High-order time-marching reinitialization for regional level-set functions

Article *in* Journal of Computational Physics · October 2017

DOI: 10.1016/j.jcp.2017.10.054

4 authors:

Shucheng Pan
Technische Universität München
29 PUBLICATIONS 66 CITATIONS
SEE PROFILE

Xiuxiu Lyu
Technische Universität München
5 PUBLICATIONS 43 CITATIONS
SEE PROFILE

Xiangyu Hu
Technische Universität München
181 PUBLICATIONS 3,731 CITATIONS
SEE PROFILE

Nikolaus Adams
Technische Universität München
445 PUBLICATIONS 9,771 CITATIONS
SEE PROFILE

High-order time-marching reinitialization for regional level-set functions

Shucheng Pan, Xiuxiu Lyu, Xiangyu Hu, Nikolaus. A. Adams

Lehrstuhl für Aerodynamik und Strömungsmechanik, Technische Universität München, 85748 Garching, Germany

Abstract

In this work, the time-marching reinitialization method is extended to compute the unsigned distance function in multi-region systems involving arbitrary number of regions. High order and interface preservation are achieved by applying a simple mapping that transforms the regional level-set function to the level-set function and a high-order two-step reinitialization method which is a combination of the closest point finding procedure and the HJ-WENO scheme. The convergence failure of the closest point finding procedure in three dimensions is addressed by employing a proposed multiple junction treatment and a directional optimization algorithm. Simple test cases show that our method exhibits 4th-order accuracy for reinitializing the regional level-set functions and strictly satisfies the interface-preserving property. The reinitialization results for more complex cases with randomly generated diagrams show the capability our method for arbitrary number of regions $\mathcal{N}$, with a computational effort independent of $\mathcal{N}$. The proposed method has been applied to dynamic interfaces with different types of flows, and the results demonstrate high accuracy and robustness.

Keywords: Reinitialization, regional level-set function, high-order accuracy, interface preserving

1. Introduction

The level-set method [15] is a well-established interface-capturing method and is being widely used for multiphase flow computation, image processing and computer vision [14]. A reinitialization process is employed to replace the distorted level-set function $\phi^0 : \mathbb{R}^d \rightarrow \mathbb{R}$ during advection by the signed distance function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ which is the solution of the Eikonal equation,

$$|\nabla\phi| = 1 \quad \text{with} \quad \text{sgn}(\phi) = \text{sgn}(\phi^0). \tag{1}$$

Successful numerical methods for directly solving this stationary boundary value problem include the fast marching method [21] and the fast sweeping method [27]. One can also transform Eq. (1) to a time-marching problem [25, 24],

$$\phi_t + \text{sgn}(\phi^0)(|\nabla\phi| - 1) = 0, \tag{2}$$

which is a Hamilton-Jacobi (HJ) equation with a discontinuous coefficient across the interface. Numerical approximations of Eq. (2) may exhibit oscillations or interface shifting [18], as the discretization of the derivatives across the interface may employ erroneous level-set information from the other side of the interface. High-order schemes specially developed for HJ equations [11] may suffer from order degeneration and large truncation errors [18, 5]. Modifications [24, 18, 13, 5, 9, 3] have been proposed to cope with spurious displacement of the interface and successfully reduce the mass loss [7].

The level-set method has been extended to simulate the interface evolution of a multi-region system involving arbitrary number of regions by using multiple level-set functions [26, 23] or a single regional level-set function [28, 20, 17], i.e., a combination of the unsigned distance function $\varphi(\mathbf{x}) \geq 0$ and the integer region indicator $\chi(\mathbf{x})$, where

Email addresses: `shucheng.pan@tum.de` (Shucheng Pan), `xiuxiu.lyu@tum.de` (Xiuxiu Lyu), `xiangyu.hu@tum.de` (Xiangyu Hu), `nikolaus.adams@tum.de` (Nikolaus. A. Adams)

$\mathbf{x} \in \Omega$ is a point in the computational domain Ω. For multiple level-set functions, reinitialization is applied to each level-set function [23]. Reinitialization for the regional level-set function can be accomplished by the fast marching method [20]. Although the time-marching method is considered to be more costly, it is more flexible and easier to parallelize [3]. When applied to the regional level-set method, high-order accuracy, to our knowledge, has not been demonstrated in the literature. For example, the time-marching reinitialization method used in Ref. [17] to regularize the regional level-set functions is limited to 1st order without preserving the interface location. Its computational cost depends on the number of regions due to the region-by-region updating. These drawbacks motivate this paper which is dedicated to demonstrating how to achieve high order and strict interface preservation for reinitializing regional level-set functions. We employ a simple mapping, which previously has been used to construct multiple local level-sets for solving regional level-set advection in Ref. [17], to transform the regional level-set function to the level-set function and apply a high-order two-step reinitialization method which is a combination of the closest point finding procedure [4] and the HJ-WENO method [11]. Compared to the time-marching regional level-set reinitilization method in Ref. [17], the present method can (i) achieve high-order accuracy, (ii) simplify the updating of the regional level-set function by imposing the interface-preserving property, and (iii) reduce the computational cost to be approximately the same as the original level-set reinitialization problem for two phases.

2. Reinitialization of a regional level-set function

Let $\mathrm{X} = \{r \in \mathbb{N} | 1 \le r \le \mathcal{N}\}$ be the index set for all regions. Regional level-set reinitialization corresponds to finding the solution of

$$|\nabla\varphi| = 1 \quad \text{with} \quad \chi = \chi^0 \tag{3}$$

on each region domain Ω^χ which is an open subset of Ω, such that Eq. (1) is decomposed into $\mathcal{N}$ sub-problems. A good reinitialization method for regional level-set functions should preserve the interface and achieve high-order accuracy with a computational effort weakly depending on the number of the regions $\mathcal{N}$. The displacement of the interface may lead to a sign change of φ, corresponding to a region-indicator change, which is difficult to handle for $\mathcal{N} > 2$. Thus, we require that the developed method does not change the sign of φ so that the indicator χ remains invariant during the reinitialization. Although many methods, especially the subcell fix scheme [18, 5, 13], work well for reinitializing original level-set or signed distance function [7], the sign of φ is not ensued to be invariant when those methods are applied to solve the regional level-set reinitialization problem.

