. These methods add mathematical and practical complexity to already quite complicated methods. Sophisticated methods exist that treat interface jump conditions, such as variations on the Immersed Interface method (IIM) by LeVeque and collaborators [38, 39, 36], including level-set methods. Additional challenges arise in these methods – while they are seen as more accurate (no unphysical "smearing"), the aforementioned stability issues are more severe.

That is not to say that multi-phase simulations are intractable – to the contrary. Much work has gone into efficient implementations of these methods that produce very impressive computational results [35, 56, 43]. The preceding remarks aim to remind that, numerically and in basic modelling terms, multi-phase flow is still a field where the motivation for more research is large.

Proceeding to surfactants, the macroscopic model considers a concentration of surfactants *on* an interface that is governed by convection and diffusion. The basic coordinate-free equation is well known, (derived in plain terms by Wong et al. [67] following [57]),

$$\frac{D\rho}{Dt} + \rho(\nabla_s \cdot \mathbf{u}) = D_\Gamma \nabla_s^2 \rho. \tag{1}$$

Here, we simply note that we are dealing with a reaction-diffusion PDE on the interface which is non-linearly coupled to the interface itself – see Section 3 for a complete statement. Additionally, when the surfactants are soluble, there can be a surfactant concentration in the bulk fluid and an exchange mechanism at the interface, see e.g. Jin et al. [29].

Of particular interest among recent work is the method by Muradoglu & Tryggvason [44]. We wish to highlight this as a valuable computational tool – the results they present, as well as subsequent work [43], clearly justifies this. Rather than being primarily a physical investigation, they set out to give a general computational method for axisymmetric cases that they then validate against theory and experiments. They build upon the well-established *finite difference/front-tracking* (FD/FT) method by Tryggvason et al. [66], and add the surfactant treatment (including soluble surfactants) as a bolt-on to the existing multiphase flow solver; the solver strategy is simply to interleave Euler-forward (first order) time steps of the interface, surfactants and flow. All time-stepping is first-order accurate, but the time step is constrained by stability to be $\Delta t \sim \Delta x^2$, because the diffusive term in the surfactant equation (1) is handled explicitly.

A method with many similarities is given by Zhang et al. [74], in a paper that predates Muradoglu & Tryggvason by about two years. It is, again, a front-tracking method for the axially symmetric case, where the emphasis is more clearly on an application – which is blood flow.

Implicit interface tracking methods have been extended to include surfactants on the interface, such as the level-set method by Xu et al. [69, 70] and various VOF methods [52, 28, 19]. The main concern in such methods is conservation of surfactant; [69] is largely concerned with modifications that deal with this. Other implicit methods recently proposed

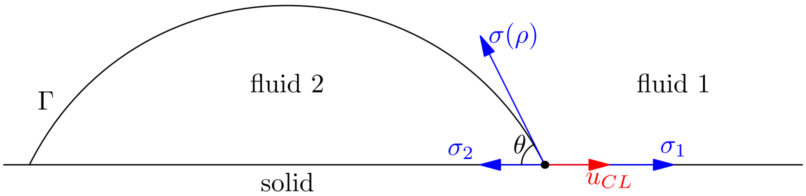


Figure 1: Droplet wetting a solid surface, forming a contact angle θ, where σ denotes the surface tension force acting on the three interfaces.

include phase-field (Cahn-Hilliard) methods by Liu & Zhang [41] and Teigen et al. [62], as well as a "smoothed particle hydrodynamics" method by Adami et al. [1].

In the realm of viscous (Stokes) flow, a method for the full 3D problem was introduced early, namely the boundary-integral method by Yon & Pozrikidis [71]. Booty & Siegel [8] give a boundary integral method that couples to a method for solving a bulk surfactant transport-diffusion equation (in the limit of weak diffusion); and they are able to resolve a thin layer around the interface with large surfactant gradients.

1.2 Wetting and contact line dynamics

The intersection line of two immiscible fluids with a solid is called the contact line, which in two dimensions reduces to a contact point. Wetting generally refers to the case of a liquid-gas-solid system when the gas is displaced by the liquid, and hence the contact line is moving. Applications involving wetting and contact line dynamics appear in many industrial processes, both macroscopic (oil recovery, reactor cooling, coating) and microscopic (drop manipulation in lab-on-a-chip devices), and is clearly of great practical importance.

In the static case when the contact line is immobile, the equilibrium (or static) contact angle[1] θ_s formed between the fluid-fluid interface and the solid surface (cf. Figure 1) is determined by Young's equation,

$$\sigma_1 - \sigma_2 - \sigma \cos\theta_s = 0, \tag{2}$$

which states that the surface tensions $\sigma_1, \sigma_2, \sigma$ at the fluid-fluid and solid-fluid interfaces are in horizontal equilibrium when the contact line is at rest [16].

If the static case can be treated with relative ease; quite the opposite can be said about the dynamic case, when the contact line is moving. We refer to the recent review articles [6, 7] for an introduction to the problem, which still is the topic of much active research. At the core of it lies the issue that the Navier-Stokes equations with the common no-slip boundary condition result in a singular shear stress at the contact line. This singularity is unphysical, and to avoid it the hydrodynamic continuum model must be modified to take into account the molecular interactions taking place at fluid-fluid and solid-fluid interfaces. Several approaches to numerical modelling of moving contact lines exist, all aiming to capture the correct behavior of the drop without resolving the microscopic interactions at the contact line. These models are *mesoscopic*, in that they rely on the flow being resolved in the inner region, which is determined by the slip length. Ren and E [49, 51] derived an effective velocity boundary condition at the contact line, relating the contact line velocity u_{CL} to the

[1] In this work we consider only the apparent/macroscopic contact angle, see e.g. de Gennes [16] for an extensive treatment.

contact angle. Conversely, Muradoglu & Tasoglu [43] simulated droplets spreading on walls by specifying the contact angle based on the flow velocity near the wall, and then updating the interface by fitting a cubic curve satisfying that contact angle between the wall and a marker point near the wall. Yet another method is to move the contact line by prescribing the contact angle to the static angle, as was done by Zahedi et al. [72].

In the presence of surfactants on the fluid-fluid interface, the wetting properties of the system will change. The surface tension σ of the interface is lowered by the surfactants, changing the equilibrium contact angle in Young's equation (2) (a water drop will typically increase its wettability when surfactants are added [53]). There can also be an exchange of surfactants between the interface and the solid surface as the contact line moves, in which case additional difficulties arise in solving the contact line singularity [15, 48]. A numerical study on the effects of surfactants and temperature on the spreading of a two-dimensional viscous droplet with surfactant exchange was performed by Clay & Miksis [14], using a lubrication approximation.

In this work we have limited ourselves to the case where there is no surfactant exchange between interface and solid, such that the wetting properties are only changed due to the modified equilibrium contact angle. A numerical simulation of this case was performed by Lai et al. [33], using an immersed boundary method in two dimensions. They made use of the Langmuir equation of state to relate the surface tension to the surfactant concentration. To move the contact line, they then set Navier slip on the entire solid surface and applied the unbalanced Young's force to the contact point to obtain movement. The work by Lai et al. is, to our knowledge, currently the only existing numerical simulation combining surfactants with moving contact lines where the full flow problem is considered.

1.3 Structure of this paper

In this paper we present a new numerical method for insoluble surfactants on dynamic interfaces in two-dimensional Navier-Stokes flow with contact line dynamics. The interface representation used in this paper is a domain decomposition approach introduced by Lindbo & Tornberg [40]. The method is an extension of the segment projection method by Tornberg & Engquist [64, 21], and is able to decompose interfaces into segments in a more general way. Here we further develop Lindbo & Tornberg's method by extending it to treat open interfaces, contact line problems and insoluble surfactants. The surfactant treatment is based on the work by Khatri & Tornberg [31] for representing surfactants on interfaces using the segment projection method. We extend this surfactant representation to fit with the method by Lindbo & Tornberg and with open interfaces.

In Section 2 we present an interface tracking method which provides a natural framework to pose the surfactant equations in (Section 3). An immersed boundary multi-phase Navier Stokes method is introduced in Section 4. We then discuss the central issues in the treatment of the contact line problem, namely what drives the movement of the contact line, and attempt to shed some light on numerical issues that have been left unresolved by related work (Sections 4.2 and 4.3). Putting the many facets of the model together, in Section 5, we discuss numerically well-motivated strategies to compute approximate solutions to the fully coupled interface/surfactant/flow/contact system in a segregated manner. We close with numerical results, in Section 6.

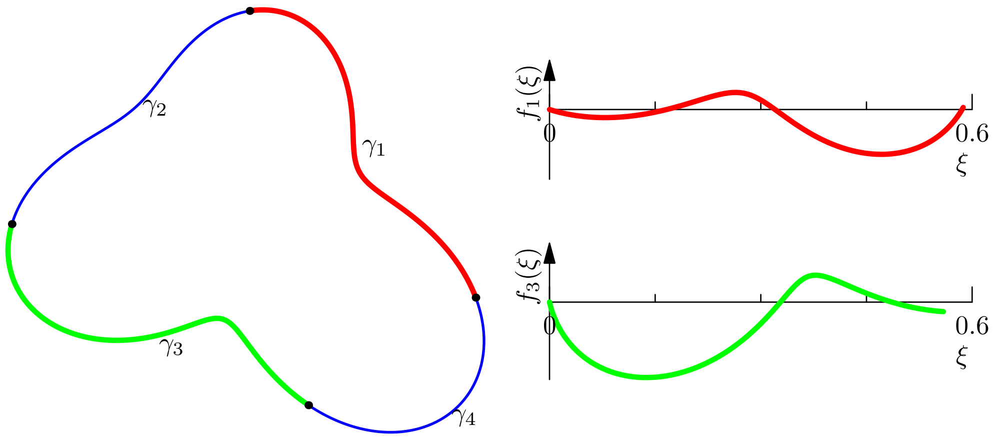


Figure 2: Interface split into segments (ends marked with ∘), with two segments drawn bold, and their local representations.

2 Interface tracking using domain decomposition

Here we give a brief introduction to the interface tracking method proposed in [40], and extend it to cope with on-interface PDE modelling. Let a simple plane curve of length L be represented by $\Gamma \subseteq \mathbb{R}^2$. Introduce a parametrization, s, of Γ at time t,

$$\Gamma = \Gamma(t) = \{x(s,t),\ y(s,t) : s \in [0,L)\}$$

and assume $x, y \in C^2[0,L)$. Let $\hat{\tau}(s)$ and $\hat{n}(s)$ be unit vectors tangential and normal to Γ, with $\hat{n}$ pointing inwards. We shall now split Γ into a number of segments, γ_j, such that $\Gamma = \bigcup_j \gamma_j$ and each segment can be represented as a single valued function with respect to a particular coordinate system as illustrated in Figure 2. The partitioning of Γ into segments is clearly not unique, the only requirement being the representation of each segment as a single valued function, and can be performed in many different ways. We here use the algorithm introduced in [40, Sec. 2.6], which is based on a maximum cumulative winding angle allowed on each segment.

