Fakultät für Maschinenwesen

Lehrstuhl für Aerodynamik und Strömungsmechanik

Numerical methods for multi-region interfacial flows

Shucheng Pan

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. Oskar J. Haidn
Prüfer der Dissertation:	1. Priv.-Doz. Dr.-Ing. habil. Xiangyu Hu
	2. Prof. Feng Xiao, Ph.D.
	3. Prof. Dr.-Ing. Nikolaus A. Adams

Die Dissertation wurde am 11.04.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 12.07.2018 angenommen.

Declaration of Authorship

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Shucheng Pan
April 11, 2018

shucheng.pan@tum.de

Released April 11, 2018
Typesetting LaTeX

Abstract

This cumulative thesis is devoted to the modeling and simulation of multi-region interfacial flows where more than two regions/phases and interface networks are involved. In particular, a high-resolution regional level-set method is proposed to capture the evolution of interface networks in multi-region systems, a high-order time marching method is developed to compute distance function for the regional level-set function, and a conservative interface-interaction method is developed to solve compressible multi-material flows.

Simulating time evolution of a system with a large number of regions is a challenging task due to the existence of interface networks, singular structures, and complex topology changes, which pose a serious accuracy and efficiency limitation for traditional interface capturing methods. The first part of this thesis contributes to developing a robust interface-network capturing method with significantly improved accuracy and computational efficiency. Using the regional level-set function to represent the multi-region systems, the memory overhead is significantly alleviated. In detail, this method contains three novel numerical strategies: (i) employing any high-order spatial discretization scheme by temporarily generating local level-sets via a simple construction operation, (ii) local transporting the interface network by solving multiple advection equations for improved representation of complex topologies, and (iii) avoiding the numerical inconsistency, e.g overlaps and voids, by applying an efficient reconstruction operation. A number of two and three dimensional numerical examples are considered to demonstrate that the present method is able to simulate a wide range of multi-region evolution problems with good robustness, high accuracy, and low computational costs.

The second part of this thesis considers to improve the accuracy of the reinitialization step in solving the multi-region system evolution. Thus a high-order time-marching numerical method is developed for redistancing the regional level-set function. This method employs the closest point finding procedure and the iterative updating procedure for different types of cells. The main features of this method include (i) high-order accuracy, (ii) simplified updating of the regional level-set function by imposing the interface-preserving property, and (iii) low computational cost which is independent of the number of regions N. In addition, to avoid convergence failure of the closest point finding procedure in three dimensions, a novel multiple junction treatment is proposed, and the directional optimization algorithm is applied near the multiple junctions. The reinitialization results for various complex cases demonstrate high accuracy and robustness of the proposed method for arbitrary number of regions N.

With the proposed high-resolution transport formulation of the regional level-set function, a conservative interface-interaction method is developed to solve compressible multi-material flows with $N > 2$. The interface network is accurately represented and evolved by the high-resolution regional level-set method. Previous two-phase conservative sharp-interface method is extended to cases with $N > 2$ by proposing a conservation correction and a simplified interface-interaction calculation in each multi-material cell. In this way, conservation is strictly maintained,

with the sharp-interface treatment of different materials. For stabilizing the high-resolution computer simulations of the complex multi-material flows, a multi-material interface scale separation model is proposed to consistently remove non-resolved interface structures. Accuracy, capability and robustness of this method are demonstrated by a range of numerical examples.

Acknowledgements

First and foremost, I would like to express my deep and sincere gratitude to Prof. Nikolaus A. Adams and PD Dr. Xiangyu Hu for their continuous guidance, kindly suggestions, and warm encouragement during my PhD study. Xiangyu is always available once I have questions. I am very grateful to you for advising this work thoroughly, inspiring me with new ideas, and encouraging me when I was frustrated. I am glad that our discussions on a broad range of research subjects finally lead to some scientific publications. I greatly appreciate Prof. Adams for providing feedback for any research progress through group meeting and email exchanges, for carefully revising my papers, and for allowing me to change research topics according to my personal interests. I have been fortunate to benefit from their great visions of research, passions in science, and deep understanding of fluid mechanics. Their advices exceptionally inspire and enrich me in my research career. I also want to thank Prof. Oskar J. Haidn for taking over the chair of my doctoral examination, and Prof. Feng Xiao for reviewing my work as a thesis committee member. I am grateful for your time reading this thesis and for your wonderful feedback.

It is always a pleasure to work with my collaborators in the Chair of Aerodynamics and Fluid Mechanics, Technical University of Munich, including Luhui Han, Xiuxiu Lyu, Jianhang Wang, Krishnamurthy Viswanathan, and Dr. Stefan Adami. Luhui helps me in studying the multi-resolution method, and works with me on numerical modelling of compressible multi-materials flows. From Jianhang, I learn numerical details on many aspects of combustion modelling. I appreciate fruitful discussions and creative collaboration with you during the last two years. Stefan, thank you for inviting me to join in one project of your "Nanoshock" group. It is a pleasure to have the chance to work with you on modelling shock-driven biomedical processes. Besides, many thanks to you for organizing two wonderful workshops in Spitzingsee and Kloster Scheyern, where I enjoy useful discussion with "Nanoshock" members (Jakob Kaiser, Aleksandr Lunkov, Vladimir Bogdanov, Nils Hoppe, etc), interesting scientific games, and good beer. Moreover, collaborating with Xiuxiu on numerical simulations of cavitation, and with Krishnamurthy on studying vorticity production mechanism, are also acknowledged. My research vision has been greatly broadened during all the collaboration, and I hope that our collaboration will continue in the future.

In addition, I truly enjoyed my time as a PhD candidate in this chair, and interact with friendly and talented people here. I wish to express my gratitude to other members of our chair for their helpful discussions on the research and accompany during the study: Dr. Jian Luo, Chi Zhang, Yujie Zhu, Xiaobing Hu, Lu Miao, Zhe Ji, Zheng Fang, Zhaoguang Wang, Dr. Rongzhong Huang, and many others. Particularly, I would like to thank my previous and current officemates, Dr Xiaopeng Xue, Dr Xin Bian, Ludger Pähler, Li Su, and Yue Li. The working environment and atmosphere in room *MW 1611* have always been suitable for research. Xin, thanks for all your helpful advices on my research and for sharing interesting stories of your research

career. Ludger, I am impressed by your knowledge of scientific communities and your self-motivation. I really enjoy working with you until very late and sharing our research experience. I also want to extend my grateful thanks to staff members of our chair, Amely Schwörer, Li Su, Angela Grygier, Hans-Gerhard Frimberger, Dr. Christian Stemmer, and many others, for providing necessary equipments, maintaining computer systems, preparing documents, reimbursing conference costs, and other supporting. I am deeply indebted to the Chinese Scholarship Council for providing the financial support of my research in Technical University of Munich.

My gratitude also goes to many other friends in Germany. I would like to give special mention to Wenfeng Li, my previous roommate, colleague, collaborator, and friend during my Master study, previously a PhD student in Institute of Aerodynamics of RWTH Aachen University, and now a Postdoctoral researcher there. It is really helpful to deal with homesickness by sharing interesting research work and exchanging our PhD progress from both side every weekend.

A very special acknowledgement goes to Prof. Jinsheng Cai, my master supervisor at Northwestern Polytechincal University, Xi'an, China. Thank you for convincing me to study fluid mechanics, supervising my Master thesis, and being the enlightener of scientific research.

In the end, I am thoroughly grateful to my parents to whom I dedicate this dissertation. Without their endless love, encouragement and understanding during my study, this thesis would never has been completed.

Contents

List of Symbols

a	CFL number
E	total energy density
e	internal energy density
i, j, k	index of grid points in the x, y, and z directions
J	Jacobian
J	multiple junction
H	Heaviside function
K	band width
t	time
N	number of regions
N	number of grid points
n	index of time steps
p	pressure
tr	trace
α	volume fraction
κ	mean curvature
κ_G	Gaussian curvature
κ_1, κ_2	principal curvatures
μ	viscosity
σ	surface tension coefficient
τ	viscous stress tensor
δ	Dirac-delta function
γ	ratio of specific heats
ρ	mass density
Δt	time step size
$\Delta x,\ \Delta y,\ \Delta z$	grid spacing sizes in the x, y, and z directions
Ω	flow domain
Γ	interface
ϕ	level-set function
$\mathbf{x}$	location of a point in the domain
$\mathbf{n}$	normal direction
$\mathbf{u}$	velocity vector
$\mathbf{g}$	gravity vector
$\mathbf{K}$	curvature tensor
$\mathbf{U}$	flow state tensor
$\mathbf{F}$	convective flux tensor
$\hat{\mathbf{X}}$	interface flux

Chapter 1

Introduction

1.1 Multi-region systems

Many systems in science and engineering can be viewed as a union of multiple regions. When the number of region $N > 2$, these multi-region systems contain interface network that separates connected regions, and multiple junctions where more than two regions meet. The interactions between different regions, determined by the local mechanical, physical and biological properties, drive the motion of interface.

1.1.1 Typical examples

Foam is an ubiquitous multi-region system where a random coarse dispersions of gas bubbles is separated by thin liquid films [1]. In a soap foam, the thin film of liquid surrounded by air forms a complex topology as shown in Fig. 1.1(a). A dry foam (referred as foam hereafter) contains four typical elements: (i) the air which is both inside and outside the bubble, (ii) the curved film which is liquid-containing

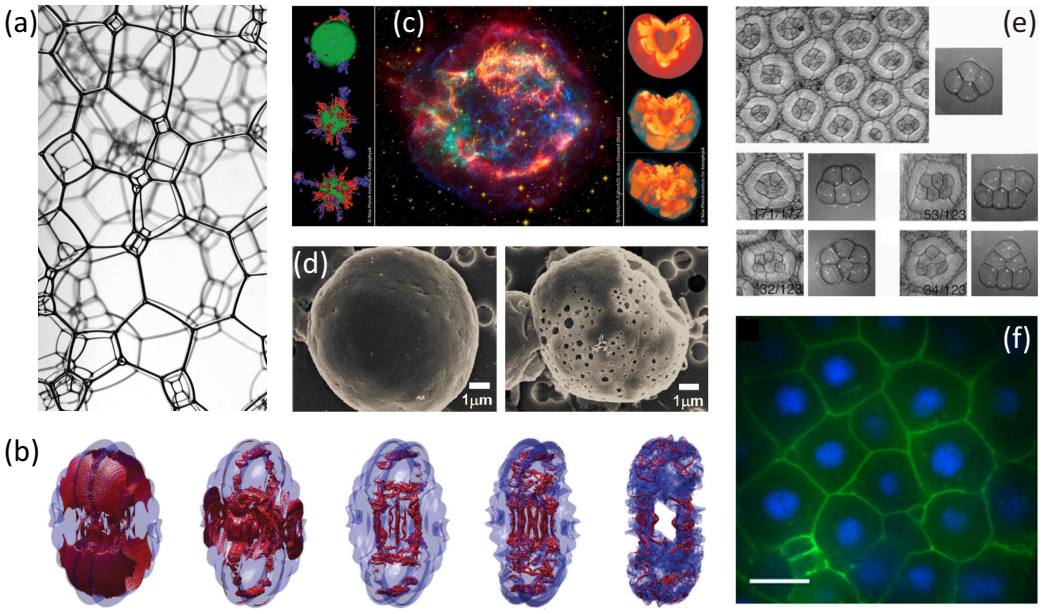


FIGURE 1.1: (a) A typical soap foam, (b) shock and interface evolution during the inertial confinement fusion (reproduced from Ref. [2]), (c) computer simulations of supernova explosions (reproduced from Ref. [3]), (d) leukemia cell damage during sonoporation (reproduced from Ref. [4]), (e) groups of cone cells inside each facet of *Drosophila* compound eye (reproduced from Ref. [5]), and (f) ectoderm cells of a zebrafish aggregate (reproduced from Ref.[6]).

membrane separating the air, and (iii) the Plateau border which is shared by three film patches, (iv) the vertex at which four Plateau border meet. The interactions between multiple gas-filled bubbles, liquid-filled films and Plateau borders complicate the dynamics of a foam. Many high energy processes occur in multi-region systems, such as the inertial confinement fusion, supernova explosion, and ultrasound driven biophysical phenomena, see Figs. 1.1(b), (c), and (d), respectively. Additionally, living cells and multicellular tissue exhibit similar multi-region structures, as shown in Figs. 1.1(e) and (f).

1.1.2 Interface networks

Let the domain Ω be an open set in $\mathbb{R}^d$, let $\mathbf{x} \in \Omega$ be an interior point of Ω, and let $\partial\Omega$ be the boundary of the domain, where d is the spatial dimension. Assuming that there are N regions within this domain, we define the index set for all regions as $\mathrm{X} = \{r \in \mathbb{N} | 1 \leqslant r \leqslant N\}$. Canonical two-region systems have regular shape, and only contains two subdomains, Ω^1 and Ω^2 (can be non-simply connected), and the interface Γ that separating them. While the multi-region system can be abstracted by a more complex structure which contain the following elements

- The region domain Ω^r which contains all interior points of the region r.
- The region boundary $\partial\Omega^r$ which contains the set of boundary points of Ω^r.
- The pairwise interfaces $\Gamma_{ab} = \partial\Omega^a \bigcap \partial\Omega^b$ which separates two connected regions a and b, with $a, b \in \mathrm{X}$ and $a \neq b$.
- The interface network $\Gamma = \bigcup_{a \neq b} \Gamma_{ab}$ is the union of all pairwise interfaces.
- Multiple junctions (high order junctions) $\mathrm{J} = \bigcap_{a \neq b} \Gamma_{ab}$ are intersections of pairwise interfaces. In most practically relevant applications they are triple points (2D) or triple lines (3D).

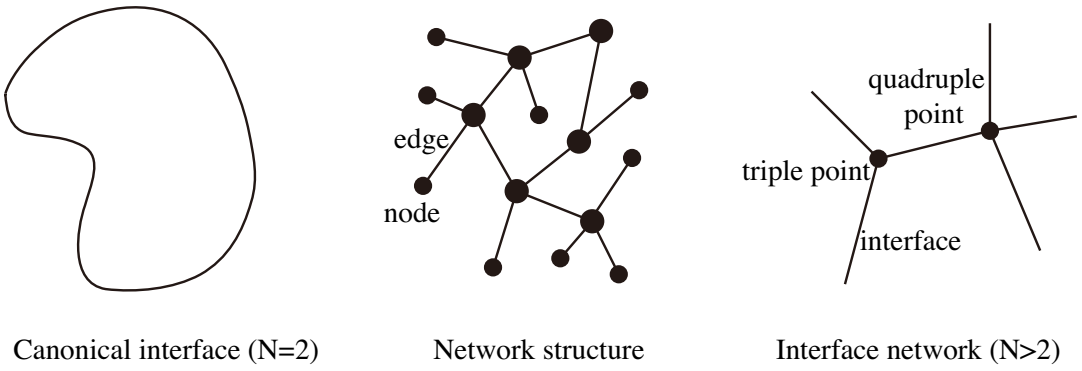


FIGURE 1.2: A schematic representation example of interface structure in two-region systems, typical network structure, and interface network in multi-region systems.

The interface structures of multi-region systems are different with those in two-region situations. As shown in Fig. 1.2, the interface of a canoical two-region system ($N = 2$) is a codimension manifold in $\mathbb{R}^d$. While the interface network of a multi-region system ($N > 2$) resembles a typical network structure while the edges are connected by the nodes. Particularly for a 2D interface network in Fig. 1.2, the multiple junctions and pairwise interfaces mimic the nodes and edges of a network structure, respectively.

1.2 Previous numerical methods

A range of numerical methods have been proposed to compute the evolution of interface networks for multi-region problems. Basically, these methods can be classified as Lagrangian or Eulerian methods according to the representation of the interface and how the interface elements are evolved.

1.2.1 Lagrangian methods

In Lagrangian methods, interface is represented explicitly by conforming discretization elements, and the coordinates of interface elements are updated according to the driven velocity fields. These methods, such as the front-tracking [7], immersed-boundary [8], and arbitrary Lagrangian–Eulerian (ALE) [9] methods, have been successfully extended to simulate interface-network evolution of multi-region systems [10, 11, 12, 13, 14, 15]. However, it is difficult to handle complex topological changes during interface-network evolution, especially in three dimensions.

1.2.2 Eulerian methods

Unlike the Lagrangian methods, in Eulerian methods the interface is reconstructed from scalar fields such as the volume-fraction field. The mostly widely used methods are the volume-of-fluid (VOF) [16] and the level-set method [17]. The interface is represented implicitly and captured by solving the corresponding transport equations. Compared to Lagrangian methods, these methods are more robust, as topological changes are automatically handled during interface evolution, and more straightforward to implement in three dimensions. The main drawback is that they usually generate large numerical dissipation due to the transport-equation discretization.

1.3 Objectives

The fundamental objective of the present work is to address the issues of previous numerical methods for multi-region problems. The proposed method should be sufficiently accurate for evolving complex interface networks, with the computation and memory costs being irrespective of the number of regions. Specifically, a recently developed method inherits the advantages of the original level-set method and the regional level-set method. This work is detailed in Paper I

- S Pan, XY Hu, NA Adams, High-resolution method for evolving complex interface networks, *Comput. Phys. Commun*, **225**, 10-27, 2018,

which has been attached in Section A.1.

As the reinitialization step is frequently required by the level-set method, especially for cases with complex topology, the accuracy of any level-set based method is dominated by that of the employed reinitialization method. The next objective is to develop a high-order reinitialization method to redistance the level-set function. Paper II,

- S Pan, X Lyu, XY Hu, NA Adams, High-order time-marching reinitialization for regional level-set functions, *J. Comp. Phys*, **354**, 311-319, 2018,

provides a time-marching reinitialization method for distance computing in multi-region systems. High-order accuracy is achieved, with a computational effort independent of N.

The final objective of this thesis is to develop a numerical method for compressible multi-material flows. Although the interface-network capturing method in Paper I can be easily coupled with the ghost fluid method [18], this non-conservative formulation may lead to wrong shock locations or nonlinear numerical instability [19] by violating the Lax-Wendroff theorem. Thus a fully conservative method, with suitable treatment of material interactions inside multi-material cells, are urgently required for solving compressible multi-material flows. The author has developed such method in Paper III,

- S Pan, L Han, XY Hu, NA Adams, A conservative interface-interaction method for compressible multi-material flows, *J. Comp. Phys*, **in press**, 2018,

1.4 Outline

This thesis is structured as follows. In Chapter 2, the author revisits the levelset method and the existing reinitialization methods which are used for solving interface evolution problems. The conservative sharp-interface method [20] for two-phase compressible flows is briefly reviewed in Chapter 3. The main work of this thesis is focused on solving numerical and physical problems related to multi-region interfacial flows. Particularly, as detailed in Chapter 4, a high-resolution multi-region level-set method [21] is proposed to solve interface-network evolution of multi-region systems, a high-order two-step regional level-set reinitialization method [22] is developed to compute distance function in multi-region systems, and a conservative interface-interaction model [23] is developed to simulate compressible multi-material flows. Finally, in Chapter 5 the state of proposed methods is concluded and potential applications in the future are discussed.

Chapter 2

The level-set method

2.1 The level-set function

2.1.1 Interface representation

The level-set method, proposed in the seminal paper by Osher & Sethian [17], is a popular interface capturing method in fluid mechanics, image processing, material science, and shape optimization [24, 25, 26]. Using the level-set function $\phi : \mathbb{R}^d \to \mathbb{R}$, the interface Γ or the boundary of the geometry is implicitly embedded as the zero contour $\Gamma := \{\mathbf{x} \in \mathbb{R}^d | \phi(\mathbf{x}) = 0\}$, where d is the spatial number. The domain Ω is partitioned into $\Omega^+ := \{\mathbf{x} \in \mathbb{R}^d | \phi(\mathbf{x}) > 0\}$ and $\Omega^- := \{\mathbf{x} \in \mathbb{R}^d | \phi(\mathbf{x}) < 0\}$. Usually the level-set function is chosen to be the sign distance function

$$\phi(\mathbf{x}) = \pm dist(\mathbf{x}) = \pm \min |\mathbf{x} - \mathbf{x}^*|,$$

to improve numerical accuracy and stability [17, 24, 25], where $\mathbf{x}^* \in \Gamma$ is the point located at the interface. The sign depends on whether the point $\mathbf{x}$ belongs to Ω^+ or Ω^-.

2.1.2 Geometric information calculation

In addition, geometric information of the interface can be accurately calculated using the level-set function. For example, the normal direction is obtained by

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|},$$

where the gradient $\nabla \phi$ is numerically approximated by finite difference schemes, e.g. the derivative $\left(\frac{\partial \phi}{\partial x}\right)_i$ at the grid point i can be computed by

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x},$$

where Δx is the grid spacing in the x direction. Then the curvature tensor $\mathbf{K}$, defined as the Jacobian of $\mathbf{n}$, becomes

$$\mathbf{K} = J(\mathbf{n}) = \begin{pmatrix} \frac{\partial \mathbf{n}_x}{\partial x} & \frac{\partial \mathbf{n}_x}{\partial y} & \frac{\partial \mathbf{n}_x}{\partial z} \\ \frac{\partial \mathbf{n}_y}{\partial x} & \frac{\partial \mathbf{n}_y}{\partial y} & \frac{\partial \mathbf{n}_y}{\partial z} \\ \frac{\partial \mathbf{n}_z}{\partial x} & \frac{\partial \mathbf{n}_z}{\partial y} & \frac{\partial \mathbf{n}_z}{\partial z} \end{pmatrix},$$

where $\mathbf{n}_x$, $\mathbf{n}_y$, and $\mathbf{n}_z$ are the normal direction components in the x, y, and z directions, respectively. The mean curvature κ is the trace of the curvature tensor $\mathbf{K}$,

$\kappa = tr(\mathbf{K})$, and the Gaussian curvature κ_G is the determinant of the matrix $\mathbf{A}$

$$\kappa_G = |\mathbf{A}| = |\mathbf{K} + \mathbf{n} \otimes \mathbf{n}|.$$

As the principal curvatures κ_1 and κ_2 satisfy the equalities, $\kappa_1 + \kappa_2 = 2\kappa$ and $\kappa_1\kappa_2 = \kappa_G$, the principal curvatures are computed by

$$\kappa_{1,2} = \kappa \pm \sqrt{\kappa^2 - \kappa_G}.$$

2.2 The level-set evolution equation

For a moving interface $\Gamma(t)$, the level-set method captures its evolution by numerically solving an advection equation,

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

where the velocity field $\mathbf{u} = (u, v, w)$ is determined by geometric information or by solving additional physical equations. Note that in some specific cases, the velocity is only available at the interface which means one needs extend the velocity to at least a narrow band around the interface [27]. One can rewrite Eq. (2.1) as a 1st-order Hamilton-Jacobi (HJ) equation

$$\frac{\partial \phi}{\partial t} + u_n |\nabla \phi| = 0, \tag{2.2}$$

where u_n is the velocity component in the normal direction of Γ. The level-set method belongs to the Eulerian interface evolution method, and unlike some interface tracking methods mentioned in Chapter 1, it does not require explicit discretization of the interface. In this way, the method becomes robust and flexible, as complex topology changes are automatically handled.

2.3 Numerical discretization

2.3.1 Space discretization

Suppose the Explicit Euler scheme is used for time marching, the semi-discrete form of Eq. (2.1) is

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + u^n \left(\frac{\partial \phi}{\partial x}\right)^n + v^n \left(\frac{\partial \phi}{\partial y}\right)^n + w^n \left(\frac{\partial \phi}{\partial z}\right)^n = 0, \quad \phi^n = \phi(t^n) \tag{2.3}$$

where the time-step Δt is determined by the Courant-Friedrichs-Lewy (CFL) condition,

$$\Delta t = \frac{a}{\max\left(|u|/\Delta x + |v|/\Delta y + |w|/\Delta z\right)}$$

with $0 < a < 1$ being the CFL number. The derivatives in Eq. (2.3) can be numerically approximated by upwind-biased finite difference schemes [1] like the essentially non-oscillatory (ENO) [32, 33] and weighted essentially non-oscillatory (WENO) [34,

[1] Other stable schemes such as the Semi-Lagrangian method [28, 29, 30, 31] has also been applied to solve the level-set evolution equation.

35, 36] schemes used in shock capturing problems. The discrete form of Eq. (2.2) is

$$\frac{\phi^{n+1}-\phi^n}{\Delta t} + u_n^n H_G(D_x^+\phi, D_x^-\phi, D_y^+\phi, D_y^-\phi, D_z^+\phi, D_z^-\phi) = 0,$$

where H_G is the Godunov numerical Hamiltonian [17] [2],

$$H_G(a,b,c,d,e,f) = \begin{cases} \sqrt{\max(|a^-|^2,|b^+|^2)+\max(|c^-|^2,|d^+|^2)+\max(|e^-|^2,|f^+|^2)} \\ \qquad\qquad \text{if } u_n^n > 0 \\ \sqrt{\max(|a^+|^2,|b^-|^2)+\max(|c^+|^2,|d^-|^2)+\max(|e^+|^2,|f^-|^2)} \\ \qquad\qquad \text{otherwise} \end{cases}$$

with $f^+ = \max(f,0)$ and $f^- = \min(f,0)$. The forward and backward derivatives in H_G are obtained by HJ-ENO [37] or HJ-WENO [38] schemes. For example, one can approximate the derivative $D_x^+\phi$ at the grid point (i,j,k) by the 5th-order HJ-WENO scheme,

$$\begin{aligned} D_x^+\phi &= w_0\frac{2\Delta_x^+\phi_{i-3,j,k} - 7\Delta_x^+\phi_{i-2,j,k} + 11\Delta_x^+\phi_{i-1,j,k}}{6\Delta x} \\ &+ w_1\frac{-\Delta_x^+\phi_{i-2,j,k} + 5\Delta_x^+\phi_{i-1,j,k} + 2\Delta_x^+\phi_{i,j,k}}{6\Delta x} \\ &+ w_2\frac{2\Delta_x^+\phi_{i-1,j,k} + 5\Delta_x^+\phi_{i,j,k} - 1\Delta_x^+\phi_{i+1,j,k}}{6\Delta x}, \end{aligned}$$

where $\Delta_x^+\phi_{i,j,k} = \phi_{i+1,j,k} - \phi_{i,j,k}$. w_0, w_1, and w_2 are the weighting factors defined in Ref. [38].

2.3.2 Time discretization

To improve the accuracy in time, the strong stability-preserving (SSP) Runge-Kutta schemes [39] are usually employed. Consider the 3rd-order SSP Runge-Kutta scheme, the ODE $\frac{\partial\phi}{\partial t} = L(\phi)$ is updated from t^n to t^{n+1} by

$$\begin{aligned} \tilde{\phi}^{n+1} &= \phi^n + \Delta t L(\phi^n) \\ \bar{\phi}^{n+1} &= \frac{3}{4}\phi^n + \frac{1}{4}\tilde{\phi}^{n+1} + \frac{1}{4}\Delta t L(\tilde{\phi}^{n+1}). \\ \phi^{n+1} &= \frac{1}{3}\phi^n + \frac{2}{3}\bar{\phi}^{n+1} + \frac{2}{3}\Delta t L(\bar{\phi}^{n+1}) \end{aligned}$$

2.4 Improved methods

Although being successful in many applications, the level-set method has some intrinsic drawbacks, e.g. the non-conservation issue, which motivates the development of numerous numerical approaches to improve the level-set method, such as the particle level-set method [40, 41], the conservative level-set method [42, 43], and the hybrid level-set/VOF method [44, 45]. In addition, to increase the efficiency, the narrow-band technique [46] is usually employed for large-scale simulations, especially in three dimensions. In two dimensions, this treatment reduces the operation complexity from $\mathcal{O}(\mathrm{N}^2)$ to $\mathcal{O}(K\mathrm{N})$, where N is the number of grid points in any direction of the domain, and K is the band width.

[2] Other monotone schemes such as the Lax-Friedrichs numerical Hamiltonian [37] can also be used.

2.5 Coupling with fluid dynamics

The level-set method is straightforward to be coupled with external physics [24, 25, 26]. As this thesis focuses on fluid dynamics, the coupling between the level-set method and numerical methods for fluid dynamics is reviewed in this section.

2.5.1 Incompressible multi-phase flows

First, consider the two-phase incompressible flow problem [47]. The fluid motion is governed by the incompressible Navier-Stokes equations

$$\begin{aligned} &\nabla \cdot \mathbf{u} = 0 \\ &\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \frac{1}{\rho}\left(-\nabla p + \nabla \cdot \tau + \sigma\kappa\delta\mathbf{n}\right) + \mathbf{g} \end{aligned} \tag{2.4}$$

where ρ is the mass density, p is the pressure, σ is the surface-tension coefficient, δ is the Dirac-delta function of the distance to the interface, $\mathbf{g}$ is the gravity vector, and τ is the viscous stress tensor. According to the Newtonian fluid assumption, $\tau = -2/3\mu\nabla \cdot \mathbf{u}\,\mathbf{I} + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$, where μ is the viscosity of the fluid and $\mathbf{I}$ is the unit tensor. As the flow states of the two fluids are usually different, smoothing treatments are applied to regularize the discontinuity of the fluid properties (density and viscosity) across the interface,

$$\begin{aligned} \rho &= H(\phi)\rho_1 + (1 - H(\phi))\rho_2 \\ \mu &= H(\phi)\mu_1 + (1 - H(\phi))\mu_2, \end{aligned}$$

where $H(\phi)$ is the Heaviside function. Numerically, the Heaviside function and the Dirac-delta function are approximated by the smoothed ones,

$$H_\epsilon(\phi) = \begin{cases} 1 & \text{if } \phi < -\epsilon \\ \frac{1}{2} + \frac{\phi}{2\epsilon} - \frac{1}{2\pi}\sin(\frac{\pi\phi}{\epsilon}) & \text{if } -\epsilon \leqslant \phi \leqslant \epsilon, \\ 0 & \text{otherwise} \end{cases}$$

$$\text{and} \quad \delta_\epsilon(\phi) = \frac{dH_\epsilon(\phi)}{d\phi}, \quad \text{with} \quad \epsilon \sim \mathcal{O}(\Delta x)$$

which effectively introduce a thickness of the interface that is proportional to the spatial mesh size [47]. Eq. (2.4) can be solved by the projection method [48, 49]. Afterwards, the velocity field $\mathbf{u}$ is used to drive the interface evolution by solving Eq. (2.1) or Eq. (2.2) with the level-set method.

2.5.2 Compressible multi-phase flows

For two-phase compressible flows, fluids are assumed to be immiscible, and the dynamics are described by the mass, momentum, and energy conservation equations,

$$\begin{aligned} &\rho_t + \nabla \cdot (\rho\mathbf{u}) = 0 \\ &(\rho\mathbf{u})_t + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \tau + \sigma\kappa\delta\mathbf{n} \\ &(\rho E)_t + \nabla \cdot ((\rho E + p)\,\mathbf{u}) = \nabla \cdot (k\nabla T) + \nabla \cdot (\tau\mathbf{u}) + \sigma\kappa\delta\mathbf{u} \cdot \mathbf{n} \end{aligned} \tag{2.5}$$

where T is the temperature, E is the total energy density, and k is the thermal conductivity. To close the equations, different equation of state (EOS) is required, e.g.

- **Ideal gas** is modelled by $p(\rho, e) = (\gamma - 1)\rho e$, where e is the internal energy density and γ is the ratio of specific heats.
- **Stiff gas** is modelled by stiffened gas EOS, $p(\rho, e) = (\gamma - 1)\rho e - \gamma B$, where B accounts for the material stiffness.
- **Weakly compressible fluid** (water) is modelled by Tait's EOS, $p = B\left(\frac{\rho}{\rho_0}\right)^\gamma - B + p_0$, where ρ_0 and p_0 are the reference density and pressure.
- **Explosive product** is modelled by the Jones-Wilkins-Lee EOS

$$p = A_0 \exp\left(\frac{-R_1\rho_0}{\rho}\right)\left(1 - \frac{\rho}{R_1\rho_0}\right) + B_0 \exp\left(\frac{-R_2\rho_0}{\rho}\right)\left(1 - \frac{\rho}{R_2\rho_0}\right) + \Gamma_0\rho(e + e_0),$$

  where Γ_0, ρ_0, A_0, B_0, R_1, R_2, and e_0 are constants.
- **Solid** is modelled by the Mie-Grüneisen EOS, $p = p_r + \Gamma(\rho)\rho(e - \frac{1}{2}p_r\frac{\rho-\rho_0}{\rho\rho_0})$, where

$$p_r = \rho_0 c_0^2 \frac{1 - \rho_0/\rho}{\left(1 - S\left(1 - \rho_0/\rho\right)\right)^2} \quad \text{and} \quad \Gamma(\rho) = \frac{\rho_0}{\rho}\Gamma_0,$$

  with c_0 and S being constant parameters.

Unlike incompressible cases, coupling of the level-set method and compressible fluid dynamic methods requires specific strategies, otherwise it leads to spurious oscillations of pressure or velocity near the interface [50]. One popular method that addresses this issue is the ghost fluid method proposed by Fedkiw et al. in Ref. [18]. The basic idea is to explicitly consider a boundary condition for the fluid interface and construct the ghost cells. The pressure and velocity defined on ghost cells are defined by duplicating those of the other fluid, while the entropy is extrapolated from the real fluid domain [3].

[3] In some cases, the isobaric fix technique [51] is required to reduce the overheating errors.

Chapter 3

Two-phase sharp-interface method

Here we first review the conservative sharp-interface method developed by Hu et al. [20] for two-phase compressible flows. Some numerical treatments will be used to develop the multi-material flow solver in Paper III. Consider two-dimensional inviscid fluids for simplicity, the governing equation of the flow, Eq. (2.4), can be rewritten as a system of conservation laws

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = 0,$$

where $\mathbf{U} = [\rho, \rho u, \rho v, \rho E]^T$ denotes the vector of flow state variables, and $\mathbf{F}$ denote the convective flux tensor. For the two-phase problems, the interface $\Gamma(t)$ decomposes the computational domain Ω into two parts, $\Omega^1(t)$ and $\Omega^2(t)$, and evolves in time by the underlying fluid dynamics. For each fluid r, Eq. (3.1) is integrated over its the own subdomain $\Omega^r(t)$,

$$\begin{aligned}
&\int_n^{n+1} \int_{\Omega^r(t)} \left(\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} \right) \, dx \, dy \, dt \\
&= \int_n^{n+1} \int_{\Omega^r(t)} \frac{\partial \mathbf{U}}{\partial t} \, dx \, dy \, dt + \int_n^{n+1} \int_{\Omega^r(t)} \mathbf{F} \cdot \mathbf{n} \, dx \, dy \, dt = 0,
\end{aligned}$$

for time interval $t \in [t^n, t^{n+1}]$.

3.1 Numerical discretization

Let the flow equation defined on two-dimensional Cartesian grids. Then the discretized form of the governing equations for each of the phases in a finite-volume cell (i, j) using explicit Euler time-marching is given by

$$\begin{aligned}
\alpha_{i,j}^{n+1} \mathbf{U}_{i,j}^{n+1} &= \alpha_{i,j}^{n} \mathbf{U}_{i,j}^{n} + \frac{\Delta t}{\Delta x} \left(A_{i-1/2,j} \hat{\mathbf{F}}_{i-1/2,j} - A_{i+1/2,j} \hat{\mathbf{F}}_{i+1/2,j} \right) \\
&+ \frac{\Delta t}{\Delta y} \left(A_{i,j-1/2} \hat{\mathbf{F}}_{i,j-1/2} - A_{i,j+1/2} \hat{\mathbf{F}}_{i,j+1/2} \right) + \frac{\Delta t}{\Delta x \Delta y} \hat{\mathbf{X}}(\Delta \Gamma_{i,j}), \quad (3.1)
\end{aligned}$$

where $\alpha_{i,j}$ is the volume fraction of the respective fluid in this cell. $\Gamma_{i,j}$ and A stand for the interface segments inside this cell and the cell face after intersection by the interface Γ, respectively. The vector of conservative states $\alpha_{i,j}\mathbf{U}_{i,j}$ is defined at the cell center of the cell (i, j) for each fluid, where $\mathbf{U}_{i,j}$ is the vector of cell-averaged states. The numerical flux at the cell faces $\hat{\mathbf{F}}$ is interpolated from the physical fluxes using the WENO scheme [39], and the calculation of interface fluxes $\hat{\mathbf{X}}$ is detailed in Sec. 3.2. Note that away from the interface, the method recovers a standard finite volume

scheme on a two-dimensional Cartesian grid [20]. For each fluid, different EOS listed in Sec. 2.5.2 can be employed to close the governing equations. To increase accuracy for time marching, Runge-Kutta schemes can be used, e.g. the 2nd-order Strong Stability-Preserving Runge-Kutta scheme [39], with Eq. (3.1) being synonymous for a substep. The time-step Δt is constrained by the CFL condition,

$$\Delta t = a \ \min\left(\frac{\Delta x}{|u| + c}, \ \frac{\Delta y}{|v| + c}\right),$$

where c is the sound speed and $a = 0.6$.

3.2 Interface treatments

To ensure strict conservation for each fluid, maintain the sharp-interface property, and improve numerical stablity, Hu et.al [20] introduced several numerical procedures. First, the momentum and energy exchange across $\Gamma_{i,j}$, represented by last term on the right-hand-side of Eq. (3.1), $\hat{\mathbf{X}}(\Delta\Gamma_{i,j})$, are determined by a local Riemann problem. To solve this problem, iterative approximate Riemann solvers [52] or the Harten-Lax-van Leer solver with contact restoration (HLLC) [53, 54] are employed to compute the interface velocity $\mathbf{u}^* = (u^*, v^*)$ and pressure p^*. Thus the interface flux becomes

$$\hat{\mathbf{X}}(\Delta\Gamma_{i,j}) = \left[0 \ , p^*\Delta\Gamma_{i,j}\mathbf{n}_x \ , p^*\Delta\Gamma_{i,j}\mathbf{n}_y \ , p^*\Delta\Gamma_{i,j}\mathbf{n}\cdot\mathbf{u}^*\right]^T,$$

where the interface segment length $\Delta\Gamma_{i,j}$ is approximated by

$$\Delta\Gamma_{i,j} = \sqrt{\left(A_{i+1/2,j} - A_{i-1/2,j}\right)^2 + \left(A_{i,j+1/2} - A_{i,j-1/2}\right)^2}.$$

Note that before applying the Riemann solver and the interpolation of physical fluxes, the flow states defined on real-fluid domain are extrapolated to ghost-fluid domain by solve the extending equation [18]

$$\frac{\partial \mathbf{W}}{\partial t} \pm \mathbf{n}\cdot\nabla\mathbf{W} = 0,$$

where $\mathbf{W} = [\rho, u, v, p]^T$ is fluid state vector to be extended.

In addition, as the time-step Δt is determined by the full cell CFL condition, numerical instabilities may occur due to the existence of small cells (the volume fraction $\alpha \ll 1$). In order to maintain numerical stability without reducing the time step, the mixing procedure [20] is applied after each Runge–Kutta sub-step. This stabilization technique is simialr with cell-merging [55], cell-linking [56], and flux redistribution methods [57]. For each fluid, the conservative quantities of a cell with small volume fraction are merged with those of the cells with large volume fraction in a conservative way. During this procedure, the exchanges of the conservative quantities, $\mathbf{M}$, are calculated according to the averaged values, see Ref. [20]. Then the conservative quantities for each fluid in the near interface cells are updated by

$$\alpha_{i,j}^{n+1}\mathbf{U}_{i,j}^{n+1} \leftarrow \alpha_{i,j}^{n+1}\mathbf{U}_{i,j}^{n+1} + \sum\mathbf{M}_x + \sum\mathbf{M}_y,$$

where $\sum\mathbf{M}_x$ and $\sum\mathbf{M}_y$ represent the sum of all mixing exchanges of this cell in the x and y directions, respectively.

3.3 Two-phase multi-resolution method

Most simulations of this work require high mesh resolution to resolve small-scale features. To increase computational efficiency and reduce memory storage of these simualtions, Han et al. [58] have developed a multi-resolution method for solving two-phase compressible flows. In this method, the two-phase sharp-interface method [20] is employed to solve fluid dynamics. And the multi-resolution method of Harten [59] is incorporated for mesh refinement due to its higher rate of data compression [60] than adaptive mesh refinement method [61, 62] and rigorous regularity analysis [63]. The projection and prediction operators in Ref. [64] are used to perform the cell-averaged multi-resolution representation. For simplicity, the one-dimensional conservative projection and prediction operators are written as

$$P_{\ell}^{\ell+1}(\mathbf{U}_{\ell+1}): \quad \mathbf{U}_{\ell,i} = \frac{1}{2}(\mathbf{U}_{\ell+1,2i} + \mathbf{U}_{\ell+1,2i+1}),$$

and

$$\begin{aligned} P_{\ell+1}^{\ell}(\mathbf{U}_{\ell}): \qquad \hat{\mathbf{U}}_{\ell+1,2i} &= \mathbf{U}_{\ell,i} + \sum_{m=1}^{r} \beta_m(\mathbf{U}_{\ell,i+m} + \mathbf{U}_{\ell,i-m}), \\ \hat{\mathbf{U}}_{\ell+1,2i+1} &= \mathbf{U}_{\ell,i} - \sum_{m=1}^{r} \beta_m(\mathbf{U}_{\ell,i+m} + \mathbf{U}_{\ell,i-m}), \end{aligned}$$

where ℓ is index of levels, β_m is the interpolation coefficient of the $(2r+1)$-th order prediction. Mesh refinement and coarsening are triggered by comparing the prediction error $\mathbf{D}_{\ell,i} = \mathbf{U}_{\ell,i} - \hat{\mathbf{U}}_{\ell,i}$ with a level-dependent threshold $\epsilon_\ell = 2^{d(\ell-\ell_{\max})}\epsilon$, where ϵ is a user-defined parameter, and $\ell_{\max}$ is the maximum level of the adaptive data structure. For time marching, a Runge-Kutta local time stepping scheme [65] is employed to achieve time adaptivity and thus obtain additional speed-up. The conservative flux correction in Ref. [65] is adopted between different levels to ensure strict conservation.

3.4 Interface scale separation

The high-resolution simulations in this work usually generate numerous non-resolved interface segments, such as thin filaments and small droplets. To avoid proliferation of these artifacts, which may lead to numerical instability, a numerical procedure that can consistently remove non-resolved interface segments is required [66, 67]. Han el al. [68] recently developed an interface-scale-separation model based on the constrained stimulus-response procedure. The first step of this method is to identifying the "oddball cells" where non-resolved interface segments exist. This is achieved by shifting the level-set function to each fluid domain and checking the local connectivity of the interface structures. Then a reinitialization procedure is revoked on the modified level-set field, which automatically separates resolvable and non-resolvable interface scales. An efficient version of the interface-scale-separation model in Ref. [68] is developed by Luo et al. [69].

Chapter 4

Accomplishments

In this chapter, the state-of-the-art of relevant numerical methods is briefly reviewed, and the major achievements of this thesis are stated.

4.1 High-resolution method for evolving complex interface networks

Multi-region interfacial flows contain more than two immiscible fluids and the interface network that separates the different fluid regions. Here a numerical method developed in Paper I [21] for solving interface-network transporting is summarized.

4.1.1 State of the art

To predict the evolution of the interface network under interactions of the different fluids across interface segments in multi-region systems, a range of numerical models have been proposed in the literature. For Lagrangian methods mentioned in Sec. 1.2, although successful extension to multi-region systems has been reported [10, 12, 13, 14], they usually suffer from complex topological changes during interface-network evolution, especially in three dimensions. Eulerian methods, such as the volume-of-fluid method and the level-set method, offer an implicit way to capture the interface evolution. However, volume-of-fluid simulations of dynamical multi-region systems require complex reconstruction algorithm to detect sub-cell interface structures from the volume-fraction field [70, 71].

