Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **9**, No. 6, pp. 1364-1382

DOI: 10.4208/aamm.OA-2016-0052
December 2017

An Adaptive Semi-Lagrangian Level-Set Method for Convection-Diffusion Equations on Evolving Interfaces

Weidong Shi[1,2], Jianjun Xu[2,*] and Shi Shu[1]

[1] *School of Mathematical and Computational Sciences, Xiangtan University, Xiangtan, Hunan 411105, China*
[2] *Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China*

Received 4 July 2016; Accepted (in revised version) 23 December 2016

Abstract. A new Semi-Lagrangian scheme is proposed to discretize the surface convection-diffusion equation. The other involved equations including the the level-set convection equation, the re-initialization equation and the extension equation are also solved by S-L schemes. The S-L method removes both the CFL condition and the stiffness caused by the surface Laplacian, allowing larger time step than the Eulerian method. The method is extended to the block-structured adaptive mesh. Numerical examples are given to demonstrate the efficiency of the S-L method.

AMS subject classifications: 65L50, 65M06, 65M20

Key words: Convection-diffusion equation, semi-Lagrangian method, level-set method, block-structured adaptive mesh, finite difference method.

1 Introduction

Solving PDEs on evolving interfaces has attracted a lot of attention in recent years (e.g., [6, 9, 11, 16, 17, 23, 26]). One of the important applications is the surfactant, which plays a critical role in numerous important industrial and biological applications (e.g., [28,30,32] and the references therein). In a fluid flow, surfactant is transported along the interface by advection via the tangential interface velocity as well by diffusion within the interface.

In [26], an Eulerian level-set method for convection-diffusion equations on evolving interfaces. The Eulerian method was applied in simulating interfacial/two-phase flows with insoluble surfactant in [28–31], two-phase flows with insoluble surfactant and moving contact line in [32], two-phase flows with insoluble surfactant under electric fields in [24], and the surface phase separation in an interfacial flow in [12]. However, all these

*Corresponding author.
Email: weidongshi123@xtu.edu.cn (W. D. Shi), xujianjun@cigit.ac.cn (J. J. Xu), shushi@xtu.edu.cn (S. Shu)

 ©2017 Global Science Press

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

works were done on uniform mesh. For practical problems, often some part of the computational domain needs higher resolution than the others. Uniform mesh refinement can be prohibitively expensive in both computer storage and computation time. Adaptive mesh refinement stragety has been an efficient way to overcome this difficulty by putting more grid points only in the under-resolved region.

It is well-known that the Eulerian level-set methods have the CFL stability constrain. On the other hand, The semi-Lagrangian (S-L) method of [5] is an alternative for problems with convection. In a S-L method, ordinary differential equations for the characteristic curves are solved backward to locate the departure points, then variable approximations at the grid points are obtained by appropriate interpolations. Thus the S-L method releases the CFL stability constraint, allowing larger time steps than its Eulerian counterpart. This feature together with the interpolation makes the S-L method convenient for adaptive mesh refinement.

In this work, we propose an adaptive finite difference level-set method for solving convection-diffusion equations on evolving surfaces. In particular, a new S-L scheme is proposed for solving the extended convection-diffusion equation. The level-set convection equation and the re-initialization are also solved by using S-L schemes. In the adaptive method, we use the block-structured adaptive mesh (BSAM) of [1–3]. The advantages of BSAM include relatively easy mesh generation and parallelization. BSAM is an important component of well-known software packages such as Chombo of [4], Bearclaw of [15] and Paramesh of [13].

The paper is organized as follows. The governing equations are given in Section 2. The S-L schemes on uniform mesh are presented in Section 3. The adaptive mesh method is described in Section 1. Numerical examples in 2D and 3D are shown Section 4. Conclusion is given in Section 5.

2 Problem description

The evolving surface $\Gamma(t)$ is represented by the zero level-set of a level set function $\phi(\mathbf{x},t)$. In this work, it is passively convected according to a given velocity $\mathbf{u}$. The level-set Hamilton-Jacobi (H-J) equation (see [18]) is following:

$$\frac{\partial \phi}{\partial t}+\mathbf{u}\cdot\nabla\phi=0. \tag{2.1}$$

The dynamics of surfactant concentration f is governed by the convection-diffusion equation:

$$\dot{f}+f(\nabla_s\cdot\mathbf{u})=D_s\nabla_s^2 f \quad \text{on } \Gamma, \tag{2.2}$$

where $\dot{f}=\frac{\partial f}{\partial t}+\nabla_s f\cdot\mathbf{u}$ is the material time derivative, $\nabla_s=(I-\mathbf{n}\otimes\mathbf{n})\nabla$ is the surface gradient, and D_s is the material diffusivity. The left-hand side of the equation corresponds to the convection of the material by the velocity field, which can be derived from the

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

following transport relation (e.g., [10]):

$$\frac{d}{dt}\int_S f ds = \int_S (\dot{f} + f\nabla_s \cdot \mathbf{u}) ds, \tag{2.3}$$

where S is an arbitrary material subsurface of the fluid interface.

Without the loss of the generality we assume that the surface diffusive coefficient $D_s = 1$.

In a Cartesian grid method, the surface quantity f is extended into a small neighborhood of the fluid interface, so is the Eq. (2.2). The extension is done by solving the following H-J equation (see [34]):

$$\begin{cases} f_\tau + S(\phi)\mathbf{n}\cdot\nabla f = 0, \\ f(\mathbf{x},0) = f_0(\mathbf{x}), \end{cases} \tag{2.4}$$

where τ is the pseudo-time, f_0 is the surfactant concentration before extension. S is the sign function defined as

$$S(x) = \begin{cases} -1, & x < 0, \\ 0, & x = 0, \\ 1, & x > 0. \end{cases} \tag{2.5}$$

Assume that $\mathbf{u}$ is solenoidal (i.e., $\nabla\cdot\mathbf{u} = 0$), which is true for incompressible flows, then Eq. (2.2) can be rewritten as (e.g., [26])

$$\frac{\partial f}{\partial t} + \mathbf{u}\cdot\nabla f - \mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n} f = \nabla^2 f - \mathbf{n}\cdot D^2 f\cdot\mathbf{n} - \kappa\mathbf{n}\cdot\nabla f, \tag{2.6}$$

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

In practice, the level set function ϕ needs to be re-initialized. This is done by solving the following H-J equation originally proposed in [21]:

$$\begin{cases} \dfrac{\partial\phi}{\partial\tau} + S(\phi_0)(|\nabla\phi| - 1) = 0, \\ \phi(\mathbf{x},0) = \phi_0(\mathbf{x}), \end{cases} \tag{2.7}$$

where ϕ_0 is the level-set function before the re-initialization.

Let $\{\mathbf{x}:\phi(\mathbf{x},t) < 0\}$ be the region enclosed by the interface, the outward normal vector and total curvature are

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \quad \kappa = \nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right). \tag{2.8}$$

In summary, the mathematical problem consists of the level-set convection equation (2.1), the re-initialization equation (2.7), surfactant transport equation (2.6), and the extension equation (2.4).

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

3 Numerical methods

We first describe the method on uniform mesh, then we extend it to the adaptive mesh.

3.1 Uniform mesh method

For simplicity, we describe the algorithms in 2D. Roughly speaking, the 3D algorithms can be obtained by adding one more dimension.

