A Fast Level Set Method with Particle Correction on Adaptive Cartesian Grid

Zhu Wang*
Michigan State University, East Lansing, MI, 48824

Z.J. Wang†
Iowa State University, Ames, IA, 50011

The level set method, devised by Osher and Sethian in 1988, is a powerful approach for tracking moving interfaces and widely used in physics, fluid mechanics, chemistry, combustion, material science, image processing etc. During the past two decades, the level set method has been under significant development. Techniques of solving the level set equation, including high-order essentially non-oscillatory (ENO) schemes, the discontinuous Galerkin (DG) method, weight ENO (WENO) schemes and hybrid particle level set method, have been developed on both structure uniform mesh and unstructured mesh. In this paper, following the work by Strain, a second-order semi-Lagrangian level set method is developed and coupled with a new efficient particle correction algorithm on 2D adaptive Cartesian mesh. This solver is general and capable of solving 2D interface moving problems under different velocity fields, including passive transport fields, first-order and second-order geometric velocity. Through the particle correction algorithm, the semi-Lagrangian level set method can achieve much better mass conservation. Numerical results are presented and compared with other simulations to demonstrate the ability of the level set solver developed in this paper.

Nomenclature

ϕ	=	Level set function or signed distance function
$\vec{V}$	=	Velocity vector
$\vec{N}$	=	Unit normal of the interface
κ	=	Curvature of interface
$\vec{n}$	=	Unit normal of the face of control volume

1. Introduction

The level set method, introduced by Osher and Sethian [1] in 1988, is a powerful numerical technique for analyzing and computing motion of interfaces in two or three dimensional space. During the past two decades, numerical algorithms for solving level set equation have been tremendously advanced as summarized by Osher and Fedkiw [2, 3] and Sethian [4, 5]. At the same time, it has been applied greatly to various areas of science, such as physics, chemistry, fluid mechanics, combustion, image processing, material science, control science, computational geometry, computer-aided-design etc. Examples of interface problems involving external physics include compressible flow [6, 7], incompressible flow [8, 9], Stefan problems [10], kinetic crystal growth, epitaxial growth of thin films, vortex dominated flows, body fitted grid generation, image processing etc.

Various algorithms for the level set equations can be mainly categorized into three groups. The first group of level set solvers was focused on solving the Hamilton-Jacobi equation and they were extended from the schemes built for hyperbolic conservation laws. They include the monotone scheme with the finite difference (FD) method by Crandall and Lions [11, 12], non-oscillatory hyperbolic schemes by Osher and Sethian [1], high-order essentially

* Graduate student (wangzhu@msu.edu), 2555 Engineering Building, East Lansing, MI 48824
† Associate Professor of Aerospace Engineering (zjw@iastate.edu), Associate Fellow of AIAA

non-oscillatory (ENO) schemes by Osher and Shu [13], Lafon and Osher [14], discontinuous Galerkin (DG) finite element method by Hu and Shu [15], Petrov-Galerkin method by Barth and Sethian [16], high order weighted essentially non-oscillatory schemes (WENO) by Jiang and Peng [17], high-order WENO schemes for triangular mesh by Zhang and Shu [18], central WENO schemes by Levy, Nayak, Shu and Zhang [19]. Although these algorithms can accurately simulate the motion of the interfaces in some cases, most of them were carried out on structured Cartesian meshes only. Since the mesh is uniform everywhere, it is not possible to have higher resolution for the grids near the interface. This may be very costly for some applications. There also have been a few attempts to use unstructured triangular meshes for the level set equation such as [16, 18]. The unstructured grids are much more flexible than uniform meshes in handling complex geometries and performing grid adaptation locally. Adaptive Cartesian grids are advocated in this study. With the help of a tree-based data structure, the generation of the adaptive Cartesian grids can be very efficient and interface-based adaptations can be performed easily.

The second group of algorithm is the finite volume (FV) level set method which was newly developed. A flux-based finite volume method was proposed by Frolkovic [20] on uniform mesh in 2003. And in 2004 Zhu Wang and Z.J. Wang [21] developed a finite volume level set method on 2D adaptive Cartesian mesh realized by a least-squares data reconstruction algorithm. Through the automatic interface-based grid refinement and adaptation, the efficiency and accuracy are further improved near the interface. The two groups of level set methods described above are Eulerian methods, which solve the level set function on some fixed background grids. Although Eulerian level set algorithms mathematically define the merging and breaking of interface well, they are numerically diffusive. In another word, Eulerian algorithms cannot conserve mass well due to the loss of information in under-resolved areas.

The third group of level set algorithms is the Lagrangian level set method. In 1999, a semi-Lagrangian scheme was developed by Strain in [22] to simulate the interface on 2D Quadtree-mesh, and it was modulated into a fast semi-Lagrangian level set method by Strain [23] in 2000. Through a fast redistancing [24] and adaptive contouring algorithm [25], the motion of interface can be simulated accurately and efficiently. One drawback of this method is that it is too intricate to extend to three dimensional problems. In order to conserve the mass or volume better, a hybrid level set method was developed by Enright et al. [26] in 2002. This method coupled the fifth-order WENO scheme for the level set equation with third-order Runge-Kutta scheme for moving the massless particles. This method dramatically improved the property of mass or volume preserving through a particle correction algorithm. In this method, the quantity of particles can be very large by distributing the particles randomly in a band of area around the interface. Thus the computation for moving particles could be time-consuming. In 2005 Enright et al. [27] employed first order semi-Lagrangian level set method coupled with particle correction algorithm. Similar results as in [26] were achieved while it is more efficient since only first order scheme was used to solve the level set equation. Even newer level set method such as hybrid particle level set on Octree adaptive Cartesian grids is under developing as stated in [28].

In this paper, following the work by Strain [23], a second-order semi-Lagrangian level set solver is developed on 2D adaptive Cartesian mesh. This solver is more general and it can maintain the signed distance function accurately through a fast redistancing algorithm. The semi-Lagrangian method is capable of solving 2D interface moving problems under different velocity fields, including passive transport fields, first-order and second-order geometric velocity. In order to further improve the accuracy and properties of conserving volume or mass, a particle correction algorithm has been developed and coupled with the semi-Lagrangian level set method.

The remainder of this paper is organized as follows. Following introduction, the generation of adaptive Cartesian grids is introduced in Section 2. Then, the numerical algorithms of semi-Lagrangian level set method and particle correction are presented in Section 3. After that, numerical examples are given to verify the accuracy and demonstrate the mass conservation properties. Finally concluding remarks based on the current study are given in Section 5.

2. Adaptive Cartesian Grid Generation

A quadtree-based adaptive Cartesian grid generator has been developed for the lever set solver based on the Object-Oriented Programming (OOP) technique. The OOP technique makes the grid generation, operation of searching neighboring cells and computation of signed distance very efficient through the recursive strategy. The detailed definition and background of quadtree-based Cartesian mesh can be found in [22]. In this section, the presentation of the 2D interface, the computation of distance function and generation of quadtree-based adaptive Cartesian mesh are presented.

2.1. Interface Representation

In this paper, two dimensional interfaces (open curves intersecting the boundary or closed curves) are under the consideration. The interfaces can have more than one curve, and each separate curve is called a component of the interfaces. Although the level set method implicitly represents the moving interfaces with the zero set of level set function, it is a better idea to create a front data structure to store the interface explicitly. By borrowing this from front-tracking method, the calculation of sign distance to the interface, the Lagrangian particle correction and post processing such as checking volume conservation can be done easily and efficiently.

A data structure that consists of points connected by elements (line segments) is employed in this study. Both particles (the ending points) and elements are stored in linked lists, which contain the pointers to the previous element and the next element. The order of all the elements is completely arbitrary hence it is not necessary to store the elements following the actual order of the interface. The advantage of using linked list is that it makes the addition and removal of particles and elements very simple. The data structure is shown in Figure 1. This data structure can be extended to 3D with similar mechanism while the elements are substituted by linear triangle, which have three ending points and three adjacent elements.