To formalize the segment representation, we let, for any $\mathbf{x} \in \mathbb{R}^2$,

$$\mathcal{M}\mathbf{x} := \mathcal{T}(\Theta)\mathbf{x} + \mathbf{b} \tag{3}$$

$$\mathcal{L}\mathbf{x} := \mathcal{M}^{-1}\mathbf{x} = \mathcal{T}(-\Theta)(\mathbf{x} - \mathbf{b}), \tag{4}$$

where $\mathcal{T}$ is the standard unitary rotation operator, Θ is an angle and $\mathbf{b}$ an offset. It is clear that one may decompose Γ into a collection of segments γ_j, and to each one associate mapping parameters $(\Theta, \mathbf{b})$, such that

$$\mathcal{M}_j = \mathcal{M}(\Theta_j, \mathbf{b}_j) : \gamma_j \longrightarrow \{\xi, f_j(\xi) : \xi \in (a_j, b_j)\}.$$

As clarity permits, we shall omit subscripts, and employ the following notation:

$$\mathcal{M}\gamma = (\xi, f(\xi)), \quad \text{and} \quad \mathcal{L}(\xi, f(\xi)) = \gamma.$$

This, with Figure 2 in mind, is quite intuitive. More subtle details shall emerge as we pose time-dependent differential equations, involving Γ, in local coordinates over the segment collection.

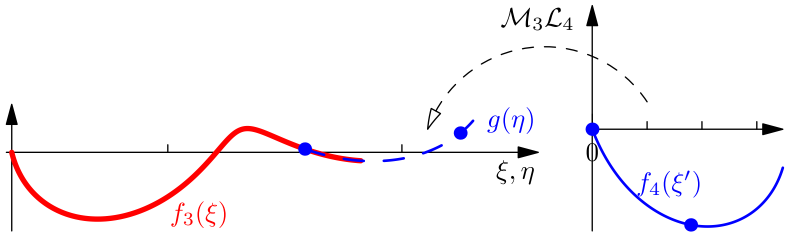


Figure 3: Schematic view of two neighboring segments that overlap (see Figure 2, segments γ_3 and γ_4). The mapping sequence $\mathcal{M}_3\mathcal{L}_4$ takes data from segment four into the frame of segment three, which can thusly be extended or given consistent "boundary" data.

2.1 Interface dynamics

The first task is to add the time-dimension and ask how interface dynamics can be expressed in terms of segments. Let $\mathbf{u}$ denote a convective field that transports the interface,

$$\frac{\mathrm{d}\Gamma}{\mathrm{d}t} = \mathbf{u}(\Gamma(t), \cdot), \tag{5}$$

where $\mathbf{u}$ may, possibly, be a something complicated, like an approximate solution for Navier-Stokes equations, or something simple, such as an explicit function, $\mathbf{u} = \mathbf{u}(\Gamma(t), t)$. In [40], we show that (5) is equivalent to solving a PDE

$$\frac{\partial}{\partial t} f(t, \xi) + v \frac{\partial}{\partial \xi} f(\xi, t) = w, \quad (v, w) = \mathcal{T}(\Theta)\mathbf{u}(\gamma(t), \cdot), \tag{6}$$

for each segment. Here we have assumed that Θ and $\mathbf{b}$ are fixed in time for each segment (see [40] for a more general setting). The important question of boundary conditions remains. To answer this, we need to clarify the relationship between adjacent segments.

2.2 Geometry of segments

For the segment advection PDE (6) to be meaningful, and for further modelling to be consistent, we need additional structure, namely elementary manifold properties. Let there be a finite overlap between adjacent segments. Then, we may view part of a segment *as a single-valued function* from the coordinate frame of its neighbor, see Figure 3. In terms of sets we have

$$(\eta, g(\eta)) = \mathcal{M}_k\mathcal{L}_j(\xi_j', f_j(\xi_j')), \tag{7}$$

where segment j and k are adjacent, and ξ_j' denotes the interval on segment j where they overlap. In this interval, the segments coincide, so

$$(\eta, g(\eta)) = (\xi_k', f_k(\xi_k')),$$

where, naturally, ξ_k' denotes the overlap interval of segment k. The operation of going from one segment to another via a mapping to global coordinates, i.e. (7), is well-defined. Note, however, that the relation $(\eta, g(\eta)) = (\xi, f(\xi))$ does not express a relationship between parameterizations. That is, given that $f(\xi) = g(\eta(\xi))$ in a point-wise sense, the map $\eta(\xi)$ is non-trivial.

2.3 Interface-interior boundary conditions

Now we return to the question of boundary conditions for the segment advection PDE (6). The PDEs (6) are of first order hyperbolic type, so information will propagate across segment boundaries. Hence, the situation demands *characteristic boundary conditions* on each segment (see e.g. the textbook by LeVeque [37]). The essence is that incoming characteristics on one segment are determined from outgoing characteristics on adjacent segments. As illustrated in Figure 3, we can map one segment function into the frame of an adjacent segment, thus providing the necessary boundary data to make the collective evolution of (6) consistent with (5).

2.4 Discrete method

We discretize each segment with a uniform grid with spacing $\Delta\xi$ and employ finite difference methods for all calculations. The transport equation (6) is time-stepped using an explicit two-step Lax-Wendroff method, as in [40],

$$\begin{aligned} f^{n+1} = f^* + \frac{\Delta t^2}{2} v(\mathbf{x}^n, t^n)\left(-D_0 w(\mathbf{x}^n, t^n)(D_0 v(\mathbf{x}^n, t^n))(D_0 f^n) + v(\mathbf{x}^n, t^n) D_+ D_- f^n\right) \\ - \frac{\Delta t^2}{2}\left(\frac{w(\mathbf{x}^*, t^{n+1}) - w(\mathbf{x}^n, t^n)}{\Delta t} - \frac{v(\mathbf{x}^*, t^{n+1}) - v(\mathbf{x}^n, t^n)}{\Delta t} D_0 f^n\right). \end{aligned} \tag{8}$$

where D_+, D_-, and D_0 are the usual one-sided and centered derivative approximations, and

$$f^* = f^n + \Delta t(w(\mathbf{x}^n, t^n) + v(\mathbf{x}^n, t^n) D_0 f^n)$$

is an Euler step so that the time-derivatives of the coefficients (v, w) can be estimated. Recall that we have multiple segments (at least three for closed interfaces), each characterized by a segment function $f(\xi, t)$. Hence, we time-step according to (8) *on each segment.* The discrete difference operators necessitate one grid-point of “boundary data” on each end of each segment at time t^n. Since the segments overlap, this is obtained by interpolating from adjacent segments, as described in Section 2.3 and Figure 3. Moreover, the partition of the interface into segments may eventually become invalid if the interface deforms sharply, in which case it needs to be redefined. We give full details surrounding this in [40].

3 Surfactants on segments

Let the concentration of an insoluble surfactant be defined as a function ρ on Γ,

$$\rho = \rho(s, t), \quad \text{on} \quad (x(s, t), y(s, t)), \quad s \in [0, L).$$

As discussed in the introduction, ρ satisfies an advection-diffusion equation (1),

$$\frac{D\rho}{Dt} + \rho(\nabla_s \cdot \mathbf{u}) = \frac{1}{\mathrm{Pe}_\Gamma}\nabla_s^2 \rho, \tag{9}$$

that takes stretching of the interface into account (second term). Here, Pe_Γ is the Peclet number, the nondimensional parameter that governs the diffusion along the interface. The surface gradient operator ∇_s is defined as $\nabla_s = (I - \hat{n} \otimes \hat{n})\nabla$, which in two dimensions simplifies to

$$\nabla_s = \hat{\tau}(\hat{\tau} \cdot \nabla) = \hat{\tau}\frac{d}{ds}.$$

We now take the surfactant representation developed by Khatri & Tornberg [30, 31] for the segment projection method and apply it to our domain decomposition framework. The concentration is then represented locally as $\rho = \rho(\xi, t)$ on each segment γ, forming the triplet $(\xi, f(\xi,t), \rho(\xi,t))$. It is a straight forward calculation to show that, in local terms, (9) is equivalent to

$$\frac{\partial \rho}{\partial t} + v\frac{\partial \rho}{\partial \xi} + \rho\phi_f(\frac{\partial v}{\partial \xi} + \frac{\partial w}{\partial \xi}\frac{\partial f}{\partial \xi}) = \frac{\phi_f}{\mathrm{Pe}_\Gamma}\frac{\partial}{\partial \xi}(\frac{\partial \rho}{\partial \xi}\phi_f), \tag{10}$$

where

$$\phi_f := \left(1 + \left(\frac{\partial f}{\partial \xi}\right)^2\right)^{-1/2}, \tag{11}$$

and $(v, w) = \mathcal{T}(\Theta)\mathbf{u}(\Gamma(t), t)$. A detailed derivation is available in [30, pp. 28-31], though in a slightly different setting.

On each segment, the surfactant equation (10) is discretized on the same uniform local grid, $\Delta\xi$, as the interface advection PDE (6) is in the Lax-Wendroff step (8), and the same mechanism for boundary conditions applies. That is, the surfactant concentration on an adjacent segment can be obtained in a small overlap region via the local-to-global mapping (3-4) in analogy with Figure 3. At the endpoints of an open interface there is a no-flux boundary condition, meaning that there is no surfactant exchange between the solid wall and the interface.

It is important to emphasize that, while being far from simple, (10) is vastly easier to deal with than (9). This is because the final problem is to solve a PDE with one spatial dimension on a uniform grid, a problem for which many standard (and well-understood) numerical methods are available.

The surfactant dynamics depend on the interface shape, conveyed by the presence of the surface gradient in (9), or derivatives of f in (10). Evidently, (6) together with (10) constitute a non-linear system of PDE's. However there are mediating factors. For one, the ρ couples to f, but not the other way around – at least not until we let $\mathbf{u}$ depend on ρ via two-way force coupling to Navier Stokes equations. The second equation is, by itself, linear if f is known. On the other hand, f characterizes the manifold on which ρ is defined, so one cannot determine ρ ahead of f in time.