Lay down a uniform Cartesian grid for the computational domain $\Omega=[a,b]\times[c,d]$, with grid length h. Let Δt be the time step. We use the cell-centered approximations for the variables. Denote $x_i=a+(i+0.5)h$, $y_j=c+(j+0.5)h$, $i=0,1,\cdots,M-1$, $j=0,1,\cdots,N-1$, $t^n=n\Delta t$, $n=0,1,\cdots$.

3.1.1 S-L scheme for the convection-diffusion equation on interface

Denote

$$A(f)\equiv\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}f-\mathbf{n}\cdot D^2f\cdot\mathbf{n}-\kappa\nabla f\cdot\mathbf{n}.$$

Along the characteristic curve $\frac{d\mathbf{x}}{dt}=\mathbf{u}$, Eq. (2.6) can be rewritten as

$$\frac{df}{dt}=\nabla^2f+A(f). \tag{3.1}$$

Integrating (3.1) over the time interval $[t^n,t^{n+1}]$, and approximating the right hand side by the trapezoidal rule, we get

$$f_{ij}^{n+1}-f(\mathbf{x}_d,t^n)=\frac{\Delta t}{2}((\nabla^2f+A)_{ij}^{n+1}+(\nabla^2f+A)^n(\mathbf{x}_d)), \tag{3.2}$$

where the departure point $\mathbf{x}_d$ is calculated by the second-order RK method:

$$\mathbf{x}^*=\mathbf{x}_{ij}-\frac{\Delta t}{2}\mathbf{u}_{ij}^n, \tag{3.3a}$$

$$\mathbf{x}_d=\mathbf{x}_{ij}-\Delta t\mathbf{u}^{n+\frac{1}{2}}(\mathbf{x}^*). \tag{3.3b}$$

When the velocity is time dependent, $\mathbf{u}(\mathbf{x}^*)^{n+\frac{1}{2}}$ is calculated by standard second-order accurate extrapolation:

$$\mathbf{u}^{n+\frac{1}{2}}(\mathbf{x}^*)=\frac{3}{2}\mathbf{u}^n(\mathbf{x}^*)-\frac{1}{2}\mathbf{u}^{n-1}(\mathbf{x}^*). \tag{3.4}$$

Since $\mathbf{x}^*$ is not a grid point, the third-order ENO scheme (see [19]) is used to interpolate $\mathbf{u}(x^*)$. The reason to use ENO interpolation is due to the possible non-smoothness of the velocity field in practice (e.g., in two-phase flows). $\phi(\mathbf{x}_d)$ is also calculated by the ENO scheme.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

The complete semi-Lagrangian scheme can be obtained by using a second-order time extrapolation for the term $A(f)_{ij}^{n+1}$ as following:

$$f_{ij}^{n+1}-f(\mathbf{x}_d,t^n)=\frac{\Delta t}{2}(\nabla_h^2 f)_{ij}^{n+1}+\frac{\Delta t}{2}[(2(A_h)_{ij}^n-(A_h)_{ij}^{n-1})+(\nabla_h^2 f+A_h)^n(\mathbf{x}_d)], \tag{3.5}$$

where the quantity at $(\mathbf{x}_d,t^n)$ is calculated by the ordinary third-order Lagrangian interpolation. $\nabla_h^2 f, A_h$ denote the discrete forms using standard second-order central difference schemes to calculate involved quantities $\nabla^2 f$, ∇f, $\mathbf{n}$, κ.

We will demonstrate through numerical experiments that the global S-L method described above is convergent in both 2D and 3D. We will also show that the S-L method allows larger time step for stability than the Eulerian method in [26, 27]. This is an advantage of the S-L method over the Eulerian method.

3.1.2 S-L scheme for the level-set convection equation

The S-L scheme for level-set equation (2.1) is simply

$$\phi_{ij}^{n+1}=\phi^n(\mathbf{x}_d), \tag{3.6}$$

where the departure point $\mathbf{x}_d$ is calculated by the second-order RK method, which results in a second-order S-L method. This second-order S-L method has been used for solving the level-set convection equation (2.1) in the literature, see e.g., [14, 20, 25].

3.1.3 S-L scheme for the level-set re-initialization

In [20], the computational geometry based method on quadtrees was used for the re-initialization. In [14], the classical Eulerian method was used for the re-initialization. In [25], a marching tetrahedra based S-L scheme was proposed. Here we adopt the method in [33], which are briefly described below. Eq. (2.7) can be rewritten as

$$\frac{\partial\phi}{\partial\tau}+S(\phi_0)\mathbf{n}\cdot\nabla\phi=S(\phi_0). \tag{3.7}$$

The velocity field $\mathbf{v}=S(\phi_0)\mathbf{n}$ has jump discontinuity across the interface. Caution needs to be taken for the grid points just adjacent to the interface (i.e., the so-called irregular grid points).

The set of irregular grid points is identified as

$$\Gamma=\{\mathbf{x}_{ij}:\beta_1\beta_2\leq 0\}, \tag{3.8}$$

where

$$\beta_1=\max\{\phi_{i+1,j},\phi_{i-1,j},\phi_{ij},\phi_{i,j+1},\phi_{i,j-1}\},$$

and

$$\beta_2=\min\{\phi_{i+1,j},\phi_{i-1,j},\phi_{ij},\phi_{i,j+1},\phi_{i,j-1}\}.$$

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

The other grid points are regular.

If $\mathbf{x}_{ij}$ is regular, the following first-order S-L scheme is used:

$$\phi_{ij}^{n+1}=\phi^n(\mathbf{x}_d)+S(\phi_0(\mathbf{x}_{ij}))\Delta\tau, \tag{3.9}$$

where the departure point $\mathbf{x}_d$ is obtained by the first-order Euler method: $\mathbf{x}_d=\mathbf{x}_{ij}-S(\phi_0(\mathbf{x}_{ij}))\mathbf{n}_{ij}\Delta\tau$. Here $\Delta\tau$ is the pseudo-time step. $\phi(\mathbf{x}_d)$ is calculated by the bilinear interpolation.

Next let $\mathbf{x}_{ij}$ be irregular. First, we find its projection point $\tilde{\mathbf{x}}=\mathbf{x}_{ij}+\alpha\mathbf{n}_{ij}$ on the interface is obtained by solving the following quadratic equation for α:

$$0=\phi_0(\mathbf{x}_{ij}+\alpha\mathbf{n}_{ij})\approx\phi_0(\mathbf{x}_{ij})+\alpha|\nabla\phi_0(\mathbf{x}_{ij})|+\frac{\alpha^2}{2}\mathbf{n}_{ij}^T\mathrm{He}(\phi_0(\mathbf{x}_{ij}))\mathbf{n}_{ij}, \tag{3.10}$$

where $\mathrm{He}(\phi_0)$ is the Hessian matrix:

$$\mathrm{He}(\phi_0)=\begin{pmatrix}\dfrac{\partial^2\phi_0}{\partial x^2} & \dfrac{\partial^2\phi_0}{\partial x\partial y}\\ \dfrac{\partial^2\phi_0}{\partial y\partial x} & \dfrac{\partial^2\phi_0}{\partial y^2}\end{pmatrix}.$$

