An Eulerian formulation for solving partial differential equations along a moving interface

Jian-Jun Xu [*] Hong-Kai Zhao [†]

Abstract

In this paper we study an Eulerian formulation for solving partial differential equations (PDE) on a moving interface. A level set function is used to represent and capture the moving interface. A dual function orthogonal to the level set function defined in a neighborhood of the interface is used to represent some associated quantity on the interface and evolves according to a PDE on the moving interface. In particular we use a convection diffusion equation for surfactant concentration on an interface passively convected in an incompressible flow as a model problem. We develop a stable and efficient semi-implicit scheme to remove the stiffness caused by surface diffusion.

KEY WORDS: Moving interfaces; level set method; surface convection and diffusion; surfactant; semi-implicit method.

1 Introduction

Many applications in fluids, materials and biology involve multiphase phenomena. In many multiphase problems the boundary between different phases can be formulated as a sharp interface. In such problems the interactions and dynamics of different phases determine the geometry and dynamics of the interface and vice versa. For a certain class of these sharp interface problems only the location and geometry of the interface is involved in the whole system. For example, for two immiscible fluids, surface tension is present on the interface and is proportional to the local curvature of the interface. In many other multiphase problems there is more complicated physics involved on the interface. For example, when there is a surfactant on the interface between two fluids, the concentration of surfactant is both advected by the ambient flow and diffused over the moving interface. On the other hand, due to the effect of surfactant,

[*]Department of Mathematics, University of California, Irvine, CA 92697, e-mail:jxu@math.uci.edu

[†]Department of Mathematics, University of California, Irvine, CA 92697, e-mail:zhao@math.uci.edu. The research is supported by NSF SCREMS grant DMS-0112416.

the surface tension depends on both the geometry of the interface and the surfactant concentration. Hence the motion and geometry of the moving interface, the surfactant distribution, and dynamics of the bulk fluids are all coupled together. Mass transport along interfaces caused by surface diffusion also occurs in many applications in materials such as in the study of sintering, grain boundary morphology, electro-migration and thin films [10, 18, 19, 17].

For multiphase problems, there is usually no uniform equation throughout the physical domain. Instead physical quantities of different phases are usually coupled together through some jump and/or flux conditions across interfaces which may have complicated geometry and topology and even develop singularities. All these complications can pose great challenges for mathematical analysis. In numerical simulations, the usual difficulties include solving PDEs in each phase coupled with jump/flux conditions across the interface, as well as tracking of the moving interface. Moreover we may need to solve PDEs along the moving interface when physics on the interface has to be involved. Typical numerical methods for the representation and tracking of moving interfaces can be classified as Lagrangian or Eulerian formulation. In Lagrangian formulation, the location of the moving interface is explicitly tracked using particles or meshes moving with the interface. When the interface undergoes complicated motion with large deformation and/or topological changes, constant remeshing and/or surgery has to be done. These procedures may become too expensive in three dimensions. Solving PDEs on the moving interface can make Lagrangian formulation even more complicated. In Eulerian formulation, such as the level set method, the moving interface is captured on a simple Cartesian grid by using a level set function. The motion and geometric information of the moving interface can then be captured in terms of the level set function. Hence, a geometric problem is turned into a PDE problem for the level set function and efficient and robust numerical schemes for PDEs can be easily adopted to deal with discontinuities and nonlinearities. Large deformations and topological changes can also be handled easily. The level set method, invented by Osher and Sethian [20], has been successfully applied to a broad range of problems in fluids, materials, image processing and computer vision. We refer the readers to the recent review article [22, 26] for more details. However, in most of previous applications of the level set method to moving interface problems, the level set function is used to capture the location and geometry of the interface only. No physics or other quantity is involved on the moving interface. In recent work [8, 6], PDEs are solved on a static implicit surface represented by the level set function, but no physical system is involved and there is a lack of analysis for the algorithms and formulations. In this paper we develop a general Eulerian framework for associating a quantity to a moving interface and solving an evolutionary PDE for the quantity on the moving interface. In particular we use the example of advection and diffusion of surfactant on a moving interface to illustrate the formulation. Based on a decomposition of the surface diffusion operator, we propose a semi-implicit numerical algorithm to remove the stiffness caused by surface diffusion. The stability and accuracy of the algorithm is studied. We present numerical examples to demonstrate the efficiency of our method. Numerical studies of the effect of surfactant using Lagrangian type of algorithms for special cases can be found in [16] and references therein. The authors have just been aware that in [3] the same problem is being studied by using level set approach at about the same time of our work.

In this paper we assume that the fluid velocity field is given and the interface is passively convected by the fluid flow, but our formulation here can be naturally combined with the immersed interface method (IIM) (see, e.g., [12, 17, 11]) to compute the velocity field of the fluid in real applications. This combination provides an Eulerian framework that can compute global quantities governed by general PDEs with jump/flux conditions at the interface.

Here is the outline of the paper. In section 2 we describe mathematical formulations for the problem and the Eulerian framework. In section 3 we develop efficient numerical algorithms to compute the solution. In section 4 we present numerical examples to illustrate the efficiency and accuracy of our formulations and algorithms. In the appendix, we analyze the stability for the proposed algorithms.

2 Basic formulations

Suppose $\Gamma(t)$ is the moving interface advected in a given velocity field $\mathbf{u}(\mathbf{x})$. Denote f to be a scalar quantity, e.g., the concentration of surfactant, defined on $\Gamma(t)$, which satisfies an evolution PDE on the moving interface. For example, surfactant on a moving interface is both advected with the interface by the fluid flow and diffused along the interface by the concentration gradient along the interface.

In the absence of diffusion, mass conservation of surfactant on a surface element $M(t)$ leads to (see e.g. [28])

$$\frac{d}{dt}\int_{M(t)} f dS = 0, \tag{1}$$

So

$$\int_{M(t)} \left(\frac{df}{dt} dS + f \frac{d}{dt} dS \right) = 0. \tag{2}$$

Let $\mathbf{n}$ be the unit outward normal of M_t. Since f is only defined along the surface, $df/dt = f_t + \mathbf{u}_s \cdot \nabla_s f$, where $\nabla_s = (I - \mathbf{n} \otimes \mathbf{n}) \cdot \nabla$ is the surface gradient and $\mathbf{u}_s = (I - \mathbf{n} \otimes \mathbf{n})\mathbf{u}$ is the tangential velocity.

In [5] it is shown that the rate of change of a material surface element with vector area $d\mathbf{S}$ is

$$\frac{d}{dt} d\mathbf{S} = d\mathbf{S} \nabla \cdot \mathbf{u} - (\nabla \mathbf{u}) \cdot d\mathbf{S} \tag{3}$$

Writing $d\mathbf{S} = \mathbf{n} dS$ and taking the inner product of $\mathbf{n}$ with (3) gives

$$\frac{d}{dt} S = dS \nabla_s \cdot \mathbf{u}. \tag{4}$$

Since $M(t)$ is an arbitrary material surface element, combining (2) and (4), we have

$$f_t + \nabla_s \cdot (f\mathbf{u}) = 0. \tag{5}$$

If we decompose the velocity $\mathbf{u}$ into the normal component $\mathbf{u}_n = \mathbf{n} \otimes \mathbf{n} \cdot \mathbf{u}$ and the tangential component $\mathbf{u}_s = \mathbf{u} - \mathbf{u}_n$, the above equation can be rewritten as

$$f_t + \nabla_s \cdot (f\mathbf{u}_s) + (\mathbf{u} \cdot \mathbf{n})\kappa f = 0, \tag{6}$$

where κ is the mean curvature (the sum of the principle curvatures). We see clearly that $\nabla_s \cdot (f\mathbf{u}_s)$ corresponds to the tangential advection of the fluid flow and the last term corresponds to surface stretching by the normal velocity. If the flow is *incompressible*, i.e. $\nabla \cdot \mathbf{u} = 0$, [13] derived the following equivalent formulation of (6):

$$f_t + \mathbf{u} \cdot \nabla f - \mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n} f = 0, \tag{7}$$

where $\mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n} = \sum_{i,j} n_i \frac{\partial u_i}{\partial x_j} n_j$.

While our later presentation is for incompressible flow, we would like to point out that the same ideas of numerical method and stability analysis can also be applied to *compressible* flow.

