UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

An Eulerian Formulation for Solving Partial Differential Equations Along a Moving Interface

Jian-Jun Xu
Hong-Kai Zhao

May 2002

CAM Report 02-27

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

http://www.math.ucla.edu/applied/cam/index.html

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 quantity on the interface and evolves according to a PDE on the moving interface. In particular we use a convection diffusion equation for surfactants concentration along a passively convected interface as a model problem. We develop a stable and efficient semi-implicit scheme to remove the stiffness caused by surface diffusion.

1 Introduction

Many applications in fluids, materials and biology involves 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 determines 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 at 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 are surfactants present at the interface between two fluids, the concentration of surfactants is both advected by the ambient flow and diffused along the moving interface. On the other hand, due to the presence of surfactants, the surface tension at the interface depends on both the geometry of the interface and the concentration of surfactants. Hence the motion and geometry of the moving interface, the distribution of surfactants on the interface, and dynamics of the bulk fluids are all coupled together. Mass transport along interfaces

*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.

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 [7, 14, 15, 13].

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 that can have complicated geometry and topology or can even develop singularities. All these complications can pose great challenges for mathematical analysis. In numerical simulations of multiphase problems, the usual difficulties include solving PDEs in each phase coupled with jump/flux conditions and complicated geometry at the interfaces as well as tracking of the moving interface that can develop large deformation and topological changes. Moreover we may need to solve PDEs along the moving interface. 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, which may become too expensive or intractable in three dimensions. Solving PDEs on the moving interface can make Lagrangian formulation even more complicated. In Eulerian formulation the moving interface is captured on a simple Cartesian grid by a level set function. The motion and geometric information of the moving interface can 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 deformation and topological changes can also be handled easily. The level set method, developed by Osher and Sethian [16], has been applied successfully to a broad range of problems in fluids, materials, image processing and computer vision. We refer the readers to the recent review article [18, 22] 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 by Cheng, et al [6], PDEs are solved on a static implicit surface represented by a level set function. 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 surfactants 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. We present numerical examples to illustrate the efficiency of out method. Some numerical studies of the effect of surfactants using Lagrangian type of algorithms for special cases can be found in [12] and references therein.

We point out that although we assume a fluid velocity field is given and the interface is passively convected by the fluid flow, our formulation here can be naturally combined with the immersed interface method (IMM) (see, e.g., [8, 13]) 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.

2 Basic formulations

Suppose Γ is a moving interface advected in a given velocity field $\mathbf{u}(\mathbf{x})$. Denote f to be a scalar quantity, e.g., the concentration of surfactants, defined on Γ and satisfies an evolution PDE on the moving interface. For example, the surfactants 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. Using mass conservation, the transport of surfactants due to the motion and distortion of the moving interface satisfies the following equation [24] in a simple divergence form

$$f_t + \nabla_s \cdot (f\mathbf{u}) = 0, \tag{1}$$

where ∇_s denotes surface gradient. Let $\mathbf{n}$ be the unit normal to the interface, then $\nabla_s = (I - \mathbf{n} \otimes \mathbf{n}) \cdot \nabla$. 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

$$f_t + \nabla_s \cdot (f\mathbf{u}_s) + (\mathbf{u} \cdot \mathbf{n})\kappa f = 0, \tag{2}$$

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. In [9] the following equivalent formulation was derived

$$f_t + \mathbf{u} \cdot \nabla f - \mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n} f = 0, \tag{3}$$

where $\mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n} = \sum_{i,j} n_i \frac{\partial u_i}{\partial x_j} n_j$. Although mathematically these formulas are equivalent, equation (3) fits into our Eulerian framework more naturally as we will see later. To include the surface diffusion, we just have to 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, e.g.,

$$f_t + \mathbf{u} \cdot \nabla f - \mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n} f - \nabla_s^2 f = 0 \tag{4}$$

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.