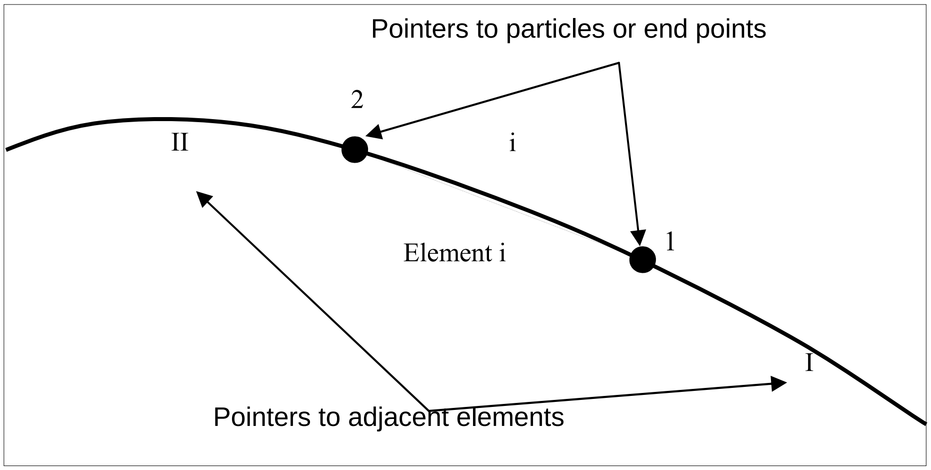
Pointers to particles or end points
2
II
i
Element i
1
I
Pointers to adjacent elements

Figure 1. Data Structure for a 2D Interface

The resolution of the interface depends on the discretization described above. Obviously the interface is more accurate when the elements are finer. In this paper, an object class named *interface* is developed and local properties of the interface such as outward normal, curvature can be calculated efficiently and accurately by coupling with the adaptive grid strategy.

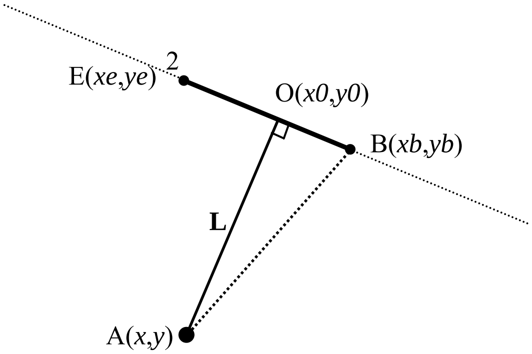

Figure 2. Distance from any Point to a Linear Element

2.2. Signed Distance Computation

The distance from any point to a linear segment can be easily calculated with a projection method and dot product, as illustrated in Figure 2. The distance L can be calculated in the following way:

(1) Calculate the square of length *BE*

$$BE^2 = (x_e - x_b)(x_e - x_b) + (y_e - y_b)(y_e - y_b) . \tag{1}$$

(2) Find the projection of *vector BA* on *vector BE* by calculating their dot production

$$dot = \overset{\rightarrow}{BA} \cdot \overset{\rightarrow}{BE} = (x_e - x_b)(x - x_b) + (y_e - y_b)(y - y_b) . \tag{2}$$

(3) Calculate the parameter θ which should belong to [0, 1] by

$$\theta = \max[0, \min(1, dot / BE^2)] . \tag{3}$$

(4) Find the closest point O on vector BE by

$$\begin{array}{l} x_0 = x_b + \theta(x_b - x_e) \\ y_0 = y_b + \theta(y_b - y_e) \end{array} . \tag{4}$$

(5) Find the distance L to the closest point O on vector BE

$$\mathrm{L} = \sqrt{(x - x_0)^2 + (y - y_0)^2} . \tag{5}$$

The signed distance function for any point (x, y) can be directly calculated by finding the distance between that point and the closest linear segment, which is described above. Then the sign can be decided by checking the outward normal of adjacent elements.

The sign of ϕ can be fixed efficiently by defining an outward unit normal vector for every element of interface Γ and checking which side of the closest element $\vec{x}$ lies on. Two special cases of fixing the sign of ϕ need to be considered carefully, and they are shown in (a) and (b) of Figure 3 for intersecting elements which form sharp corners. In the first case, O is the closest point hence both elements are nearest segments to point $\vec{x}$. The point $\vec{x}$ is outside of element *i* and inside of element *i*+1, while the sign of ϕ is positive. For the convex case (a), the sign is negative only when the point $\vec{x}$ falls inside of both element *i* and element *i*+1, otherwise the sign is positive. For the concave case (b), the sign is positive only when $\vec{x}$ falls outside of both element *i* and element *i*+1, otherwise the sign is negative. It is noted that directly computing the signed distance is extremely time-consuming if one loops all over the segments. But with the help of distance tree introduced in [24], it can be efficiently computed in (*N*log*N*) time on adaptive Cartesian grids with quadtree-based data structure. Interested readers can refer to the fast redistancing algorithm introduced in [24].

2.3. Quadtree-Based Adaptive Cartesian Mesh

Quadtree-based adaptive Cartesian grids have been developed for the lever set solver based on the Object-Oriented Programming (OOP) techniques. The OOP techniques make the grid generation, operation of searching neighboring cells of all the sides and computation of signed distance very efficient.

A special data structure, called cell-based tree, is used for generation of adaptive Cartesian mesh. The basic unit of this data structure is cell (or called node) and all the cells are arranged in a tree. In 2D, regularly each node or cell at level *l* has one parent at level *l*-1 (except root) and four children cells at level *l*+1 by splitting along both *x* and *y* direction, hence it leads to a name of *Quadtree* mesh. Note that a root is the cell which has no parent and its level *l* is set to be 0, and a leaf cell is a cell which has no children. A simple level 3 quadtree mesh and its tree structure are depicted in Figure 4 and it has 23 points and 13 leaf cells.

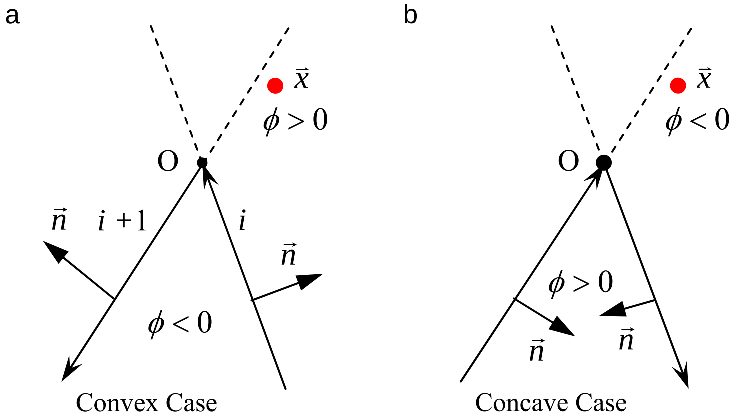

Figure 3. Fixing the Sign of Level Set Function:
(a) Convex Interface; (b) Concave Interface