When the quadratic equation (3.10) has two real roots, we choose the one with smaller absolute value as α since $\mathbf{x}_{ij}$ is close to interface already. In the case there is no real root, we choose a grid point adjacent to $\mathbf{x}_{ij}$, say, $\mathbf{x}_{i+1,j}$, such that $\phi_0(\mathbf{x}_{ij})\phi_0(\mathbf{x}_{i+1,j})\le 0$. Then ϕ is linearly approximated by setting the projection point $\tilde{\mathbf{x}}=\mathbf{x}_{ij}-\phi_0(\mathbf{x}_{ij})(\mathbf{x}_{i+1,j}-\mathbf{x}_{ij})/(\phi_0(\mathbf{x}_{i+1,j})-\phi_0(\mathbf{x}_{ij}))$. Then $\tilde{\mathbf{x}}$ is used to mimic the signed distance function at the irregular grid point $\mathbf{x}_{ij}$ as follows. Since the level-set function is only Lipschitz continuous, in order to prevent fake projection point $\tilde{\mathbf{x}}$ due to the non-smoothness of ϕ_0, we set

$$\phi_{ij}=\begin{cases}\max\{\phi_0(\mathbf{x}_{ij}),-|\mathbf{x}_{ij}-\tilde{\mathbf{x}}|\}, & \text{if } \phi_0(\mathbf{x}_{ij})<0,\\ \min\{\phi_0(\mathbf{x}_{ij}),|\mathbf{x}_{ij}-\tilde{\mathbf{x}}|\}, & \text{if } \phi_0(\mathbf{x}_{ij})>0.\end{cases} \tag{3.11}$$

In evaluating the normal $\mathbf{n}_{ij}=\nabla\phi/|\nabla\phi|(\mathbf{x}_{ij})$, the modified difference scheme of [7] may be used. For example, $\frac{\partial\phi}{\partial x}(\mathbf{x}_{ij})$ is approximated by standard center difference scheme $(\phi_{i+1,j}-\phi_{i-1,j})/(2h)$, if $\phi_{i+1,j}-\phi_{i,j}$ and $\phi_{i,j}-\phi_{i-1,j}$ have the same sign; by maxmod$\{(\phi_{i+1,j}-\phi_{i,j})/h,(\phi_{i,j}-\phi_{i-1,j})/h\}$ otherwise, where

$$\mathrm{maxmod}(a,b)=\begin{cases}a, & \text{if } |a|>|b|,\\ b, & \text{otherwise,}\end{cases}$$

and similarly for the evaluation of $\frac{\partial\phi}{\partial y}(\mathbf{x}_{ij})$. This modification gives a more accurate normal direction of the interface in the unresolved region around the interface, e.g., near the places where the interfaces are about to have topological changes.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

3.1.4 S-L scheme for the extension equation

The velocity of the extension equation (2.4) is $\mathbf{v}=S(\phi_0)\mathbf{n}$, the same as of the re-initialization.

The S-L scheme for the extension is following:

$$f_{ij}^{n+1}=\begin{cases} f^n(\mathbf{x}_d), & \text{if } \mathbf{x}_{ij} \text{ is regular,} \\ f_{ij}^n, & \text{if } bx_{ij} \text{ is irregular,} \end{cases} \tag{3.12}$$

where the departure point $\mathbf{x}_d=\mathbf{x}_{ij}-\mathbf{v}_{ij}^n\Delta\tau$, here $\Delta\tau$ is the pseudo-time step. The bilinear interpolation is used for calculating $f^n(\mathbf{x}_d)$.

3.2 Adaptive mesh method

In this section, we extend the above described method to the BSAM. The adaptive mesh of domain Ω is a hierarchy of nested rectangular mesh levels, indexed by $k=0,\cdots,K$ in the increasing order of mesh fineness. Let Ω_k be the domain covered by the k-level mesh, $R_{i,k}$ be the open region covered by the ith patch $G_{i,k}$ of the k-level mesh, h_k be the grid length of the k-level mesh. An illustration is given in Fig. 1.

The adaptive mesh satisfies the following three rules:

- $\Omega_0=\Omega$.
- $\Omega_k\subset\Omega_{k-1}$, $R_{i,k}\cap R_{j,k}=\emptyset$, $i\neq j$, $k=1,\cdots,K$,$i,j=1,.\cdots,n_k$, where n_k is the total number of patches of the k-level mesh.
- $h_k=\frac{h_{k-1}}{2}$, $k=1,\cdots,K$.

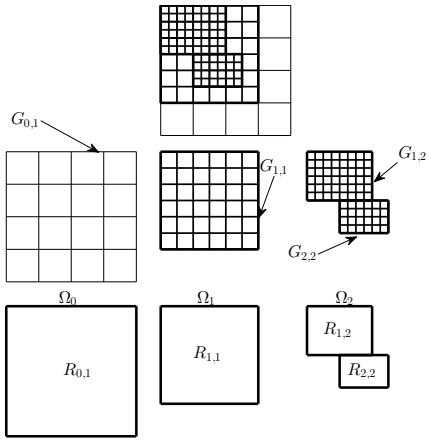


Figure 1: An example of the adaptive mesh of 3 levels. Top: the global composite mesh; Middle: the k-level mesh patches, $k=0,1,2$; Bottom: domains covered at different levels.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

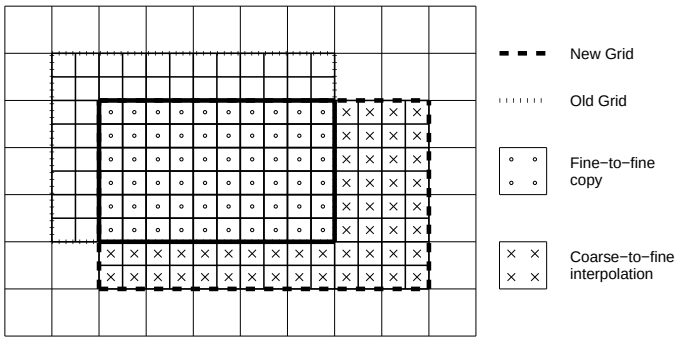


Figure 2: Illustration of data transfer from old mesh to new mesh.

For simplicity the same time step is used on every level mesh, though this may be generalized through a "subcycling" time stepping procedure in [2].

Now starting from a multilevel adaptive mesh on which a numerical solution is given, we mark grid cells where refinement is needed according to certain criterion. In this work we refine mesh at the cells close to the interface:

$$|\phi| \leq w, \tag{3.13}$$

where ϕ is the level-set function, w is a fixed small number. The point clustering algorithm in [3] is used to regroup the marked cells into rectangular mesh patches. The clustering algorithm keeps a good balance between the number of patches and the unnecessary region covered by the patches. The mesh refinement is done on every patch. This procedure is repeated on every level of the old mesh. Finally the data on the old mesh are transferred to the newly generated mesh via copying restriction or bilinear interpolation, as illustrated in Fig. 2.

Note that in the adaptive method of [22], the level-set convection equation is solved on all level meshes. We found that it is only needed to be solved on the finest level mesh. In particular, we do the following steps to evolve the surface convection-diffusion problem:

Step 1 Solve the level-set convection equation (2.1) only on K-level mesh, i.e., the finest mesh.

Step 2 Solve the re-initialization equation (2.7) on every level mesh.

Step 3 Solve the extension equation (2.4) only on K-level mesh.

Step 4 Solve the surface convection-diffusion equation (2.6) only on K-level mesh.

Step 5 Generate new adaptive mesh. Go to Step 1.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

Below we give more explanations on Step 2. Suppose that Eq. (2.1) is solved on K-level mesh in a tube T_K with width $5h_K$, i.e., $T_K=\{\mathbf{x}_{ij}^K:|\phi(\mathbf{x}_{ij}^K)|\leq 5h_K\}$, where $\mathbf{x}_{ij}^K$ represents the cell center of K-level mesh. The re-initialization is done in a larger tube with width $10h_K$. Then the level-set function is restricted to the $(K-1)$-level mesh by averaging in a tube roughly with width $5h_{K-1}$, since $h_{K-1}=\frac{h_K}{2}$. Then the re-initialization is done on the $(K-1)$-level mesh in a tube with width $10h_{K-1}$. Keep going until the 0-level mesh. In this way, the level-set function is defined on every k-level mesh in a tube of width $10h_k$. Thus new adaptive mesh can be generated.

