Numerical Modeling of Surfactant Effects in Interfacial Fluid Dynamics

Ashley J. James and John Lowengrub

Abstract. Surfactants are substances that preferentially accumulate at interfaces between two fluids, altering the local surface tension. An imposed flow can produce a non-uniform distribution of surfactant. In regions of high surfactant concentration the surface tension is low, so the interface offers less resistance to deformation and can become highly curved, allowing very small droplets or bubbles to pinch off. A numerical method to simulate interfacial surfactant mechanics within a volume of fluid method has been developed. To conserve surfactant, the surfactant mass and the interfacial surface area are tracked as the interface evolves, and then the surfactant concentration is reconstructed. The algorithm is coupled to an incompressible flow solver that uses a continuum method to incorporate both the normal and tangential components of the surface tension force into the momentum equation.

1. Introduction

Surfactant plays a critical role in numerous important industrial and biomedical applications. For example, the formation of very small drops or bubbles by tip streaming relies on the presence of surfactant. The production of such tiny droplets is useful in drug delivery, industrial emulsification, liquid/liquid extraction, polymer blending and plastic production, and other applications.

Surfactants adhere to interfaces resulting in a lowered, non-uniform surface tension along the interface. This makes the capillary force non-uniform and introduces the Marangoni force. Interfacial surfactant is transported with the interface by convection, and may diffuse along the interface in the presence of a surfactant concentration gradient. Additionally, compression or stretching of the interface causes a corresponding increase or decrease in the concentration. The equation that governs these dynamics has been derived in various forms in [1, 2]. The motion of the surfactant and of the surrounding bulk fluids are coupled through the Marangoni force.

We simulate interface dynamics using the volume-of-fluid (VOF) method [3]. The main advantages of the method are that the interface shape is not constrained,

Received by the editors September 1, 2005.

Key words and phrases. surfactant, VOF, interfacial flow, surface tension.
changes in topology are handled automatically, and mass of each flow component is conserved exactly. The interface location is captured as it moves through the grid by tracking the local volume fraction. Flow discontinuities are smoothed and the surface tension force is distributed over a thin layer near the interface.

Continuum formulations of the interface governing equations have been implemented numerically primarily for clean drops, although Jan and Tryggvason [4] studied the effect of surfactants on rising bubbles using an immersed boundary/front tracking method and Ceniceros [5] used a hybrid level-set/front tracking method to study the effect of surfactants on capillary waves. Xu and Zhao [6] simulated surfactant transport on a deformable interface in conjunction with a level set method. They did not couple their method to a flow solver, but presented test cases in which a velocity field is prescribed. Renardy et al. [7] presented simulations of drops with surfactant using the VOF method. This work thus far has been limited to assuming a linear relation between the surfactant concentration and surface tension.

In the current paper we present a numerical method that incorporates surfactant dynamics in an axisymmetric, incompressible Navier-Stokes solver based on the VOF method for interface capturing. We focus on the case of insoluble surfactant. The surfactant mass is exactly conserved along the interface by our algorithm. An arbitrary equation of state relating the surfactant concentration to the surface tension may be used. Further details and verification are given in [8].

2. Governing Equations

We assume that the flow is incompressible in both fluids, so the velocity, \( \mathbf{u} \), is divergence free, \( \nabla \cdot \mathbf{u} = 0 \). The VOF method is used to track the interface between the two fluids, called fluid 1 and fluid 2. In this method a volume fraction, \( F \), is defined in each grid cell as the fraction of the cell that contains fluid 1. The volume fraction evolution is governed by a convection equation that ensures the interface moves with the velocity of the fluid,

\[
\frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0.
\]