$$\begin{aligned}
\nabla_s^2 &= (I-\mathbf{n}\otimes\mathbf{n})\nabla\cdot(I-\mathbf{n}\otimes\mathbf{n})\nabla \\
&= [\nabla-\mathbf{n}(\mathbf{n}\cdot\nabla)]\cdot[\nabla-\mathbf{n}(\mathbf{n}\cdot\nabla)] \\
&= \nabla\cdot[\nabla-\mathbf{n}(\mathbf{n}\cdot\nabla)]-(\mathbf{n}\cdot\nabla)\mathbf{n}\cdot[\nabla-\mathbf{n}(\mathbf{n}\cdot\nabla)] \\
&= \nabla^2-\nabla(\nabla\cdot\mathbf{n})\cdot\mathbf{n}-(\nabla\cdot\mathbf{n})(\mathbf{n}\cdot\nabla) \\
&= \nabla^2-\frac{\partial^2}{\partial\mathbf{n}^2}-\kappa\frac{\partial}{\partial\mathbf{n}} \qquad (5)
\end{aligned}$$

If a quantity f is defined in a neighborhood of Γ, we have

$$\nabla_s^2 f=\nabla^2 f-\frac{\partial^2 f}{\partial\mathbf{n}^2}-\kappa\frac{\partial f}{\partial\mathbf{n}}=\nabla^2 f-\mathbf{n}D^2(f)\mathbf{n}-\kappa\nabla f\cdot\mathbf{n} \qquad (6)$$

where ∇^2 is the standard Laplacian operator and D^2 is the Hessian.

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

$$\phi_t+\mathbf{u}\cdot\nabla\phi=0. \qquad (7)$$

To capture the evolution of a quantity f on the moving interface, we introduce another scalar function $\tilde{f}(\mathbf{x})$ in a neighborhood of the interface such that $\tilde{f}(\mathbf{x})=f(\mathbf{x})$ for $\mathbf{x}\in\Gamma$, (i.e., $\phi(\mathbf{x})=0$). 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 $\tilde{f}$ is also 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 example, if the surfactants have a bulk distribution. In other applications, f may be a physical quantity only defined on the interface. Then we can use a numerical extension procedure to extend the quantity off the interface while keeping its values on the interface unchanged. The procedure will be discussed in the next section. From the level set function ϕ, we can compute geometric quantities of the interface easily, e.g.,

$$\mathbf{n}(\mathbf{x})=\frac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|},\quad \kappa(\mathbf{x})=\nabla\cdot\left(\frac{\nabla\phi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}\right) \qquad (8)$$

So the surface gradient operator and surface Laplacian operator in (5) are all well defined and we can solve the corresponding PDE for f such as (4) 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 as all level sets of ϕ are convected in the velocity field $\mathbf{u}$ in (7), instead of solving the PDE for f solely on the zero level set of ϕ, which is the moving interface, we solve the same PDE for $\tilde{f}$ on every level set of ϕ in a neighborhood of the zero level set. 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.

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 [28] for the extension. We will discuss this in detail in the next section. We also combine with the local level set method to reduce computation cost.

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

3 Numerical algorithms

Based on the Eulerian formulation developed in section 2, we can capture a moving interface as well as solving PDE(s) for some quantity f along the moving interface on a simple Cartesian grid. Start with an initial setup that 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), here is the outline of the main procedures in one time step:

step 1: Evolve the quantity f on the moving interface, e.g., by (4).

step 2: Evolve the interface in the velocity field by (7).

step 3: (if needed) Reinitialize the level set function ϕ to 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 scheme in time, where Δx is the spatial grid size. This can be the bottle-neck for the whole numerical computation. For example, in our model problem for a moving interface with surfactants, the interface is simply convected in a velocity field. The corresponding convection equation (7) 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 (5), 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 Laplacain minus the second derivative in the normal direction, i.e., an isotropic diffusion minus diffusion in the normal. 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 [4]. By adding and subtracting an isotropic diffusion term that is large enough to dominate the anisotropic diffusion term and make the isotropic diffusion term with right sign implicit and all other terms explicit, the scheme becomes unconditionally stable. For example, to solve

$$u_t = \nabla \cdot (a(\mathbf{x})\nabla u), \quad \text{where } \; C \geq a(\mathbf{x}) \geq c > 0, \tag{9}$$

which is mathematically the same as

$$u_t = A\nabla^2 u + \nabla \cdot (a(\mathbf{x})\nabla u) - A\nabla^2 u, \tag{10}$$