4 Numerical examples

In this section, we demonstrate the efficiency of the S-L level-set method by several numerical examples in 2D and 3D, including comparison among the uniform mesh S-L (UMSL) method, the classical Eulerian (UMCE) method in [26,27], and the adaptive mesh S-L (AMSL) method. The local level-set technique is used, i.e., all the equations are solved in small tubes containing the interface. The tube widths are just a few multiple of grid lengths. For example, the widths are chosen as $5h$, $8h$, $5h$ and $11h$ for the surfactant transport, the extension, the level-set convection, and the re-initialization, respectively. Note that though the curvature is constant in Examples 4.1 and 4.2, we still use standard central difference scheme to calculate it.

Example 4.1. We check the accuracy in 2D on a uniform grid. We choose a simple velocity field $\mathbf{u}=(1,0)^T$ so that the exact solution is available. The initial level set function $\phi(x,y,0)=\sqrt{x^2+y^2}-2$ and the initial distribution of surfactant

$$f(x,y,0)=\sin(\theta)+2=\frac{y}{\sqrt{x^2+y^2}}+2.$$

$\Omega=[-3,5]\times[-3,3]$. The exact solution is

$$f(x,y,t)=e^{-\frac{t}{4}}\sin(\theta(t))+2, \tag{4.1}$$

where

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

Let f_h, ϕ_h be numerical solutions of the surfactant concentration f and the level-set function ϕ respectively. We measure the error $e_h=f-f_h$ in a small neighborhood of the interface, i.e.,

$$\|e_h\|_\infty=\max_{|\phi_h|<1.5h}\{e_h\}. \tag{4.2}$$

The errors and the orders of accuracy at time $t=2$ are given in Table 1.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

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

h	$\|e_h\|_\infty$	order
0.2	3.22×10^{-2}	-
0.1	9.60×10^{-3}	1.75
0.05	2.51×10^{-3}	1.93

Table 2: Errors for surfactant concentration at time $t=1.0$, $\Delta t=h/4$, h is the grid length of the finest mesh.

h	$\|e_h\|_\infty$	order
0.1	2.36×10^{-3}	-
0.05	7.27×10^{-4}	1.70
0.025	2.17×10^{-4}	1.73

Example 4.2. We check accuracy in 3D using the adaptive mesh. Velocity field $\mathbf{u}=(1,0,0)^T$. $\Omega=[-4,4]\times[-2,2]\times[-2,2]$. We introduce a source term

$$g=-\frac{1}{2}e^{-\frac{t}{2}}\frac{z((x-t)^2+y^2+z^2-4)}{((x-t)^2+y^2+z^2)^{3/2}} \tag{4.3}$$

in the surface convection-diffusion equation (2.6), i.e., we will solve the following equation:

$$f_t+\mathbf{u}\cdot\nabla f-\mathbf{n}\cdot\nabla\mathbf{u}\cdot\mathbf{n}f-\nabla_s^2 f=g. \tag{4.4}$$

The exact solution

$$f(x,y,z,t)=\frac{e^{-t/2}z}{\sqrt{(x-t)^2+y^2+z^2}}.$$

$\phi=\sqrt{(x-t)^2+y^2+z^2}-1$ Errors at time $t=1$ are given in Table 2. In the adaptive mesh, the gird length of the root level mesh is $h=0.1$.

In the following the interface is deformed. The exact solutions for the level-set function and surfactant concentration are not available.

Example 4.3. $\phi(x,y,0)=\sqrt{x^2+y^2}-1$, $f(x,y,0)=1$, $\Omega=[-3,3]\times[-3,3]$. The velocity field is $\mathbf{u}=(y,0)^T$. The uniform mesh size is 240×240. The adaptive mesh has 3 levels with the root level mesh size being 60×60. $\Delta t=h/2$.

We first give a comparison among the UMSL method, the UMCE method, and the AMSL method. Three methods work well and produce similar results, as shown in Fig. 3. Here surfactant concentrations are plotted as functions of the arc length s. To make this plot, the interface is reconstructed by projecting the irregular grid points onto interface control points. A piecewise linear representation of the interface is used to calculate arc length. The starting point $s=0$ corresponds to the control point closest to the positive x-axis and s increases in the counterclockwise direction. The surfactant concentration on the interface are obtained the third-order ENO interpolation at the control points.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

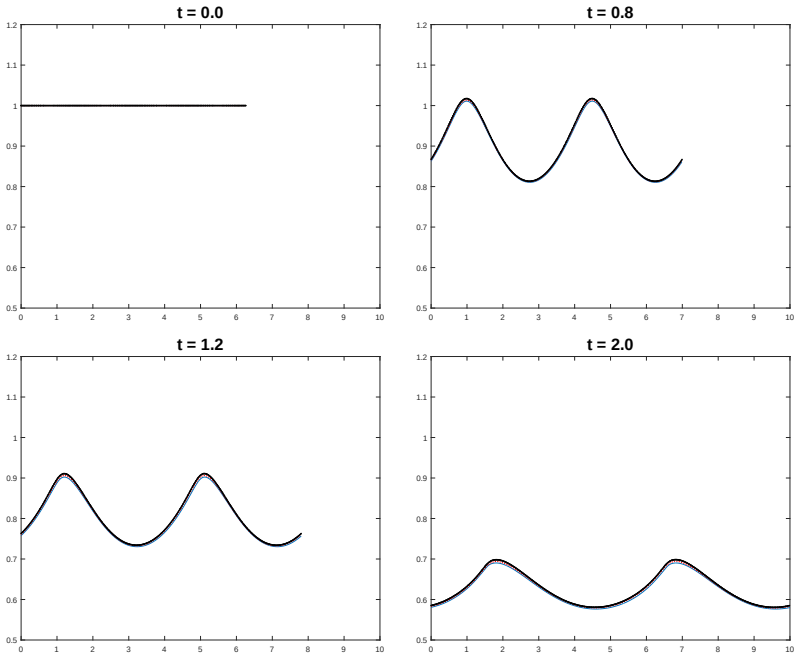


Figure 3: Surfactant concentrations as function of arc-length at different times: the UMSL method (red dashed line), the UMCE method (blue solid line), and the AMSL method (dark dash-dotted line). Three methods agree quite well.