3.1 Interface forces and equations of state

In the familiar multi-phase flow setting, an interface gives rise to forces that are localized on the interface but act on the embedding continuum. In [40] we consider only clean interfaces (i.e. without surfactant) and only the force due to surface tension – an interface-normal force directly proportional to the interface curvature [9],

$$\mathbf{f}_\perp(s) = \sigma\kappa(s)\hat{n}(s). \tag{12}$$

The presence of surfactants is well-known to change the strength of the surface tension locally [34, 45]. This is commonly modelled by the Langmuir equation of state [20],

$$\sigma = \sigma(\rho(s)) = 1 + \mathrm{E}\ln(1 - \rho(s)), \tag{13}$$

where $\mathrm{E} = \frac{RT\rho_\infty}{\sigma_c}$ is the dimensionless elasticity number measuring to what extent the presence of surfactants on the interface modifies the surface tension. In this paper ρ is nondimensionalized by ρ_∞, the maximum surfactant concentration possible on the interface, and σ by σ_c, the surface tension of a clean interface. Moreover, the presence of surfactants generates a tangential force proportional to the surfactant gradient, known as the Marangoni stress,

$$\mathbf{f}_{\|} = \nabla_s \sigma(\rho) = \hat{\tau}\frac{\mathrm{E}}{1-\rho}\frac{\mathrm{d}\rho}{\mathrm{d}s}. \tag{14}$$

These quantities are naturally computed in the local coordinate frame of each segment,

$$\begin{aligned} \kappa(\xi) &= \frac{f''(\xi)}{(1 + f'(\xi)^2)^{3/2}}, \quad \hat{n}(\xi) = \phi_f[f'(\xi),\ -1], \\ \frac{\mathrm{d}\rho}{\mathrm{d}s} &= \phi_f\frac{\partial\rho}{\partial\xi}, \quad \hat{\tau}(\xi) = \phi_f[1,\ -f'(\xi)], \end{aligned}$$

as this is the concrete interface representation of the present method, and a parametrization in terms of ξ is provided on a uniform grid, such that derivatives with respect to ξ can be computed using centered finite differences. The total force becomes

$$\begin{aligned} \mathbf{f}(s) &= \mathbf{f}_\perp(s) + \mathbf{f}_{\|}(s) = \sigma(\rho)\kappa(s)\hat{n}(s) + \hat{\tau}(s)\frac{E}{1-\rho}\frac{\mathrm{d}\rho}{\mathrm{d}s} \\ &= \mathcal{T}(-\Theta)\left(\sigma(\rho(\xi))\frac{f''(\xi)}{(1+f'(\xi)^2)^{3/2}}\hat{n}(\xi) + \frac{E}{1-\rho(\xi)}\frac{\mathrm{d}\rho}{\mathrm{d}\xi}\hat{\tau}(\xi)\right) \end{aligned} \tag{15}$$

for any point $\{\Gamma(s), \rho(s)\} \longleftrightarrow (\xi, f(\xi), \rho(\xi))$.

4 Two-phase Navier-Stokes system

The Navier-Stokes equations express the momentum balance and incompressibility of a fluid subject to a body force $\mathbf{F}$, here in nondimensional form,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla^2\mathbf{u} + \frac{1}{\mathrm{ReCa}}\mathbf{F}, \tag{16}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{17}$$

$$+\ \text{Boundary conditions},$$

where the Reynolds number $\mathrm{Re} = \frac{\pi U L}{\mu}$ is the ratio of inertial to viscous forces, and the capillary number $\mathrm{Ca} = \frac{\mu U}{\sigma_c}$ is the ratio of viscous to surface tension forces, both for a fluid with density π, viscosity μ, characteristic length L and characteristic velocity U. We shall discuss three aspects: (*i*) a second order accurate discretization of (16-17), (*ii*) an immersed boundary method for the two-phase aspect of the problem, and (*iii*) the overall interface-surfactant-flow splitting procedure (Section 5).

An overwhelming volume of work concerns numerical methods for the Navier-Stokes equations; we shall not attempt a survey here, but focus on the class referred to as *projection methods* – more precisely *pressure correction* methods – that time-step (16) and then apply a correction such that (17) is (approximately) satisfied. This idea goes back to classical work by Chorin [13], and was elegantly surveyed by Guermond et al. [26].

Here we use the *incremental pressure-correction method in rotational form*, due to Timmermans et al. [63], discussed also in [26, Sec. 3.3]. Letting superscripts denote time level, and introducing a provisional velocity field $\tilde{\mathbf{u}}$, the momentum equation (16) is discretized in time using a second order backward differentiation formula[2],

$$\frac{1}{2\Delta t}\left(3\tilde{\mathbf{u}} - 4\mathbf{u}^k + \mathbf{u}^{k-1}\right) + 2(\mathbf{u}^k \cdot \nabla)\mathbf{u}^k - (\mathbf{u}^{k-1} \cdot \nabla)\mathbf{u}^{k-1} = -\nabla p^k + \frac{1}{\mathrm{Re}}\nabla^2\tilde{\mathbf{u}} + \mathbf{F}^{k+1/2}. \quad (18)$$

A Poisson equation is then solved for an auxiliary field, ϕ,

$$\nabla^2\phi = \frac{3}{2\Delta t}\nabla \cdot \tilde{\mathbf{u}}, \quad \hat{n} \cdot \nabla\phi|_{\partial\Omega} = 0, \quad (19)$$

from which the pressure, p, and velocity, $\mathbf{u}$, is updated,

$$\mathbf{u}^{k+1} = \tilde{\mathbf{u}} - \frac{2\Delta t}{3}\nabla\phi, \quad (20)$$

$$p^{k+1} = \phi + p^k - \frac{1}{\mathrm{Re}}\nabla \cdot \tilde{\mathbf{u}}. \quad (21)$$

In space, we employ centered finite difference approximations on a staggered grid. That this method is second order accurate in time, given sufficient boundary regularity, was proven by Guermond and Shen [27]. Note that if the boundary, $\partial\Omega$, is only Lipschitz, then $\Delta t^{3/2}$ appears to be the maximal convergence attainable. In the present paper a periodic channel is used, for which second order accuracy is both implied by theory and observed in practice. A similar method was used by Lai et al. in [32].

4.1 Immersed boundary method

Moving on to the two-phase part, we employ an immersed boundary method (IBM). That method has its roots in cardiac simulations [46], and was surveyed in [47]. The presence of an interface is accounted for by introducing a singular source term into the Navier-Stokes momentum equation (16).

In terms of Lagrangian variables, the force $\mathbf{f}(s,t)$ generated by the interface curvature and surfactant concentration is given in (15). The force $\mathbf{F}(\mathbf{x},t)$ in (16) can then be computed in terms of Eulerian variables as [47]

$$\mathbf{F}(\mathbf{x},t) = \int_\Gamma \mathbf{f}(s,t)\delta(\Gamma(s,t);\mathbf{x})\mathrm{d}s, \quad (22)$$

where $\delta(\Gamma;\cdot)$ denotes a Dirac measure on $\mathbb{R}^2$ with support on Γ. Conversely, the velocity field that advects the interface in (5) is given by

$$\mathbf{u}(\Gamma(s),t) = \int_\Omega \mathbf{u}(\mathbf{x},t)\delta(\Gamma(s,t);\mathbf{x})\mathrm{d}\mathbf{x}. \quad (23)$$

[2] The interface forces represented by $\mathbf{F}$ are available at time step $k + \frac{1}{2}$ due to the time splitting scheme introduced in Section 5.

Both (22) and (23) are evaluated using the trapezoidal rule. Following the treatment of singular source terms by Tornberg and Engquist [65], we regularize the delta function to the grid using a product of one dimensional delta function approximations,

$$\delta_\epsilon(\mathbf{x}-\Gamma(s)) = \delta_{\epsilon_x}(x - X(s))\delta_{\epsilon_y}(y - Y(s)), \quad (X(s), Y(s)) \in \Gamma, \tag{24}$$

where

$$\delta_{\epsilon_i}(x) = \begin{cases} \frac{1}{h_i}\phi\left(\frac{x}{h_i}\right), & |x| \leq \epsilon_i, \\ 0, & |x| > \epsilon_i. \end{cases} \tag{25}$$

The error when integrating by the trapezoidal rule,

$$E = \left| h_x h_y \sum_{i,j} \int_\Gamma f(s)\delta_\epsilon\left((x_i, y_i) - \Gamma(s)\right) \mathrm{d}s - \int_\Gamma f(s)\mathrm{d}s \right| \tag{26}$$

depends on the delta functions used. A one dimensional δ_ϵ satisfies q discrete moment conditions M_r if

$$M_r(\delta_\epsilon, \bar{x}, h) = h \sum_i \delta_\epsilon(x_i - \bar{x})(x_i - \bar{x})^r = \begin{cases} 1, & r = 0, \\ 0, & 0 < r < q, \end{cases} \quad \forall \bar{x}. \tag{27}$$

It has been shown that $E \leq Ch^q$, where $h = \max(h_x, h_y)$, provided δ_{ϵ_i} satisfies q moment conditions. For a partial differential equation with a singular source term – the interface force in our case – this will contribute to an $O(h^q)$ error in the solution away from the singularity and an $O(h)$ error near the singularity. There are additional concerns when the source terms are close to the boundary of the computational domain – we will discuss this in Section 4.3, where we will also define the delta function approximation used in this work.

When using an immersed boundary method together with a pressure correction method, as we do here, there will in general be a numerical imbalance between the pressure and the surface tension force, resulting in spurious currents of magnitude $O(h)$ in the solution. This phenomenon is particularly pronounced when solving for static interfaces and high density ratios. It is however not unique for immersed boundary methods, and improving the pressure jump representation and reducing spurious currents is an active field of research for many classes of methods [18, 23, 73]. The focus of the present work is the treatment of surfactants and wetting boundary conditions, but a natural direction of future work would be the improvement of the pressure jump representation together with support for density and viscosity ratios other than unity.