When applied to simulate the interface-network evolution in multi-region systems, two main level-set based approaches are developed, i.e., the multiple level-set method and the regional level-set method. The multiple level-set method, first developed by Merriman et al. in Ref. [72], defines multiple level-set functions which are assigned to corresponding regions. Then level-set transport equations are solved for all level-set functions to capture the evolution of multiple regions. Two numerical issues arise when this method is applied to multi-region problems. The first one is the memory and computation overhead due to the increased number of scalar fields [73]. The other issue, observed in many papers [74, 73, 75], is the spuriously generated voids and overlaps, due to the inconsistent numerical dissipation near the multiple junctions. To address these issues, various numerical methods have been developed in the literature. The method of Zhao et.al [74], named as the variational level-set method, applies the Lagrange multiplier to prevent the generation of voids and overlaps during the interface reconstruction. Starinshak et al. [75] propose to use a pairwise voting strategy to remove overlaps and voids. However, the algorithm requires $N(N-1)/2$ level-set functions to be stored and evolved in time. The method of Vese and Chan [76] reduces the number of level-set fields from N to

$\log_2 N$ [73], with the generation of voids and overlaps being remedied. These methods does not completely avoid artifacts or reduce the computation/memory cost to the same with of the original level-set method.

The optimal strategy to address the computation/memory overhead is using a single level-set function to represent arbitrary number of regions. In the community of image science, Lie et al. [77] and Chung and Vese [78] develop such methods for image segmentation. Recently, the regional level-set method [79] addresses the problem of multiple level-set fields by employing a single unsigned level-set field and an integer region indicator function. A low-order Semi-Lagrangian scheme is applied to capture the interface motion. The drawback is that this low-order Semi-Lagrangian scheme is more dissipative than high-order finite-difference schemes. The Voronoi implicit interface method [80, 81], as an improved regional level-set method, uses a transported ϵ-level-set and reconstructs the interface network by a reinitialization step based on Voronoi diagrams. This method usually requires frequently reinitialization which is significantly more costly than solving the level-set advection equation. The conflict between high accuracy and low computational efficiency has not been addressed for the regional level-set method.

4.1.2 Summary of the presented method

To overcome the drawbacks of previous regional level-set methods, i.e, high computational cost and low prediction accuracy, Paper I proposes a high-resolution interface-network capturing method based on the regional level-set function. This method inherits the advantages of the original level-set method and the low-order regional level-set method [79] but is modified in such a way that it is suitable for high- resolution finite-difference discretization of the level-set transport equations.

By defining cell neighborhoods and cell types and using the regional level-set function (a combination of a unsigned level-set function and an integer indicator function), the elements of a multi-region system, including the region domain, region boundary, pairwise interface, and interface network, are implicitly represented in an efficient way to reduce the computational effort significantly. Specifically, the square-shaped near neighborhood and the cross-shaped stencil neighborhood are defined for each finite-volume cell to detect potential indicator change during evolution and perform local construction of the auxiliary level-set functions, respectively. According to the number of regions existing in the square-shaped near neighborhood, a finite-volume cell is categorized as a cell with full type, with two-region type, or with complex-region type.

A simple construction operator is proposed to generate multiple local signed level-set fields from the global unsigned level-set field. Thus the normal direction and curvature can be computed by the constructed local level-set functions. And numerical approximation of derivatives in the level-set evolution operating on constructed local level-set functions in the cross-shaped stencil neighborhood permits the implementation of high-order schemes in a straightforward way.

According to the cell type, different strategies for solving the evolution equation are developed, which further increase the efficiency of the interface-network capturing method. For each local constructed level-set function, the corresponding local level-set advection equation is formulated and numerically solved by high-order schemes. This method recovers the high-order accuracy of the original level-set method for cells that are sufficiently far away from multiple junctions. After solving the advection equation we obtain multiple updated level-set values, defined at the cell center, with each being approximately the distance to the corresponding region

boundary. Paper I also proposes a physically reasonable reconstruction operator in the sense that no artificial overlaps or voids are generated when reconstructing the global unsigned level-set from multiple evolved local level-set functions. The combination of the construction and the reconstruction procedures is consistent for static interface networks.

Simple triple-point advection cases demonstrate that this interface-network capturing method has high-order accuracy away from the multiple junctions and 1st-order near the multiple junctions. A range of pure interface-network evolution test cases, such as the normal-driven flow and mean curvature flow, show that this method is accurate and robust. Its computational efficiency is approximately similar to the Semi-Lagrangian regional level-set method, with significantly improved accuracy. This method has been successfully coupled with a high-order finite-volume Navier-Stokes method to solve foam dynamics, and with the ghost fluid method to simulation multiple material shock-bubble interaction problem.

My contribution to this work was the development of the method and the corresponding computer code for its implementation. I have performed simulations, analyzed the results, and wrote the manuscript for the publication.

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

Multi-region level-set methods, such as the multiple level-set method and regional level-set method, require frequent redistancing of the level-set during interface-network evolution. This section provides an overview of level-set reinitialization, especially for multi-region systems, and a conclusion of a novel regional level-set reinitialization method developed in Paper II.

4.2.1 State of the art

In fact, as the signed distance function is not the solution of Hamilton-Jacobi equation, the level-set method does not intrinsically preserve the signed distance property during evolution [82, 83, 84]. Thus, the level-set contours near the interface become distorted, leading to large numerical errors. To address this issue, the level-set function needs to be reinitialized periodically throughout the simulation. Let the level-set function after the advection be ϕ^0. The objective of the reinitialization procedure is to ensure the function ϕ after reinitialization is the distance to the interface implicitly represented by $\Gamma^0 = \{\mathbf{x}|\phi^0(\mathbf{x}) = 0\}$, with the sign of ϕ being the same with that of ϕ^0.

A number of successful numerical methods have been developed for solving this problem, such as the fast marching method [85, 86, 87], the fast sweeping method [88, 89, 90], and the Hopf-Lax formula based method [91, 92, 93]. Indeed, all these methods solve the Eikonal equation

$$|\phi| = 1 \quad \text{with} \quad \text{sgn}(\phi) = \text{sgn}(\phi^0). \tag{4.1}$$

The most widely used method, introduced in Ref. [94], transforms the stationary boundary value problem (4.1) to a time-marching problem

$$\frac{\partial \phi}{\partial t} + \text{sgn}(\phi^0)(|\phi| - 1) = 0, \tag{4.2}$$

which is a HJ equation with a discontinuous coefficient across the interface. Compared to sequential-updating-based methods like the fast marching method, this time-marching method is considered to be more flexible and easier to parallelize [95]. However, as the numerical discretization may employ erroneous level-set information from the other side of the interface [96], this method generates distinct interface displacement which in turn leads to order degeneration of high-order schemes specially developed for HJ equations [38] and mass loss during the iterations [97]. Improvement of the methods has been proposed to cope with spurious interface oscillations in the literature [97, 96, 95, 98, 99, 100].

In the multiple level-set methods, the reinitialization procedure is usually applied to each level-set function [75]. While for regional level-set methods, the fast marching method has been employed [81], however, high-order accuracy has not been demonstrated in the literature for regional level-set reinitialization. In Paper I, the employed reinitialization method is limited to 1st-order accuracy without preserving the interface location, and its computational cost depends on the number of regions due to the region-by-region updating.

4.2.2 Summary of the presented method

The drawbacks of previous regional level-set reinitialization methods motivate the work in Paper II to achieve high order accuracy, preserve interface location, and reduce the computational cost. The proposed regional level-set reinitialization method is based on the time-marching PDE (4.2), which is flexible and easy to parallelize.

A high-order two-step time marching method is developed in Paper II, which is a combination of the closest point method and the numerical HJ method. During the fist step, like the fast marching method [85], the cell is tagged as *Alive* if it is close to the interface network, others as *Far*. For every *Alive* cell, multiple bicubic/tricubic polynomials are constructed using the local level-set functions obtained by applying the construction operator proposed in Paper I, implying a 4th-order accuracy near the interface. Then within the cells containing interface segments, a modified Newton method [101] is applied to find the closest points which are subsequently used to update the level-set values of *Alive* cells.

In the second step, a 5th-order HJ-WENO scheme is employed to iteratively solve the time-dependent reinitialization equation for *Far* cells. Here, unlike the interface-network capturing methods in Paper I, the construction operator is only revoked with respect to the indicator define on the cell center. This ensure the invariant of indicator function defined at *Far* cells. The data on *Alive* cells is frozen during iteration to achieve strict interface-preserving property.

In addition, to address the convergence issue encountered especially in three-dimensional cases during the closest finding procedure, Paper II has managed to incorporate the directional optimization method of Elsey & Esedoglu [102] with a modified formulation. An efficient multiple-junction treatment algorithm is proposed to handle potential convergence issue near multi-material cells, and to provide a fast and accurate explicit interface extraction method for post-processing of multi-region interface networks.

The proposed numerical method is validated by a range of static and dynamical reinitialization examples. It has been demonstrated that this 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.

My contribution to this work was the development of the method and the corresponding computer code for its implementation. I have performed simulations, analyzed the results, and wrote the manuscript for the publication.

4.3 A conservative interface-interaction method for compressible multi-material flows

In a broad range of scientific and engineering applications, especially high energy physics and astrophysics, compressible multi-material problems are commonly encountered, e.g. inertial confinement fusion (ICF) [103], core-collapse supernova [104], and hypervelocity impact [105]. The numerical method for solving these problems in Paper III is discussed in this section, with a brief review on related work in the literature.

4.3.1 State of the art

In compressible multi-material flows, different materials separated by the interface exhibit distinct material properties, e.g. viscosity and equation of state. Numerical modelling has received increasing attention in recent years due to its versatility as complement to experimental investigations. However, the discontinuity of density or pressure profile across the interface requires high accuracy of the numerical method to solve such complex flow field and interface evolution.

Among many well-established numerical approaches for two-phase compressible flows, such as the front-tracking method, the arbitrary-Lagrangian-Eulerian (ALE) method, the volume-of-fluid (VOF) and the level-set method, methods with sharp interface property have the advantage of introducing no numerical mixing artifacts for immiscible materials and fast shock-driven progresses of miscible materials. Compared to front-tracking and ALE methods, VOF and level-set methods are more robust as complex geometries of interfaces are automatically captured as due to implicit representation of the interface. Regarding the extension of the VOF method to multi-region flows, explicit interface reconstruction algorithms, such as the onion-skin [70] and serial-dissection [71], are required to map the volume-fraction field to the interface location, which are computationally costly and numerically inconsistent due to the material ordering strategy. Due to mixed treatment inside multi-material cells, the VOF method lacks the sharp interface property.

For multi-material flows containing more than two fluids and the interface network, in previous work Paper I, the coupling of the high-resolution regional level-set method and the ghost fluid method successfully capture interface-network evolution in a 3-material shock bubble interaction problem, with a sharp interface property. However, the conservation is violated, which may cause the wrong solution of shock-wave locations. Conservative interface interaction models have been developed for two-phase flows [20, 42, 54, 106, 107, 108, 58, 109] and solid/fluid interactions [106, 110, 111], however, have not yet been proposed for more than two fluids.

4.3.2 Summary of the presented method

To achieve the sharp-interface property and conservation for multiple compressible fluids with $N > 2$, Paper III provides a conservative interface-interaction method

based on the high-resolution interface-network capturing scheme proposed in Paper I. This method is robust, efficient, accurate, and has the ability to handle different types of materials, including perfect gases, stiffened gases, water-like fluids, detonation products, and solids.

First, the conservative sharp-interface method in Ref. [20] is extended to multiple materials. The finite-volume method and the high-resolution regional level-set method [21] are employed for solving the fluid dynamics and the interface-network evolution, respectively. Second, to avoid complex interface fluxes calculation and explicit interface reconstruction, a reduced interface-interaction model is proposed inside multi-material cells. The interaction between each connected material pair is computed by solving a local Riemann problem using the HLLC Riemann solver, following Ref. [54]. In addition, an efficient volume-fraction correction procedure is proposed to strictly preserve conservation for multi-material cells, without explicitly reconstructing the sub-cell structures. This method is successfully coupled with the multi-resolution method and local time stepping scheme to achieve significant speedup for high-resolution simulations.

One drawback of the conservative sharp-interface method in Ref. [20] is the numerical instabilities for high-resolution simulations due to the non-resolved interface segments. Thus, at a given spatial resolution non-resolved interfacial scales, such as thin filaments and small droplets, need to be removed in order to avoid proliferation of artifacts. Although the interface scale separation methods for two-phase flows have been developed [66, 67, 68, 69], currently there are no successful numerical methods for separating of resolvable and non-resolvable interface scales in flows involving more than 3 materials. Another achievement of Paper III is the developing a numerical procedure to consistently remove non-resolved interface segments and thus to increase the robustness of high-resolution low numerical dissipation multi-material simulations. Following the constrained stimulus-response concept in Ref. [68], the non-resolved interface structures in multi-material cells are deleted. Then new interfaces are constructed between these materials in a way that these different materials stay separated, however with a different connection relation.

A variety of numerical test cases, including the 3-material shock tube problem, inertial confinement fusion implosion, triple-point shock interaction, and shock interaction with multi-material bubbles, demonstrate that the proposed method is robust and accurate for simulating a wide range of complex compressible multi-material flows.

My contribution to this work was the development of the method and the corresponding computer code for its implementation. I have performed simulations, analyzed the results, and wrote the manuscript for the publication.

Chapter 5

Conclusions and outlooks

In this thesis, a series of numerical methods have been proposed to simulate the interface-network evolution, level-set reinitialization, and the compressible fluid dynamics in the multi-region interfacial flows where the number of phases (or fluids) is larger than 2. A high-resolution regional level-set method is developed for capturing the motion of interface networks with high accuracy and high efficiency. A high-order time-marching reinitialization approach is developed to impose the distance function property for the regional level-set function. In addition, a conservative interface-interaction model is proposed to simulate compressible material flows.

5.1 Concluding remarks

The drawbacks of previous multiple level-set methods and regional level-set methods, e.g. numerical artifacts generation, low accuracy, and high computational cost, are addressed by the proposed high-resolution regional level-set method. Using the regional level-set function, the multi-region system is efficiently represented. Unlike the multiple level-set functions used previously, the memory cost is approximately the same with that of the single level-set function for two-phase systems. By defining a construction operator on the regional level-set function and applying this operator on a small stencil of each finite-volume cell, classical high-order spatial discretization schemes are able to be applied to solve local evolution equations, which increases the accuracy of this method. By formulating a reconstruction operator based on the evolved local level-set values, the global regional level-set data is efficiently restored, without generating voids or overlaps. A number of multi-region validation tests involving interface networks show that the proposed method is robust, accurate, and capable of handling complex systems with arbitrary number of regions in an efficient way. Moreover, this method has been coupled with the ghost fluid method and successfully applied to solve the 3-material shock-bubble interaction problem.

Although the high-resolution regional level-set method mentioned above is suitable to solve complex multi-region systems evolution, its accuracy is dominated by the reinitialization procedure which is frequently revoked during simulations. Considering the high-order accuracy has not been achieved for reinitializing regional level-set function, a high-order time marching method is developed for computing distance in multi-region systems. The closet point finding algorithm and the 5th-order finite-difference WENO scheme are applied for redistancing in the *Alive* and *Far* finite-volume cells, respectively. In this way, the high-order accuracy and strict interface-preserving property are maintained. Also, an efficient multiple-junction reconstruction algorithm is developed to ensure convergence near multi-material cells. Two-dimensional and three-dimensional static and dynamic numerical example problems indicate that this method outperforms previous reinitialization methods for multi-region systems.

Based on the proposed interface-network capturing method in Paper I, a conservative interface-interaction method is developed to simulate compressible multi-material flows. Using the finite-volume method and an efficient volume-fraction correction algorithm, conservation of each fluid is strictly preserved. Complex interface fluxes calculation inside the multi-material cells are simplified by introducing a reduced interface-interaction model. Besides, this method achieve improved robustness in high-resolution simulations by developing a multi-material interface-scale separation method. Numerous challenging test cases indicate that this method has the intended properties and demonstrate that the method is very robust, flexible and efficient for simulations of compressible flows with multiple fluid-like materials.

5.2 Potential applications and future work

The main part of this thesis focuses on developing a numerical method for multi-region interfacial problems and its application in simulating multi-material flows. This work offers a general and flexible framework to represent the interface network and track its motion driven by external physics, implying many other potential applications. Several possibilities for future work are

- *Multi-phase image segmentation and multi-region shape optimization.* Similar to the numerical cases in this thesis, the image segmentation and shape optimization problems requires evolution of the interface networks embedded in a multi-region system subject to minimizing specific energy functionals. For example, prevoius work uses the multiple level-set method for multi-phase image segmentation [76] and shape optimization [112, 113]. With the method presented in this thesis, one may achieve additional efficiency and accuracy in solving these problems.

- *Domain decomposition.* Domain decomposition can be viewed as a multi-region problems. As pointed out by Zhao et al. [74], the density of the computational workload on each computing node and the density of communication data drive the motion of the boundary of each region which represent the union of assigned to the corresponding node. In Ref. [114], the interface-network capturing method of Paper I has been successfully applied to solve domain decomposition problems for block-structured meshes. Additionally, as mentioned in Ref. [114], by replacing the energy functional, this method can be applied to other applications such as the development of multi-scale coupling frameworks [115, 116].

- *Fluid-structure interaction.* Although the work in this thesis is limited to simulation of pure fluid materials, the proposed method can be extended to more complex materials by adding additional physical models, such as elastic-plastic model for fluid-structure interaction problems or cell membrane models for simulating biological systems.

Bibliography

[1] Denis L Weaire and Stefan Hutzler. *The physics of foams*. Oxford University Press, 2001.

[2] Brian M Haines, Fernando F Grinstein, and James R Fincke. "Three-dimensional simulation strategy to determine the effects of turbulent mixing on inertial-confinement-fusion capsule performance". In: *Physical Review E* 89.5 (2014), p. 053302.

[3] NJ Hammer, H-Th Janka, and E Müller. "Three-dimensional simulations of mixing instabilities in supernova explosions". In: *The Astrophysical Journal* 714.2 (2010), p. 1371.

[4] Wenjing Zhong et al. "Sonoporation induces apoptosis and cell cycle arrest in human promyelocytic leukemia cells". In: *Ultrasound in Medicine and Biology* 37.12 (2011), pp. 2149–2159.

[5] Takashi Hayashi and Richard W Carthew. "Surface mechanics mediate pattern formation in the developing retina". In: *Nature* 431.7009 (2004), p. 647.

[6] M Lisa Manning et al. "Coaction of intercellular adhesion and cortical tension specifies tissue surface tension". In: *Proceedings of the National Academy of Sciences* 107.28 (2010), pp. 12517–12522.

[7] Salih Ozen Unverdi and Grétar Tryggvason. "A front-tracking method for viscous, incompressible, multi-fluid flows". In: *Journal of Computational Physics* 100.1 (1992), pp. 25–37.

[8] Charles S Peskin. "The immersed boundary method". In: *Acta numerica* 11 (2002), pp. 479–517.

[9] CW Hirt, Anthony A Amsden, and JL Cook. "An arbitrary Lagrangian-Eulerian computing method for all flow speeds". In: *Journal of Computational Physics* 14.3 (1974), pp. 227–253.

[10] Kenneth A Brakke. "The surface evolver". In: *Experimental Mathematics* 1.2 (1992), pp. 141–165.

[11] Stéphane Galera, Pierre-Henri Maire, and Jérôme Breil. "A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction". In: *Journal of Computational Physics* 229.16 (2010), pp. 5755–5787.

[12] Raphaël Loubère et al. "ReALE: A reconnection-based arbitrary-Lagrangian–Eulerian method". In: *Journal of Computational Physics* 229.12 (2010), pp. 4724–4761.

[13] Milan Kucharik et al. "A comparative study of interface reconstruction methods for multi-material ALE simulations". In: *Journal of Computational Physics* 229.7 (2010), pp. 2432–2452.

[14] Yongsam Kim, Ming-Chih Lai, and Charles S Peskin. "Numerical simulations of two-dimensional foam by the immersed boundary method". In: *Journal of Computational Physics* 229.13 (2010), pp. 5194–5207.

[15] Yongsam Kim et al. "Numerical simulations of three-dimensional foam by the immersed boundary method". In: *Journal of Computational Physics* 269 (2014), pp. 1–21.

[16] Cyril W Hirt and Billy D Nichols. "Volume of fluid (VOF) method for the dynamics of free boundaries". In: *Journal of Computational Physics* 39.1 (1981), pp. 201–225.

[17] Stanley Osher and James A Sethian. "Fronts propagating with curvature–dependent speed: algorithms based on Hamilton–Jacobi formulations". In: *Journal of Computational Physics* 79.1 (1988), pp. 12–49.

[18] Ronald P Fedkiw et al. "A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)". In: *Journal of Computational Physics* 152.2 (1999), pp. 457–492.

[19] Bernd Einfeldt et al. "On Godunov-type methods near low densities". In: *Journal of Computational Physics* 92.2 (1991), pp. 273–295.

[20] XY Hu et al. "A conservative interface method for compressible flows". In: *Journal of Computational Physics* 219.2 (2006), pp. 553–578.

[21] Shucheng Pan, Xiangyu Hu, and Nikolaus A Adams. "High-resolution method for evolving complex interface networks". In: *Computer Physics Communications* (2018).

[22] Shucheng Pan et al. "High-order time-marching reinitialization for regional level-set functions". In: *Journal of Computational Physics* 354 (2018), pp. 311–319.

[23] Shucheng Pan et al. "A conservative interface-interaction method for compressible multi-material flows". In: *Journal of Computational Physics* (2018).

[24] James Albert Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Vol. 3. Cambridge university press, 1999.

[25] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Applied Mathematical Sciences, 2003.

[26] Frederic Gibou, Ronald Fedkiw, and Stanley Osher. "A review of level-set methods and some recent applications". In: *Journal of Computational Physics* (2017).

[27] David Adalsteinsson and James A Sethian. "The fast construction of extension velocities in level set methods". In: *Journal of Computational Physics* 148.1 (1999), pp. 2–22.

[28] John Strain. "Semi-Lagrangian methods for level set equations". In: *Journal of Computational Physics* 151.2 (1999), pp. 498–533.

[29] Leo Miguel González Gutiérrez and Rodolfo Bermejo. "A semi-Lagrangian level set method for incompressible Navier–Stokes equations with free surface". In: *International Journal for Numerical Methods in Fluids* 49.10 (2005), pp. 1111–1146.

[30] Chohong Min and Frédéric Gibou. "A second order accurate level set method on non-graded adaptive cartesian grids". In: *Journal of Computational Physics* 225.1 (2007), pp. 300–321.

[31] Michael Lentine, Jón Tómas Grétarsson, and Ronald Fedkiw. "An unconditionally stable fully conservative semi–Lagrangian method". In: *Journal of Computational Physics* 230.8 (2011), pp. 2857–2879.

[32] Ami Harten and Stanley Osher. "Uniformly high-order accurate nonoscillatory schemes. I". In: *SIAM Journal on Numerical Analysis* 24.2 (1987), pp. 279–309.

[33] Ami Harten et al. "Uniformly high order accurate essentially non–oscillatory schemes, III". In: *Journal of Computational Physics* 71.2 (1987), pp. 231–303.

[34] Xu-Dong Liu, Stanley Osher, and Tony Chan. "Weighted essentially non–oscillatory schemes". In: *Journal of Computational Physics* 115.1 (1994), pp. 200–212.

[35] Guang-Shan Jiang and Chi-Wang Shu. "Efficient implementation of weighted ENO schemes". In: *Journal of Computational Physics* 126.1 (1996), pp. 202–228.

[36] Chi-Wang Shu. "Essentially non-oscillatory and weighted essentially non–oscillatory schemes for hyperbolic conservation laws". In: *Advanced numerical approximation of nonlinear hyperbolic equations*. Springer, 1998, pp. 325–432.

[37] Stanley Osher and Chi-Wang Shu. "High-order essentially non–oscillatory schemes for Hamilton–Jacobi equations". In: *SIAM Journal on Numerical Analysis* 28.4 (1991), pp. 907–922.

[38] Guang-Shan Jiang and Danping Peng. "Weighted ENO schemes for Hamilton–Jacobi equations". In: *SIAM Journal on Scientific computing* 21.6 (2000), pp. 2126–2143.

[39] Chi-Wang Shu and Stanley Osher. "Efficient implementation of essentially non–oscillatory shock-capturing schemes". In: *Journal of Computational Physics* 77.2 (1988), pp. 439–471.

[40] Douglas Enright et al. "A hybrid particle level set method for improved interface capturing". In: *Journal of Computational Physics* 183.1 (2002), pp. 83–116.

[41] Simone E Hieber and Petros Koumoutsakos. "A Lagrangian particle level set method". In: *Journal of Computational Physics* 210.1 (2005), pp. 342–367.

[42] Elin Olsson and Gunilla Kreiss. "A conservative level set method for two phase flow". In: *Journal of Computational Physics* 210.1 (2005), pp. 225–246.

[43] Olivier Desjardins, Vincent Moureau, and Heinz Pitsch. "An accurate conservative level set/ghost fluid method for simulating turbulent atomization". In: *Journal of Computational Physics* 227.18 (2008), pp. 8395–8416.

[44] Mark Sussman and Elbridge Gerry Puckett. "A coupled level set and volume–of–fluid method for computing 3D and axisymmetric incompressible two-phase flows". In: *Journal of Computational Physics* 162.2 (2000), pp. 301–337.

[45] Mark Sussman. "A second order coupled level set and volume–of–fluid method for computing growth and collapse of vapor bubbles". In: *Journal of Computational Physics* 187.1 (2003), pp. 110–136.

[46] David Adalsteinsson and James A Sethian. "A fast level set method for propagating interfaces". In: *Journal of Computational Physics* 118.2 (1995), pp. 269–277.

[47] Mark Sussman et al. "An improved level set method for incompressible two-phase flows". In: *Computers & Fluids* 27.5-6 (1998), pp. 663–680.

[48] Alexandre Joel Chorin. "Numerical solution of the Navier-Stokes equations". In: *Mathematics of Computation* 22.104 (1968), pp. 745–762.

[49] Ann S Almgren et al. "A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations". In: *Journal of Computational Physics* 142.1 (1998), pp. 1–46.

[50] W Mulder, S Osher, and James A Sethian. "Computing interface motion in compressible gas dynamics". In: *Journal of Computational Physics* 100.2 (1992), pp. 209–228.

[51] Ronald P Fedkiw, Antonio Marquina, and Barry Merriman. "An isobaric fix for the overheating problem in multimaterial compressible flows". In: *Journal of Computational Physics* 148.2 (1999), pp. 545–578.

[52] Xiang Yu Hu and Boo Cheong Khoo. "An interface interaction method for compressible multifluids". In: *Journal of Computational Physics* 198.1 (2004), pp. 35–64.

[53] Eleuterio F Toro, Michael Spruce, and William Speares. "Restoration of the contact surface in the HLL-Riemann solver". In: *Shock Waves* 4.1 (1994), pp. 25–34.

[54] XY Hu, NA Adams, and Gianluca Iaccarino. "On the HLLC Riemann solver for interface interaction in compressible multi-fluid flow". In: *Journal of Computational Physics* 228.17 (2009), pp. 6572–6589.

[55] Xiang Zhao, Sami Bayyuk, and Sijun Zhang. "Aeroelastic response of rocket nozzles to asymmetric thrust loading". In: *Computers & Fluids* 76 (2013), pp. 128–148.

[56] MP Kirkpatrick, SW Armfield, and JH Kent. "A representation of curved boundaries for the solution of the Navier–Stokes equations on a staggered three-dimensional Cartesian grid". In: *Journal of Computational Physics* 184.1 (2003), pp. 1–36.

[57] Phillip Colella et al. "A Cartesian grid embedded boundary method for hyperbolic conservation laws". In: *Journal of Computational Physics* 211.1 (2006), pp. 347–366.

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

[59] Ami Harten. "Multiresolution algorithms for the numerical solution of hyperbolic conservation laws". In: *Communications on Pure and Applied Mathematics* 48.12 (1995), pp. 1305–1342.

[60] Ralf Deiterding et al. "Adaptive multiresolution or adaptive mesh refinement? A case study for 2D Euler equations". In: *ESAIM: Proceedings*. Vol. 29. EDP Sciences. 2009, pp. 28–42.

[61] Marsha J Berger and Joseph Oliger. "Adaptive mesh refinement for hyperbolic partial differential equations". In: *Journal of Computational Physics* 53.3 (1984), pp. 484–512.

[62] Marsha J Berger and Phillip Colella. "Local adaptive mesh refinement for shock hydrodynamics". In: *Journal of Computational Physics* 82.1 (1989), pp. 64–84.

[63] Siegfried Müller. *Adaptive multiscale schemes for conservation laws*. Vol. 27. Springer Science & Business Media, 2012.

[64] Olivier Roussel et al. "A conservative fully adaptive multiresolution algorithm for parabolic PDEs". In: *Journal of Computational Physics* 188.2 (2003), pp. 493–523.

[65] Margarete O Domingues et al. "An adaptive multiresolution scheme with local time stepping for evolutionary PDEs". In: *Journal of Computational Physics* 227.8 (2008), pp. 3758–3780.

[66] Marcus Herrmann. "A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids". In: *Journal of Computational Physics* 227.4 (2008), pp. 2674–2706.

[67] XY Hu et al. "Multi-scale modeling of compressible multi-fluid flows with conservative interface method". In: *Proceedings of the Summer Program*. Vol. 301. Center for Turbulence Research. 2010.

[68] LH Han, XY Hu, and Nikolaus A Adams. "Scale separation for multi-scale modeling of free-surface and two-phase flows with the conservative sharp interface method". In: *Journal of Computational Physics* 280 (2015), pp. 387–403.

[69] Jun Luo, XY Hu, and NA Adams. "Efficient formulation of scale separation for multi-scale modeling of interfacial flows". In: *Journal of Computational Physics* 308 (2016), pp. 411–420.

[70] David J Benson. "Volume of fluid interface reconstruction methods for multi-material problems". In: *Applied Mechanics Reviews* 55.2 (2002), pp. 151–165.

[71] Vadim Dyadechko and Mikhail Shashkov. "Reconstruction of multi-material interfaces from moment data". In: *Journal of Computational Physics* 227.11 (2008), pp. 5361–5384.

[72] Barry Merriman, James K Bence, and Stanley J Osher. "Motion of multiple junctions: A level set approach". In: *Journal of Computational Physics* 112.2 (1994), pp. 334–363.

[73] Tony F Chan, Mark Moelich, and Berta Sandberg. "Some recent developments in variational image segmentation". In: *Image processing based on partial differential equations*. Springer, 2007, pp. 175–210.

[74] Hong-Kai Zhao et al. "A variational level set approach to multiphase motion". In: *Journal of Computational Physics* 127.1 (1996), pp. 179–195.

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

[76] Luminita A Vese and Tony F Chan. "A multiphase level set framework for image segmentation using the Mumford and Shah model". In: *International Journal of Computer Vision* 50.3 (2002), pp. 271–293.

[77] Johan Lie, Marius Lysaker, and Xue-Cheng Tai. "A variant of the level set method and applications to image segmentation". In: *Mathematics of Computation* 75.255 (2006), pp. 1155–1174.

[78] Ginmo Chung and Luminita A Vese. "Image segmentation using a multilayer level-set approach". In: *Computing and Visualization in Science* 12.6 (2009), pp. 267–285.

[79] Wen Zheng, Jun-Hai Yong, and Jean-Claude Paul. "Simulation of bubbles". In: *Graphical Models* 71.6 (2009), pp. 229–239.

[80] Robert I Saye and James A Sethian. "The Voronoi implicit interface method for computing multiphase physics". In: *Proceedings of the National Academy of Sciences* 108.49 (2011), pp. 19498–19503.

[81] RI Saye and James A Sethian. "Analysis and applications of the Voronoi implicit interface method". In: *Journal of Computational Physics* 231.18 (2012), pp. 6051–6085.

[82] Guy Barles, H Mete Soner, and Panagiotis E Souganidis. "Front propagation and phase field theory". In: *SIAM Journal on Control and Optimization* 31.2 (1993), pp. 439–469.

[83] José Gomes and Olivier Faugeras. "Reconciling distance functions and level sets". In: *Journal of Visual Communication and Image Representation* 11.2 (2000), pp. 209–223.

[84] Virginia Estellers et al. "Efficient algorithm for level set method preserving distance function". In: *IEEE Transactions on Image Processing* 21.12 (2012), pp. 4722–4734.

[85] James A Sethian. "A fast marching level set method for monotonically advancing fronts". In: *Proceedings of the National Academy of Sciences* 93.4 (1996), pp. 1591–1595.

[86] James A Sethian. "Fast marching methods". In: *SIAM Review* 41.2 (1999), pp. 199–235.

[87] James A Sethian. "Evolution, implementation, and application of level set and fast marching methods for advancing fronts". In: *Journal of Computational Physics* 169.2 (2001), pp. 503–555.

[88] Yen-Hsi Richard Tsai et al. "Fast sweeping algorithms for a class of Hamilton–Jacobi equations". In: *SIAM journal on Numerical Analysis* 41.2 (2003), pp. 673–694.

[89] Hongkai Zhao. "A fast sweeping method for eikonal equations". In: *Mathematics of Computation* 74.250 (2005), pp. 603–627.

[90] Chiu-Yen Kao, Stanley Osher, and Yen-Hsi Tsai. "Fast Sweeping Methods for Static Hamilton–Jacobi Equations". In: *SIAM Journal on Numerical Analysis* 42.6 (2005), pp. 2612–2632.

[91] Michael Wayne Royston. "A Hopf-Lax formulation of the eikonal equation for parallel redistancing and oblique projection". PhD thesis. UCLA, 2017.

[92] Byungjoon Lee et al. "Revisiting the redistancing problem using the Hopf–Lax formula". In: *Journal of Computational Physics* 330 (2017), pp. 268–281.

[93] Michael Royston et al. "Parallel redistancing using the Hopf–Lax formula". In: *Journal of Computational Physics* (2018).

[94] Mark Sussman, Peter Smereka, and Stanley Osher. "A level set approach for computing solutions to incompressible two-phase flow". In: *Journal of Computational Physics* 114.1 (1994), pp. 146–159.

[95] Li-Tien Cheng and Yen-Hsi Tsai. "Redistancing by flow of time dependent eikonal equation". In: *Journal of Computational Physics* 227.8 (2008), pp. 4002–4017.

[96] Giovanni Russo and Peter Smereka. "A remark on computing distance functions". In: *Journal of Computational Physics* 163.1 (2000), pp. 51–67.

[97] Mark Sussman and Emad Fatemi. "An efficient, interface–preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow". In: *SIAM Journal on scientific computing* 20.4 (1999), pp. 1165–1191.

[98] 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". In: *Journal of Scientific Computing* 35.2-3 (2008), pp. 114–131.

[99] Daniel Hartmann, Matthias Meinke, and Wolfgang Schröder. "The constrained reinitialization equation for level set methods". In: *Journal of Computational Physics* 229.5 (2010), pp. 1514–1535.

[100] Chohong Min. "On reinitializing level set functions". In: *Journal of Computational Physics* 229.8 (2010), pp. 2764–2772.

[101] David L Chopp. "Some improvements of the fast marching method". In: *SIAM Journal on Scientific Computing* 23.1 (2001), pp. 230–244.

[102] Matt Elsey and Selim Esedoglu. "Fast and accurate redistancing by directional optimization". In: *SIAM Journal on Scientific Computing* 36.1 (2014), A219–A231.

[103] Vincent A Thomas and Robert J Kares. "Drive asymmetry and the origin of turbulence in an ICF implosion". In: *Physical Review Letters* 109.7 (2012), p. 075004.

[104] Eric J Lentz et al. "Three-dimensional core-collapse supernova simulated using a 15 M$\odot$ progenitor". In: *The Astrophysical Journal Letters* 807.2 (2015), p. L31.

[105] Galen Gisler et al. "Two-and three-dimensional simulations of asteroid ocean impacts la-ur 02-66-30". In: *Science of Tsunami Hazards* 21.2 (2003), p. 119.

[106] Marcel Meyer et al. "A conservative immersed interface method for large-eddy simulation of incompressible flows". In: *Journal of Computational Physics* 229.18 (2010), pp. 6300–6317.

[107] E Lauer et al. "Numerical modelling and investigation of symmetric and asymmetric cavitation bubble dynamics". In: *Computers & Fluids* 69 (2012), pp. 1–19.

[108] E Lauer et al. "Numerical investigation of collapsing cavity arrays". In: *Physics of Fluids* 24.5 (2012), p. 052104.

[109] Jian Luo, XY Hu, and NA Adams. "A conservative sharp interface method for incompressible multiphase flows". In: *Journal of Computational Physics* 284 (2015), pp. 547–565.

[110] Philip T Barton, B Obadia, and Dimitris Drikakis. "A conservative level-set based method for compressible solid/fluid problems on fixed grids". In: *Journal of Computational Physics* 230.21 (2011), pp. 7867–7890.

[111] Vito Pasquariello et al. "A cut-cell finite volume–finite element coupling approach for fluid–structure interaction in compressible flow". In: *Journal of Computational Physics* 307 (2016), pp. 670–695.

[112] Michael Yu Wang and Xiaoming Wang. ""Color" level sets: a multi-phase method for structural topology optimization with multiple materials". In: *Computer Methods in Applied Mechanics and Engineering* 193.6-8 (2004), pp. 469–496.

[113] Miaojun Yao et al. "Level-set-based partitioning and packing optimization of a printable model". In: *ACM Transactions on Graphics (TOG)* 34.6 (2015), p. 214.

[114] Shucheng Pan, Xiangyu Hu, Nikolaus Adams, et al. "A variational-level-set based partitioning method for block-structured meshes". In: *arXiv preprint arXiv:1801.03685* (2018).

[115] Martin Robinson, Mark Flegg, and Radek Erban. "Adaptive two-regime method: application to front propagation". In: *The Journal of chemical physics* 140.12 (2014), p. 124109.

[116] Jonathan U Harrison and Christian A Yates. "A hybrid algorithm for coupling partial differential equation and compartment-based dynamics". In: *Journal of The Royal Society Interface* 13.122 (2016), p. 20160335.

Appendix A

Original journal papers

Here, the peer-reviewed journal publications of the present work are attached.

A.1 Paper I

Shucheng Pan, Xiangyu Y. Hu, Nikolaus A. Adams

High-resolution method for evolving complex interface networks

In *Computer Physics Communications*, Volume 225, 2018, pp. 10-27, DOI: `https://doi.org/10.1016/j.cpc.2018.01.001`.

Contribution: My contribution to this work was the development of the method and the corresponding computer code for its implementation. I performed simulations and analyzed the results, and wrote the manuscript for the publication.

Computer Physics Communications 225 (2018) 10–27

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

High-resolution method for evolving complex interface networks

Shucheng Pan, Xiangyu Y. Hu *, Nikolaus A. Adams

Chair of Aerodynamics and Fluid mechanics, Department of Mechanical Engineering, Technical University of Munich, 85748 Garching, Germany

ARTICLE INFO

Article history:
Received 8 February 2017
Received in revised form 26 October 2017
Accepted 4 January 2018
Available online 11 January 2018

Keywords:
Multi-region problem
Interface network
Level-set method
Interface capturing
High-order scheme
Multiphase flow

ABSTRACT

In this paper we describe a high-resolution transport formulation of the regional level-set approach for an improved prediction of the evolution of complex interface networks. The novelty of this method is twofold: (i) construction of local level sets and reconstruction of a global level set, (ii) local transport of the interface network by employing high-order spatial discretization schemes for improved representation of complex topologies. Various numerical test cases of multi-region flow problems, including triple-point advection, single vortex flow, mean curvature flow, normal driven flow, dry foam dynamics and shock–bubble interaction show that the method is accurate and suitable for a wide range of complex interface-network evolutions. Its overall computational cost is comparable to the Semi-Lagrangian regional level-set method while the prediction accuracy is significantly improved. The approach thus offers a viable alternative to previous interface-network level-set method.

1. Introduction

Multi-region problems can occur when the motion of more than two immiscible fluids is to be described. In this case the interface network, separating the different fluid regions, evolves in time due to interactions of the different fluids across interface segments. These interactions often can be described by local fluid properties. Important applications include shock-driven multiphase flows [1–3], astrophysical events [4–6], foam dynamics [7–10], multi-cellular tissue dynamics [11–15], and grain coarsening in polycrystalline materials [16–19].

A range of numerical models have been proposed to compute the evolution of interface networks for multi-region problems. Generally, they can be classified as Lagrangian or Eulerian methods according to the representation of the interface. With Lagrangian methods, such as front-tracking [20], immersed-boundary [21], or arbitrary Lagrangian–Eulerian (ALE) [22] methods, the interface is represented explicitly by conforming discretization elements. Although these methods have been extended to multi-region systems [23–27,10], it is difficult to handle complex topological changes during interface-network evolution, especially in three dimensions. With Eulerian methods, such as volume-of-fluid (VOF) [28] and level-set methods [29], the interface is reconstructed from scalar fields, i.e., volume fraction or level-set field. The interface is represented implicitly and captured by solving the corresponding transport equations. These methods generally can handle complex interface evolution with topological changes and are straightforward to implement in three dimensions. However, they often exhibit low accuracy due to numerical dissipation introduced by the transport-equation discretization.

Two additional problems are encountered when Eulerian methods are applied to multi-region problems [30–32]. One is that the number of scalar fields increases with the number of regions and entails additional computational operations and memory cost [33]. The other is that the interface reconstruction can produce voids and overlaps where more than two regions meet [31–33]. There are two main level-set-based approaches for capturing the evolution of multiple regions. One is to define multiple level-set functions (referred to as multiple level-set method in this paper) and to assign these to corresponding regions [30–32], followed by solving separate level-set transport equations for each region. Different numerical procedures may be employed to prevent the generation of voids and overlaps during the interface reconstruction. For example, Starinshak et al. [32] propose $N(N-1)/2$ level-set functions to represent all interfaces of N regions and an additional pairwise voting strategy to remove overlaps and voids. The algorithm copes with the interface reconstruction problem but requires that a larger number of different level-set fields are stored and evolved in time than with multiple level-set methods [30,31]. The method of Vese and Chan [34] reduces the number of level-set fields from N to $\log_2 N$ [33] and it avoids the generation of voids and overlaps. Naturally, using a single level-set function to represent an arbitrary number of regions is the optimal strategy to address the memory overhead [33]. Lie et al. [35] and Chung and Vese [36] develop such methods for image segmentation. Also, the regional level-set method [37] addresses the problem of multiple level-set fields [38] by employing a combination of a single unsigned

* Corresponding author.
E-mail addresses: xiangyu.hu@aer.mw.tum.de, xiangyu.hu@tum.de (X.Y. Hu).

https://doi.org/10.1016/j.cpc.2018.01.001
0010-4655/

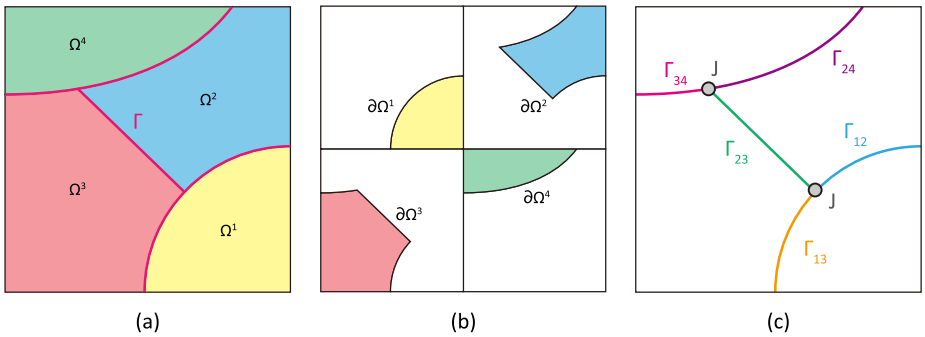


Fig. 1. A 4-region system on a square domain Ω. The index set for all regions is $X = \{1, 2, 3, 4\}$. The region domain and the interface network Γ are shown in (a). The region boundary $\partial\Omega^\chi$ for each region is shown in (b). The pairwise interfaces $\Gamma_{\alpha\beta}$ ($\alpha, \beta \in$ X and $\alpha \neq \beta$) are represented by different colors in (c) where two multiple junctions J are observed.

level-set field and an integer region indicator function which is termed as the global level-set function in this paper. Interface reconstruction for more than two regions, is handled by employing a low-order Semi-Lagrangian scheme. We note that this low-order Semi-Lagrangian scheme is more dissipative than high-order finite-difference schemes. As further development of the regional level-set method, the Voronoi implicit interface method [39] uses a transported ϵ-level-set and reconstructs the interface network by a reinitialization step based on Voronoi diagrams.