To include the surface diffusion, the divergence of flux which is proportional the surface gradient of the concentration, we plug in $\nabla_s \cdot (D(\mathbf{x})\nabla_s f)$ in the above formulas, where D is the diffusion tensor. Without loss of generality for our numerical formulation, we assume isotropic diffusion and normalize the diffusion constant to be one which results in a convection-diffusion equation for the surfactant concentration on the moving interface:

$$f_t + \mathbf{u} \cdot \nabla f - \mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n} f - \nabla_s^2 f = 0. \tag{8}$$

Now we derive an equivalent representation of the surface Laplacian operator which is very convenient for our Eulerian formulation and is crucial for the design of the semi-implicit scheme to remove the stiffness of surface diffusion.

Lemma 1 *If a quantity f is defined smoothly in a neighborhood of a surface Γ, the surface Laplacian of f on Γ can be decomposed as the following:*

$$\nabla_s^2 f \quad = \quad \nabla^2 f - \frac{\partial^2 f}{\partial \mathbf{n}^2} - \kappa \frac{\partial f}{\partial \mathbf{n}}. \tag{9}$$

Proof.

$$\begin{aligned} \nabla_s^2 f &= (I - \mathbf{n} \otimes \mathbf{n})\nabla \cdot (\nabla f - (\mathbf{n} \cdot \nabla f)\mathbf{n}) \\ &= \nabla \cdot (\nabla f - (\mathbf{n} \cdot \nabla f)\mathbf{n}) - (\mathbf{n} \otimes \mathbf{n})\nabla \cdot (\nabla f - (\mathbf{n} \cdot \nabla f)\mathbf{n}) \end{aligned} \tag{10}$$

Using chain rule and tensor calculus,

$$\begin{aligned} &\nabla \cdot (\nabla f - (\mathbf{n} \cdot \nabla f)\mathbf{n}) \\ &= \nabla^2 f - (\mathbf{n} \cdot \nabla f)(\nabla \cdot \mathbf{n}) - \nabla(\mathbf{n} \cdot \nabla f) \cdot \mathbf{n} \\ &= \nabla^2 f - \kappa \frac{\partial f}{\partial \mathbf{n}} - \mathbf{n} \cdot (\nabla \mathbf{n}) \cdot \nabla f - \mathbf{n} \cdot D^2 f \cdot \mathbf{n}, \end{aligned} \tag{11}$$

where $D^2 f$ is the Hessian of f.

For any vector function $\mathbf{v}$, $(\mathbf{n}\otimes\mathbf{n})\nabla\cdot\mathbf{v} = \mathbf{n}\cdot(\nabla\mathbf{v})\cdot\mathbf{n}$, so

$$(\mathbf{n}\otimes\mathbf{n})\nabla\cdot\nabla f = \mathbf{n}\cdot D^2(f)\cdot\mathbf{n}, \tag{12}$$

and

$$\begin{aligned}
&(\mathbf{n}\otimes\mathbf{n})\nabla\cdot[(\mathbf{n}\cdot\nabla f)\mathbf{n}] \\
&= \mathbf{n}\cdot[(\mathbf{n}\cdot\nabla f)(\nabla\mathbf{n}) + \nabla(\mathbf{n}\cdot\nabla f)\otimes\mathbf{n}]\cdot\mathbf{n} \\
&= (\mathbf{n}\cdot\nabla f)\mathbf{n}\cdot(\nabla\mathbf{n})\cdot\mathbf{n} + \mathbf{n}\cdot(\nabla\mathbf{n})\cdot\nabla f + \mathbf{n}\cdot D^2 f\cdot\mathbf{n} \\
&= \mathbf{n}\cdot(\nabla\mathbf{n})\cdot\nabla f + \mathbf{n}D^2 f\mathbf{n}
\end{aligned} \tag{13}$$

here we used $\mathbf{n}\cdot\mathbf{n}=1$, so $(\nabla\mathbf{n})\cdot\mathbf{n}=0$. Plugging (11), (12) and (13) into (10), we get (9). $\diamond$

In our Eulerian formulation we use the level set method to capture the moving interface $\Gamma(t)$ which is convected in a velocity field $\mathbf{u}(\mathbf{x})$, i.e., $\Gamma(t)$ is represented as the zero level set of a level set function $\phi(\mathbf{x},t)$ which satisfies

$$\phi_t + \mathbf{u}\cdot\nabla\phi = 0. \tag{14}$$

To capture the evolution of a quantity $f(\mathbf{x},t)$ on the moving interface, we introduce another scalar function $\tilde{f}(\mathbf{x},t)$ in a neighborhood of the interface such that $\tilde{f}(\mathbf{x},t) = f(\mathbf{x},t)$ for $\mathbf{x}\in\Gamma(t)$, (i.e., $\phi(\mathbf{x},t)=0$). Just like in the level set formulation, the definition of a level set function is arbitrary as long as the zero level set agrees with the interface, the scalar function $\tilde{f}$ can also be defined arbitrarily as long as it agrees with f at the interface. In some applications, the quantity f may have a natural extension off the interface. For instance, if the surfactant has a bulk distribution. In other applications, f may be a physical quantity only defined on the interface. Then we can use a standard numerical extension procedure used in level set method, which was first proposed and studied in [33], to extend the quantity off the interface while keeping its values on the interface unchanged.

From the level set function ϕ, we can compute geometric quantities of the interface easily, e.g., the unit normal $n(\mathbf{x})$ and the mean curvature $\kappa(\mathbf{x})$,

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}, \quad \kappa(\mathbf{x}) = \nabla\cdot\left(\frac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}\right). \tag{15}$$

So the surface gradient operator and surface Laplacian operator in (9) are all well defined and we can solve the corresponding PDE for f such as (8) in a neighborhood of the interface in an Eulerian framework. Actually $\mathbf{n}(\mathbf{x})$ and $\kappa(\mathbf{x})$ are the unit normal and mean curvature respectively of the level set that passes through $\mathbf{x}$. Just like in the level set formulation, the motion law for the zero level set of ϕ, which corresponds to that of the moving interface Γ, is extended to other level sets of ϕ, the quantity f and the associated PDE on the interface are also extended to other level sets of ϕ in a neighborhood of the interface. Since the PDE of f is purely tangential to the level set, the evolution of $\tilde{f}$ on different level sets does not interfere with each other.

Mathematically, the definition of a level set function is arbitrary as long as the zero level set agrees with the interface. However, for numerical reasons, the best choice of the level set function is the signed distance function, i.e., $|\nabla\phi| = 1$, to the interface so that the underlying grid yields the best resolution and accuracy for the level set function. For the same reason, the best extension of the quantity f off the interface is the orthogonal extension, i.e., $\nabla\tilde{f} \cdot \nabla\phi = 0$. Just as the level set function does not remain as a signed distance function to the moving interface and we have to use a reinitialization process to redistance the level set function near the interface, we need an extension process to maintain the orthogonality between $\tilde{f}$ and ϕ. A PDE based method was proposed in [33] for the extension and was later further discussed and used in [9, 2]. We also combine with the local level set method to reduce computation cost. Details will be discussed in the next section.

With no confusion, we drop the $\tilde{}$ for $\tilde{f}$ from now on.

3 Numerical algorithms

Based on the Eulerian formulations developed in section 2, we can capture a moving interface and solve PDE(s) for some quantity f on the moving interface using a simple Cartesian grid. The initial setup includes the velocity field $\mathbf{u}$, the interface (represented by a level set function ϕ), and the distribution of f on the interface (extended in a neighborhood of the interface). The outline of the main procedures in one time step is as follows.

step 1: Evolve the quantity f on the moving interface, e.g., by (8).

step 2: Evolve the interface in the velocity field by (14).

step 3: (if needed) Reinitialize the level set function ϕ to be the signed distance function and extend the quantity f off the interface orthogonal to ϕ.

step 4: Use the updated interface and distribution of f to update the velocity field.