4.2 Wetting and boundary conditions

To drive the movement of the contact line, we have here implemented the velocity boundary condition of Ren and E [49], which combines Navier slip on the entire wall with a set slip velocity at the contact line. The tangential velocity of the contact line $u_{CL} = \mathbf{u}_{\parallel,CL}$ is then computed from the contact angle θ as

$$\beta_{CL} u_{CL} = \sigma_2 - \sigma_1 - \sigma(\rho)\cos\theta,$$

where β_{CL} is the effective friction coefficient and the right hand side is the unbalanced Young's force. The contact angle $\theta_{s,c}$ of the clean interface can be determined by Young's equation, $\sigma_c \cos\theta_{s,c} = (\sigma_2 - \sigma_1)$. We can then write the boundary condition as

$$\beta_{CL} u_{CL} = \sigma_c \cos\theta_{s,c} - \sigma(\rho)\cos\theta,$$

which when nondimensionalized by U and σ_c takes the form

$$\mathcal{B} u_{CL} = \cos\theta_{s,c} - \sigma(\rho)\cos\theta, \tag{28}$$

where u_{CL} and σ are now nondimensional and the nondimensional friction number $\mathcal{B} = \frac{\beta_{CL} U}{\sigma_c}$ represents the relative effect of friction forces versus capillary forces at the contact line. The behaviour of this boundary condition is illustrated in Figure 4. On the wall away from the contact line, we have Navier slip with slip length λ_s,

$$\mathbf{u}_{\|} = -\lambda_s (\hat{n} \cdot \nabla)\mathbf{u}_{\|}. \tag{29}$$

Care must be taken when implementing the boundary condition (28), since the contact line will almost inevitably lie between two points on the boundary of the Navier-Stokes grid. To make the implementation consistent we have here introduced a slip width w_s around the contact line, over which the flow boundary condition varies from Navier slip to contact line velocity as

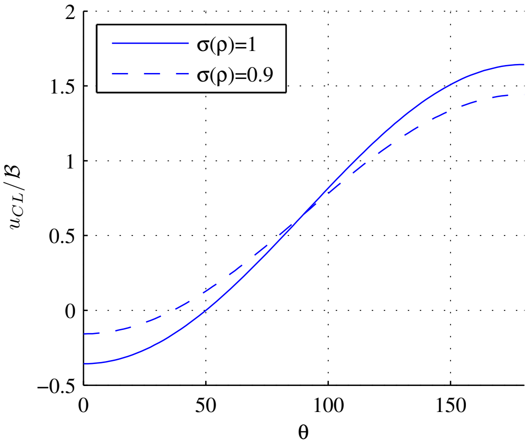


Figure 4: Velocity boundary condition (28) as a function of the contact angle for a clean interface ($\sigma = 1$) and for an interface with surfactants ($\sigma = 0.9$), where σ is the nondimensional surface tension. The static contact angle is here $\theta_{s,c} = 50°$.

$$\mathbf{u}_{\|} = \Phi(x)(u_{CL}) + (1 - \Phi(x))(-\lambda_s \frac{\partial u}{\partial y}), \tag{30}$$

where $\Phi(x) = \exp\left(-4(x - x_{CL})^2/w_s^2\right)$ and x is the coordinate parallel to the wall. The slip width is set to a number of grid sizes, $w_s = n\Delta x$, to minimize grid effects as the contact line moves along the wall. We have found $n = 5$ to give a good compromise between a sharp interface representation and a smooth velocity profile on the wall. This regularization of (28) is necessary because we are imposing a condition located at a moving point on a fixed grid, but it does introduce a new length scale into the numerical solution, something which must be considered when tuning parameters to a physical model. A similar regularization of the contact line boundary condition was used by Ren & E in [50], with the difference that they added the contact line slip to the Navier slip, whereas we use a partition of unity transition between the two.

An alternative that exists in the literature (Lai et al. [33, p. 743]) is to apply the Navier slip condition (29) on the entire boundary, and instead drive the motion of the contact line by adding the unbalanced Young's force to (15) at the contact points, e.g. at $s = 0, L$,

$$\mathbf{f}_y(s) = \begin{cases} \mathbf{f}(s), & s \in (0, L) \\ \mathbf{f}(0) + -(\sigma_1 - \sigma_2 - \sigma(0, \rho(0)) \cos(\theta_0))\hat{y}, & s = 0 \\ \mathbf{f}(L) + (\sigma_1 - \sigma_2 - \sigma(L, \rho(L)) \cos(\theta_L))\hat{y}, & s = L. \end{cases} \tag{31}$$

We will comment some on this second approach, while using the first one for our own implementation.

4.3 Getting force regularization right near domain boundary

Consistent force regularization is of critical importance in the immersed boundary method, since the regularized force is the *only* coupling of the interface dynamics to the underlying flow solver. As one might suspect – from the fact that it deals with replacing a distribution with a continuous function – this is a subtle matter. Certain commonly used approaches have been shown [65] to introduce $\mathcal{O}(1)$ errors. Further complications arise in the context of contact-line problems, where the interface force (15) is non-zero near the solid boundary. The question is how we construct consistent δ-function approximations near the boundary, and how to maintain the consistency between force regularization and boundary conditions. The second issue, to which we shall return shortly, relates strongly to the question of what drives the flow: either the velocity boundary condition (28) by Ren and E [49] or the application of the unbalanced Young's force in the contact point, as in Lai et al. [33][3].

First though, we make a few basic points regarding consistent regularization near the boundary. Clearly, point-wise measures of errors are meaningless, so what do we want a δ-function approximation to satisfy? Naturally one may start from,

$$\int_\Gamma \mathbf{f}(s)\mathrm{d}s = \int_\Omega \mathbf{F}(\mathbf{x})\delta(\Gamma; \mathbf{x})\mathrm{d}\mathbf{x}.$$

Consider the following 1D example: Let $g(x) > 0$ on $x \in [0, L]$ and $\delta_{\varepsilon,y}(x)$ a regularized δ-function with support ε around y. At the least, a consistent δ-function approximation should satisfy the zeroth moment condition

$$\int_0^L \delta_\varepsilon(x - y)\mathrm{d}x = 1 \ \forall y \in [0, L]. \tag{32}$$

However, if we let $\delta_{\varepsilon,y}$ be the linear hat function of support h (33), it is evident from the illustration in Figure 5 that this condition will be violated if y lies near the boundary, in this case if $y < h$. This was mentioned by Zahedi et al. [72] in the context of finite element based

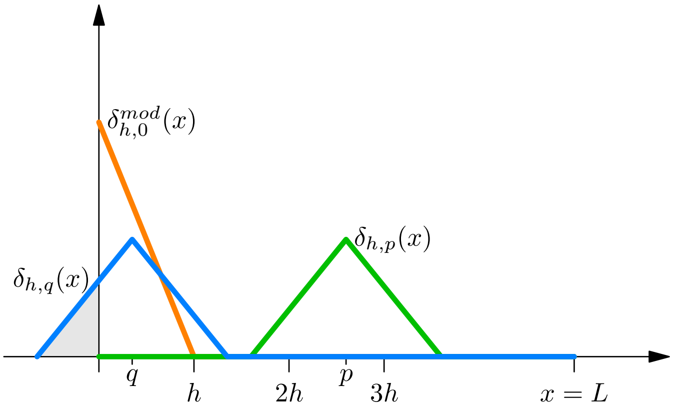


Figure 5: Linear hat functions as δ-function approximations: $\delta_{h,p}(x)$ interior (green); $\delta_{h,q}(x), p < h$ (blue), is not compactly supported on $[0, L]$ but some mass (gray) is "lost"; $\delta_{h,0}^{mod}(x)$ example of a modified function such that (32) is satisfied (orange).

level set methods, where they suggested that the surface tension force be evaluated through a line integral along the interface to avoid this problem.

Returning to the relationship between the near-boundary force regularization and the contact point model, we first consider the approach proposed by Lai et al. [33]: using the Navier slip condition (29) on the wall and applying a point force (31) in the contact point. Since it is a point force that drives the flow and causes the contact point to move, the manner in which the force is spread to the grid, $\mathbf{f}(s) \to \mathbf{F}(\mathbf{x})$, is critical. It should be evident from Figure 5 that without special treatment near the boundary one would have $\int_\Omega |\mathbf{F}(\mathbf{x})\delta(\Gamma;\mathbf{x})|\mathrm{d}\mathbf{x} < \int_\Gamma |\mathbf{f}(s)|\mathrm{d}s$. In principle, this could cause the contact point to move at half the intended speed. Moreover, the manner in which the Navier slip condition (29) is discretized becomes important – in the present configuration the derivative $\frac{\partial u}{\partial y}$ is approximated, and one notices that this will have to take the contact point force into account (otherwise the force regularized to a boundary point will implicitly be excluded from the discrete method by the application of the boundary condition). Lai et al. [33] mention neither of these issues.

The approach provided by Ren and E [49], prescribing the condition (28), can, in light of the preceding remarks, be expected to be less sensitive. The flow velocity at the contact point, $\mathbf{u}_{CL}$, is consistent with the unbalanced Young's force at that point. Therefore, only the force contribution from surface tension and Marangoni stress on the interface need to be regularized near the boundary. There is still a need to ensure the δ-regularization is consistent near the boundary, but one can expect the wetting properties of the problem to be less sensitive to the regularization used.

To make sure we have a consistent regularization near the boundary, and at the same time avoid incorporating regularized forces into our boundary conditions, we have chosen to use a linear combination of two different delta function — none of which will have support on the boundary. Inside the domain we use the narrow linear hat function,

$$\delta_h^{hat}(\bar{x}, x) = \begin{cases} \frac{1}{h}\left(1 - |\frac{x-\bar{x}}{h}|\right), & |x-\bar{x}| \le h, \\ 0, & |x-\bar{x}| > h, \end{cases} \tag{33}$$

which is centered around the source point and satisfies the first two moment conditions (27).

[3] There exists at least two other approaches for "driving" the flow: reconstruction [43] and diffusion [72].

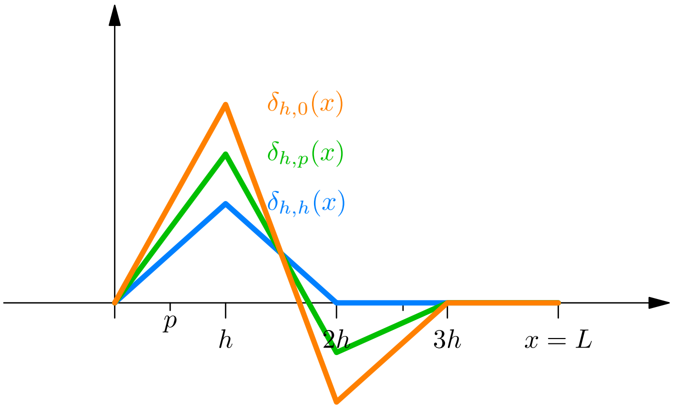


Figure 6: Linear δ-function approximation that satisfies two discrete moment conditions even as the source point approaches the boundary.

For the case when the source point is near or on the boundary, we have constructed a modified delta function by combining two hat functions centered around the two grid points closest to the boundary. For an interval $x \in L$ discretized by $x_1, ..., x_N$, where x_1 and x_N lie on the boundary and $h = x_{i+1} - x_i$, the final delta function is then written

$$\delta_\epsilon(\bar{x}, x) = \begin{cases} d_{3h}(\bar{x}, x, x_2, x_3), & \bar{x} < x_2, \\ \delta_h^{hat}(\bar{x}, x), & x_2 \leq \bar{x} \leq x_{N-1}, \\ d_{3_h}(\bar{x}, x, x_{N-1}, x_{N-2}), & x_{N-1} < \bar{x}, \end{cases} \tag{34}$$

where

$$d_{3h}(\bar{x}, x, x_a, x_b) = \delta_h^{hat}(x_a, x) + \frac{|x_a - \bar{x}|}{h} \left(\delta_h^{hat}(x_a, x) - \delta_h^{hat}(x_b, x) \right). \tag{35}$$

This combination ensures that the regularization involves only interior points $x_2, ..., x_{N-1}$ and that the regularization to the grid is continuous as $\bar{x}$ moves over L. Furthermore, the resulting delta function satisfies the first two moment conditions, i.e. $q = 2$ in (27). The shape of the final δ_ε as the source point moves closer to the boundary is illustrated in Figure 6.