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

$$\frac{u^{m+1} - u^m}{\Delta t} = A\nabla^2 u^{m+1} + \nabla \cdot (a(\mathbf{x})\nabla u^m) - A\nabla^2 u^m, \tag{11}$$

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. In order to achieve the second order accuracy in time, we use the following second order semi-implicit Crank-Nicholson scheme in time for the convection diffusion equation (4) for f on the interface,

$$\begin{aligned}\frac{f^{m+1} - f^m}{\Delta t} = \; & \frac{\nabla^2 f^{m+1} + \nabla^2 f^m}{2} + \frac{3}{2}\left[-\kappa\frac{\partial f}{\partial \mathbf{n}} - \frac{\partial^2 f}{\partial \mathbf{n}^2} - \mathbf{u}\cdot\nabla f + \mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}f\right]^m \\ & -\frac{1}{2}\left[-\kappa\frac{\partial f}{\partial \mathbf{n}} - \frac{\partial^2 f}{\partial \mathbf{n}^2} - \mathbf{u}\cdot\nabla f + \mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}f\right]^{m-1}\end{aligned} \tag{12}$$

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

$$(u)^+ D_x^- f_{ij} + (u)^- D_x^+ f_{ij} + (v)^+ D_y^- f_{ij} + (v)^- D_y^+ f_{ij}, \tag{13}$$

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 [10, 11]. 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

$$D_x^- g_i = \frac{[(\Delta^+ g_{i-1} + \Delta^+ g_i) - \omega_-(\Delta^+ g_{i-2} - 2\Delta^+ g_{i-1} + \Delta^+ g_i)]}{(2\Delta x)} \tag{14}$$

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

$$r = \frac{\epsilon + (\Delta^- \Delta^+ g_{i-1})^2}{\epsilon + (\Delta^- \Delta^+ g_i)^2} \tag{15}$$

where ϵ is a small positive number. 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

$$D_x^+ g_i = \frac{[(\Delta^+ g_{i-1} + \Delta^+ g_i) + \omega_+(\Delta^+ g_{i+1} - 2\Delta^+ g_i + \Delta^+ g_{i-1})]}{(2\Delta x)} \tag{16}$$

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

$$r = \frac{\epsilon + (\Delta^- \Delta^+ g_{i+1})^2}{\epsilon + (\Delta^- \Delta^+ g_i)^2} \tag{17}$$

The linear system for f^{m+1} from (12) is symmetric positive definite as for the standard heat equation. It can be solved easily by conjugate gradient method or even SOR method [20]. If the quantity f is extended orthogonal to the interface approximately, i.e., $\nabla f \cdot \nabla \phi \approx 0$, then $\dfrac{\partial f}{\partial \mathbf{n}} \approx 0$, $\dfrac{\partial^2 f}{\partial \mathbf{n}^2} \approx 0$. Since our semi-implicit scheme is a two-step method, we use the following one step semi-implicit scheme,

$$\frac{f^1 - f^0}{\Delta t} = \nabla^2 f^1 + \left[-\kappa \frac{\partial f}{\partial \mathbf{n}} - \frac{\partial^2 f}{\partial \mathbf{n}^2} - \mathbf{u} \cdot \nabla f + \mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n} f\right]^0. \tag{18}$$

for the first time step. We will use numerical examples to show 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 (7) 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 [23]. Consider the following time dependent PDE,

$$\frac{dg}{dt} = L(g), \tag{19}$$

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

$$\left\{ \begin{array}{ll} g_1 & = g^m + \Delta t L(g^m) \\ g_2 & = \frac{3}{4} g^m + \frac{1}{4} g_1 + \frac{\Delta t}{4} L(g^1) \\ g^{m+1} & = \frac{3}{4} g^m + \frac{2}{3} g_2 + \frac{3\Delta t}{4} L(g_2). \end{array} \right. \tag{20}$$