Here are more detailed descriptions for each step.

step 1: When the evolution of the quantity f on the moving interface involves surface diffusion, CFL condition for stability would require the time step $\Delta t = O(\Delta x^2)$ for any explicit discretization scheme in time, where Δx is the spatial grid size. This can be a bottle-neck for the whole numerical computation. For example, in our model problem for a moving interface with surfactant, the interface is simply convected in a velocity field. The corresponding convection equation (14) allows a time step $\Delta t = O(\Delta x)$. Here we design a second order semi-implicit scheme for the surface Laplacian operator to remove the stiffness of surface diffusion. According to (9), the surface Laplacian operator ∇_s^2 can be decomposed as

$$\nabla_s^2 = \nabla^2 - \frac{\partial^2}{\partial \mathbf{n}^2} - \kappa \frac{\partial}{\partial \mathbf{n}}.$$

The leading order term can be interpreted as the standard Laplacian minus the second derivative in the normal direction, i.e., an isotropic diffusion minus diffusion in the normal direction. From this structure, we use an implicit scheme on the standard Laplacian and an explicit one on all remaining terms. Since the standard Laplacian already includes the diffusion in the normal direction which is treated implicitly, the explicit treatment of the term $-\frac{\partial^2 f}{\partial \mathbf{n}^2}$ will not cause instability. In some sense this decomposition and treatment of surface Laplacian operator is similar to the idea of T. Dupont for parabolic equations with anisotropic diffusion, which the authors learned from [7]. By adding and subtracting an isotropic diffusion term that dominates the anisotropic diffusion term and by making the isotropic diffusion term with right sign implicit and all other terms explicit, the scheme becomes unconditionally stable. For example, to solve

$$u_t = \nabla \cdot (a(\mathbf{x})\nabla u), \quad \text{where} \quad C \geq a(\mathbf{x}) \geq c > 0, \tag{16}$$

which is mathematically the same as

$$u_t = A\nabla^2 u + \nabla \cdot (a(\mathbf{x})\nabla u) - A\nabla^2 u, \tag{17}$$

the following semi-implicit scheme, which is stable as long as $A \geq C$, can be used:

$$\frac{u^{m+1} - u^m}{\Delta t} = A\nabla^2 u^{m+1} + \nabla \cdot (a(\mathbf{x})\nabla u^m) - A\nabla^2 u^m, \tag{18}$$

where m is the time step. In our semi-implicit scheme we add and subtract the diffusion in the normal direction (plus some lower order terms) to make the surface diffusion operator a standard Laplacain operator. One can construct semi-implicit backward Euler scheme as (25), however, it is only of first-order accuracy in time. The standard Crank-Nicholson scheme (55) achieves second-order accuracy, however, it is difficult to implement. We propose the following semi-implicit modified Crank-Nicholson scheme for the time discretization of the convection diffusion equation (8), which is of second-order accuracy in time and stable under condition $\Delta t = O(\Delta x)$.

$$\begin{aligned} \frac{f^{m+1} - f^m}{\Delta t} = \quad & \frac{\nabla^2 f^{m+1} + \nabla^2 f^m}{2} + \frac{3}{2}\left[-\kappa \frac{\partial f}{\partial \mathbf{n}} - \frac{\partial^2 f}{\partial \mathbf{n}^2} - \mathbf{u}\cdot\nabla f + \mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}f\right]^m \\ & -\frac{1}{2}\left[-\kappa \frac{\partial f}{\partial \mathbf{n}} - \frac{\partial^2 f}{\partial \mathbf{n}^2} - \mathbf{u}\cdot\nabla f + \mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}f\right]^{m-1} \end{aligned} \tag{19}$$

We will provide stability analyses for these schemes in appendix.

For the spatial discretization, central difference schemes are used to compute $\nabla^2 f$, $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$, $\kappa = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}$, $\frac{\partial f}{\partial \mathbf{n}} = \mathbf{n}\cdot\nabla f$, $\frac{\partial^2 f}{\partial \mathbf{n}^2} = \mathbf{n}\cdot D^2 f \cdot \mathbf{n}$, and $\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}$.

Denote $\mathbf{u} = (u, v)$, the following upwind scheme is used for the convection term $\mathbf{u} \cdot \nabla f$ at grid point (i, j),

$$(u)^+ D_x^- f_{ij} + (u)^- D_x^+ f_{ij} + (v)^+ D_y^- f_{ij} + (v)^- D_y^+ f_{ij}, \tag{20}$$

where $(x)^+ = \max(x, 0)$, $(x)^- = \min(x, 0)$, and $D_x^{\pm} f_{ij}, D_y^{\pm} f_{ij}$ are the one-sided divided differences for which we use the third order weighted essentially non-oscillatory (WENO) scheme derived in [14, 15]. For example, the third order WENO approximation to $\frac{\partial g}{\partial x}$ at x_i using the left-biased stencil $\{x_k, k = i-2, i-1, i, i+1\}$ is

$$D_x^- g_i = \frac{[(\Delta^+ g_{i-1} + \Delta^+ g_i) - \omega_-(\Delta^+ g_{i-2} - 2\Delta^+ g_{i-1} + \Delta^+ g_i)]}{(2\Delta x)}, \tag{21}$$

where we use the notation Δ^+, Δ^- for the forward and backward difference operators respectively. And $\omega_- = 1/(1 + 2r^2)$ with

$$r = \frac{\epsilon + (\Delta^- \Delta^+ g_{i-1})^2}{\epsilon + (\Delta^- \Delta^+ g_i)^2}, \tag{22}$$

where ϵ is a small positive number, in our computation we take it to be 10^{-6}. The third order WENO approximation to $\frac{\partial g}{\partial x}$ at x_i on the right-biased stencil $\{x_k, k = i-1, i, i+1, i+2\}$ is

$$D_x^+ g_i = \frac{[(\Delta^+ g_{i-1} + \Delta^+ g_i) + \omega_+(\Delta^+ g_{i+1} - 2\Delta^+ g_i + \Delta^+ g_{i-1})]}{(2\Delta x)} \tag{23}$$

where $\omega_+ = 1/(1 + 2r^2)$ with

$$r = \frac{\epsilon + (\Delta^- \Delta^+ g_{i+1})^2}{\epsilon + (\Delta^- \Delta^+ g_i)^2} \tag{24}$$

The linear system for f^{m+1} from (19) is symmetric positive definite as for the standard heat equation. It can be easily solved by using SOR method or conjugate gradient method [24]. In our numerical computation in section 4 we simply use SOR method.

Since our semi-implicit scheme is a two-step method, we use the following one step semi-implicit scheme,

$$\frac{f^1 - f^0}{\Delta t} = \nabla^2 f^1 - \left[-\kappa \frac{\partial f}{\partial \mathbf{n}} - \frac{\partial^2 f}{\partial \mathbf{n}^2} - \mathbf{u} \cdot \nabla f + \mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n} f \right]^0 . \tag{25}$$

for the first time step. We use numerical examples to verify that our semi-implicit scheme is of second order accuracy and the CFL condition is $\Delta t = O(\Delta x)$ in section 4.

step 2: Evolving the level set function according to the linear convection equation (14) with a given velocity field $\mathbf{u}$ is straight forward. The same upwind scheme using third order WENO described as above is used for the discretization of $\mathbf{u} \cdot \nabla \phi$. For time discretization we use the following total variation diminishing (TVD) Runge-Kutta scheme of third-order devised in [27]. Consider the following time dependent PDE,

$$\frac{dg}{dt} = L(g), \tag{26}$$

with initial data $g(0) = g_0$, where L is some spatial operator. We march from mth step to $(m+1)$th step by

$$\begin{cases} g_1 & = g^m + \Delta t L(g^m) \\ g_2 & = \frac{3}{4} g^m + \frac{1}{4} g_1 + \frac{\Delta t}{4} L(g^1) \\ g^{m+1} & = \frac{1}{3} g^m + \frac{2}{3} g_2 + \frac{2\Delta t}{3} L(g_2). \end{cases} \tag{27}$$

step3: As we discussed before, reinitialization for the level set function ϕ to the signed distance function and extension of the quantity f off the interface orthogonally may be needed for numerical reasons. In particular, if we use the local level set method to save the computation cost, these two procedures have to be done frequently. To reinitialize the level set function, we use the PDE based approach which was proposed in [30] and was discussed in detail in [23]. After the evolution of the level set function at step m, we reinitialize it by solving the equation to steady state:

$$\begin{cases} \phi_\tau + S(\phi_0)(|\nabla \phi| - 1) = 0 \\ \phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}) = \phi^m(\mathbf{x}) \end{cases} \tag{28}$$

Where τ is the pseudo time and $S(x)$ is the sign function of x defined as

$$S(x) = \begin{cases} -1 & \text{if} \quad x < 0 \\ 0 & \text{if} \quad x = 0 \\ +1 & \text{if} \quad x > 0. \end{cases} \tag{29}$$