5 Treating the full system

We have introduced all the components of the mathematical model for the two-phase Navier-Stokes system involving surfactants and contact point dynamics, summarized as follows:

a) *Interface tracking* using the domain-decomposition method in Section 2, involving the evolution equation (6) on each segment;

b) *Surfactant dynamics*, which are governed by an advection-diffusion equation (10) in the local coordinates of each segment, as discussed in Section 3;

c) *Navier-Stokes equations* (16, 17) using the pressure-correction method of Timmermans et al. [63], given in Section 4;

d) *Contact-point boundary conditions* (30) following Ren and E [49], to account for the contact point dynamics;

e) *Flow-interface coupling* using an immersed boundary method [47], which amounts to computing (15), (22) and (23).

We shall make the natural distinction between items (a, b) and (c, d), with (e) providing the coupling between the two pairs; this constitutes a *splitting method.* The term “operator splitting” abounds in computational mathematics, and takes on various related meanings. An early reference is Strang [58] and more recent work includes informative surveys by Budd and Iserles [11], and McLachlan and Quispel [42].

For our purposes, a productive view is given by J. Geiser, see e.g. the recent text-book [25], where the focus is, as here, the efficient and accurate treatment of complex modelling problems involving PDEs. The terminology is “decomposition methods”, and aptly so: the original problem is decomposed into simpler problems and (approximate) solutions of these problems is “composed” into a consistent approximation of the original (hard) problem. We refer the reader to [25, pp. 33-39] for an overview of classical splitting methods, including sequential and Strang splitting.

In this light, we split the interface tracking and surfactant problem (a, b) from the Navier-Stokes calculation (c, d) as (a) was split from (c) in [40],

$$\begin{cases} \mathbf{u}^{n-1/2} \to \mathbf{u}^{n+1/2} & \text{using } (\Gamma^n, \rho^n) \\ (\Gamma^n, \rho^n) \to (\Gamma^{n+1}, \rho^{n+1}) & \text{using } \mathbf{u}^{n+1/2}. \end{cases}$$

This can be seen as a classical Strang splitting. From Section 4 it should be clear how the Navier-Stokes step is computed. We now turn our attention to how to tackle the interface-surfactant problem.

5.1 Splitting the interface-surfactant system

Let q denote (f, ρ) so that the interface/surfactant system from (6) and (10),

$$\begin{cases} \frac{\partial}{\partial t} f(t,\xi) + v \frac{\partial}{\partial \xi} f(\xi, t) = w \\ \frac{\partial \rho}{\partial t} + v \frac{\partial \rho}{\partial \xi} + \rho \phi_f \left(\frac{\partial v}{\partial \xi} + \frac{\partial w}{\partial \xi}\frac{\partial f}{\partial \xi}\right) = \frac{\phi_f}{\mathrm{Pe}_\Gamma} \frac{\partial}{\partial \xi}\left(\frac{\partial \rho}{\partial \xi} \phi_f\right), \end{cases} \tag{36}$$

may be written as

$$q_t + F\left(\frac{\partial}{\partial \xi}, \frac{\partial^2}{\partial \xi^2}, q\right) + G = 0, \tag{37}$$

with

$$q(t,\xi) := \begin{bmatrix} f(t,\xi) \\ \rho(t,\xi) \end{bmatrix}, \quad F(q) := \begin{bmatrix} Aq_1 \\ Aq_2 + B(q_1)q_2 + C(q_1)q_2 \end{bmatrix}, \quad G := \begin{bmatrix} -w \\ 0 \end{bmatrix}$$

and

$$\begin{aligned} A = A(v) &:= v \frac{\partial}{\partial \xi} \\ B = B(q_1, v, w) &:= \phi_f \left(\frac{\partial v}{\partial \xi} + \frac{\partial w}{\partial \xi} \frac{\partial q_1}{\partial \xi} \right) \\ C = C(q_1) &:= \frac{\phi_f}{\mathrm{Pe}_\Gamma} \left(\phi_f \frac{\partial^2}{\partial \xi^2} + \frac{\partial \phi_f}{\partial \xi} \frac{\partial}{\partial \xi} \right). \end{aligned}$$

Note that $\phi_f = \phi_f(q_1)$ was defined in (11). We now decompose the system as follows:

$$q_t + Q_1(q) + Q_2(q) = 0, \quad Q_1 := \begin{bmatrix} A \\ A \end{bmatrix} + G, \quad Q_2 := \begin{bmatrix} 0 \\ B + C \end{bmatrix}, \tag{38}$$

based on the observation that $A + G$ is hyperbolic, whereas $B + C$ is elliptic.

A typical way to approach the decomposed system (38) is the Strang method [58]

$$\begin{cases} q_t^I \ + Q_1(q^I) \ \ = 0, & \text{with } q^I(t) = q(t), & t \in (t, t + \Delta t/2) \\ q_t^{II} + Q_2(q^{II}) = 0, & \text{with } q^{II}(t) = q^I(t + \Delta t/2), & t \in (t, t + \Delta t) \\ q_t^I \ + Q_1(q^I) \ \ = 0, & \text{with } q^I(t + \Delta t/2) = q^{II}(t + \Delta t), & t \in (t + \Delta t/2, t + \Delta t) \end{cases} \tag{39}$$

where the final result is $q(t + \Delta t) := q^I(t + \Delta t)$. This method is consistently observed to be fully second order accurate, though a proof of this is not readily available[4].

The discretization of the Q_1 steps in (39) was discussed in Section 2.4; the transport part of the surfactant PDE (10) is of the same form as the segment advection equation. The Q_2 step has the form of a heat-equation, and we use an implicit Crank-Nicholson scheme with the usual centered difference operators on the uniform grid in local coordinates. As is natural given that the interface tracking method is based on domain decomposition, we use a Schwarz iteration to solve the resulting system (see [17, Sec. 6.10.2] or [31]).

6 Numerical results

In this section we report the results of three simulations performed using our method. We show grid convergence and give examples of flow cases where the addition of surfactants to the interface significantly influences the behavior of the flow.

The full system, as described in the previous section, contains a large number of parameters that characterize the flow. The Navier-Stokes equations (16) are directly influenced by the Reynolds number Re and the Capillary number Ca. The diffusion of surfactants on the interface is governed by the Peclet number Pe_Γ (9), while the elasticity number E determines the coupling between surfactants and surface tension (13). The problems including wetting have a contact line friction number $\mathcal{B}$ controlling the wetting speed (28). On the wall away from the contact point the slip velocity is controlled by the slip length λ_s (29). The flow domain is discretized using a staggered grid with $\Delta y = \Delta x$, while the segment equations (36) are discretized using $\Delta\xi = \Delta x$. The velocity boundary condition (30) is regularized using a slip width w_s, which is kept constant when performing convergence tests, to ensure that the same physical problem is solved for all grid refinements. The time step Δt is set such that the CFL condition is satisfied; the implicit treatment of the surfactant diffusion (Q_2 in (39)) removes the $\Delta t \sim \Delta x^2$ limitation that an explicit treatment would have required.

6.1 Migrating drop

To illustrate the effect of Marangoni forces on a drop, we consider the case of a circular drop of unit diameter which initially has a locally high concentration of surfactant on one side, leading to tangential (Marangoni) forces (14) due to surfactant gradient. At $t = 0$, the

[4] An informal proof that the Strang splitting error is second order can be found in e.g. [37, p. 387] for linear operators.

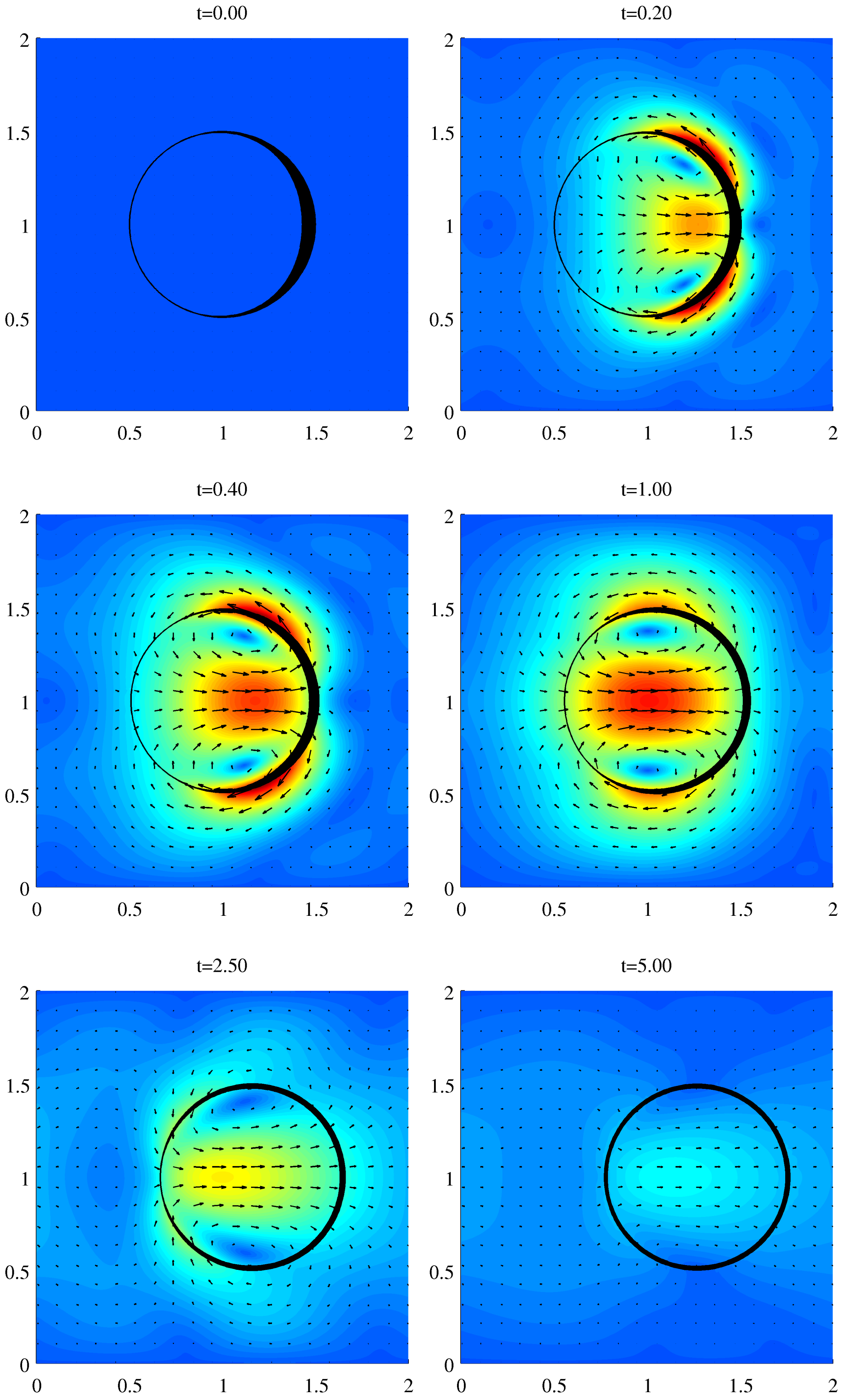


Figure 7: Drop migrating due to Marangoni force. Interface thickness shows surfactant concentration. Note that at the final time, the drop has moved horizontally.