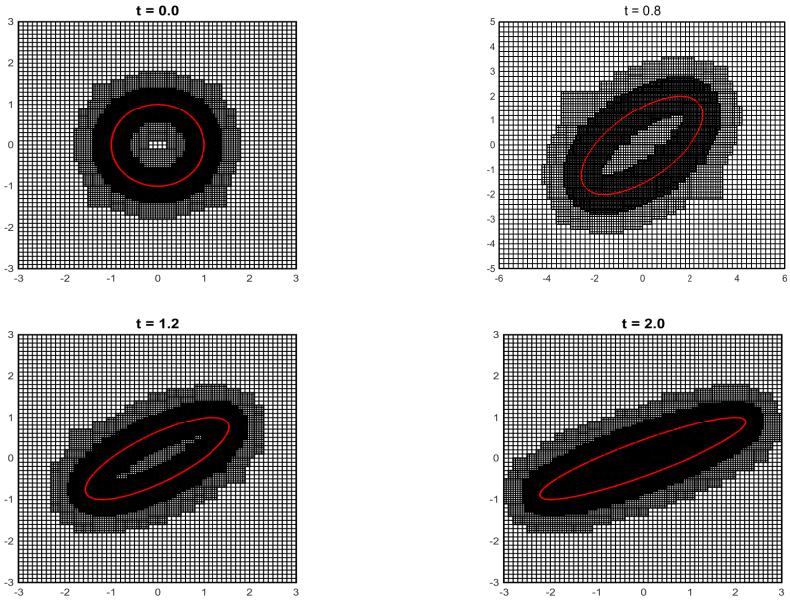


Figure 4: Adaptive meshes together with drop shapes at different times.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

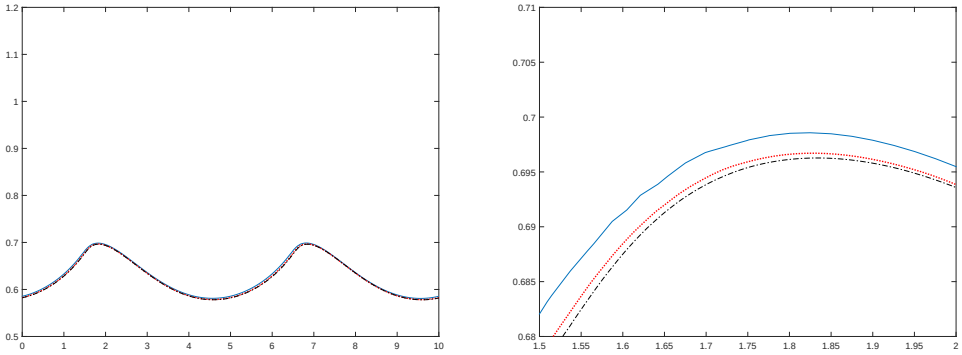


Figure 5: Left: Surfactant concentrations as function of arc-length at $t=2$ for three different adaptive mesh refinements ; Right: A zoomed plot around a drop tip. The finest grid length $h=0.025$ (blue solid line), 0.0125 (red dotted line), 0.00625 (black dash-dotted line). Numerical convergence is observed.

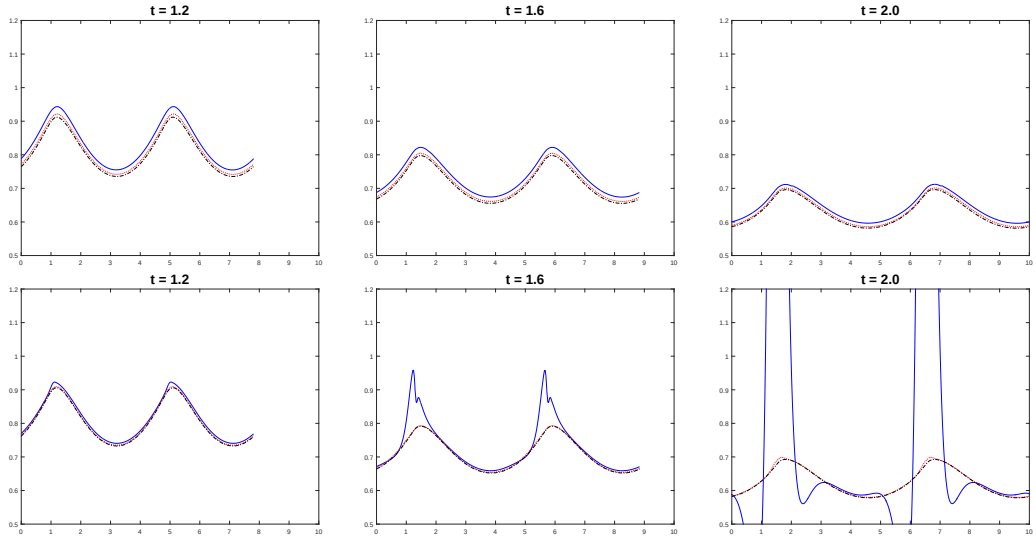


Figure 6: Surfactant concentrations as function of arc-length for three different time steps. Top row: UMSL method; Bottom row: UMCE method. $\Delta t=1.6h$ (blue solid line), $\Delta t=h$ (red dotted line), $\Delta t=h/2$ (black dash-dotted line), $h=0.025$.

The adaptive meshes at different times are given in Fig. 4.

A numerical convergence study for the adaptive S-L method is given in Fig. 5.

In Fig. 6, the UMSL method is stable for the three choices of time steps. The UMCE method is unstable when $\Delta t=1.6h$ due to the violation of the CFL condition.

Example 4.4. In this example we show that the method can handle discontinuous initial surfactant concentration. The setting is the same as in Example 4.3 except that the initial surfactant concentration is

$$f(x,y,0)=\begin{cases} 1, & \text{if } x<0, \\ 0.5, & \text{if } x\geq 0, \end{cases} \tag{4.5}$$

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

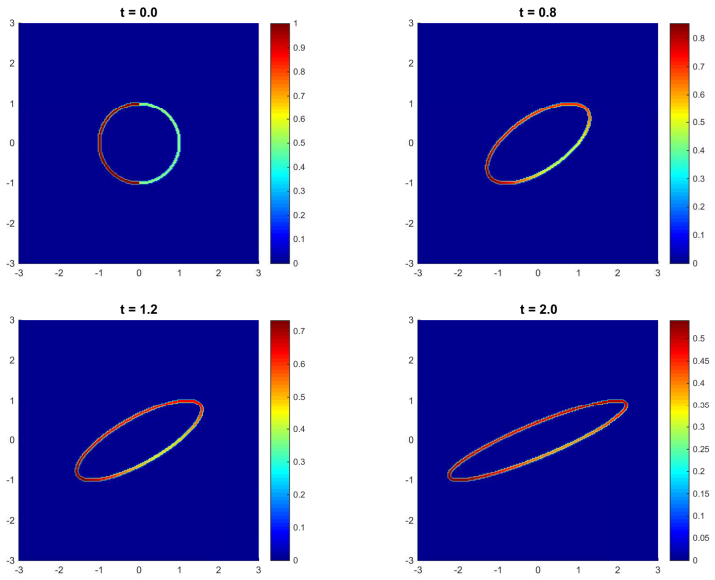


Figure 7: The moving interface together with surfactant concentration at different times. $\Delta t = h/2$.

Due to the same mesh refinement criterion (3.13), the time-dependent adaptive mesh is the same as in Example 4.3. The contour plots of surfactant concentration at different times are showed in Fig. 7.

Similar surfactant concentrations (as function of arc-length) produced by the the UMSL method, the UMCE method and the AMSL method are given in Fig. 8. It is also found that the UMSL method allows larger time step than the UMCE method (not shown here).

A numerical convergence study is given in Fig. 9.

Example 4.5. We consider a shear velocity $\mathbf{u} = (y,0,0)^T$. The initial level-set function $\phi(x,y,z,0) = \sqrt{x^2+y^2+z^2}-1$, and the initial surfactant concentration on the interface is uniformly 1. The computational domain $\Omega = [-4,4]\times[-2,2]\times[-2,2]$. In the adaptive mesh, the root level mesh size is $80\times 40\times 40$.

The drop shape together with surfactant concentration at different times are showed in Fig. 10. In this figure, the root level mesh is refined 3 times ($h_3 = 0.0125$). Total number of grid cells in all level meshes is 5793303, while the uniform mesh with grid length h_3 has 65536000 $(=640\times 320\times 320)$ grid cells. Adaptive mesh refinement saves the computer storage greatly.