We use the following spatial discretization for the Hamiltonian $S(\phi_0)(|\nabla \phi| - 1)$

$$\begin{aligned} & s_{ij}^+ \left(\sqrt{(a^+)^2 + (b^-)^2 + (c^+)^2 + (d^-)^2} - 1 \right) \\ & + s_{ij}^- \left(\sqrt{(a^-)^2 + (b^+)^2 + (c^-)^2 + (d^+)^2} - 1 \right) \end{aligned} \tag{30}$$

where s_{ij} is the numerical approximation to $S(\phi_{ij}^0)$:

$$s_{ij} = \frac{\phi_{ij}^0}{\sqrt{(\phi_{ij}^0)^2 + \Delta x^2}}. \tag{31}$$

a, b, c, d in (30) denote the following one-sided difference quotients, which are computed using the third order WENO method as before:

$$\begin{aligned} a = D_x^- \phi_{ij}, \quad b = D_x^+ \phi_{ij} \\ c = D_y^- \phi_{ij}, \quad d = D_y^+ \phi_{ij} \end{aligned}$$

To extend the quantity f off the interface so that it is orthogonal to the level set function ϕ at least in a neighborhood of the interface we solve the following simple linear convection equation to steady state

$$\begin{cases} f_\tau + S(\phi)\mathbf{n}\cdot\nabla f = 0 \\ f(\mathbf{x},0) = f_0(\mathbf{x}), \end{cases} \tag{32}$$

where $S(\phi)$ is again the sign function of ϕ and $\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}$ is the unit normal. So $f(\mathbf{x},\tau) = f_0(\mathbf{x})$ if $\phi(\mathbf{x}) = 0$, i.e., for $\mathbf{x}$ on the interface. The orthogonality condition, $\nabla\phi\cdot\nabla f = 0$, becomes true very quickly near the interface and then propagates further away. The above extension method was first introduced and analyzed in [33] and later discussed in more detail in [23]. It has been successfully applied to the Stefan problem in [9]. In [2], a discrete version of the scheme was also proposed. Numerically, the second order central difference scheme is used to compute $\mathbf{n}$, and the upwind third order WENO scheme is used for ∇f in $\mathbf{n}\cdot\nabla f$.

Since in two successive time steps the solutions are nearly same, we only need a few number of iterations (i.e. pseudo time steps) for the reinitialization or extension. In the local level set method, in order to get the correct distance values for the level set function ϕ or the extended values for f at points newly added to the moving tube from outside, the number of iterations or pseudo time steps is proportional to $\frac{w}{\Delta t}$, where w is the width of the narrow tube that follows the moving interface, because the information propagates with speed one in both cases. In our computation we use the third order TVD Runge-Kutta scheme (27) for (28) and (32) in the pseudo time with $\Delta\tau = 0.2\Delta x$.

step 4: In this paper we mainly address the numerical framework and algorithms for solving PDEs on a moving interface. We assume the velocity field $\mathbf{u}(\mathbf{x})$ is a given function. In our future study we will combine our numerical methods with the immersed interface method with applications to real physical problems.

Remark 3.1: We present the method here in two dimensional formulation. The extension to three dimensions is straightforward.

Local level set method. In the local level set method, instead of solving the PDE for the level set function, e.g., (14), in the whole computation domain, we only need to restrict the computations in a small neighborhood of the interface. Similarly, we only solve the PDE for the quantity f, e.g., (8), in a neighborhood of the zero level set. In our computations we adopt the PDE based local level set method discussed in [23]. A narrow tube is constructed and updated following the moving interface. The size of the tube is fixed and can be just a few grid size wide. There is also a different localization technique introduced in [1]. The difference is that the PDE based local level set formulation involves PDEs and values of the level set function only, not the explicit location of grid points in the domain.

In our numerical algorithm, we construct three tubes around the interface with widths $0 < \gamma_1 < \gamma_2 < \gamma_3$ respectively,

$$T_1 = \{(x_i, y_j) : |\phi(x_i, y_j)| < \gamma_1\}$$

$$
\begin{aligned}
T_2 &= \{(x_i, y_j) : |\phi(x_i, y_j)| < \gamma_2\} \\
T_3 &= \{(x_i, y_j) : \min_{l,k=-1,0,1} |\phi(x_{i+l}, y_{j+k})| < \gamma_3\}
\end{aligned}
$$

The widths, γ_i's, and their differences are usually a few multiple of grid size. The choice of parameters $\gamma_1, \gamma_2, \gamma_3$ may depend on the stencils of the spatial discretization. For instance, the widest stencil used in our numerical algorithm is the third-order WENO. So we can choose $\gamma_1 = 3\Delta x, \gamma_2 = \gamma_1 + 3\Delta x$, $\gamma_3 = \gamma_2 + 3\Delta x$ for instance. These tubes have to be updated following the moving interface by adding and deleting neighboring grid points appropriately.

Since the semi-implicit scheme for the surfactant concentration f is a two-step method, we apply the backward Euler method (25) at the first time step. Suppose we have initial data: ϕ^0, ϕ^1, f^0, f^1, and initial tubes T_1, T_2 and T_3. We use steps 1-3 described above to evolve from time t^m to time t^{m+1}. Due to the use of local level set method there are a few more implementation technicalities to be clarified here.

Step 1: Solve the equation (8) for f using the semi-implicit scheme (19) to get f^{m+1} in tube T_1. The spatial discretization in (19) involves not only points in tube T_1 but also points outside of T_1 but near the boundary of T_1. Since the extension process for f is done in a larger tube T_2, we can impose the Dirichlet boundary condition for f^{m+1} at those points using extrapolations:

$$
f_{ij}^{m+1} = 2f_{ij}^m - f_{ij}^{m-1}, \forall (x_i, y_j) \in T_2 - T_1. \tag{33}
$$

step 2: We introduce the following cutoff function $c_m(\phi)$ used in [23]

$$
c_m(\phi) = \begin{cases} 0 & \text{if} \quad \phi^m > \gamma_2 \\ \frac{(|\phi^m| - \gamma_2)^2 (2|\phi^m| + \gamma_2 - 3\gamma_1)}{(\gamma_2 - \gamma_1)^3} & \text{if} \quad \gamma_1 < |\phi^m| \le \gamma_2 \\ 1 & \text{if} \quad |\phi^m| \le \gamma_1 \end{cases} \tag{34}
$$

and solve

$$
\phi_t + c_m \mathbf{u} \cdot \nabla \phi = 0 \tag{35}
$$

for the level set function in tube T_2. The reason for the introduction of the cutoff function is to prevent discontinuities in the coefficients and numerical oscillations at the tube boundary. Since we use explicit scheme in time and the reinitialization for ϕ is done in T_3 (which includes T_2), no boundary condition is needed for points near the boundary of T_2.

step 3: Reinitialize ϕ^{m+1} using (28) in tube T_3 and then extend f using (32) in tube T_2. In both equations information propagates away from the interface, and hence we do not need explicit boundary conditions when using the upwind scheme.

The reconstruction of the three tubes around the new interface from the previous ones is done by

$$
\begin{aligned}
T_1 \leftarrow \; & \{(x_i, y_j) : |\phi_{ij}^{m+1}| < \gamma_1, (x_i, y_j) \in T_3\} \\
T_2 \leftarrow \; & \{(x_i, y_j) : |\phi_{ij}^{m+1}| < \gamma_2, (x_i, y_j) \in T_3\} \\
T_3 \leftarrow \; & \{(x_i, y_j) : |\phi_{i,j}^{m+1}| < \gamma_3, (x_i, y_j) \in T_3\} \\
& \cup \{(x_{i+l}, y_{j+k}), l, k = -1, 0, 1 : |\phi_{ij}^{m+1}| < \gamma_3, (x_i, y_j) \in T_3\}
\end{aligned}
$$

4 Numerical results

In this section, we present numerical results for the model problem for surfactant concentration to demonstrate our Eulerian framework. We are going to show that the numerical algorithm we developed is stable with CFL condition $\Delta t = O(\Delta x)$ and can achieve second order accuracy. Another important property we monitor is the mass conservation. The total mass of surfactant on the interface is conserved since the surfactant is convected and diffused only along the interface. The volume (area) of the interior region enclosed by the moving interface is also conserved since we use a divergence free velocity field. In the level set formulation, the surface integral of a function $g(\mathbf{x})$ on the interface Γ represented by the level set function ϕ can be written as [33]:

$$\int_\Gamma g(\mathbf{x})ds = \int g(\mathbf{x})\delta(\phi)|\nabla\phi|d\mathbf{x}. \tag{36}$$