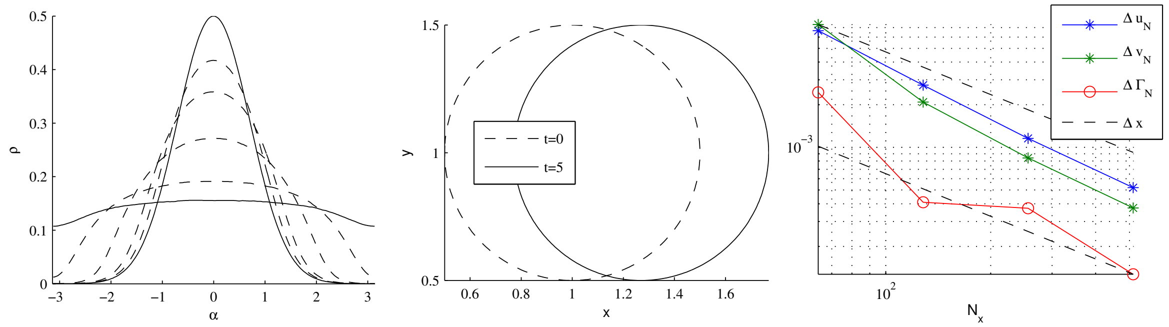


Figure 8: Left: Surfactant concentration plotted against angle from drop centre during time stepping. Concentration at $t = 0$ and $t = 5$ drawn in solid lines. Center: Interface position at the initial and final times. Right: Convergence of flow field and interface.

surfactant concentration on the drop surface is $\rho(\alpha) = \frac{1}{2}\exp(-\alpha^2)$, α being the angle from the center of the drop to the point on the surface. The simulation is performed using $\Delta x = 0.01$ and $\Delta t = \Delta x/2$ until $t = 5$, with dimensionless numbers $\mathrm{Re} = 10$, $\mathrm{Ca} = 0.1$, $\mathrm{E} = 0.5$ and $\mathrm{Pe}_\Gamma = 100$. Figure 7 shows how the Marangoni forces due to the gradient in surfactant concentration (illustrated by line thickness) create a local flow along the interface of the drop. The flow generates recirculation inside the drop, moving it forward. As the concentration gets evened out the flow slows down, until the drop comes to a rest a its new position, to the right of the initial position. Figure 8 (left and center) shows how the surfactant concentration profiles flattens out as the drop moves to its new position.

To check the numerical convergence of our solution, we measure the difference in the flow field at time $t = 1$ between consecutive refinements in L^2-norm, $\Delta u_N = \|u_N - u_{2N}\|_2$ and $\Delta v_N = \|v_N - v_{2N}\|_2$. The convergence of the interface is measured by comparing the right halves of the interface at time $t = 1$ between refinements in maximum norm, $\Delta\Gamma_N = \|\Gamma_N - \Gamma_{2N}\|_\infty$. The domain $(0, 2) \times (0, 2)$ is discretized into $N \times N$ points, $N = \{32, 64, 128, 256\}$. The solutions on the two staggered grids are compared by interpolating from the refined grid onto the coarse grid. Figure 8 (right) shows the convergence as the grid is a refined, with an average convergence rate around 1.3.

6.2 Moving drop on solid surface

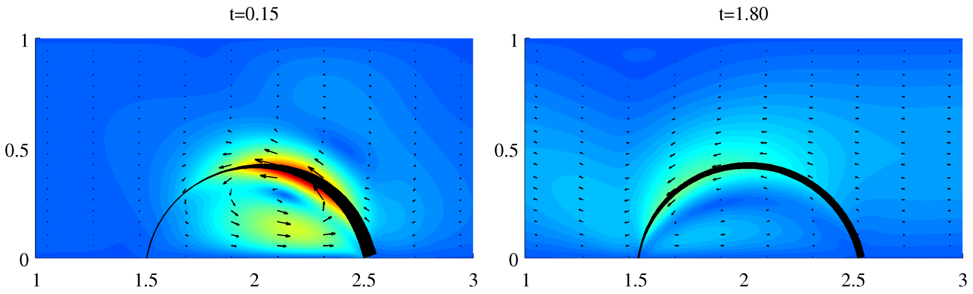


Figure 9: Drop migrating due to Marangoni force. Interface thickness shows surfactant concentration.

Here we simulate a drop of radius $R = 0.5$ which is initially at rest on a solid surface, with contact angle $\theta = \theta_{s,c} = 80°$ and a surfactant distribution that has a maximum at the right contact point. We set $\mathrm{Re} = 10$, $\mathrm{Ca} = 0.1$, $\mathrm{E} = 0.5$, $\mathrm{Pe}_\Gamma = 100$, $\mathcal{B} = 1$, $w_s = 0.05$, $\lambda_s = 0.05$, $\Delta t = \Delta x/2$ and run until $t = 3$ in the domain $(0, 4) \times (0, 1)$. Figure 9 shows the solution at different times, illustrating how Marangoni forces create a flow that redistributes the surfactant concentration on the interface. The drop moves very slightly to the right in the process and the contact angles also change due to the presence of surfactant, though the visible effects are mainly in the transient flow.

To check the numerical convergence of our contact line treatment, we measure the average difference in x-position between times $t = 0$ and $t = 1$ (during the transient state) as $e(\Delta x) = \int_0^1 \|x(\Delta x) - x(2\Delta x)\|_1 \mathrm{d}t$, between refinements $\Delta x = \{1/16,\ 1/32,\ 1/64,\ 1/128\}$. This is done for the points on the interface that lie in the box $\{x > 2,\ 0 \leq y \leq 0.3\}$, which is the part of the interface where the change of position is most pronounced. The results displayed in Figure 10 indicate that the convergence is linear.

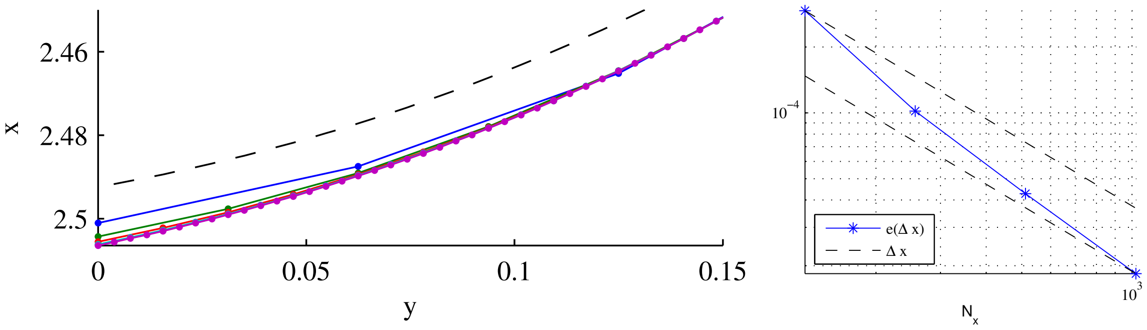


Figure 10: Left: Points near the right contact line (rotated by 90° in plot) at $t = 1$ for a number of grid refinements. The dashed line represents the interface at $t = 0$. Right: Average difference in x-position between refinements.

6.3 Steady state of drop on solid surface in shear flow

We now turn our attention to a moving drop on a channel wall in shear flow, with the purpose of investigating the effect surfactants have on the shape of the drop. The drop is initially at rest with contact angle $\theta = \theta_{s,c} = 80°$ and a uniform surfactant distribution $\rho = \rho_0$. At time $t = 0$ the top wall starts to move with unit velocity, and as the shear flow develops the drop starts moving along the bottom wall. We let the simulation run until the shape and surfactant distribution of the drop have reached a steady state, at which the drop is simply being translated horizontally and the flow transport of surfactant along the interface is balanced by the diffusion. We do this for a number of combinations of ρ_0 and Pe_Γ, while fixing the remaining parameters at $\mathcal{B} = 1$, $E = 1$, $\lambda_s = 0.01$, $w_s = 0.025$ and $\Delta x = 0.005$.

The flow field is displayed in Figure 11 with the translational velocity of the drop subtracted, meaning that it is viewed from the drop's frame of reference, where the velocity on the interface is tangential. The addition of surfactants to the drop evidently makes the flow undergo a sharp change at the interface; note the pronounced change in velocity profiles as we move from left to right in the top row of the figure, corresponding to increasing the level of surfactants on the interface while keeping the diffusion rate constant. A similar, but less pronounced, effect is visible on the bottom row, where the surfactant level is kept constant while the diffusion is decreased (Pe_Γ increased). The velocity change at the interface is caused by the Marangoni force (14), which increases both when the surfactant concentration and its surface gradient increase. Decreasing the diffusion causes a stronger surfactant gradient on the interface (see Figure 12), hence the larger Maragoni force when Pe_Γ is increased.

It is worth noting that the conservation error $|R(t) - R(0)|/|R(0)|$, $R(t) = \int_{\Gamma(t)} \rho(s,t)\mathrm{d}s$, is less than 1% at the final time $t = 5$ in all of these cases, suggesting that the method shows good conservation of surfactants even though it is not exactly numerically conservative.

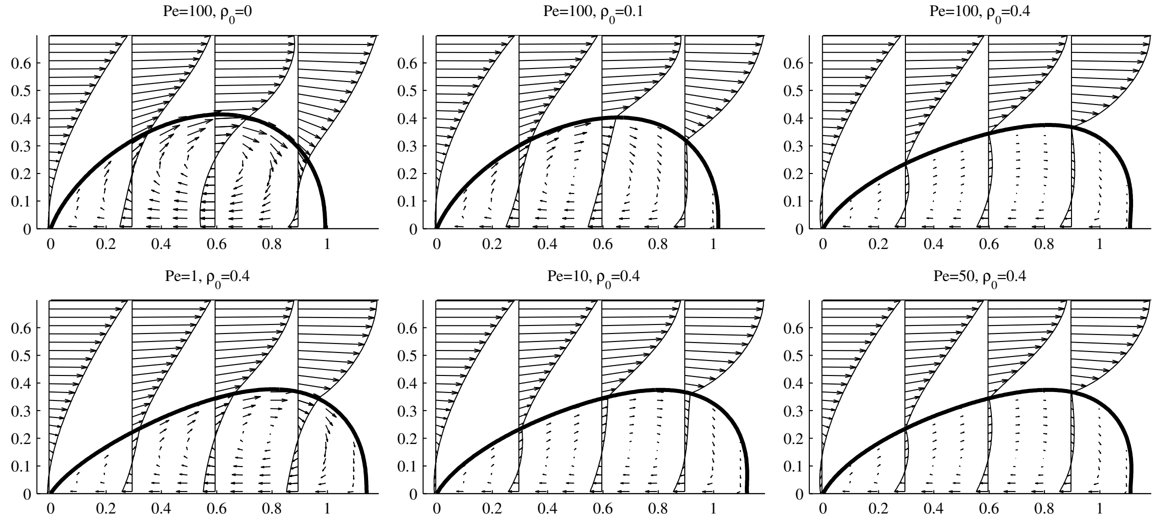


Figure 11: Drop shape at steady state with velocity field viewed from the drop's frame of reference. Top row: $\rho_0 = 0$, $\rho_0 = 0.1$ and $\rho_0 = 0.4$ with $\mathrm{Pe}_\Gamma = 100$. Bottom row: $\mathrm{Pe}_\Gamma = 1$, $\mathrm{Pe}_\Gamma = 10$ and $\mathrm{Pe}_\Gamma = 50$ with $\rho_0 = 0.4$.