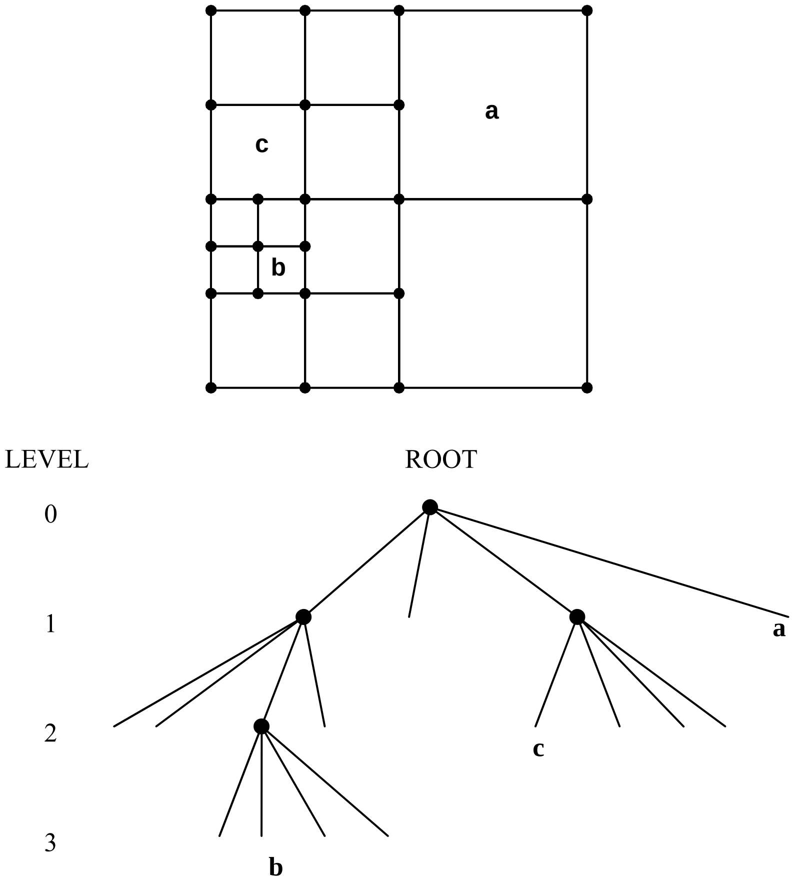

Figure 4. Level 3 Quadtree Mesh and the Tree Structure

The adaptation including refinement and coarsening can be easily realized by splitting a cell or simply delete its children on a Quadtree-based adaptive Cartesian grids. Following some adaptation criteria, desired mesh with high quality can be generated to satisfy different requirements.

A graded (also called balanced) adaptive Cartesian mesh is generated under several different criteria of splitting from a root cell, which represents the complete computational domain. These criterions are:

(1) Split any cell uniformly until the size of each leaf cell is less than the maximum size (minimal level).

(2) Split any cell whose size is larger than the distance between cell centroid and the interface.

(3) Split any cell whose size is large than the minimum size (maximum level) in a band nearby.

(4) Smoothing the entire mesh to ensure that neighbor cells differ in size by no more than a factor of 2.

Through recursive quad-tree based subdivisions, one can efficiently and automatically generate fine cells near particular flow features such as a material interface which can have multiple components. If one is interested in the interface only, the level set equation can be solved locally and the Narrow Band method is preferred because of its efficiency. With the Cartesian adaptive mesh, the minimum grid resolution is achieved near the interface within the narrow band thus high accuracy is guaranteed there.

3. Numerical Method

There are several important techniques for the semi-Lagrangian level set method in [24]. They are first order and second order numerical schemes, Whitney extension of velocity, redistancing algorithm for maintaining the signed distance function, contouring algorithm for the zero level set. The details of these algorithms can be found in [24]. In this section the first order and second-order semi-Lagrangian schemes are reviewed firstly, then the particle correction algorithm for the semi-Lagrangian level set method are described. For passive transport velocity field, the particle level set method is applied to achieve better mass conservation and capture very thin filaments.

3.1. Semi-Lagrangian Level Set Method

The level set equation is given by

$$\frac{\partial \phi}{\partial t} + \vec{V} \cdot \nabla \phi = 0 . \tag{6}$$

Starting from a Lagrangian view, one can observe that the level set equation propagates level set function ϕ along the characteristic curve $\vec{x} = \vec{s}(t)$ which is defined by

$$\dot{\vec{x}}(t) = \dot{\vec{s}}(t) = \vec{V}(x,t) \tag{7}$$

with the initial condition given by $x_0 = s(t_0)$. Hence the value of the level set function ϕ at time step n+1 ($t + dt$) can be computed by finding the characteristic curve passing through point $(\vec{x}, t + dt)$ and following it backwards to some previous point $(\tilde{x}, t)$, where the value of ϕ is already known at the time t, then setting $\phi^{n+1}(\vec{x}, t + dt) = \phi^{n}(\tilde{x}, t)$. This procedure can be numerically realized by solving the characteristic ordinary difference equation (ODE) (7). Once the velocity field $\vec{V}(x,t)$ is extended from the interface or computed by external physics, it is easy to locate the off-grid point $(\tilde{x}, t)$, on which the signed distance function $\phi(\tilde{x}, t)$ can be evaluated by numerical interpolation or directly comuting the distance to the discretized interface with the help of an auxiliary distance tree. The simplest way of calculating $\tilde{x}$ is to use the first order approximation

$$\tilde{x} = \vec{x} - dt \cdot \vec{V}(\vec{x}, t), \tag{8}$$

and linear interpolation of $\phi(\tilde{x}, t)$ can calculated by bilinear interpolation

$$\phi(\tilde{x},t)=(1-\alpha)(1-\beta)\phi_{00}+\alpha(1-\beta)\phi_{10+}(1-\alpha)\beta\phi_{01}+\alpha\beta\phi_{11}\,, \tag{9}$$

where ϕ_{00}, ϕ_{01}, ϕ_{11}, ϕ_{10}, α and β are defined in Figure 5.

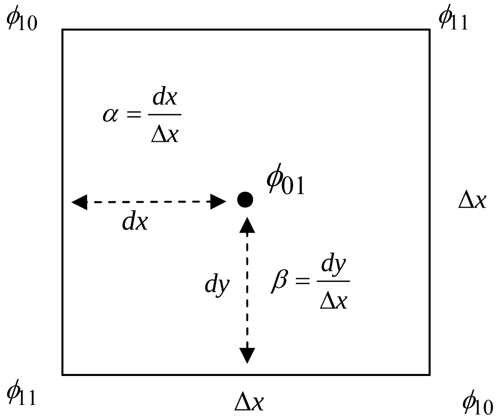

Figure 5. Bilinear Interpolation of off-grid Value

The first-order semi-Lagrangian scheme is called Courant-Issaacson-Rees (CIR) scheme [29], which was developed in [22] and applied in [23-25]. The formulation of CIR scheme is

$$\phi(\vec{x},t+dt)=\phi\left(\vec{x}-dt\cdot\vec{V}(\vec{x},t),t\right). \tag{10}$$

In order to obtain second-order accuracy in time, a CIR predictor (10) and a second-order trapezoidal corrector are combined together. Firstly a predicted function $\tilde{\phi}$ is acquired by using the CIR scheme, i.e.,

$$\tilde{\phi}(\vec{x},t+dt)=\phi(\tilde{x},t)=\phi\left(\vec{x}-dt\cdot\vec{V}(x,t),\ t\right). \tag{11}$$

Then the extensive velocity $\tilde{\tilde{\vec{V}}}$ is evaluated from the predicted function $\tilde{\phi}(\vec{x},t+dt)$ at time *t* + *dt*. The second-order semi-Lagrangian scheme is given by

$$\phi^{n+1}(\vec{x},t+dt)=\phi^{n}\left(\vec{x}-\frac{dt}{2}\cdot\vec{V}(\tilde{x},t)-\frac{dt}{2}\cdot\tilde{\tilde{\vec{V}}}(\vec{x},t+dt),\ t\right). \tag{12}$$

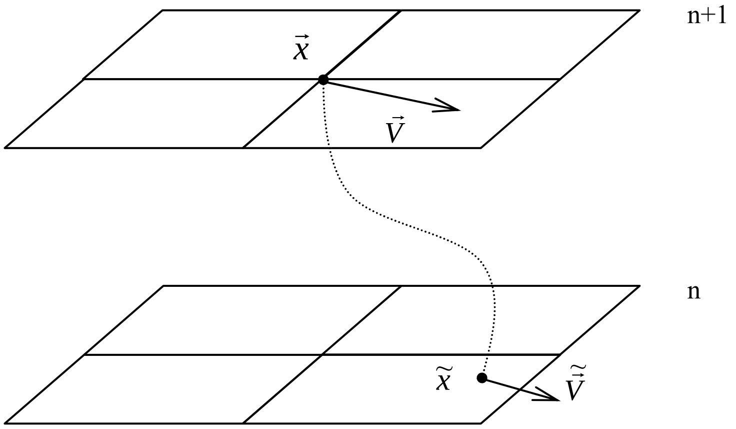

Figure 6. Fast Semi-Lagrangian Level Set Method

The fast semi-Lagrangian level set method is illustrated in Figure 6. The CIR and the second-order semi-Lagrangian schemes are explicit yet unconditionally stable and satisfy the CFL condition. The solutions can converge with large time step ($dt = O(\Delta x)$) even for parabolic problems where the velocity is depended on curvature terms. Compared to second-order semi-Lagrangian scheme, the CIR scheme is only first-order accurate in time and sometimes dissipative. In this study, both the CIR scheme and second-order scheme are tested and numerical results shown in this paper are computed by the second order semi-Lagrangian scheme if there is no extra explanation.