And the integral of $g(\mathbf{x})$ in the interior of Γ, denoted by Ω and in which the level set function ϕ is defined to be negative, can be written as:

$$\int_\Omega g(\mathbf{x})d\mathbf{x} = \int g(\mathbf{x})H(-\phi)d\mathbf{x}. \tag{37}$$

Here $\delta(x)$ is the $1D$ $\delta-$function and $H(x)$ is the $1D$ Heaviside function. Numerically we approximate the Heaviside function and δ-function by the following formulas:

$$\tilde{H}(x) = \begin{cases} 0, & \text{if} \quad x < -w, \\ -\frac{1}{6\pi}\left(1 + \frac{x}{w} + \sin\left(\frac{\pi x}{w}\right)\right), & \text{if} \quad -w \le x < -0.5w, \\ -\frac{1}{6\pi}\left(1 + \frac{x}{w} + \sin\left(\frac{\pi x}{w}\right)\right) & \\ +\frac{1}{3}\left(2 + \frac{4x}{w} + \frac{2}{\pi}\sin\left(\frac{2\pi x}{w}\right)\right), & \text{if} \quad -0.5w \le x \le 0.5w, \\ -\frac{1}{6}\left(1 + \frac{x}{w} + \frac{1}{\pi}\sin\left(\frac{\pi x}{w}\right)\right) + \frac{4}{3}, & \text{if} \quad 0.5w < x \le w \\ 1.0, & \text{if} \quad x > w \end{cases}$$

$$\tilde{\delta}(x) = \begin{cases} 0, & \text{if} \quad |x| > w, \\ -\frac{1}{6w}\left(1 + \cos\left(\frac{\pi x}{w}\right)\right) & \\ +\frac{4}{3w}\left(1 + \cos\left(\frac{2\pi x}{w}\right)\right), & \text{if} \quad |x| < 0.5w, \\ -\frac{1}{6w}\left(1 + \cos\left(\frac{\pi x}{w}\right)\right), & \text{if} \quad 0.5w \le |x| \le w \end{cases}$$

We use $w = 1.5\Delta x$ and a simple quadrature rule to compute these integrals. Our approximate δ-function satisfies $\int \tilde{\delta}(x)x^p = 0,\ p = 0, 1, 2$ and $\frac{d}{dx}\tilde{H}(x) = \tilde{\delta}(x)$.

Example 1: Here we make up a simple but non-trivial example with explicit analytical solution to check the accuracy and stability of our algorithm. Assume the interface is a stationary circle centered at the origin with radius r_0. Then the surface diffusion operator on the circle becomes $\nabla_s^2 = \frac{1}{r_0^2}\frac{\partial^2}{\partial\theta^2}$, where $\theta = \arcsin(\frac{y}{\sqrt{x^2+y^2}})$ denotes the central angle with x-axis. If the initial distribution of f is a function of θ, then the solution depends on θ only at any later time

t. The surface diffusion equation becomes the standard heat equation with periodic boundary condition,

$$\begin{cases} \dfrac{\partial f(\theta,t)}{\partial t} = \dfrac{1}{r_0^2}\dfrac{\partial^2 f(\theta,t)}{\partial \theta^2} & 0 \le \theta \le 2\pi \\ f(\theta,0) = f_0(\theta) & \\ f(0,t) = f(2\pi,t) & \end{cases}, \tag{38}$$

which can be solved using Fourier series. In particular, if $f_0(\theta) = \sin(n\theta) + c$, where c is a constant, then solution on the interface is

$$f(\theta,t) = e^{-\frac{n^2 t}{r_0^2}} \sin(n\theta) + c \tag{39}$$

In our computation, we use the level set function $\phi(x,y) = \sqrt{x^2+y^2} - r_0$, whose level sets are concentric circles. Denote $r = \sqrt{x^2+y^2}$, then

$$f(x,y,t) = e^{-\frac{n^2 t}{r^2}} \sin(n\theta) + c, \tag{40}$$

is a global solution to the PDE

$$\frac{\partial f(x,y,t)}{\partial t} = \nabla_s^2 f(x,y,t) = \nabla^2 f(x,y,t) - \frac{\partial^2 f(x,y,t)}{\partial \mathbf{n}^2(x,y)} - \kappa(x,y)\frac{\partial f(x,y,t)}{\partial \mathbf{n}(x,y)}, \tag{41}$$

where

$$\mathbf{n}(x,y) = \frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|},\ \kappa(x,y) = \nabla\cdot\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}.$$

In another word, f diffuses along every level set of ϕ according to $f_t = \nabla_s^2 f$, where s is the arc length. In our first numerical test, we choose $r_0 = 1$ and $f_0(\theta) = \sin\theta + 2$ and fix the level set function $\phi(x,y) = \sqrt{x^2+y^2} - 1$. In order to avoid singularities in the diffusion coefficient $\frac{1}{r^2}$, we solve the PDE (41) in the domain $[-2,2]\times[-2,2] - \{(x,y) : \sqrt{x^2+y^2} < 0.8\}$ using the semi-implicit scheme (19) with Dirichlet boundary condition from the exact solution. Neither reinitialization nor extension is used here.

Errors in different norm and the order of accuracy at time $t = 2$ are presented in Table 1. Here errors are measured over the entire computational domain. This example clearly shows that our semi-implicit scheme is second-order accuracy and is stable with $\Delta t = O(\Delta x)$.

EXAMPLE 2. In this example we put the above example in a simple velocity field $\mathbf{u} = (1,0)$. We compute the evolution of the interface as well as the surfactant concentration on the moving interface by (14) and (7) respectively using the algorithm described in Section 3. Local level set method is used in our computations. The computational domain is $[-3,5]\times[-3,3]$, the initial distribution of surfactant is $f(x,y,0) = \sin\theta + 2 = \frac{y}{\sqrt{x^2+y^2}} + 2$, and the initial level set function is $\phi(x,y,t) = \sqrt{x^2+y^2} - 2$. The exact solution is a simple translation of the solution in example 1 with speed 1. However, since we use the reinitialization and extension procedure in the local level set method, the solution at the interface is extended to a neighborhood of the interface. So we compare our numerical solution to the following function in a small neighborhood of the interface.

$$f(x,y,t) = e^{-\frac{t}{4}} \sin(\theta(t)) + 2, \tag{42}$$

where $\theta(t) = \arcsin(\frac{y}{\sqrt{(x-t)^2+y^2}})$

Let f_h, ϕ_h be the numerical solutions for surfactant concentration and level set function for the moving interface respectively on a grid with grid size h. Denote $e_h = |f_h - f|$ the numerical error. We measure the error by three norms:

$$\|e_h\|_{L_\infty} = max_{|\phi_h|<1.5\Delta x}\{e_h\} \tag{43}$$

$$\|e_h\|_{L_1} = \int_\Gamma e_h ds \tag{44}$$

$$\|e_h\|_{L_2} = \left(\int_\Gamma e_h^2 ds\right)^{\frac{1}{2}} \tag{45}$$

The surface integrals in $\|\cdot\|_{L_1}$ and $\|\cdot\|_{L_2}$ are computed according to (36). The errors and order of accuracy at time $t = 2$ is shown in tables 2.

Remark 4.1: There are a few issues needed to be clarified for the degeneracy of accuracy in this example. Since the local level set method is used, we have to use the reinitialization and extension procedure at every time step. There are two possible contributions of errors by this procedure. One is due to the discontinuous sign function in both (28) and (32). The other one is because we may not reach the steady state solution of (28) and (32) for just a few iterations in our computation, i.e., f may not be constant in the normal direction. So (42) may not be the exact solution we should compare to. We use the following example to illustrate our points here.