With the deformation of the drop, its surface area increases, leading to surfactant dilution. Meanwhile, more surfactant tends to accumulate at the drop tips due to the convection.

A view of the adaptive meshes is given in Fig. 11.

Numerical convergence studies for surfactant concentrations at the cross-sections on yz-plane and xz-plane are given in Fig. 12.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

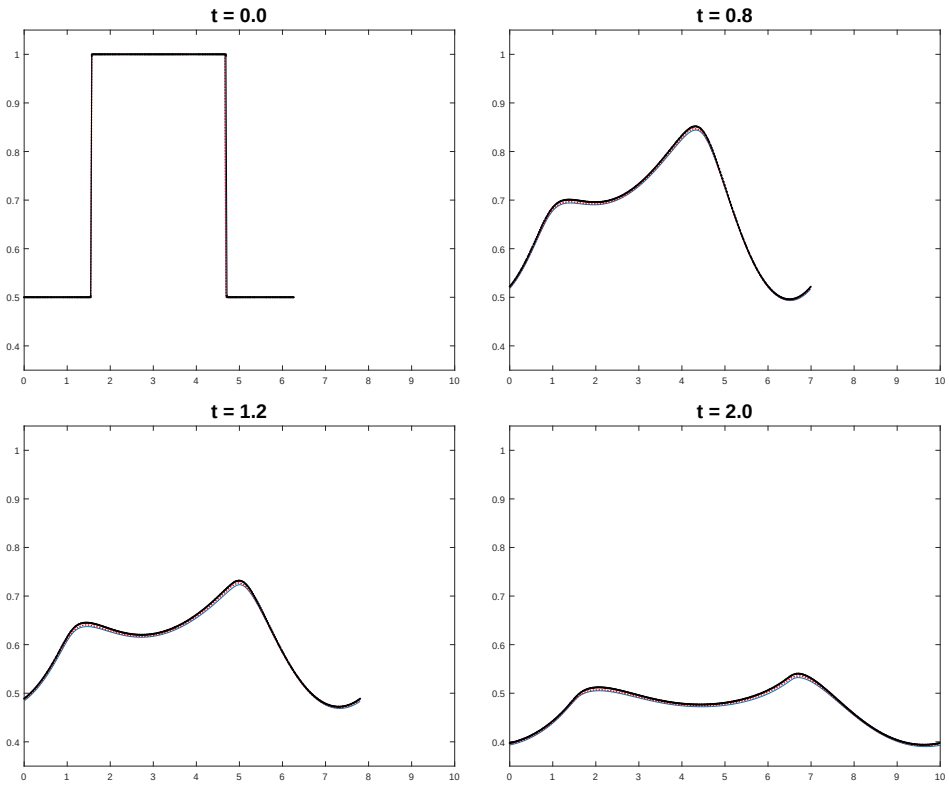


Figure 8: Surfactant concentrations as function of arc length for the three methods at different times: the UMSL method (red dashed line), the UMCE method (blue solid line), and the AMSL method (dark dash-dotted line). They agree quite well. $\Delta t = h/2$.

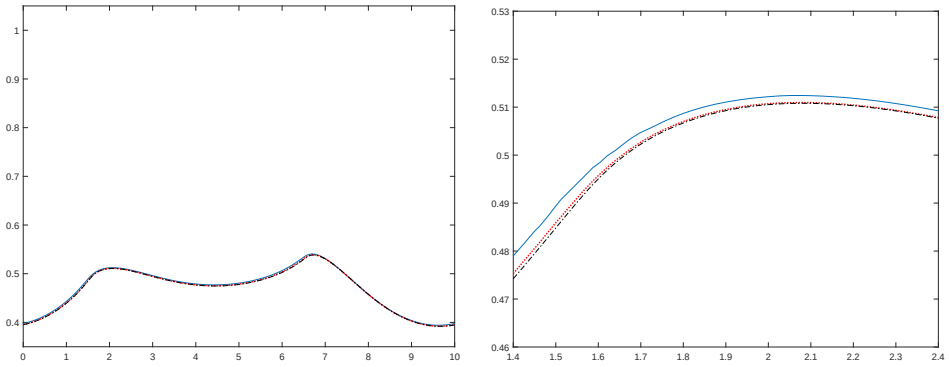


Figure 9: Left: Surfactant concentrations as functions of arc-length at $t = 2$ for three different adaptive mesh refinements; Right: A zoomed plot around a drop tip. The finest grid length $h = 0.025$ (blue solid line), 0.0125 (red dotted line), 0.00625 (black dash-dotted line). $\Delta t = h/2$. Numerical convergence is observed.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

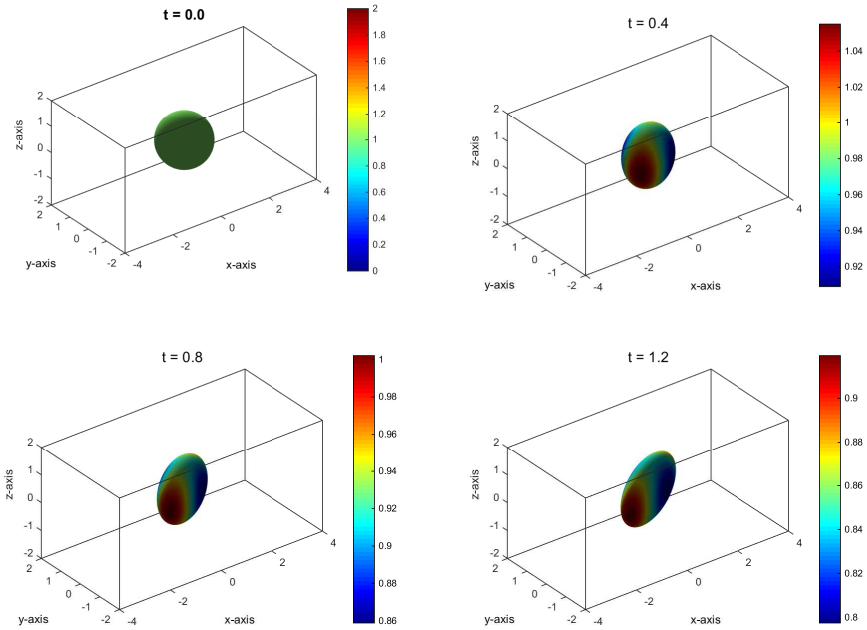


Figure 10: Contour plots of surfactant concentration at different times. $\Delta t = h/2$, h is the finest grid length.