3.2. Particle Correction Approach

Particle level set method was firstly developed by Enright et al [26], and in this method a layer of Lagrangian marker particles near the interface are randomly distributed near the interface. The particles are then moved with high order Runge-Kutta schemes and used to correct the interface in under-resolved area and reinitialize the signed distance function. This hybrid method couples the H-J WENO scheme and particle correction algorithm, and it dramatically improves the conservation of mass.

Based on work by Enright et al [26], Enright, Losasso and Fedkiw [27] developed a fast and accurate semi-Lagrangian particle level set method, in which the level set method are solved by the first order CIR scheme and the interface same particle correction algorithm are employed. Since the CIR scheme is much simpler than high order WENO H-J scheme, the method is more efficient.

Compared to these two methods, a more efficient particle correction algorithm is developed in this paper. The particles are treated as the vertices of linear elements, which is stored with a linked list data structure described in section 2.1. Much fewer number of particles are needed to move with high order Runge-Kutta method compared to the above two methods since the particles are only placed on the interface itself (the particles will represent the interface). By using the linked list, removing unnecessary particles and adding new particles are quite simple. With the fast redistancing algorithm to evaluate the signed distance function, the signed distance can be maintained accurately and efficiently.

The algorithms of particle level set method in this study are:

(1) Evaluate the signed distance function with the first-order CIR scheme;

(2) Move the particles with second or three-order Runge-Kutta schemes by using the extensive velocity field;

(3) Use the particles and elements stored in the linked list to correct the level set function locally (within a narrow band of interface);

(4) Update the particles and elements and check the size of elements to decide whether adding particles or removing particles.

The particle correction algorithms are tested in this study. The numerical results are excellent for passive transported velocity field and comparison to regular semi-Lagrangian level set method without particle correction will be shown in next section.

4. Numerical Results

Versatile motions of interfaces under passive transport velocity field, first order and second order geometric velocity field are simulated with the semi-Lagrangian level set method to show its ability. The particle correction algorithm is also tested for passive transport velocity field. The numerical results are compared to regular semi-Lagrangian level set method without particle correction.

4.1. Zalesak's Disk Rotation Problem

The rigid body rotation of Zalesak's disk (notched circle in 2D) under a constant vorticity velocity field [30] is employed as the first example to demonstrate the ability of semi-Lagrangian level set method in conserving mass. This case has been studied by Sussman [9], Enright and Fedkiw [26], and Enright, Losasso and Fedkiw [27] for the purpose of improving mass conservation of their numerical schemes. It has been known that many classic algorithms for the level set equations are diffusive and tend to lose mass while the hybrid level set method in [26, 27] can achieve much better mass conservation.

The initial interface and velocity field over the computational domain are shown in Figure 7. The notched circle centers at (50, 75) with radius 15 and the slot has a width of 5 and length of 24.79. And the velocity is given by

$$\begin{aligned} u &= (\pi/314)(50-y) \\ v &= (\pi/314)(x-50) \end{aligned}. \tag{13}$$

The period of rotation is 628.

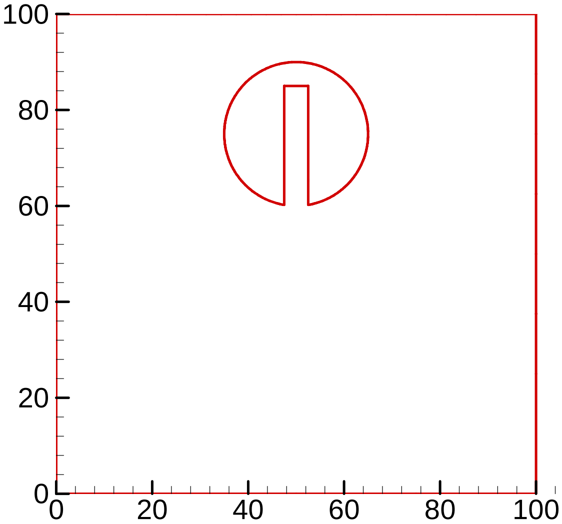

Figure 7. Initial Zalesak's Disk

For this case, the adaptive Cartesian grid technology and the second-order fast semi-Lagrangian scheme are applied to Zalesak's disk rotation problem. The numerical results acquired by linear contouring, adaptive contouring and particle correction under different levels of adaptive Cartesian grids are shown and compared next.

Figure 8 shows the numerical results after one period under level 6, 7, 8 and 9 adaptive Cartesian grids with linear contouring. The second-order semi-Lagrangian level set scheme is used for all simulations. From the figure, it is noticed that by refining the mesh with higher level quadtree mesh the accuracy of the numerical results and mass conservation are improved, while the solution near the under-resolved area near the sharp corners of the interface is still diffusive.

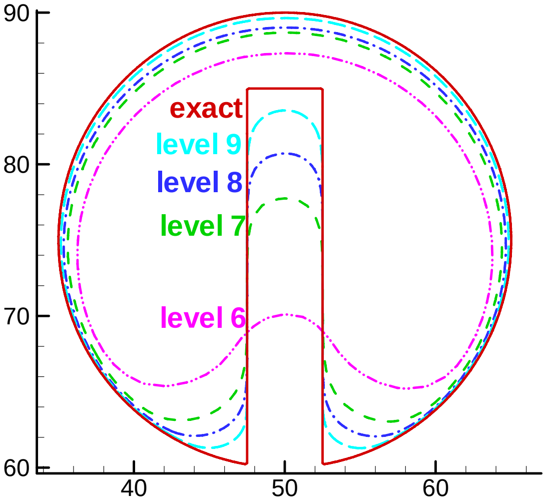

Figure 8. Zalesak's Disk Problem: T=628, Level 6, 7, 8 and 9 Adaptive Cartesian Grid, Linear Contouring

Figure 9(a) shows the numerical results after one period on the level 7 adaptive Cartesian grids with level 3 adaptive contouring techniques. The accuracy and mass conservation are dramatically improved compared to those of linear contouring strategy. The diffusive property of the level set method is greatly reduced and limited within one or two finest cells at the under resolved area, which can be seen in Figure 9(b).