EXAMPLE 3. In this example we use the same setup as in example 2 except that we introduce an extra forcing term,

$$g(x,y,t) = -\frac{1}{4}e^{-\frac{t}{4}}\frac{y((x-t)^2+y^2-4)}{((x-t)^2+y^2)^{3/2}}, \tag{46}$$

in the equation for surfactant concentration

$$f_t = \nabla_s^2 f - \mathbf{u}\cdot\nabla f + f\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n} + g, \tag{47}$$

where $\mathbf{u} = (1,0)$. At the interface $g(x,y,t) = 0$. The purpose to add g is to make the exact solution $f(x,y,t)$ to be a constant along the normal direction of the interface, i.e.,

$$f(x,y,t) = e^{-\frac{t}{4}} \sin(\theta(t)) + 2 = e^{-\frac{t}{4}}\frac{y}{\sqrt{(x-t)^2+y^2}} + 2 \tag{48}$$

is the exact solution. So the extension process for f does not play a significant role near the interface. The errors at time $t = 2$ is shown in tables 3. Second order accuracy is observed.

Remark 4.2: In examples 2 and 3, we fix the widths of tubes for different grid sizes in order to check the accuracy.

EXAMPLE 4. We consider the same model problem in example 2 except that the initial level set function is changed to $\phi(x, y, 0) = \sqrt{x^2 + y^2} - 1$ and the computation domain is changed to $[-2, 8] \times [-2, 2]$ with $\Delta x = 0.04$, $\Delta t = \Delta x/4$. We choose $\gamma_1 = 3\Delta x, \gamma_2 = 6\Delta x$ and $\gamma_3 = 9\Delta x$, and the number of iterations for both re-initialization and extension is 3.

We present the picture of the moving interface as well as the surfactant concentration on it at different time in figure 1. The exact solution for the surfactant concentration on the moving interface is $f(\theta, t) = e^{-t}\sin\theta + 2$, where θ is the central angle. In figure 2 we show our numerical solution of the surfactant concentration as a function of θ at different time. Figure 3 is another plot corresponding to the initial surfactant concentration $f(x, y, 0) = \sin(3\theta) + 2$.

EXAMPLE 5. In this example we use a velocity field like a shear flow,

$$\mathbf{u}(x, y) = \begin{cases} (y^2, 0) & \text{if} \quad y \geq 0 \\ (-y^2, 0) & \text{if} \quad y < 0 \end{cases} \tag{49}$$

in which the interface moves and deforms. The computation domain is $\Omega = [-3, 3]^2$, the initial interface is given by the zero level set of $\phi(x, y, 0) = \sqrt{x^2 + y^2} - 1$, and the initial surfactant concentration on the interface is $f(x, y, 0) = \sin\theta + 2 = y/\sqrt{x^2 + y^2} + 2$. The grid size is $\Delta x = 0.04$, and $\Delta t = \Delta x/4$. We choose $\gamma_1 = 3\Delta x, \gamma_2 = 6\Delta x$ and $\gamma_3 = 9\Delta x$, and the number of iterations for both re-initialization and extension is 3. The moving interface and surfactant concentration on the interface at different time are shown in figure 4. Figure 5 and 6 show the relative change of the total mass of surfactant on the interface and the area enclosed by the interface respectively.

EXAMPLE 6. In this example we change the velocity field in example 5 to

$$\mathbf{u}(x, y) = \left(\frac{(y + 2)^2}{3}, 0 \right). \tag{50}$$

The initial setup is the same as in example 5. The computation domain is $\Omega = [-2, 6] \times [-2, 2]$ and the grid size is $\Delta x = 0.04$. Due to the increase of the magnitude of the velocity field we use $\Delta t = \Delta x/8$. Figure 7 shows the moving interface and surfactant concentration on the interface at different time. Figure 8 and 9 show the relative change of the total mass of surfactant on the interface and the area enclosed by the interface respectively.

A Stability analysis for the semi-implicit schemes

We use the Von Neumann analysis (see, e.g., [29]) with frozen coefficients to analyze the stability of the semi-implicit backward Euler method, Crank-Nicholson and the modified semi-implicit Crank-Nicholson scheme (19) for the equation (8). The frozen coefficient problems are constant coefficient problems obtained by fixing the coefficients at their values attained at each point in the domain of the computation. If each frozen coefficient problem is stable, then the variable coefficient problem is also stable. For simplicity, we decouple the motion of the interface from our analysis, i.e., we study the stability of our schemes on a fixed interface. We also neglect the lower order terms in the equation (8) since the stiffness comes from the second order parabolic term and the lower order terms cause a lower order perturbation. We show that all these three schemes are unconditionally stable neglecting the first order terms. Hence the explicit treatment of the lower order terms only requires $\Delta x = O(\Delta t)$ for stability, which is verified by our numerical experiments in section 4.

We use the standard five point stencil central scheme to discretize the Laplacian operator,

$$\nabla^2 f^m = \frac{f^m_{i+1,j} - 2f^m_{i,j} + f^m_{i-1,j}}{\Delta x^2} + \frac{f^m_{i,j+1} - 2f^m_{i,j} + f^m_{i,j-1}}{\Delta y^2}. \tag{51}$$

Fix $\mathbf{n} = (\cos\alpha, \sin\alpha)$ to be the normal direction at an arbitrary grid point and

$$\frac{\partial^2 f}{\partial \mathbf{n}^2} = \mathbf{n}\cdot D^2 f \cdot \mathbf{n} = \cos^2\alpha \frac{\partial^2 f}{\partial x^2} + 2\cos\alpha\sin\alpha\frac{\partial^2 f}{\partial x \partial y} + \sin^2\alpha\frac{\partial^2 f}{\partial y^2}. \tag{52}$$

Again we use central difference scheme to discretize this term as follows

$$\begin{aligned}\frac{\partial^2 f^m}{\partial \mathbf{n}^2} = \quad & \cos^2\alpha\frac{f^m_{i+1,j} - 2f^m_{i,j} + f^m_{i-1,j}}{\Delta x^2} + \sin^2\alpha\frac{f^m_{i,j+1} - 2f^m_{i,j} + f^m_{i,j-1}}{\Delta y^2}. \\ & +2\cos\alpha\sin\alpha\frac{f^m_{i+1,j+1} + f^m_{i-1,j-1} - f^m_{i-1,j+1} - f^m_{i+1,j-1}}{4\Delta x \Delta y}\end{aligned} \tag{53}$$

Neglecting the lower order terms $\kappa\frac{\partial f}{\partial \mathbf{n}}, \mathbf{u}\cdot\nabla f$ and $\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}f$, we have the semi-implicit backward Euler scheme

$$\frac{f^{m+1} - f^m}{\Delta t} = \nabla^2 f^{m+1} - \frac{\partial^2 f^m}{\partial \mathbf{n}^2}, \tag{54}$$

the Crank-Nicholson scheme is

$$\frac{f^{m+1} - f^m}{\Delta t} = \frac{\nabla^2 f^{m+1} + \nabla^2 f^m}{2} - \frac{1}{2}\left(\frac{\partial^2 f^{m+1}}{\partial \mathbf{n}^2} + \frac{\partial^2 f^m}{\partial \mathbf{n}^2}\right), \tag{55}$$

and the modified semi-implicit Crank-Nicholson scheme (19) is

$$\frac{f^{m+1} - f^m}{\Delta t} = \frac{\nabla^2 f^{m+1} + \nabla^2 f^m}{2} - \left(\frac{3}{2}\frac{\partial^2 f^m}{\partial \mathbf{n}^2} - \frac{1}{2}\frac{\partial^2 f^{m-1}}{\partial \mathbf{n}^2}\right). \tag{56}$$

Without loss of generality we assume that $\Delta x = \Delta y$.

Theorem 1 *The semi-implicit backward Euler scheme (54) and the Crank-Nicholson scheme (55) are unconditionally stable.*

Proof. Both of these two schemes are one-step method. We use the standard Von Neumann analysis. Plugging $f^m_{i,j} = g^m e^{I(i\xi_1+j\xi_2)}$ into the corresponding finite difference scheme, where I denotes the imaginary unit, we can get the amplification factor

$$g = \frac{1+4\lambda\left(\cos^2\alpha\sin^2\frac{\xi_1}{2}+2\cos\alpha\sin\alpha\sin\frac{\xi_1}{2}\cos\frac{\xi_1}{2}\sin\frac{\xi_2}{2}\cos\frac{\xi_2}{2}+\sin^2\alpha\sin^2\frac{\xi_2}{2}\right)}{1+4\lambda\left(\sin^2\frac{\xi_1}{2}+\sin^2\frac{\xi_2}{2}\right)} \tag{57}$$

for the semi-implicit backward Euler scheme and

$$g = \frac{1-2\lambda\left(\sin^2\alpha\sin^2\frac{\xi_1}{2}-2\cos\alpha\sin\alpha\sin\frac{\xi_1}{2}\cos\frac{\xi_1}{2}\sin\frac{\xi_2}{2}\cos\frac{\xi_2}{2}+\cos^2\alpha\sin^2\frac{\xi_2}{2}\right)}{1+2\lambda\left(\sin^2\alpha\sin^2\frac{\xi_1}{2}-2\cos\alpha\sin\alpha\sin\frac{\xi_1}{2}\cos\frac{\xi_1}{2}\sin\frac{\xi_2}{2}\cos\frac{\xi_2}{2}+\cos^2\alpha\sin^2\frac{\xi_2}{2}\right)} \tag{58}$$

for the Crank-Nicholson scheme. Here $\lambda = \Delta t/\Delta x$. It is easy to check that $|g| \le 1$ in both cases. $\diamond$

Remark A1: Including the lower order terms such as $\kappa\frac{\partial f}{\partial \mathbf{n}}, \mathbf{u}\cdot\nabla f$ and $\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}f$, will contribute a small perturbation of $O(\Delta t)$ (if $\Delta t = O(\Delta x)$) for the amplification factor, so the resulting scheme is still stable [29].