Direct implementation of the time-marching reinitialization method on a regional level-set function gives

$$\varphi_t + |\nabla\varphi| = 1 \tag{4}$$

which is incorrect near the interface for reinitialization as $\nabla\varphi$ gives the wrong characteristic direction across the interface. To address this issue, we employ a mapping $\mathbf{C}_r : \mathbb{R} \times \mathbb{N} \to \mathbb{R}$ defined as

$$\phi^r(\mathbf{x}) = \mathbf{C}_r\left(\varphi(\mathbf{x}), \chi(\mathbf{x})\right) = \begin{cases} \varphi(\mathbf{x}) & \text{if } \chi(\mathbf{x}) = \chi_r \\ -\varphi(\mathbf{x}) & \text{otherwise} \end{cases}, \quad \chi_r \in \mathrm{X}. \tag{5}$$

It transforms the regional level-set function to a level-set function for each region χ_r. Eq. (4) is reformulated as

$$\varphi_t + |\nabla\phi^r| = \varphi_t + |\nabla\mathbf{C}_r(\varphi)| = 1, \tag{6}$$

which is a standard HJ equation. Now, reinitialization can be performed region by region [17], or by solving $\mathcal{N}$ scalar evolution equations [3]. Both have computational cost scaling linearly with $\mathcal{N}$ due to global mapping from φ and χ to ϕ. The same cost can be achieved with time-marching reinitialization methods [25, 13, 5, 9].

To reduce the computational cost to be approximately the same as for the original level-set reinitialization method, we can locally apply the mapping on every stencil of the spatial discretization schemes. Considering a 2D multi-region system defined by $\varphi_{i,j}$ and $\chi_{i,j}$ on a uniform Cartesian grid, with i and j being the indices of the coordinate directions, we apply a mapping for each grid point on the stencil of the discretization scheme,

$$\phi^r_{i+k,j} = \mathbf{C}_r\left(\varphi_{i+k,j}, \chi_{i+k,j}\right), \quad \phi^r_{i,j+l} = \mathbf{C}_r\left(\varphi_{i,j+l}, \chi_{i,j+l}\right), \quad \chi_r = \chi_{i,j} \tag{7}$$

where $-t \leqslant k \leqslant t$, $-t \leqslant l \leqslant t$, and $2t+1$ is the required stencil width. Note that Eq. (7) is a local operation and the local level-set functions, $\phi^r_{i+k,j}$ and $\phi^r_{i,j+l}$, are temporary variables, unlike the regional level-set (φ, χ) which is defined globally. Thus (φ, χ) at other grid points remains invariant. Now the information is propagated from the region boundary $\partial\Omega^{\chi_{i,j}}$ to the interior of the region domain $\Omega^{\chi_{i,j}}$. The mapping $\mathbf{C}_r$ serves the same purpose as the signum term in Eq. (2). The semi-discrete form of Eq. (6) is

$$\varphi_t + H^G(D_x^+\mathbf{C}_r(\varphi), D_x^-\mathbf{C}_r(\varphi), D_y^+\mathbf{C}_r(\varphi), D_y^-\mathbf{C}_r(\varphi)) = 1 \tag{8}$$

where H^G is the Godunov numerical Hamiltonian [15, 11], $H^G(a,b,c,d) = \sqrt{\max(|a^-|^2, |b^+|^2) + \max(|c^-|^2, |d^+|^2)}$, with $f^+ = \max(f, 0)$ and $f^- = \min(f, 0)$. If $\mathcal{N} = 2$, this is identical to traditional time-marching level-set formulation which can be solved by existing high-order schemes [16, 11]. However directly applying these high-order schemes requires additional operations when $\mathcal{N} > 2$. When the solution of Eq. (8) changes its sign after one time-step, updating of χ is required, which is not encountered with the original level-set problem. We emphasize that unlike traditional level-set reinitialization where interface-preservation serves to achieve good mass conservation and high accuracy of the interface location, in our problem we require the interface-preservation property as it implies that the sign change of the solution in Eq. (8) is avoided and thus the corresponding χ updating. Note that only strict interface-preservation allows to omit updating of χ, although high-order may be achieved without satisfying this condition [13, 5].

2.1. *High-order two-step reinitialization method*

To ensure interface-preservation and high-order accuracy, we perform 2 operations. First, as with the initialization step of the fast marching method [21], we tag the grid point (i, j) adjacent to the interface as *Alive*, others as *Far*. A cell $[i-1, i] \times [j-1, j]$ is a cut-cell if it contains an *Alive* grid point. Inside every cut-cell, the level-set function is approximated by a bicubic polynomial [4] whose coefficients are determined by interpolation using all 16 grid-point values around this cell to achieve 4th-order accuracy. If the cut-cell $[i-1, i] \times [j-1, j]$ is shared by more than 2 regions, say $\{\chi_a, \chi_b, \chi_c\}$, we need to reconstruct multiple polynomials,

$$p^r_{i,j}(x, y) = \sum_{\xi=0}^{3}\sum_{\eta=0}^{3} c^r_{\xi\eta} x^\xi y^\eta, \qquad r = a, b, c \quad , \tag{9}$$

where $c^r_{\xi\eta}$ is determined by interpolating $\phi^r_{i+i',j+j'} = \mathbf{C}_r(\varphi_{i+i',j+j'})$, with $\chi_r \in \{\chi_a, \chi_b, \chi_c\}$ and $i', j' \in [-2, 1]$. A modified Newton method [4] is used to find the closest point $\mathbf{x}'$ that satisfies $p^r(\mathbf{x}') = 0$ and $\nabla p^r(\mathbf{x}') \cdot (\mathbf{x}' - \mathbf{x}) = 0$ simultaneously, where $\mathbf{x}$ is the location of an *Alive* grid point. The distance of an *Alive* grid point (i, j) to its region boundary is $\varphi_{i,j} = \min_{p \in \mathcal{P}^r} \|\mathbf{x}_{i,j} - \mathbf{x}'_p\|$ with $\chi_r = \chi_{i,j}$, where $\mathcal{P}^r$ is the set of all polynomials in cut-cells that share the grid point (i, j).