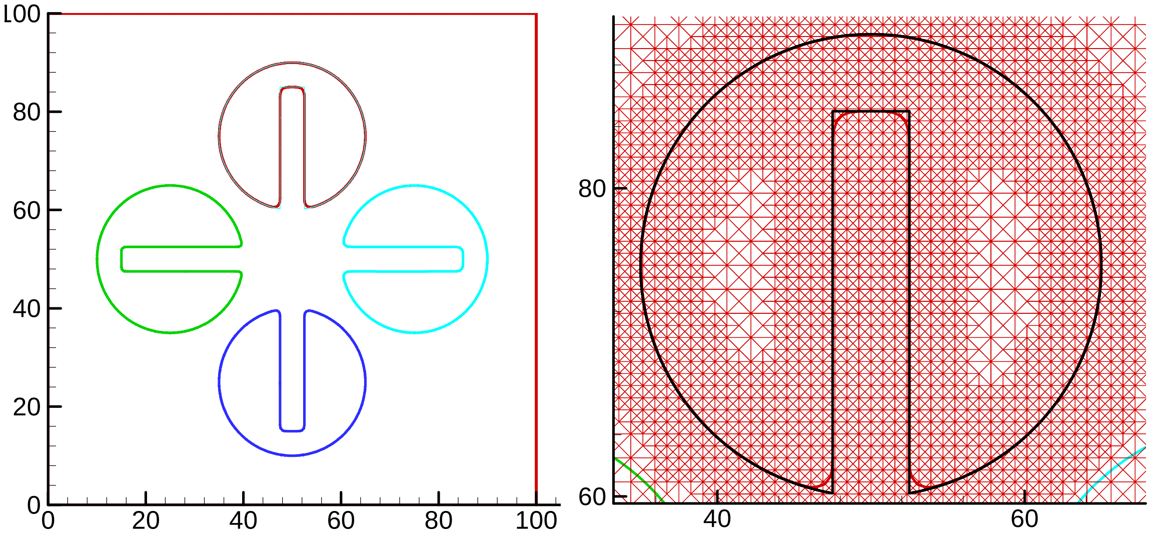

Figure 9. (a) Level 7 Adaptive Cartesian Grid, Level 3 Adaptive Contouring, T=0, 628 (157)
(b) Diffusive Property of Level Set Method: Interface near the Sharp Corners

Figure 10a shows the numerical results after two periods respectively on level 7 adaptive Cartesian grids. In this simulation, the particle correction algorithm is applied. Figure 10b shows the numerical results and grids after two periods on a level 8 adaptive Cartesian grid. For these simulations, the interface is discretized with 3000 linear elements and the particles (vertices of the elements) are moved with third-order Runge-Kutta scheme with the predefined external velocity field. Based on the numerical results, the interface is captured very sharply with excellent conservation property. The mass conservation error is very small (less than 0.01% after two rotation periods on level 8 adaptive Cartesian grids). In order to compare these methods, the volume loss and CPU time with linear and adaptive contouring methods are shown in Table 1 while results with particle level set method are shown in Table 2.

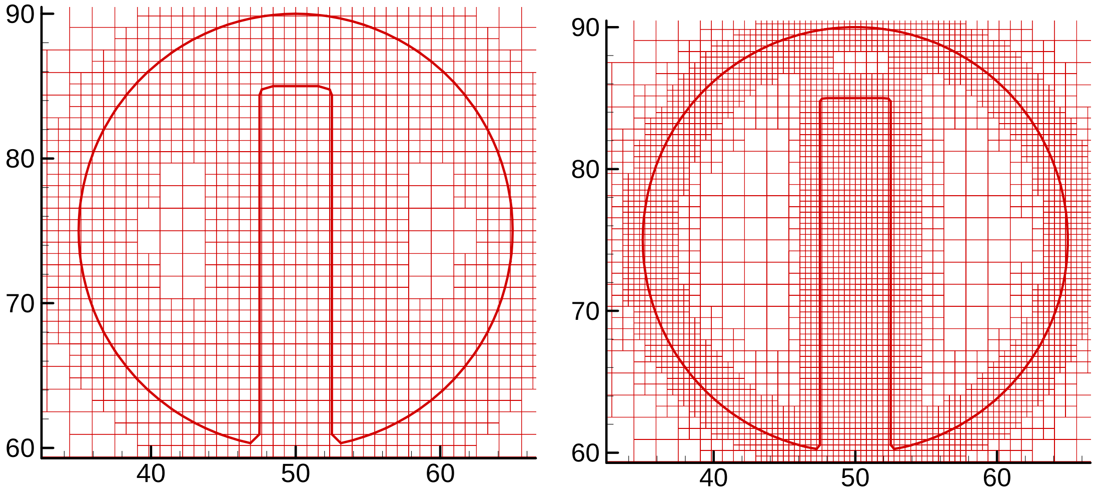

Figure 10. (a) Level 7 Adaptive Cartesian Grid, Particle Level Set Method, T=1256
(b) Level 8 Adaptive Cartesian Grid, Particle Level Set Method, T=1256

Table 1. Zalesak's Disk: Results with Linear and Adaptive Contouring

	Adaptive Cartesian Grid	Volume	Volume lost	CPU Time/step
	Exact Solution	582.20	--	--
One Period (T=628)	Level 6 + Linear Contour	471.52	19.01%	0.052
	Level 7 + Linear Contour	529.78	9.00%	0.176
	Level 8 + Linear Contour	550.89	5.38%	0.378
	Level 9 + Linear Contour	569.43	2.19%	0.697
	Level 6 + Adaptive Contour	579.17	0.52%	0.773
	Level 7 + Adaptive Contour	579.86	0.40%	2.434
	Level 8 + Adaptive Contour	581.17	0.18%	3.520

Table 2. Zalesak's Disk : Semi-Lagrangian + Particle Correction

	Adaptive Cartesian Grid	Volume	Volume lost	CPU Time/step
	Exact Solution	582.20	--	--
One Period (T=628)	Level 6	581.94	0.045%	0.984
	Level 7	582.12	0.014%	1.545
	Level 8	582.16	0.007%	2.143
Two Period (T=1256)	Level 7	582.12	0.014%	--
	Level 8	582.16	0.007%	--

4.2. Single Vortex

Zalesak's disk rotation problem is used to demonstrate the ability of our method to conserve mass. While the single vortex problem is used to test the ability of the semi-Lagrangian level set method to accurately resolve thin filaments in stretching and tearing flows. An example of such flows was introduced by Bell et al [31] and called "single vortex" problem. This case has been studied with the particle level set methods in [26, 27].

The velocity field is defined by the stream function

$$\Psi = -\frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y). \tag{14}$$

Hence the velocity field is computed by

$$\begin{aligned} u &= \frac{\partial\Psi}{\partial y} = -\sin^2(\pi x)\sin(2\pi y) \\ v &= -\frac{\partial\Psi}{\partial x} = \sin^2(\pi x)\sin(2\pi y) \end{aligned} \quad . \qquad (15).$$

The initial interface and velocity field are shown in Figure 11. The interface is a circle center at (0.5, 0.75) and the computational domain is 1 by 1. When time increases, the interface keeps stretching into a long and slim filament-like interface. Classic level set method will fail to capture the interface at a later time because of diffusive property. Adaptive Cartesian grids technologies are applied in this example and the finest cells with highest level are generated around the interface. Higher accuracy can be achieved than uniform mesh with comparable number of cells and it is much more efficient than using finest cell everywhere. By coupling the particle correction algorithm, the interface can be captured very accurately even the width of interface is less than the scale of finest Cartesian cells. After *t* = 1, the interface stretches longer and slimmer very fast. And Figure 12 shows the numerical results at *t* = 3 on level 8 adaptive Cartesian mesh with level 3 adaptive Contouring. The end of interface is too slim and can not be resolved thus it is smeared. The results start getting worse after *t* = 3 since the filaments are too slim to be resolved even with the level 10 adaptive Cartesian grids, which can achieve a resolution of 1024 by 1024 uniform mesh around the interface.