Surface tension is included in the momentum equation via the continuum surface force (CSF) method [9]. The momentum equation satisfies the stress balance boundary condition on the interface. The surface tension force is nonzero only near the interface. The VOF and CSF methods make it unnecessary to apply boundary conditions at the interface and one set of governing equations applies to the entire domain. Thus, the density and viscosity must be retained as variables in the momentum equation even though they are both constant in each fluid. Using inertial time and pressure scales, the momentum equation is

\[
\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \frac{1}{Re} \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] + \frac{1}{ReCa} F_S,
\]

where \( \rho \) is the density, \( \mu \) is the dynamic viscosity, \( \nabla \) is the gradient operator, \( \nabla \cdot \) is the divergence operator, \( \nabla^T \) is the transpose of the gradient, and \( F_S \) is the surface force due to surface tension.
where \( \rho \) is the density, \( p \) is the pressure, \( Re \) is the Reynolds number, \( \mu \) is the viscosity and \( Ca \) is the capillary number. The surface force \( F_S \) is

\[
F_S = \nabla \cdot \left[ \sigma \left( \mathbf{I} - \mathbf{n} \mathbf{n}^T \right) \delta_S \right] = -\sigma \kappa \delta_S \mathbf{n} + \delta_S \nabla_S \sigma
\]

where \( \sigma \) is the surface tension, \( \mathbf{n} \) is the unit vector normal to the interface, \( \delta_S \) is the surface delta function, \( \kappa \) is the interface curvature and \( \nabla_S \) is the surface gradient. The first term on the right hand side of Eq. (2.3) is the capillary force and the second term is the Marangoni force. The normal vector and the surface delta function are determined from the gradient of the volume fraction,

\[
\mathbf{n} = -\frac{\nabla F}{|\nabla F|}, \quad \delta_S = |\nabla F|.
\]

The surfactant concentration evolution is governed by a convection-diffusion equation with a source term to account for interfacial stretching,

\[
\frac{\partial \Gamma}{\partial t} + \mathbf{u} \cdot \nabla \Gamma = \frac{1}{Pe_S} \nabla^2 \Gamma + \Gamma \mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n},
\]

where \( \Gamma \) is the interfacial surfactant concentration, \( Pe_S \) is the surface Peclet number, and \( \nabla^2 \) is the surface Laplacian operator.

In our finite volume method, we do not solve Eq. (2.5) directly and instead relate the surfactant concentration in a finite volume to the ratio of the surfactant mass \( M \) and surface area \( A \) in that volume, i.e. \( \Gamma = M/A \). The surfactant mass and surface area are tracked independently as described below. Siegel [10] has also proposed decomposing concentration into mass and area.

The equation governing \( A \) can be written in differential form as

\[
\frac{DA}{Dt} = \frac{\partial A}{\partial t} + \mathbf{u} \cdot \nabla A = -A \left( \mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n} \right).
\]

The left-hand side of equation (2.6) is the time rate of change of the area of a material element of the interface. The right-hand side represents changes in interfacial area due to stretching. The mass of surfactant on a material element of the interface can change if there is diffusion along the interface. The governing equation is

\[
\frac{DM}{Dt} = \frac{\partial M}{\partial t} + \mathbf{u} \cdot \nabla M = \frac{A}{Pe_S} \nabla^2 \Gamma.
\]

Equation (2.5) is regained by combining equations (2.6) and (2.7) with \( \Gamma = M/A \).

Finally, an equation of state is given for the surface tension as a function of surfactant concentration. For example, the Langmuir equation of state is

\[
\sigma = \frac{1 + E \ln (1 - x\Gamma)}{1 + E \ln (1 - x)}.
\]

where \( E \) is the surfactant elasticity and \( x \) is a measure of surfactant coverage. The scaling is constructed so that the equilibrium dimensionless concentration, \( \Gamma = 1 \), corresponds to the equilibrium dimensionless surface tension, \( \sigma = 1 \).
3. **Computational Method**

3.1. **Introduction**

The axisymmetric governing equations are discretized using a finite-volume method, on a fixed, structured, uniform, staggered grid, in the \( r - z \) plane. In the staggered grid arrangement the velocity components are defined on cell faces and all other variables are defined at cell centers. The continuity and momentum equations are discretized using second-order central differences, except for the surface stress, which is described in section 3.6.