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 at every time step. To reinitialize the level set function, we use the PDE based approach which was proposed in [25] and was discussed in detail in [19]. After the evolution of the level set function at step m, we reinitialize it by the equation:

$$\begin{cases} \phi_\tau + S(\phi_0)(|\nabla\phi| - 1) = 0 \\ \phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}) = \phi^m(\mathbf{x}) \end{cases} \tag{21}$$

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

$$S(x) = \begin{cases} -1 & \text{if} \quad x < 0 \\ 0 & \text{if} \quad x = 0 \\ +1 & \text{if} \quad x > 0. \end{cases} \tag{22}$$

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

$$\begin{aligned} & s_{ij}^+ \left(\sqrt{(a^+)^2 + (b^-)^2 + (c^+)^2 + (d^-)^2} - 1 \right) \\ & + s_{ij}^- \left(\sqrt{(a^-)^2 + (b^+)^2 + (c^-)^2 + (d^+)^2} - 1 \right) \end{aligned} \tag{23}$$

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

$$s_{ij} = \frac{\phi_{ij}^0}{\sqrt{(\phi_{ij}^0)^2 + \Delta x^2}}. \tag{24}$$

a, b, c, d in (23) 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 some 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 use the following simple linear convection equation

$$\begin{cases} f_\tau + S(\phi)\mathbf{n} \cdot \nabla f = 0 \\ f(\mathbf{x}, 0) = f_0(\mathbf{x}), \end{cases} \tag{25}$$

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 [28] and later discussed in more detail in [19]. It has been successfully applied to the Stefan problem in [5]. 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 $\mathbf{n} \cdot \nabla f$.

In practice we do not need to solve equations (21) or (25) to steady state. At each time step since the solutions does not change much from the previous ones, which are supposed to satisfy these equations, we only need a few number of iterations or pseudo time steps for the reinitialization or extension. However 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 (20) for (21) and (25) 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. We present the method here in two dimensional formulation, the extension to three dimensions is straight forward.

Local level set method For the level set method, instead of solving the PDE for the level set function, e.g., (7), 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., (4), in a neighborhood of the zero level set. In our computations we adopt the PDE based local level set method discussed in [19]. 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,

$$\begin{aligned} T_1 &= \{(x_i, y_j) : |\phi(x_i, y_j)| < \gamma_1\} \\ T_2 &= \{(x_i, y_j) : |\phi(x_i, y_j)| < \gamma_2\} \\ T_3 &= \{(x_i, y_j) : \min_{l,k=-1,0,1} |\phi(x_{i+l}, y_{j+k})| < \gamma_3\} \end{aligned}$$

The widths, γ_i's, and their differences are usually a few multiple of grid size. The choice of parameters $\gamma_1, \gamma_2, \gamma_3$ depends 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 $\gamma_1 = 3\Delta x, \gamma_2 = \gamma_1 + 3\Delta x$, $\gamma_3 = \gamma_1 + 3\Delta x$ 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 (18) 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 (4) for f using the semi-implicit scheme (12) to get f^{m+1} in tube T_1. The spatial discretization in (12) involves not only points in tube $T1$ 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 interpolate the boundary condition for f^{m+1} as follows:

$$f_{ij}^{m+1} = 2f_{ij}^{m} - f_{ij}^{m-1}, \forall (x_i, y_j) \in T_2 - T_1. \tag{26}$$

step 2: We introduce the following cutoff function $c_m(\phi)$ used in [19]

$$c_m(\phi) = \begin{cases} 0 & \text{if} \quad \phi^m > \gamma_2 \\ \frac{(|\phi^m| - \gamma_2)^2 (2|\phi^m| + \gamma_2 - 3\gamma_1)}{(\gamma_2 - \gamma_1)^3} & \text{if} \quad \gamma_1 < |\phi^m| \le \gamma_2 \\ 1 & \text{if} \quad |\phi^m| \le \gamma_1 \end{cases} \tag{27}$$

and solve

$$\phi_t + c_m \mathbf{u} \cdot \nabla \phi = 0 \tag{28}$$

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 (21) in tube T_3 and then extend f using (25) in tube T_2. In both equations information propagates away from the interface, and hence we do not need explicit boundary conditions using the upwind scheme.