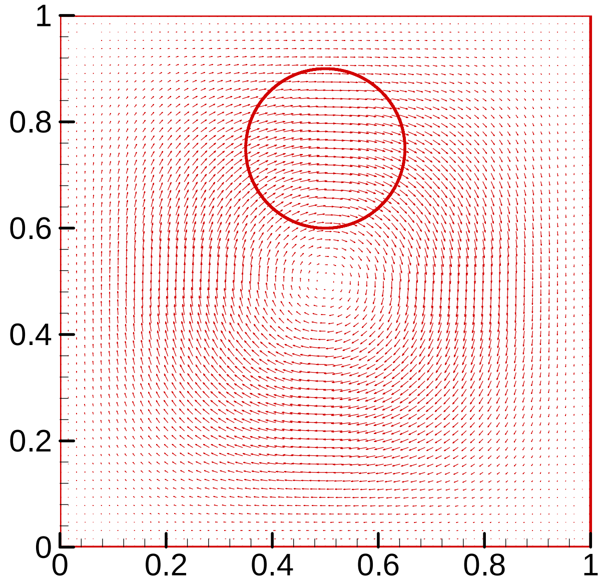

Figure 11. Initial Interface and Single Vortex Velocity Field

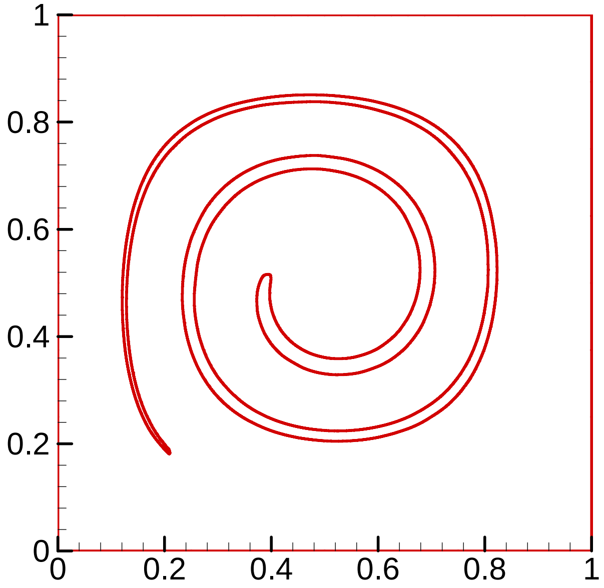

Figure 12. T = 3, Level 8 Adaptive Cartesian Grid, dt= 628/2400,
Level 3 Adaptive Contouring, Second-Order SL Scheme

The particle correction algorithm needs to be applied to simulate the motion after t=3. Initial circular interface are discretized by linear elements with comparable size, and the resolution is at least three orders higher than the resolution of background grids. For example, if level 8 adaptive Cartesian grids are used, then the size of each element is comparable to $1/2^{11}$. Since the interface becomes longer, a new particle is added at the midpoint by checking the length of each element during the simulation. Figure 13 shows the adaptive Cartesian grids and interface at *t*=3 solved on level 8 adaptive Cartesian grids by particle level set method. Compared with the interface in Figure 12, the interface captured here by particle level set method is much more accurate. Compared to the particle level set methods in [26, 27], a more efficient particle correction algorithm is developed in this paper. Much less number of particles are needed to move with high order Runge-Kutta method since the particles are only placed on the interface itself (the particles will represent the interface). By using the linked list, the removing unnecessary particles and adding new particles are quite simple.

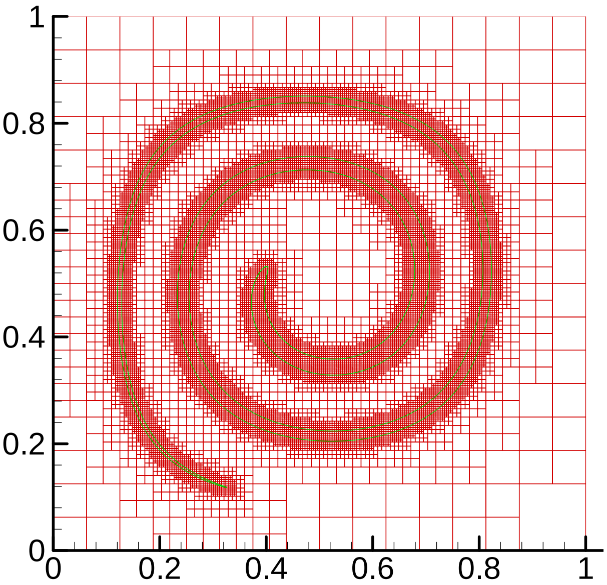

Figure 13. Particle Level Set Method, Level 8 Adaptive Cartesian Grid, T = 3

Figure 14 shows the numerical results and adaptive Cartesian at *t*=5. It can be seen that the semi-Lagrangian level set solver with particle correction algorithm developed in this study are capable of simulating slim interfaces and it is very robust and powerful.

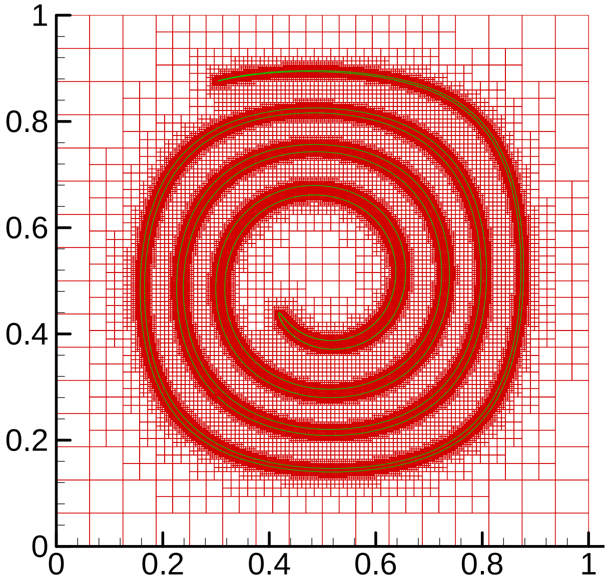

Figure 14. Particle Level Set Method, Level 9 Adaptive Cartesian Grid, T = 5

4.3. Motion under First-Order Geometric Velocity

In this section, the motion of interface under first order geometric velocity field are simulated. The velocity $\vec{V}$ is dependent on the first-order derivatives of level set function ϕ, i.e, $\vec{V}$ is equal to $\vec{V}(x,\phi_x,\phi_y,t)$. In the simulation, ϕ_x and ϕ_y are computed locally with second-order ENO schemes, then velocities of all vertices of interface are evaluated by numerical interpolation. After that Whitney numerical extension is employed to extend the velocity only near the interface. Thus level set function is solved locally within ± 5Δh of the interface.

The first test case is expansion of a single circle initially centered at (0, 0) with a radius of 1. Here it is solved with second-order semi-Lagrangian method. The numerical results are quite accurate with a volume of 28.22 at $t = 2$ (compare to exact solution 28.27 and volume loss is 0.17%). The evolution history is shown in Figure 15.

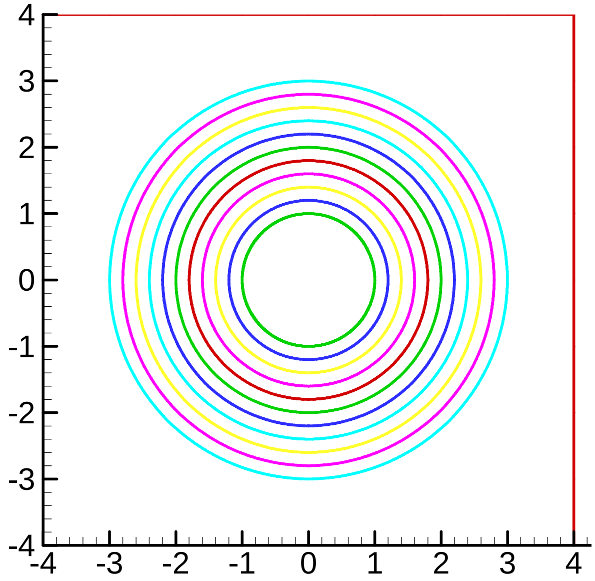

Figure 15. T=0, 2 (0.2), dt = 0.02, Second-Order SL, Level 7 Adaptive Cartesian Grid

As the second case, expansion and shrinking of a triangle are simulated to demonstrate that viscosity solutions can be achieved by semi-Lagrangian method for non-smooth interfaces with sharp corners, where the velocity is non-smooth or even doesn't exist. The expansion history is shown in Figure 16(a) and the shrinking case is shown in Figure 16(b).