An explicit Euler time integration method is used, except that surfactant diffusion is discretized implicitly as described in section 3.5. At each time step, first the velocity and pressure are updated, using a projection method, and then the volume fraction and the surfactant distribution are updated as described below.

The volume of fluid 1 in a grid cell at the beginning of a time step is the cell volume times the volume fraction. During a time step the volume flux of fluid 1 that moves between adjacent grid cells is computed. The volume fraction at the end of a time step is then the initial volume of fluid 1 minus the net volume flux out of the cell, divided by the cell volume. Thus, although equation (2.1) governs the evolution of the volume fraction, the method actually tracks fluid volumes.

Our approach to surfactant evolution is analogous to volume fraction evolution. Surfactant mass fluxes due to convection and diffusion are computed. The concentration in a grid cell is then the surfactant mass in the cell divided by the surface area of the interface in the cell. The surface area in a cell may vary, unlike the cell volume, so it is critical to accurately track the surface area.

Surface area evolution is described below, as well as volume fraction and surfactant mass evolution. However, the “reconstruction” of the interface and the surfactant concentration are described first.

3.2. **Volume fraction and surfactant concentration reconstruction**

To convect volumes of fluid while preventing smearing of the interface normal to itself it is first necessary to reconstruct the interface from the volume fraction field. This interface reconstruction locates where the volume of fluid 1 resides in the cell, rather than assuming both fluids are distributed uniformly. Convection of surfactant similarly suffers from excessive numerical diffusion if the surfactant is assumed to uniformly distributed along the interface, so its distribution is also reconstructed.

The volume fraction distribution in a cell is determined by approximating the interface in a cell as a straight line. The line segment approximation of the interface is defined independently in each cell, so the approximate interface need not be continuous from one cell to the next. First, the normal vector is computed as the volume fraction gradient using a finite difference method with a nine-point stencil. The normal vector defines the slope of the line. The intercept is calculated iteratively so that the volume of fluid 1 defined by the line divided equals that defined by the known cell volume fraction, \( F_{i,j} \).
The surfactant concentration is reconstructed as a linear function of position, $s$, along the straight-line interface reconstruction, $\Gamma = (\nabla_S \Gamma) s + c$, where the surface gradient, $\nabla_S \Gamma$, is taken to be constant in each cell. As for the interface reconstruction, the function need not be continuous from one cell to the next. Since the concentration is only defined on the interface the surface gradient cannot be computed using a simple finite difference formula, as the normal vector is for the interface reconstruction. Instead the gradient is computed using only the two adjacent cells that contain an interface segment. This is illustrated in Figure 1 for the case in which the adjacent cells that contain an interface segment are cells $(i+1,j)$ and $(i-1,j)$. The procedure is analogous if other adjacent cells are used. The concentration gradient is the difference in concentration between the two cells divided by the distance, $L$, between their interface midpoints. For the case illustrated in Figure 1, for example

$$\nabla_S \Gamma)_{i,j} = \frac{\Gamma_{i+1,j} - \Gamma_{i-1,j}}{L}.$$  \hspace{2em} (3.1)

The intercept, $c$, in the concentration reconstruction is then computed to ensure that the average concentration defined by the reconstruction equals the known average concentration in the cell, $\Gamma_{i,j}$.