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

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

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 surfactants on the interface is conserved due to the divergence form of the equation

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

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 [28]:

$$\int_\Gamma g(\mathbf{x})ds = \int g(\mathbf{x})\delta(\phi)|\nabla\phi|d\mathbf{x}. \tag{29}$$

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:

$$\int_\Omega g(\mathbf{x})d\mathbf{x} = \int g(\mathbf{x})H(-\phi)d\mathbf{x}. \tag{30}$$

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,

$$\begin{cases} \dfrac{\partial f(\theta,t)}{\partial t} = \dfrac{1}{r_0^2}\dfrac{\partial^2 f(\theta,t)}{\partial\theta^2} \quad 0 \le \theta \le 2\pi \\ f(\theta,0) = f_0(\theta) \\ f(0,t) = f(2\pi,t) \end{cases}, \tag{31}$$

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

$$f(\theta, t) = e^{-\frac{n^2 t}{r_0^2}} \sin(n\theta) + c \tag{32}$$

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

$$f(x, y, t) = e^{-\frac{n^2 t}{r^2}} \sin(n\theta) + c, \tag{33}$$

is a global solution to the PDE

$$\frac{\partial f(x,y,t)}{\partial t} = \nabla_s^2 f(x,y,t) = \nabla^2 f(x,y,t) - \frac{\partial^2 f(x,y,t)}{\partial \mathbf{n}^2(x,y)} - \kappa(x,y)\frac{\partial f(x,y,t)}{\partial \mathbf{n}(x,y)}, \tag{34}$$

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 (34) in the domain $[-2, 2] \times [-2, 2] - \{(x, y) : \sqrt{x^2 + y^2} < 0.8\}$ using the semi-implicit scheme (12) 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 $\Delta t = O(\Delta x)$.

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

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 (7) and (3) 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 surfactants 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.

$$f(x,y,t) = e^{-\frac{t}{4}} \sin(\theta(t)) + 2, \tag{35}$$

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:

$$\begin{aligned}
||e_h||_{L_\infty} &= max_{|\phi_h|<1.5\Delta x}\{e_h\} && (36)\\
||e_h||_{L_1} &= \int_\Gamma e_h ds && (37)\\
||e_h||_{L_2} &= \left(\int_\Gamma e_h^2 ds\right)^{\frac{1}{2}} && (38)
\end{aligned}$$

The surface integrals in $||\cdot||_{L_1}$ and $||\cdot||_{L_2}$ are computed according to (29). The errors and order of accuracy at time $t = 2$ is shown in tables 2.

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

Remark 1: There are a few issues need 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 (21) and (25). The other one is because we may not reach the steady state solution of (21) and (25) for just a few iterations in our computation, i.e., f may not be constant in the normal direction. So (35) 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,

$$g(x,y,t) = -\frac{1}{4} e^{-\frac{t}{4}} \frac{y((x-t)^2+y^2-4)}{((x-t)^2+y^2)^{3/2}}, \tag{39}$$

in the equation for surfactant concentration

$$f_t = \nabla_s^2 f - \mathbf{u} \cdot \nabla f + f \mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{n} + g, \tag{40}$$

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.,

$$f(x, y, t) = e^{-\frac{t}{4}} sin(\theta(t)) + 2 = e^{-\frac{t}{4}} \frac{y}{\sqrt{(x-t)^2 + y^2}} + 2 \tag{41}$$

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.