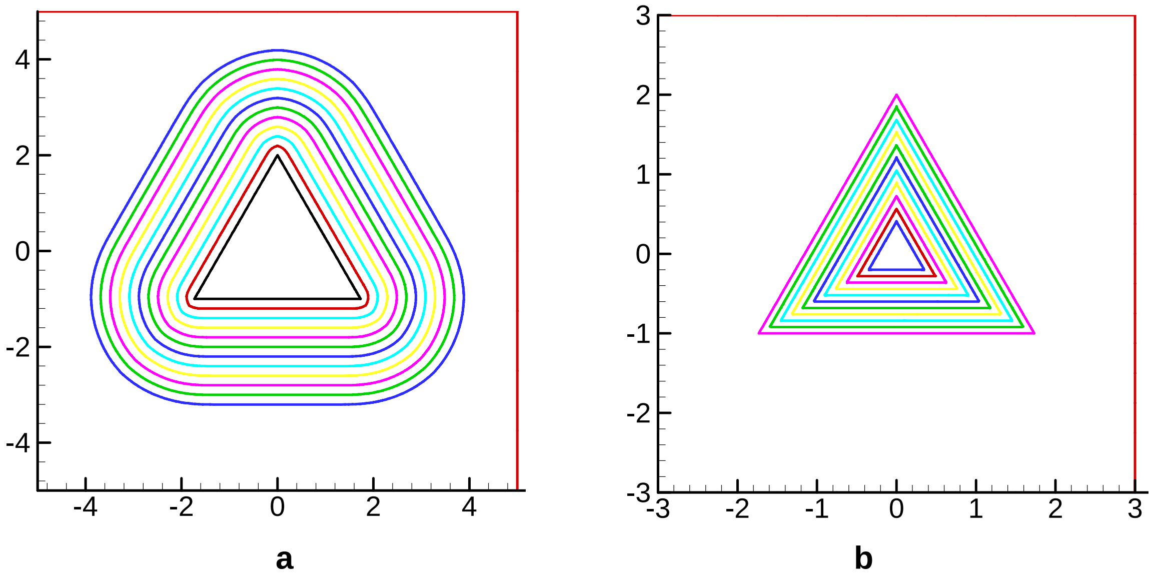

Figure 16. Motion of Triangle
(a) Expansion, T=0, 2.2 (0.1), dt = 0.0125, Second-Order SL, Level 7 Adaptive Cartesian Grid
(b) Shrinking, T=0, 0.8 (0.08), dt = 2.667e-3, Second-Order SL, Level 8 Adaptive Cartesian Grid

The third case is the expansion of a non-convex seven-point star-shaped interface. The results on level 7 adaptive Cartesian grids by second-order semi-Lagrangian level set method are shown in Figure 17. It demonstrates the ability of solving unsmooth problems. Based on the results, similar viscosity solution is obtained as in [1].

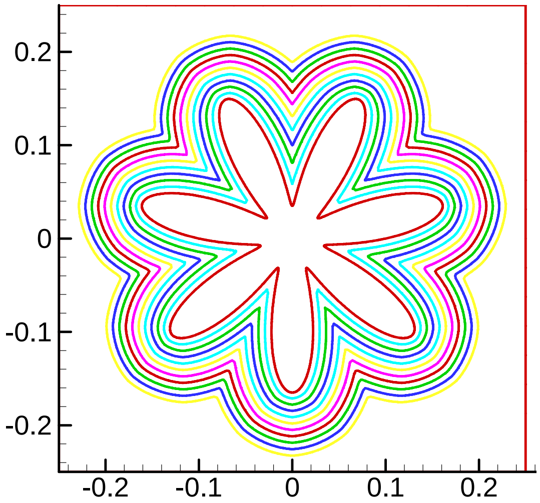

Figure 17. T=0, 0.07 (0.007), dt = 0.0001, Second-Order SL, Level 8 Adaptive Cartesian Grid

4.4. Motion under Second-Order Geometric Velocity

In this case, the motions of interface under second-order geometric velocity field are simulated, that is, $\vec{V}$ is equal to $\vec{V}(\vec{x}, \vec{n}, \vec{\kappa}, t)$. In the simulations, all the derivatives are computed locally with second-order ENO schemes, and then velocities of all vertices of interface are evaluated. After that Whitney numerical extension is used to extend the velocity locally and ϕ is only updated locally.

The first test case is the motion of interface composed of multiple circles under curvature flow, where the speed function is equal to $V = -\kappa$. Initially a small circle centers at (-2.1, -2.1) with a radius of 1.0 and a large circle centers at (1.1, 1.1) with a radius of $\sqrt{5}$. Simulation starts from t = 0 to t = 2. For a circle under curvature flow, the speed of shrinking is equal to $V = 1/r$. It is easy to construct the following ODE

$$-V \cdot dt = \frac{1}{r} dt = dr \qquad (16)$$

and integrate Equation (16) further to find that

$$r^2(t) = r_0^2 - 2(t - t_0). \qquad (17)$$

According to (17), the small circle will vanish at t=0.5 second and the large circle shrinks to a circle with radius equals to 1 at t = 2. The numerical results are shown in Figure 18 and they match analytical solution very well.

The second case is motion of an initially non-convex seven-point star-shaped interface under the curvature flow. According the theory of moving interface, any closed curves will shrink into a circle and finally vanish. This case is used to validate the semi-Lagrangian solver developed in this study. The simulation starts from t =0 to t = 0.002. The numerical results are shown in Figure 19 a, b, c and d.

5. Conclusion

A semi-Lagrangian method for the level set equation has been realized and extended with a new efficient particle correction algorithm for 2D adaptive Cartesian grids. Both the first-order CIR scheme and second-order trapezoidal scheme are tested. With the help of fast redistancing algorithm, numerical velocity extension, fast contouring algorithm, the semi-Lagrangian method is robust and accurate for interfacial simulation. Versatile motions of interfaces under passive transport velocity field, first and second order geometric velocity fields are simulated with the semi-Lagrangian method to show its ability. The particle correction algorithm has applied to further improve the accuracy. The results are excellent and have a very good mass-conserving property. The semi-Lagrangian level set method can be extended to 3D adaptive Cartesian grids further based on this study.

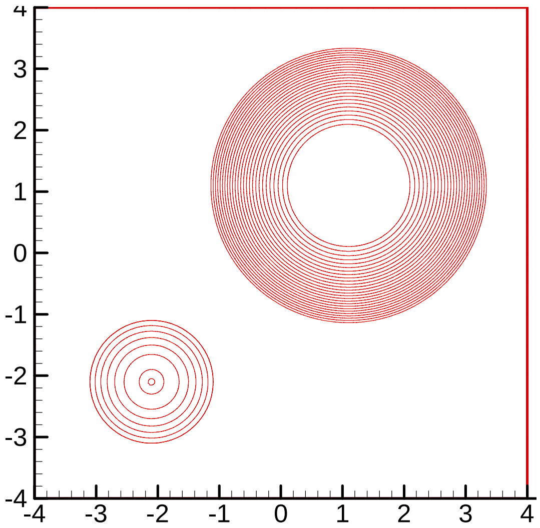

Figure18. Shrinking of Multiple Circles under Curvature Flow, T=0, 2 Second-Order SL. Level 8 Adaptive Cartesian Grid