### 3.3. Volume fraction evolution

The axisymmetric, conservative governing equation for the volume fraction is

$$\frac{\partial F}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (ruF) + \frac{\partial}{\partial z} (vF) = F \left[ \frac{1}{r} \frac{\partial}{\partial r} (ru) + \frac{\partial v}{\partial z} \right].$$  \hspace{2em} (3.2)

where $u$ is the radial velocity and $v$ is the vertical velocity. The equation is split into radial and vertical directions using an intermediate volume fraction, $\tilde{F}$,

$$\frac{\tilde{F}_{i,j} - F_{i,j}}{\Delta t} + \frac{RF_{i+1/2,j}^n - RF_{i-1/2,j}^n}{2\pi r_i \Delta r \Delta z \Delta t} =$$

$$\tilde{F}_{i,j} \left( \frac{r_{i+1/2}u_{i+1/2,j}^{n+1} - r_{i-1/2}u_{i-1/2,j}^{n+1}}{r_i \Delta r} \right)$$

\hspace{2em} (3.3)
Figure 2. Calculation of volume flux. The volume flux, \( RF_{i+1/2,j} \), is denoted by the darker gray area.

\[
\frac{F_{i,j}^{n+1} - \tilde{F}_{i,j}}{\Delta t} + \frac{\tilde{Z} F_{i,j+1/2} - \tilde{Z} F_{i,j-1/2}}{2\pi r_i \Delta r \Delta z \Delta t} = \tilde{F}_{i,j} \left( \frac{v_{i,j+1/2}^{n+1} - v_{i,j-1/2}^{n+1}}{\Delta z} \right).
\]

\( RF_{i+1/2,j} \) is the volume flux of fluid 1 in the radial direction across the \((i + \frac{1}{2}, j)\) face, and \( Z F_{i,j+1/2} \) is the volume flux of fluid 1 in the vertical direction across the \((i, j + \frac{1}{2})\) face. The fluxes are calculated in one direction and used to update the volume fraction to the intermediate level. Then, using the intermediate volume fraction, the fluxes are calculated in the other direction and used to update the intermediate volume fraction to the next time level. The direction computed first is switched at each time step.

The volume flux is the amount of fluid 1 that passes through the face during the time step. This flux equals the amount of fluid 1 in the domain of dependence of the face, at the beginning of the time step. This is illustrated in Figure 2 for the case of flux across the \((i + \frac{1}{2}, j)\) face with positive radial velocity, \( u_{i+1/2,j} \). The domain of dependence is approximated by the region bounded by the face of interest, the two adjacent perpendicular grid lines, and a line parallel to the face of interest that is a distance of \( U \Delta t \) away from the face, where \( U \) is the velocity normal to the face. The flux is the intersection of the domain of dependence and the portion of the cell volume that contains fluid 1, as defined by a straight-line reconstruction of the interface.

3.4. Surface area evolution

The interfacial area in a grid cell is governed by equation (2.6). The area is updated in three steps. Two account for convection, one step in each direction, and are taken in tandem with the volume fraction. An additional step to account for stretching is taken in between the two convective steps.
\[
\begin{align*}
(3.5) \quad & \tilde{A}_{i,j} - A_{i,j}^n + RA_{i+1/2,j}^n - RA_{i-1/2,j}^n = 0 \\
(3.6) \quad & \tilde{A}_{i,j} = \tilde{A}_{i,j} - \Delta t \tilde{A}_{i,j} (n \cdot \nabla u \cdot n)_{i,j}^{n+1} \\
(3.7) \quad & A_{i,j}^{n+1} - \tilde{A}_{i,j} + \tilde{Z} A_{i,j+1/2}^{n+1} - \tilde{Z} A_{i,j-1/2}^{n+1} = 0,
\end{align*}
\]

where \( RA_{i+1/2,j}^n \) is the interfacial area flux in the radial direction across the \((i + \frac{1}{2},j)\) face, and \(ZA_{i,j+1/2}^n\) is the interfacial area flux in the vertical direction across the \((i,j + \frac{1}{2})\) face. The stretching term in equation (3.6) is evaluated with 2nd order central differences. Convective fluxes of area are computed analogously to the volume fraction fluxes, and, as for the volume fraction, the direction computed first is switched at each time step. The fluxes in one direction are used to update the area to an intermediate value in all cells, \( \tilde{A} \), at the same time the volume fraction is updated by convection in the same direction. The straight line interface reconstruction is then updated. Next, stretching is applied to update the area in all cells to \( \tilde{A} \). Finally, convective fluxes in the other direction complete the update of the area in all cells to the new time step, \( A_{i,j}^{n+1} \). This is done in conjunction with the final update of the volume fraction by convection in the same direction.

The area flux is the area in the domain of dependence at the beginning of the time step. In Figure 2 this domain is the region \( u_{i+1/2,j} \Delta t \) wide. The area of the straight line in the domain of dependence, \( A_{dod} \), and the area of the straight line in the whole cell, \( A_{il} \), are computed from the interface geometry. Unlike in the volume fraction computation, \( A_{dod} \) is not an accurate representation of the area flux. However, the straight line does provide a good representation of which part of the cell the interface is in. Thus, we assume that the fraction of the actual area in the domain of dependence equals the fraction of the area of the straight line in the domain of dependence. Thus, the area flux is computed as the fraction of the area of the straight line in the domain of dependence, \( A_{dod}/A_{il} \), times the actual cell interfacial area, \( A_{i,j} \).

### 3.5. Surfactant evolution

The evolution of the surfactant mass in a cell is governed by equation (2.7). At each time step, the mass equations are updated in three steps that correspond to convection and diffusion.

\[
\begin{align*}
(3.8) \quad & \tilde{M}_{i,j} - M_{i,j}^n + RM_{i+1/2,j}^n - RM_{i-1/2,j}^n = 0 \\
(3.9) \quad & \tilde{M}_{i,j} - \tilde{M}_{i,j} + \tilde{Z} M_{i,j+1/2}^n - \tilde{Z} M_{i,j-1/2}^n = 0 \\
(3.10) \quad & M_{i,j}^{n+1} - \tilde{M}_{i,j} = DR_{i+1/2,j}^{n+1} - DR_{i-1/2,j}^{n+1} + DZ_{i,j+1/2}^{n+1} - DZ_{i,j-1/2}^{n+1}.
\end{align*}
\]

First, the mass is updated in every cell to an intermediate level, \( \tilde{M} \), by convection in one direction, along the convection of volume fraction and interfacial area in the same direction. After this the interface approximation is reconstructed, the
area is stretched, the average concentration is updated as \( \Gamma = M/A \), and the concentration approximation is reconstructed. Next, the mass is updated in every cell by convection in the other direction to \( \dot{M} \), along with convection of volume fraction and interfacial area. The direction in which \( F \), \( A \) and \( M \) are convected first is switched at every time step to avoid skew. Then, once again, the interface approximation is reconstructed, the average concentration is updated, and the concentration approximation is reconstructed. Finally, the mass is updated in every cell to the next time level, \( M^{n+1} \), by diffusion in both directions simultaneously.

The mass fluxed by convection through a cell face during a time step equals the mass in the domain of dependence at the beginning of the step, as for volume of fluid and interfacial area. Its computation is analogous to the area flux computation. A first approximation to the flux is the integral of the concentration over the straight line in the domain of dependence, \( M_{dod} \). In computing \( M_{dod} \) it is crucial to use the linear reconstruction of the concentration, instead of simply its average value, to avoid excessive numerical diffusion. As for the area, this does not accurately represent the flux since \( M_{dod} \) is obtained using the straight line. However, \( M_{dod}/A_{dod} \) gives a consistent value for the average concentration on the portion of the interface that is convected. This is multiplied by the area flux to obtain a mass flux that is consistent with the area flux.

\[
\text{Mass Flux} = \left( \frac{M_{dod}}{A_{dod}} \right) \left( \frac{A_{i,j}}{A_{i+1,j}} \right).
\]