Now we prove the stability for the parabolic part of semi-implicit modified Crank-Nicholson method (19).

Theorem 2 *The semi-implicit scheme (56) is unconditionally stable.*

Proof. Plugging $f^m_{i,j} = g^m e^{I(i\xi_1+j\xi_2)}$ into (56), we get the characteristic polynomial for the amplification factor g,

$$P(g) = ag^2 + bg + c \tag{59}$$

where

$$\begin{aligned}
&a = 1+2\lambda\left(\sin^2\tfrac{\xi_1}{2}+\sin^2\tfrac{\xi_2}{2}\right)\\
&b = -1+2\lambda\left(\sin^2\tfrac{\xi_1}{2}+\sin^2\tfrac{\xi_2}{2}\right)\\
&-6\lambda\left(\cos^2\alpha\sin^2\tfrac{\xi_1}{2}+2\cos\alpha\sin\alpha\sin\tfrac{\xi_1}{2}\cos\tfrac{\xi_1}{2}\sin\tfrac{\xi_2}{2}\cos\tfrac{\xi_2}{2}+\sin^2\alpha\sin^2\tfrac{\xi_2}{2}\right)\\
&c = 2\lambda\left(\cos^2\alpha\sin^2\tfrac{\xi_1}{2}+2\cos\alpha\sin\alpha\sin\tfrac{\xi_1}{2}\cos\tfrac{\xi_1}{2}\sin\tfrac{\xi_2}{2}\cos\tfrac{\xi_2}{2}+\sin^2\alpha\sin^2\tfrac{\xi_2}{2}\right)
\end{aligned}$$

where again $\lambda = \Delta t/\Delta x$.

We want to show that $P(g)$ is a simple Von Neumann polynomial, i.e., both roots of $P(g)$ are not outside the unit circle, and its roots on the unit circle (if there is any) are simple roots. By Theorem 4.3.2 in [29], we only need to show that $|P(0)| < |P^*(0)|$ and

$$P_1(g) = \frac{P^*(0)P(g) - P(0)P^*(g)}{g} \tag{60}$$

is a simple Von Neumann polynomial, where

$$P^*(g) = a + bg + cg^2. \tag{61}$$

It is easy to check $|P(0)| < |P^*(0)|$, i.e., $|c| < |a|$, and

$$P_1(g) = (a^2 - c^2)g + b(a - c). \tag{62}$$

The root of P_1 is $g^* = -b/(a + c)$. Again it is easy to see that $|g^*| \leq 1$. $\diamond$

References

[1] D. Adalsteinsson and J.A. Sethian, A fast level set method for propagating interfaces, *J. Comput. Phys.*, **118**, 269(1995)

[2] D. Adalsteinsson and J.A. Sethian, The fast construction of extension velocities in the level set methods, *J. Comput. Phys.*, **148**, 2(1999).

[3] D. Adalsteinsson and J.A. Sethian, Transport and Diffusion of Material Quantities on Propagating Interfaces via Level Set Methods, *preprint.*

[4] R. Aris, Vectors, Tensors, and the Basic Equations of Fluid, Prentice-Hall, Englewood Cliffs, NJ, 1962.

[5] G.K. Batchlor, An introduction to Fluid Dynamics, Cambridge University Press, 1967

[6] P. Burchard, L.-T. Cheng, B. Merriman and S. Osher, Motion of curves in three spatial dimensions using a level set approach, *J. Comput. Phys.*, **170**, 2, 720-741 (2001).

[7] H. Ceniceros and T.Y. Hou An efficient dynamically adaptive mesh for potentially singular solutions, *J. Comput. Phys.*, **172**, 2, 1-31 (2001).

[8] M. Bertalmio, L.-T. Cheng, S. Osher, and G. Sapiro, Variational problems and partial differential equations on implicit surfaces: the framework and examples in image processing and pattern formation, *J. Comput. Phys.*, **174**, 759-780 (2001).

[9] S. Chen, B. Merriman, S. Osher, and P. Smereka, A simple level set method for solving Stefan problems. *J. Comput. Phys.*, **135**, 8, 1997.

[10] C. Herring, Surface diffusion as a motivation for sintering, *The Physics of Powder Metallurgy*, McGraw-Hill, New York, NY, 1951.

[11] J.K. Hunter, Z. Li, and H.K. Zhao, Reactive autophobic spreading of drops, *to appear in J. Comp. Phys.*

[12] T.Y. Hou, Z. Li, S. Osher, and H. Zhao, A hybrid method for moving interface problems with applications to the Hele-Shaw flows, *J. Comput. Phys.*, **134**, 236(1997)

[13] A. James, J. Lowengrub, and M. Seigel, Distribution formulation of interfaces with surfactant, preprint.

[14] G.-S. Jiang and C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.*, **126**, 202-228(1996)

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

[16] X. Li and C. Pozrikidis, Effect of surfactants on drop deformation and on the rheology of dilute emulsions in Stokes flow, *J. Fluid Mech.*, **341**, 441-454(1997).

[17] Z.-L. Li, H.-K. Zhao, and H.-J., Gao, A numerical study of electro-migration voiding by evolving level set functions on a fixed Cartesian grid, *J. Comput. Phys.*, **152**, 281(1999).

[18] W.W. Mullins, Mass transport at interfaces in single component system, *Metallurgical and Materials Trans. A*, **26**, 1917-1925, (1995).

[19] F.A. Nichols and W.W. Mullins, Surface (interface) and volume-diffusion contributions to morphological changes driven by capillarity, *Trans. Metall. Soc. AIME*, **233**, 1840-1847, (1965).

[20] S. Osher and J.A. Sethian, Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.*, **79**, 12(1988).

[21] S. Osher and C.W. Shu, High order essentially non-oscillatory schemes for Hamilton-Jacobi equations, *SIAM J. Numer. Anal.*, **28**(4), 907(1991).

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

[23] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, and M. Kang, A PDE-based fast local level set method, *J. Comput. Phys.*, **155**, 410(1999).

[24] Y. Saad, Iterative methods for sparse linear systems, PWS, 1996.

[25] L.E. Scriven, *Chem. Eng. Sci.*, **12**, 98(1960).

[26] J.A. Sethian, Evolution, implementation, and application of level set and fast marching methods for advancing fronts, *J. Comput. Phys.*, **169**, 503(2001).

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

[28] H.A. Stone, A simple derivation of the time-dependent convective-diffusion equation for surfactant transport along a deforming interface, *Phys. Fluids A* **2** (1), 111(1989).

[29] J.C. Strikwerda, Finite difference schemes and partial differential equations, Wadsworth & Brooks/Cole Advanced Books & Software, Pacific grove, California, 1989.

[30] M. Sussman, P. Smereka, and S. Osher, A levelset approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.*, **114**, 146-159 (1994)

[31] A.M. Waxman, *Stud. Appl. Math.* , **70**, 63(1984).

[32] H. Wong, D. Rumschitzki, and C. Maldarelli, On the surfactant mass balance at a deforming fluid interface, *Phys. Fluids*, **8**(11), 3203(1996).

[33] H. Zhao, T. Chan, B. Merriman, and S. Osher, A variational level set approach to multiphase motion, *J. Comput. Phys.*, **127**, 179(1996).