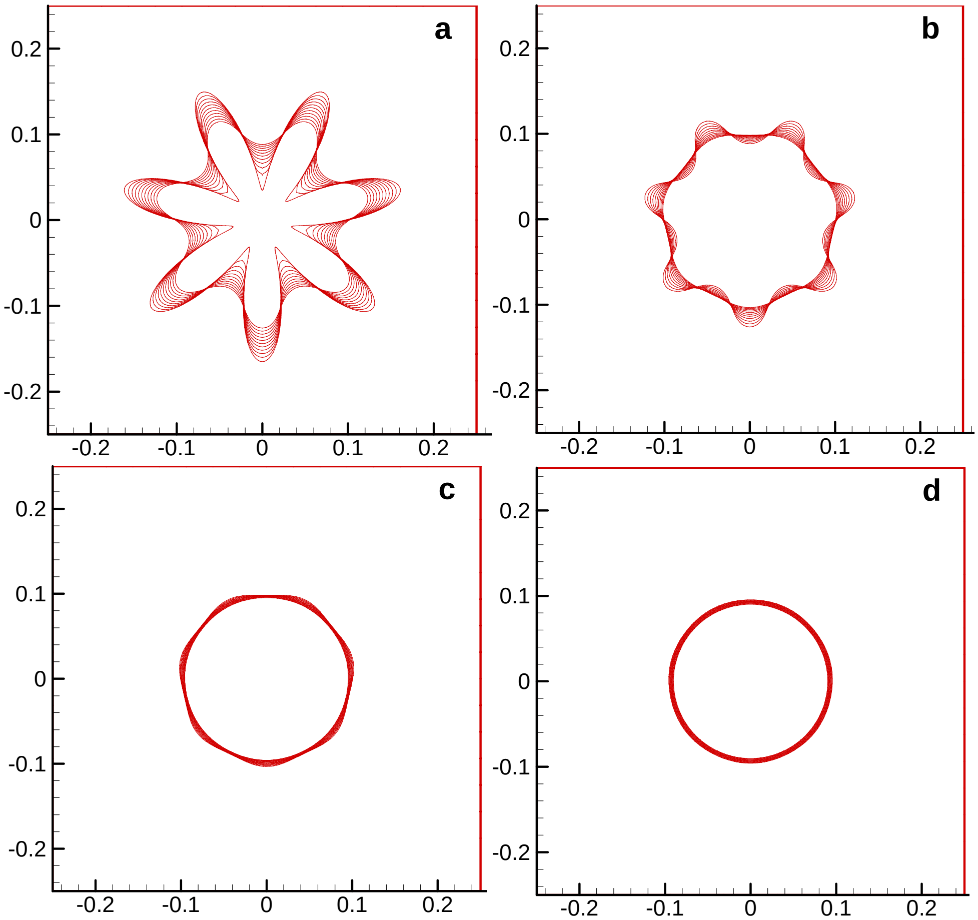

Figure 19. Motion of Star-shaped Interface under Curvature Flow, T=0, 0.002 (0.0005), Second-Order SL, Level 7 Adaptive Cartesian Grid

Acknowledgments

The authors gratefully acknowledge support from the National Science Foundation under contracts DCC-0305504 and DSC-0325760. The authors are grateful to Drs. John Strain for many helpful discussions.

References

1. Osher, S. and Sethian, J.A., “Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations”, J. Comput. Phys., Vol. 79, pp.12-49, 1988.

2. Osher, S. and Fedkiw, R., “Level Set Methods: An Overview and Some Recent Results”, J. Comput. Phys., Vol. 169, pp.463-502, 2001.

3. Osher, S. and Fedkiw, R., “Level Set Methods and Dynamic Implicit Surfaces”, New York : Springer, 2003.

4. Sethian, J.A., “Level Set Methods and Fast Marching Methods”, Cambridge University Press, 1999.

5. Sethian, J. A. and Smereka, P., “Level Set Methods for Fluid Interface”, Annu. Rev. Fluid Mech., 35:341–72, 2003.

6. Fedkiw, R., Aslam, T., Merriman, B., and Osher, S., “A Non-Oscillatory Eulerian Approach to Interface in Multimaterial Flow (The Ghost Fluid Method)”, J. Comput. Phys. Vol. 152(2), pp.457-492, 1999.

7. Mulder, W., Osher, S., and Sethian, J.A., “Computing Interface Motion in Compressible Gas Dynamics”, J. Comput. Phys., Vol. 100, pp.209-228, 1992.

8. Sussman, M., Smereka, P., and Osher, S., “A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow”, J. Comput. Phys., Vol. 114, pp.146-159, 1994.

9. Sussman, M., Fatemi, E., Smereka, P., and Osher, S., “An Improved Level Set Method for Incompressible Two-Phase Flows”, Computers and Fluids, Vol. 27, No. 5-6, pp. 663-680, 1998.

10. Chen, S., Merriman, B., Osher, S., and Smereka, P., “A Simple Level Set Method For Solving Stefan Problems”, J. Comput. Phys., Vol. 135, pp.8-29, 1997.

11. Crandall, M.G. and Lions, P-L., “Viscosity Solutions of Hamilton-Jacobi Equations”, Tran. AMS, Vol. 277, pp. 1-43, 1983.

12. Crandall, M.G. and Lions, P-L., “Two Approximations of Solutions of Hamilton-Jacobi Equations”, Math. Comp., Vol.43, pp.1-19, 1984.

13. Osher S. and Shu C.W., “High-Order Essentially Non-oscillatory Schemes for Hamilton-Jacobi Equations”, SIAM J. Numer. Anal., Vol. 28, No. 4, pp.907-922, 1991.

14. Lafon, F. and Osher, S., “High Order Two Dimensional Non-Oscillatory Methods for Solving Hamilton-Jacobi Scalar Equations”, J. Comput. Phys., Vol. 123, pp.235-253, 1996.

15. Hu, C. and Shu, C.W., “A Discontinuous Galerkin Finite Element Method for Hamilton-Jacobi Equations”, ICASE Report No.98-2, NASA/CR-1998-206903, 1998.

16. Barth, T.J. and Sethian, J.A., “Numerical Schemes for the Hamilton-Jacobi and Level Set Equations on Triangulated Domains”, J. Comput. Phys., Vol. 145, pp.1-40, 1998

17. Jiang, G.S. and Peng, D., “Weighted ENO Schemes for Hamilton-Jacobi Equations”, SIAM J. Sci. Comput., Vol. 21, No.6, pp.2126-2143, 2000.

18. Zhang, Y. and Shu, C.W., “High Order WENO Schemes for Hamilton-Jacobi Equations on Triangular Meshes”, SIAM J. Sci. Comput., Vol. 24, pp.1005-1030, 2003.

19. Levy, D., Nayak, S., Shu, C.W., and Zhang, Y., “Central WENO Schemes for Hamilton-Jacobi Equations on Triangular Meshes”, Scientific computing report, BrownSC-2004-08, 2004.

20. Frolkovic, P. and Mikula, K., “Flux Based Level Set Method: A Finite Volume Method for Evolving interfaces”, Preprint IWR/SFB No. 2003-15, Interdisciplinary Center for Scientific Computing, University of Heidelberg, 2003

21. Wang, Zhu and Wang, Z.J., “The Level Set Method on Adaptive Cartesian Grid for Interface Capturing" AIAA paper, no.2004-0082, 2004.

22. Strain, J., “Tree Methods for Moving Interfaces”, J. Comput. Phys., Vol. 151, pp.616-648, 1999.

23. Strain, J., “A Fast Modular Semi-Lagrangian Method for Moving Interfaces”, J. Comput. Phys., Vol. 161, pp.512-536, 2000.

24. Strain, J., “Fast Tree-based Redistancing for Level Set Computations”, J. Comput. Phys., Vol.152, pp.664-686, 1999.

25. Strain, J., “A Fast Semi-Lagrangian Contouring Method for Moving Interfaces”, J. Comput. Phys.169, pp.1-22, 2000.

26. Enright, D., Fedkiw, R., Ferziger, J., and Mitchell, I., “A Hybrid Particle Level Set Method for Improved Interface Capturing”, J. Comput. Phys., Vol. 183, pp.83-116, 2002.

27. Enright, D., Losasso, F., and Fedkiw, R., “A Fast and Accurate Semi- Lagrangian Particle Level Set Method”, Computers and Structures, Vol. 83, 479-490, 2005.

28. Losasso, F., Fedkiw, R., and Osher, S., “Spatially Adaptive Techniques for Level Set Methods and Incompressible Flow”, Computers and Fluids, (in press).

29. Courant, R., Isaacson, E., and Rees, M., “On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Differences”, Comm. Pure Appl. Math., Vol.5, pp.243-249, 1952.

30. Zalesak, S., “Fully Multidimensional Flux-Corrected Transport Algorithm for Fluids”, J. Comput. Phys., Vol. 31, pp.335-362., 1979.

31. Bell, J., Colella, P., and Glaz, H., “A Second-Order Projection Method for the Incompressible Navier-Stokes Equations”, J. Comput. Phys., Vol. 85, pp. 257-283, 1989.