The second step is to update iteratively all *Far* grid-point values by solving Eq. (8). To ensure that global accuracy is at least the same as that of the 4th-order accurate approximation of *Alive* grid values, we employ a 5th-order HJ-WENO scheme [11] for approximation of the derivatives in H^G,

$$\begin{aligned} D_x^-\mathbf{C}_r(\varphi) \quad = \quad & w_0 \frac{2\Delta_x^+\phi^r_{i-3,j} - 7\Delta_x^+\phi^r_{i-2,j} + 11\Delta_x^+\phi^r_{i-1,j}}{6\Delta x} + w_1 \frac{-\Delta_x^+\phi^r_{i-2,j} + 5\Delta_x^+\phi^r_{i-1,j} + 2\Delta_x^+\phi^r_{i,j}}{6\Delta x} \\ + \quad & w_2 \frac{2\Delta_x^+\phi^r_{i-1,j} + 5\Delta_x^+\phi^r_{i,j} - \Delta_x^+\phi^r_{i+1,j}}{6\Delta x} \end{aligned} \tag{10}$$

where $\Delta_x^+\phi^r_{i,j} = \phi^r_{i+1,j} - \phi^r_{i,j}$. The weighting factors, w_0, w_1 and w_2, are defined in Ref. [11]. Thus a fully 4th-order reinitialization method is achieved that prevents spurious sign change of φ. In the following sections we compare the high-order method with a low-order two-step regional level-set reinitialization method which uses piecewise linear functions to approximate *Alive* values and an upwind finite difference scheme for *Far* values. The high-order subcell fix method in Ref. [13] is applied to the solution of Eq.(8) for reducing the interface oscillation. Note that we replace the one-sided ENO scheme with the 3rd-order HJ-WENO scheme [11] for grid points away from the interface to achieve fully 3rd-order accuracy.

Note that bicubic interpolation degenerates to 1st-order accuracy near triple points. In this case, one can use piecewise linear interpolation which gives 2nd-order accuracy. A simple possibility is to employ a common triangulation algorithm, such as marching cubes [12] or marching tetrahedra [1], after mapping $\mathbf{C}_r$. Such simple methods, however, may generate invalid interface segments in a cut-cell with more than two regions. As shown in Fig. 1, the interface deviates from the expected symmetric configuration for the first case. Also, it can not resolve quadruple points and octuple points. We propose a method to improve the triangulation results for multi-region systems. A complete interface is the union of all segments between different regions, $\Gamma = \bigcup_{\alpha,\beta\in[1,\mathcal{N}]\times[1,\mathcal{N}]} \Gamma_{\alpha\beta}$. Suppose that the cut-cell $[i-1,i]\times[j-1,j]$ is shared by $\mathcal{N}_{ij}$ regions, where $\mathcal{N}_{ij} \leqslant 4$ in 2D for resolved cases. The main steps to find $\Gamma_{\alpha\beta}$ are: (i) apply mapping $\mathbf{C}_\alpha(\varphi)$ for $\alpha \in [1, \mathcal{N}_{ij}]$; (ii) for every vertex of the cell apply piecewise linear interpolation $p^\alpha(x,y) = c_0^\alpha + c_1^\alpha x + c_2^\alpha y$ and modify the vertex level-set data by $\varphi_\alpha^* = \min_{p\in\mathcal{P}^\alpha} d_p$, where d_p is the distance to the plane $p^\alpha = 0$ and $\mathcal{P}^\alpha$ is the set of all piecewise linear functions in every simplex that contains the current vertex; (iii) construct the piecewise linear function $p^{\alpha\beta}(x,y)$ by interpolating $\mathbf{C}_\alpha(\varphi_\alpha^*) - \mathbf{C}_\beta(\varphi_\beta^*)$ and extract the interface segments of $p^{\alpha\beta}(x,y) = 0$; (iv) apply the chop operation of Ref. [19], i.e., cut the interface where $\mathbf{C}_\alpha(\varphi_\alpha^*) < \mathbf{C}_\gamma(\varphi_\gamma^*)$ for $\gamma \neq \alpha,\beta$. Typical examples shown in Fig. 1(b) indicate that this procedure improves the triangulation results in Fig. 1(a). Unlike with the method in Ref. [19], it is not necessary to calculate the distance function by solving the reinitialization equation. Moreover, this procedure is needed only near triple points. It can be used also for explicit interface extraction of multi-region systems, as shown in Fig. 4(f). Finally, we can update the level-set data by simply computing the distance to each interface element, $\varphi_{i,j} = \min_{p\in\mathcal{P}^r_{\alpha\beta}} \|\mathbf{x}_{i,j} - \mathbf{x}'_p\|$ with $\chi_r = \chi_{i,j}$, where $\mathcal{P}^r_{\alpha\beta}$ is the set of all piecewise linear polynomials with $r = \alpha$ or $r = \gamma$.

Extension to three dimensions (3D) encounters an issue with the closest point finding algorithm of Ref. [4] which may not converge within a prescribed number of iterations, as reported in Ref. [10] for two region cases. This issue becomes more serious near multiple junctions due to the existence of discontinuities in the multi-region cases. Without additional modification, the multi-region 3D test cases with an initially highly distorted level-set function such as that in Fig. 4(f) fail to converge near the sharp corner. For cut-cells with $\mathcal{N}_{ij} > 2$, the above piecewise linear method has no convergence problems. For cut-cells near multiple junctions with $\mathcal{N}_{ij} = 2$, we can address this issue by an improved closest point finding algorithm. We choose the directional optimization algorithm developed in Ref. [6] as a safeguard to guarantee convergence if the closest point finding algorithm does not converge. This algorithm is more robust as it employs Newton iteration in the azimuthal direction. In addition, we make minor modifications to achieve the high-order convergence in the L_∞ norm. In the line search procedure of Ref. [6] which is used to find a projection $\mathbf{x}_0$ of a point $\mathbf{x}$ on the interface, the magnitude of $\mathbf{I}_\phi(\mathbf{x}_0)$ is enforced to be lower than a given tolerance, e.g. $0.01\Delta x^3$, where $\mathbf{I}_\phi$ is the tricubic interpolation operation of ϕ. In most of our numerical examples, this safeguard is rarely invoked. For example, if the grid resolution is 256^3, about 0.47% and 1.93% of all *Far* grid points employ the directional optimization algorithm for the 3D case with $\mathcal{N} = 3$ and the 3D randomly generated Voronoi diagrams case with $\mathcal{N} = 5$ shown in Fig. 6, respectively.

3. Numerical examples

In this section, we assess the capability and accuracy of our time-marching regional level-set reinitialization method by a range of numerical examples. Both 2D and 3D cases are considered. For all test cases, Eq. (8) is solved by a 3rd-order strongly stable Runge-Kutta scheme [22] with a CFL number of 0.5.

3.1. Two-region system

Two cases with 2 regions are considered to test the accuracy and interface-preserving property of our reinitialization method for regional level-set functions. First, we test a simple case with an initial unsigned regional level-set function $\varphi^{0,\chi}$ given by

$$\varphi^0(x,y) = |f(x,y)|, \quad f(x,y) = e^{\sqrt{(x-5)^2+(y-5)^2}-2.313} - 1, \quad \chi^0(x,y) = \begin{cases} 0 & \text{if } f(x,y) < 0 \\ 1 & \text{otherwise} \end{cases}, \tag{11}$$

on a rectangular domain $[0,10]\times[0,10]$. For comparison of accuracy, we compute the error measures, $L_1(\varphi)$, $L_\infty(\varphi)$, $L_1(\kappa)$ and $L_\infty(\kappa)$, where $\kappa = \nabla\cdot\left(\frac{\nabla\phi^r}{|\nabla\phi^r|}\right)$ is the mean curvature. The errors are measured on the full domain, with the

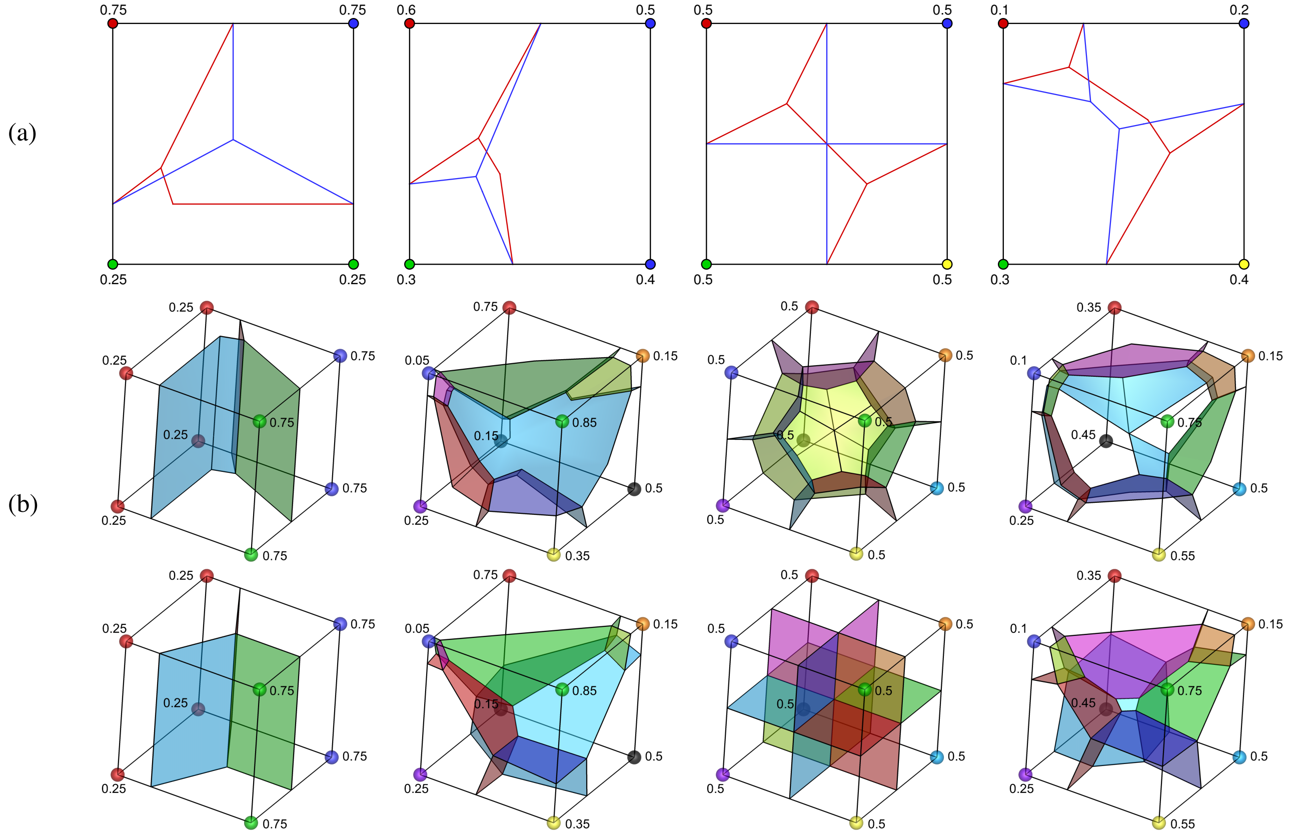

Figure 1: Triangulation results for (a) 2D and (b) 3D cases. The regional level-set functions are defined at vertices, where color indicates χ and data indicate φ. The extracted interface patches are labelled by different colors for each pairwise interface. The results of the simple linear interpolation method and the proposed method are represented by red and blue lines in (a) and upper and lower rows in (b).

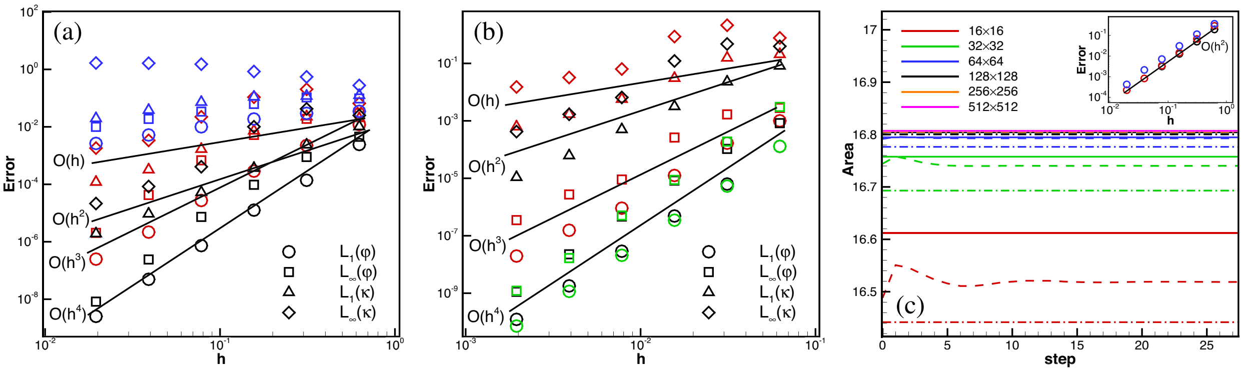

Figure 2: Reinitialization for 2-region cases: errors of the first (a) and second (b) case, and (c) time history of area of the fist case. The blue, red, and black symbols indicate results of the low-order, subcell fix, and the high-order two-step method, respectively. The solid, dashed, and dash-dotted lines indicate results of high-order two-step, subcell fix and low-order methods, respectively. The green symbols are the 3D results of the high-order two-step methods.

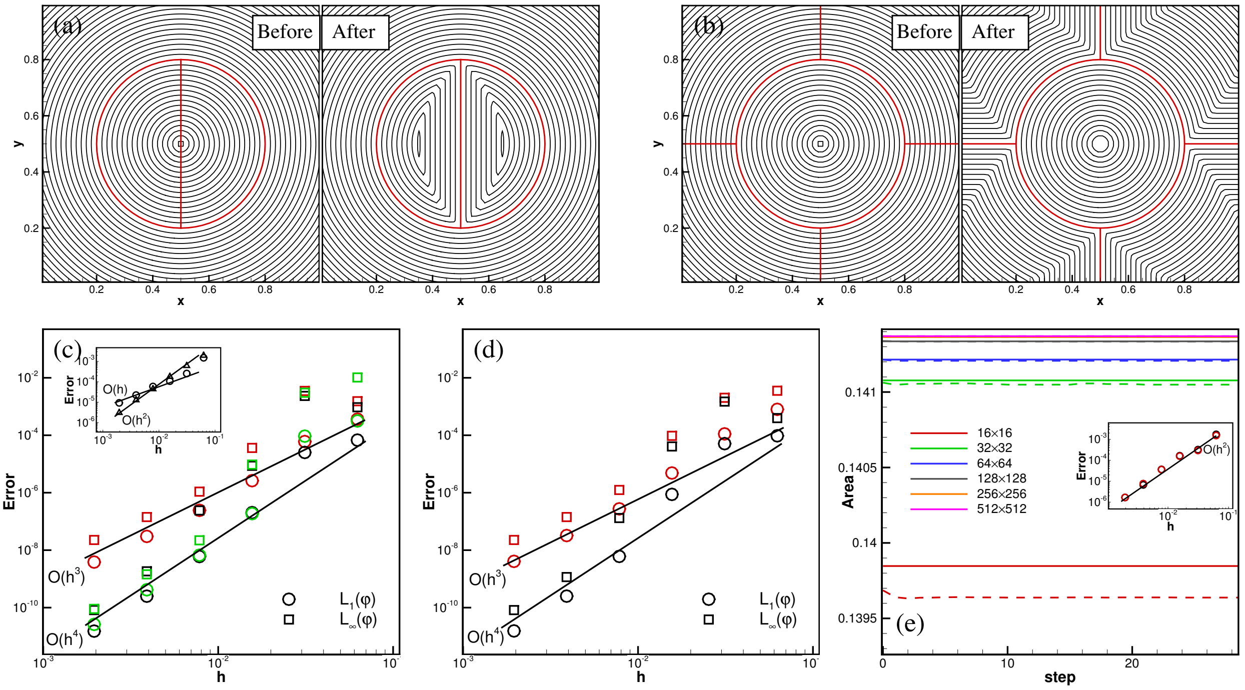

Figure 3: Reinitialization for multi-region cases: φ contours for the 3-region (a) and the 5-region (b) case, errors of the 3-region (c) and the 5-region (d) case, and (e) time history of area of the 3-region case. The red and black symbols stand for results of the subcell fix and the high-order two-step methods, respectively. The solid and dashed lines indicate results of high-order two-step and subcell fix methods, respectively. The green symbols are the 3D results of the high-order two-step methods. The insert of (c) shows the L_∞ errors near the triple points.

kink of (5, 5) excluded. As shown in Fig. 2(a), the high-order two-step method achieves 4th-order accuracy for error norms of φ and the corresponding 2nd-order accuracy for error norms of κ, which are higher than those for the subcell fix method (3rd-order φ and 1st-order κ) and the low-order method (1st-order φ and nonconvergent κ), indicating significantly reduced truncation errors. The area-conservation property of region $\chi = 0$ is given in Fig. 2(c), and corresponds to interface preservation. Although the high-order subcell fix method estimates the area better than the low-order method, it still exhibits distict interface oscillations. Such oscillations are eliminated by our high-order two-step method. All methods show 2nd-order accuracy for area errors due to the 2nd-order approximation of the area estimate [8]. The absolute error level of the low-order method is higher than for the high-order methods. The second 2-region case has a asymmetrical distribution of initial regional level-set function [13], $f(x,y) = [(x-0.75)^2 + (y-0.75)^2 + 0.025][\sqrt{(x-0.5)^2+(y-0.5)^2} - 0.25]$, on a unit square. The errors in Fig. 2(b) indicate similar accuracy for 2D. The results of the 3D test case with

$$\varphi^0(x,y,z) = \left[(x-0.75)^2 + (y-0.75)^2 + (z-0.75)^2 + 0.025\right]\left|\sqrt{(x-0.5)^2+(y-0.5)^2+(z-0.5)^2} - 0.25\right|, \qquad (12)$$

are represented by green symbols in Fig. 2(b) and show that the expected order of accuracy is achieved.

3.2. *Multi-region system*

The first case is a 2D circle of radius $r = 0.3$ which is divided into two equal parts on the computational domain $[0,1] \times [0,1]$ with initial regional level-set function being

$$\varphi^0(x,y) = |f(x,y)| = |\sqrt{(x-0.5)^2+(y-0.5)^2} - 0.3|, \quad \chi^0(x,y) = \begin{cases} 0 & \text{if } f(x,y) < 0 \quad \text{and} \quad x < 0.5 \\ 1 & \text{if } f(x,y) < 0 \quad \text{and} \quad x \geqslant 0.5 \\ 2 & \text{otherwise} \end{cases}, \qquad (13)$$

Table 1: CPU times (in seconds) of the reinitialization for two randomly generated diagrams. The reinitialization is performed by the high-order two-step method with a grid resolution of 512 × 512.