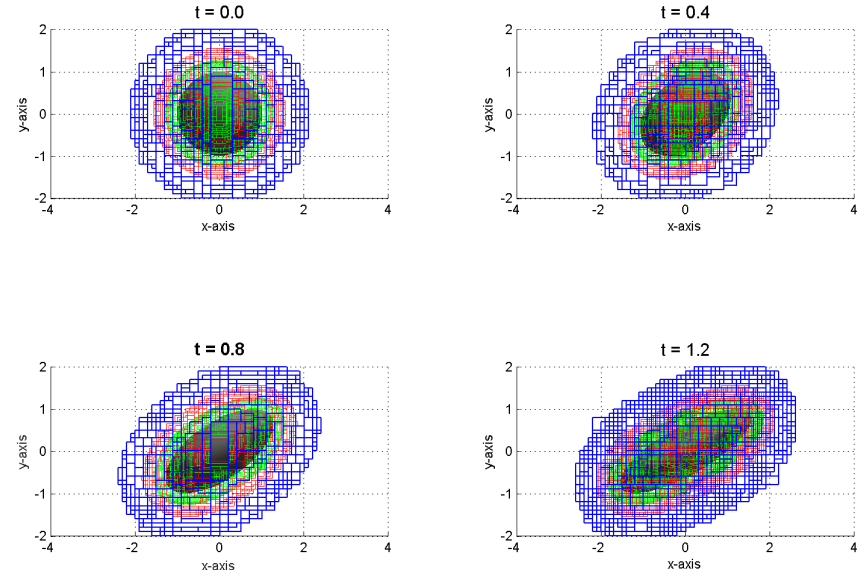


Figure 11: Adaptive meshes together with the drop shapes at different times. Same setting as in Fig. 10.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

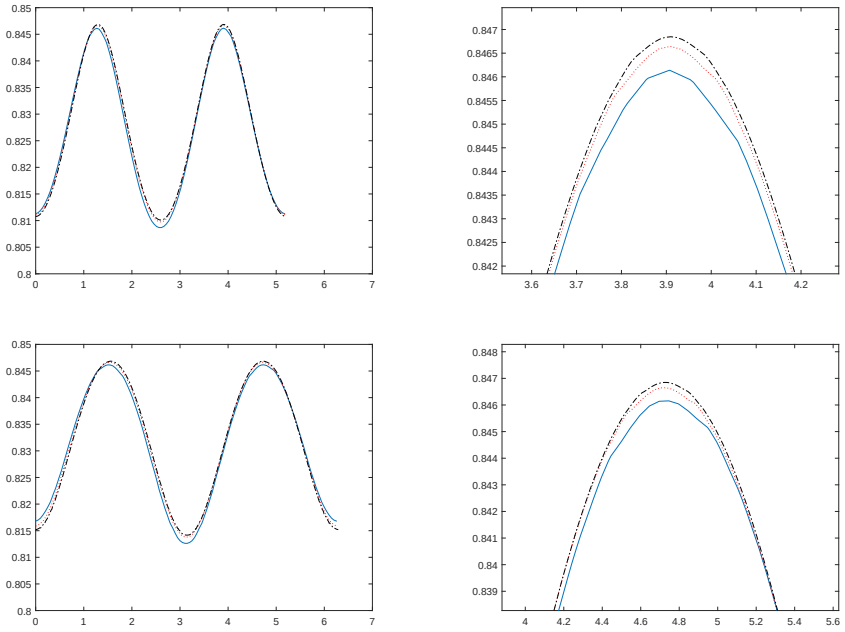


Figure 12: At two different cross-sections, surfactant concentrations as functions of arc-length at $t=1.2$ for three different adaptive mesh refinements. Top panel: at the cross-section on yz-plane (left) together with a zoomed plot (right). Bottom panel: at the cross-section on xz-plane (left) together with a zoomed plot (right). Due to the shear flow, drop is shrunk on yz-plane and elongated on xz-plane. The finest grid length $h=0.05$ (blue solid line), 0.025 (red dotted line), 0.0125 (black dash-dotted line). $\Delta t=h/2$. Numerical convergence is observed.

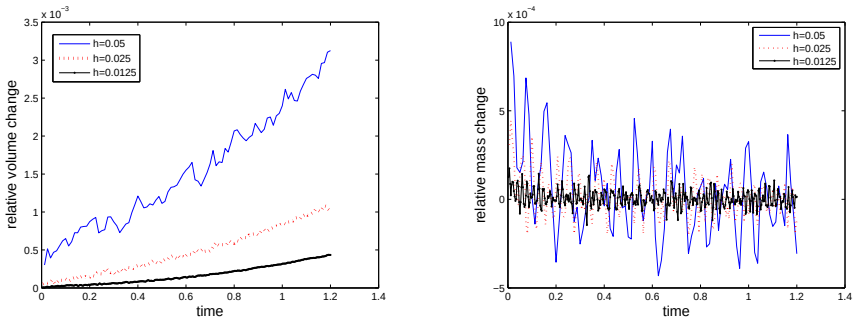


Figure 13: The relative changes of the volume enclosed by the interface (left) and of the total mass of surfactant on the interface (right). $\Delta t=h/2$, h is the finest grid length.

Fig. 13 shows a mesh refinement study for the relative changes of the total mass of surfactant on the interface and the volume enclosed by the interface. It can be seen that mesh refinement reduces errors significantly. The formulas for surfactant mass and the volume are the same as in [26].

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

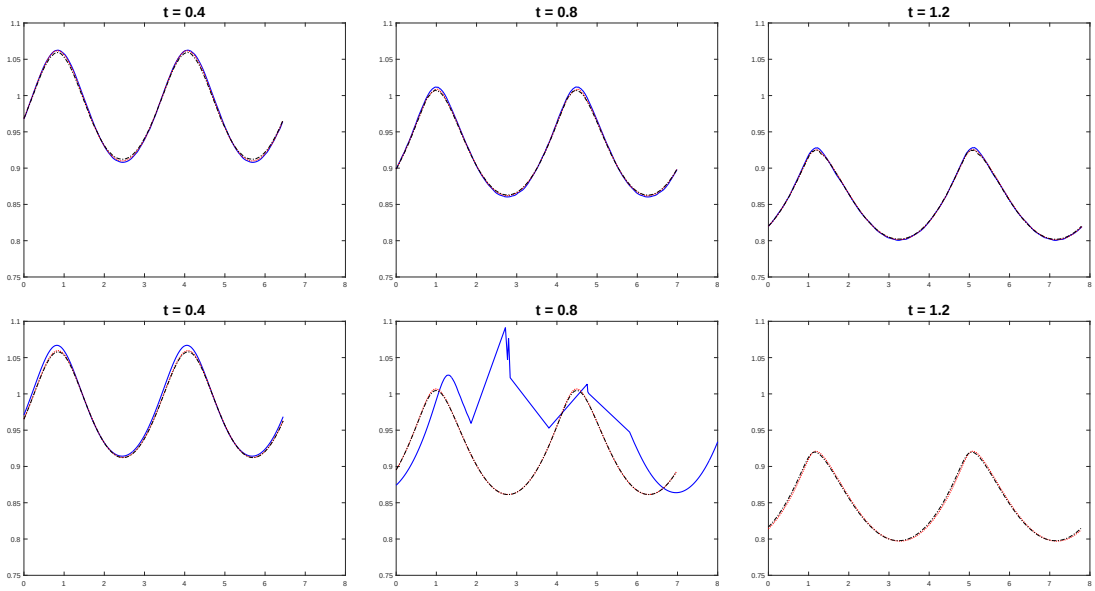


Figure 14: At the cross-section on xy-plane, surfactant concentrations as function of arc-length for three different time steps. Top row: UMSL method; Bottom row: UMCE method. $\Delta t=2h$ (blue solid line), $\Delta t=h$ (red dotted line), $\Delta t=h/2$ (black dash-dotted line),$h=0.05$. Code crashed for UMCE method when $\Delta t=2h$.

Lastly we show that the S-L method is more stable than the Eulerian method in Fig. 14. The Eulerian method is unstable when $\Delta t=2h$ due to the violation of the CFL condition.