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

Remark 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,

$$\mathbf{u}(x, y) = \begin{cases} (y^2, 0) & \text{if } \ y \geq 0 \\ (-y^2, 0) & \text{if } \ y < 0 \end{cases} \tag{42}$$

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

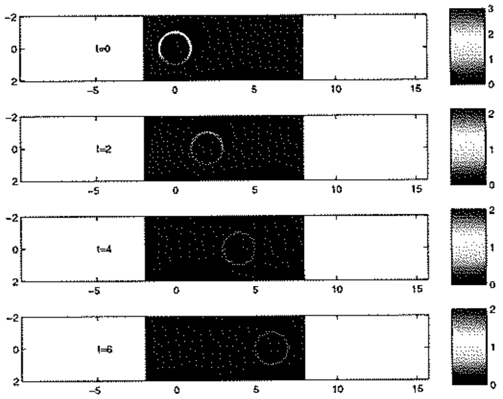

Figure 1: The moving interface and surfactant concentration at different times

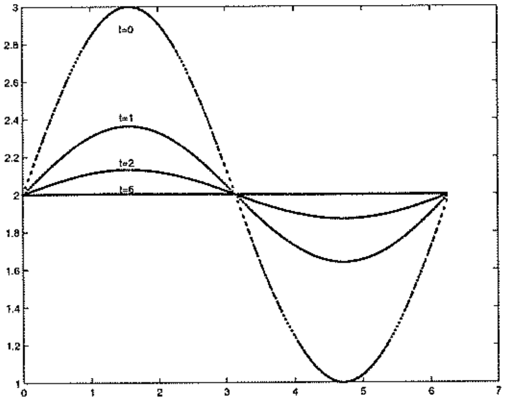

Figure 2: Surfactant concentration as a function of the central angle at different times

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 surfactants 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

$$\mathbf{u}(x,y) = \left(\frac{(y+2)^2}{3}, 0\right). \tag{43}$$

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 surfactants on the interface and the area enclosed by the interface respectively.

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

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

[5] S. Chen, B. Berriman, S. Osher, and P. Smereka, A simple level set method for solving Stefan problems, *J. Comput. Phys.*, **135**, 8 (1995).

[6] 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).

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

[8] 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)

[9] A. James, J. Lowengrub and M. Seigel, Distribution Formulation of Interfaces with Surfactants, preprint.

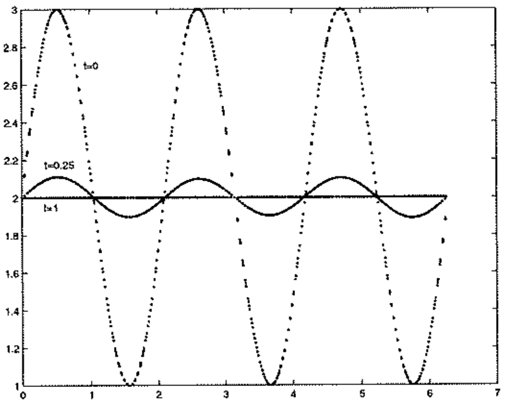

Figure 3: Surfactant concentration as a function of the central angle at different times