Our objective is to develop a regional level-set method that has computational cost comparable to Semi-Lagrangian regional level-set methods [37,38], but significantly improves prediction accuracy. Specifically, the method inherits the advantages of the original level-set method and the Semi-Lagrangian regional level-set method but is modified in such a way that it is suitable for high-resolution finite-difference discretization of the level-set transport equations. We define local level-set fields to capture the evolution of the interface network by a simple construction operator, followed by a reconstruction operator to obtain the global level-set field. The multi-region system and the definition of the global level-set field are revisited in Section 2. The proposed multi-region method is detailed in Section 3. Accuracy and robustness are assessed in Section 4, followed by a brief conclusion in Section 5.

2. Multi-region system and regional level set method

First we introduce the representation of a multi-region system by implicit functions. Let the domain Ω be an open set in $\mathbb{R}^d$, let $\mathbf{x} \in \Omega$ be an interior point of Ω, and let $\partial\Omega$ be the boundary of the domain, where d is the spatial dimension. Assuming that there are $\mathcal{N}$ regions within this domain, Ω is the union of a family of disjoint subsets, $\Omega = \bigcup_{\chi \in \mathrm{X}} \Omega^\chi$, where Ω^χ is the subdomain of the region χ, and $\mathrm{X} = \{\chi \in \mathbb{N} | 1 \leq \chi \leq \mathcal{N}\}$ is the index set for all regions. The entire multi-region system consists of the following elements:

- The region domain Ω^χ which contains all interior points of the region χ.
- The region boundary $\partial\Omega^\chi$ which contains the set of boundary points of Ω^χ.
- The pairwise interfaces $\Gamma_{\alpha\beta} = \partial\Omega^\alpha \cap \partial\Omega^\beta$ ($\alpha, \beta \in$ X and $\alpha \neq \beta$), each as codimension manifold in $\mathbb{R}^d$ that separates two connected regions.
- The interface network $\Gamma = \bigcup_{\alpha,\beta \in \mathrm{X}, \alpha \neq \beta} \Gamma_{\alpha\beta}$ is the union of all pairwise interfaces.
- Multiple junctions (high order junctions) $\mathrm{J} = \bigcap_{\alpha,\beta \in \mathrm{X}, \alpha \neq \beta} \Gamma_{\alpha\beta}$ are intersections of pairwise interfaces. In most practically relevant applications they are triple points (2D) or triple lines (3D).

An example with 4 regions on a square domain Ω in Fig. 1 illustrates these definitions. In Fig. 1(a), the 4 region domains, Ω^χ, with $\chi \in [1, 4]$, are separated by the interface network Γ. The region boundaries, $\partial\Omega^1$, $\partial\Omega^2$, $\partial\Omega^3$ and $\partial\Omega^4$, Fig. 1(b). The system contains 5 pairwise interfaces (Γ_{12}, Γ_{13}, Γ_{23}, Γ_{24} and Γ_{34}) and 2 multiple junctions (J), Fig. 1(c).

If there are only two regions ($\mathcal{N} = 2$) in the system, the above elements can be implicitly defined by a level-set field (or signed distance function) [29] $\phi : \mathbb{R}^d \to \mathbb{R}$. The two region domains, Ω^1 and Ω^2, are identified by the sign of the level-set function: $\Omega^1 = \{\mathbf{x} \in \mathbb{R}^d | \phi(\mathbf{x}) > 0\}$ and $\Omega^2 = \{\mathbf{x} \in \mathbb{R}^d | \phi(\mathbf{x}) < 0\}$. The region boundaries of Ω^1 and Ω^2 coincide and are identical to the pairwise interface and interface network, $\partial\Omega^1 = \partial\Omega^2 = \Gamma_{1,2} = \Gamma = \{\mathbf{x} \in \mathbb{R}^d | \phi(\mathbf{x}) = 0\}$. There are no multiple junctions in this system. When more than two regions ($\mathcal{N} > 2$) exist, as shown in Fig. 2(a), the global level-set function used in the regional level-set method [37,39] can be used to represent the system implicitly. For the global level-set function, a mapping $\varphi^\chi : \Omega \subset \mathbb{R}^d \to \mathbb{R} \times \mathbb{N}$ is defined as $\varphi^\chi(\mathbf{x}) = (\varphi(\mathbf{x}), \chi(\mathbf{x}))$, where $\varphi(\mathbf{x}) \geq 0$ is the unsigned distance function and $\chi(\mathbf{x})$ is a positive integer region indicator. A subdomain Ω^α is identified by region indicators, $\Omega^\alpha = \{\mathbf{x} \in \Omega | \chi(\mathbf{x}) = \alpha\}$ and the interface network is defined as the zero distance contour $\Gamma = \{\mathbf{x} \in \mathbb{R}^d | \varphi(\mathbf{x}) = 0\}$. However, the region boundary, the pairwise interface and multiple junctions cannot be identified directly by the global level-set field φ^χ. An additional operation, such as the Voronoi method described in [39] is required for this purpose. In this paper we propose as an alternative a simple construction operator to define these surface contours.

3. Numerical method

Before we proceed to numerical details of our method, we first list the main notations in Table 1 to enhance the readability of our algorithm.

3.1. Global and local index sets of regions

In this section, we develop the numerical method for the representation of the multi-region system and the evolution of the interface network. For this purpose we consider the multi-region system defined by $\varphi^\chi_{i,j} = (\varphi_{i,j}, \chi_{i,j})$ at the centers of finite-volume (FV) cells with indices i in x_1 and j in x_2 coordinate directions on a two-dimensional uniform Cartesian grid, Fig. 2(a). An extension of the two-dimensional definitions to three dimensions is straightforward. We recall that globally there are $\mathcal{N}$ regions in the considered domain. We assume that for a multi-region system, although $\mathcal{N}$ may be very large, locally the number of regions that occur in a neighborhood of a cell is limited to a small integer number. Interface networks typically exhibit only very few FV cells which contain multiple junctions. Thus, multiple signed level-set functions defined only locally to capture the evolution of the interface network allow to reduce the computational effort significantly. The

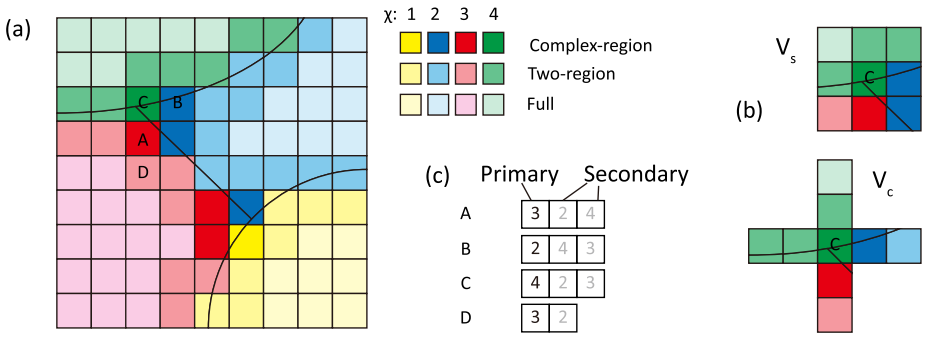


Fig. 2. A schematic representation example of cell neighborhoods and cell types. In (a), a 4-region system is defined on 9 × 9 FV cells. Cells are colored by yellow ($\chi = 1$), blue ($\chi = 2$), red ($\chi = 3$) and green ($\chi = 3$) according to the indicator on cell centers. The brightness of each color indicates cells with different type of V_s. The definition of cell neighborhoods V_s and V_c of the cell C in (a) is illustrated in (b). In (c), the local indicators in V_s of the chosen 4 cells (A, B, C and D) in (a) are categorized as the primary and secondary indicators which are represented by black and gray, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Summary of mathematical notations.

Symbols	Descriptions
$\mathcal{N}$	The number of regions in the domain
$\mathcal{N}_s$	The number of regions in V_s
$\mathcal{N}_C$	The number of regions that share a particular finite-volume cell
X	Global index set, a vector with $\mathcal{N}$ elements, $\{1, 2, \ldots \mathcal{N}\}$
X_s	Local index set for V_s, a vector with $\mathcal{N}_s$ elements
X_C	Local index set for a particular finite-volume cell, a vector with $\mathcal{N}_C$ elements
V_s	Square-shaped near neighborhood, a 3 × 3 cell matrix
V_c	Cross-shaped stencil neighborhood, a vector with $2S + 1$ cells
φ^χ	Global level-setfield, $n \times n$ matrix with elements $\in \mathbb{R} \times \mathbb{N}$
ϕ	Local level-set field: a 3 × 3 matrix on V_s, a vector with $2S + 1$ elements on V_s and a vector with $\min(3^d, \mathcal{N})$ elements for advection

required global and local definitions on a two-dimensional uniform Cartesian grid are as follows:

• $\mathcal{N}$ global regions are labeled by the global index set X. The center $\mathbf{x}_{i,j}$ of each FV cell $C_{i,j}$ is assigned to one region by an indicator $\chi_{i,j}$ even though $C_{i,j}$ may contain more than one region. Here i, j are the global indices for cells.

• A local subdomain is defined as $\omega \subset \Omega$. The $\mathcal{N}_\omega$ regions contained in this local subdomain are labeled by a local index set $X_\omega = \{r \in \mathbb{N} | 1 \le r \le \mathcal{N}_\omega\}$. We map the local region indicators into the global index set X, and identify the corresponding global region indicator as χ_r. A local subdomain Ω_ω^α of the global region Ω^α is defined by $\Omega_\omega^\alpha = \{\mathbf{x} \in \omega | \chi_r(\mathbf{x}) = \alpha, r \in X_\omega\} \subset \Omega^\alpha$. Here, ω can be a FV cell $C_{i,j}$ or a neighborhood, i.e., a set of cells $C_{k,l}$ of $C_{i,j}$, which will be defined in Section 3.2. (k, l) denotes the local index pair for cells in such a neighborhood of $C_{i,j}$.

3.2. Cell neighborhoods and cell types

For a FV cell $C_{i,j}$ one can define two types of neighborhoods, i.e. cell sets: a square-shaped near neighborhood

$$V_s = \{C_{k,l} | i - 1 < k < i + 1, j - 1 < l < j + 1\}, \tag{1}$$

and a cross-shaped stencil neighborhood

$$V_c = \{C_{k,l} | i - S < k < i + S, l = j\} \cup \{C_{k,l} | k = i, j - S < l < j + S\}, \tag{2}$$

where $2S + 1$ is the required width of the stencil for the level-set transport discretization schemes. The number of separate regions $\mathcal{N}_C$ contained in cell $C_{i,j}$ and the number of region indicators in the square-shaped near neighborhood $\mathcal{N}_s$ have the relation $1 \le \mathcal{N}_C \le \mathcal{N}_s \ll \mathcal{N}$ when $\mathcal{N}$ is large. We also have $\mathcal{N}_s \le \min(3^d, \mathcal{N})$ as the size of V_s is 3 × 3 (2D) or 3 × 3 × 3 (3D). And $\mathcal{N}_C \le \min(2^d, \mathcal{N})$ as there are at most 2^d different indicators assigned at 2^d vertices of this cell if the interface is resolved. Accordingly, the local index sets X_ω of Section 3.1 specifically for V_s and V_c are $X_s = \{r \in \mathbb{N} | 1 \le r \le \mathcal{N}_s\}$ and $X_C = \{r \in \mathbb{N} | 1 \le r \le \mathcal{N}_C\}$, respectively. Note that V_c, V_s, $\mathcal{N}_C$ and $\mathcal{N}_s$ are defined for each cell $C_{i,j}$ and thus depend on the index pair (i, j).

According to the value of $\mathcal{N}_s$, a cell $C_{i,j}$ is categorized as a cell with full type V_s ($\mathcal{N}_s = 1$), as a cell with two-region type V_s ($\mathcal{N}_s = 2$), or as a cell with complex-region type V_s ($\mathcal{N}_s > 2$). The brightness of each cell in Fig. 2(a) represents cell with different types of V_s. The definitions of V_s and V_c for cell C ($i = 3, j = 11$) in Fig. 2(a) are illustrated in Fig. 2(b). By counting the number of unique indicators in V_s, we find that this cell is a cell with complex-region type V_s and its local index set is $X_s = \{1, 2, 3\}$. We clarify that the square-shaped near neighborhood V_s serves as a search stencil to identify all local regions that may share the cell $C_{i,j}$ and to generate the local index set X_s. As shown in Fig. 2(c), we identify $\chi_1 = \chi_{i,j}$ as "primary indicator", and other region indicators χ_r, $r \in X_s$, $r > 1$, if they exist, are called "secondary indicators".

After knowing the type of V_s, a cell is denoted as a "full cell" if it is not intersected by an interface, i.e. $\mathcal{N}_C = 1$, as a "two-region cut cell" if it is intersected by an interface segment shared by two regions, i.e. $\mathcal{N}_C = 2$, or as a "complex-region cut cell" if the interface segment is shared by more than two regions, i.e. $\mathcal{N}_C > 2$, see Fig. 3. To identify the cell types, the first step is to construct $\mathcal{N}_s$ (3 for Fig. 3) local level-set functions on cell centers ("∘" in Fig. 3) by a mapping (5), followed by interpolating data on vertices ("×" in Fig. 3). Then the cell is shared by the region χ_r if its vertices values ϕ^r have different signs. The cell is determined by the number of regions that share this cell, as shown in Algorithm 1 and Fig. 3. For example, the cell in Fig. 3(a) is intersected by regions $\chi_1 = 3$ and $\chi_2 = 1$, indicating it is a "two-region cut cell". While the cell in Fig. 3(b) is a "complex-region cut cell" as it is intersected by 3 regions. We note here that the identification of cell types is required for post-processing and coupling with external physics.

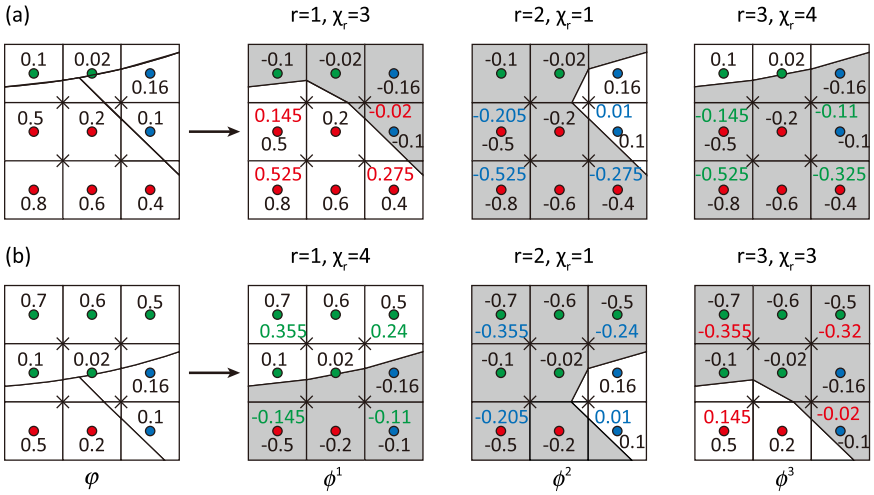


Fig. 3. Cell type determination for (a) cell A and (b) cell C in Fig. 2(a). The data on the left side of the arrow is the unsigned level-set φ and local level-set functions on cell centers ($\circ$) of the right side, ϕ^1, ϕ^2 and ϕ^3, are obtained by mapping (5). The local level-set field on vertices ($\times$) is calculated by bilinear interpolation.

And the region number $\mathcal{N}_C$ indicates the number of region boundaries segments reconstructed in a multi-region cell. On the cross-shaped stencil neighborhood V_c, $\mathcal{N}_s$ auxiliary local level-set fields (see Section 3.3) corresponding to the local regions identified in V_s are constructed in order to evolve the interface network by discretized evolution equations and represent the region boundaries from the zero level set.

Algorithm 1 Determination of the cell type for $C_{i,j}$. Assume the type of V_s for this cell has been determined.

if $\mathcal{N}_s = 1$ **then**
  $\mathcal{N}_C \leftarrow 1$ ▷ as $1 \le \mathcal{N}_C \le \mathcal{N}_s = 1$
else if $\mathcal{N}_s \ge 2$ **then**
  $\mathcal{N}_C \leftarrow 0$
  for $r = 1$ **to** $\mathcal{N}_s$ **do**
    for $k = i-1$ **to** $i+1$, $l = j-1$ **to** $j+1$ **do**
      $\phi^r_{k,l} \leftarrow \mathbf{C}_r(\varphi^\chi_{k,l})$ ▷ Constructing local level-set fields in V_s
    end for
    Let (p, q) be the index of vertices ("$\times$" in Fig. 3) of cell $C_{i,j}$,
    for $p = 0$ **to** 1, $q = 0$ **to** 1 **do**
      $\phi^{r,v}_{p,q} = 0.25\left(\phi^r_{i+p-1,j+q-1} + \phi^r_{i+p,j+q-1} + \phi^r_{i+p-1,j+q} + \phi^r_{i+p,j+q}\right)$ ▷ Bilinear interpolation
    end for
    if the vertices values $\phi^{r,v}_{p,q}$ have different signs **then**
      $\mathcal{N}_C \leftarrow \mathcal{N}_C + 1$ ▷ $C_{i,j}$ is intersected by $\partial\Omega^{\chi_r}$
    end if
  end for
end if
Determine cell type according to $\mathcal{N}_C$ by the definition in Sec. 3.2.

3.3. Description of the local construction and reconstruction operators

For any cell $C_{k,l}$ in the neighborhood of the cell $C_{i,j}$, we can apply a construction operator for generating the local multiple signed level-set fields and a reconstruction operator for reconstructing the global level-set field from the local level-set fields, see Fig. 4. Note that in our method there are one global level-set function and multiple local level-set functions. The global one, defined on the whole domain, is a $i_{\max} \times j_{\max}$ matrix if there are $i_{\max}$ cells in the x direction and $j_{\max}$ cells in the y direction. The local one is temporarily computed on a small size cell neighborhood (V_s or V_c).

3.3.1. The local construction operator

The construction operator is used to generate the $\mathcal{N}_s$ local level-set fields $\phi^r_{k,l} := \phi_{k,l}(\chi_r)$ from the global level-set field $\varphi^\chi_{k,l}$ at the center of cell $C_{k,l}$ for each region indicator $\chi_r \in \mathrm{X}_s$. The construction operator $\mathbf{C}_r : \mathbb{R} \times \mathbb{N} \to \mathbb{R}$ is defined as

$$\phi^r_{k,l} = \mathbf{C}_r\left(\varphi^\chi_{k,l}\right) = \begin{cases} \varphi_{k,l} & \text{if } \chi_{k,l} = \chi_r \\ -\varphi_{k,l} & \text{otherwise,} \end{cases} \tag{3}$$

and the local level-set function $\phi^r(\mathbf{x}) := \phi(\mathbf{x}, \chi_r)$ at any point can be obtained by interpolation.

Upon construction of the local multiple signed level-set fields, the normal direction and curvature are obtained at the FV cells by

$$\mathbf{n} = \frac{\nabla\phi^r}{|\nabla\phi^r|} \quad \text{and} \quad \kappa = \nabla \cdot \frac{\nabla\phi^r}{|\nabla\phi^r|}, \tag{4}$$

respectively. Unlike with the original level-set method, the normal direction always points away from the interface network. The local construction operator $\mathbf{C}_r$ may be inaccurate for curvature calculation when the signed level set is not strictly a distance function, which may happen near multiple junctions, as demonstrated in 4.1.4. This issue can be handled by a construction operator $\mathbf{C}^\star_r$ on V_s which additionally invokes a re-initialization procedure for recovering the signed-distance property of the local multiple signed level-set fields

$$\phi^r_{k,l} = \mathbf{C}^\star_r\left(\varphi^\chi_{k,l}\right) = \begin{cases} \mathbf{C}_r\left(\varphi^\chi_{k,l}\right) & \text{if } \chi_{k,l} = \chi_r \\ \mathbf{C}_r\left(\min\left(\Delta x, d_s(k,l)\right)\right) & \text{otherwise,} \end{cases} \tag{5}$$

where $d_s(k, l)$ is the distance of the cell center $\mathbf{x}_{k,l}$ from the local interface segment in V_s.

Note that discrete derivatives in the local level-set advection equation Eq. (8), see below, which operate on constructed local level-set functions in the stencil neighborhood V_c, now can be calculated by high-resolution spatial discretization schemes.

3.3.2. The local reconstruction operator

The global level-set field at an arbitrary point can be reconstructed from the multiple auxiliary local level sets for all region indicators in V_s by the reconstruction operator $\mathbf{R} : \mathbb{R}^{\mathcal{N}_s} \to \mathbb{R} \times \mathbb{N}$ given by

$$\begin{aligned} \varphi^\chi(x, y) &= \mathbf{R}\left(\phi^r(x, y), r \in \mathrm{X}_s\right) \\ &= \left(\left|\max \phi^r(\mathbf{x})\right|, \arg\max_{\chi_r} \phi^r(\mathbf{x})\right). \end{aligned} \tag{6}$$

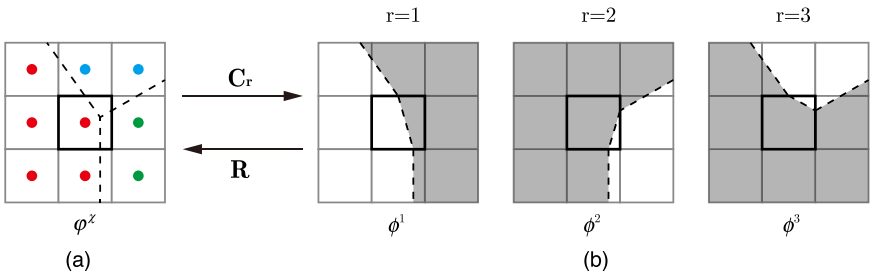


Fig. 4. A schematic representation example of local construction and reconstruction operators. Globally the system has 5 regions and the local index set in this 3×3 stencil is $X_s = \{r | r = 1, 2, 3\}$. The corresponding region index χ_r is $\{\chi_r | r \in X_s\} = \{4, 2, 5\}$. The mapping from (a) to each field in (b) is defined as the construction operator $\mathbf{C} : \mathbb{R} \times \mathbb{N} \to \mathbb{R}$. The inverse mapping from all fields in (b) to (a) is defined as the reconstruction operator $\mathbf{R} : \mathbb{R}^3 \to \mathbb{R} \times \mathbb{N}$. (a) The global level-set field is φ^{χ}. The center of cells whose region indicator equals 4, 2, and 5 is colored by red, green, and blue, respectively. (b) The multiple local level-set functions are ϕ^1, ϕ^2 and ϕ^3 after applying $\mathbf{C}$ on φ^{χ}. The gray part and white part correspond to the negative and positive ϕ^r, respectively. The dashed line in each local level-set field is the region boundary and is represented by the zero contour of ϕ^r. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Note that if the interface network is static, application of the construction operation followed by the reconstruction operation leaves the original global level-set field invariant. Consider the example in Fig. 4. If the global level set at the central cell (colored by red) is $\varphi^{\chi} = (0.2, 4)$, upon local construction, the 3 generated local level-set data $\{\phi^r | r = 1, 2, 3\}$ at this point are $\{0.2, -0.2, -0.2\}$. Upon applying the reconstruction, we obtain the original global level-set data $\varphi^{\chi} = (0.2, 4)$.

This reconstructed field is unique. The reconstruction generates topologies without artificial overlaps or voids. Thus it is suitable for determining the global level set when the interface network has been evolved in time through updating multiple local level-set fields. After solving the advection equation we obtain multiple updated level-set values, ϕ^r, $1 \le r \le \mathcal{N}_s$, defined at the cell center, with each being the distance to the corresponding region boundary $\partial\Omega^{\chi_r}$, see Section 3.4. Naturally, if $\phi^r > 0$, the larger value of ϕ^r indicates that the cell center is farther away from the boundary of the region χ^r, and thus the cell center is located further inside the region Ω^{χ_r}. Similarly, if $\phi^r < 0$, smaller ϕ^r (larger magnitude) indicates the current cell center is further away from the region Ω^{χ_r}. Eq. (6) assigns the indicator with maximum positive values to the cell center.

3.4. Evolution of the interface network

The evolution of the interface network in a multi-region system is equivalent to that of a signed level-set field whose interface advection is determined by the advection equation

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0, \tag{7}$$

where $\mathbf{v}$ is the advection velocity. The advection equation in our method is formulated locally for each region, identified by the region indicator, and recovers the original level-set method for cells that are sufficiently far away from a multiple junction. More detailed considerations are necessary to predict the evolution of the interface for cells in which more than two regions meet. Consider the situation in Fig. 5. The cell contains three regions colored by red, blue, and yellow. Initially, the largest fraction of the cell is occupied by region Ω^{χ_1}, with χ_1 being the primary indicator. After advection by one time step, the interface has three possible configurations, as illustrated in Fig. 5(a). The first is that the cell center still resides within Ω^{χ_1}, so that the primary indicator of the cell does not change. In the other two cases, the primary indicator changes to the secondary indicator, either χ_2 or χ_3. After solving an advection equation for the local level-set field of the three regions separately, each region boundary may shrink (dashed line) or expand (dash-dotted line), see Fig. 5(b). For example, when the boundary $\partial\Omega^{\chi_1}$ of Ω^{χ_1} moves away from the cell center its local level-set ϕ^{χ_1} value increases (positive). For the converse case, $\partial\Omega^{\chi_1}$ moves across the cell center, so that it does not belong to Ω^{χ_1} any longer, resulting in a corresponding sign change to $\phi^{\chi_1} < 0$. That the cell center is not in Ω^{χ_1} does not necessarily imply that it is in Ω^{χ_2} or Ω^{χ_3}. A direct combination of all three independently advected region boundaries may introduce an overlap or a void. To address this issue, we apply the reconstruction operator $\mathbf{R}$ to determine the global level set φ^{χ} from the three candidate local level-set fields $\{\phi^{\chi_1}, \phi^{\chi_2}, \phi^{\chi_3}\}$. This is physically reasonable because $\mathbf{R}$ identifies the most likely indicator corresponding to the region domain in which the current cell center will be located after one advection time step.

The update of the global level set of each FV cell $\varphi_{i,j}^{\chi,n} = (\varphi_{i,j}, \chi_1)^n$ at time-step n, by a sub-step of an explicit time-integration scheme, such as a strongly stable Runge–Kutta scheme [40], consists of three sub-steps: (1) construction of the local multiple signed level-set fields in the stencil neighborhood V_c; (2) computation of the new intermediate multiple local level-set fields by updating the locally constructed advection equations, see below; (3) reconstruction of the global level set from the multiple local level-set fields. The local construction of the advection equations depends on the type of V_s (full, two-region or complex-region). Note that we assume that under the Courant–Friedrichs–Lewy (CFL) condition, i.e. CFL $\le$ 1.0, the type of V_s does not change and the region indicators within the square-shaped near neighborhood remain unchanged during one sub-step of the time-integration scheme, whereas the actual FV cell may change its type. Only $\mathcal{N}_s$ local level-sets are constructed and evolved as the cell center can only be occupied by the region in V_s if CFL $\le$ 1.0. Unlike the "repairing" (or "modification") procedure in Ref. [30], no *a posteriori* operations are required after advection. A simple T-junction advection is shown in Fig. 6 as an illustrative example to describe the evolution method (Section 3.4.1, 3.4.2 and 3.4.3) for different types of cells. In Fig. 6, the computational domain is a 4×4 square and contains 4×4 cells. The background velocity is $\mathbf{v} = (0, 1)$ and time step is set as $\Delta t = 0.5$.

3.4.1. Updating a cell with full type V_s

For a cell with full type V_s, the local level set $\phi_{i,j}^{1,(n)} = \phi_{i,j}^n(\chi_1)$ and the intermediate data $\phi_{k,l}^{1,(s)} = \phi_{k,l}^{(s)}(\chi_1)$ in the stencil neighborhood are constructed referring to the primary indicator χ_1. The intermediate-step update $\phi_{i,j}^{1,(s+1)}$ is obtained from

$$\begin{aligned} \phi_{i,j}^{1,(s+1)} &= \alpha_s \phi_{i,j}^{1,(s)} + (1 - \alpha_s)\left[\phi_{i,j}^{1,(s)} - \Delta t \mathbf{v}_{i,j}^n \cdot \left(\nabla \phi^1\right)_{i,j}^{(s)}\right], \\ & \qquad s = 0, \ldots, m, \\ \phi_{i,j}^{1,(0)} &= \phi_{i,j}^{1,n}, \quad \phi_{i,j}^{1,(m+1)} = \phi_{i,j}^{1,n+1} \end{aligned} \tag{8}$$

where m is the number of sub-steps, α_s is the parameter of Runge–Kutta sub-step s, and $\left(\nabla \phi^1\right)_{i,j}^{(s)}$ is the finite difference (FD) approximation of the spatial derivative at the center of a FV cell $C_{i,j}$.

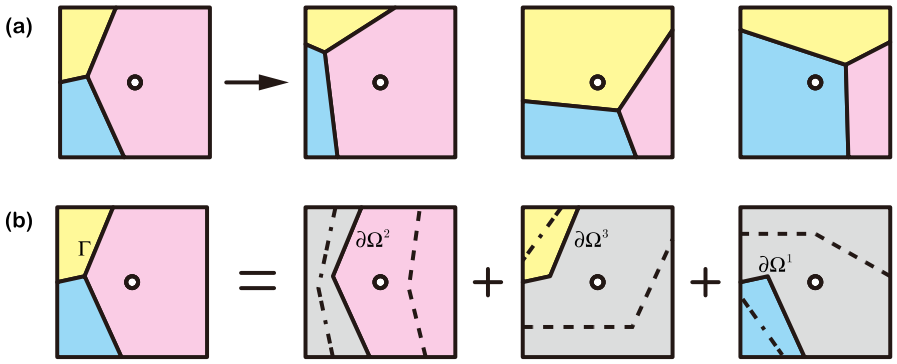


Fig. 5. A schematic representations of a 3-region (region domains are colored by blue ($\chi = 1$), yellow ($\chi = 3$) and red ($\chi = 2$)) cell. (a) The evolution of this complex-region cut cell consisting of 3 regions, $\mathcal{N}_s = 3$. Initially the primary indicator is $\chi_1 = \chi_{i,j} = 2$, and the secondary indicators are $\chi_2 = 1$ and $\chi_3 = 3$. After one sub-step of advection, the primary indicator may be unchanged, or have changed to $\chi_1 = 1$ or $\chi_1 = 3$. (b) the surface for each individual region may shrink (dashed line) or expand (dash-dotted line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The updated global level set at the sub-step $s + 1$ is $\varphi_{i,j}^{\chi,(s+1)} = (\phi_{i,j}^{1,(s+1)}, \chi_1)$. For example, consider the cell A in Fig. 6(a) which is a cell with full type V_s. After solving the advection equation (8), the evolved local level-set value is $\phi^1(\Delta t) = 0.7$. Thus the global level set at $t = \Delta t$ is updated by $\varphi^{\chi}(\Delta t) = (0.7, 1)$.

3.4.2. Updating a cell with two-region type V_s

For a cell with two-region type V_s, the construction of the multiple local level-set fields and the global level-set field updates are obtained essentially by the same operations as for a full cell. The difference is that the intermediate global level-set field update depends on the sign of $\phi_{i,j}^{1,(s+1)}$

$$\varphi_{i,j}^{\chi,(s+1)} = \begin{cases} (\phi_{i,j}^{1,(s+1)}, \chi_1) & \text{if } \phi_{i,j}^{1,(s+1)} \geq 0 \\ (-\phi_{i,j}^{1,(s+1)}, \chi_2) & \text{otherwise.} \end{cases} \tag{9}$$

As shown in Fig. 6(a), cell B changes its region indicator after one time-step. As the evolved local level-set value becomes negative, $\phi^1(\Delta t) = -0.3$, the global level-set becomes $\varphi^{\chi}(\Delta t) = (-(\phi^1(\Delta t)), \chi_2) = (0.3, 1)$.

3.4.3. Updating a cell with complex-region type V_s

For a cell with complex-region type V_s, the local level-set field is constructed and updated referring to the primary indicator χ_1. If $\phi_{i,j}^{1,(s+1)}$ does not change sign, the new intermediate local level set is $\varphi_{i,j}^{\chi,(s+1)} = (\phi_{i,j}^{1,(s+1)}, \chi_1)$. Otherwise, the local level-set field is constructed and updated referring to the secondary indicators

$$\phi_{i,j}^{m,(s+1)} = \alpha_s \phi_{i,j}^{m,(s)} + (1 - \alpha_s)\left[\phi_{i,j}^{m,(s)} - \Delta t \mathbf{v}_{i,j}^{n} \cdot \left(\nabla \phi^m\right)_{i,j}^{(s)}\right], \quad \chi_m \in V_s, \tag{10}$$

and the global level set is reconstructed by the operator $\mathbf{R}$. This case is illustrated by Fig. 6(d) where the cell C in Fig. 6(a) is advected from $t = 0$ to $t = \Delta t$. After solving 3 advection equations, the corresponding local level-set values are $\phi^1(\Delta t) = -0.3$, $\phi^2(\Delta t) = 0.3$ and $\phi^3(\Delta t) = -0.32$, respectively. Then according to the reconstruction operator (6), the global level-set value is $\varphi^{\chi}(\Delta t) = \mathbf{R}(\phi^1(\Delta t), \phi^2(\Delta t), \phi^3(\Delta t)) = (0.3, 1)$.

Note that our algorithm does not depend on how the indicator function χ is assigned for different regions as χ is only an auxiliary function to identify whether different cells in the domain belong to the same region or not. For example, if one changes the χ values in Fig. 6(a), say $\chi = 1$ for blue, $\chi = 2$ for yellow and $\chi = 3$ for red, one obtains the same interface locations after advection. The reason for using different advection strategies based on the cell types is the following. A direct transport of the global unsigned level-set function which exhibits a discontinuity across the interface

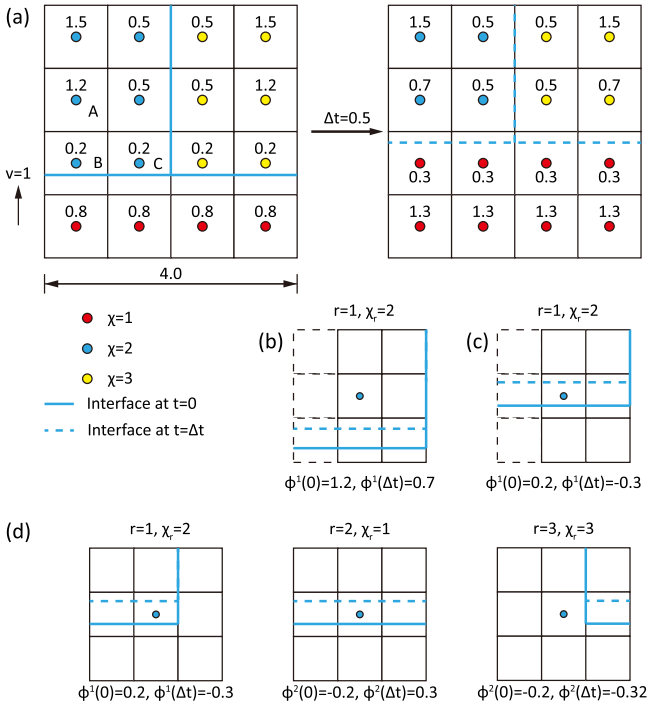


Fig. 6. A schematic representation example of evolution step for different types of cells. In (a), a T-junction is advected from $t = 0$ to $t = \Delta t = 0.5$ under a constant velocity field $\mathbf{v} = (0, 1)$. The interface location at $t = \Delta t$ is represented by dashed line and the corresponding global level-set function are updated in the right side of (a). The three chosen cells, A, B, and C , have different types of V_s and are updated in (b), (c) and (d), respectively.

requires numerical diffusion for stabilization and thus produces a smeared interface. Consequently, the unsigned scalar function does not maintain the distance function property. Advection with a high-order, low dissipation scheme of the local signed level-set functions, however, maintains the sharp interface and the distance property much more accurately. We advect the unsigned level-set function for full type cells and the local constructed signed level-set functions for other cells. This essentially is an adaptive algorithm which first determines the cell type as the indicator of the local smoothness of the unsigned level-set. Then the unsigned level-set field is advected wherever the unsigned level-set is smooth or the signed level-set field is advected wherever the unsigned level-set is singular. The re-initialization is not subject to regularization constraints, and our numerical examples show that it is required only as frequently as with the original level-set method.

Computational efficiency can be further increased by employing the narrow band technique [41]. For simple test cases, such as in Sections 4.1.2 and 4.1.3, this is not necessary. For more complex cases, such as in Sections 4.2.2 and 4.3.1, it is applied. The operation count of our method per time step is $\mathcal{O}(n^2)$ (same as with the Semi-Lagrangian regional level-set method and original level-set method) and can be reduced to $\mathcal{O}(kn)$ by employing the narrow band technique [41], where k is the band width and n is the number of cells in any direction. For the entire system the construction and reconstruction operators are applied $d(2t+1)N_1 + 2d(2t+1)N_2 + \mathcal{N}_s d(2t+1)N_3$ and N_3 times respectively, where N_1, N_2 and N_3 are the number of cells with full type, two-region type and complex-region type V_s in the narrow band of interface network, respectively. At every Runge–Kutta sub-step we solve the local advection equations $N_1 + N_2 + \mathcal{N}_s N_3$ times.

3.5. Re-initialization

It is important to re-initialize the global level set when necessary for maintaining its distance-function property with respect to the interface network. We employ two re-initialization methods. One widely used method is that of Sussman et al. [42] where a re-initialization equation is solved iteratively until steady-state is reached. The implementation for the regional level-set method involves the following steps region by region. For each subdomain Ω^χ, $\chi \in \mathrm{X}$, first the local level set ϕ^r is constructed and the re-initialization equation according to Ref. [42] is iterated until a steady state is reached. Alternatively, the explicit one-step method developed by Fu et al. [43] can be used which is significantly faster than iterative methods.

3.6. Summary of the numerical method

Here we summarize our high-resolution regional level-set method and comment on the description above. The entire numerical method contains the following steps:

1. **The initialization step.** According to the description in Section 3.1, we define an initial global level set $\varphi^\chi_{i,j} = (\varphi_{i,j}, \chi_{i,j})$ at each FV cell $C_{i,j}$ center $\mathbf{x}_{i,j}$, where $\varphi_{i,j}$ is the distance from the interface network and $\chi_{i,j}$ indicates which region domain the center of $C_{i,j}$ is located at.
2. **The evolution step** contains three sub-steps for $C_{i,j}$: (1) construction of the $\mathcal{N}_s$ local level-set fields $\phi^{r,n}_{k,l}$ for the current time-step n at the center of each cell $C_{k,l}$ which belongs to the stencil V_c of $C_{i,j}$; (2) computation of $\phi^{r,n+1}_{i,j}$ at the next time-step $n+1$ by solving the $\mathcal{N}_s$ local advection equations; (3) reconstruction of the new global level set $\varphi^\chi_{i,j}$ at the center of $C_{i,j}$ from the $\mathcal{N}_s$ new local level-set fields $\phi^{r,n+1}_{i,j}$ by the reconstruction operator $\mathbf{R}$.
3. **The re-initialization step.** Enforce the distance function property of the global level set $\varphi^\chi_{i,j}$ as the distance from the interface network. We emphasize that the interface-network transport re-initialization is not always necessary, see Section 4.
4. **The postprocessing step.** If necessary, extract the interface network by the triangulation method of Ref. [44].

The detailed scheme for numerical implementation is listed in Algorithm 2 . Our method only requires two global fields, φ and χ ($i_{\max} \times j_{\max}$ matrix). The operational complexity is almost the same as that of the Semi-Lagrangian regional level-set method. It should be mentioned that except for a postprocessing step the method does not need to extract explicitly the interface network, so the interface is fully implicit for multi-region problems. Another important feature of our method is that unlike using neighboring ϵ-level-set contours to reconstruct the interface network [39], our method directly captures the evolution of the interface network. The method is not a hybrid of the multiple level-set method and the original level-set method. It rather can be viewed as a regional level-set method employing locally signed level-set fields. As most of the operations are localized, the current method, like the original level-set method, is well suited for to parallelization. It is possible that the number of local region indicators is $\mathcal{N}_s = 3^d$ for every cell, for example, $\mathcal{N}_s = 9$ everywhere for a 2D 9-region system. In this case, the efficiency of our method is reduced to that of the multiple level-set method. However, for most multi-region systems such as that in Fig. 3(a), there are a small number of cells that have more than two local region indicators.

Algorithm 2 Algorithm for evolving a global level-set function defined on $i_{\max} \times j_{\max}$ FV cells, part I.

1: Allocate memory for φ and χ ($i_{\max} \times j_{\max}$ matrix) and initialize their values. Initialize the velocity field $\mathbf{v}$ ($i_{\max} \times j_{\max}$ matrix). ▷ the initialization step
2: Define cell neighborhoods, V_s (3×3 cell pointers) and V_c (two $2S+1 \times 1$ cell pointers), for each cell.
3: Construct the local level-set functions by a vector ($\mathcal{N}$ elements) ϕ^r. Create constructed intermediate local level-set fields by a 3×3 matrix (ϕ^s) and a vector (ϕ^c) with $2S+1$ elements.
4: Update the velocity field $\mathbf{v}$ (Optional).
5: Update the time step Δt (Optional).
6: Apply boundary conditions.
7: **for** $i = 1$ **to** $i_{\max}$, $j = 1$ **to** $j_{\max}$ **do**
8: Update the local index set X_s for $C_{i,j}$ according to V_s, e.g., for cell C in Fig. 6, $\mathrm{X}_s = \{1, 2, 3\}$ with $\chi_1 = 2$, $\chi_2 = 1$, and $\chi_3 = 3$. And update $\mathcal{N}_s \leftarrow n(\mathrm{X}_s)$.
9: **for** $r = 1$ **to** n **do** ▷ If $\mathcal{N}_s > 2$, $n \leftarrow \mathcal{N}_s$, otherwise $n \leftarrow 1$.
10: **for** $k = i - S$ **to** $i + S$ **do**
11: $\phi^s_k \leftarrow \mathbf{C}_r(\varphi^\chi_{k,j})$ ▷ Constructing local level sets in x direction of V_s
12: Use high-order FD scheme, e.g. WENO, to approximate $\partial\phi^s/\partial x$.
13: **end for**
14: **for** $l = j - S$ **to** $j + S$ **do**
15: $\phi^s_k \leftarrow \mathbf{C}_r(\varphi^\chi_{i,l})$ ▷ Constructing local level sets in y direction of V_s
16: Use high-order FD scheme to approximate $\partial\phi^s/\partial y$.
17: **end for**

Algorithm 3 Algorithm for evolving a global level-set function defined on $i_{\max} \times j_{\max}$ FV cells, part II.

18: Update $\mathbf{n}$ and κ by local level-set ϕ^c in V_c (Optional).
19: Update local level-set value, $\phi^r \leftarrow \phi^s_{1,1} - \mathbf{v}_{i,j} \cdot \nabla\phi^s$ ▷ Explicit Euler time-integration scheme
20: **end for**
21: Reconstruct new global level set $\varphi^\chi_{i,j}$ from the $\mathcal{N}_s$ evolved local level-set fields ϕ^r:
22: **if** $\mathcal{N}_s = 1$ **then**
23: $\varphi^\chi_{i,j} \leftarrow (\phi^1, \chi_1)$.
24: **else if** $\mathcal{N}_s = 2$ **then** ▷ Cell D in Fig. 6(a)
25: **if** $\phi^1 > 0$ **then**
26: $\varphi^\chi_{i,j} \leftarrow (\phi^1, \chi_1)$.
27: **else**
28: $\varphi^\chi_{i,j} \leftarrow (-\phi^1, \chi_2)$.
29: **end if**
30: **else** ▷ Cell A in Fig. 6(a)
31: $\varphi^\chi_{i,j} = \mathbf{R}(\phi^r, r \in \mathrm{X}_s)$.
32: **end if**
33: **end for**
34: Re-initialize the level-set field φ (Optional).
35: Postprocessing step (Optional).
36: update $t \leftarrow t + \Delta t$ and continue to step 4 until the final time.