Next, the mass is updated to the new time step by diffusion in a single implicit step. Diffusion of surfactant across a cell face occurs only when there is an interface in both cells adjacent to the face. From Fick's Law, the radial flux across the face \((i + \frac{1}{2}, j)\), for example, is

\[
DR_{i+1/2,j}^{n+1} = \frac{\Delta t}{\rho_s} 2\pi r_{i+1/2} \left( \frac{\Gamma_{i+1,j}^{n+1} - \Gamma_{i,j}^{n+1}}{L_{i+1/2,j}} \right),
\]

where the surface gradient is approximated as the difference in the average concentration between the two adjacent cells, divided by \( L \), the distance between the midpoints of the straight-line interface reconstructions in the cells. Note that equation (3.10) is implicit, since the concentration in the flux is evaluated at the new time step. In practice this is written as an equation for concentration by dividing by the area. Since the fluxes depend on \( \Gamma_{i,j}^{n+1} \) this coupled system is solved iteratively for \( \Gamma_{i,j}^{n+1} \). The surfactant mass is then updated as \( M = \Gamma A \).

### 3.6. Surface tension force

Once the surfactant concentration distribution is known the average surface tension in each grid cell can be computed from the equation of state, (2.8). The surface stress, equation (2.3), can be written as

\[
F_s = \sigma \kappa \nabla F + \frac{\partial \sigma}{\partial s} \left| \nabla F \right| s.
\]
In the staggered grid arrangement the stress components are evaluated at cell faces. First, the curvature is computed in each grid cell center using standard methods. Next, the curvature is evaluated at each face as the average of the curvature in the two adjacent cells. The surface tension at cells faces is also computed as the average of the surface tension in the two adjacent cells, if both cells contain an interface segment. If only one of the adjacent cells contains an interface segment the surface tension in that cell is used as the surface tension at the face. If there is not an interface segment in either adjacent cell the surface tension at the face is set to zero and there is no normal force.

The surface gradient of the surface tension is non-zero only at faces for which both adjacent grid cells contain an interface segment. For such faces the gradient is computed exactly as the surface gradient of concentration is computed in evaluating surface diffusion. The magnitude of the volume fraction gradient at each face is computed using straightforward 2nd order finite difference approximations. Finally, the radial component of the surface stress, $F_R$, becomes

$$ F_{i+1/2,j} = (\sigma \kappa)_{i+1/2,j} \left( \frac{F_{i+1,j} - F_{i,j}}{\Delta r} \right) + \left( \frac{\sigma_{i+1,j} - \sigma_{i,j}}{L_{i+1/2,j}} \right) |\nabla F|_{i+1/2,j}. $$

The vertical component is analogous.

4. Conclusions and Future Work

In this paper, we presented a volume of fluid method that accounts for an evolving surface distribution of insoluble surfactant and the associated Marangoni force in an axisymmetric geometry. The masses of the fluid components and of the surfactant are exactly conserved. An arbitrary equation of state relating the surfactant concentration to the surface tension may be used.

There are several directions we will pursue in the future. To resolve the wide-ranging length and time scales inherent in interfacial flows with surfactants, we will implement adaptive mesh refinement. This is necessary to resolve, for example, very small secondary drops that may pin off from the ends of a primary drop in the presence of surfactant (tip streaming). Additionally, coupling the VOF method to a level set method can improve the accuracy of the surface tension computation [11]. We have implemented a 2D coupled level-set/volume-of-fluid algorithm on an adaptive mesh for clean drops [12]. The method uses an arbitrary Lagrangian-Eulerian method to capture the interface evolution. An analogous method to track the surfactant is currently undergoing testing.

In additional future work, we will link the volume fraction and interface area advection routines to create a higher-order, self-consistent interface reconstruction. We will also simulate the transport of soluble surfactant in the fluid bulk and transfer of surfactant between the bulk and the interface. Finally, the simulations will be generalized to three dimensions.
References


Acknowledgment
This work utilized the resources of the Minnesota Supercomputing Institute.

Department of Aerospace Engineering and Mechanics, University of Minnesota
*E-mail address: a.james@aem.umn.edu*

Department of Mathematics, University of California, Irvine
*E-mail address: lowengrb@math.uci.edu*