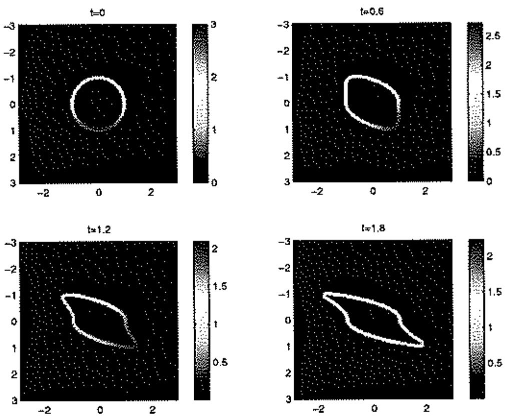

Figure 4: The moving interface and surfactant concentration at different times

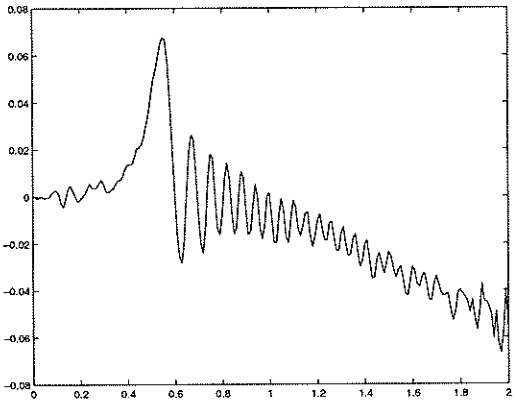

Figure 5: The relative change of the total mass of surfactants in time

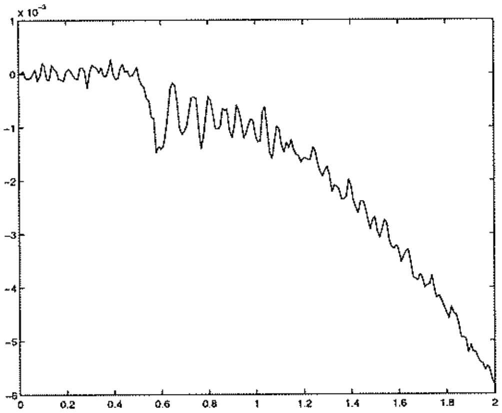

Figure 6: The relative change of the area enclosed by the interface in time

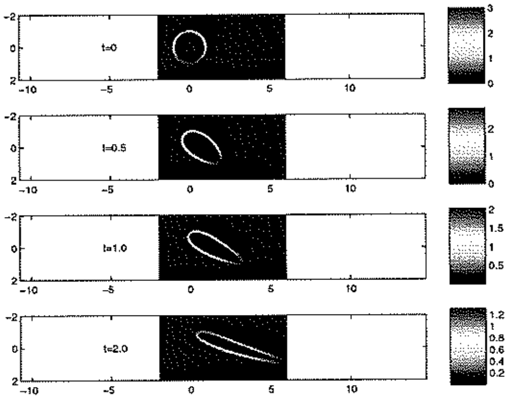

Figure 7: The moving interface and surfactant concentration at different times

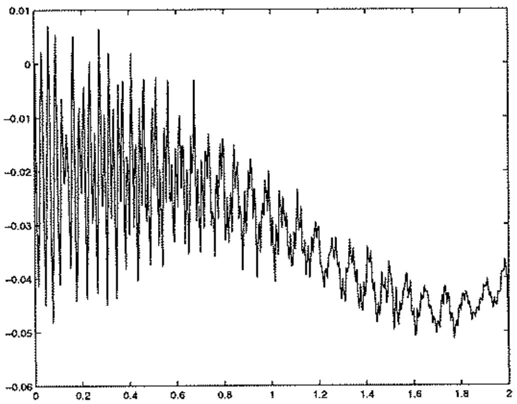

Figure 8: The relative change of the total mass of surfactants in time

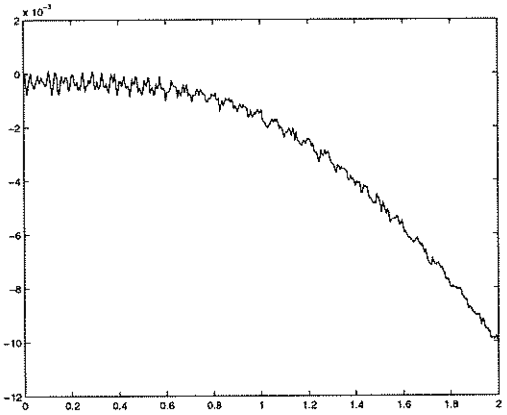

Figure 9: The relative change of the area enclosed by the interface in time

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

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

[12] 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(1995).

[13] 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).

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

[15] 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).

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

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

[18] S. Osher and R.P. Fedkiw, Level Set Methods: An Overview and Some recent Results, *J. Comput. Phys.*, **169**, 463(2001).

[19] 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).

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

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

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

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

[24] 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).

[25] 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)

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

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

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