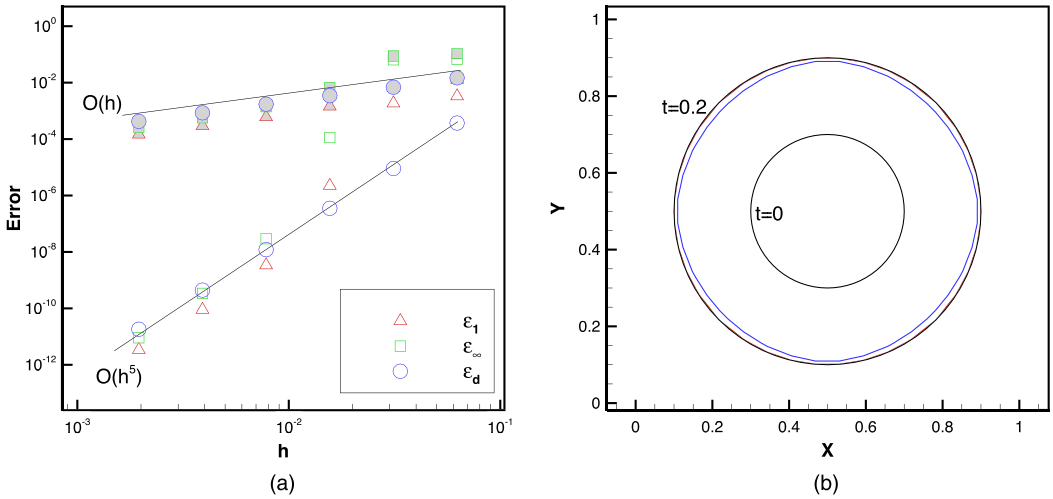


Fig. 7. Errors ε_1, ε_∞ and ε_d with increasing resolution (a) and the interfaces segmentation (16×16 grid points) at $t = 0.2$ (b) for a circle expansion. Present method (empty symbols and red line) is compared with Semi-Lagrangian regional level-set method (gray symbols and blue line). The exact interfaces at $t = 0$ and $t = 0.2$ are plotted with solid lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Numerical validation

In this section, we assess accuracy and efficiency, the present method by a range of numerical examples. Our intention is to show that we recover the high computational efficiency of the Semi-Lagrangian (SL) method, whilst significantly improving the prediction accuracy. First, we compare different high-order discretizations of the level-set transport equation with SL results for constant rotation motion of a three-region case. Afterwards, two simple mean curvature flows are considered to verify the construction operators and re-initialization methods. Subsequently, the single vortex flow is used to demonstrate the ability of the present method to resolve long thin filaments. The computations of normal driven flow, mean curvature flow and their combination serve to assess the accuracy of the present method. Finally, we couple the present method with Navier–Stokes equations applied to dry-foam dynamics which undergo sudden breakups and interface evolution driven by shock–bubble interactions, in order to demonstrate the capability for coping with complex configurations.

4.1. Simple test cases

Three simple cases are considered to test suitable high-resolution finite-difference schemes, construction operators, and re-initialization methods for the simulation of multi-region problems. We consider the 5th-order weighted essentially non-oscillatory scheme (WENO) [45] and the central-upwind weighted essentially non-oscillatory scheme (WENO_CU6) [46]. Both are compared with the SL scheme of the regional level-set method [37].

4.1.1. Circle expansion

We start with a 2-region expansion case in Ref. [44]. A circle with radius of 0.2 expands with a uniform speed to $t = 0.2$. The computational domain is $[0, 1] \times [0, 1]$. The explicit Euler scheme is used for time marching with a CFL number of 0.5. The 5th-order WENO scheme is employed for spatial discretization. Fig. 7(a) shows three error measures

$$\varepsilon_1 = \int_0^T \|\varphi\|_1 \, \mathrm{d}x, \quad \varepsilon_\infty = \int_0^T \|\varphi\|_\infty \, \mathrm{d}x, \quad \varepsilon_d = \int_0^T d_H(\Gamma^n, \Gamma^e) \, \mathrm{d}x, \tag{11}$$

with

$$\|\varphi\|_1 = \frac{1}{n(B)} \sum_{(i,j)\in B} |\varphi^n_{i,j} - \varphi^e_{i,j}|, \qquad \|\varphi\|_\infty = \max_{(i,j)\in B} |\varphi^n_{i,j} - \varphi^e_{i,j}|,$$

$$B = \{(i,j) | \varphi^n_{i,j} < 10\Delta x\}. \tag{12}$$

The Hausdorff distance is

$$d_H(\Gamma^n, \Gamma^e) = \max\Big(\sup_{\mathbf{y}\in\Gamma^e} \inf_{\mathbf{x}\in\Gamma^n} \|\mathbf{x} - \mathbf{y}\|_2, \ \sup_{\mathbf{x}\in\Gamma^e} \inf_{\mathbf{y}\in\Gamma^n} \|\mathbf{x} - \mathbf{y}\|_2 \Big). \tag{13}$$

The superscripts 'e' and 'n' stand for the exact and the numerical solutions. Our method achieves the expected 5th-order convergence rate for all three error measures. The SL method is 1st-order, as expected, as is the Voronoi implicit interface method, see Fig. 4 of Ref. [44]. Fig. 7(b) shows the interfaces of our method and SL method at $t = 0.2$, compared the exact solution.

4.1.2. Triple point advection

We assess the prediction accuracy of our method by triple point advection cases which have analytical solutions. The first configuration contains only one triple point which initially is located at $(0.2, 0.5)$. The second is a circle of radius $r_0 = 0.3$ divided into two parts. The computational domain is $[0, 1] \times [0, 1]$. The velocity field is given by $(u, v) = (1.0, 0.0)$. The explicit Euler scheme is used for time marching with a CFL number of 0.6. The 5th-order WENO scheme is used for spatial discretization. The simulations are performed until time $t = 0.4$. Re-initialization is not employed. Error measures of Eq. (11) are computed within a narrow band

$$B = \{(i,j) | \varphi^n_{i,j} < 1.2\Delta x \cap \|\mathbf{x}_{i,j} - \mathbf{x}_s\|_2 < 0.05L\}, \tag{14}$$

where $\mathbf{x}_{i,j}$ and $\mathbf{x}_s$ are the locations of cell center and triple points, respectively. Due to the smoothness properties of the exact solution at most first order accuracy in ε_d can be achieved. This is reproduced by our method, as shown in Fig. 8(a) and (b). The benefit of the high-resolution discretization becomes evident for the global measures ε_1 and ε_∞. The interface recovers the exact solution as the resolution increases. A comparison with the SL regional level-set method, shown in Fig. 8(b), demonstrates that the error magnitude of the SL regional level-set method is significantly larger than that of our method.

4.1.3. Constant rotation of three regions

A two-dimensional circle of radius $r_0 = 0.3$ is divided into two equal parts which undergo constant rotation. The computational

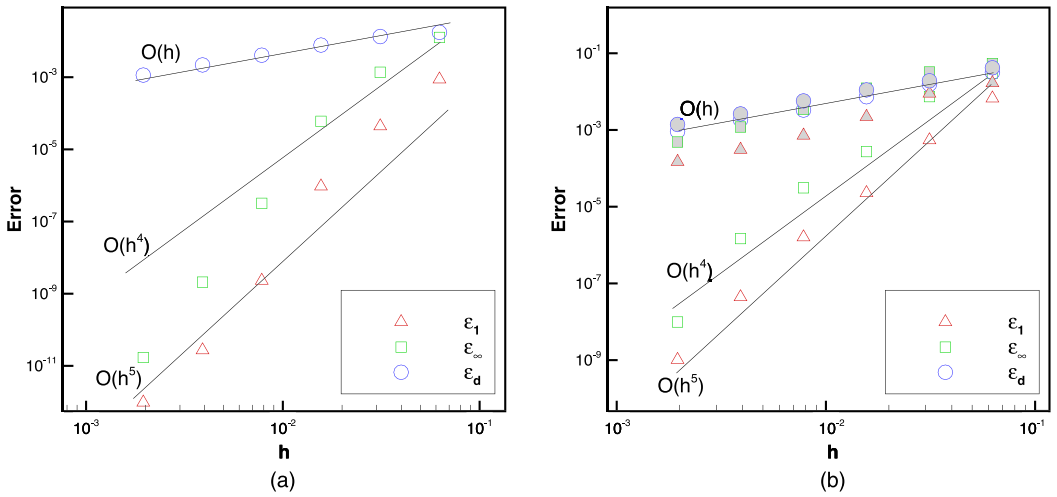


Fig. 8. Errors ε_1, ε_∞ and ε_d with increasing resolution for the single (a) and double (b) triple point advection. Present method (empty symbols) is compared with SL regional level-set method (gray symbols).

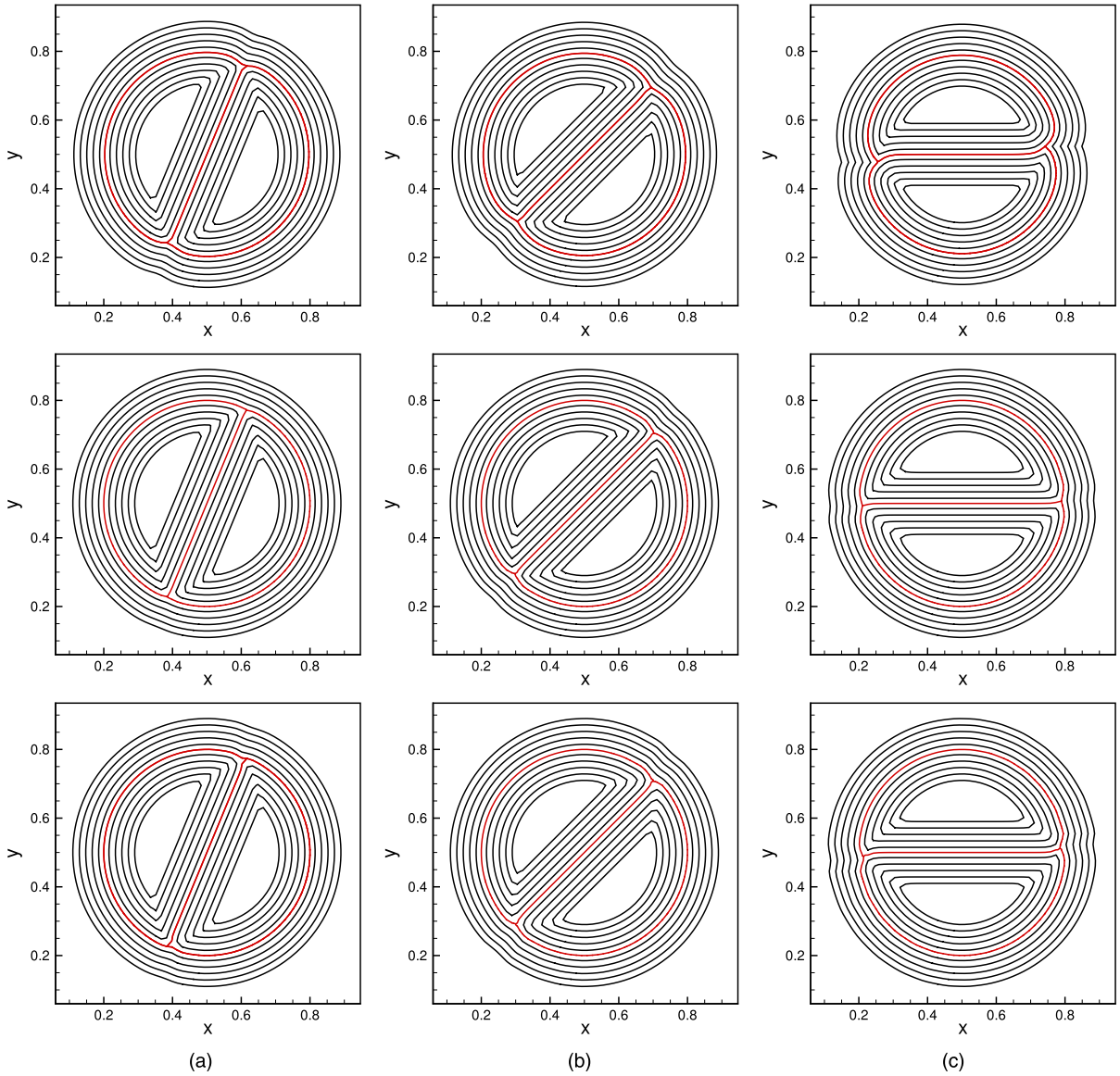


Fig. 9. Unsigned level-set contours ranging from 0.015 to 0.07: (a) $t = \frac{1}{8}\pi$, (b) $t = \frac{1}{4}\pi$, (c) $t = \frac{1}{2}\pi$. The results are obtained by using three different schemes: SL (first row), 5th-order WENO (middle row) and WENO_CU6 (last row).

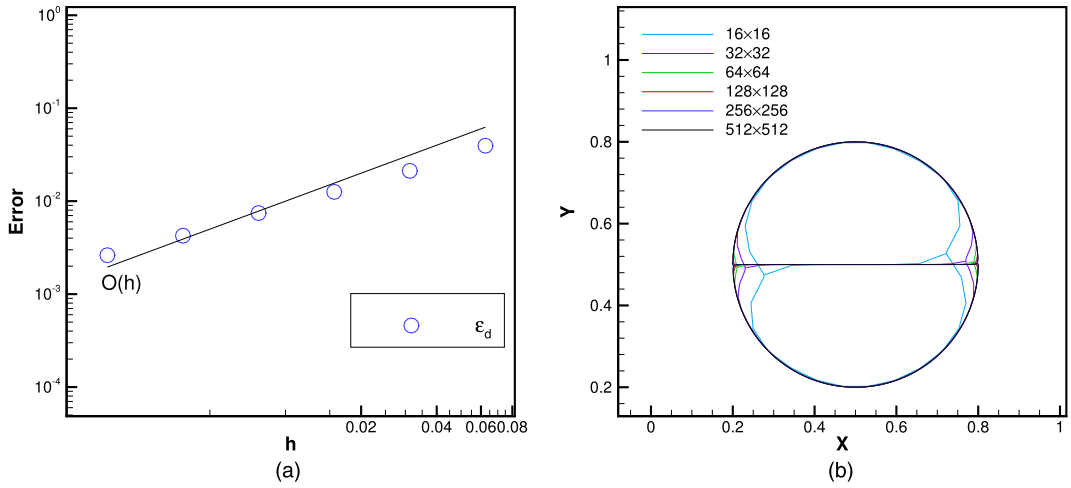


Fig. 10. (a) Error ε_d with increasing grid resolution for constant rotation case. (b) Segmentation of the interface networks at $t = \frac{1}{2}\pi$.

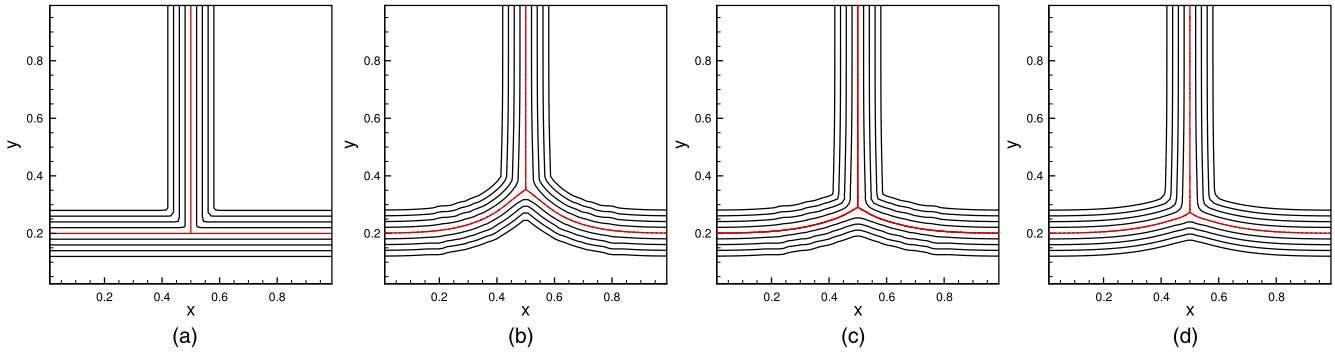


Fig. 11. Unsigned level-set contours ranging from 0.02 to 0.06: (a) initial contours; (b) $t = 0.02$, explicit re-initialization method and $\mathbf{C}_\Gamma$; (c) $t = 0.02$, explicit re-initialization method and $\mathbf{C}_\Gamma^\star$; (d) $t = 0.02$, iterative re-initialization method and $\mathbf{C}_\Gamma^\star$.

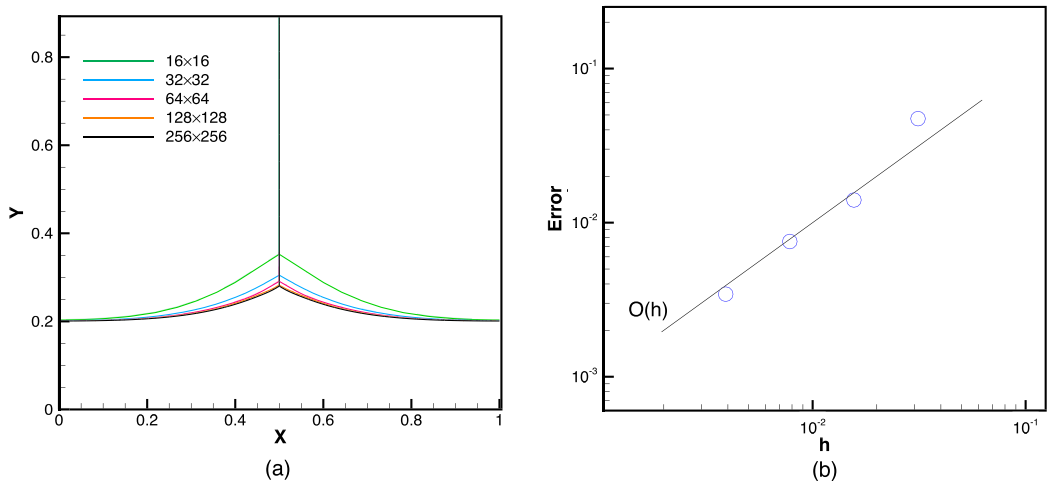


Fig. 12. Convergence for mean curvature flows in Fig. 11: (a) interface segmentation and (b) error of triple point locations.

domain is $[0, 1] \times [0, 1]$. Symmetry conditions are employed at all boundaries. The grid spacing is $h = \frac{1}{64}$. The explicit Euler scheme is used for time marching with a CFL number of 0.6. Two high-resolution schemes (5th-order WENO and WENO_CU6), see the last two rows of Fig. 9, are compared with respect to their ability to capture the two triple points, in comparison with the results of the 1st-order SL scheme, see the first row of Fig. 9. Three unsigned level-set contours are shown in Fig. 9 at $t = \frac{1}{8}\pi$, $t = \frac{1}{4}\pi$, and $t = \frac{1}{2}\pi$. Apparently, in each simulation, the deflection angles of the separation interface at all times agree with the theoretical values. However, the level-set contours surrounding the interface alter slightly. The SL method results in more smeared

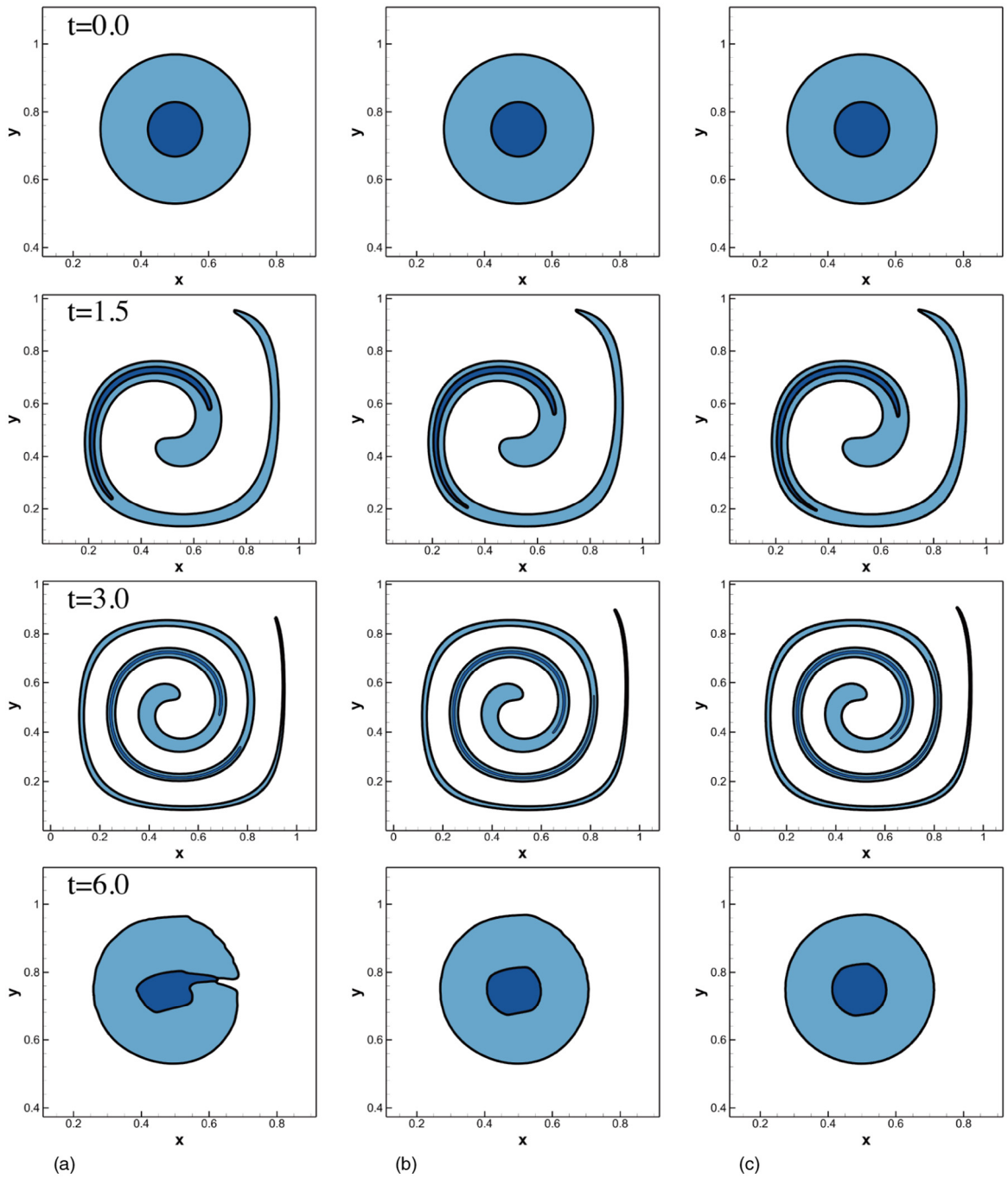


Fig. 13. Interface deformation for the single vortex flow at $t = 0$, 1.5, 3.0, and 6.0 with different resolutions: (a) $h = \frac{1}{256}$, (b) $h = \frac{1}{512}$, and (c) $h = \frac{1}{1024}$.

contours near the triple points due to numerical dissipation. The two high-resolution schemes have less numerical dissipation and thus preserve sharp corners. The relative radius difference $|\triangle r|/r_0$ of the circle for the WENO and WENO_CU6 results at $t = 0.5\pi$ are 0.211% and 0.285%, respectively, and are much smaller than for the SL result which is 3.42%, indicating better area conservation with WENO and WENO_CU6. An error analysis for this case, shown in Fig. 10(a), indicates 1st-order convergence of the interface location. The extracted interface network converges to the exact solution asymptotically as the resolution increases, as shown in Fig. 10(b).

4.1.4. Two dimensional mean curvature flows

The interface network of three-region and five-region systems is evolved under mean curvature flow with $\mathbf{u} = \kappa\mathbf{n}$, where $\mathbf{n}$ and κ are calculated by Eq. (4). In our simulation we employ the 5th-order WENO scheme for advection and consider different construction operators and re-initialization methods. Since the mean curvature uses 2nd-order derivatives of ϕ^r, it is more sensitive to the local constructed level-set fields. The computational domain extent is $[0, 1] \times [0, 1]$ in x and y directions. Symmetry boundary conditions are employed at all domain boundaries. The explicit Euler scheme is used for time marching, and the time step Δt is the same as that in [44], $\Delta t = \frac{h^2}{4}$. As depicted in Fig. 11, the T junction transforms to a Y junction under the effect of mean curvature. The different columns in this figure show results for different re-initialization methods and construction operators. When we use the construction operator $\mathbf{C}_r$, the displacement of the triple point is overestimated in both cases, see Fig. 11(b). Significant

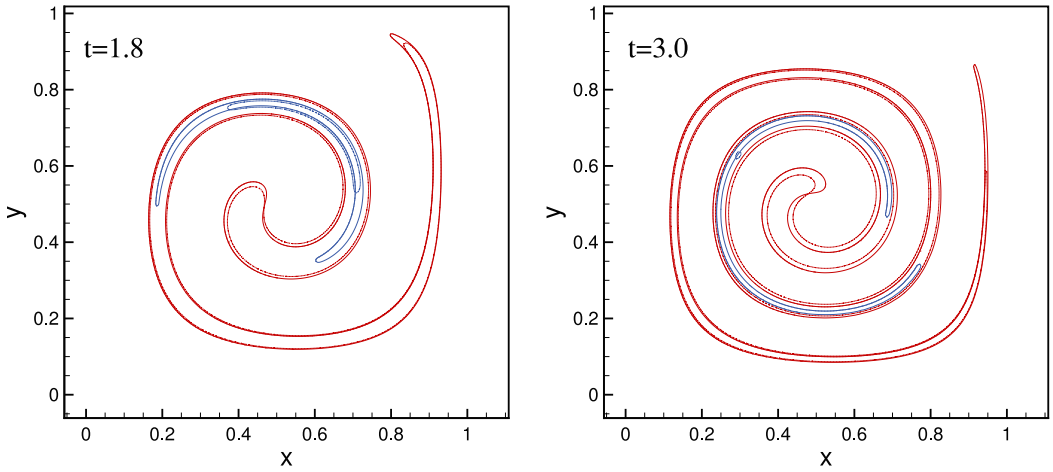


Fig. 14. Outer (red line) and inner (blue line) interfaces deformation for the single vortex flow with Semi-Lagrangian scheme (dotted line) and 5th-order WENO scheme (solid line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

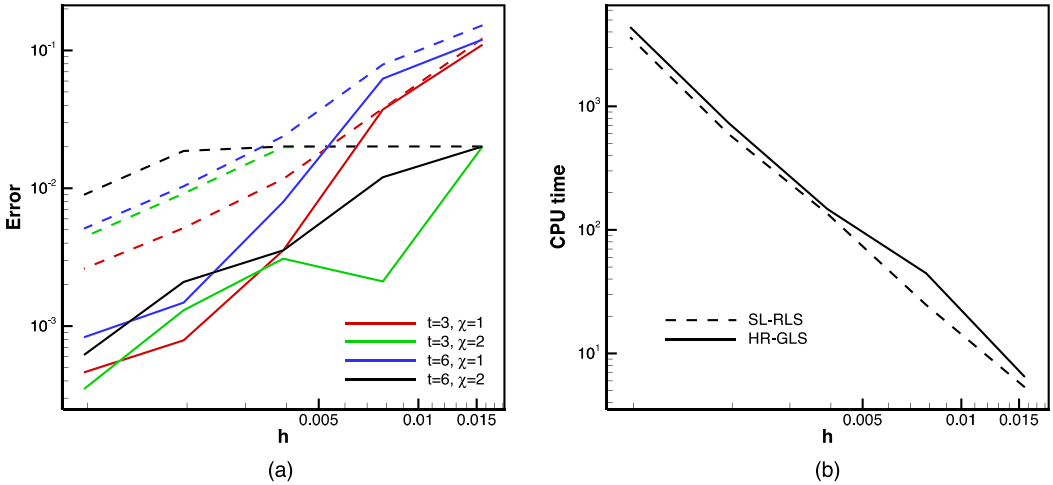


Fig. 15. (a) Area conservation error (%) and (b) CPU times (in seconds) of the Semi-Lagrangian regional level-set method (dashed lines) and our high-resolution method (solid lines) for simulating the single vortex flow. The area loss $\Delta A(t_0) = (\Delta A^1(t_0), \Delta A^2(t_0))$ and $\Delta A(t_1) = (\Delta A^1(t_1), \Delta A^2(t_1))$ are measured at physical time $t_0 = 3.0$ and $t_1 = 6.0$, where the superscripts 1 and 2 indicate the outer and inner circles, respectively. The CPU time is measured at $t = 3.0$.

improvement is observed by the operator $\mathbf{C}_r^\star$ where $\chi(\mathbf{x}) \neq \chi_r$, irrespective of the employed re-initialization method, see Fig. 11(c) and (d). Note that noise in the contours far away from the interface visible for the explicit re-initialization is irrelevant to our method as only the smooth inner contours are used for extracting the interface.

We conclude that the construction operator $\mathbf{C}_r^\star$ is more suitable for mean curvature flows. Both re-initialization methods are suitable for capturing the interface network. We obtain converged interface locations, as shown in Fig. 12(b), where the interface networks are extracted by the triangulation method of Ref. [44]. As shown in Fig. 12(b), first order convergence is achieved for the triple-point locations.

4.2. Interface advection

4.2.1. Single vortex flow

For the single vortex flow case, we use the setup introduced by Bell et al. [47] to test the ability and accuracy on resolving thin filaments under the deformation by the velocity field:

$$\begin{aligned} u &= -\sin^2(\pi x)\sin(2\pi y)\cos(2\pi y) \\ v &= \sin^2(\pi y)\sin(2\pi x)\,. \end{aligned} \tag{15}$$

Again, we employ the 5th-order WENO scheme for advection. The initial circle deforms into a filament wrapping around the center of the domain. This structure wraps back into the initial circle upon reversing the velocity at $t = 3$. The centers of concentric circles are $(0.5, 0.75)$ and their radii are 0.08 and 0.22, respectively. Symmetric boundary conditions and a second-order strongly stable Runge–Kutta scheme [48] with a CFL number of 0.6 are employed. The grid size is refined from $h = \frac{1}{256}$ to $h = \frac{1}{1024}$.

It can be seen from Fig. 13 that two spirals of long filaments are successfully captured at $t = 1.5$ and 3.0. With increasing grid resolution the filaments become longer, especially the inner one. At $t = 6.0$ with $h = \frac{1}{256}$ two additional unresolvable triple points are generated such that the two initial circle contours are connected. This phenomenon can be attributed to the fact that for extremely stretched filaments the two interfaces may reside in the same cell, and a small numerical perturbation leads to a topology change while reverse rotation does not disconnect the interface. We emphasize that although we use 256×256 cells overall, the number of cells across the initial inner circle is only 40×40. Increasing mesh resolution removes such artifacts, as shown in Fig. 13(b) and (c). Both circles recover their initial shapes with good area conservation.

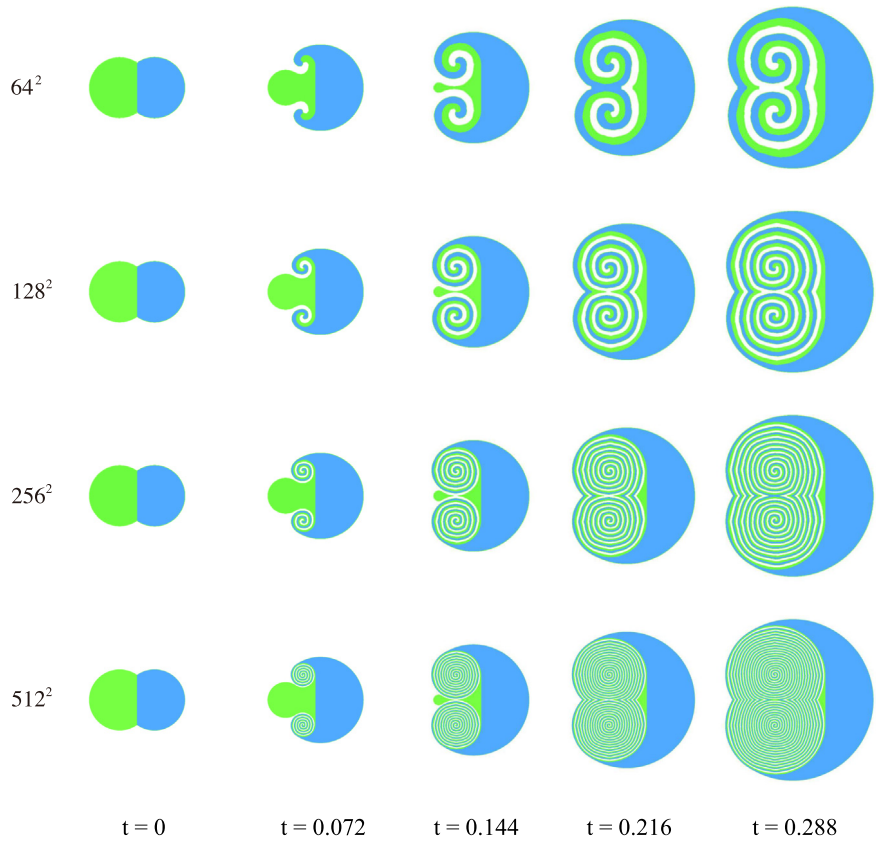


Fig. 16. Constant normal driven flow of three regions at $t = 0$, 0.072, 0.144, 0.216, and 0.288 with different resolutions ($h = \frac{1}{64}$, $h = \frac{1}{128}$, $h = \frac{1}{256}$, and $h = \frac{1}{512}$).

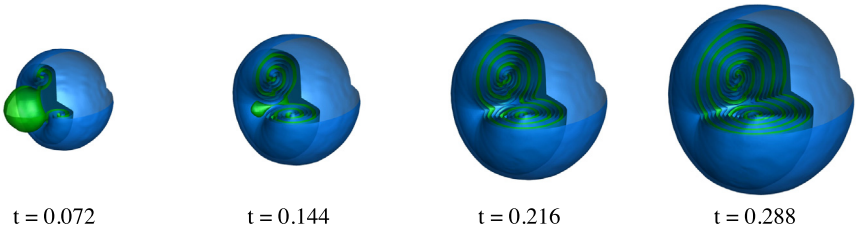


Fig. 17. 3D normal driven flow at $t = 0.072$, 0.144, 0.216, and 0.288 with $h = \frac{1}{128}$.

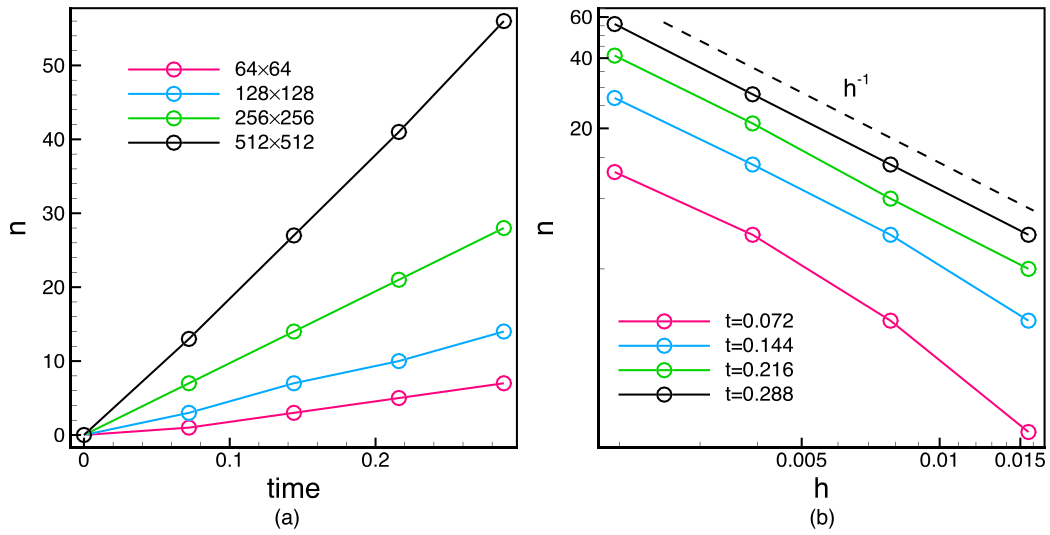


Fig. 18. Number of spirals for normal driven flow: (a) temporal evolution and (b) relation with resolution.

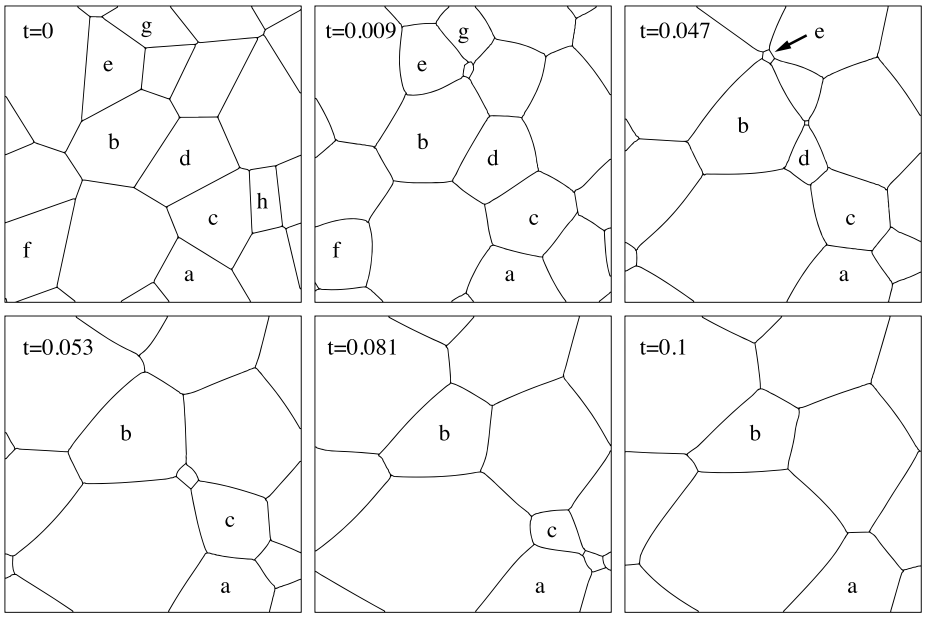


Fig. 19. Numerical result of the 15-region system evolution under mean curvature.

In Fig. 14, we compare the results of our method with that of the SL regional level-set method. We observe that the SL method shows large numerical dissipation and cannot reproduce the long filament. The area conservation errors are listed in Fig. 15(a). Note that the inner circle is very poorly resolved, i.e., for $h = \frac{1}{64}$, we have only 12×12 cells across the initial inner circle. With such low resolution 100% mass loss of the inner circle is inevitable for any method. At all three grid resolutions, our method exhibits significantly smaller area loss compared to the SL method. The CPU time measurement indicates that our method is nearly as fast as the SL regional level-set method even though we employ the high-order spatial schemes, as shown in Fig. 15(b). We conclude that our approach achieves improved accuracy and comparable computational cost compared with the SL regional level-set method.

4.2.2. Normal driven flow

The constant normal driven flow has been studied in [44]. The pairwise Γ_{ab} interface separating region domains Ω^a and Ω^b moves in its normal direction with a constant speed. As shown in Fig. 16, two initially neighboring region domains Ω^b (colored by green) and Ω^c (colored by blue) are surrounded by a background region domain Ω^a (colored by white). We define the normal velocity of the interface of each region as

$$\mathbf{u}_{\Gamma_{ab}} = \mathbf{n}_b, \mathbf{u}_{\Gamma_{bc}} = \mathbf{n}_c, \mathbf{u}_{\Gamma_{ca}} = \mathbf{n}_a, \tag{16}$$

where $\mathbf{u}_{\Gamma_{ab}}$ is the velocity at the pairwise interface Γ_{ab} and $\mathbf{n}_b$ is the normal direction of the region boundary $\partial\Omega_b$ at Γ_{ab}. Thus the flow wraps Ω^a into Ω^b which in turn is wrapped into Ω^c. The computation is carried out on a unit square with different resolutions employing the 5th-order WENO schemes for advection. Symmetry boundary conditions and the explicit Euler scheme are employed with a CFL number of 0.6.

For a comparison between the results of our method and those in [44] we plot five snapshots at the same time instants as those of [44], from $t = 0$ to $t = 0.288$. During evolution, the number of spirals increases quickly, and we observe that our results actually exhibit more visible spirals compared to that in [44], both on the same 256×256 grid, see third row of Fig. 16. This observation can be attributed to the high-resolution scheme employed in our method. Another reason is that we directly advect the interface network.

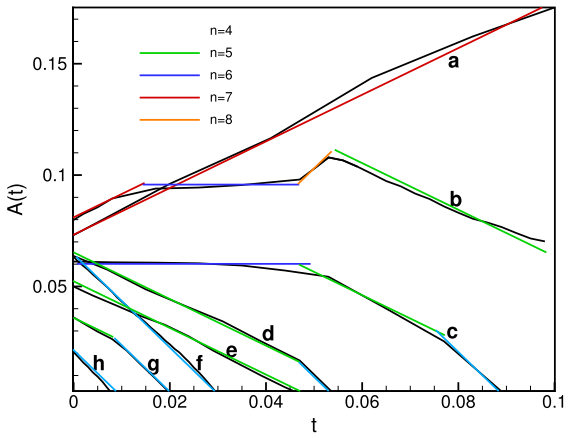


Fig. 20. Area as a function of time in the 15-region system. The colored lines indicate von Neumann-Mullins' law in Eq. (17). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The number of spirals increases linearly with time and proportionally to grid resolution, see Fig. 18(a) and 18(b), respectively. In order to demonstrate the validity of our method in three dimensions, a 3D example of normal driven flow on a $128 \times 128 \times 128$ grid, is shown in Fig. 17.

4.2.3. Von Neumann-Mullins' law validation

We consider a case involving more regions and which serves to verify our method in a configuration with multiple triple points within a multi-region system. Initially 15 regions are randomly placed in a domain $[0, 1] \times [0, 1]$ and evolve under a mean curvature generated velocity $\mathbf{u} = \kappa\mathbf{n}$. Periodic boundary conditions are imposed at the domain boundaries. The explicit Euler scheme is employed for temporal discretization. The time-step size is $\Delta t = \frac{h^2}{4}$ and the grid spacing is $h = \frac{1}{128}$. We want to verify that the von Neumann-Mullins' law is reproduced, which states that the rate of area A growth or decay is a function of the number of edges

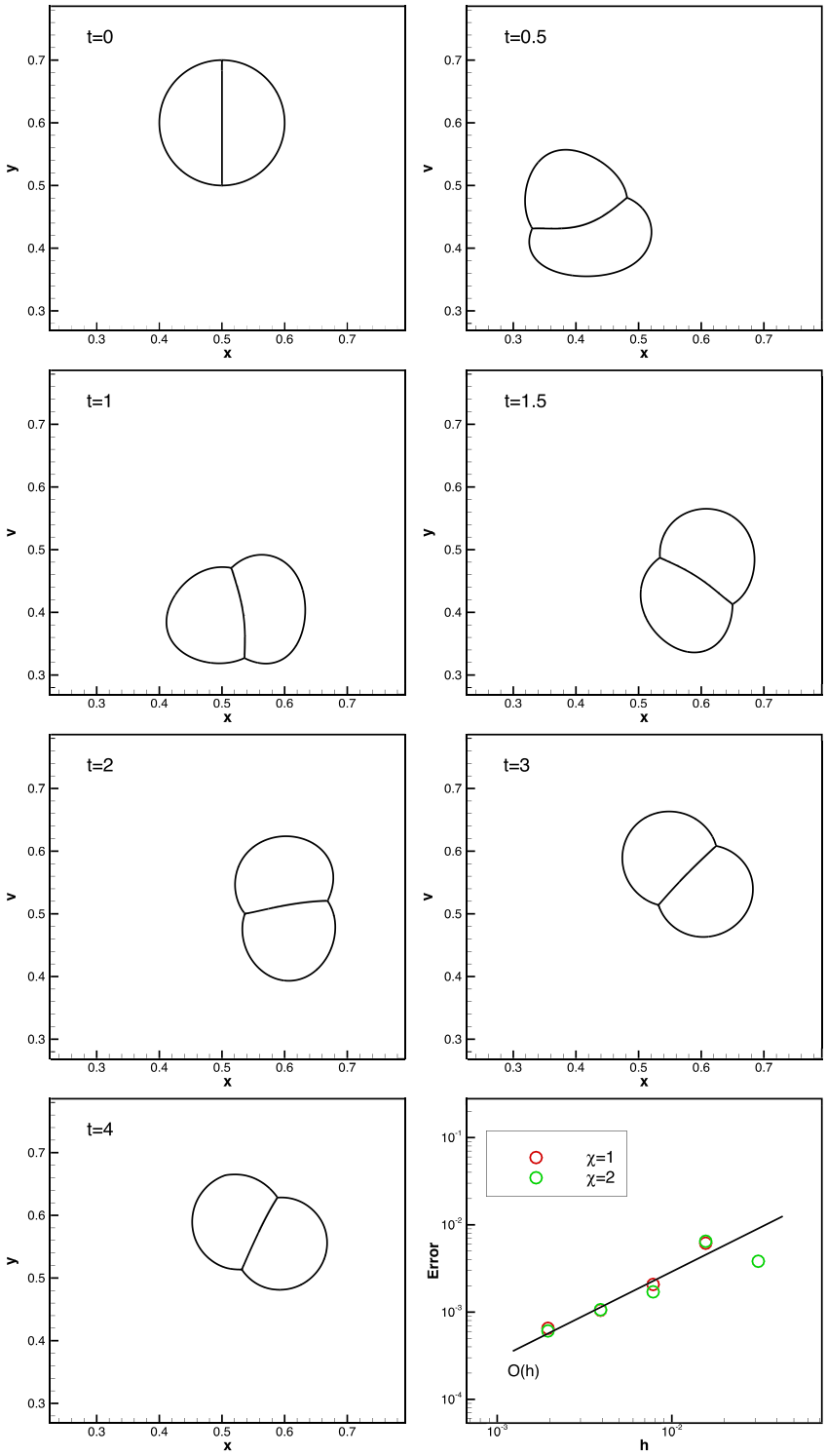


Fig. 21. Foam dynamics driven by surface tension and the background Taylor–Green vortex flow. The interface network is obtained by solving the N–S equations and global level-set advection equation with a grid size of $h = \frac{1}{512}$. The area loss for the two bubbles is measured at $t = 3.5$.

n of the phase [44]

$$\frac{dA}{dt} = 2\pi\gamma\left(\frac{n}{6} - 1\right), \tag{17}$$

where we set $\gamma = 1.0$. The initial number of regions 15 is successively reduced to 6 under the effect of mean curvature, as shown in Fig. 19. According to Fig. 19, region "h" initially has four edges and shrinks under mean curvature, leading to region destruction, consistent with von Neumann-Mullins' law. More specifically, the temporal growth and decay of the selected 8 regions is shown in Fig. 20 which exhibits a piecewise linear profile for each region, in agreement with von Neumann-Mullins' law which is indicated by a colored line in Fig. 20.

4.3. Coupled with fluid dynamics

To demonstrate the capability of our method to handle complex applications we couple the high-resolution regional level-set

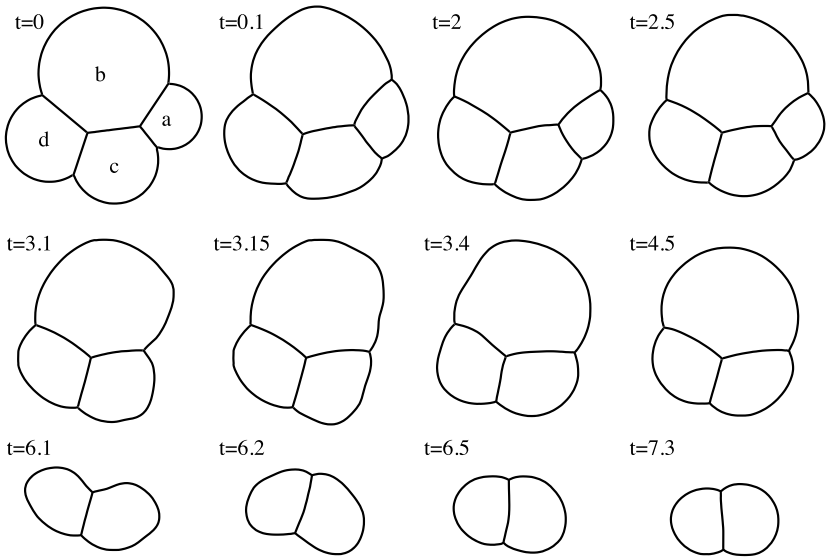


Fig. 22. Four-bubble cluster dynamics subject to sudden rupture. The breakup of regions Ω^{χ_a} and Ω^{χ_b} occurs at $t = 3.0$ and $t = 6.0$, respectively.

method with the Navier–Stokes (N–S) equations,

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}_c = \nabla \cdot \mathbf{F}_v + \sigma\kappa\delta\mathbf{n}, \tag{18}$$

where $\mathbf{U} = (\rho, \rho u, \rho v, \rho w, \rho E)^T$, in which ρ, u, v, ρw, and ρE are the density, the three velocity components, and the total energy, respectively. $\mathbf{F}_c$ and $\mathbf{F}_v$ are the convective and viscous flux tensor, respectively. The surface tension term $\sigma\kappa\delta\mathbf{n}$ on the right-hand side describes foam dynamics, with σ being the surface tension and δ being a smoothed Dirac delta function. The surface tension force is calculated by Eq. (14) of Ref. [44]. We note that the above surface-tension model may be improved by Eq. (16) in Ref. [44], which is, however, not our main concern here. We use the 5th-order WENO scheme and a 4th-order central scheme for spatial discretization and a second-order strongly stable Runge–Kutta scheme for temporal discretization, both for solving the N–S equation and level-set advection. The CFL number is 0.6.

4.3.1. Dry foam dynamics

First we simulate dry-foam dynamics driven by surface tension and a background Taylor–Green vortex flow [49],

$$\begin{aligned} u &= -U\cos(2\pi x)\sin(2\pi y) \\ v &= U\sin(2\pi x)\cos(2\pi y). \end{aligned} \tag{19}$$

In a dry-foam cluster the gas bubbles are separated by thin liquid films, which corresponds to the interface network of our method. As we are not concerned with a specific physical problem, inertia and gravity effects of liquid membranes, gas exchange across permeable membranes, and Marangoni forces at the liquid–gas interface are neglected here for simplicity. We consider the interconnected membranes as massless and infinitely thin. Periodic boundary conditions are applied. The Reynolds number is $\mathrm{Re} = \rho UL/\mu = 100$ and the Ohnesorge number is $\mathrm{Oh} = \mu/\sqrt{\rho\sigma L} = 0.1$. So the inertial force, viscous force and surface tension force cannot be neglected. We show the interface evolution at different time instants in Fig. 21. The initial foam cluster is not at the equilibrium state and evolves under the background vortex flow. The surface tension takes long time to enforce the foam cluster to an equilibrium state ($t \geqslant 3.5$) as the Ohnesorge number is large. The area loss is measured at $t = 3.5$ and exhibits 1st-order convergence in Fig. 21.

The next foam dynamics problem has an Ohnesorge number of $\mathrm{Oh} = 0.00024$, i.e., the flow is surface-tension dominated. Symmetry boundary conditions are applied and the grid size is $h = \frac{1}{256}$. Initially, four bubbles evolve from an artificial initial configuration to an equilibrium state, as shown in the first row of Fig. 22. Following the breakup of Ω^a, a second equilibrium state is attained under the surface tension force, see the second row of Fig. 22. Then Ω^b breaks up, and a similar process is observed for the remaining two bubbles. Note that the bubble breakup is triggered explicitly, which is unphysical. However, it serves our purpose of demonstrating that our method has the capability to capture the interface evolution of a foam cluster subjected to stimulated bubble breakup.

4.3.2. Shock–bubble interaction driven interface evolution

We consider the interface evolution is driven by a shock–bubble interaction, where a light gas bubble (helium) is initially accelerated by a Mach 6.0 planar shock and then impacts on a heavy bubble (R22). The shock wave, helium bubble center, and R22 bubble center are located at $x = 0.1$, $x = 0.15$ and $x = 0.22$, respectively. The bubble has a radius of 0.025. The initial condition is given by

$$(\rho, p, u, v, \gamma) = \begin{cases} (5.268, 41.83, 5.752, 0, 1.4) & \text{post-shocked air} \\ (1, 1, 0, 0, 1.4) & \text{pre-shocked air} \\ (0.138, 1, 0, 0, 1.667) & \text{helium bubble} \\ (3.154, 1, 0, 0, 1.249) & \text{R22 bubble} \end{cases} \tag{20}$$

and the ideal gas equation of state is used. The surface tension and viscosity are neglected, as the flow is advection dominated. This is a typical compressible multi-phase problem and we notice that our interface capturing method can be straightforward coupled with the ghost fluid method proposed in Ref. [50] to solve the dynamics of the three different materials. The computational domain is $[0, 0.356] \times [0, 0.0445]$. Symmetry boundary conditions are employed at the upper and lower boundaries, while inflow and outflow conditions are prescribed at the left and right boundaries. We perform high-resolution simulation ($h = 4.35 \times 10^{-5}$) to generate complex flow fields and validate our interface capturing method in resolving complex topology changes. As shown in Fig. 23(a) and (b), the density gradient and vorticity are plotted and complex shock reflection patterns are observed. The helium and R22 bubbles are colored by blue and green, respectively. After the helium bubble impacts on the heavy bubble, many triple points,

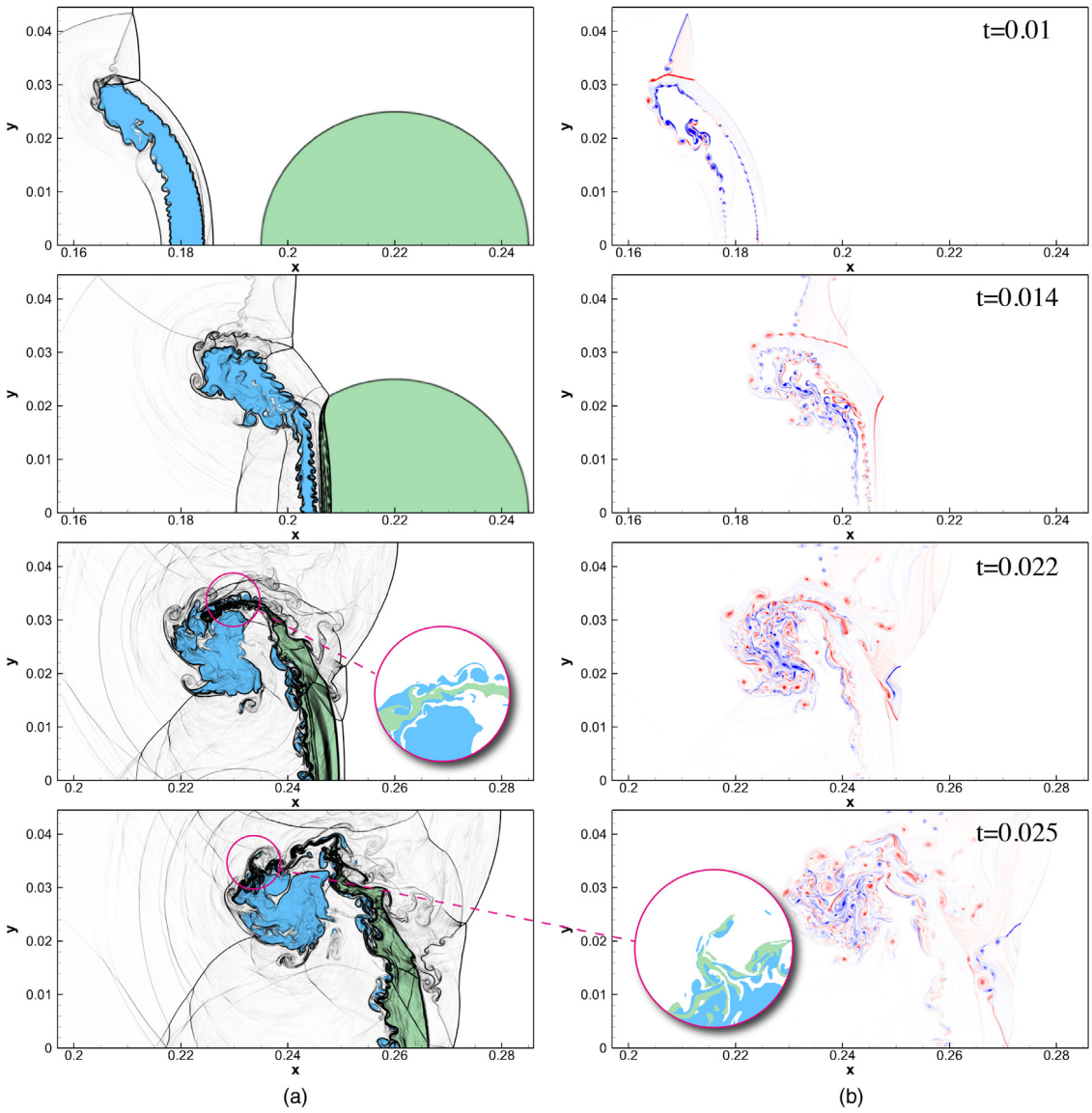


Fig. 23. Interface evolution driven by shock–bubble interactions at $t = 1 \times 10^{-2}$, $t = 1.4 \times 10^{-2}$, $t = 2.2 \times 10^{-2}$ and $t = 2.5 \times 10^{-2}$. The density gradient (a) and vorticity fields are plotted. The regions of helium bubble, R22 bubble and the ambient air are colored by blue, green and white, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

droplets and filaments are generated, as shown in the inserts of Fig. 23(a), indicating the ability of our method for resolving the evolution of these interface structures.

5. Concluding remarks

The proposed method employs locally constructed level-set fields and the regional level-set method to overcome typical problems encountered with the numerical simulation of multi-region problems. The proposed numerical approach for multi-region problems and its algorithmic formulation have the following main properties: (1) As the proposed regional level-set method employs local signed level-set fields generated from the global level-set field with a simple construction operator, it permits the implementation of high-resolution schemes for level-set transport in a straightforward way. (2) Instead of explicitly constructing the interface at every time step, we use a reconstruction operator to assemble the global level set from multiple local level-set fields. This way we can ensure that the implicitly defined topology has no void or overlap artifacts. Moreover, a simple algorithm allows to distinguish different types of cells. It can be concluded from a range of test cases that the proposed method is more accurate than the Semi-Lagrangian regional level-set method. High order accuracy is demonstrated for some simple test cases where analytical results are known. Several increasingly complex configurations serve to demonstrate that improved accuracy and efficiency transfer to such test cases. A region-deconstruction example in foam dynamics and a shock–bubble interaction driven interface deformation demonstrate the feasibility of the method for complex applications.

Acknowledgments

This work is partially supported by China Scholarship Council (No. 201306290030), National Natural Science Foundation of China (No. 11628206) and Deutsche Forschungsgemeinschaft (HU 1527/6-1). The project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (No. 667483).

References

[1] M. Betney, B. Tully, N. Hawker, Y. Ventikos, Phys. Fluids (1994-Present) 27 (3) (2015) 036101.
[2] V.A. Thomas, R.J. Kares, Phys. Rev. Lett. 109 (7) (2012) 075004.
[3] B.M. Haines, F.F. Grinstein, J.R. Fincke, Phys. Rev. E 89 (5) (2014) 053302.
[4] K.-J. Chen, S. Woosley, A. Heger, A. Almgren, D.J. Whalen, Astrophys. J. 792 (1) (2014) 28.
[5] E.J. Lentz, S.W. Bruenn, W.R. Hix, A. Mezzacappa, O.B. Messer, E. Endeve, J.M. Blondin, J.A. Harris, P. Marronetti, K.N. Yakunin, Astrophys. J. Lett. 807 (2) (2015) L31.
[6] A. Wongwathanarat, E. Mueller, H.-T. Janka, Astron. Astrophys. 577 (2015) A48.
[7] D.L. Weaire, S. Hutzler, The Physics of Foams, Oxford University Press, 2001.
[8] A.-L. Biance, A. Delbos, O. Pitois, Phys. Rev. Lett. 106 (6) (2011) 068301.
[9] R.I. Saye, J.A. Sethian, Science 340 (6133) (2013) 720–724.
[10] Y. Kim, M.-C. Lai, C.S. Peskin, Y. Seol, J. Comput. Phys. 269 (2014) 1–21.
[11] S. Hilgenfeldt, S. Erisken, R.W. Carthew, Proc. Natl. Acad. Sci. 105 (3) (2008) 907–911.
[12] M. Rauzi, P. Verant, T. Lecuit, P.-F. Lenne, Nature Cell Biol. 10 (12) (2008) 1401–1410.
[13] M.L. Manning, R.A. Foty, M.S. Steinberg, E.-M. Schoetz, Proc. Natl. Acad. Sci. 107 (28) (2010) 12517–12522.
[14] E. Marinari, A. Mehonic, S. Curran, J. Gale, T. Duke, B. Baum, Nature 484 (7395) (2012) 542–545.
[15] M. Osterfield, X. Du, T. Schüpbach, E. Wieschaus, S.Y. Shvartsman, Dev. Cell 24 (4) (2013) 400–410.
[16] J. Geiger, A. Roosz, P. Barkoczy, Acta Mater. 49 (4) (2001) 623–629.
[17] C. Krill Iii, L.-Q. Chen, Acta Mater. 50 (12) (2002) 3059–3075.
[18] M. Elsey, S. Esedoğlu, P. Smereka, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 467 (2011) 381–401.
[19] C. Torres, M. Emelianenko, D. Golovaty, D. Kinderlehrer, S. Ta'asan, SIAM J. Appl. Math. 75 (2) (2015) 762–786.
[20] S.O. Unverdi, G. Tryggvason, J. Comput. Phys. 100 (1) (1992) 25–37.
[21] C.S. Peskin, Acta Numer. 11 (2002) 479–517.
[22] C. Hirt, A.A. Amsden, J. Cook, J. Comput. Phys. 14 (3) (1974) 227–253.
[23] K.A. Brakke, Exp. Math. 1 (2) (1992) 141–165.
[24] S. Galera, P.-H. Maire, J. Breil, J. Comput. Phys. 229 (16) (2010) 5755–5787.
[25] R. Loubère, P.-H. Maire, M. Shashkov, J. Breil, S. Galera, J. Comput. Phys. 229 (12) (2010) 4724–4761.
[26] M. Kucharik, R.V. Garimella, S.P. Schofield, M.J. Shashkov, J. Comput. Phys. 229 (7) (2010) 2432–2452.
[27] Y. Kim, M.-C. Lai, C.S. Peskin, J. Comput. Phys. 229 (13) (2010) 5194–5207.
[28] C.W. Hirt, B.D. Nichols, J. Comput. Phys. 39 (1) (1981) 201–225.
[29] S. Osher, J.A. Sethian, J. Comput. Phys. 79 (1) (1988) 12–49.
[30] B. Merriman, J.K. Bence, S.J. Osher, J. Comput. Phys. 112 (2) (1994) 334–363.
[31] H.-K. Zhao, T. Chan, B. Merriman, S. Osher, J. Comput. Phys. 127 (1) (1996) 179–195.
[32] D.P. Starinshak, S. Karni, P.L. Roe, J. Comput. Phys. 262 (2014) 1–16.
[33] T.F. Chan, M. Moelich, B. Sandberg, Image Processing Based on Partial Differential Equations, Springer, 2007, pp. 175–210.
[34] L.A. Vese, T.F. Chan, Int. J. Comput. Vis. 50 (3) (2002) 271–293.
[35] J. Lie, M. Lysaker, X.-C. Tai, Math. Comput. 75 (255) (2006) 1155–1174.
[36] G. Chung, L.A. Vese, Comput. Vis. Sci. 12 (6) (2009) 267–285.
[37] W. Zheng, J.-H. Yong, J.-C. Paul, Graph. Models 71 (6) (2009) 229–239.
[38] B. Kim, ACM Trans. Graph. 29 (6) (2010) 175.
[39] R.I. Saye, J.A. Sethian, Proc. Natl. Acad. Sci. 108 (49) (2011) 19498–19503.
[40] C.-W. Shu, S. Osher, J. Comput. Phys. 77 (2) (1988) 439–471.
[41] D. Adalsteinsson, J.A. Sethian, J. Comput. Phys. 118 (2) (1995) 269–277.
[42] M. Sussman, P. Smereka, S. Osher, J. Comput. Phys. 114 (1) (1994) 146–159.
[43] L. Fu, X. Hu, N. Adams, The Ninth Symposium on Turbulence and Shear Flow Phenomena: Melbourne, Australia, 30 June - 3 July, 2015.
[44] R. Saye, J.A. Sethian, J. Comput. Phys. 231 (18) (2012) 6051–6085.
[45] G.-S. Jiang, C.-W. Shu, Efficient Implementation of Weighted ENO Schemes. Tech. Rep., DTIC Document, 1995.
[46] X. Hu, Q. Wang, N.A. Adams, J. Comput. Phys. 229 (23) (2010) 8952–8965.
[47] J.B. Bell, P. Colella, H.M. Glaz, J. Comput. Phys. 85 (2) (1989) 257–283.
[48] M.O. Domingues, S.M. Gomes, O. Roussel, K. Schneider, J. Comput. Phys. 227 (8) (2008) 3758–3780.
[49] G. Taylor, A. Green, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 158 (895) (1937) 499–521.
[50] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, J. Comput. Phys. 152 (2) (1999) 457–492.

A.2 Paper II

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

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

In *Journal of Computational Physics*, Volume 354, 2018, pp. 311-319, DOI: `https://doi.org/10.1016/j.jcp.2017.10.054`.

Contribution: My contribution to this work was the development of the method and the corresponding computer code for its implementation. I performed simulations and analyzed the results, and wrote the manuscript for the publication.

Journal of Computational Physics 354 (2018) 311–319

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

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

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

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

ARTICLE INFO

Article history:
Received 23 March 2017
Received in revised form 26 October 2017
Accepted 27 October 2017
Available online 31 October 2017

Keywords:
Reinitialization
Regional level-set function
High-order accuracy
Interface preserving

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.

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 $\phi^0 : \mathbb{R}^d \to \mathbb{R}$ during advection by the signed distance function $\phi : \mathbb{R}^d \to \mathbb{R}$ which is the solution of the Eikonal equation,

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

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],

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

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

* Corresponding author.
E-mail addresses: shucheng.pan@tum.de (S. Pan), xiuxiu.lyu@tum.de (X. Lyu), xiangyu.hu@tum.de (X.Y. Hu), nikolaus.adams@tum.de (N.A. Adams).

https://doi.org/10.1016/j.jcp.2017.10.054
0021-9991/

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 $\mathbf{x} \in \Omega$ 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 reinitialization 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 $\mathrm{X} = \{r \in \mathbb{N} | 1 \leq r \leq \mathcal{N}\}$ 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

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

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,

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

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,

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

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 $r = \gamma$.

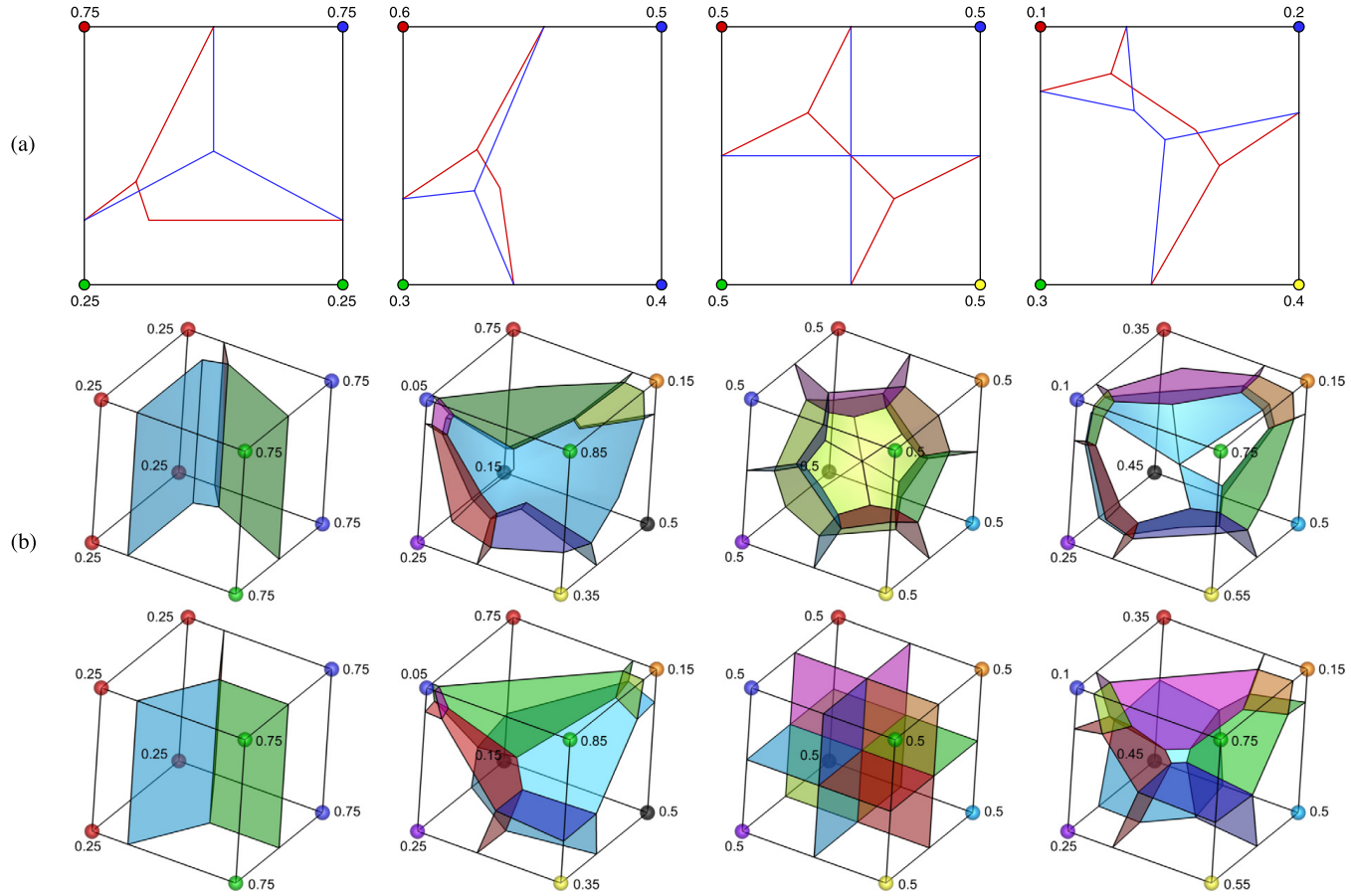


Fig. 1. Triangulation results for (a) 2D and (b) 3D cases. The regional level-set functions are defined at vertices, where color indicates χ and data indicate φ. The extracted interface patches are labeled by different colors for each pairwise interface. The results of the simple linear interpolation method and the proposed method are represented by red and blue lines in (a) and upper and lower rows in (b). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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

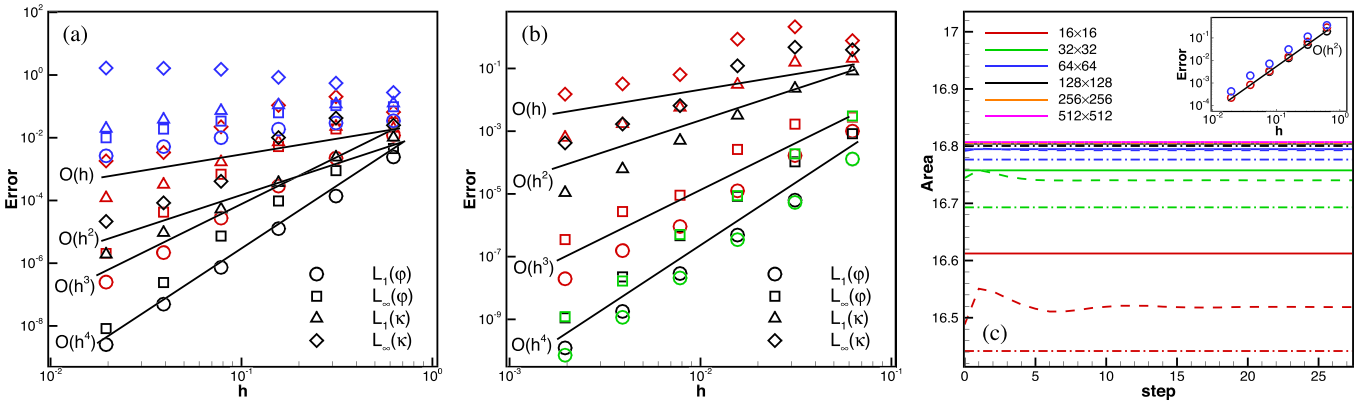


Fig. 2. Reinitialization for 2-region cases: errors of the first (a) and second (b) case, and (c) time history of area of the fist case. The blue, red, and black symbols indicate results of the low-order, subcell fix, and the high-order two-step method, respectively. The solid, dashed, and dash-dotted 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. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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

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 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 distinct 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

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

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

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

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

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

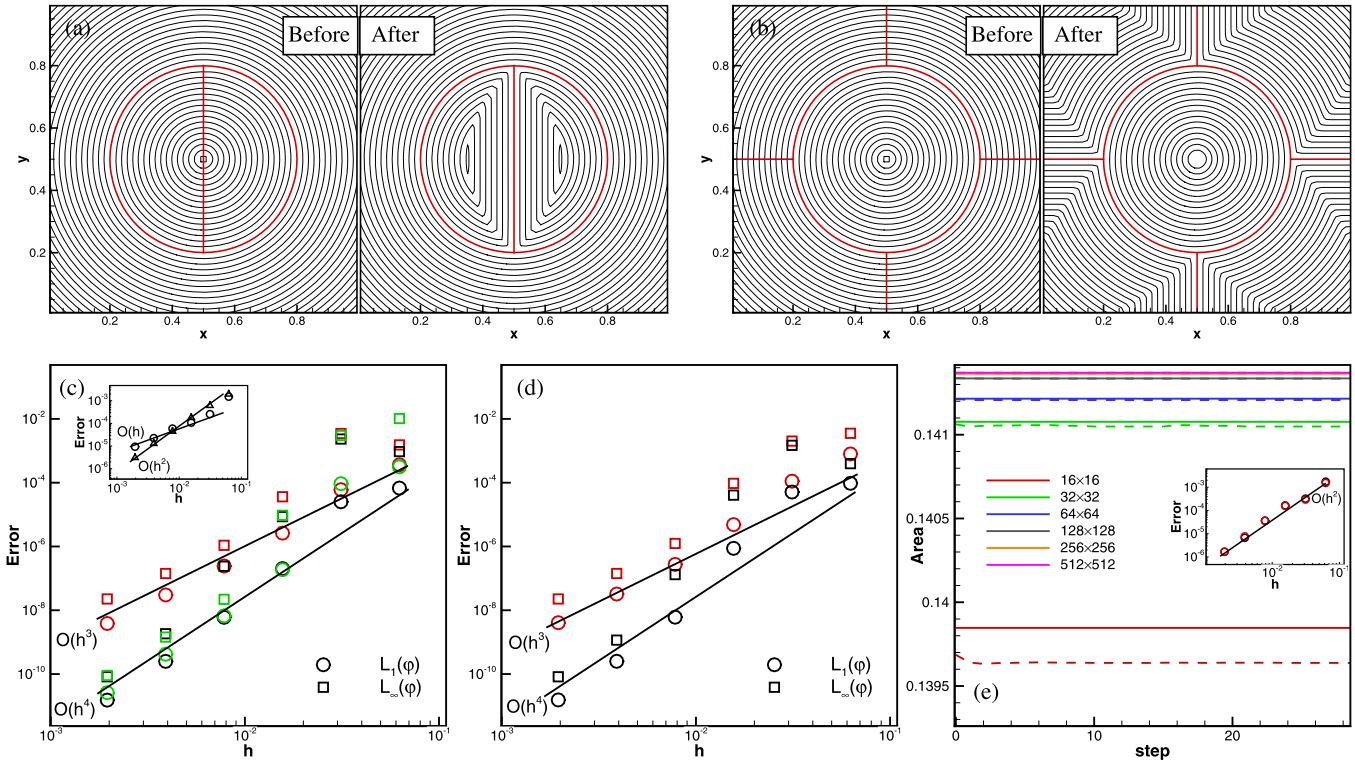


Fig. 3. Reinitialization for multi-region cases: φ contours for the 3-region (a) and the 5-region (b) case, errors of the 3-region (c) and the 5-region (d) case, and (e) time history of area of the 3-region case. The red and black symbols stand for results of the subcell fix and the high-order two-step 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. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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.

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).

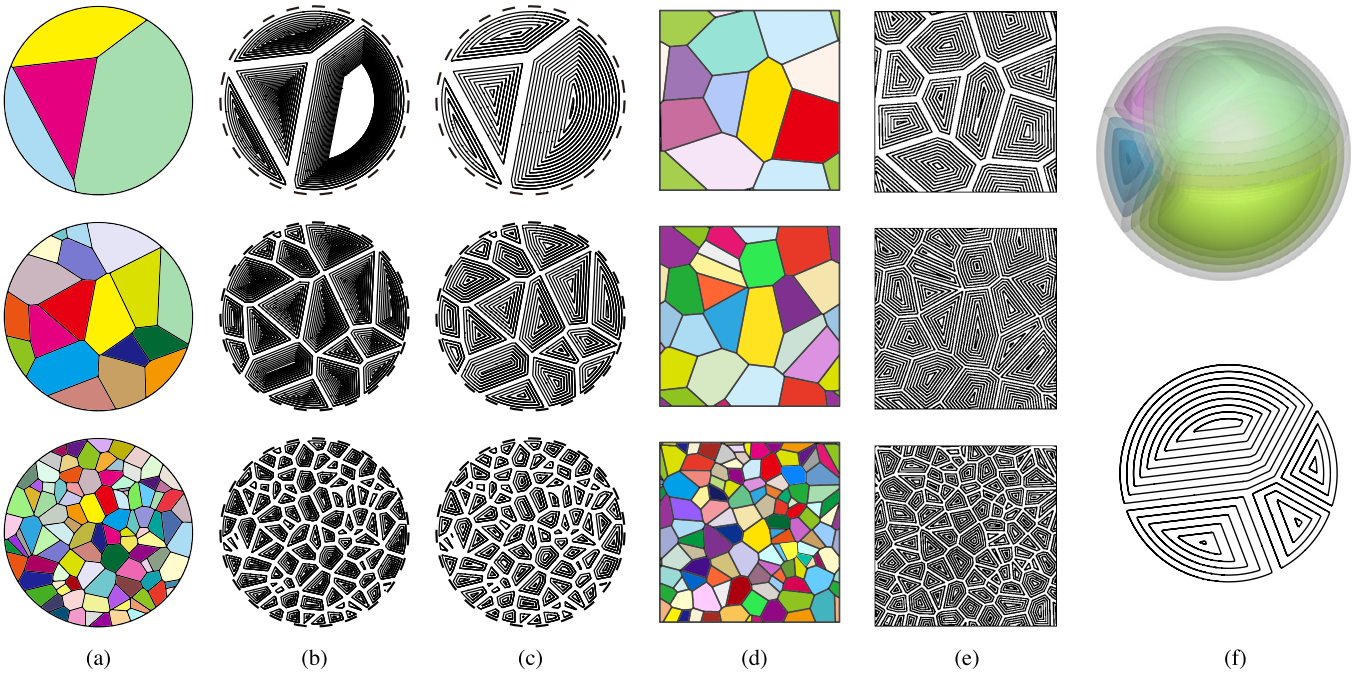


Fig. 4. Reinitialization for randomly generated cases with different $\mathcal{N}$. The value of the χ indicator for the cases with random diagrams inside a circle and periodic random diagrams are represented by different colors in (a) and (d), respectively, where the Voronoi edges and the circle are 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). (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

Table 1
CPU time (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[a]	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

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

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 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

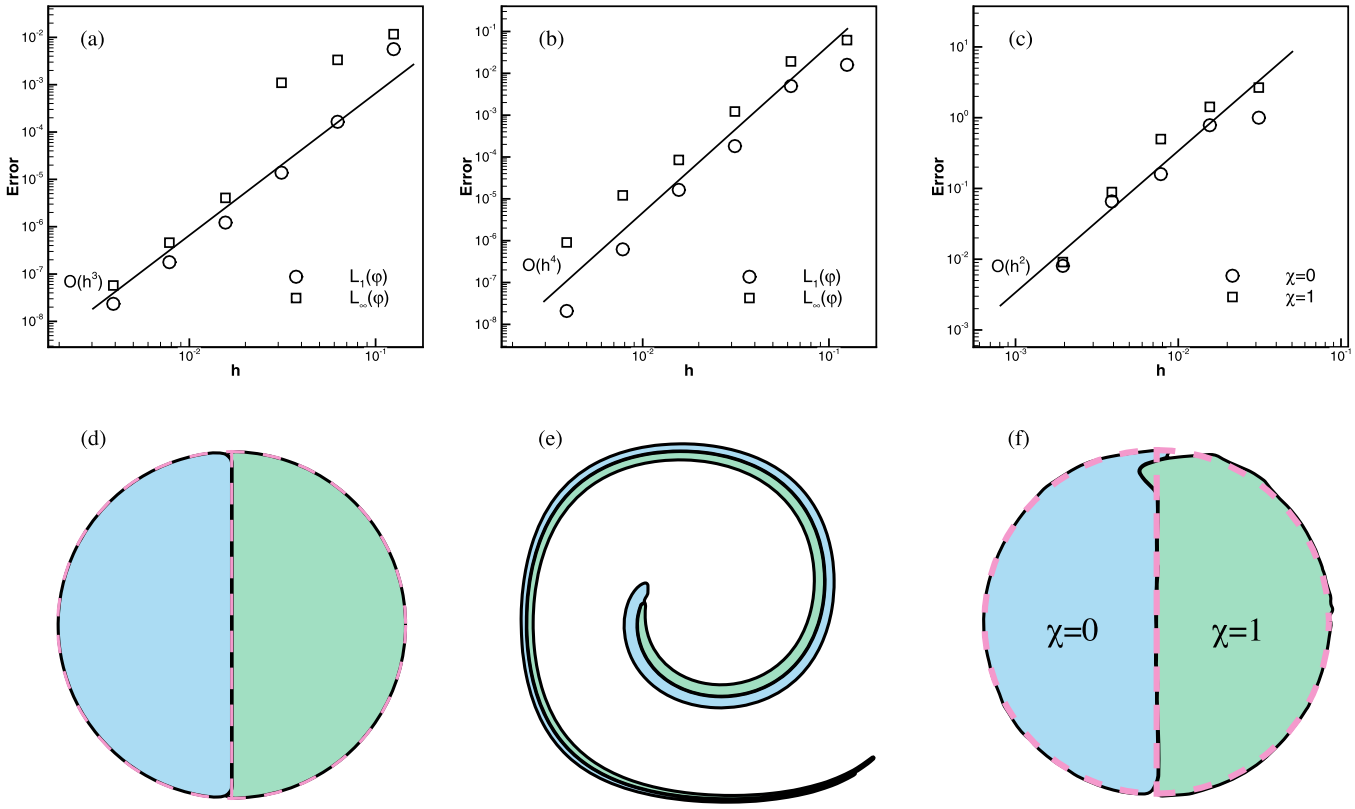


Fig. 5. Applications of the regional reinitialization method for different types of flows. The convergence results of the normal driven flow, triple-point advection and vortex flows are shown in (a), (b) and (c), respectively. The interface of the vortex flows is extracted at $t = 3$ (d) and $t = 6$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

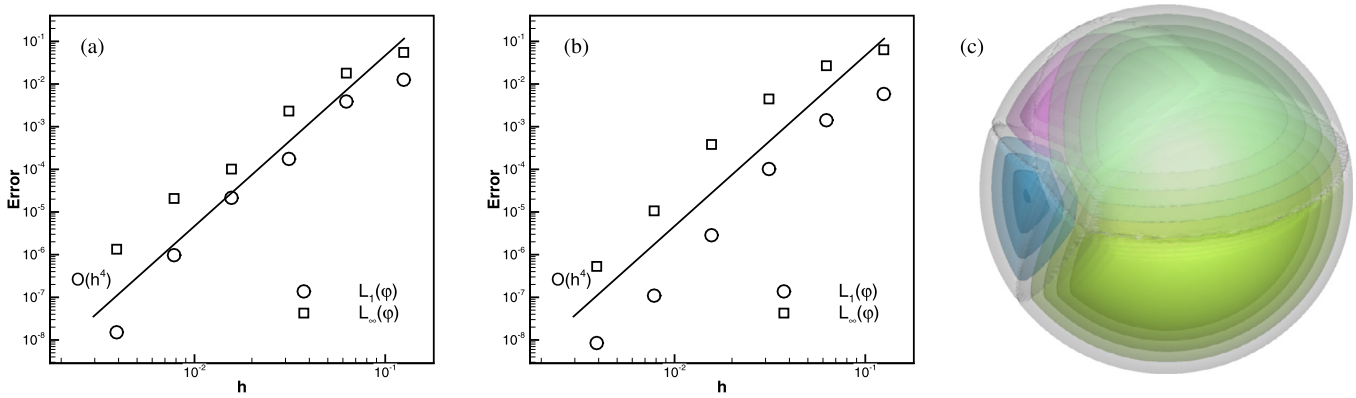


Fig. 6. Three-dimensional advection with the regional reinitialization method. The convergence results of the 3-region and 5-region advection are shown in (a) and (b), respectively. The extracted interfaces, $\varphi = 0$, 0.02, 0.04, 0.06 and 0.08, are shown in (c) for $t = 0.2$.

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 $\mathcal{N}$ cases with a computational effort that effectively is independent of $\mathcal{N}$. 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.

Acknowledgements

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] Akio Doi, Akio Koide, An efficient method of triangulating equi-valued surfaces by using tetrahedral cells, IEICE Trans. Inf. Syst. 74 (1) (1991) 214–224.
[2] John B. Bell, Phillip Colella, Harland M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (2) (1989) 257–283.
[3] Li-Tien Cheng, Yen-Hsi Tsai, Redistancing by flow of time dependent eikonal equation, J. Comput. Phys. 227 (8) (2008) 4002–4017.
[4] David L. Chopp, Some improvements of the fast marching method, SIAM J. Sci. Comput. 23 (1) (2001) 230–244.
[5] Antoine du Chéné, Chohong Min, Frédéric Gibou, Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes, J. Sci. Comput. 35 (2) (2008) 114–131.
[6] Matt Elsey, Selim Esedoglu, Fast and accurate redistancing by directional optimization, SIAM J. Sci. Comput. 36 (1) (2014) A219–A231.
[7] Frederic Gibou, Ronald Fedkiw, Stanley Osher, A review of level-set methods and some recent applications, J. Comput. Phys. 353 (2018) 82–109, https://doi.org/10.1016/j.jcp.2017.10.006.
[8] L.H. Han, X.Y. Hu, N.A. Adams, Adaptive multi-resolution method for compressible multi-phase flows with sharp interface model and pyramid data structure, J. Comput. Phys. 262 (2014) 131–152.
[9] Daniel Hartmann, Matthias Meinke, Wolfgang Schröder, The constrained reinitialization equation for level set methods, J. Comput. Phys. 229 (5) (2010) 1514–1535.
[10] Marcus Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, J. Comput. Phys. 227 (4) (2008) 2674–2706.
[11] Guang-Shan Jiang, Danping Peng, Weighted ENO schemes for Hamilton–Jacobi equations, SIAM J. Sci. Comput. 21 (6) (2000) 2126–2143.
[12] William E. Lorensen, Harvey E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, in: ACM SIGGRAPH Computer Graphics, vol. 21, ACM, 1987, pp. 163–169.
[13] Chohong Min, On reinitializing level set functions, J. Comput. Phys. 229 (8) (2010) 2764–2772.
[14] Stanley Osher, Ronald Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, vol. 153, Springer Science & Business Media, 2006.
[15] Stanley Osher, James A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12–49.
[16] Stanley Osher, Chi-Wang Shu, High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations, SIAM J. Numer. Anal. 28 (4) (1991) 907–922.
[17] Shucheng Pan, Xiangyu Hu, Nikolaus A. Adams, High-resolution transport of regional level sets for evolving complex interface networks, arXiv preprint arXiv:1702.02880, 2017.
[18] Giovanni Russo, Peter Smereka, A remark on computing distance functions, J. Comput. Phys. 163 (1) (2000) 51–67.
[19] R.I. Saye, James A. Sethian, Analysis and applications of the Voronoi implicit interface method, J. Comput. Phys. 231 (18) (2012) 6051–6085.
[20] Robert I. Saye, James A. Sethian, The Voronoi implicit interface method for computing multiphase physics, Proc. Natl. Acad. Sci. USA 108 (49) (2011) 19498–19503.
[21] James A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Natl. Acad. Sci. USA 93 (4) (1996) 1591–1595.
[22] Chi-Wang Shu, Stanley Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (2) (1988) 439–471.
[23] David P. Starinshak, Smadar Karni, Philip L. Roe, A new level set model for multimaterial flows, J. Comput. Phys. 262 (2014) 1–16.
[24] Mark Sussman, Emad Fatemi, An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, SIAM J. Sci. Comput. 20 (4) (1999) 1165–1191.
[25] Mark Sussman, Peter Smereka, Stanley Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1) (1994) 146–159.
[26] Hong-Kai Zhao, Tony Chan, Barry Merriman, Stanley Osher, A variational level set approach to multiphase motion, J. Comput. Phys. 127 (1) (1996) 179–195.
[27] Hongkai Zhao, A fast sweeping method for eikonal equations, Math. Comput. 74 (250) (2005) 603–627.
[28] Wen Zheng, Jun-Hai Yong, Jean-Claude Paul, Simulation of bubbles, Graph. Models 71 (6) (2009) 229–239.

A.3 Paper III

Shucheng Pan, Luhui Han, Xiangyu Hu, Nikolaus A. Adams

A conservative interface-interaction method for compressible multi-material flows

In *Journal of Computational Physics*, In Press, DOI: https://doi.org/10.1016/j.jcp.2018.02.007.

Contribution: My contribution to this work was the development of the method and the corresponding computer code for its implementation. I performed simulations and analyzed the results, and wrote the manuscript for the publication.

Journal of Computational Physics ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A conservative interface-interaction method for compressible multi-material flows

Shucheng Pan, Luhui Han, Xiangyu Hu, Nikolaus A. Adams *

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

ARTICLE INFO

Article history:
Received 10 July 2017
Received in revised form 15 January 2018
Accepted 5 February 2018
Available online xxxx

Keywords:
Compressible multi-material flows
Sharp interface method
Multi-resolution simulations
Level-set method
Interface scale separation

ABSTRACT

In this paper we develop a conservative interface-interaction method dedicated to simulating multiple compressible fluids with sharp interfaces. Numerical models for finite-volume cells cut by more than two material-interface are proposed. First, we simplify the interface interaction inside such a cell to avoid the need for explicit interface reconstruction and very complex flux calculation. Second, conservation is strictly preserved by an efficient conservation correction procedure for the cut cell. To improve robustness, a multi-material scale separation model is developed to remove consistently non-resolved interface scales. In addition, a multi-resolution method and a local time-stepping scheme are incorporated into the proposed multi-material method to speed up high-resolution simulations. Various numerical test cases, including the multi-material shock tube problem, inertial confinement fusion implosion, triple-point shock interaction and shock interaction with multi-material bubbles, show that the method is suitable for a wide range of complex compressible multi-material flows.

1. Introduction

Compressible multi-material problems occur in a broad range of scientific and engineering areas such as high energy physics and astrophysics. Typical examples include inertial confinement fusion (ICF) [1], core-collapse supernova [2] and hypervelocity impact [3]. In these problems, different materials separated by the interface have significantly different material properties and equation of states (EOS). Large density or pressure jumps inside the material or across the interface may occur and may lead to complex flow fields and interface evolution. For such compressible multi-material problems, numerical modeling has received increasing attention in recent years due to its versatility as complement to experimental investigations. Many well-established methods have been proposed to simulate two-phase compressible flows, such as the front-tracking method [4], the arbitrary-Lagrangian–Eulerian (ALE) method [5], the volume-of-fluid (VOF) [6] and the level-set method [7]. Among these, sharp interface methods have the advantage of introducing no numerical mixing artifacts due to the interface model for immiscible materials or shock-driven miscible materials.

Unlike front-tracking and ALE methods, VOF and level-set methods can avoid the problem of capturing complex geometries of interfaces as they implicitly define the interface and solve its evolution. Although VOF inherently is conservative, it treats the interface with a smeared interface representation which is disadvantageous for immiscible materials problems or extremely fast high energy processes. When more than two materials are involved, complex interface reconstruction algorithms, such as onion-skin [8] and serial-dissection [9], are needed to generate interface locations from the volume-faction

* Corresponding author.
E-mail addresses: shucheng.pan@tum.de (S. Pan), luhui.han@tum.de (L. Han), xiangyu.hu@tum.de (X. Hu), nikolaus.adams@tum.de (N.A. Adams).

https://doi.org/10.1016/j.jcp.2018.02.007
0021-9991/

data, usually relying on a material ordering strategy. The mixed treatment for fluid states inside a cell occupied by more than two materials (hereafter referred to as multi-material-cell) in the VOF method does not ensure a sharp-interface representation [10]. In the level-set method, the interface reconstruction is straightforward via a signed distance function as the interface implicitly is represented by the zero contour which can be considered as a non-smeared interface representation. The sharp-interface property can be imposed by interface-interaction treatment [11] for two-phase flows. For multi-material flows involving more than two fluids, problems arise when the interface-network motion needs to be captured and interface interaction occurs inside a multi-material-cell. The former issue can be addressed by a recently developed multi-region level-set method [12] which offers higher computational efficiency than the multiple level-set method [13] and better accuracy than the regional level-set method [14]. The latter issue can be handled by applying the ghost fluid method [15] on the multiple level-set functions to avoid complex interface interaction. This, however, leads to violation of conservation [16,13]. Conservative interface interaction models have been developed for two-phase flows [17,18], but, to our knowledge, have not yet been proposed for more than two fluids.

Numerical methods may suffer from a lack of robustness when complex interface topology changes, such as sudden generation and destruction of thin filaments and isolated droplets, are encountered. Interface scale separation models based on the refined level-set method [19,20] and identifying resolved/non-resolved interface segments [21] have been proposed for two-phase flows to remove non-resolved structures. More recently a model employing the constrained stimulus-response procedure [22,23] has been developed for interface scale separation and increases the robustness for the simulations of compressible interfacial flows. However, scale separation for more than two materials has not been presented in the literature, and the extension of previous models [19,20,22,23] is not straightforward.

The objective of the present paper is to develop an efficient and robust numerical method for compressible multi-material flows. In order to ensure conservation and the sharp-interface property, several operations related to the interface network are proposed. First, the interface-network evolution is captured accurately and efficiently by a multi-region level-set method, which is a combination of the original level-set method and he regional level-set method, to adopt the respective advantages of these two methods. In order to maintain conservation and impose a sharp-interface treatment, we extend the two-phase conservative sharp-interface method by introducing a conservation correction and a reduced interface-interaction model in each multi-material-cell. A multi-material interface scale separation model is proposed to remove non-resolved interface segments and thus to increase the robustness of high-resolution low numerical dissipation simulations. The paper is structured as follows. In Sec. 2, the multi-material sharp-interface method, including the multi-region level-set method, conservative finite volume method, reduced interface-interface model and multi-material interface scale separation operation, is detailed. The proposed method gains computational efficiency by the employed multi-resolution method and a local time-stepping scheme. Accuracy, capability and robustness of the method are demonstrated in Sec. 3 by a range of numerical examples, followed by a concluding remark.

2. Numerical method

2.1. Governing equations

The governing equations of inviscid compressible flows are

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0, \tag{1}$$

where $\mathbf{U} = (\rho, \rho u, \rho v, \rho w, \rho E)^T$, in which ρ, u, v, ρw, and ρE are the density, the three velocity components and the total energy with relation $E = e + \frac{1}{2}(u^2 + v^2 + w^2)$, with e being the internal energy per unit mass. The inviscid flux tensor $\mathbf{F}$ is

$$\mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho u & \rho v & \rho w \\ \rho u^2 + p & \rho v u & \rho w u \\ \rho u v & \rho v^2 + p & \rho w v \\ \rho u w & \rho v w & \rho w^2 + p \\ u(\rho E + p) & v(\rho E + p) & w(\rho E + p) \end{bmatrix}. \tag{2}$$

To close the governing equations EOS is required to describe the thermodynamic properties of the materials. The EOS for an idea gas is

$$p = (\gamma - 1)\rho, \tag{3}$$

where γ is the ratio of specific heats. For water-like fluids, the pressure can be determined by Tait's equation of state

$$p = B\left(\frac{\rho}{\rho_0}\right)^{\gamma} - B + p_0, \tag{4}$$

where ρ_0 and p_0 are the reference density and pressure, and B is a constant. Other EOS may also be employed in compressible multi-material flows, such as the stiffened gas EOS for water under very high pressure, the Jones–Wilkins–Lee (JWL) EOS for gaseous detonation-products and the Mie–Grüneisen (MG) EOS for solids [24].

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

Multi-material fluid dynamic problems contain multiple different fluids. We assume that there are $\mathcal{N}$ materials (or fluids) in the problem and partition the entire domain, $\Omega = \bigcup_{\chi\in \mathrm{X}} \Omega^{\chi}$, where Ω^{χ} is the material domain and $\mathrm{X} = \{\chi \in \mathbb{N} | 1 \le \chi \le \mathcal{N}\}$ is the index set for all materials. Given $\xi, \eta \in \mathrm{X}$ and $\xi \neq \eta$, we define $\partial\Omega^{\chi}$ as the material boundary, $\Gamma_{\xi\eta} = \partial\Omega^{\xi} \bigcap \partial\Omega^{\eta}$ as the pairwise material interface that separates two material domains, $\Gamma = \bigcup \Gamma_{\xi\eta}$ as the interface network, and $\mathrm{J} = \bigcap \Gamma_{\xi\eta}$ as multiple junctions. Each material has its individual material parameters (e.g. viscosity) and EOS.

2.2. Interface capturing

To capture the evolution of complex interface networks, we have developed the multi-region level-set method which is a combination of the original level-set method [7] and the regional level-set method [14]. Globally we use a single regional level-set for representing the multi-region (or multi-material) system to significantly reduce the memory cost. Locally we construct multiple local level-set functions to capture directly the evolution of the interface network in order to save computational effort and avoid inaccuracies in reconstructing the interface network. Although the total number of materials may be very large, locally the number is limited, such that solving multiple locally constructed level-set advection equations is efficient. This method has demonstrated to achieve high-order accuracy for typical test cases without generation of artifacts [12].

The total system can be represented by the regional level-set function $\varphi^{\chi}(\mathbf{x}) = (\varphi(\mathbf{x}), \chi(\mathbf{x}))$, where $\varphi(\mathbf{x}) \ge 0$ is the unsigned distance function and $\chi(\mathbf{x})$ is an integer material indicator. The material domain Ω^{a} is identified by the indicator, $\Omega^{\xi} = \{\mathbf{x} | \chi(\mathbf{x}) = \xi\}$, and the interface network is defined as $\Gamma = \{\mathbf{x} | \varphi(\mathbf{x}) = 0\}$. On a two-dimensional uniform Cartesian grid, the regional level-set $\varphi^{\chi}_{i,j} = (\varphi_{i,j}, \chi_{i,j})$ is defined at the center of the finite-volume cell $C_{i,j}$. The $\mathcal{N}_s$ materials contained in a local set of cells

$$V_s = \{C_{k,l} | i-1 < k < i+1, j-1 < l < j+1\} \tag{5}$$

are identified by a local index set $\mathrm{X}_s = \{r \in \mathbb{N} | 1 \le r \le \mathcal{N}_s\}$.

By using a construction operator for generating the local multiple level-set fields and a reconstruction operator for reconstructing the global regional level-set field from the local level-set fields, the evolution step contains three main operations for $C_{i,j}$:

(1) Construct $\mathcal{N}_s$ local level-set fields $\phi^{r,n}_{k,l}$ for the current time-step n at the center of each cell $C_{k,l}$ which belongs to the spatial discretization stencil of $C_{i,j}$

$$\phi^{r,n}_{k,l} = \mathbf{C}_r\left(\varphi^{\chi}_{k,l}\right) = \begin{cases} \varphi_{k,l} & \text{if } \chi_{k,l} = \chi_r \\ -\varphi_{k,l} & \text{otherwise,} \end{cases} \quad r \in \mathrm{X}_s \tag{6}$$

(2) Compute the updated $\phi^{r,n+1}_{i,j}$ at the next time-step $n+1$ by solving $\mathcal{N}_s$ local advection equations. The formulation of the local advection equation depends on $\mathcal{N}_s$. If $\mathcal{N}_s \le 2$ it recovers the original level-set advection equation while if $\mathcal{N}_s \ge 3$ it can be rewritten as

$$\begin{gathered} \phi^{r,(s+1)}_{i,j} = \beta_p \phi^{r,n}_{i,j} + (1-\beta_p)\left[\phi^{r,(s)}_{i,j} - \Delta t \mathbf{v}^{n}_{i,j} \cdot \left(\nabla\phi^{r}\right)^{(p)}_{i,j}\right], \quad 0 \le p \le m, r \in \mathrm{X}_s, \\ \phi^{r,(0)}_{i,j} = \phi^{r,n}_{i,j}, \quad \phi^{r,(m+1)}_{i,j} = \phi^{r,n+1}_{i,j}, \end{gathered} \tag{7}$$

where m is the number of Runge–Kutta sub-steps, β_p is the parameter in the p-th sub-step, $\mathbf{v}$ is the advection velocity, and $\left(\nabla\phi^{r}\right)^{(p)}_{i,j}$ is the finite difference approximation of the spatial derivative at the center of a finite-volume cell $C_{i,j}$.

(3) Reconstruct the new regional level-set $\varphi^{\chi}_{i,j}$ at the center of $C_{i,j}$ from the $\mathcal{N}_s$ new local level-set fields $\phi^{r,n+1}_{i,j}$ by the reconstruction operator $\mathbf{R}$ as

$$\varphi^{\chi}_{i,j} = \mathbf{R}\left(\phi^{r,n+1}_{i,j}, r \in \mathrm{X}_s\right) = \left(\left|\max \phi^{r,n+1}_{i,j}\right|, \arg\max_{\chi_r} \phi^{r,n+1}_{i,j}\right). \tag{8}$$

We consider a 1D two-region case in Fig. 1 to illustrate the interface capturing scheme above. Suppose we have the regional level-set function $\varphi^{\chi,n}$ (solid line in Fig. 1(a)) at time step n, and we want to obtain $\varphi^{\chi,n+1}$ (dashed line in Fig. 1(a)). The velocity field u is set as constant for simplicity. Following the algorithm above, first we use the construction operator to construct the local level-set functions. For instance, as the local index set of the cut-cell $i=6$ is $\mathrm{X}_s = \{1,2\}$, the two local level-set functions defined on the stencil of cell $i=6$, $\phi^{1,n}$, $\phi^{2,n}$, are shown in Fig. 1(b). We use Eq. (7) to advect these local level-sets,

$$\phi^{1,n+1}_{i=6} = \phi^{1,n}_{i=6} + \Delta t u \left(\frac{\partial\phi^{1}}{\partial x}\right)^{n}_{i=6}, \quad \phi^{2,n+1}_{i=6} = \phi^{2,n}_{i=6} + \Delta t u \left(\frac{\partial\phi^{2}}{\partial x}\right)^{n}_{i=6}, \tag{9}$$

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

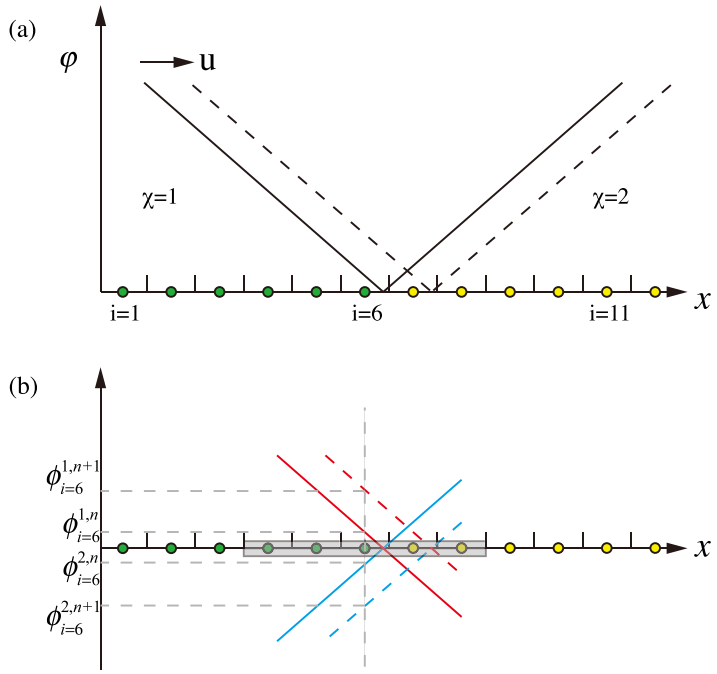


Fig. 1. Schematic of the interface capturing method in Sec. 2.2 for a 1D two-region system. (a) The regional level-set function $\varphi^{\chi,n}$ at t^n is illustrated by the solid line. The dashed line indicates the exact $\varphi^{\chi,n+1}$ profile at t^{n+1} driven by a constant velocity u. (b) The local level-set functions $\phi^{1,n}$ and $\phi^{2,n}$ on the stencil (the gray box) of cell $i=6$ are plotted by the blue solid line and the red solid line for the regions $\chi=1$ and $\chi=2$, respectively. The corresponding evolved local level-set functions, $\phi^{1,n+1}$ and $\phi^{2,n+1}$, are shown by the dashed lines. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

where for simplicity an explicit Euler time marching is used for illustration. The numerical approximation of $\left(\frac{\partial\phi^1}{\partial x}\right)^n$ depends on the specific spatial discretization scheme. The updated values for cell $i=6$, $\phi^{1,n+1}_{i=6}$ and $\phi^{2,n+1}_{i=6}$, are plotted in Fig. 1(b). Applying the reconstruction operator gives

$$\varphi^{\chi}_{i=6} = \mathbf{R}\left(\phi^{1,n+1}_{i=6}, \phi^{2,n+1}_{i=6}\right) = \left(\left|\phi^{1,n+1}_{i=6}\right|, 1\right), \tag{10}$$

which indicates that cell $i=6$ does not change its material indicator. For cell $i=7$ one has

$$\varphi^{\chi}_{i=7} = \left(\left|\phi^{1,n+1}_{i=7}\right|, 1\right) \tag{11}$$

which corresponds to an indicator change from $\chi^n_{i=7}=2$ to $\chi^{n+1}_{i=7}=1$. For more complex 2D examples involving multiple junctions we refer to Ref. [12].

2.3. Conservative sharp-interface method

We extend the two-phase discretized governing equation of the conservative sharp-interface method [18] to multiple materials. On a two-dimensional uniform Cartesian grid with grid spacings Δx and Δy, the flow variable $\mathbf{U}$ is defined at the center of each finite-volume cell. For each material χ residing in cell $C_{i,j}$, we can integrate Eq. (2) over the space-time volume $C_{i,j}\cap\Omega^{\chi}(t)$ and apply the Gauss theorem to obtain

$$\int_n^{n+1} \mathrm{d}t \int_{\alpha^{\chi}_{i,j}(t)} \mathrm{d}x\mathrm{d}y\, \frac{\partial \mathbf{U}}{\partial t} + \int_n^{n+1} \mathrm{d}t \int_{\partial C_{i,j}\cap\Omega^{\chi}(t)} \mathrm{d}x\mathrm{d}y\, \mathbf{F}\cdot\mathbf{n} = 0, \tag{12}$$

where $\alpha^{\chi}_{i,j}(t)$ is the time dependent volume fraction of material χ in $C_{i,j}$. $\partial C_{i,j}\cap\Omega^{\chi}(t)$ contains two parts: one is the combination of the four segments of the cell faces after being cut by the material interface, which can be written in the form of $A^{\chi}_{i+1/2,j}(t)\Delta y$, $A^{\chi}_{i,j+1/2}(t)\Delta x$, $A^{\chi}_{i-1/2,j}(t)\Delta y$, and $A^{\chi}_{i,j-1/2}(t)\Delta x$, where $A^{\chi}(t)$ is the aperture, see Fig. 3(a); the other one, denoted as $\Delta\Omega^{\chi}_{i,j}$, is the segment of material boundary $\partial\Omega^{\chi}$ inside cell $C_{i,j}$. This integral equation can be discretized with an explicit Euler time marching scheme as

$$\begin{aligned}\alpha^{\chi,n+1}_{i,j}\mathbf{U}^{n+1}_{i,j} = \alpha^{\chi,n}_{i,j}\mathbf{U}^{n}_{i,j} &+ \frac{\Delta t}{\Delta x}\left[A^{\chi}_{i-1/2,j}\hat{\mathbf{F}}_{i-1/2,j} - A^{\chi}_{i+1/2,j}\hat{\mathbf{F}}_{i+1/2,j}\right] \\ &+ \frac{\Delta t}{\Delta x}\left[A^{\chi}_{i,j-1/2}\hat{\mathbf{F}}_{i,j-1/2} - A^{\chi}_{i,j+1/2}\hat{\mathbf{F}}_{i,j+1/2}\right] + \frac{\Delta t}{\Delta x\Delta y}\hat{\mathbf{X}}(\Delta\Omega^{\chi}_{i,j}),\end{aligned} \tag{13}$$

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

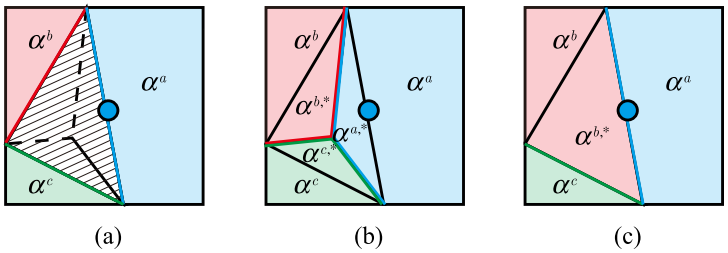


Fig. 2. A schematic representation of volume fraction correction for a multi-material finite-volume cell to maintain conservation: (a) without modification, (b) sub-cell reconstruction and (c) conservation modification.

where Δt is the time step size determined by CFL condition. $\alpha^{\chi}_{i,j}\mathbf{U}_{i,j}$ is the conservative quantity vector in $C_{i,j}$, with $\mathbf{U}_{i,j}$ being the cell-averaged quantity vector of the material χ. $\hat{\mathbf{F}}$ is the numerical flux at cell-face and $\hat{\mathbf{X}}(\Delta\Omega^{\chi}_{i,j})$ is the momentum and energy exchange flux determined by the interface interaction model discussed in Sec. 2.3.3. For Runge–Kutta time discretization, Eq. (13) is synonymous for a substep.

To update the volume fraction from the regional level-set function we use the algorithm in Ref. [25]. For computing the volume fraction α_r of a particular material r in a cut-cell with $\mathcal{N}_s$ materials the algorithm, as in Ref. [26], recursively subdivides a cut-cell into 4 (8 in 3D) sub-cells with equal size. The volume fraction of each sub-cell α_s is determined by the sign of its local level-set data ϕ^r at vertices (obtained by bilinear/trilinear interpolation), i.e., $\alpha_s = 1$ for full sub-cells (all vertex level-set values are positive), $\alpha_s = 0$ for empty sub-cells (all vertex level-set values are negative), and $\alpha_s = 0.5$ for cut sub-cells. We set the maximal subdivision level to be 5 for 2D cases. For 3D cases this algorithm is computationally expensive, and we prefer to use the volume-estimation method of Ref. [23] (see Appendix A therein) or a recently derived analytical formulation [27].

2.3.1. Conservation correction for a multi-material-cell

For the material χ summing Eq. (13) over its material domain Ω^{χ} yields

$$\sum_{i,j}\alpha^{\chi,n+1}_{i,j}\mathbf{U}^{n+1}_{i,j} = \sum_{i,j}\alpha^{\chi,n}_{i,j}\mathbf{U}^{n}_{i,j} + \sum_{i,j}\frac{\Delta t}{\Delta x\Delta y}\hat{\mathbf{X}}(\Delta\Omega^{\chi}_{i,j}) + \text{boundary terms} \tag{14}$$

For two-phase flows, overall conservation can be achieved by summing Eq. (14) for the two fluids because interface-exchange terms in cut cell always have opposite sign for each of the interacting materials, and the sum of volume fractions of the two fluids equals 1. For multi-material flows, although the volume fractions in a two-material-cell satisfy $\alpha^{\chi_1} + \alpha^{\chi_2} = 1$, in a multi-material-cell where more than two materials meet the sum of volume fractions may be $\sum_{r\in\mathrm{X}_s}\alpha^{\chi_r} \neq 1.0$ if we assume that the material boundary segments are piecewise linear inside the cell. For example in Fig. 2(a), the linear material boundary segments of three materials never coincide, resulting in a void region, and thus $\alpha^a + \alpha^b + \alpha^c < 1.0$. To correct this total volume fraction deficiency, the piecewise linear assumption no longer can be applied to this cell. A straightforward way is to explicitly reconstruct the sub-cell topology inside this multi-material-cell as shown in Fig. 2(b). The new volume fraction for each material is updated by

$$\alpha^a = \alpha^a + \alpha^{a,*},\ \alpha^b = \alpha^b + \alpha^{b,*},\ \alpha^c = \alpha^c + \alpha^{c,*}. \tag{15}$$

Although this procedure is considered to be accurate as long as the explicit reconstruction method is sufficiently accurate, it is computationally expensive, especially in three dimensions. An alternative efficient way is shown in Fig. 2(c). Instead of explicitly reconstructing the sub-cell structure, we only modify the volume fraction corresponding to the smallest mass fraction inside this cell:

$$\alpha^{\chi^*} = \alpha^{\chi^*} + \alpha^{\chi^*,*},\quad \alpha^{\chi^*,*} = 1 - \sum\alpha^{\chi},\quad \chi^* = \arg\min_{\chi}(\alpha^{\chi}\rho_{\chi}). \tag{16}$$

Thus this method is efficient especially in 3D as no reconstruction is needed. Although it does not resolve the sub-cell structure and is generally less accurate than the first method, the error is limited as the material dominating the cell properties labeled by the primary indicator) has a volume fraction larger than 0.5 which is not affected by the correction.

2.3.2. Material interface interaction model

The interface interaction for each connected material pair $(\Omega^{\xi}, \Omega^{\eta})$ occurs at the pairwise interface $\Gamma_{\xi\eta}$ and can be described by a Riemann problem. The solution of this Riemann problem produces the interface condition which is used to calculate the exchange flux $\hat{\mathbf{X}}$ in Eq. (13) and the interface advection velocity $\mathbf{v}$ in Eq. (7). Hu et al. [24] have shown that the HLLC Riemann solver [28] is robust, accurate and efficient in handling two-phase flow with very strong interactions and large jumps of material properties. In this paper, we employ this approach to solve multi-material interface interactions.

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

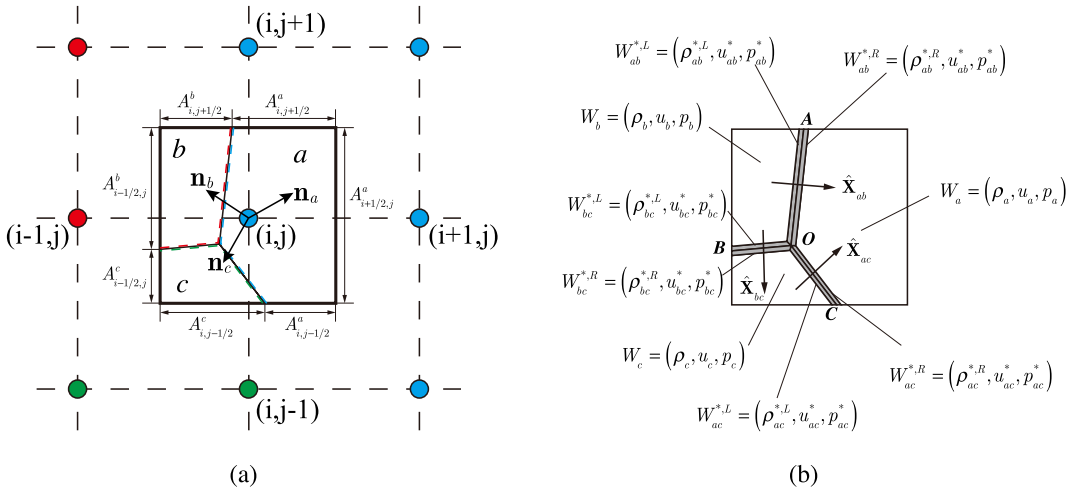


Fig. 3. Schematic of conservative discretization for a 3-material cell. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Consider that the finite-volume cell $C_{i,j}$ in Fig. 3(a) is occupied by three materials a, b and c, which are colored by blue, red and green, respectively. With our method, the multi-material cell $C_{i,j}$ contains one regional level-set $\varphi^{\chi}_{i,j}$ and multiple fluid states. For the cell in Fig. 3(a), the flow states of material a are obtained by solving Eq. (13), and the states of the other material are extended from the respective material domains by using the extending algorithm [15,18]. According to Fig. 3(b), the interface condition at each pairwise interface segment $\Gamma_{\xi\eta}$ is obtained by solving a corresponding 1D Riemann problem $\mathsf{R}(W_L, W_R)$ along a local normal direction $\mathbf{n}_{\xi\eta}$ of $\Gamma_{\xi\eta}$. The constant left and right states of the Riemann problem, (W_L, W_R), are defined as

$$(W_L, W_R) = \begin{cases} (\hat{W}_\xi, \hat{W}_\eta) & \text{if } \alpha^\xi \rho_\xi < \alpha^\eta \rho_\eta \\ (\hat{W}_\eta, \hat{W}_\xi) & \text{otherwise,} \end{cases} \qquad \hat{W}_\xi = \begin{cases} W_\xi & \text{if } \xi = \chi_1 \\ W^g_\xi & \text{otherwise,} \end{cases} \tag{17}$$

where the superscript "g" indicates the ghost state at the cell center and χ_1 is the primary material indicator with $\alpha^{\chi_1} > 0.5$. We rewrite the EOS with a general form

$$p = p(\rho, e), \tag{18}$$

and the speed of sound

$$c_s = \left.\frac{\partial p}{\partial \rho}\right|_e + \frac{p}{\rho^2}\left.\frac{\partial p}{\partial e}\right|_\rho = \Psi + \Upsilon\frac{p}{\rho}, \tag{19}$$

where Υ is the Grüneisen coefficient and Ψ determines the material properties [24]. Afterwards, the HLLC approximate Riemann solver is invoked to obtain the two intermediate states, $W^{*,L}_{\xi\eta}$ and $W^{*,R}_{\xi\eta}$, see Fig. 3(b), which have the relation

$$u^{*,L}_{\xi\eta} = u^{*,R}_{\xi\eta} = S_M = u^*, \quad p^{*,L}_{\xi\eta} = p^{*,R}_{\xi\eta} = p^*, \tag{20}$$

where S_M is the speed of the contact wave or the material interface. Toro et al. [28] use the jump conditions and the integral form of the conservation law to obtain the normal contact wave velocity

$$u^* = S_M = \frac{(S_R - u_R)\rho_R u_R + (u_L - S_L)\rho_L u_L + p_L - p_R}{(S_R - u_R)\rho_R + (u_L - S_L)\rho_L} \tag{21}$$

and the intermediate pressure

$$p^* = p_L + \rho_L(S_L - u_L)(S_M - u_L) = p_R + \rho_R(S_R - u_R)(S_M - u_R) \tag{22}$$

which describe the interface condition. The minimum and maximum wave speed S_L and S_R are estimated by

$$S_L = \min[u_L - c_{s,L}, \tilde{u} - \tilde{c_s}], \quad S_R = \max[\tilde{u} + \tilde{c_s}, u_R + c_{s,R}]. \tag{23}$$

For two adjacent states described by different EOS, the average speed of sound $\tilde{c_s}$ is obtained by

$$\tilde{c_s}^2 = \tilde{\Psi} + \tilde{\Upsilon}\widetilde{\left(\frac{p}{\rho}\right)}, \tag{24}$$

where the tilde on the right side indicates Roe-averaged values

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

$$\tilde{\rho} = \sqrt{\rho_L \rho_R}, \quad \tilde{f} = \mu(f) = \frac{\sqrt{\rho_L} f_L + \sqrt{\rho_R} f_R}{\sqrt{\rho_L} + \sqrt{\rho_R}}, f = \Psi, \Upsilon \tag{25}$$

and

$$\widetilde{\left(\frac{p}{\rho}\right)} = \mu\left(\frac{p}{\rho}\right) + \frac{1}{2}\tilde{\rho}\left(\frac{u_R - u_L}{\sqrt{\rho_L} + \sqrt{\rho_R}}\right)^2. \tag{26}$$

We can calculate the interface flux between materials ξ and η according to the interface condition $(\mathbf{u}^*_{\xi\eta}, p^*_{\xi\eta})$ obtained above, where $\mathbf{u}^*_{\xi\eta} = u^*_{\xi\eta}\mathbf{n}_{\xi\eta}$. For a two-material-cell, the flux $\hat{\mathbf{X}}$ in Eq. (13) has the same form as that in Ref. [11]. The multi-material-cell, e.g. the three-material-cell in Fig. 3, has the more complex interface flux

$$\begin{aligned}
\hat{\mathbf{X}}(\Delta\Omega^a_{i,j}) &= \hat{\mathbf{X}}(\Delta\Gamma_{ab}) + \hat{\mathbf{X}}(\Delta\Gamma_{ac}) = \hat{\mathbf{X}}(\Delta\Gamma_{\overline{OA}}) + \hat{\mathbf{X}}(\Delta\Gamma_{\overline{OC}}) \\
\hat{\mathbf{X}}(\Delta\Omega^b_{i,j}) &= -\hat{\mathbf{X}}(\Delta\Gamma_{ab}) - \hat{\mathbf{X}}(\Delta\Gamma_{bc}) = -\hat{\mathbf{X}}(\Delta\Gamma_{\overline{OA}}) - \hat{\mathbf{X}}(\Delta\Gamma_{\overline{OB}}) \quad , \\
\hat{\mathbf{X}}(\Delta\Omega^c_{i,j}) &= \hat{\mathbf{X}}(\Delta\Gamma_{bc}) - \hat{\mathbf{X}}(\Delta\Gamma_{ac}) = \hat{\mathbf{X}}(\Delta\Gamma_{\overline{OB}}) - \hat{\mathbf{X}}(\Delta\Gamma_{\overline{OC}})
\end{aligned} \tag{27}$$

where the interface inside the cell, $\Delta\Gamma_{\overline{OC}}$, are the line segments shown in Fig. 3(b). The interface condition is used to calculate these fluxes, e.g.

$$\hat{\mathbf{X}}(\Delta\Gamma_{\overline{OA}}) = \left[0,\ p^*_{ab}\Delta\Gamma_{\overline{OA}}\mathbf{n}^x_{ab},\ p^*_{ab}\Delta\Gamma_{\overline{OA}}\mathbf{n}^y_{ab},\ p^*_{ab}\Delta\Gamma_{\overline{OA}}\mathbf{n}_{ab}\cdot\mathbf{u}^*_{ab}\right]^T. \tag{28}$$

This model represents the full set of interactions. However, it is computationally expensive due to the fact that an explicit reconstruction of the interface-network is necessary to calculate the length and normal direction of $\Delta\Gamma_{\overline{OA}}$, $\Delta\Gamma_{\overline{OB}}$ and $\Delta\Gamma_{\overline{OC}}$. Moreover, it is difficult to extend to 3D since the number of interface segments for each material may become every large. As alternative, we propose a reduced interaction model for a multi-material-cell. The basic idea is that one only considers the interaction between the two materials with largest mass fraction inside a multi-material-cell. Assuming the mass fraction of each material is ordered as $\alpha^a\rho_a > \alpha^b\rho_b > \alpha^c\rho_c$ in Fig. 3, the reduced model is

$$\begin{aligned}
\hat{\mathbf{X}}(\Delta\Omega^a_{i,j}) &= \hat{\mathbf{X}}(\Delta\Gamma_{ab}) \approx \hat{\mathbf{X}}(\Delta\Omega^a_{\overline{AC}}) \\
\hat{\mathbf{X}}(\Delta\Omega^b_{i,j}) &= -\hat{\mathbf{X}}(\Delta\Gamma_{ab}) \approx -\hat{\mathbf{X}}(\Delta\Omega^a_{\overline{AC}}),
\end{aligned} \tag{29}$$

where $\Delta\Omega^a_{\overline{AC}}$ is the material boundary of material a represented by its local level set. The normal direction $\mathbf{n}_{\xi\eta}$ in the full interaction model can be approximated by the normal of a particular material boundary. As shown in Fig. 2(a), the three normal directions of material a, b and c are defined at the cell center and can be calculated by their corresponding local level-set fields

$$\mathbf{n}_{\chi_r} = \frac{\nabla\phi^r}{|\nabla\phi^r|}, \quad \chi_r = a, b, c. \tag{30}$$

Note that the normal direction points into the respective material. The direction of the local Riemann problem $\mathsf{R}(W_L, W_R)$ is

$$\mathbf{n}_{\xi\eta} = \begin{cases} \mathbf{n}_\xi & \text{if } \alpha^\xi\rho_\xi \geq \alpha^\eta\rho_\eta \\ \mathbf{n}_\eta & \text{otherwise.} \end{cases} \tag{31}$$

Additionally, the interface flux is modified by

$$\hat{\mathbf{X}}(\Delta\Omega^a_{\overline{AC}}) = \left[0,\ p^*_{ab}\Delta\Omega^a_{\overline{AC}}\mathbf{n}^x_a,\ p^*_{ab}\Delta\Omega^a_{\overline{AC}}\mathbf{n}^y_a,\ p^*_{ab}\Delta\Omega^a_{\overline{AC}}\mathbf{n}_a\cdot\mathbf{u}^*_{ab}\right]^T, \tag{32}$$

where $\mathbf{n}_a$ replaces $\mathbf{n}_{ac}$ as $\alpha^a\rho_a > \alpha^c\rho_c$. The averaged flow variable of the two subvolumes separated by $\Delta\Omega^a_{\overline{OA}}$ are W_a and W_b, respectively, indicating that the flow state of material c is approximated by the data of material b. By this procedure we reduce the multiple interactions to a single interaction, irrespectively of the number of materials inside a cell. Moreover an explicit interface reconstruction no longer is needed to calculate the normal directions. Thus the reduced model significantly improves computational efficiency, especially in 3D.

The rationale of the reduced-model formulation is based on the assumption that the material with largest mass fraction $\alpha_a\rho_a$ in a multi-material cell dominates inertial dynamics. Under this assumption it suffices to determine the interaction between this material a with the mixture of b and c, without resolving the subcell details of the multi-material interactions. The flow state $\mathbf{U}_m$ of the mixture (denoted by "m") can be determined following e.g. [29] as

$$\rho_m = \sum_{r=b,c}\alpha_r\rho_r, \quad (\rho\mathbf{u})_m = \sum_{r=b,c}\alpha_r\rho_r\mathbf{u}_r, \quad (\rho E)_m = \sum_{r=b,c}\alpha_r\rho_r(e_r + \frac{1}{2}|\mathbf{u}_r|^2), \tag{33}$$

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

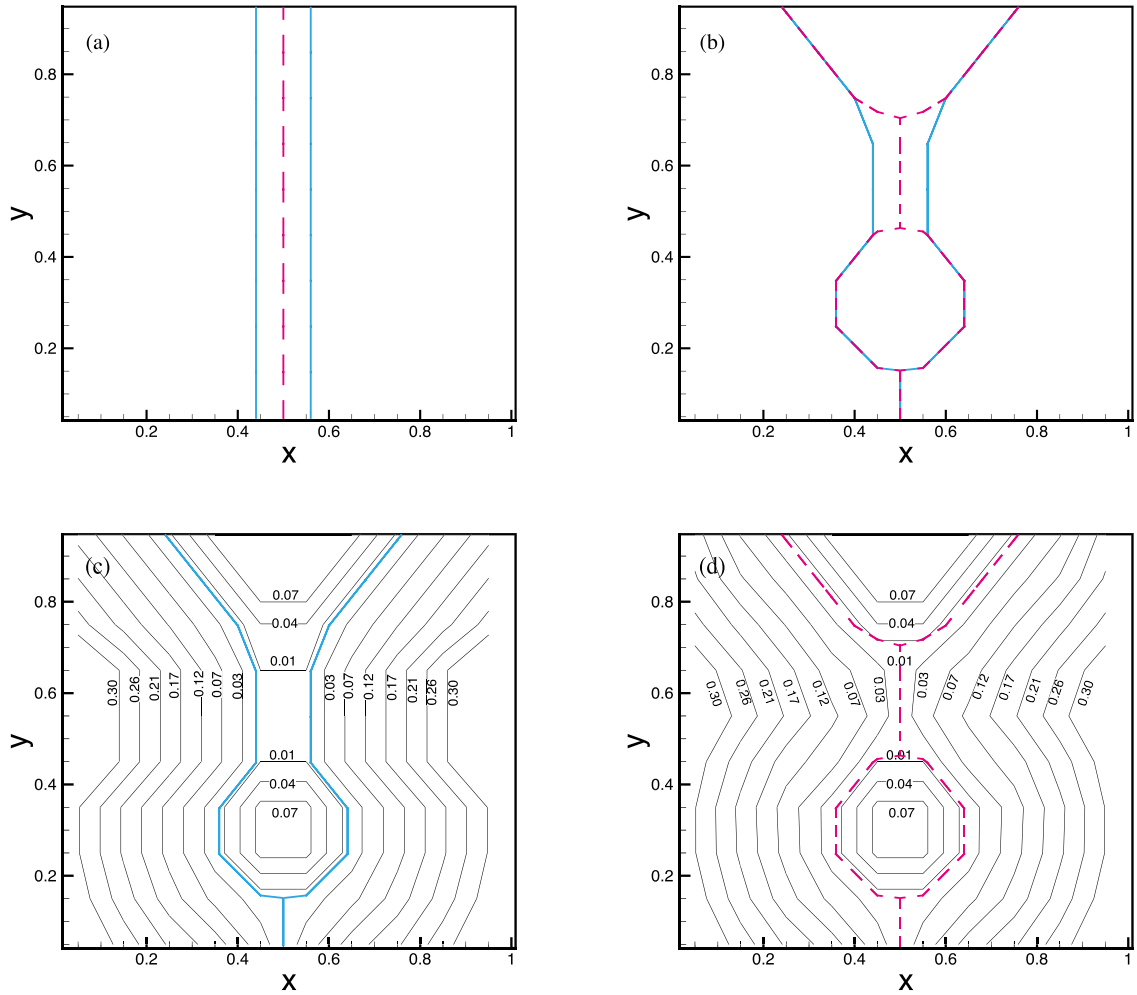


Fig. 4. Simple test cases for multi-material interface scale separation model: (a) A thin filament and (b) a small droplet. The solid blue line and dashed red line indicate the interfaces before and after applying the scale separation model. (c) and (d) show the level-set contours before and after applying the scale separation model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where the volume fractions have been normalized by the total volume of the mixture $\sum_{r=b,c}\alpha_r$. If the mass fraction of c is sufficiently small,

$$\alpha_c\rho_c \ll \alpha_b\rho_b, \tag{34}$$

and the density of the materials constituting the mixture satisfies

$$\rho_c \simeq \rho_b, \tag{35}$$

the state of the mixture can be approximated by

$$\rho_m \simeq \rho_b, \quad (\rho\mathbf{u})_m \simeq \rho_b\mathbf{u}_b, \quad (\rho E)_m \simeq \rho_b(e_b + \frac{1}{2}|\mathbf{u}_b|^2)\,, \tag{36}$$

which essentially implies that the interaction between material a and the mixture m can be replaced by the interaction of the reduced model (29). Note that for highly compressible flow the assumptions (34) and (35) may be not valid. Nevertheless even for complex test cases shown in Sec. 3 the simplified model produces good results with high computational efficiency. We emphasize that the current model formulation is limited to fluid-like materials. For more complex material combinations, such as fluid–solid interactions, e.g. Ref. [30], the model requires further extensions as the materials cannot be described by the same EOS family.