Table 1: Errors for surfactant concentration at time $t = 2.0$, with $\Delta t = \Delta x/4$

Grid Size	L_∞	order	L_1	order	L_2	order
40×40	6.20D-3		9.60D-3		5.37D-3	
80×80	1.87D-3	1.73	2.53D-3	1.92	1.49D-3	1.85
160×160	5.48D-4	1.77	6.60D-4	1.94	4.06D-4	1.88

Table 2: Errors for surfactant concentration at time $t = 2.0$ with $\Delta t = \Delta x/4$

Grid Size	L_∞	order	L_1	order	L_2	order
20×15	5.20D-2		1.88D-1		6.91D-2	
40×30	1.58D-2	1.72	8.51D-2	1.14	2.98D-2	1.21
80×60	5.10D-3	1.63	3.32D-2	1.34	1.12D-2	1.41

Table 3: Errors for surfactant concentration at time t=2.0 with $\Delta t = \Delta x/4$

Grid Size	L_∞	order	L_1	order	L_2	order
20×15	5.21D-3		1.94D-2		7.06D-3	
40×30	8.65D-4	2.59	5.07D-3	1.94	1.55D-3	2.19
80×60	1.40D-4	2.62	8.78D-4	2.53	2.68D-4	2.53

Figure Captions:

Figure 1: The moving interface and surfactant concentration at different times

Figure 2: Surfactant concentration as a function of the central angle at different times

Figure 3: Surfactant concentration as a function of the central angle at different times

Figure 4: The moving interface and surfactant concentration at different times

Figure 5: The relative change of the total mass of surfactant in time

Figure 6: The relative change of the area enclosed by the interface in time

Figure 7: The moving interface and surfactant concentration at different times

Figure 8: The relative change of the total mass of surfactant in time

Figure 9: The relative change of the area enclosed by the interface in time

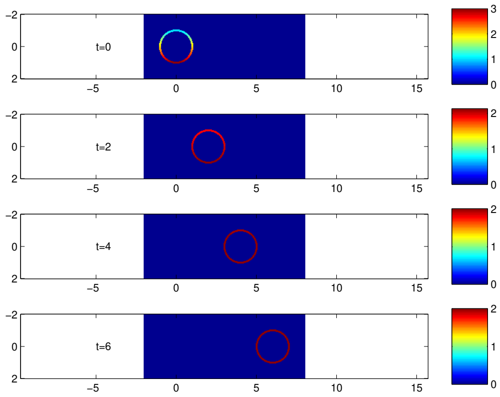

Figure 1:

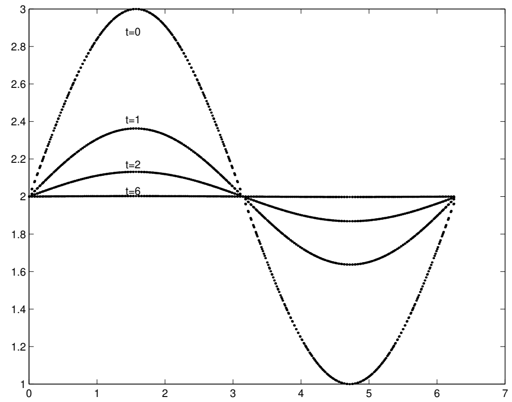

Figure 2:

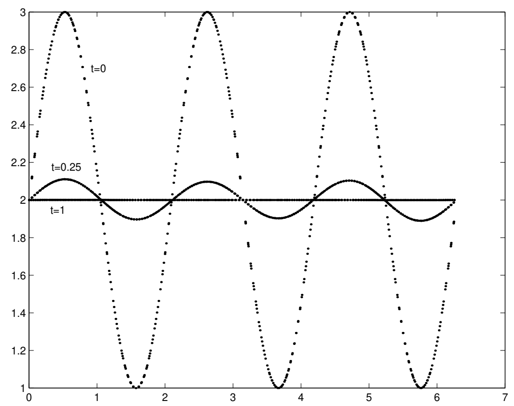

Figure 3:

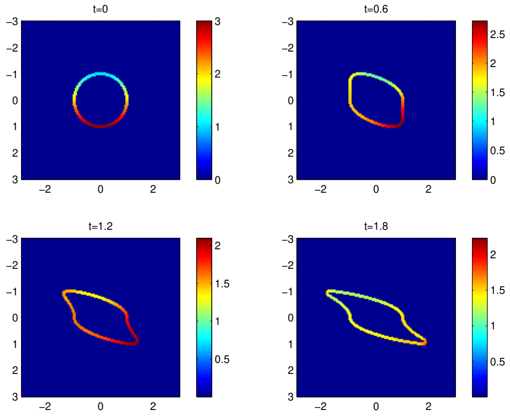

Figure 4:

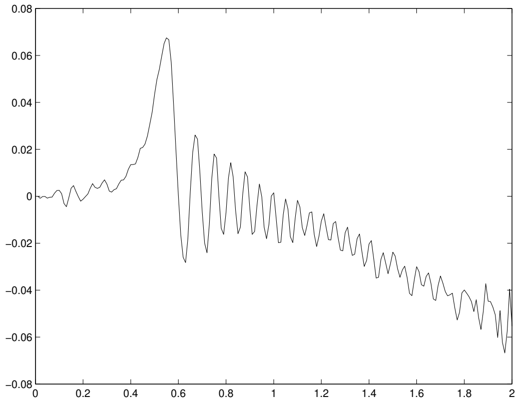

Figure 5:

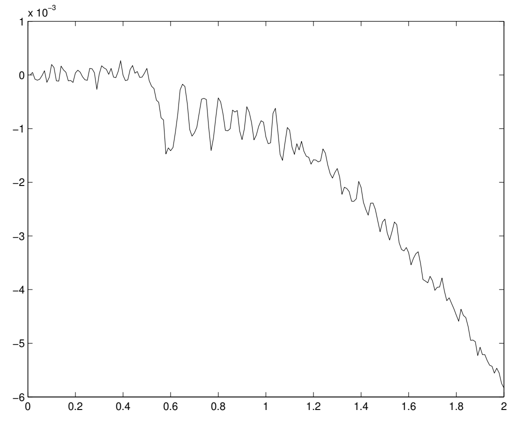

Figure 6:

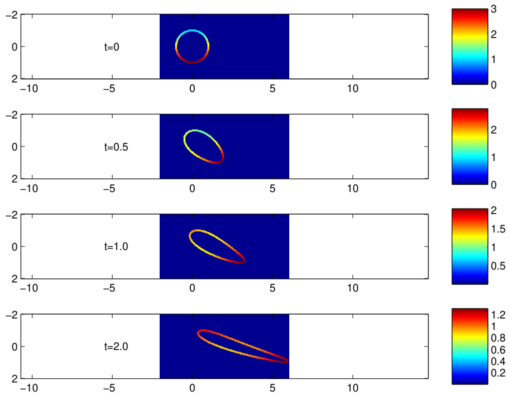

Figure 7:

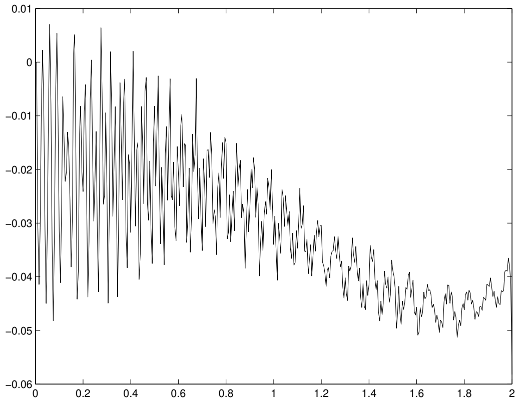

Figure 8:

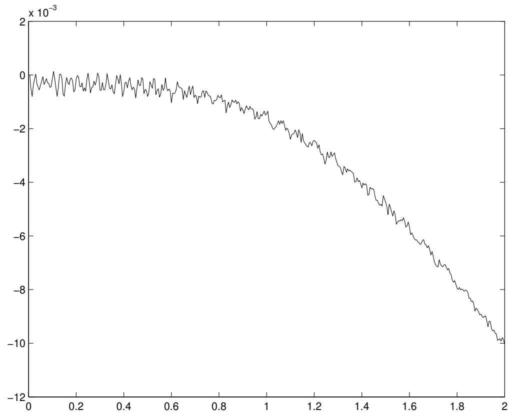

Figure 9: