Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **9**, No. 1, pp. 104-124

DOI: 10.4208/aamm.2015.m1305
February 2017

A Simple Implementation of the Semi-Lagrangian Level-Set Method

Weidong Shi[1,2], Jian-Jun 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 21 September 2015; Accepted (in revised version) 31 December 2015

Abstract. Semi-Lagrangian (S-L) methods have no CFL stability constraint, and are more stable than the Eulerian methods. In the literature, the S-L method for the level-set re-initialization equation was complicated, which may be unnecessary. Since the re-initialization procedure is auxiliary, we propose to use the first-order S-L scheme coupled with a projection technique to improve the accuracy at the grid points just adjacent to the interface. Standard second-order S-L method is used for evolving the level-set convection equation. The implementation is simple, including on the block-structured adaptive mesh. The efficiency of the S-L method is demonstrated by extensive numerical examples including passive convection of interfaces with corners/kinks/large deformation under given velocity fields, a geometrical flow with topological changes, simulations of bubble/ droplet dynamics in incompressible two-phase flows. In terms of accuracy it is comparable to the other existing methods.

AMS subject classifications: 65M06, 65M20, 76T10

Key words: Semi-Lagrangian method, level-set method, interface motion, two-phase flow, bubble/ droplet dynamics, block-structured adaptive mesh.

1 Introduction

The level-set method of Osher and Sethian [11] has been an invaluable tool in computing interface problems for a wide range of applications.

The level-set convection equation reads

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0, \tag{1.1}$$

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

 ©2017 Global Science Press

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

where ϕ is the level-set function, its zero level-set being the interface. $\mathbf{u}$ is the convective velocity field.

In practice, the level-set function ϕ needs to be re-initialized to prevent the formation of too large or small gradient of ϕ around the interface. Originally proposed in [22], the following auxiliary Hamilton-Jacobi equation has been widely used and studied for the purpose of re-initialization:

$$\begin{cases} \dfrac{\partial \phi}{\partial \tau} + S(\phi_0)(|\nabla \phi| - 1) = 0, \\ \phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}), \end{cases} \tag{1.2}$$

where ϕ_0 is the level-set function before the re-initialization, τ is the pseudo-time and $S(x)$ is the sign function of x defined as

$$S(x) = \begin{cases} -1, & \text{if } x < 0, \\ 0, & \text{if } x = 0, \\ 1, & \text{if } x > 0. \end{cases} \tag{1.3}$$

As $\tau \to \infty$, the solution of (1.2) approaches to a steady state, i.e., the so-called signed distance function. In many practical applications, the re-initialization equation (1.2) does not need to be solved to the steady state. Instead, just a few time iterations are enough. Though the level-set function ϕ is one dimension higher than the interface, the computational cost can be reduced significantly by using the local level-set technique proposed in [13]. In the local level-set approach, the Eqs. (1.1) and (1.2) are solved only in small tubes containing the interface, since only the zero level set is physical.

The semi-Lagrangian (S-L) methods trace back along the characteristic curves to locate the departure points by solving ordinary differential equations, and then use appropriate interpolations to get the approximation of solution at the grid points. Generally speaking the S-L methods automatically satisfy the CFL condition by shifting the stencil. Thus it is more stable than the classical Eulerian methods. The S-L methods do not handle shock discontinuities well. However, there is no shock for the level-set function since it is at least Lipschitz continuous (e.g., [17]).

The S-L methods date back at least to the Courant-Isaacson-Rees (CIR) method [3]. They have been popular in numerical computations in atmospheric sciences (e.g., [16, 29]). The S-L methods have received a lot of interest in the level-set community recently. In a series of works [17–21], J. Strain studied the S-L schemes for the level-set equation (1.1), the computational geometry based method on quadtrees was used for the re-initialization. Adaptive mesh methods on tree-structured Cartesian grid were also developed. In [4,5,14], the first-order semi-Lagrangian scheme was used as building blocks and the differential equations were evolved forward and backward in time to get an error estimate, then the error information was exploited in another forward evolution step to obtain more accurate solution. In [7], a hybrid particle level-set method was proposed to improve the accuracy of the first-order S-L scheme. In [10], a non-graded adaptive mesh

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

method on tree-structured Cartesian grid was developed, which used a second-order S-L scheme for Eq. (1.1), coupled with a C-E scheme for the re-initialization equation (1.2). More recently, an adaptive S-L method for the level-set equations on block-structured Cartesian grid was developed and applied to incompressible two-phase flows in [23]. However, the re-initialization algorithm together with the volume constraint procedure increased the complexity of the implementation, which may be unnecessary.

In this work we study a simple S-L method for the level-set equations. In the method, standard second-order S-L scheme (as in [10, 23]) is used for Eq. (1.1). For the re-initialization equation (1.2), we simply use the first-order S-L scheme due to the jump discontinuity of the velocity across the interface, and a second-order projection technique is used to improve the accuracy. The S-L schemes are studied by truncation error analysis, numerical examples including interfaces with corners/kinks, topological changes, simulations of bubble/droplet dynamics in two-phase flows, and comparison with the other methods.

The paper is organized as follows. The S-L method for the level-set equations is described and analyzed in Section 2. The numerical examples are given in Section 3. Conclusions and discussion are given in Section 4.

2 Semi-Lagrangian method

We first briefly introduce the concept of the semi-Lagrangian method. Consider the general convection equation of the following form:

$$\frac{\partial f(\mathbf{x},t)}{\partial t}+\mathbf{u}\cdot\nabla f(\mathbf{x},t)=g(\mathbf{x},t), \tag{2.1}$$

with given initial value $f(\mathbf{x},0)$.

The characteristic curve $\mathbf{x}=\mathbf{x}(t)$ of Eq. (2.1) solves the following ordinary differential equation:

$$\frac{d\mathbf{x}}{dt}=\mathbf{u}(x,t). \tag{2.2}$$

Along the characteristic curve, Eq. (2.1) can be rewritten as

$$\frac{df(\mathbf{x}(t),t)}{dt}=g(\mathbf{x}(t),t). \tag{2.3}$$

For simplicity, we describe the S-L method in 2D. The 3D method can be obtained by simply 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 uniform time step, $x_i=a+ih$, $y_j=c+jh$, $t^n=n\Delta t$.

Given f^n, a S-L scheme marches from t^n to t^{n+1} by integrating equation (2.1) along the characteristic curve backward over the time interval $[t^n,t^{n+1}]$ with the starting point

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

$(\mathbf{x}_{ij}, t^{n+1})$:

$$f_{ij}^{n+1} - f^n(\mathbf{x}_d) = \int_{t^n}^{t^{n+1}} g(\mathbf{x}(t), t)dt, \tag{2.4}$$

where $\mathbf{x}_d$ is the so-called departure point, which solves (2.2) backward to time t^n with the starting value $\mathbf{x}(t^{n+1}) = \mathbf{x}_{ij}$.

For the integration in the right hand side of Eq. (2.4), one can use the trapezoidal rule to get a second-order S-L scheme for Eq. (2.1) as following:

$$f_{ij}^{n+1} = f^n(\mathbf{x}_d) + \frac{\Delta t}{2}(g_{ij}^{n+1} + g^n(\mathbf{x}_d)). \tag{2.5}$$

Or one can use the left point rule to to get a first-order S-L scheme as following:

$$f_{ij}^{n+1} = f^n(\mathbf{x}_d) + \Delta t g_{ij}^n. \tag{2.6}$$

An alternative way to construct semi-Lagrangian schemes is to differentiate along the characteristic curve instead of the integration (e.g., [24, 25]).

2.1 Locating the departure point

The departure point $\mathbf{x}_d$ is obtained by the methods for solving ODE (2.2) as follows

- the first-order Euler method: $\mathbf{x}_d = \mathbf{x}_{ij} - \Delta t \mathbf{u}_{ij}^n$.
- the second-order Runge-Kutta(RK) method:

$$\mathbf{x}^* = \mathbf{x}_{ij} - \frac{\Delta t}{2}\mathbf{u}_{ij}^n, \tag{2.7a}$$

$$\mathbf{x}_d = \mathbf{x}_{ij} - \Delta t \mathbf{u}^{n+\frac{1}{2}}(\mathbf{x}^*). \tag{2.7b}$$

When the velocity is time dependent, $\mathbf{u}(\mathbf{x}^*)^{n+\frac{1}{2}}$ in the second stage of RK method can be calculated by the second-order accurate time extrapolation:

$$\mathbf{u}^{n+\frac{1}{2}}(\mathbf{x}^*) = \frac{3}{2}\mathbf{u}^n(\mathbf{x}^*) - \frac{1}{2}\mathbf{u}^{n-1}(\mathbf{x}^*) + \mathcal{O}(\Delta t^2). \tag{2.8}$$

In the S-L scheme, the points $\mathbf{x}_d, \mathbf{x}^*$ are not grid points, the terms $f^n(\mathbf{x}_d)$, $\mathbf{u}^n(\mathbf{x}^*)$, $\mathbf{u}^{n-1}(\mathbf{x}^*)$ need to be approximated by appropriate interpolations with data at grid points.

For the first-order Euler method, the following bilinear interpolation is used.

If $\tilde{\mathbf{x}} \equiv (\tilde{x}, \tilde{y}) \in [x_i, x_{i+1}] \times [y_j, y_{j+1}]$, then

$$\begin{aligned} f(\tilde{\mathbf{x}}) = &[(x_{i+1} - \tilde{x})(y_{j+1} - \tilde{y})f_{i+1,j+1} + (x_i - \tilde{x})(y_j - \tilde{y})f_{ij} \\ &- (x_{i+1} - \tilde{x})(y_j - \tilde{y})f_{i+1,j} - (x_i - \tilde{x})(y_{j+1} - \tilde{y})f_{i,j+1}]/h^2. \end{aligned} \tag{2.9}$$

For the second-order R-K method, we use the third-order ENO scheme (e.g., [15]). The ENO interpolation is done in a way of dimension by dimension. The ENO construction of the third-order interpolation polynomial in 1D can be found in e.g., [15]. The ENO interpolation is efficient in handling spatially non-smooth velocity fields.

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

2.2 Second-order S-L scheme for the level-set convection

The S-L scheme for level-set equation (1.1) is simply

$$\phi_{ij}^{n+1}=\phi^n(\mathbf{x}_d), \tag{2.10}$$

where the departure point $\mathbf{x}_d$ is calculated by the second order RK method, and the interpolation is done by using the third-order ENO scheme.

2.3 S-L schemes for the level-set re-initialization

The re-initialization equation (1.2) can be rewritten as

$$\frac{\partial\phi}{\partial\tau}+S(\phi_0)\mathbf{n}\cdot\nabla\phi=S(\phi_0). \tag{2.11}$$

This is a nonlinear H-J equation with the velocity $\mathbf{u}=S(\phi_0)\mathbf{n}$ and the source term $g=S(\phi_0)$.

The set of grid points just adjacent to the interface is identified as

$$\Gamma=\{\mathbf{x}_{ij}:\beta_1\beta_2\leq 0\}, \tag{2.12}$$

where

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

and

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

Such grid points shall be called irregular grid points, while the other grid points are regular.

If $\mathbf{x}_{ij}$ is irregular, the departure point $\mathbf{x}_d$ often lies in the other side of the interface. This causes that the level-set function changes sign after the re-initialization. Therefore the level-set function at the irregular grid points should be re-initialized separately. Or one may simply keep the level-set function unchanged at the irregular grid points, as in [5].

2.3.1 S-L schemes

Let $\mathbf{x}_{ij}$ be regular. If the pseudo time step $\Delta\tau<h$, $\mathbf{x}_d$ and $\mathbf{x}_{ij}$ are on the same side of the interface. The firt-order S-L scheme reads follows:

$$\phi_{ij}^{n+1}=\begin{cases}\phi^n(\mathbf{x}_d)+\Delta\tau, & \text{if } \phi_0(\mathbf{x}_{ij})>0,\\ \phi^n(\mathbf{x}_d)-\Delta\tau, & \text{if } \phi_0(\mathbf{x}_{ij})<0,\end{cases} \tag{2.13}$$

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

where the departure point $\mathbf{x}_d$ is calculated by using the first-order Euler method described in Section 2.1.

The second-order S-L scheme is the same above except that $\mathbf{x}_d$ is obtained by the second-order RK scheme described in Section 2.1.

In evaluating the normal $\mathbf{n}=\nabla\phi/|\nabla\phi|$, the modified difference scheme of [5] 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

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

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.

2.3.2 Projection technique for irregular grid points

Here we propose a projection technique to improve the accuracy at the irregular grid points.

Let $\mathbf{x}_{ij}$ be irregular, its projection point $\tilde{\mathbf{x}}=\mathbf{x}_{ij}+\alpha\mathbf{n}$ at the interface is obtained by solving the following quadratic equation for α:

$$0=\phi_0(\mathbf{x}_{ij}+\alpha\mathbf{n})\approx\phi_0(\mathbf{x}_{ij})+\alpha|\nabla\phi_0(\mathbf{x}_{ij})|+\frac{\alpha^2}{2}\mathbf{n}^T\text{He}(\phi_0(\mathbf{x}_{ij}))\mathbf{n}, \tag{2.14}$$

where $\mathbf{n}$ is the normal of ϕ_0 at $\mathbf{x}_{ij}$, and He(ϕ_0) is the Hessian matrix:

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

When the quadratic equation (2.14) 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 $\tilde{\mathbf{x}}=\mathbf{x}_{ij}-\phi_0(\mathbf{x}_{ij})(\mathbf{x}_{i+1,j}-\mathbf{x}_{ij})/(\phi_0(\mathbf{x}_{i+1,j})-\phi_0(\mathbf{x}_{ij}))$.

$\tilde{\mathbf{x}}$ 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

$$\phi_{ij}=\begin{cases}\max\{\phi_0(\mathbf{x}_{ij}),-|\mathbf{x}_{ij}-\tilde{\mathbf{x}}|\}, & \text{if } \phi_0(\mathbf{x}_{ij})<0,\\ \min\{\phi_0(\mathbf{x}_{ij}),|\mathbf{x}_{ij}-\tilde{\mathbf{x}}|\}, & \text{if } \phi_0(\mathbf{x}_{ij})>0.\end{cases} \tag{2.15}$$

Note that the zero level set remains unchanged. Once the projection step is done at the beginning of the re-initialization, the values of ϕ at the irregular grid points are kept unchanged in the remaining re-initialization process.

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

2.4 Truncation error and stability

We assume that the level-set function ϕ and the velocity $\mathbf{u}$ are smooth.

Let us denote the interpolation operator as I, then $Ig = g + \mathcal{O}(h^p)$ for any smooth function g, where $p=3$ for the bilinear interpolation, and $p=4$ for the ENO interpolation. Also we denote the time extrapolation operator in (2.8) as E. Then $Eg = g + \mathcal{O}(\Delta t^2)$.

The truncation error e for the S-L scheme (2.10) is determined by plugging the exact solution into it:

$$\phi(\mathbf{x}(t_{n+1}), t^{n+1}) = I\phi(\mathbf{x}_d, t^n) + e\Delta t, \tag{2.16}$$

where

$$\begin{aligned}
\mathbf{x}_d &= \mathbf{x}(t^{n+1}) - \Delta t IE\mathbf{u}\left(\mathbf{x}(t^{n+1}) - \frac{\Delta t}{2}\mathbf{u}(\mathbf{x}^{n+1}, t^n), t^{n+\frac{1}{2}}\right)\right) \\
&= \mathbf{x}(t^{n+1}) - \Delta t\mathbf{u}\left(\mathbf{x}(t^{n+1}) - \frac{\Delta t}{2}\mathbf{u}(\mathbf{x}^{n+1}, t^n), t^{n+\frac{1}{2}}\right) + \mathcal{O}(\Delta t^3) + \Delta t\mathcal{O}(h^4) \\
&= \mathbf{x}(t^n) + \mathcal{O}(\Delta t^3) + \Delta t\mathcal{O}(h^4),
\end{aligned}$$

where the last equality can be found in any classic book in RK method for ODEs (see e.g., [9]). Thus (2.16) becomes

$$\phi(\mathbf{x}(t^{n+1}), t^{n+1}) = \phi(\mathbf{x}(t^n), t^n) + \mathcal{O}(\Delta t^3) + \Delta t\mathcal{O}(h^4) + \mathcal{O}(h^4) + e\Delta t. \tag{2.17}$$

Two terms involving ϕ are canceled. Thus we obtain the local truncation error for the SL scheme (2.10)

$$e = \mathcal{O}(\Delta t^2) + \mathcal{O}\left(\frac{h^4}{\Delta t}\right). \tag{2.18}$$

Similar results can be obtained for S-L schemes for the re-initialization. The truncation error at the irregular grid points $\mathcal{O}(h^2)$ due to the projection technique.

The truncation error analysis indicates the errors of the S-L schemes are non-monotonic with respect to time step Δt.

In [17], truncation error was analyzed for the first-order S-L scheme for the 1D level-set linear convection equation. It was also shown that the S-L scheme is unconditionally stable in maximum norm when the linear interpolation was used. Unfortunately it may be difficult to analyse that stability for the S-L schemes in this paper, due to the fact that the higher-order interpolations (such as the ENO scheme) may allow the maximum norm of the solution to increase, and the re-initialization equation (1.2) is nonlinear. Nevertheless the S-L methods are generally more stable than the Eulerian methods since the S-L methods have no CFL stability constraint.

3 Numerical examples

In this section, we demonstrate the efficiency of the S-L method (i.e., the 2nd-order S-L scheme for the level-set convection coupled with the 1st-order S-L scheme for the re-

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

initialization) by several numerical examples. In all the examples, the level-set equation (1.1) and the re-initialization equation (1.2) are solved in the tubes containing the interface. For example, we choose $\{\mathbf{x}:|\phi(\mathbf{x},t)|\leq 6h\}$ for the level-set advection, and $\{\mathbf{x}:|\phi(\mathbf{x},t)|\leq 12h\}$ for the re-initialization. Also Eq. (1.2) is not solved to steady state. Unless specified otherwise, we do 9 pseudo time iterations with pseudo time step $\Delta\tau=h/5$. In the following the S-L method refers the second-order S-L scheme for the level-set convection coupled with the first-order S-L scheme for the re-initialization. The S-L method is compared with two other methods: the classical high-order Eulerian (C-E) method of [8] (i.e., the third-order WENO scheme for spatial discretization and the third-order TVD RK scheme for time marching for both the level-set convection and the re-initialization), and a hybrid S-L-Eulerian (S-L-E) method similar to [10] (i.e., the second-order S-L scheme for level-set convection, and a high-order Eulerian method for the re-initialization).

It is found that there is no significant difference between the results of the first-order and the second-order schemes for the re-initialization. So we mainly present the results of the 1st-order re-initialization scheme, except in Example 3.3.

3.1 Passive interface motion under a given velocity field

Example 3.1. Consider an equilateral triangle initially centered at the origin. The computational domain $\Omega=[-2,2]\times[-2,2]$. We compute the rotation of the triangle around the origin with the velocity $\mathbf{u}=2\pi(-y,x)^T$. The S-L method is compared with the S-L-E method and the C-E method. The results are given in Fig. 1. The S-L method gives the most accurate result.

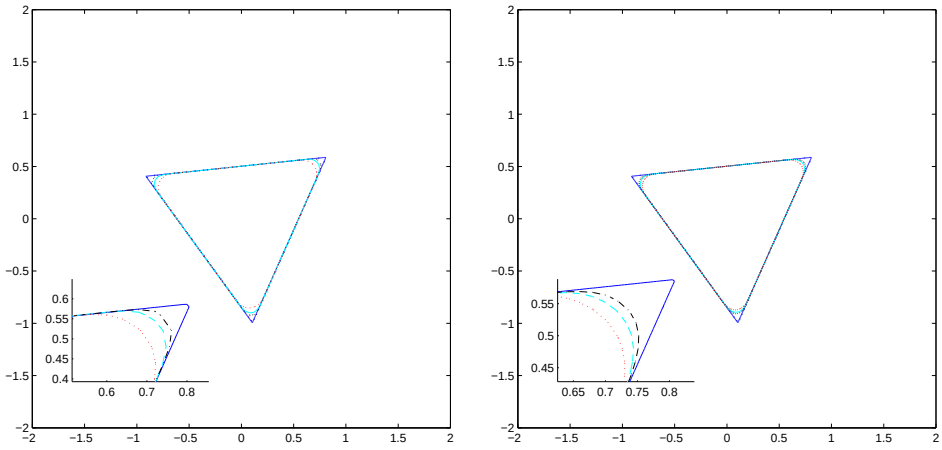


Figure 1: Comparison among the S-L method (black dash-dotted), the S-L-E method (cyan dashed) and the C-E (red dotted) after one round rotation. The exact interface is plotted in blue solid line. The upper right corner is zoomed for visualization purpose. The Mesh size $=200\times200$. Left: $\Delta t=h/4$, $\Delta\tau=h/2$. Right: $\Delta t=h/8$, $\Delta\tau=h/5$.

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

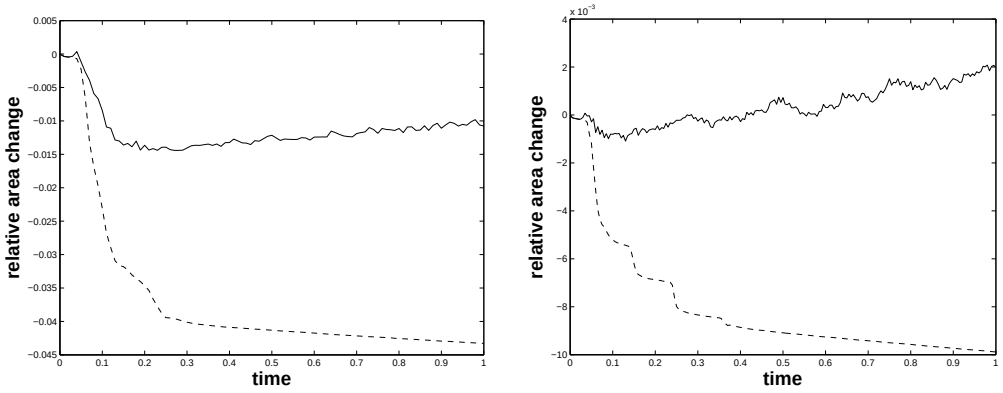


Figure 2: Comparison between the S-L method (solid line) and the S-L method without the projection technique (dashed line). The Mesh size $=200\times200$ (left), and 400×400 (right). $\Delta t=h/2$, $\Delta\tau=h/4$.

The C-E method is unstable when $\Delta t=h/2$ due to the violation of the CFL condition. The semi-Lagrangian methods allow larger time step. In Fig. 2, the errors for the area conservation when $\Delta t=h/2$ are plotted for the S-L method and the S-L method without the projection technique (i.e., the values of the level-set function at the irregular grid points are kept unchanged in the re-initialization). It can be seen that the projection technique reduces the errors significantly. The area/volume of the region enclosed by the interface is calculated by applying the trapezoidal rule to the following integral:

$$\text{area}=\int_{\Omega}H(-\phi)d\mathbf{x}. \tag{3.1}$$

The Heaviside function is smoothed as

$$H_h(x)=\begin{cases}0, & \text{if } x<-\epsilon,\\ \frac{1}{2}\Big(1+\frac{x}{\epsilon}+\frac{1}{\pi}\sin\Big(\frac{\pi x}{\epsilon}\Big)\Big), & \text{if } |x|\le\epsilon,\\ 1, & \text{else,}\end{cases} \tag{3.2}$$

where $\epsilon=2h$.

Example 3.2. The Zalesak disk problem [28] is one of the difficult test problems for Eulerian interface capturing methods because of their implicit representation of the interface. The challenge for the computation is that the disk has corner points, curves and straight lines. The computational domain $\Omega=[-2,2]\times[-2,2]$. Initially the cutout circle is centered at the origin with radius 1. The slot being cut out has width $2\sin\frac{\pi}{18}$, and length 1. The disk is rotated under the same velocity field as in Example 3.1. At time $t=1$, the disk returns to its initial position. Comparisons among the S-L method, the C-E method and the C-E-L method are shown in Fig. 3 for several mesh sizes and time steps. The S-L method produces the best results. Note that the C-E method is unstable when $\Delta t=h/2$.

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

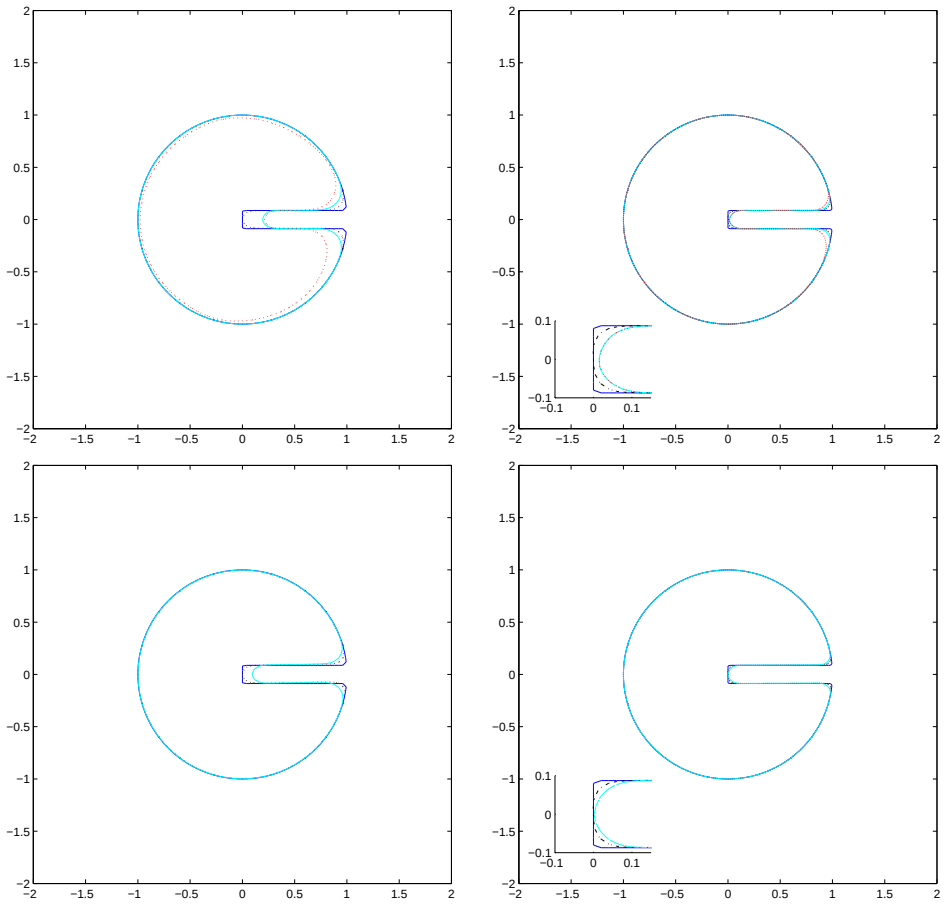


Figure 3: Comparison of the S-L method (black dash-dotted), the S-L-E method (cyan dashed) and the C-E method (red dotted) for the Zalesak disk. The exact solution is in blud solid line. Mesh size $=100\times100$ (left column), 200×200 (right column), $\Delta t=h/4$ (top row), $h/2$ (bottom row). The C-E method is unstable when $\Delta t=h/2$.

In the next, we perform a mesh refinement study. We measure the error by using the following quantity:

$$E_h=\max\{|\phi(\mathbf{x},t)-\phi_h(\mathbf{x},t)|:|\phi(\mathbf{x},t)|\leq 1.5h\}, \tag{3.3}$$

where ϕ and ϕ_h are the exact and numerical solutions respectively. Here the exact solution is taken to be the signed distance function of the Zalesak disk. The numerical solution ϕ_h at $t=1$ is re-initialized a few more pseudo time steps so that it is also approximately a signed distance function.

The errors and the estimated convergence rates for the three methods are given in Table 1. Though the convergence rate of the S-L method is lower than the other methods, the errors of the S-L method is the smallest on coarse grids, and comparable to the other

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

Table 1: Errors and convergence rates of the methods for the Zalesak disk problem. $\Delta t = h/4$.

	S-L				S-L-E				C-E			
h	e_∞	rate	e_2	rate	e_∞	rate	e_2	rate	e_∞	rate	e_2	rate
0.04	8.12D-2		1.22D-2		2.01D-1		4.48D-2		2.37D-1		7.40D-2	
0.02	5.04D-2	0.69	4.36D-3	1.48	6.26D-2	1.68	6.96D-3	2.69	1.04D-1	1.18	1.17D-2	2.66
0.01	3.14D-2	0.68	1.48D-3	1.56	3.18D-2	0.98	1.75D-3	1.99	4.43D-2	1.23	2.39D-3	2.29
0.005	2.00D-2	0.65	5.34D-5	1.47	1.54D-2	1.05	4.04D-4	2.11	2.00D-2	1.14	5.05D-5	2.24

Table 2: The running times (in seconds) by the methods for one round rotation of the Zalesak disk. $\Delta t = h/4$.

h	S-L	S-L-E	C-E
0.04	2.65	9.38	11.25
0.02	15.6	60.4	86.72
0.01	114.33	433	710.82
0.005	1255.39	3347.62	6422.22

methods on the finest grid. The running times of the methods are given in Table 2. The S-L method is significantly faster.

Example 3.3. The interface curvature plays an important role in many applications, for example, two-phase flows driven by surface tension (see Section 3.4). In the level-set techniques, curvature is represented via the level-set function as following:

$$\kappa = \nabla \cdot \mathbf{n} = \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right). \tag{3.4}$$

Standard central difference scheme is used in calculation. We consider that an ellipse is rotated one full round under the same velocity as in Example 3.1. The initial level-set function $\phi = \sqrt{(x/1.5)^2 + (y/0.75)^2} - 1$.

In Fig. 4 the curvature at different times is plotted as functions of arc-length. 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. The S-L method produces stable curvature calculations, though some oscillations are observed.

A mesh refinement study is presented in Fig. 5. In most part of the interface, the calculation of curvature is improved as the mesh is refined.

Comparisons in Fig. 6 shows that the S-L method gives the best result in curvature calculation.

3.2 A geometrical flow

One of the advantages of the level-set method is the capability of handling topological change automatically.

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

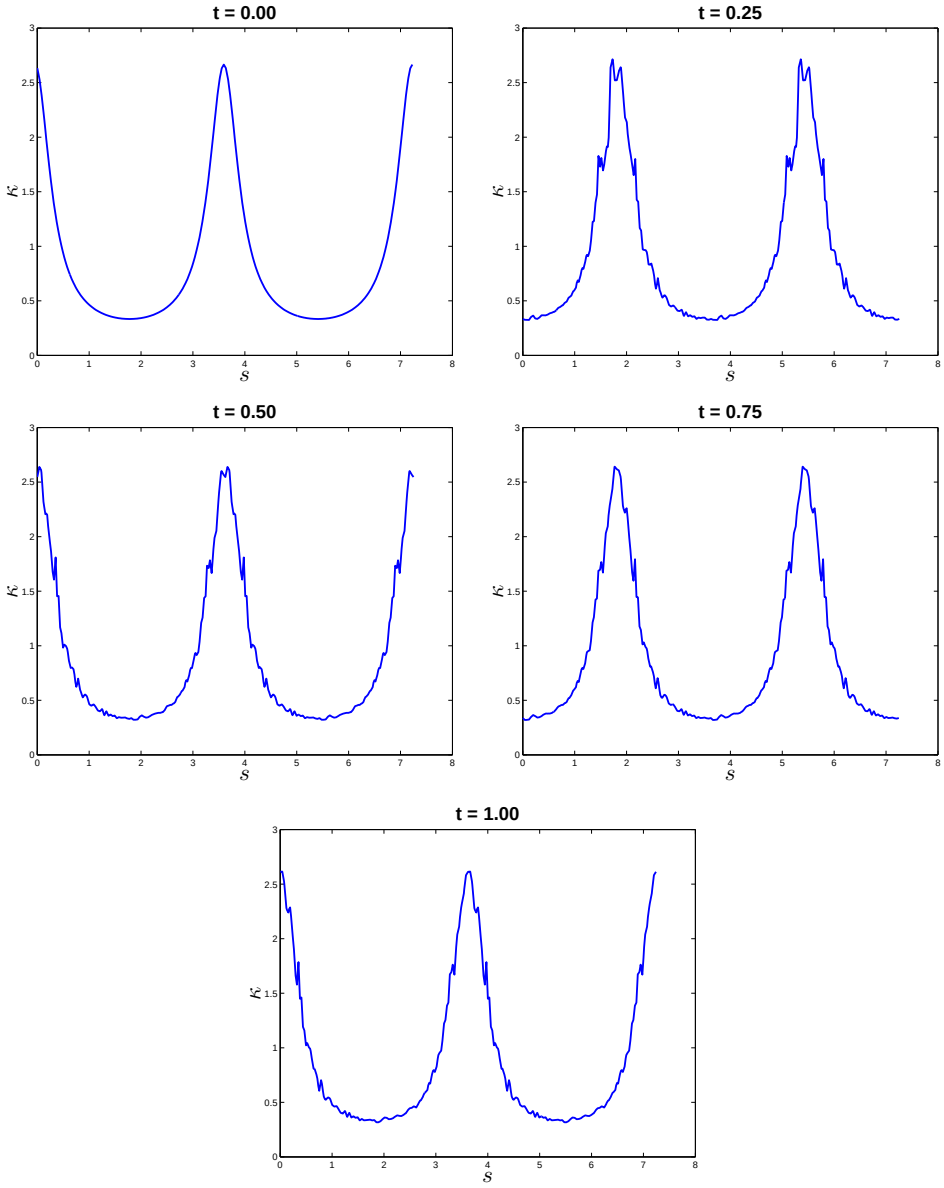


Figure 4: Curvature as functions of arc-length at different times. The curvature calculation is stable. Mesh size$=100\times100$, $\Delta t=h/4$.

Example 3.4. In this example the S-L method is used to compute a geometrical flow with the following anisotropic normal velocity for the interface:

$$\mathbf{u}=(2+\cos(3\theta+0.3))\mathbf{n}, \quad \cos\theta=\frac{\phi_x}{|\nabla\phi|}. \tag{3.5}$$

The corresponding Hamiltonian is non-convex, causing some Hamilton-Jacobi methods

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

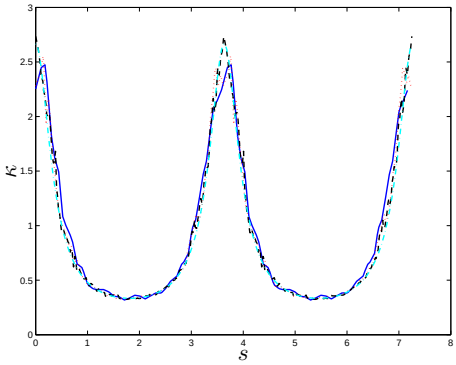


Figure 5: Mesh refinement study for the curvature at $t=1$. Mesh size $=40\times40$ (blue solid), 80×80 (red dotted), 160×160 (black dashed), the exact solution is plotted in cyan dash-dotted line. $\Delta t=h/4$.

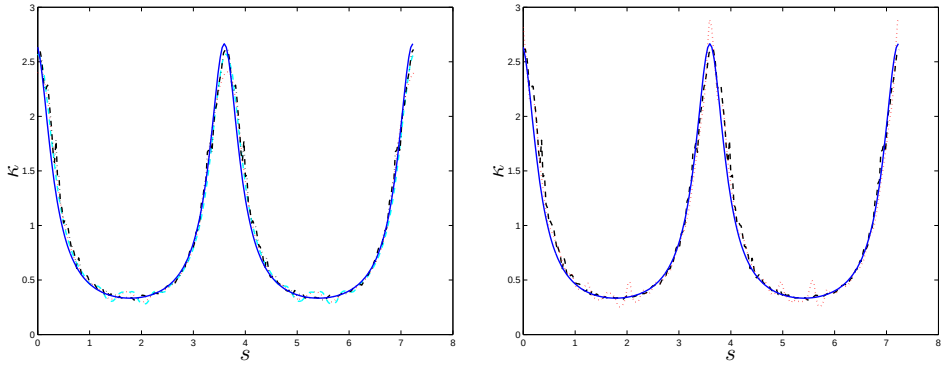


Figure 6: Left: S-L method (black dash-dotted), S-L-E method (cyan dashed line) and C-E method (red dotted) for the curvature calculation. Right: S-L method (black dash-dotted), the method of the second-order S-L scheme for the level-set convection coupled with the second-order S-L scheme for the re-initialization (red dotted). Exact solution is in blue solid. Mesh size $=100\times100$, $\Delta t=h/8$.

to break down.

Consider that three trefoil shapes are randomly placed in the computational domain $\Omega=[-3,3]\times[-3,3]$. The initial level-set function is given by $\phi=\min\{\phi_1,\phi_2,\phi_3\}$, where

$$\phi_1=\sqrt{\left(x-\frac{1}{2\pi}\right)^2+\left(y-\frac{1}{2\pi}\right)^2}-0.4-0.2\cos(3\theta_1),\qquad \theta_1=\tan^{-1}\frac{x-\frac{1}{2\pi}}{y-\frac{1}{2\pi}},$$

$$\phi_2=\sqrt{\left(x+\frac{2}{\pi}\right)^2+\left(y+\frac{2}{\pi}\right)^2}-0.3+0.1\cos(3\theta_2),\qquad \theta_2=\tan^{-1}\frac{x+\frac{2}{\pi}}{y+\frac{2}{\pi}},$$

$$\phi_3=\sqrt{\left(x+\frac{2}{\pi}\right)^2+\left(y-\frac{2}{\pi}\right)^2}-0.3+0.15\sin(3\theta_3),\qquad \theta_3=\tan^{-1}\frac{x+\frac{2}{\pi}}{y-\frac{2}{\pi}}.$$

Their evolution according to the normal velocity (3.5) involves considerable topological

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

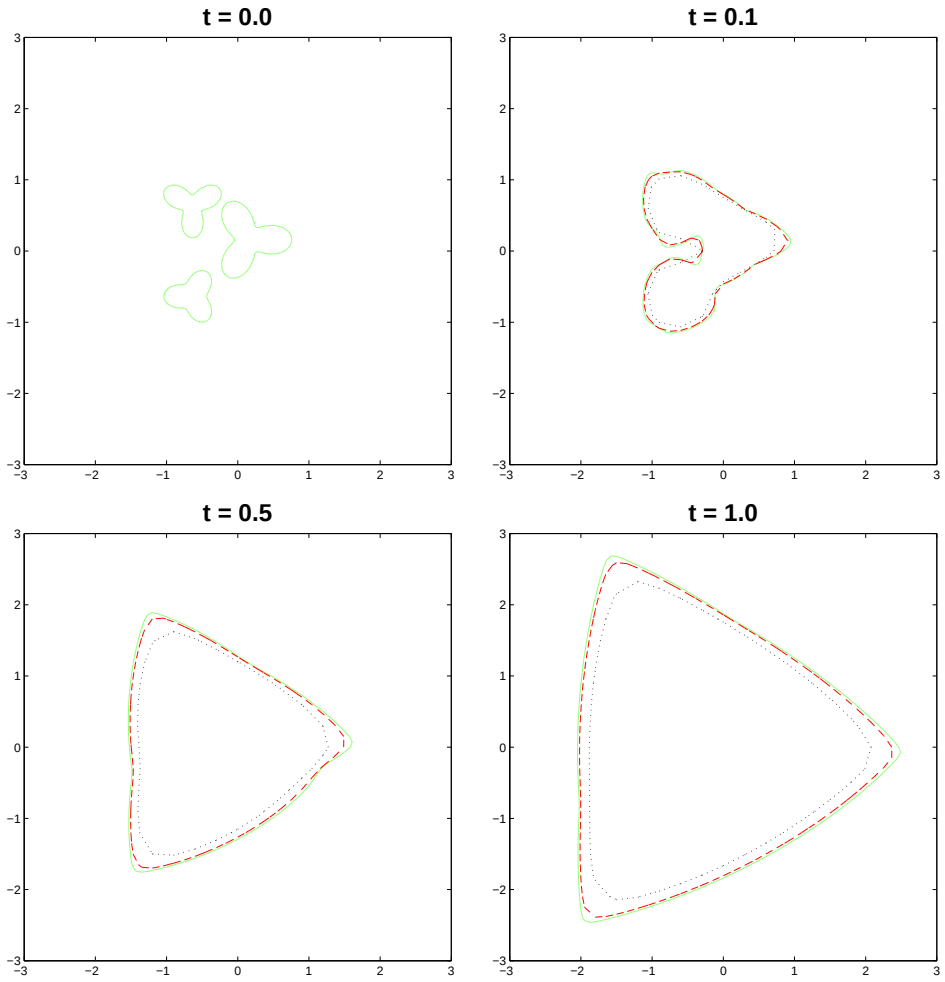


Figure 7: The triangular Wulff shape limit is computed using the S-L method with three mesh sizes: 320×320 (solid), 160×160 (dashed) and 80×80 (dotted). $\Delta t=h/8$. Numerical convergence is observed.

changes. The predicted asymptotic Wulff shape limit (e.g., [12]) is correctly computed in Fig. 7. In the computation, the following smoothing technique (also see [17]) for the velocity is used. After the velocity $\mathbf{u}$ is calculated at grid points in the tubes, set

$$\mathbf{u}_{ij}=\frac{1}{9}\Big(\sum_{i1=-1}^{1}\sum_{j1=-1}^{1}\mathbf{u}_{i+i1,j+j1}\Big). \tag{3.6}$$

3.3 The 3D examples

Example 3.5. In this example, we consider a Zalesak sphere which is rotated under the velocity $\mathbf{u}=2\pi(-y,x,0)^T$. The computational domain is $[-2,2]^3$. The drop is centered at

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

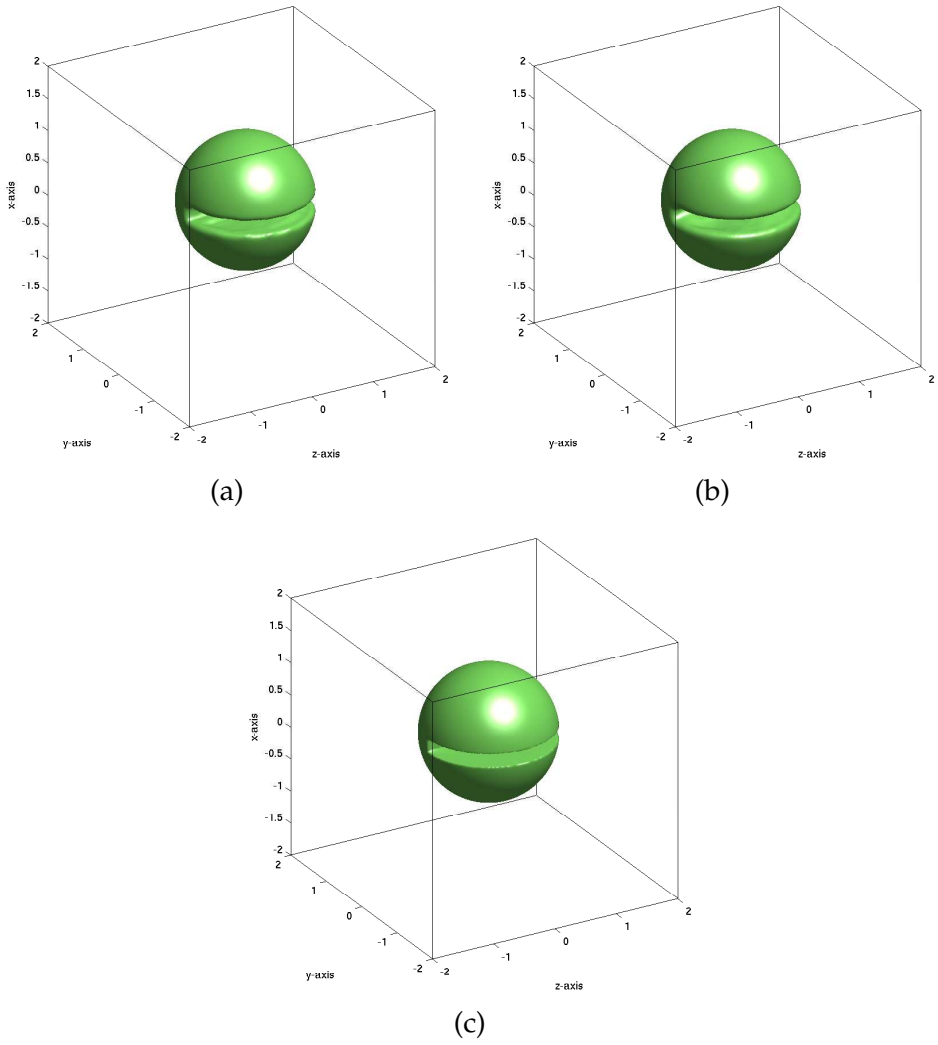


Figure 8: Zalesak sphere after one round of rotation by using the S-L method (a) and the S-L-E method (b); The exact solution is given in (c). Mesh size $= 160\times160\times160$, $\Delta t = h/2$.

the origin with radius 1, and the part being cut-out has width 0.2 and length 1.

The computed spheres together with the exact solution after one round of rotation are given in Fig. 8. Both the S-L method and the S-L-E method give reasonably good results. Somehow the C-E method does not work well for this problem, thus no result of the C-E method is presented here.

Next we compare the S-L method and the S-L-E method by plotting the cross-sections at $z=0$. Fig. 9 shows that the S-L method is better than the S-L-E method for two different time steps: $\Delta t = h/2$ and $h/4$.

The S-L method is also significantly faster than the S-L-E method. For example, when

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

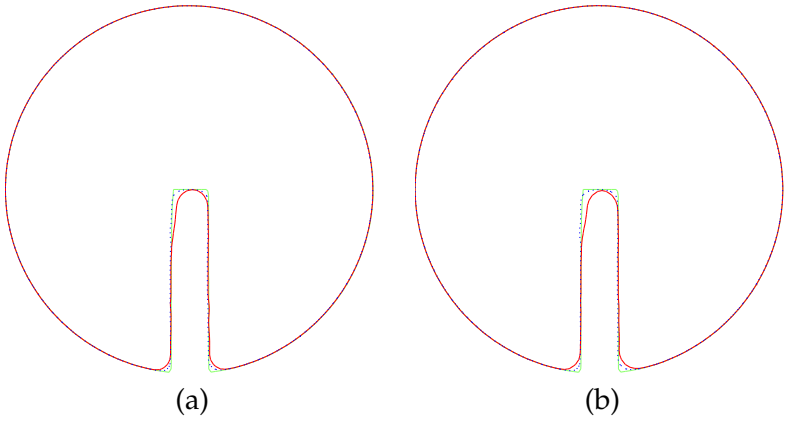


Figure 9: Comparison between the S-L method (blue dotted line) and the S-L-E method (red dashed line) for the cross-section at $z=0$. (a) $\Delta t=h/2$. (b) $\Delta t=h/4$. The green solid line is the exact solution. Mesh size $=160^3$.

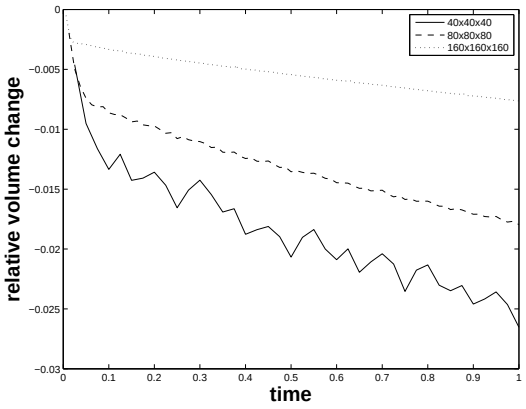


Figure 10: Mesh refinement study for volume loss of the Zalesak sphere for the S-L method. $\Delta t=h/4$.

the mesh size $=160^3$, $\Delta t=h/2$, the running times for the S-L method and the S-L-E method are 2797 seconds and 4956 seconds respectively.

Finally a mesh refinement study for the volume conservation of the S-L method is presented in Fig. 10. Mesh refinement improves the volume conservation significantly.

Example 3.6. We consider the Enright test problem [5, 6]. A 3D sphere is severally deformed by the following solenoidal velocity field:

$$\begin{cases} u(x,y,z,t)=2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)g(t), \\ v(x,y,z,t)=-\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)g(t), \\ w(x,y,z,t)=-\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)g(t), \end{cases} \tag{3.7}$$

where $g(t)=\cos(\pi t/3)$. The computational domain $\Omega=[0,1]^3$. The initial sphere is

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

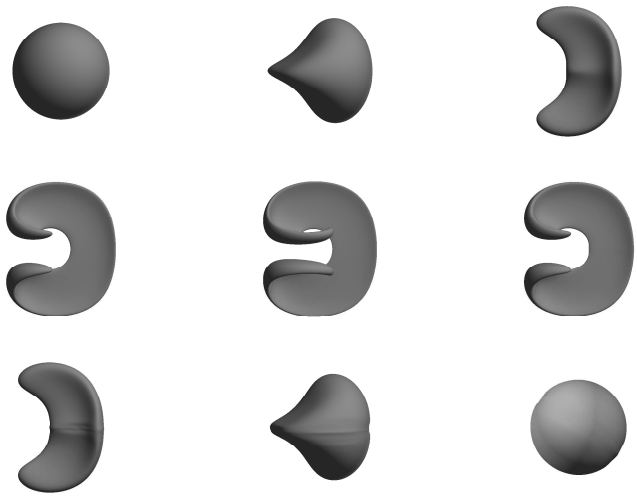


Figure 11: Deformation of drop at times $t=0.0,0.25,0.5,1.0,1.5,2.0,2.5,2.75,3.0$, from top to bottom and from left to right. In the end the total volume loss is about 3%.

centered at $(0.35,0.35,0.35)$ with radius 0.15. During the process, the drop undergoes severe distortion. Part of the drop becomes very thin at time $t=1.5$. After that it gradually returns to its original spherical shape. For this problem, uniform mesh would lead to large error due to the lack of resolution. It is prohibitively expensive to perform uniform mesh refinement due to the limit of computer storage. Our method is extended to the block-structured adaptive mesh (see [1,2]). In Fig. 11, a 4-level mesh with the finest mesh size $h_{\min}=1.5625D-3$ is used. The time step is $dt=h_{\min}/2$.

3.4 Simulations of incompressible two-phase flows

In the following we present simulations of bubble/droplet dynamics in incompressible two-phase flows in 2D.

The dimensionless Navier-Stokes equations governing the two-phase flows are following (see e.g., [26,27]):

$$\begin{cases} \rho(\phi)\Big(\dfrac{\partial \mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{u}\Big)=-\nabla p+\dfrac{1}{Re}\nabla\cdot\mathbf{T}+\dfrac{1}{ReCa}\mathbf{F}+\rho(\phi)\mathbf{g}, \\ \nabla\cdot\mathbf{u}=0, \end{cases} \tag{3.8}$$

where $\mathbf{T}=\mu(\phi)(\nabla\mathbf{u}+(\nabla\mathbf{u})^T)$ is the stress tensor. With the dimensionless surface tension being 1, $\mathbf{F}=-\kappa\mathbf{n}\delta(\phi)|\nabla\phi|$.

The dimensionless viscosity and density are given by

$$\rho(\phi)=H(\phi)+\lambda_\rho(1-H(\phi)),$$
$$\mu(\phi)=H(\phi)+\lambda_\mu(1-H(\phi)),$$

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

where H is the Heaviside function, λ_ρ, λ_μ are the density ratio and the viscosity ratio respectively. The other dimensionless parameters are the capillary number Ca and the Reynolds number Re.

In the numerical method, the S-L level-set method is used to capture the interfacial dynamics. As in [26, 27], the density weighted projection method is used to solve the N-S equations with the same smoothed Heaviside function and Dirac function. Then the resulting velocity is used to evolve the level-set function.

Example 3.7. We consider the deformation of a neutrally buoyant drop under a shear flow. The gravity force is neglected. Theoretically there is a critical capillary number beyond that the drop keeps stretching and no steady state exists. The computational domain $\Omega=[-5,5]\times[-2,2]$. The boundary condition is $\mathbf{u}=(y,0)^T$. Initially the drop is a unit circle centered at the origin. The drop deformation is measured by the maximum distance from the interface to its center. Comparison between the result of the S-L method and that of the C-E method (as in [26]) is plotted in Fig. 12 for several values of Ca. Two methods produce very similar results.

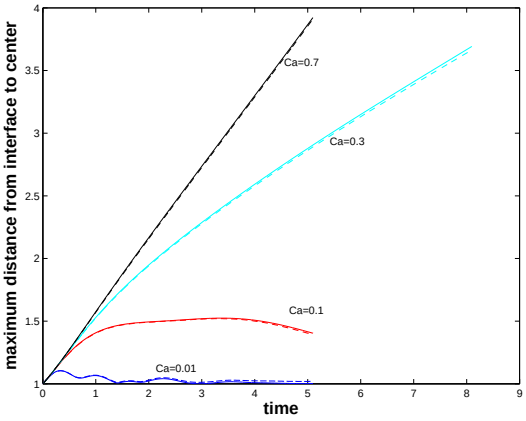


Figure 12: Comparison between the S-L method (dashed line) and the C-E method (solid line) for different values of Ca. $Re=10$, $\lambda_\rho=\lambda_\mu=1$. Mesh size $=320\times128$, $\Delta t=h/8$.

Example 3.8. We consider a bubble rising due to the buoyant force. The dimensionless gravity acceleration $\mathbf{g}=(0,-1)^T$. The computational domain $\Omega=[-2,2]\times[-2,5]$. The boundary condition is $\mathbf{u}=(0,0)^T$. Initially the bubble is centered at the origin with radius 1.

We take $Ca=0.2$, $Re=800$, and a very small density ratio $\lambda_\rho=0.001$. The effects of surface tension and the viscosity are relatively small. In this case the bubble is largely deformed and eventually pinches off. This phenomenon has been observed in the literature, see e.g., [22].

We refine the meshes with three different mesh sizes, the results are shown in Fig. 13. Numerical convergence is observed.

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

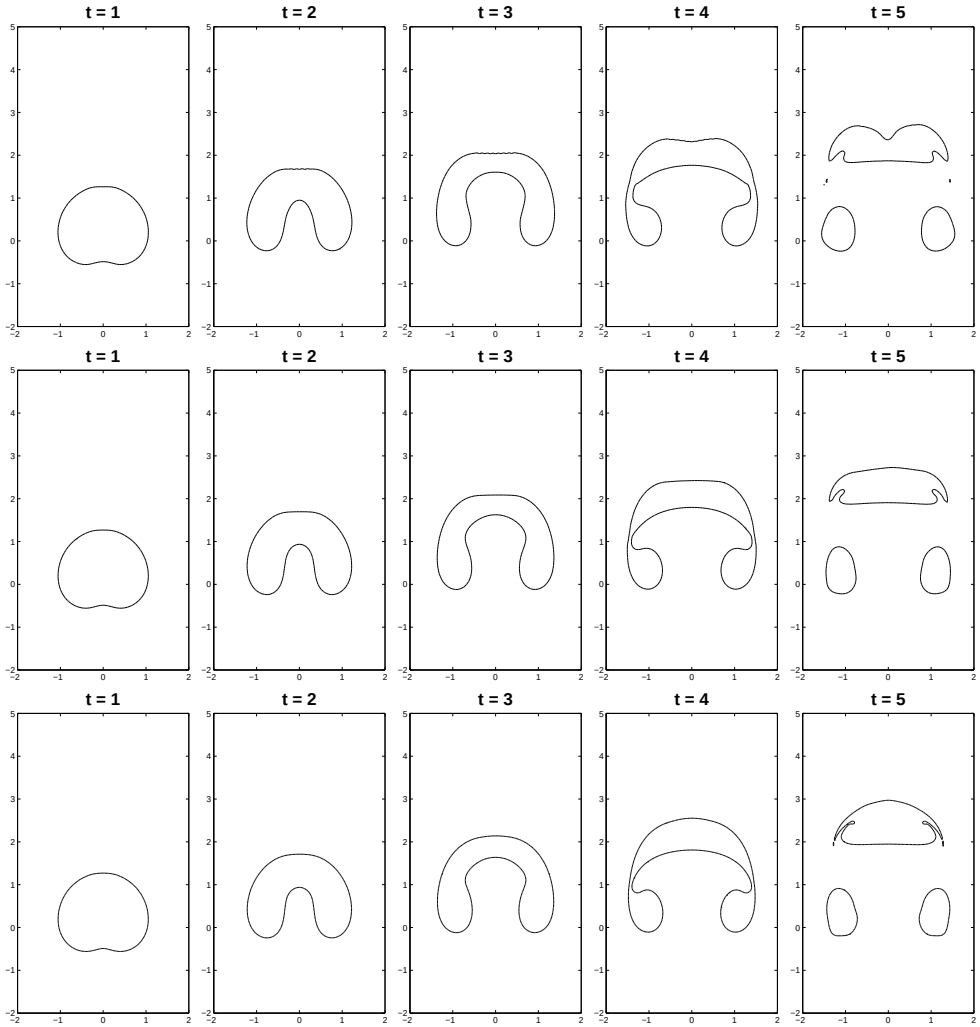


Figure 13: Bubble shapes at different times with three different mesh sizes: 64×112 (top panel), 128×224 (mid panel), 256×448 (bottom panel). $Ca=0.2$, $Re=800$, $\lambda_\mu=0.01$, $\lambda_\rho=0.001$. $\Delta t=h/32$.

The errors of the area conservation for the simulation is plotted in Fig. 14. It can be seen that the mesh refinement reduces the area loss significantly before the pinch-off. After the pinch-off, the numerical quadrature for the integral (3.1) may be less accurate for such complex geometry.

4 Conclusions and discussion

A simple semi-Lagrangian (S-L) method is proposed for the auxiliary level-set re-initialization equation. It consists of a first-order S-L scheme for the auxiliary re-

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

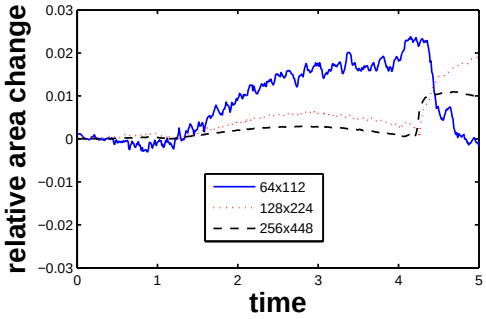


Figure 14: Mesh refinement study for the area conservation of the bubble. Same setting as in Fig. 13.

initialization equation, coupled with a projection technique for updating the level-set function at the irregular grid points. This S-L re-initialization scheme together with the standard second-order S-L scheme for the level-set convection equation forms an efficient S-L level-set method for interface motion. The efficiency of the S-L method has been demonstrated by extensive numerical examples including motion of interfaces with corners/kinks, topological changes, the Enright problem of large deformation, comparisons with the other methods, and simulations of two-phase flows.

Acknowledgements

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).

References

[1] M. J. Berger and J. Oliger, *Adaptive mesh refinement for hyperbolic partial differential equations,* J. Comput. Phys., 53 (1984), pp. 484–512.
[2] M. J. Berger and I. Rigoutsos, *An algorithm for point clustering and grid generation,* IEEE Trans. Syst. Man. Cybern, 21 (1991), pp. 1278–1286.
[3] 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.
[4] T. F. Dupont and Y. Liu, *Back and forth error compensation and corrections methods for removing errors induced by uneven gradients of the level set function*, J. Comput. Phys., 190 (2003), pp. 311–324.
[5] 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.
[6] D. Enright, R. Fedkiw, J. Ferziger and I. Mitchell, *A hybrid particle level set method for improved interface capturing,* J. Comput. Phys., 183 (2002), pp. 83–116.

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press

[7] D. ENRIGHT, F. LOSASSO AND R. FEDKIW, *A fast and accurate Semi-Lagrangian particle level set method,* Comput. Structures, 83 (2005), pp. 479–490.
[8] G.-S. JIANG AND D. PENG, *Weighted ENO schemes for Hamilton-Jacobi equations,* SIAM J. Sci. Comput., 21 (2000), 2126.
[9] R. LEVEQUE, *Finite difference methods for ordinary and partial differential equations,* SIAM, 2007.
[10] 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.
[11] S. OSHER AND J. A. SETHIAN, *Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations,* J. Comput. Phys., 79 (1988), 12.
[12] S. OSHER AND B. MERRIMAN, *The Wulff shape as the asymptotic limit of a growing crystal interface,* Asian J. Math., 1 (1997), 560.
[13] D. PENG, B. MERRIMAN, S. OSHER, H. ZHAO AND M. KANG, *A PDE-based fast local level set method,* J. Comput. Phys., 155 (1999), 410.
[14] A. SELLE, R. FEDKIW, B. KIM, Y. LIU AND J. ROSSIGNAC, *An unconditionally stable MacCormack method,* J. Sci. Comput., 35 (2008), pp. 350–371.
[15] C.-W. SHU, Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws, Springer-Verlag, 1998.
[16] A. STANIFORTH AND J. COTE, *Semi-Lagrangian schemes for atmospheric models–a review,* Monthly Weather Rev., 119 (1991), pp. 2206–2223.
[17] J. STRAIN, *Semi-Lagrangian methods for level set equations,* J. Comput. Phys., 151 (1999), 498.
[18] J. STRAIN, *Tree methods for moving interfaces,* J. Comput. Phys., 151 (1999), 616.
[19] J. STRAIN, *A fast modular semi-lagrangian method for moving interfaces,* J. Comput. Phys., 161 (2000), pp. 512–536.
[20] J. STRAIN, *A fast semi-lagrangian contouring method for moving interfaces,* J. Comput. Phys., 170 (2001), pp. 373–394.
[21] J. STRAIN, *Fast tree-based redistancing for level set computations,* J. Comput. Phys., 152 (1999), pp. 664–686.
[22] 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.
[23] 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.
[24] D. XIU AND G. E. KARNIADAKIS, *A semi-Lagrangian high-order method for Navier-Stokes equations,* J. Comput. Phys., 172 (2001), 658.
[25] D. XIU, S. J. SHERWIN, S. DONG AND G. E. KARNIADAKIS, *Strong and auxiliary forms of the semi-Lagrangian method for incompressible flows,* J. Sci. Comput., 25 (2005), pp. 323–346.
[26] 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.
[27] 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.
[28] S. T. ZALESAK, *Full multidimensional flux-corrected transport,* J. Comput. Phys., 31 (1979), pp. 335–362.
[29] M. ZERROUKAT, N. WOOD AND A. STANIFORTH, *Application of the parabolic spline method to a multi-dimensional conservative semi-Lagrangian transport scheme,* J. Comput. Phys., 225 (2007), pp. 935–948.

https://doi.org/10.4208/aamm.2015.m1305 Published online by Cambridge University Press