Global conservation is also achieved as can be seen from summing over all materials

$$\sum_{\chi\in\mathrm{X}}\sum_{i,j}\alpha_{i,j}^{\chi,n+1}\mathbf{U}_{i,j}^{n+1} = \sum_{\chi\in\mathrm{X}}\sum_{i,j}\alpha_{i,j}^{\chi,n}\mathbf{U}_{i,j}^{n} + \text{boundary terms}. \tag{37}$$

The interface fluxes for the two interacting materials have opposite sign in all multi-material-cells, i.e. $\hat{\mathbf{X}}(\Delta\Omega_{i,j}^{\xi}) + \hat{\mathbf{X}}(\Delta\Omega_{i,j}^{\eta}) = 0$.

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

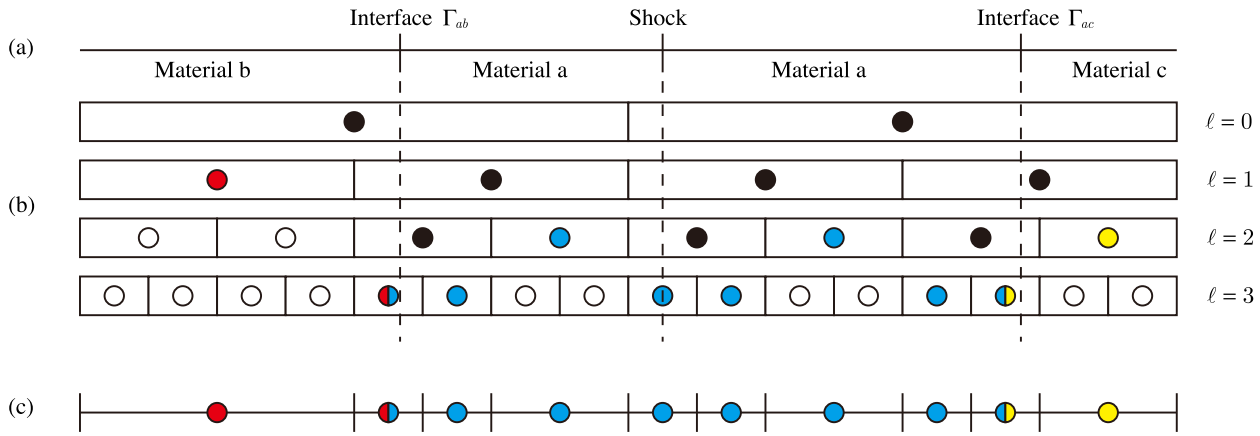


Fig. 5. The multi-resolution representation of 3-material compressible flows in 1D. (a) The computational domain is partitioned by 3 materials (a, b, c). Two interfaces Γ_{ab} and Γ_{ac} separate these materials, and a shock wave occurs inside material a. (b) The circles indicate the non-existing blocks. The black dots are the non-leaf blocks which have child blocks. The leaf blocks, colored by blue, red and yellow are blocks containing single material a, b and c, respectively. (c) The final multi-resolution representation shows the single-material block of material b is located at the coarsest level as no shock wave or interface occur near it. The single-material block of material c is refined to $\ell = 2$ because it is close to the interface Γ_{ac}. The single-material blocks of material a are either refined by interface or shock wave. All multi-material blocks are refined to the finest level. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Initial conditions for 1D test cases I and III.

Location	Case I[a]			Case III		
	$0 \le x < 0.4$	$0.4 \le x < 0.6$	$0.6 \le x \le 1.0$	$0 \le r < 1.0$	$1.0 \le r < 1.2$	$1.2 \le r \le 1.5$
χ	1	2	3	1	2	3
ρ	0.125	1.0	0.125	0.05	1.0	0.1
p	0.1	1.0	0.1	0.1	0.1	13.0
γ	1.667	1.4	1.667	1.667	1.667	1.667

Note: the velocity u is zero everywhere.
[a] For the helium–air–R22 shock tube problem, just change γ to 1.249 in $0.6 \le x \le 1.0$.

Note that the small volume fraction of a particular material may lead to numerical instability if the time step of an explicit time integration scheme is calculated according to the full cell CFL condition. In order to maintain numerical stability without reducing the time step, we apply the mixing procedure [18] after each Runge–Kutta sub-step. For each material, the conservative quantities of a small volume fraction cell are mixed with those of the larger neighboring cells in a conservative way. The exchanges of the conservative quantities $\mathbf{M}$ are calculated according to the averaged values, see Ref. [18]. The conservative quantities for each material in the near interface cells are updated by

$$\alpha_{i,j}^{\chi,n+1}\mathbf{U}_{i,j}^{n+1} = (\alpha_{i,j}^{\chi,n+1}\mathbf{U}_{i,j}^{n+1})^* + \sum \mathbf{M}^x + \sum \mathbf{M}^y, \tag{38}$$

where the second and third terms on the right hand side represent the sum of all mixing exchanges of cell $C_{i,j}$ in the x and y directions, respectively.

Given the flow states $\alpha^n\mathbf{U}^n$ and the regional level-set data $\varphi^{\chi,n}$ at time step n, the coupling between the interface capturing method in Sec. 2.2 and the fluid evolution described by Eq. (13) advances the interface $\varphi^{\chi,n+1}$ and flow states $\alpha^{n+1}\mathbf{U}^{n+1}$ to the next time step $n+1$ through the following operation sequence:

- Update the volume fraction α^n, aperture A, and normal direction $\mathbf{n}$ using the regional level-set function.
- Obtain the flow states $\mathbf{U}^n$ from $\alpha^n\mathbf{U}^n$ and the updated volume fractions.
- For each material χ, extend the fluid states defined on Ω^χ to the ghost fluid domain $\Omega \setminus \Omega^\chi$.
- Solve the material interaction to obtain the exchange flux $\hat{\mathbf{X}}$ and the interface velocity $\mathbf{u}^*$ by the reduced model (29).
- Evolve the conservative flow states by Eq. (13) to obtain $\alpha^{n+1}\mathbf{U}^{n+1}$.
- Solve the interface advection equation (7) to obtain the new regional level-set function $\varphi^{\chi,n+1}$.
- Invoke the mixing procedure (38) for small cut cells.

2.3.3. Interface scale separation model

In this section, we discuss a numerical procedure for consistent removal of non-resolved interface segments during multi-material simulations. For a given spatial resolution non-resolved interfacial scales, such as thin filaments and small droplets, need to be removed in order to avoid proliferation of artifacts. For this purpose, the separation of resolvable and non-resolvable interface scales is necessary. The scale separation operation for two-material-cells is performed with the model in Refs. [22,23] without need for modifications. In multi-material-cells non-resolved interface structures may occur

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

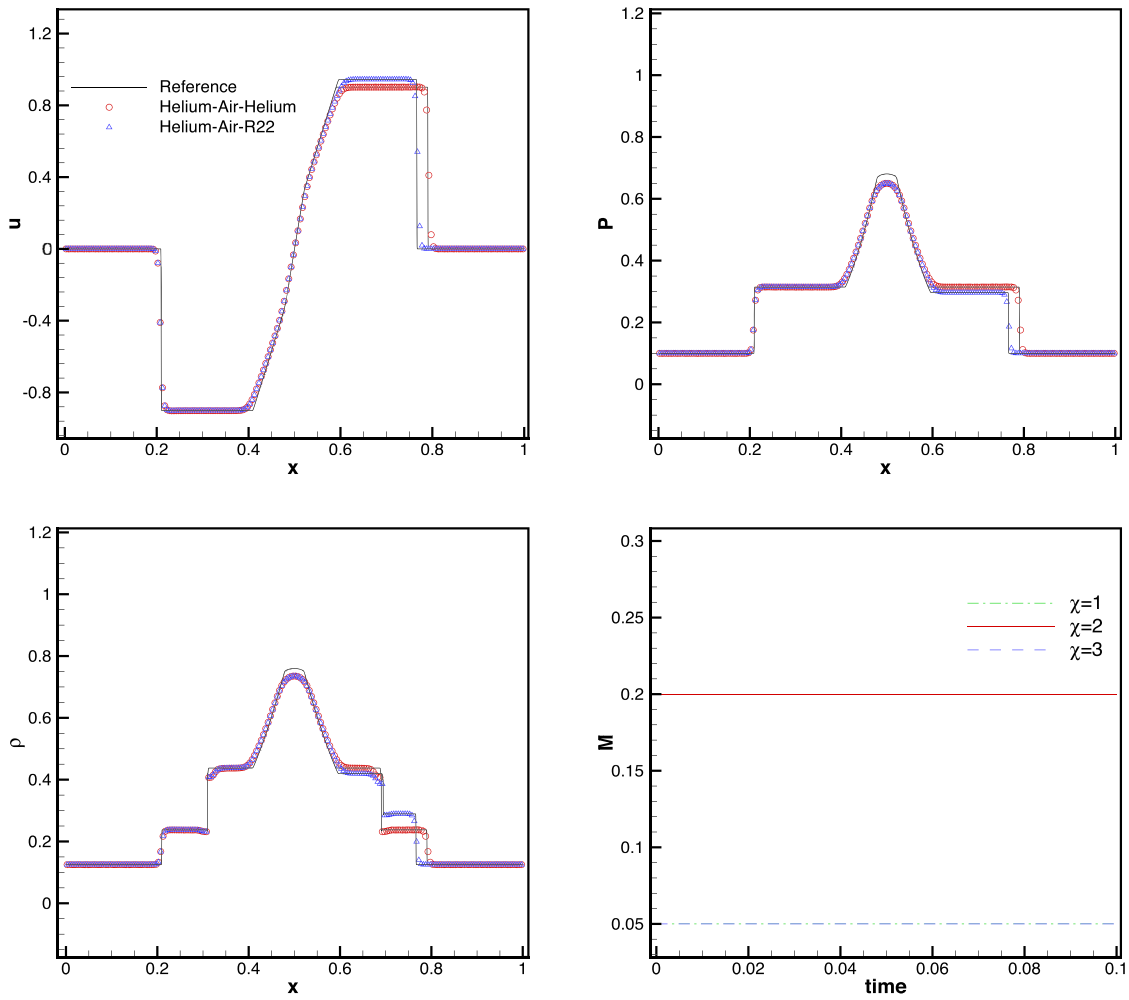


Fig. 6. Three-material shock tube problem.

across different material domains, so that the scale separation model must be adapted to multi-material interfaces. After the removal of non-resolved structures the new interface needs to be constructed between these materials in such a way that these different materials stay separated, however with a different connection relation.

The basic idea of the multi-material-interface scale separation model resembles that of the two-material model [22,23]: each material domain is simply connected as long as the interface is resolved; non-resolved interface segments, however, show up as not simply connected sub-domains when the level-set field is slightly shifted. This observation is used to find "oddball cells" where non-resolved interface segments exist. We define the $\epsilon^{\pm}$-material-interface as

$$\Gamma^{\chi}_{\epsilon^{\pm}} = \{\mathbf{x} \subset \Omega^{\chi} | \phi^{\chi}(\mathbf{x}) \pm \epsilon = 0, \chi \in \mathrm{X}\} \tag{39}$$

where ϵ is a small positive parameter [22] and can be set to $0.75h$ for 2D and $0.9h$ for 3D in order to remove interface segment whose scales are smaller than grid scale h [22]. The set of cut cells S_0 which contains the segments of the interface network is defined as

$$S_0 = \{C_{i,j} | \partial C_{i,j} \cap \Gamma \neq \emptyset\}. \tag{40}$$

For each material χ we define $S^{\chi}_{\pm}$ as the sets of cut cells containing the segments of $\partial\Omega^{\chi}_{\epsilon^{\pm}}$

$$S^{\chi}_{\pm} = \{C_{i,j} | \partial C_{i,j} \cap \partial\Omega^{\chi}_{\epsilon^{\pm}} \neq \emptyset\}. \tag{41}$$

The scale separation operation consists of three main steps. First, we identify the oddball cells where non-resolved interface segments reside. The selection criterion of oddball cells is similar as that of Ref. [23], which requires that to each oddball no neighbor exists that belongs to $S^{\chi_1}_{-}$

$$S_* = \{C_{i,j} | (C_{i,j} \in S_0) \wedge (\forall i_0, j_0 \in \{-1, 0, 1\}, C_{i+i_0, j+j_0} \notin S^{\chi_1}_{-})\}, \tag{42}$$

where $\chi_1 = \chi_{i,j}$ is the primary indicator of $C_{i,j}$. Second, the local non-resolved topology of the oddball cell in S_* is altered to a new resolved topology by replacing its indicator with the indicator corresponding to the second largest volume fraction

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

Table 2
Initial conditions and EOS parameters for 1D copper–TNT–copper and aluminum–TNT–copper impact problems.

Location	Case II[a]		
	$0 \le x < 0.3$ (copper plate)	$0.3 \le x < 0.7$ (detonation products)	$0.7 \le x \le 1.0$ (copper plate)
χ	1	2	3
ρ	8.924	2.48537	8.924
p	0	37	0
Γ_0	1.96	0.25	1.96
ρ_0	8.924	1.84	8.924
A_0	/	854.5	/
B_0	/	20.5	/
R_1	/	4.6	/
R_2	/	1.35	/
e_0	/	8.15	/
c_0	3.91	/	3.91
S	1.51	/	1.51

Note: the velocity u is zero everywhere.
[a] For the aluminum–TNT–copper impact problem, just change ρ to 2.785 in $0 \le x < 0.3$. Its EOS parameters are modified by $\Gamma_0 = 3$, $\rho_0 = 2.785$, $c_0 = 5.238$, and $S = 1.338$.

$$\chi_1^* = \arg\max_{\chi_r \neq \chi_1} \alpha_{\chi_r} \tag{43}$$

as changing from χ_1 to χ_1^* requires the least stimulus of level-set values for this cell and thus χ_1^* has the largest possibility to occupy this cell after scale separation. Note that this procedure automatically generates a new interface. Third, the unsigned distance function of cell containing non-resolved interface segment is assigned with new data that fit the interface network of the new resolved topology. This is accomplished by the same operation as in Ref. [23]. Because the considered cell $C_{i,j}$ in S_* belongs to $\Omega^{\chi_1^*}$ after scale separation, we calculate the distance from its center to $\partial\Omega_{\epsilon^+}^{\chi_1^*}$

$$d = \sqrt{\left[i - i_0\Delta x + (\varphi_{i_0,j_0} + \epsilon)n_x\right]^2 + \left[j - j_0\Delta y + (\varphi_{i_0,j_0} + \epsilon)n_y\right]^2}, \tag{44}$$

where (i_0, j_0) is the index pair of the cell in $S_+^{\chi_1^*}$ within a search stencil $i-3 \le i_0 \le i+3$, $j-3 \le j_0 \le j+3$. $\{n_x, n_y\}$ is the unit normal vector at the center of C_{i_0,j_0}. Subsequently, we update the regional level-set with

$$\phi_{i,j}^* = (\varphi_{i,j}^*, \chi_1^*), \quad \varphi_{i,j}^* = |d_{min} - \epsilon|. \tag{45}$$

Fig. 4 shows two illustrative test cases (a thin filament and a small droplet) where non-resolved interface scales exist. The computational domain is a unit square with the grid size $\Delta x = \Delta y = 0.1$. The thickness of the thin filament is $1.2\Delta x$, and the radius of the small droplet is $1.5\Delta x$. In Fig. 4(a), two interfaces that separates three materials merge to one after using the scale separation model, indicating that the material in the filament is entirely removed. The small droplet in Fig. 4(b) becomes an isolated bubble and the filament connecting the droplet and the main body collapses to a single interface while the other interface segments remain invariant. Two additional triple points are generated as expected. The level-set contours remain regular after scale separation and the contours inside the bubble do not change by the scale-separation operation, as shown in Figs. 4(c) and 4(d).

2.4. Space-time adaptivity

To achieve high computational efficiency and low memory storage, the space-time adaptivity strategy developed in Ref. [25] is incorporated into our compressible multi-material method with minor changes. In detail, the multi-resolution method [31] is used for mesh refinement due to its high rate of data compression. The projection and prediction operators [32] are defined based on the cell-averaged multi-resolution representation. For simplicity, the 1D operators with 5th-order interpolation are

$$P_{\ell+1\to\ell}: \quad \bar{u}_{\ell,i} = \frac{1}{2}(\bar{u}_{\ell+1,2i} + \bar{u}_{\ell+1,2i+1}), \tag{46}$$

and

$$P_{\ell\to\ell+1}: \quad \hat{u}_{\ell+1,2i} = \bar{u}_{\ell,i} + \sum_{m=1}^{2} \gamma_m(\bar{u}_{\ell,i+m} + \bar{u}_{\ell,i-m}), \tag{47}$$

$$\hat{u}_{\ell+1,2i+1} = \bar{u}_{\ell,i} - \sum_{m=1}^{2} \gamma_m(\bar{u}_{\ell,i+m} + \bar{u}_{\ell,i-m}),$$

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

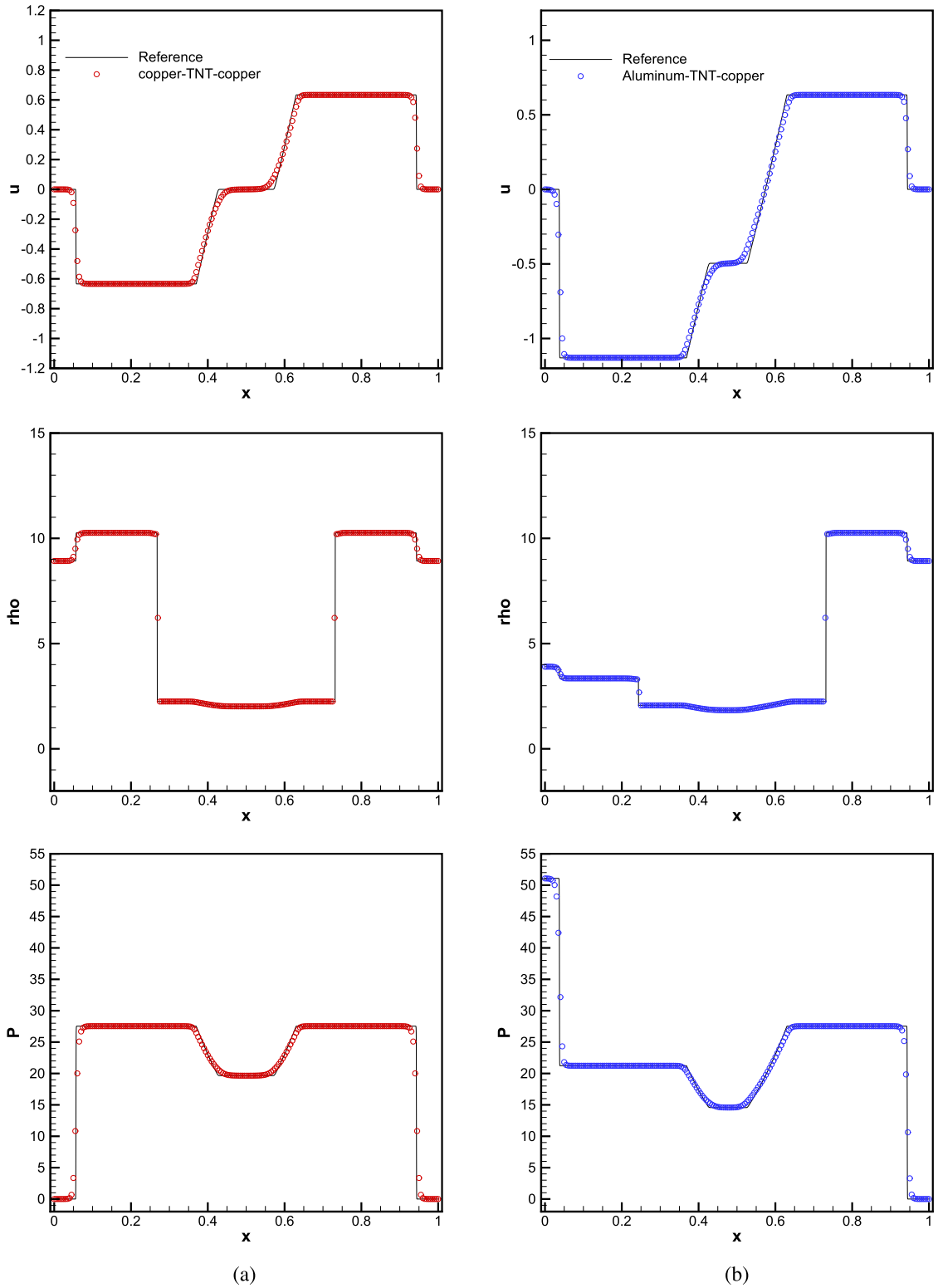


Fig. 7. Three-material explosive driving and high-speed impact problems: (a) the copper–TNT–copper configuration and (b) the aluminum–TNT–copper configuration.

where ℓ is the index of levels, and γ_m is the interpolation coefficient. Mesh refinement and coarsening are accomplished by comparing the prediction error $\bar{d}_{\ell,i} = \bar{u}_{\ell,i} - \hat{u}_{\ell,i}$ with a level-dependent threshold [25]. A multi-step Runge–Kutta local time stepping scheme [33] is employed to achieve time adaptivity and thus obtain additional speed-up. To maintain strict conservation a conservative flux correction [33] is adopted between cells with different levels.

The pyramid data structure and storage-and-operation-splitting approach proposed in Ref. [25] are used here. The block containing cells which reside in the narrow band of the interface network is refined to the finest level ℓ_{max} and denoted

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

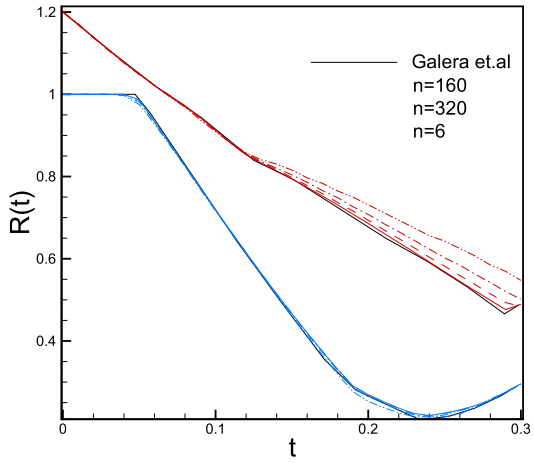
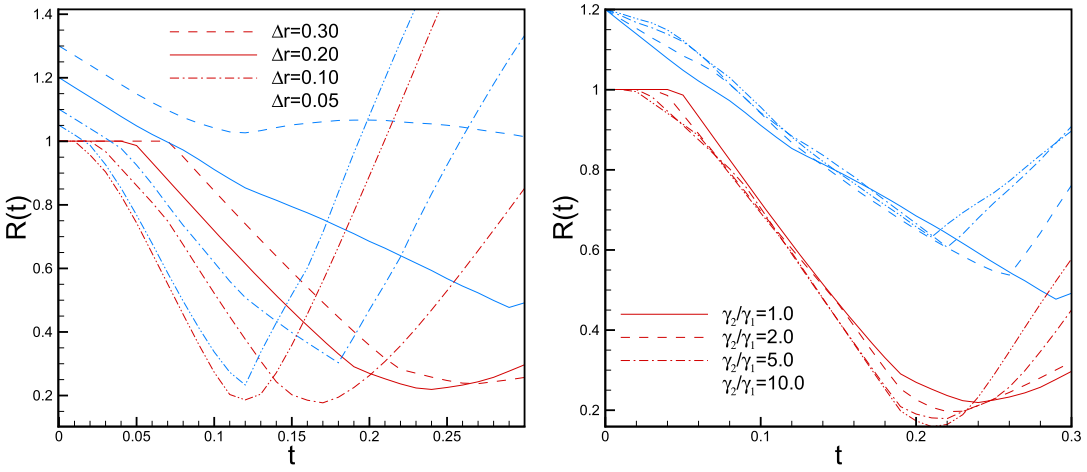


Fig. 8. Trajectories of the inner (blue lines) and outer (red lines) interfaces of the shell in 1D ICF implosion problems. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) (b)

Fig. 9. Parameter study of the ICF implosion problem: (a) different shell thickness and (b) different γ_2/γ_1.

as "multi-material block", otherwise as "single-material block". In such way operations related to the interface, including interface interaction, mixing procedure, scale separation and level-set advection, are only conducted at the finest level. The block position identifier [25] is used to distinguish the block location. The identifier at the finest level is 1 whenever the block has cells which occur in the narrow band of any cut cell, otherwise it is 0. A cell $C_{i,j}$ contains the interface network if it is intersected by the zero contour of local level-set field

$$\exists r \in \mathrm{X}_s,\, C_{i,j} \cap \{\mathbf{x} | \phi^r(\mathbf{x}) = 0\} \neq \emptyset, \tag{48}$$

or the indicator field in V_s differs from $\chi_{i,j}$,

$$\exists C_{k,l} \in V_s,\, \chi_{k,l} \neq \chi_{i,j}. \tag{49}$$

The position identifier of blocks at other levels are obtained according to Ref. [25]. The final multi-resolution representation of a multi-material problem is generated by locations of interface and shock waves, see e.g. Fig. 5.

Note that a "single-material block" may also exist at the finest level if a shock wave resides in this block. For a "multi-material block", we allocate memory for one single regional level-set field and $\mathcal{N}_b$ flow state fields, where $\mathcal{N}_b$ is determined by searching all unique indicator inside the inner and buffer zone of this block and satisfies the relation $\mathcal{N}_b \ll \mathcal{N}$ when $\mathcal{N}$ is large. For a "single-material block", the volume fraction and apertures become unity, and the interface exchange terms vanish. Thus the governing equation Eq. (13) degenerates to a standard finite volume scheme on a Cartesian grid.

3. Numerical validation

In this section, we assess the accuracy and robustness of the present method by a number of test cases. The materials are assumed to be inviscid compressible fluids whose dynamics are governed by Eq. (1), with different EOS for different

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

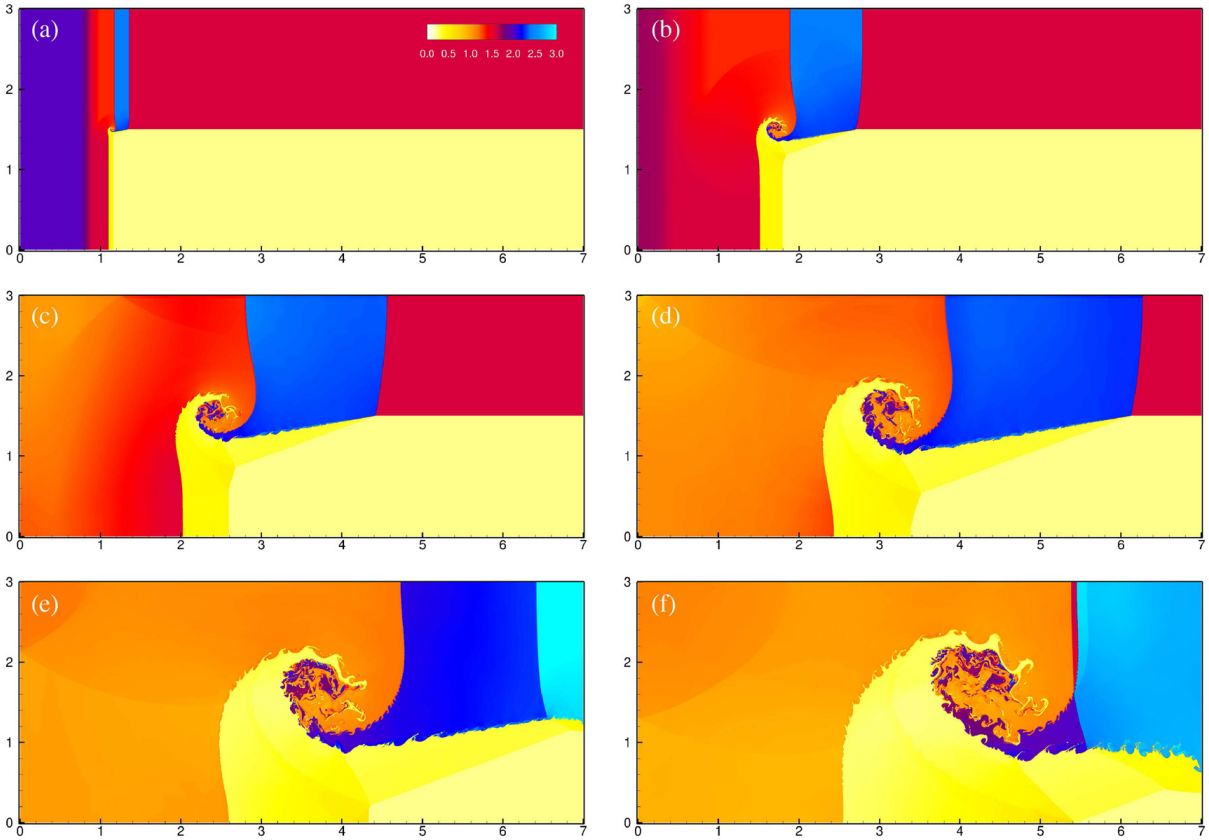


Fig. 10. Internal energy distributions at $t = 0.2$, 1.0, 2.0, 3.0, 4.0 and 5.0 for 2D compressible triple point problem with $\ell = 5$.

materials, and for which the interaction between each material pair can be written as in Eq. (29). A model extension may be required for other types of materials. Although a recently developed interface extraction algorithm [34] helps to determine the sub-cell interface geometrical information (normal direction and volume fraction), the development of a computationally efficient model that fully takes into account the interaction of more than three materials within a single cell is not straightforward. Applications of the present numerical model are limited to fluid-like materials.

First, the 1D multi-material shock tube and ICF implosion are considered. Then more complex 2D cases, including 2D ICF implosion, compressible triple point and shock wave interactions in multiple materials serve to demonstrate the robustness of interface interaction model and multi-material scale separation method in high-resolution simulations. For all test cases, discretization of the flow and interface evolution equation is performed by a 5th-order WENO [35] and a 2nd-order strongly stable Runge–Kutta scheme [36], with a CFL number of 0.6.

3.1. Shock-tube problem (I)

The three-material shock-tube problem of two helium gases and one air gas modeled by an ideal-gas EOS is an extension of the two-material shock-tube problem in Refs. [18,15]. Reflective boundary conditions are applied at $x = 0$ and $x = 1$. The initial condition is listed in Table 1. The grid spacing is $\Delta x = 5.0 \times 10^{-3}$ which we compare with a high-resolution result with $\Delta x = 2.5 \times 10^{-4}$. At $t = 0$, two Riemann problems occur at $x = 0.4$ and $x = 0.6$ and generate symmetric wave types. Two shock waves move towards the left and right boundaries while two rarefaction waves approach to each other. Finally, at $t = 0.1$ these two rarefaction waves impact and interact with each other, as shown in Fig. 6. Good agreement with the reference solution is observed and the distributions of flow variables exhibits symmetric profiles. When we modify the ratio of specific heats in $0.5 \leq x \leq 1.0$ to R22, i.e. $\gamma = 1.249$. The problem becomes asymmetric, as can be seen in Fig. 6. The results demonstrate the conservation of mass for each individual material.

3.2. 1D Explosive driving and high-speed impact problem (II)

Now we consider three-material explosive driving and high-speed impact problems which contain more complex EOS. Following Refs. [37,29,24], we model the explosive by the JWL EOS

$$p = A_0 \exp\left(\frac{-R_1\rho_0}{\rho}\right)\left(1 - \frac{\rho}{R_1\rho_0}\right) + B_0 \exp\left(\frac{-R_2\rho_0}{\rho}\right)\left(1 - \frac{\rho}{R_2\rho_0}\right) + \Gamma_0\rho(e + e_0), \tag{50}$$

and the impacting materials by the MG EOS,

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

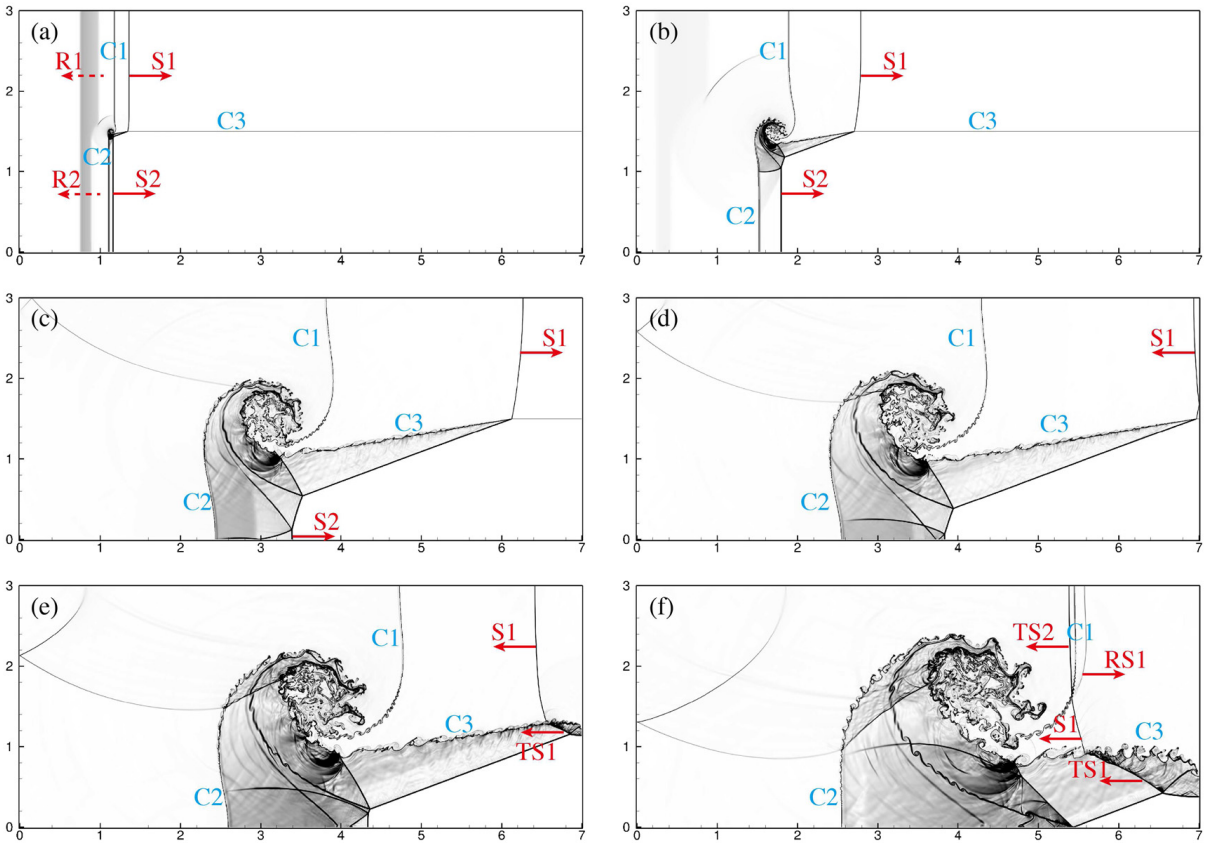


Fig. 11. Snapshots of density gradient at $t = 0.2$, 1.0, 3.0, 3.5, 4.0 and 5.0 for 2D compressible triple point problem with $\ell = 5$.

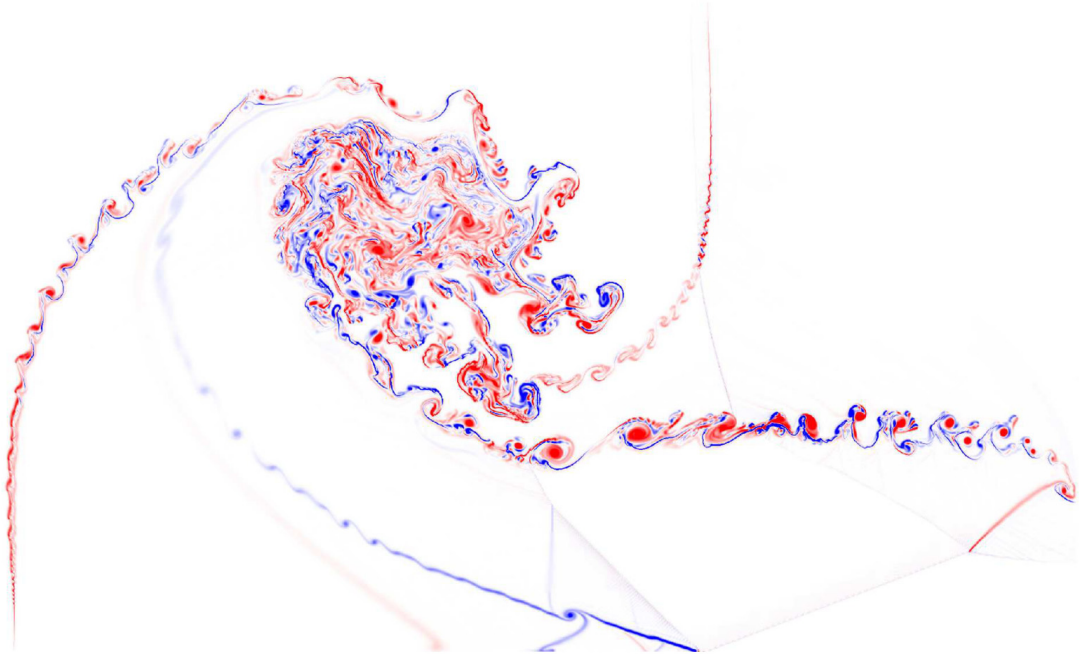


Fig. 12. Vorticity contours at 5.0 for 2D compressible triple point problem with $\ell = 5$.

$$p = p_r + \Gamma(\rho)\rho(e - \frac{1}{2}p_r\frac{\rho - \rho_0}{\rho\rho_0}), \tag{51}$$

where

$$p_r = \rho_0 c_0^2 \frac{1 - \rho_0/\rho}{(1 - S\,(1 - \rho_0/\rho))^2} \quad \text{and} \quad \Gamma(\rho) = \frac{\rho_0}{\rho}\Gamma_0. \tag{52}$$

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

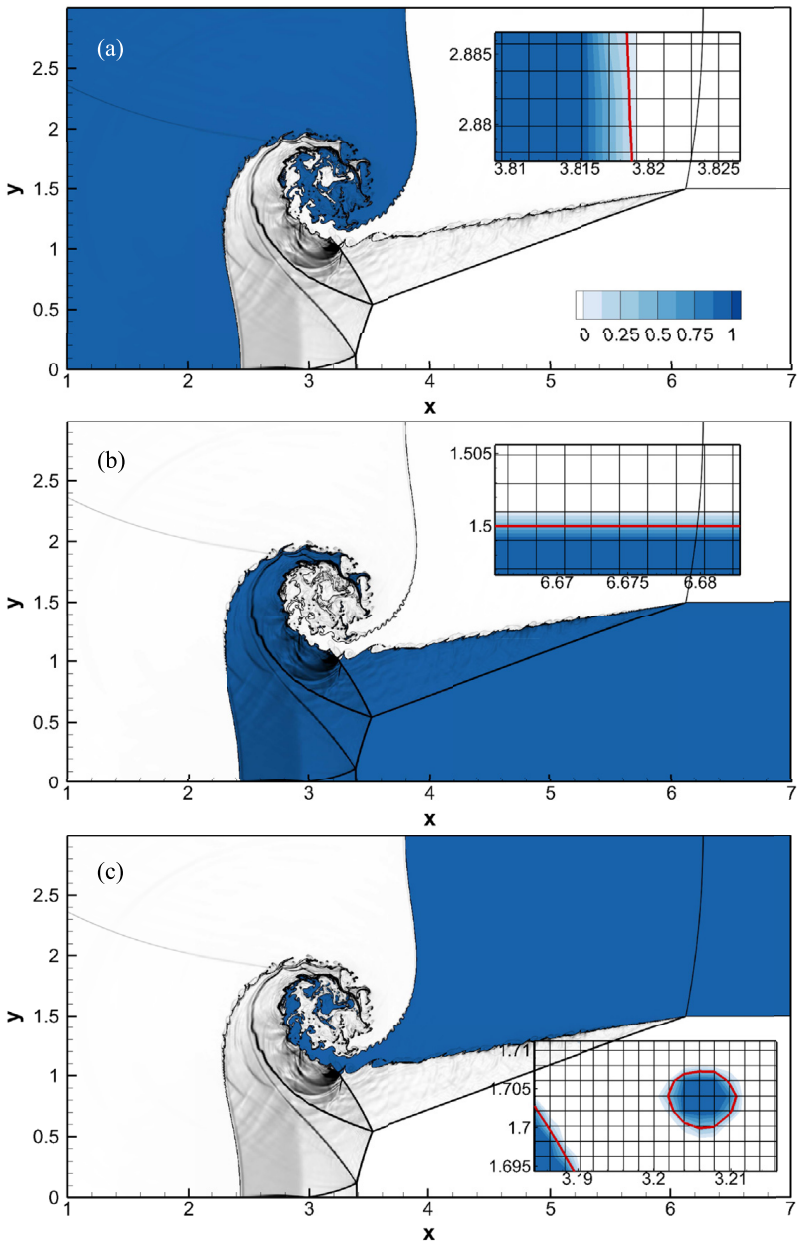


Fig. 13. Volume fraction contours of $\chi = 1$ (a), $\chi = 2$ (b), and $\chi = 3$ (c) at time $t = 3.0$ for 2D compressible triple point problem with $\ell = 5$. The extracted interfaces (red lines) are shown in the insets. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The parameters in Eqs. (50) and (52), Γ_0, ρ_0, A_0, B_0, R_1, R_2, e_0, c_0, S, are listed in Table 2 for the different materials. The mesh resolution and boundary conditions are the same as that of the above shock-tube case. First we test a symmetric configuration where the domain is decomposed into three parts: (i) a copper plate ($0 \le x < 0.3$), (ii) a product gas of the explosive TNT ($0.3 < x \le 0.7$), and (iii) another copper plate ($0.7 < x \le 1$). The initial condition is shown in Table 2, and the final time is $t = 0.05$. Then we replace the material of the first part ($0 \le x < 0.3$) by an aluminum plate. As shown in Fig. 7, our coarse-resolution ($\Delta x = 5.0 \times 10^{-3}$) numerical results agree well with high-resolution results ($\Delta x = 2.5 \times 10^{-4}$).