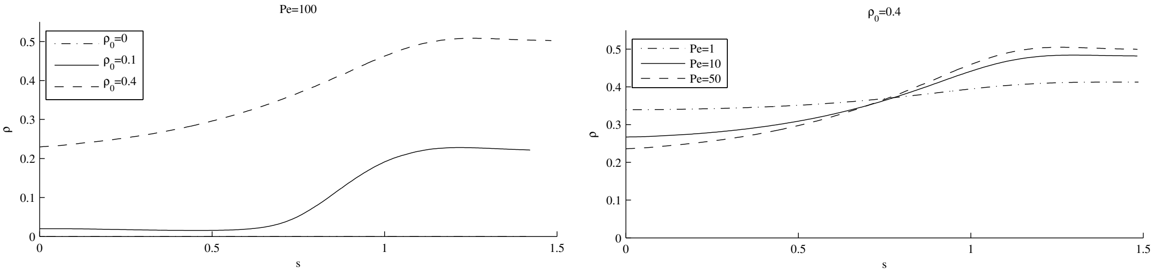


Figure 12: Surfactant concentration against arc length. Left: $\mathrm{Pe}_\Gamma = 100$ and varying ρ_0. Right: $\rho_0 = 0.4$ and varying Pe_Γ.

6.4 Spreading drop

As an illustration of what happens to a wetting drop when surfactants are added, we consider a case where surfactants are added to the surface of an initially clean, partially wetting drop ($\theta_{s,c} = 70°$) resting on a substrate, thus increasing the wetting surface of the drop. As a test of robustness, the same total amount of surfactants are added to the interface in two different initial distributions, one flat and one in the shape of a bell curve (Figure 13a). The results of the numerical simulation, presented in Figure 13, show that there is a brief period of transition during which the drop relaxes to a new equilibrium state with a smaller contact angle. The same final state is reached from both initial distributions, even though the paths taken through the transition differ, with differences between the final interface positions and surfactant distributions of magnitude $O(10^{-4})$ in maximum norm. This indicates that the computational method is robust, in the sense the equilibrium state reached is independent of the initial state, as long as it is symmetric (otherwise there will be a net translational movement of the drop).

Parameters used in the simulation are $\mathrm{Re} = 10$, $\mathrm{Ca} = 0.1$, $\mathrm{Pe}_\Gamma = 100$, $\mathrm{E} = 0.95$, $\mathcal{B} = 0.5$, $\lambda_s = 0.05$, $w_s = 0.05$, $\Delta x = 0.005$ and $\Delta t = \Delta x/2$.

7 Conclusions

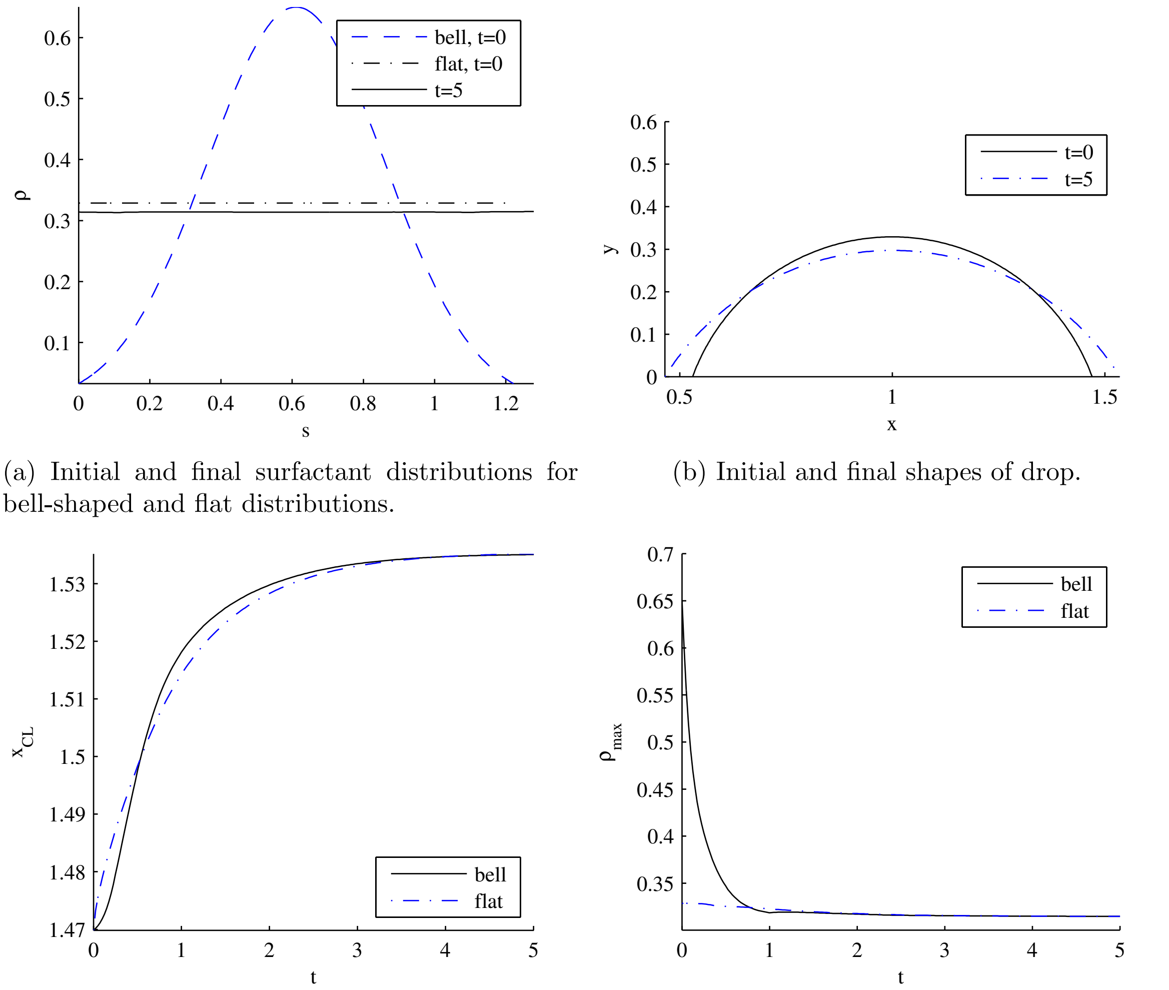
In this paper we have presented a numerical method for two-phase flow with insoluble surfactant and contact line dynamics in two dimensions. We use a domain decomposition approach to represent the interface and formulate its time-evolution. The interface is represented as a union of segments, where each segment defines a mapping (3) to a local coordinate frame and a function, $f(\xi, t)$, that defines the interface location in local coordinates (cf. Figure 2). On each segment, a PDE in local coordinates (6) describes the motion of the interface given a convective field, $\mathbf{u}$.

We argue that this is a natural framework to pose complex modelling problems in. To demonstrate this, we consider a surfactant concentration, ρ, defined on the interface. We apply the domain decomposition idea to the PDE that governs surfactant transport and diffusion (1), obtaining a set of equations (10) for the surfactant concentration $\rho(\xi, t)$ in local coordinates on each segment.

The resulting system is a parametrization, $(\xi, f(\xi), \rho(\xi))$, of both the interface and its corresponding surfactant concentration. We contend that this is a significant feature, because a vast family of numerical methods are available, efficient and well understood for PDEs on uniform grids. In Section 5 we deploy such methods for the resulting systems of equations, and we are able to keep the time step Δt proportional to the grid size Δx by solving the surfactant diffusion equation using an implicit method. *Vis-à-vis* front tracking methods we do not need to discretize equations with respect to e.g. arc-length parametrization.

Moreover, we consider the contact-point (wetting) problem [16]. This adds a solid-interface contact angle (cf. Figure 1) that introduces an additional set of physical models to the computation (Section 4.2). We discuss this in the context of an immersed boundary method for the Navier-Stokes equations (Section 4). In the literature there are several approaches to incorporating the forces generated at the contact point into the multi-phase flow solver in a consistent manner; this is an area of ongoing research. While we do not claim to have a contribution to the theoretical understanding of wetting, we do believe that given a model for the contact point dynamics, such as the one by Ren and E [49], our method can be used to study the dynamics implied by the model.

We give numerical results in Section 6, including results for the full problem with both surfactants and contact point dynamics. We observe that the presence of surfactants can have large effects on the dynamics of two-phase flows, mainly through the addition of Marangoni forces tangential to the interface.



(a) Initial and final surfactant distributions for bell-shaped and flat distributions.

(b) Initial and final shapes of drop.

(c) Path of right contact point during drop relaxation for the two distributions, $|x_{CL,\text{bell}}(5) - x_{CL,\text{flat}}(5)| = 1.42 \cdot 10^{-4}$.

(d) Maximum surfactant distribution on interface over time for the two distributions, $|\rho_{max,\text{bell}}(5) - \rho_{max,\text{flat}}(5)| = 1.49 \cdot 10^{-4}$.

Figure 13: Drop initially at rest spreading due to the addition of surfactants. The same total amount of surfactants is added in two different initial distributions. The same final state is reached in both cases. Note that the length of the interface increases during the transition, causing a decrease in the average value of ρ.

Acknowledgments

This work has been supported by the Linné Flow Centre and by the Swedish Research Council under Contract No. 621-2007-6375. The authors gratefully acknowledge this support.

References

[1] S. Adami, X. Y. Hu, and N. A. Adams. A conservative SPH method for surfactant dynamics. *J. Comput. Phys.*, 229:1909–1926, 2010.

[2] D. M. Anderson, G. B. McFadden, and A. A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annu. Rev. Fluid Mech.*, 30:139–165, 1998.

[3] S. L. Anna, N. Bontoux, and H. A. Stone. Formation of dispersions using "flow focusing" in microchannels. *Appl. Phys. Lett.*, 82:364–366, 2003.