		$\mathcal{N}$				
		5	10	20	50	100
Circle bounded diagrams*	constant m	65.782	66.122	65.755	60.703	59.126
	case dependent m	13.100	9.204	5.908	4.217	3.028
Periodic diagrams	constant m	65.893	65.997	64.931	59.764	57.868
	case dependent m	16.565	13.912	9.472	5.849	4.113

* In this case, the number of Voronoi cell is $\mathcal{N} - 1$.

which means the φ^0 function is not consistent with the χ function, as shown in Fig. 3(a). After reinitialization, the unsigned level-set function matches the χ function very well. Error measures, $L_1(\varphi)$ and $L_\infty(\varphi)$, are computed within a narrow band, $\{(i, j)|\varphi_{i,j} < 3\Delta x \wedge \|\mathbf{x}_{i,j} - \mathbf{x}_s\| < 0.05\}$, where $\mathbf{x}_{i,j}$ and $\mathbf{x}_s$ are the locations of grid point and triple points, respectively. As expected, in Fig. 3(c), we achieve the same order of accuracy as with that for the 2-region cases. The errors of the 3D case with

$$\varphi^0(x, y, z) = |f(x, y, z)| = |\sqrt{(x-0.5)^2 + (y-0.5)^2 + (z-0.5)^2} - 0.3|, \quad \chi^0(x, y, z) = \begin{cases} 0 & \text{if } f(x,y,z) < 0 \quad \text{and} \quad x < 0.5 \\ 1 & \text{if } f(x,y,z) < 0 \quad \text{and} \quad x \geqslant 0.5 \\ 2 & \text{otherwise} \end{cases}, \quad (14)$$

indicate similar accuracy, and the subfigure shows that the bicubic interpolation exhibits order degeneration near the triple points, which can be improved by the piecewise linear method. The subcell fix produces oscillatory results for the area of region $\chi = 0$, indicating an oscillatory interface during iterations. The two-step method provides a better interface-preservation, see Fig. 3(e). Analogously, for 5 regions our method produces a reasonable φ function after reinitialization in Fig. 3(b) and exhibits 4th-order convergence rate of error measures, $L_1(\varphi)$ and $L_\infty(\varphi)$, in Fig. 3(d).

We consider cases with randomly generated Voronoi diagrams to test the capability and performance of our method for a multi-region system with arbitrary $\mathcal{N}$. The diagrams are either generated inside a circle or on a square with a periodic configuration. Both cases have a region number $\mathcal{N}$ ranging from 5 to 100 and the initial φ is assigned to be $\varphi^0 = e^d - 1$, where d is the distance to the Voronoi edges. The reinitialization is performed on a domain $[0, 10] \times [0, 10]$ with 512×512 grid points. As shown in Figs. 4(b) and (c), after the reinitialization the incorrect φ of the circle-bounded case is replaced by the distance to Voronoi edges. The reconstructed interface network in Fig. 4(a) coincides with exact one. The errors in the case $\mathcal{N} = 100$ are 4.5×10^{-6} for L_1 norm and 1.8×10^{-5} for L_∞ norm. The second random case shows similar results, see Figs. 4 (d) and (e). The execution times in Table 1 indicate that the computational effort is approximately independent of the region number if the number of iterations is constant, $m = 512/\text{CFL}$ [13]. The largest Voronoi cell in Fig. 4(a) with $\mathcal{N} = 100$ is only $1/10$ of the domain size, thus a very small m suffices in this situation, and the CPU time can be reduced by 20 times, as shown in Table 1. A 3D random generated diagram case with $\mathcal{N} = 5$ is also considered, as shown in Fig. 4(f). The initial distorted level-set function is defined the same way for the 2D cases. The computation is performed with 256^3 grid points and its errors are 2.6×10^{-7} for L_1 norm and 1.1×10^{-6} for L_∞ norm. The interface extracted by the triangulation method is colored by gray, and the isosurfaces with φ ranging from 0.02 to 0.08 are also shown. The 2D contours plotted in the plane $y = 0.5$ indicate a regularized level-set field.

3.3. Level-set motion with reinitialization

Different types of flows are tested to demonstrate the ability of the present regional level-set reinitialization method. To make sure the numerical error generated during advection is not dominant, the advection of the interface is solved by a high-order regional level-set method [17], where the 5th-order HJ-WENO scheme [11] and 3rd-order SSP Runge-Kutta scheme [22] are used to discretize the level-set advection equation. Reinitialization is performed after every timestep.

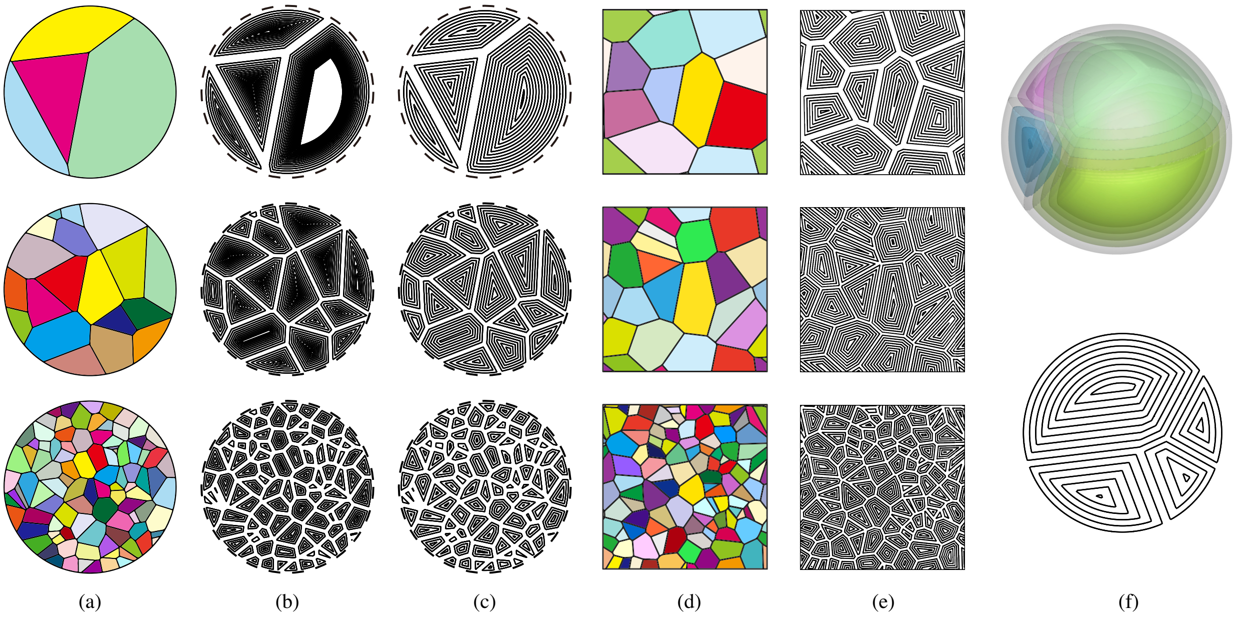

Figure 4: Reinitialization for randomly generated cases with different $\mathcal{N}$. The value of the χ indicator for the cases with random diagrams inside a circle and periodic random diagrams are represented by different colors in (a) and (d), respectively, where the Voronoi edges and the circle are reconstructed from the φ function after reinitialization. φ contours before and after the reinitialization for the first case are plotted in (b) and (c), respectively, where the dashed lines indicate the circular boundary. (e) shows φ contours after the reinitialization for the second case. The extracted interface and φ contours at $y = 0.5$ of a 3D case are shown in (f).

The first case is the normal driven flow with 2 regions, where the velocity $\mathbf{u}$ is determined by the normal direction $\mathbf{n}$. A circle with a radius of $r = 0.2$ at $t = 0$ expands with a uniform speed to $r = 0.4$ at $t = 0.2$. As expected, a 4th-order method for level-set functions produces 3rd-order errors, as $\mathbf{u}$ is the first derivative of the level-set function, see Fig. 5(a). The second example is the triple-point advection under constant velocity $\mathbf{u} = (1, 0)$. A 2D circle of radius $r = 0.2$ is divided into two equal parts and transported by the flow field until $t = 0.4$. As shown in Fig. 5(b), the L_1 and L_∞ norms indicate that the reinitialization errors dominate. To assess the robustness of the reinitialization method in more complex flows, we employ the vortex flow $\mathbf{u} = (\frac{\partial \Psi}{\partial y}, -\frac{\partial \Psi}{\partial x})$, where $\Psi = \sin^2(\pi x)\cos^2(\pi y)\cos(\pi t/3)/\pi$ [2]. The circle of the previous case is located at $(0.5, 0.75)$ with a radius of 0.15. The initial two triple points move due to the background flow fields, and the interface deforms into a filament wrapping around the center of the domain. At $t = 3$ the interface reaches its maximum deformation, and each region becomes a thin filament with sharp corners, as shown in Fig. 5(d). Then the interface wraps back into the initial configuration at $t = 6$, see Fig. 5(d). In Fig. 5(c), the error of area loss is measured at $t = 6$ and exhibits 2nd-order convergence rate.

Finally we consider 3D cases with velocity $\mathbf{u} = (1, 0, 0)$. First we extend the 2D triple-point advection to 3D, where a sphere of radius $r = 0.2$ is divided into two equal parts. This system is transported until $t = 0.4$. Second we advect the 3D cases with randomly generated Voronoi diagrams, Fig. 4(f), and $\mathcal{N} = 5$ until $t = 0.2$. As shown in Fig. 6, the L_1 and L_∞ norms indicate that the reinitialization errors dominate. The extracted interface network ($\varphi = 0$) and interior iso-surfaces ($\varphi = 0.02 - 0.08$) at $t = 0.2$ correspond to a regularized level-set field with 256^3 grid points.

4. Concluding remarks

In this short note we demonstrate how to achieve high-order accuracy for computing distance functions in multi-region systems involving arbitrary number of regions by solving the time-marching reinitialization equation. We employ a simple map of the regional level-set function onto the level-set function inside the stencil of a finite difference scheme, followed by solving the discretized HJ equation by a high-order two-step reinitialization method, which

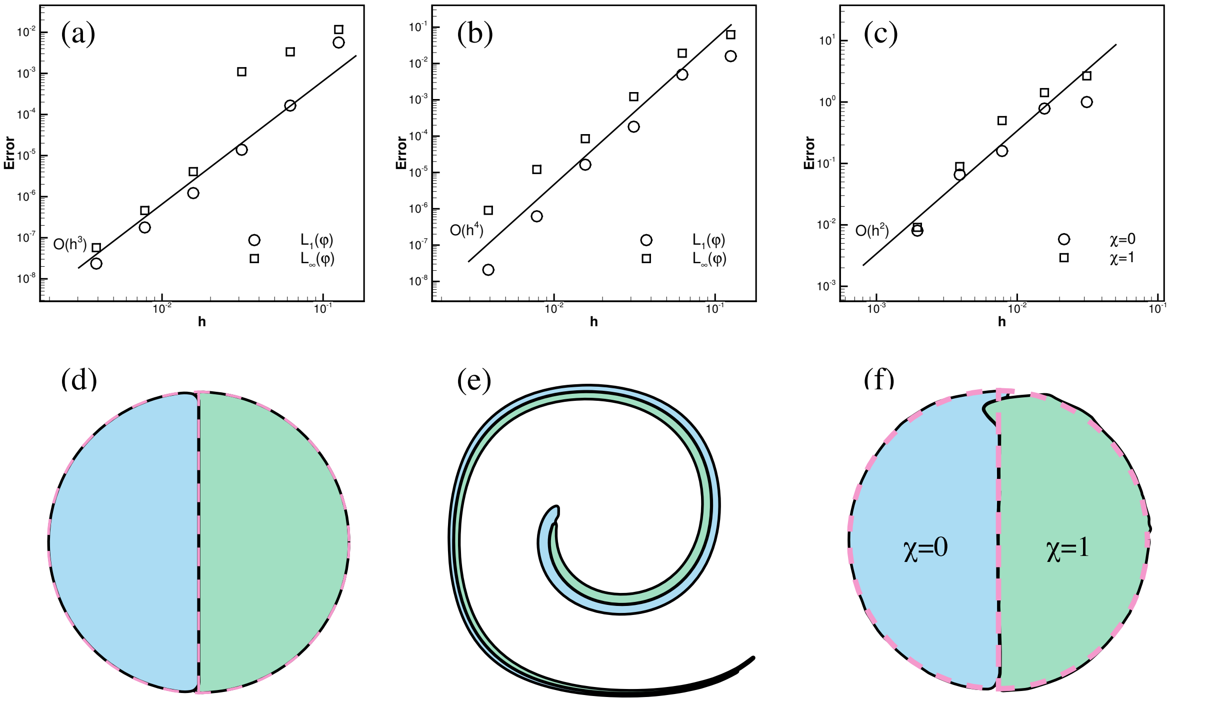

Figure 5: Applications of the regional reinitialization method for different types of flows. The convergence results of the normal driven flow, triple-point advection and vortex flows are shown in (a), (b) and (c), respectively. The interface of the vortex flows is extracted at $t = 3$ (d) and $t = 6$ (e). The dashed line indicates the exact interface.

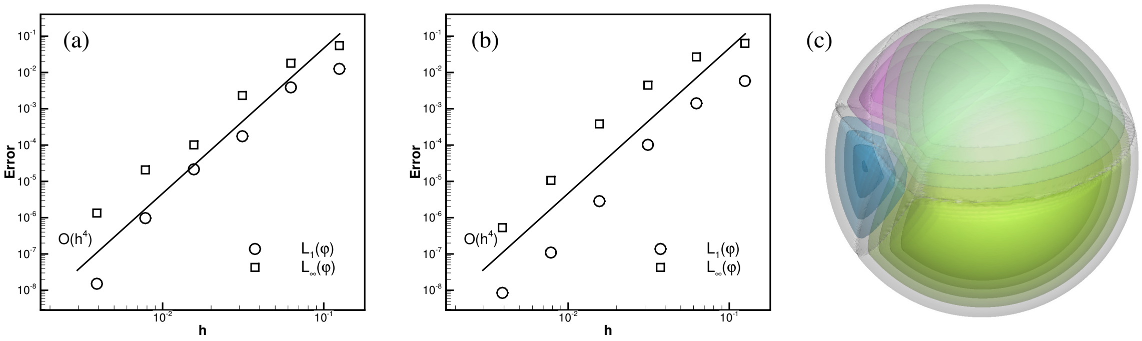

Figure 6: Three-dimensional advection with the regional reinitialization method. The convergence results of the 3-region and 5-region advection are shown in (a) and (b), respectively. The extracted interfaces, $\varphi = 0, 0.02, 0.04, 0.06$ and 0.08, are shown in (c) for $t = 0.2$.

is a combination of a closest point finding procedure and the 5th-order HJ-WENO scheme. In 3D, the accuracy and robustness is improved by employing a proposed multiple junction treatment and the directional optimization algorithm. 2D and 3D test cases demonstrate that our method exhibits 4th-order accuracy for reinitializing the regional level-set functions and strictly satisfies the interface-preserving property. The reinitialization results for more complex cases with randomly generated diagrams show the capability our method for arbitrary N cases with a computational effort that effectively is independent of N. The time-marching regional level-set reinitialization method has been applied successfully to level-set motion with different types of flows to assess its robustness and accuracy.

Acknowledgment

This work is supported by China Scholarship Council under No. 201306290030 and No. 201406120010. XYH acknowledges funding from National Natural Science Foundation of China (No. 11628206) and Deutsche Forschungsgemeinschaft (HU 1527/6-1). NAA acknowledges funding from the European Research Council (ERC) under the Horizon 2020 grant agreement 667483. The authors thank Dr. Matt Elsey for his helpful suggestions on implementation of the directional optimization algorithm and the anonymous reviewers for their constructive comments on addressing the convergence issue in 3D.

References

[1] Doi Akio and Akio Koide. An efficient method of triangulating equi-valued surfaces by using tetrahedral cells. *IEICE Transactions on Information and Systems*, 74(1):214–224, 1991.

[2] John B Bell, Phillip Colella, and Harland M Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 85(2):257–283, 1989.

[3] Li-Tien Cheng and Yen-Hsi Tsai. Redistancing by flow of time dependent eikonal equation. *Journal of Computational Physics*, 227(8):4002–4017, 2008.

[4] David L Chopp. Some improvements of the fast marching method. *SIAM Journal on Scientific Computing*, 23(1):230–244, 2001.

[5] Antoine du Chéné, Chohong Min, and Frédéric Gibou. Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes. *Journal of Scientific Computing*, 35(2):114–131, 2008.

[6] Matt Elsey and Selim Esedoglu. Fast and accurate redistancing by directional optimization. *SIAM Journal on Scientific Computing*, 36(1):A219–A231, 2014.

[7] Frederic Gibou, Ronald Fedkiwc, and Stanley Osher. A review of level-set methods and some recent applications. *Journal of Computational Physics*, in press, 2017.

[8] LH Han, XY Hu, and NA Adams. Adaptive multi-resolution method for compressible multi-phase flows with sharp interface model and pyramid data structure. *Journal of Computational Physics*, 262:131–152, 2014.

[9] Daniel Hartmann, Matthias Meinke, and Wolfgang Schröder. The constrained reinitialization equation for level set methods. *Journal of Computational Physics*, 229(5):1514–1535, 2010.

[10] Marcus Herrmann. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *Journal of Computational Physics*, 227(4):2674–2706, 2008.

[11] Guang-Shan Jiang and Danping Peng. Weighted ENO schemes for Hamilton–Jacobi equations. *SIAM Journal on Scientific Computing*, 21(6):2126–2143, 2000.

[12] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987.

[13] Chohong Min. On reinitializing level set functions. *Journal of Computational Physics*, 229(8):2764–2772, 2010.

[14] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.

[15] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[16] Stanley Osher and Chi-Wang Shu. High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations. *SIAM Journal on Numerical Analysis*, 28(4):907–922, 1991.

[17] Shucheng Pan, Xiangyu Hu, and Nikolaus A Adams. High-resolution transport of regional level sets for evolving complex interface networks. *arXiv preprint arXiv:1702.02880*, 2017.

[18] Giovanni Russo and Peter Smereka. A remark on computing distance functions. *Journal of Computational Physics*, 163(1):51–67, 2000.

[19] RI Saye and James A Sethian. Analysis and applications of the Voronoi implicit interface method. *Journal of Computational Physics*, 231(18):6051–6085, 2012.

[20] Robert I Saye and James A Sethian. The Voronoi implicit interface method for computing multiphase physics. *Proceedings of the National Academy of Sciences*, 108(49):19498–19503, 2011.

[21] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[22] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439–471, 1988.

[23] David P Starinshak, Smadar Karni, and Philip L Roe. A new level set model for multimaterial flows. *Journal of Computational Physics*, 262:1–16, 2014.

[24] Mark Sussman and Emad Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on Scientific Computing*, 20(4):1165–1191, 1999.

[25] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1):146–159, 1994.

[26] Hong-Kai Zhao, Tony Chan, Barry Merriman, and Stanley Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127(1):179–195, 1996.

[27] Hongkai Zhao. A fast sweeping method for eikonal equations. *Mathematics of Computation*, 74(250):603–627, 2005.

[28] Wen Zheng, Jun-Hai Yong, and Jean-Claude Paul. Simulation of bubbles. *Graphical Models*, 71(6):229–239, 2009.