5 Conclusions

An adaptive finite difference S-L level-set method has been proposed for solving the convection-diffusion equation on an evolving surface under a prescribed solenoidal velocity field. The S-L schemes removes both the CFL stability constraint and the stiffness of surface Laplacian, allowing large time step than the Eulerian method. In the adaptive mesh method, the block structured adaptive mesh refinement is used. It reduces the need of computer storage significantly. The level-set reinitialization equation is solved on all levels of the adaptive mesh, the other equations are only solved on the finest level mesh. This makes the adaptive mesh efficient.

Acknowledgements

We would like to thank the referees for the valuable comments. This work is partially supported by National natural science fund of China (No. 91430213 and No. 11571293) and Hunan Provincial Innovation Foundation for Postgraduate (No. CX2015B208).

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

References

[1] M. J. BERGER AND J. OLIGER, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comput. Phys., 53 (1984), pp. 484.

[2] M. J. BERGER AND P. COLELLA, *Local adaptive mesh refinement for shock hydrodynamics*, J. Comput. Phys., 82 (1989), pp. 64–84.

[3] M. BERGER AND I. RIGOUTSOS, *An algorithm for point clustering and grid generation*, IEEE Trans. Syst. Man. Cybern, 21 (1991), pp. 1278–1286.

[4] P. COLELLA, D. T. GRAVES, T. J. LIGOCKI, D. F. MARTIN, D. MODIANO, D. B. SERAFINI AND B. V. STRAALEN, *CHOMBO: software package for AMR applications: design document*, Technical report, Lawrence Berkeley National Laboratory, Applied Numerical Algorithms Group, NERSC Division; CA, USA, 2003

[5] R. COURANT, E. ISSACSON AND M. REES, *On the solution of nonlinear hyperbolic differential equations by finite difference*, Commun. Pure Appl. Math., 5 (1952), pp. 243–255.

[6] G. DZIUK AND C. ELLIOTT, *Finite element methods for surface PDEs*, Acta Numer., 22 (2013), pp. 289–396.

[7] T. F. DUPONT AND Y. LIU, *Back and forth error compensation and correction methods for semi-Lagrangian schemes with applications to level set interface computations*, Math. Comput., 76 (2007), pp. 647–668.

[8] C. M. ELLIOTT, B. STINNER, V. STYLES AND R. WELFORD, *Numerical computation of advection and diffusion on evolving diffuse interfaces*, IMA J. Numer. Anal., 31 (2011), pp. 786.

[9] J. GRANDE, *Eulerian finite element methods for parabolic equations on moving surfaces*, SIAM J. Sci. Comput., 36 (2014), pp. B248–B271.

[10] S. GROSS AND A. REUSKEN, Numerical Methods for Two-Phase Incompressible Flows, Springer, 2011.

[11] P. HANSBOA, M. G. LARSONB AND S. ZAHEDI, *Characteristic cut finite element methods for convection-diffusion problems on time dependent surfaces*, Comput. Meth. Appl. Mech. Eng., 293 (2015), pp. 431–461.

[12] J. LOWENGRUB, J.-J. XU AND A. VOIGT, *Surface phase separation and flow in a simple model of multicomponent drops and vesicles*, Fluid Dyn. Material Pro., 3 (2007), pp. 1–19.

[13] P. MACNEICE, K. M. OLSON, C. MOBARRY, R. DEFAINCHTEIN AND C. PACKER, *PARAMESH: A parallel adaptive mesh refinement community toolkit*, Comput. Phys. Commun., v126 (2000), pp. 330–354.

[14] C. MIN AND F. GIBOU, *A second order accurate level set method on non-graded adaptive cartesian grids*, J. Comput. Phys., 225 (2007), pp. 300–321.

[15] S. MITRAN, *BEARCLAW: a code for multiphysics applications with embedded boundaries: users manual*, Technical report, Dept. of Math., Univ. of North Carolina, NC, USA, 2006.

[16] M. A. OLSHANSKII, A. REUSKEN AND X. XU, *An Eulerian space-time finite element method for diffusion problems on evolving surfaces*, SIAM J. Numer. Anal., 52 (2014), pp. 1354–1377.

[17] M. A. OLSHANSKII AND A. REUSKEN, *Error analysis of a space-time finite element method for solving PDEs on evolving surfaces*, SIAM J. Numer. Anal., 52 (2014), pp. 2092–2120.

[18] S. OSHER AND J. SETHIAN, *Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12.

[19] C. Shu, Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws, Springer, 1998.

[20] J. STRAIN, *Semi-Lagrangian methods for level set equations*, J. Comput. Phys., 151 (1999), pp. 498–533.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press

[21] M. SUSSMAN, P. SMEREKA AND S. OSHER, *A level set approach for computing solutions to incompressible two-phase flow*, J. Comput. Phys., 114 (1994), pp. 146–159.
[22] M. SUSSMAN, A. S. ALMGREN, J. B. BELL, P. COLELLA, L. H. HOWELL AND M. L. WELCOMEY, *An adaptive level set approach for incompressible two-Phase flows*, J. Comput. Phys., 184 (1999), pp. 81–124.
[23] K. E. TEIGEN, X. LI, J. LOWENGRUB, F. WANG AND A. VOIGT, *A diffuse-interface approach for modeling transport, diffusion and adsorption/desorption of material quantities on a deformable interface*, Commun. Math. Sci., 7 (2009), pp. 1009–1037.
[24] K. E. TEIGEN AND S. T. MUNKEJORD, *Influence of surfactant on drop deformation in an electric field*, Phys. Fluids, 22 (2010), 112104.
[25] Y. WANG, S. SIMAKHINA AND M. SUSSMAN, *A hybrid level set-volume constraint method for incompressible two-phase flow*, J. Comput. Phys., 231 (2012), pp. 6438–6407.
[26] J.-J. XU AND H. ZHAO, *An Eulerian formulation for solving partial differential equations along a moving interface*, J. Sci. Comput., 19 (2003), pp. 573–594.
[27] J.-J. XU, H. Z. YUAN AND Y. Q. HUANG, *A 3D level-set method for solving convection-diffusion along moving surfaces (in Chinese)*, Sci. Sinna Math., 42(5) (2012), pp. 445–454.
[28] J.-J. XU, Z. LI, J. LOWENGRUB AND H. ZHAO, *A level set method for solving interfacial flows with surfactant*, J. Comput. Phys., 212 (2006), pp. 590–616.
[29] J.-J. XU, Z. LI, J. LOWENGRUB AND H. ZHAO, *Numerical study of surfactant-laden drop-drop interactions*, Commun. Comput. Phys., 10 (2011), pp. 453–473.
[30] J.-J. XU, Y. YANG AND J. LOWENGRUB, *A level-set continuum method for two-phase flows with insoluble surfactant*, J. Comput. Phys., 231 (2012), pp. 5897–5909.
[31] J.-J. XU, Y. HUANG, M.-C. LAI AND Z. LI, *A coupled immersed interface and level set method for three-dimensional interfacial flows with insoluble surfactant*, Commun. Comput. Phys., 15 (2014), pp. 451–469.
[32] J.-J. XU AND W. REN, *A level-set method for two-phase flows with moving contact line and insoluble surfactant*, J. Comput. Phys., 263 (2014), pp. 71–90.
[33] W. D. SHI, J.-J. XU AND S. SHI, *A simple implementation of the semi-Lagrangian level-set method*, Adv. Appl. Math. Mech., 2016 (in press).
[34] H. ZHAO, T. CHAN, B. MERRIMAN AND S. OSHER, *A variational level set approach to multiphase motion*, J. Comput. Phys., 127 (1996), pp. 179–195.

https://doi.org/10.4208/aamm.OA-2016-0052 Published online by Cambridge University Press