[4] S. L. Anna and H. Mayer. Microscale tipstreaming in a microfluidic flow focusing device. *Phys. Fluids.*, 18:121512, 2006.

[5] E. Aulisa, S. Manservisi, and R. Scardovelli. A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows. *J. Comput. Phys.*, 188:611–639, 2003.

[6] T. D. Blake. The physics of moving wetting lines. *J. Colloid Interf. Sci,*, 299:1–13, 2006.

[7] D. Bonn, J. Eggers, J. Indekeu, J. Meunier, and E. Rolley. Wetting and spreading. *Rev. Mod. Phys.*, 81:739–805, 2009.

[8] M. R. Booty and M. Siegel. A hybrid numerical method for interfacial fluid flow with soluble surfactant. *J. Comput. Phys.*, 229:3864–3883, 2010.

[9] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface-tension. *J. Comput. Phys*, 100:335–354, 1992.

[10] A. Branger and D. Eckmann. Accelerated arteriolar gas embolism reabsorption by an exogenous surfactant. *Anesthesiology*, 96:971–979, 2002.

[11] C. J. Budd and A. Iserles. Geometric integration: numerical solution of differential equations on manifolds. *Philos. T. R. Soc. A*, 357:945–956, 1999.

[12] H. D. Ceniceros and J. E. Fisher. A fast, robust, and non-stiff immersed boundary method. *J. Comput. Phys.*, 230:5133 – 5153, 2011.

[13] A. J. Chorin. Numerical Solution of the Navier-Stokes Equations. *Math. Comput.*, 22:745–762, 1968.

[14] M. A. Clay and M. J. Miksis. Effects of surfactant on droplet spreading. *Phys. Fluids*, 16:3070, 2004.

[15] R. G. Cox. The dynamics of the spreading of liquids on a solid surface. Part 2. Surfactants. *J. Fluid Mech.*, 168:195–220, 1986.

[16] P. G. de Gennes. Wetting: statics and dynamics. *Rev. Mod. Phys.*, 57:827–863, 1985.

[17] J. Demmel. *Applied Numerical Linear Algebra.* SIAM, 1997.

[18] W. Dijkhuizen, I. Roghair, M. Van Sint Annaland, and J.a.M. Kuipers. DNS of gas bubbles behaviour using an improved 3D front tracking mode–Model development. *Chem. Eng. Sci.*, 65(4):1427–1437, 2010.

[19] M. Drumright-Clarke and Y. Renardy. The effect of insoluble surfactant at dilute concentration on drop breakup under shear with inertia. *Phys. Fluids.*, 16:14–21, 2004.

[20] C. Eggleton and K. Stebe. An adsorption-desorption-controlled surfactant on a deforming droplet. *J. Colloid Interf. Sci.*, 208:68–80, 1998.

[21] B. Engquist, O. Runborg, and A.-K. Tornberg. High-frequency wave propagation by the segment projection method. *J. Comput. Phys*, 178(2):373–390, 2002.

[22] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys*, 183:83–116, 2002.

[23] M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, and M. W. Williams. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *J. Comput. Phys.*, 213(1):141–173, 2006.

[24] D. Gaudlitz and N. A. Adams. On improving mass-conservation properties of the hybrid particle-level-set method. *Comput. Fluids*, 37:1320–1331, 2008.

[25] J. Geiser. *Decomposition Methods for Differential Equations: Theory and Applications.* CRC Press, 2009.

[26] J. L. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Comput. Meth. Appl. Mech. Eng.*, 195:6011–6045, 2006.

[27] J. L. Guermond and J. Shen. On the error estimates for the rotational pressure-correction projection methods. *Math. Comput.*, 73:1719–1737, 2004.

[28] A. James and J. Lowengrub. A surfactant-conserving volume-of-fluid method for interfacial flows with insoluble surfactants. *J. Comput. Phys.*, 201:685–722, 2004.

[29] F. Jin, N. Gupta, and K. Stebe. The detachment of a viscous drop in a viscous solution in the presence of a soluble surfactant. *Phys. Fluids.*, 18:022103, 2006.

[30] S. Khatri. *A Numerical Method for Two Phase Flows with Insoluble and Soluble Surfactants.* PhD thesis, Courant Institute of Mathematical Sciences, New York University, 2009.

[31] S. Khatri and A.-K. Tornberg. A numerical method for two phase flows with insoluble surfactants. *Comput. Fluids*, 49:150–165, 2011.

[32] M. Lai, Y.-H. Tseng, and H. Huang. An immersed boundary method for interfacial flows with insoluble surfactants. *J. Comput. Phys.*, 227:7279–7293, 2008.

[33] M. Lai, Y.-H. Tseng, and H. Huang. Numerical simulation of moving contact lines with surfactant by immersed boundary method. *Commun. Comput. Phys.*, 8:735–757, 2010.

[34] I. Langmuir. Surface chemistry. Nobel Lecture, 1932. Retreived Oct. 24 2011, http://www.nobelprize.org/nobel_prizes/chemistry/laureates/1932/langmuir-lecture.pdf.

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

[36] L. Lee and R. J. Leveque. An immersed interface method for incompressible Navier-Stokes equations. *SIAM J. Sci. Comput.*, 25:832–856, 2003.

[37] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002.

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

[39] R. J. LeVeque and Z. Li. Immersed interface methods for stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput.*, 18:709–735, 1997.

[40] D. Lindbo and A.-K. Tornberg. Interface tracking using patches. *arXiv:1302.4648 [physics.flu-dyn]*, 2010.

[41] H. Liu and Y. Zhang. Phase-field modeling droplet dynamics with soluble surfactants. *J. Comput. Phys.*, 229:9166–9187, 2010.

[42] R. I. McLachlan and G. R. W. Quispel. Splitting methods. *Acta Numerica*, 11:341–434, 2002.

[43] M. Muradoglu and S. Tasoglu. A front-tracking method for computational modeling of impact and spreading of viscous droplets on solid walls. *Comput. Fluids*, 39:615–625, 2010.

[44] M. Muradoglu and G. Tryggvason. A front-tracking method for computation of interfacial flows with soluble surfactants. *J. Comput. Phys.*, 227:2238–2262, 2008.

[45] Y. Pawar and K. Stebe. Marangoni effects on drop deformations in an extensional flow: The role of surfactant physical chemistry. I. insoluble surfactants. *Phys. Fluids*, 8:1738–1751, 1996.

[46] C. S. Peskin. Numerical-analysis of blood-flow in heart. *J. Comput. Phys.*, 25:220–252, 1977.

[47] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.

[48] E. Ramé. The spreading of surfactant-laden liquids with surfactant transfer through the contact line. *J. Fluid Mech.*, 440:205–234, 2001.

[49] W. Ren and W. E. Boundary conditions for the moving contact line problem. *Phys. Fluids*, 19, 2007.

[50] W. Ren and W. E. Contact line dynamics on heterogeneous surfaces. *Phys. Fluids*, 23(7):072103, 2011.

[51] W. Ren, D. Hu, and W. E. Continuum models for the contact line problem. *Phys. Fluids*, 22:102103, 2010.

[52] Y. Renardy, M. Renardy, and V. Cristini. A new volume-of-fluid formulation for surfactants and simulations of drop deformation under shear at a low viscosity ratio. *Eur. J. Mech. B - Fluid*, 21:49–59, 2002.

[53] M. J. Rosen. *Surfactants and interfacial phenomena.* Wiley, Hoboken, N.J., 3. ed. edition, 2004.

[54] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech.*, 31:567–603, 1999.

[55] J. A. Sethian and P. Smereka. Level set methods for fluid interfaces. *Annu. Rev. Fluid Mech.*, 35:341–372, 2003.

[56] S. Shin and D. Juric. Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity. *J. Comput. Phys.*, 180:427–470, 2002.

[57] H. A. Stone. A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface. *Phys. Fluids A-Fluid*, 2:111–112, 1990.

[58] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:pp. 506–517, 1968.

[59] M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *J. Comput. Phys.*, 187:110–136, 2003.

[60] M. Sussman and M. Ohta. A stable and efficient method for treating surface tension in incompressible two-phase flow. *SIAM J. Sci. Comput.*, 31(4):2447–2471, 2009.

[61] M. Sussman and E. G. Puckett. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *J. Comput. Phys.*, 162:301–337, 2000.

[62] K. E. Teigen, P. Song, J. Lowengrub, and A. Voigt. A diffuse-interface method for two-phase flows with soluble surfactants. *J. Comput. Phys.*, 230:375–393, 2011.

[63] L. J. P. Timmermans, P. D. Minev, and F. N. van de Vosse. An approximate projection scheme for incompressible flow using spectral elements. *Int. J. Num. Meth. Fl.*, 22:673–688, 1996.

[64] A.-K. Tornberg and B. Engquist. The segment projection method for interface tracking. *Commun. on Pure and Appl. Math.*, 56(1):47–79, 2003.

[65] A.-K. Tornberg and B. Engquist. Numerical approximations of singular source terms in differential equations. *J. Comput. Phys*, 200:462–488, 2004.

[66] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y. J. Jan. A front-tracking method for the computations of multiphase flow. *J. Comput. Phys*, 169:708–759, 2001.

[67] H. Wong, D. Rumschitzki, and C. Maldarelli. On the surfactant mass balance at a deforming fluid interface. *Phys. Fluids*, 8:3203–3204, 1996.

[68] M. Wörner. Numerical modeling of multiphase flows in microfluidics and micro process engineering: a review of methods and applications. *Microfluid. and Nanofluidics*, 12(6):841–886, 2012.

[69] J. J. Xu, Z. L. Li, J. Lowengrub, and H. K. Zhao. A level-set method for interfacial flows with surfactant. *J. Comput. Phys.*, 212:590–616, 2006.

[70] J. J. Xu, Yin Y., and J. Lowengrub. A level-set continuum method for two-phase flows with insoluble surfactant. *J. Comput. Phys.*, 231:5897–5909, 2012.

[71] S. Yon and C. Pozrikidis. A finite-volume/boundary-element method for flow past interfaces in the presence of surfactants, with application to shear flow past a viscous drop. *Comput. Fluids*, 27:879–902, 1998.

[72] S. Zahedi, K. Gustavsson, and G. Kreiss. A conservative level set method for contact line dynamics. *J. Comput. Phys.*, 228:6361–6375, 2009.

[73] S. Zahedi, M. Kronbichler, and G. Kreiss. Spurious currents in finite element based level set methods for two-phase flow. *Int. J. Num. Meth. Fl.*, 69(9):1433–1456, 2012.

[74] J. Zhang, D. M. Eckmann, and P. S. Ayyaswamy. A front tracking method for a deformable intravascular bubble in a tube with soluble surfactant transport. *J. Comput. Phys.*, 214:366–396, 2006.