3.3. 1D cylindrical ICF implosion (III)

The setup of the ICF implosion simulations in a cylindrical geometry is taken from Ref. [38]. The computational domain is $[0, 1]$ and 1D axisymmetrical grids with different resolutions are used. There are three materials in the domain. A light fluid is located in the core region of the target and is surrounded by a shell of dense fluid. Outside the shell is an ambient material which is not solved during the simulation. Thus we treat the interface between the shell and the ambient material as a free surface boundary. To drive the implosion the pressures $p(t)$ imposed on that boundary are initially constant and then decrease linearly as

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

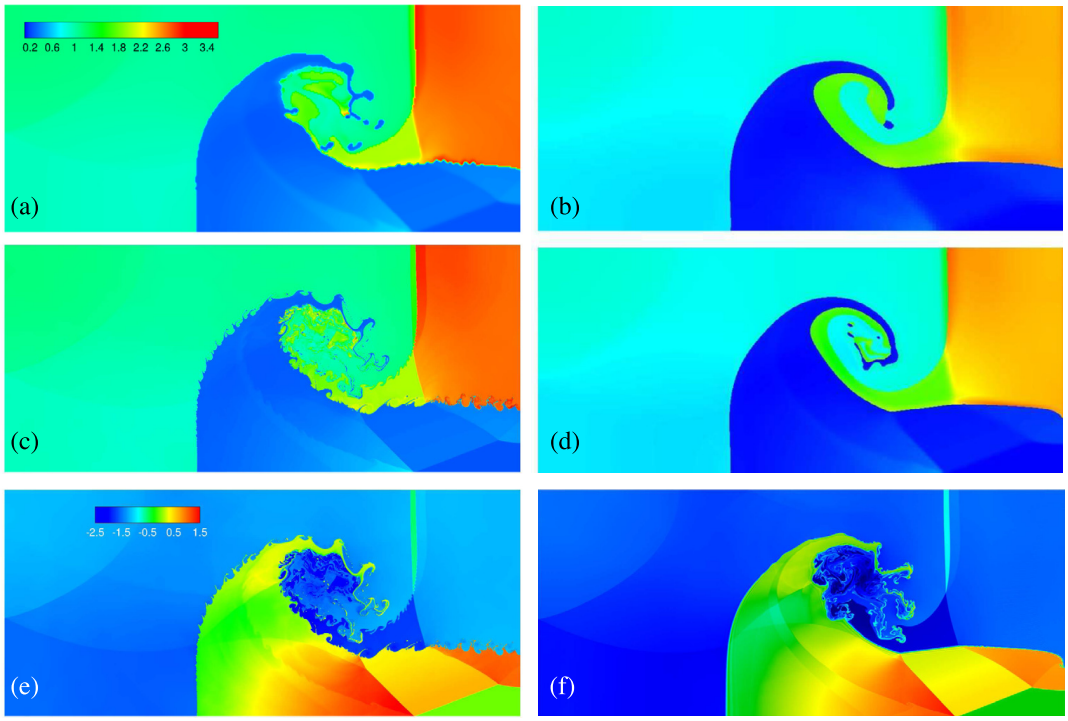


Fig. 14. Comparison between the present numerical method (left column) and other existing methods [42,41] (right column) for the 2D compressible triple point problem. The internal energy contours obtained by our numerical method with $\ell = 2$ (an effective resolution of 448×192) and $\ell = 5$ (an effective resolution of 3584×1536) are plotted in (a) and (c), respectively. The internal energy contours of the ALE method (reproduced from [41] with permission of Elsevier BV 2017) with resolution 140×60 and 560×240 are shown in (b) and (d), respectively. The density contours (logarithm scale) of our numerical method with $\ell = 5$ and the curvilinear finite element method [42] (high-resolution results are available at https://computation.llnl.gov/projects/blast/triple-point-shock-interaction) are shown in (e) and (f), respectively.

$$p(t) = 13 - \frac{12.5(t - 0.04)}{0.125 - 0.04}, \tag{53}$$

according to Ref. [38].

The evolution of the locations of inner and outer interfaces of the shell are shown in Fig. 8. Fig. 8(a) shows that our results converge to the Lagrangian result [38] with the cell number increasing from 160 to 1280. At the initial stage the outer interface moves inward under a constant pressure, leading to a shrinking of the shell until $t = 0.047$. Both the interfaces move towards the core. As the contraction speed of the inner interface is larger than that of the outer one, the thickness of the shell increases. The deceleration of the light fluid is observed from $t = 0.17$ to $t = 0.24$ when the radius of inner interface reaches its minimum value (referred to as "stagnation time" [38]). Afterwards the light fluid exhibits a expansion.

In order to demonstrate the versatility of the method, we conduct multiple simulations with varying parameters r_2 and γ_2, which are the radius of the outer interface and the ratio of specific heats of the heavy fluid, respectively. Such a parameter study represents a typical design issue for an ICF capsule, as the thickness of the shell and the EOS used in the previous simulations [38] are not realistic. As shown in Fig. 9(a), the thickness of the shell during the simulation, the stagnation time and the compression rate reduce with the initial thickness $\Delta r = r_2 - r_1$ decreasing from 0.30 to 0.05. When the ratio γ_2/γ_1 varies from 1.0 to 10.0, the profiles for the inner and outer interfaces are shifted downwards and upwards, respectively, as shown in Fig. 9(b). Thus the achieved minimum radius of inner interface decreases as γ_2/γ_1 increase, indicating that stiffer shell materials generate higher compression rates.

3.4. Compressible triple point problem (IV)

In this section we consider a compressible triple point problem which contains three perfect gases and is often used to validate the accuracy and robustness of Lagrangian or ALE methods for multi-material compressible flow simulations [38,39]. The computational domain is a rectangle $[0, 7] \times [0, 3]$ and is partitioned into three sub-domains: (i) $[0, 1] \times [0, 3]$ which is filled with a high pressure high density fluid $(\rho, p, \gamma, \chi) = (1, 1, 1.5, 1)$, (ii) $[1, 1] \times [0, 1.5]$ which is occupied by a low pressure high density fluid $(\rho, p, \gamma, \chi) = (1, 0.1, 1.4, 2)$, and (iii) $[1, 7] \times [1.5, 3]$ which has a low pressure low density flow state $(\rho, p, \gamma, \chi) = (0.125, 0.1, 1.5, 3)$. Accordingly, a triple point exists initially at $(1, 1.5)$. Reflective boundary conditions are employed, and the final time of the simulation is 5.0. The coarsest level has 7×3 blocks and are refined to the finest level by the mesh refinement criterion in Sec. 2.4.

First, we show numerical results of a high-resolution simulation conducted with $\ell_{max} = 5$ and an effective resolution of 3584×1536 at the finest level. Upon comparing Fig. 10 with previous results in the literature [38,40], it can be seen that the

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

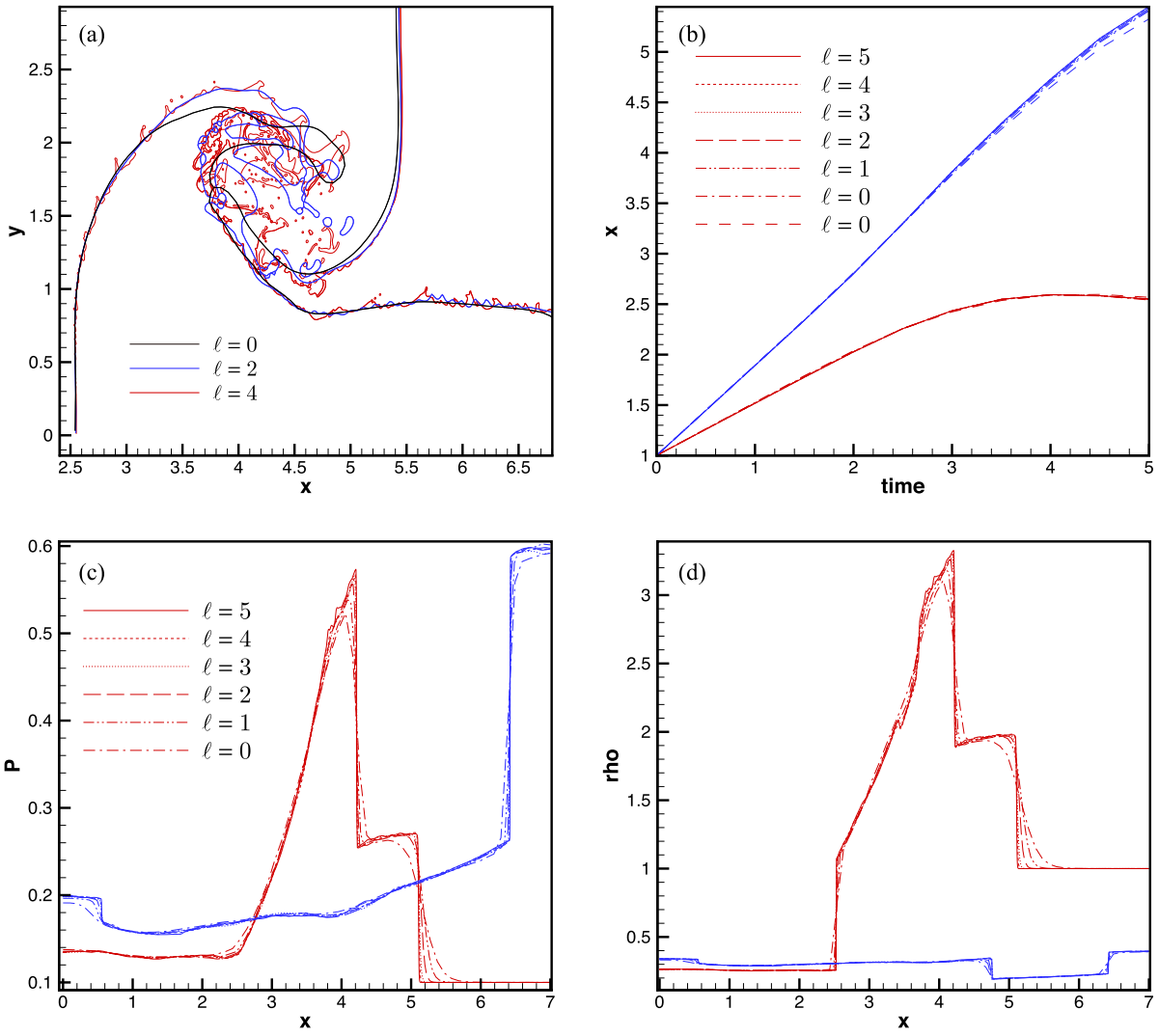


Fig. 15. Grid convergence tests for 2D compressible triple point problem. (a) The interface networks at $t = 5.0$ with different finest levels, $\ell = 0$, $\ell = 2$ and $\ell = 4$. (b) Convergence study of the interface location at the lower (red lines) and upper (blue lines) boundaries with increasing ℓ from 0 (the second resolution with $\ell = 0$ is half of the first one) to 5. The coarsest resolution is obtained with $\ell = 0$ and the number of inner cells is 8 while in other simulations the number of inner cells is 16. (c) The pressure profiles along x direction at $y = 0.5$ (red lines) and $y = 2.5$ (blue lines) with t being 4.0. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

internal energy contours at $t = 0.2$, 1.0, 3.0, 3.5, 4.0 and 5.0 are in good agreement. The development of the shock system is illustrated in Fig. 11. The initial Riemann problems at the discontinuities Γ_{13} and Γ_{12} generate a contact discontinuity ($C1$ or $C2$), a leftward rarefaction wave ($R1$ or $R2$) and a rightward shock wave ($S1$ or $S2$), see Fig. 11(a). The shock $S1$ moves faster than $S2$ as the acoustic impedance has a relation $\rho_2 c_{s,2} > \rho_3 c_{s,3}$. As a consequence, a distinct roll-up region appears around the triple point, as shown in Fig. 11(b). Near the triple point the shock reflection pattern is more complex as different waves interact with each other. Meanwhile, the development of disturbance can be observed in Fig. 11(c) due to the strong shear along all contact discontinuities $C1$, $C2$ and $C3$, which is not observed in numerical results of previous papers [38,40] and is indicative of less numerical dissipation. Fig. 11(d) shows the shock wave system just after $S1$ impacts the right boundary and is reflected. At this time instant, the interface inside the roll-up is strongly perturbed, leading to the shedding of small droplets. Afterwards, $S1$ moves upstream and is partially refracted to generate a transmitted shock $TS1$, as shown in Fig. 11(e). After $S1$ has reached the contact discontinuity $C1$, it becomes a transmitted shock $TS2$ inside the material 1 and a reflected shock $RS1$ is produced to maintain the mechanical equilibrium at the interface Γ_{13}, see Fig. 11(f). Due to the Kelvin–Helmholtz instabilities along the contact discontinuities, vortical structures are produced, as shown in Fig. 12. The volume fraction contours, computed by the recursive subdivision algorithm [25], are consistent with the density gradient contours and interface locations for all materials, as shown in Fig. 13.

In Fig. 14 we compare our results with that of an ALE method [41] and of a curvilinear-coordinate finite element method [42]. Generally, all results show similar large-scale vortex structures and shock reflection patterns at $t = 0.5$. Our results show the generation of finer small-scale structures upon grid refinement, as shown in Figs. 14(a) and (c), which is consistent with inviscid dynamics. A similar observation is made for the ALE simulations of Ref. [41], see Figs. 14(b) and (d). Also, our results exhibit enhanced physical interface instabilities with decreasing grid size, which again is consistent with

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

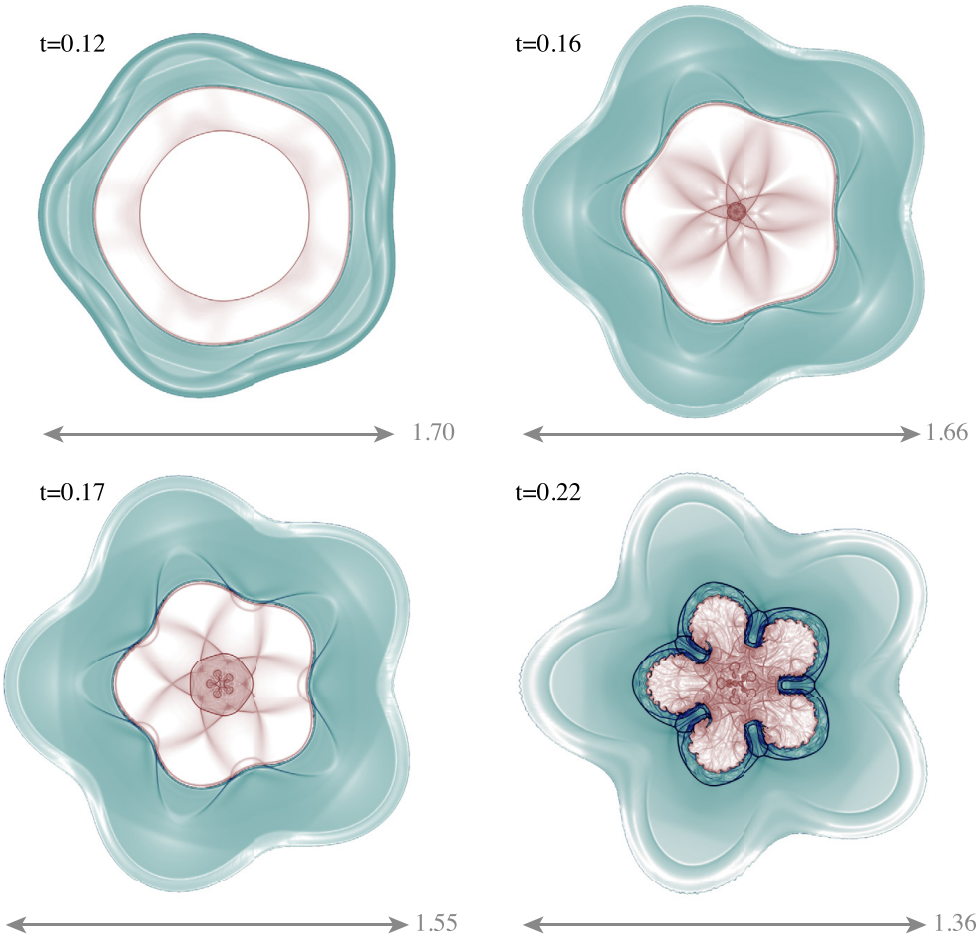


Fig. 16. Snapshots of density gradient before the stagnation time for 2D cylindrical ICF implosion with a 5-mode perturbation.

inviscid dynamics as smaller-scale Kelvin–Helmholtz instabilities along the contact discontinuities (e.g. C3) can be resolved on finer grids and are no longer suppressed by numerical dissipation. For high resolution simulations, vortical structures and smallscale features inside the primary vortex core are predicted consistently with the results of the curvilinear-coordinate finite-element method [42], see Figs. 14(e) and (f). As shown in Fig. 15(b), the interface locations at the upper and lower boundaries exhibit grid convergence. At $t = 4.0$, the shock waves and contact discontinuities corresponding to Fig. 11(e) are represented by the pressure and density profiles along the x direction at $y = 0.5$ and $y = 2.5$, see Figs. 15(c) and (d). The pressure and density jumps become sharper with increasing resolution from $\ell_{max} = 0$ to $\ell_{max} = 5$ and do not develop overshoots.

3.5. 2D cylindrical ICF implosion with perturbed interface (V)

Increasing the complexity of the configuration considered in Sec. 3.3, we investigate a 2D cylindrical ICF implosion with initial perturbations. This configuration demonstrates that the method can handle interface and small-scale flow instabilities from linear to highly nonlinear stages. The interface between light and heavy fluid has a single-mode perturbation [38,43] with

$$r_0' = r_0[1 + A\cos(m\theta)], \tag{54}$$

where r_0 and r_0' are the initial unperturbed and perturbed radii of the light fluid, and θ is the polar coordinate. The mode number m is 5 and 47 for low-mode and high-mode perturbations, respectively. The amplitude A of the initial perturbation is 2% of the wavelength of low-mode perturbation. A free surface boundary condition is prescribed at the outer interface. There is one block at the coarsest level and the maximum refinement level is $\ell_{max} = 8$. With each block containing 16×16 inner cells, the effective grid resolution at the finest level is 4096^2. Other parameters follow from the 2D extension of the 1D case in Section 3.3.

Figs. 16 and 17 show four snapshots of the density gradient before and after the stagnation time, respectively. At the early stage, $t = 0.12$ in Fig. 16, the shock wave passes the inner perturbed interface, and a slight distortion of the shell and light gas bubble is observed due to Richtmyer–Meshkov instability. The length scale in Fig. 16 decreases slowly. After maximum compression, Rayleigh–Taylor instability develops as the high pressure inside the light fluid bubble accelerates

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

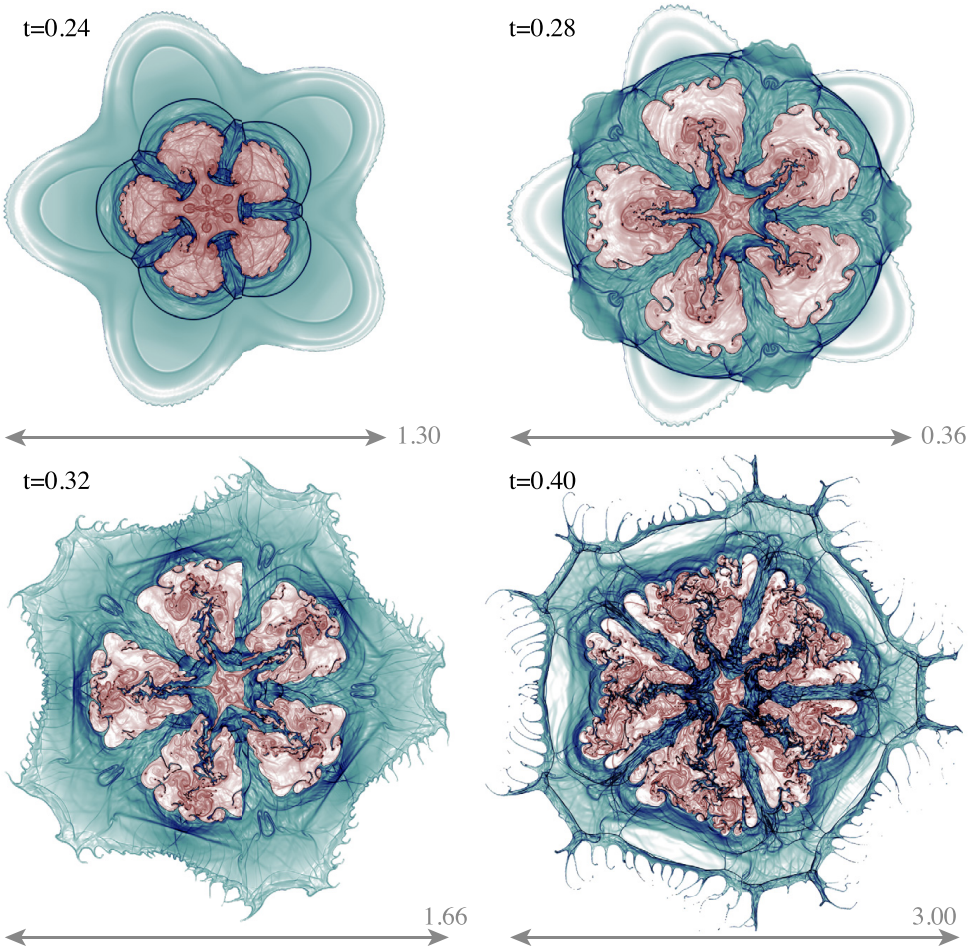


Fig. 17. Snapshots of density gradient after the stagnation time for 2D cylindrical ICF implosion with a high-mode perturbation.

the heavy shell. The low-mode instability has grown substantially from the stagnation time, which is confirmed by the rapid increase of the length scale in Fig. 17. Secondary instabilities near the inner interface can be seen, as shown in the snapshot $t = 0.28$. At time $t = 0.40$, the length scale shown in Fig. 17 indicates that the entire structure has significantly expanded and exhibits small-scale mixing [43]. For the high-mode case, the selected two snapshots in Fig. 18, $t = 0.22$, $t = 0.27$, are just before and after the maximum compression time. The outer boundary is composed of 47 fingers and show good symmetry preservation. The mixing zone containing most small scales at $t = 0.28$ is larger than that at $t = 0.27$, see the vorticity contours in Fig. 18. A wide range of resolved small interface scales are generated, indicating our method is very robust due to the multi-material scale separation model.

3.6. Shock wave interaction with a multi-material bubble (VI)

This problem represents a complex shock-accelerated inhomogeneous flow [44], and is a combination of a 2D air–helium [45,18,46,25] and an air-R22 [25,47,48] shock bubble interaction. A Mach 6.0 shock wave in air interacts with a cylindrical helium bubble with a R22 shell. The computational domain and initial conditions are shown in Fig. 19(a) and Table. 3, respectively. Symmetry boundary conditions are employed at the upper and lower boundaries, while inflow and outflow conditions are prescribed at the left and right boundaries. Simulations are performed with 4×1 blocks at the coarsest level and $\ell_{max} = 7$, leading to an effective resolution of 8192×2048 at the finest level.

In Fig. 20, the density gradient fields and materials distributions at 6 time instants illustrate the development of the shock system and the bubble deformation inside a medium with inhomogeneous flow states. At $t = 5.0 \times 10^{-3}$, the incident shock wave is refracted after crossing the upstream front of the R22 shell. As a result of the acoustic impedance mismatch $\rho_3 c_{s,3} > \rho_1 c_{s,1}$ at the air–helium interface, the transmitted shock wave has a concave curvature while the shock in the ambient air keeps planar. This convergent shock refraction pattern agrees with the numerical results in Refs. [25,48]. A reflected shock is generated at the upstream front and then propagates upstream inside the air [49,44]. When the concave transmitted shock wave impacts on the R22–helium interface, the reflected rarefaction occurs and moves upstream inside the R22 shell. The acoustic impedance mismatch $\rho_2 c_{s,2} > \rho_3 c_{s,3}$ at the air–helium interface produces a convex transmitted shock inside the helium bubble. This shock propagates downstream and subsequently impacts on the downstream surface of helium bubble. At $t = 1.0 \times 10^{-2}$, it moves across the interface entirely and becomes a re-transmitted shock in the R22

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

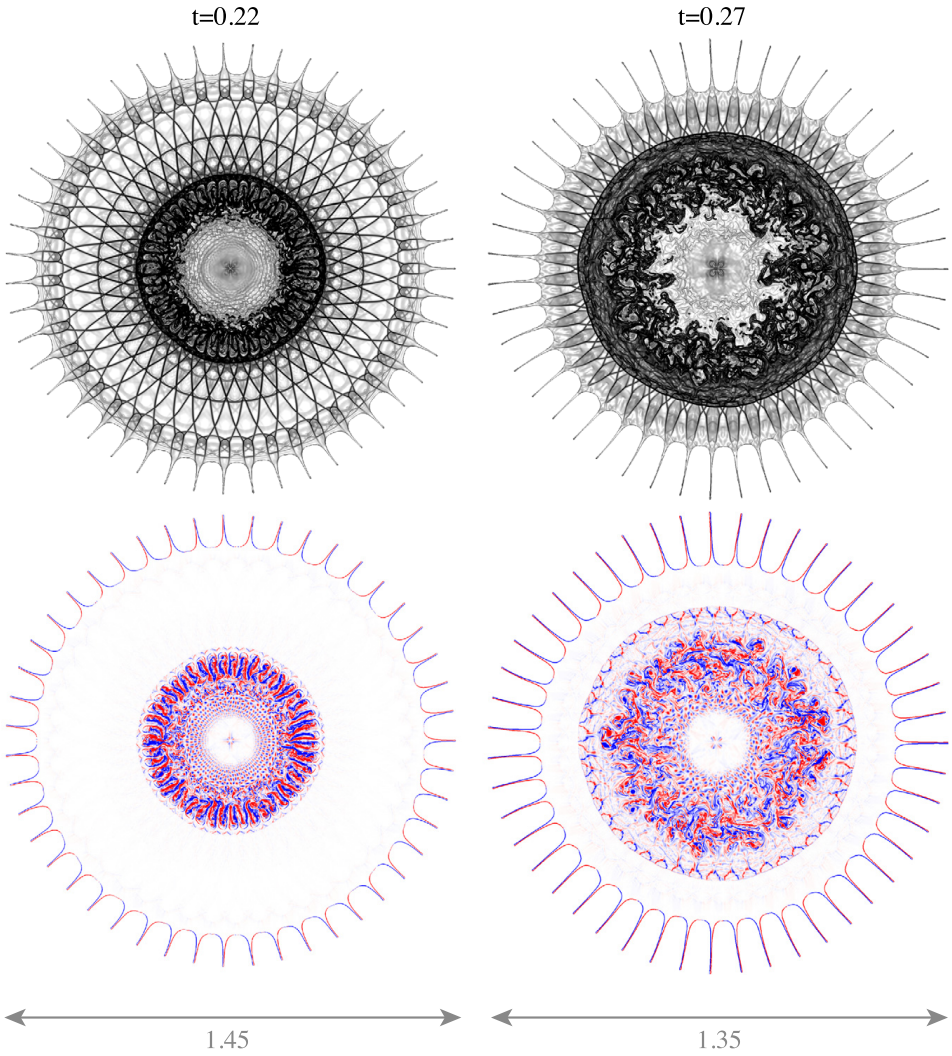


Fig. 18. Snapshots of density gradient and vorticity at $t = 0.22$ and 0.27 for 2D cylindrical ICF implosion with a high-mode perturbation.

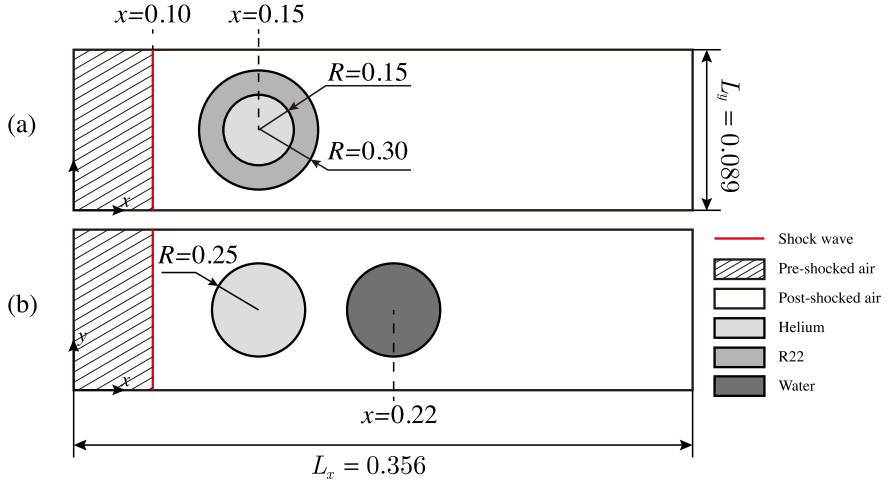


Fig. 19. Computational domains for 2D compressible 3-material flows: (a) case V and (b) case VI.

material, as shown in Fig. 20(b). Correspondingly, a Mach stem, triple point, slip line and re-transmitted reflected shock (moves to the upstream of the helium bubble in Fig. 20(b)) are produced in R22 material during this process [49]. Along the vertical direction one can observe a "fast–slow–fast–slow–fast" type of shock speed according to definition of Zabusky and Zeng [50]. The deformed helium bubble has a similar shape with previous numerical results [18,46,25], although its incident shock is not planar. The materials helium and R22 are accelerated with significantly different shock speed. The

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

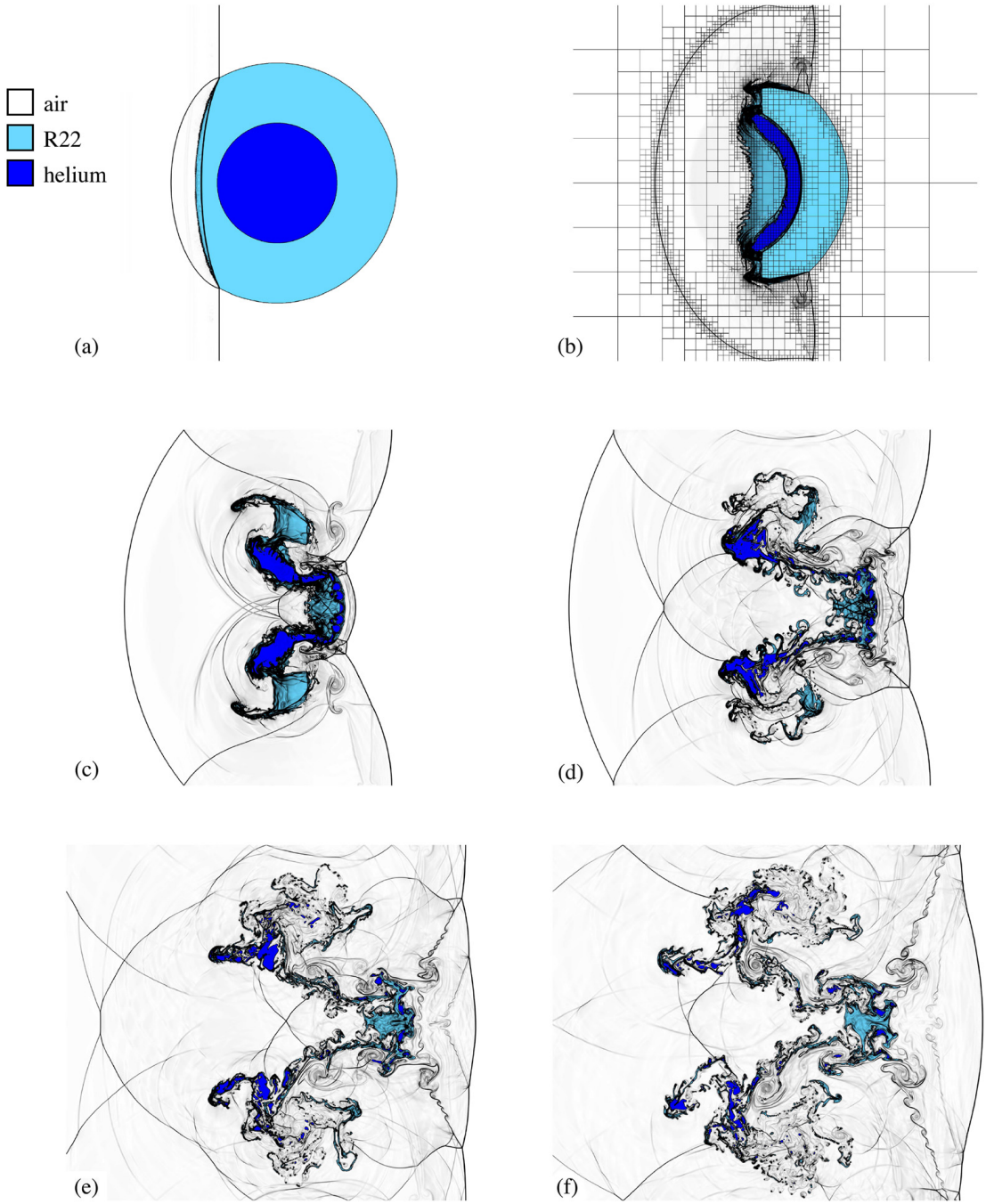


Fig. 20. Density gradient fields and materials distribution of case V at (a) $t = 5.0 \times 10^{-3}$, (b) $t = 1.0 \times 10^{-2}$, (c) $t = 1.5 \times 10^{-2}$, (d) $t = 2.0 \times 10^{-2}$, (e) $t = 2.5 \times 10^{-2}$ and (f) $t = 3.0 \times 10^{-2}$. A multi-resolution representation is outlined at $t = 1.0 \times 10^{-2}$.

global shape is extremely distorted, corresponding to "fast–slow–fast–slow–fast" shock wave, see Figs. 20(c) and 20(d). Several upstream-directed reflected shock waves with different shock strength occur at different locations in either R22 shell or helium bubble. These shocks interact with each other and finally with the reflected shock in Fig. 20(a), see Figs. 20(e) and 20(f). From $t = 1.0 \times 10^{-2}$ to $t = 3.0 \times 10^{-2}$, R22 and helium are mixed with the ambient air. The generation of many isolated R22 and helium droplets are captured by our simulation, as shown in Figs. 20(e) and 20(f) and in Fig. 21. The multi-resolution representation shown in Fig. 20(b) demonstrates that all blocks near the interfaces and the shock structures are refined to the finest level, $\ell_{max} = 7$.

3.7. Shock wave interaction with an Helium bubble and a water column (VII)

In this case, we consider a combination of shock-helium interaction [45,18,46,25] and shock-water interaction [25,48,51], where a helium bubble is initially accelerated by a Mach 6.0 planar shock and then impacts on a water column. The computational domain and initial conditions are detailed in Fig. 19(b) and Table 3 while the boundary conditions are

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

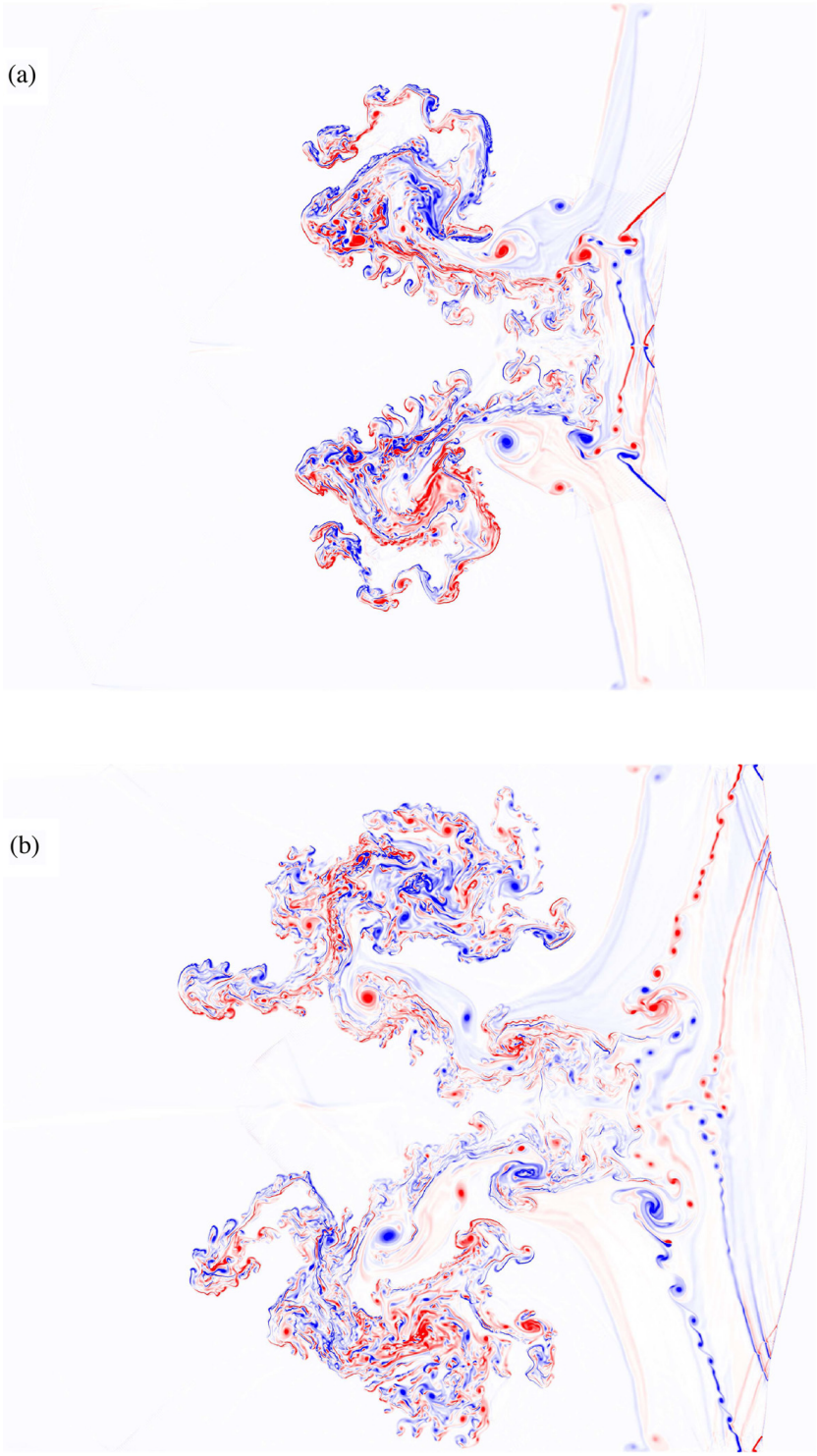


Fig. 21. Vorticity contours of case V at (a) $t = 2.0 \times 10^{-2}$ and (b) $t = 3.0 \times 10^{-2}$.

the same as in Sec. 3.6. We refine 4×1 blocks at the coarsest level to $\ell_{max} = 6$ to obtain an effective grid resolution of 4096×1024 at the finest level.

We plot 4 density gradient fields in Fig. 22 at $t = 1.0 \times 10^{-2}$, 1.2×10^{-2}, 1.5×10^{-2} and 1.8×10^{-2}, with helium colored by blue to track its evolution after impact on the water column. Note that the bubble deformation, the transmitted and reflected shocks, and the reflected rarefaction wave at $t = 1.0 \times 10^{-2}$ are in good agreement with those in Ref. [25], see Fig. 22(a). Triple point, Mach stem and slip line are observed at this time instant. The deformed helium bubble moves downstream and impacts on the water column at $t = 1.2 \times 10^{-2}$. A transmitted shock and reflected shock are generated at the helium–water interface. As the speed of sound in helium is comparable with that in water, the transmitted shock in water and re-transmitted shock in the ambient air together form a bow shock wave. Due to the high stiffness of the

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

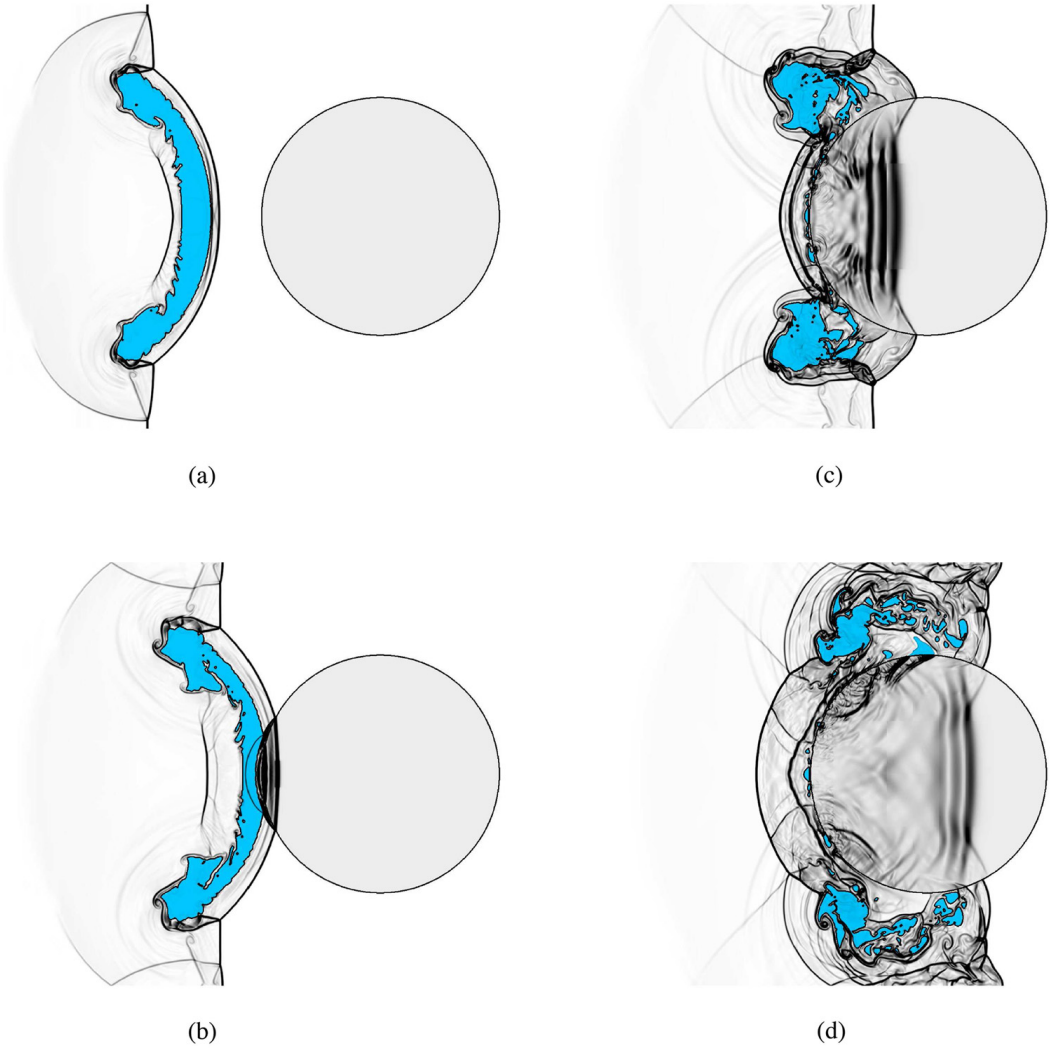


Fig. 22. Density gradient fields of case VI at (a) $t = 1.0 \times 10^{-2}$, (b) $t = 1.2 \times 10^{-2}$, (c) $t = 1.5 \times 10^{-2}$ and (d) $t = 1.8 \times 10^{-2}$. The distribution of helium is colored by blue to capture its interaction with water column (light gray). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Initial conditions for 2D test cases.

Location[a]	Case VI				Case VII			
	Post-shocked air	Pre-shocked air	Helium bubble	R22 shell	Post-shocked air	Pre-shocked air	Helium bubble	Water column
χ	1	1	2	3	1	1	2	3
ρ	1.0	5.268	0.138	3.154	1.0	5.268	0.138	1000.0
p	1.0	41.83	1.0	1.0	1.0	41.83	1.0	1.0
u	5.752	0.0	0.0	0.0	5.752	0.0	0.0	0.0
γ[b]	1.4	1.4	1.667	1.249	1.4	1.4	1.667	7.15

[a] See Fig. 19(a) for details.
[b] Tait's EOS is used for water column and idea gas EOS is for other gaseous materials.

water column, the helium keeps closely touched to the water column, as shown in Fig. 22(b). From $t = 1.5 \times 10^{-2}$ to $t = 1.8 \times 10^{-2}$, near two roll-up regions of helium propagate across the water column, while little helium material resides along the upstream surface of water column. Small droplets are generated due to the displacement effect of the water column. Complex shock refraction pattern develop, indicating the increased complexity as compared to configuration considered before [49,44].

4. Conclusion

The proposed conservative multi-material interface-interaction method comprises a multi-region level-set method, a conservative finite volume method, an interface interaction model and an interface scale separation model to cope with typical

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

problems in the numerical simulations of compressible multi-material flows. The advection of interface networks is handled by the multi-region level-set method. Conservation is strictly satisfied by solving the discretized governing equations with consistent multi-material-interaction models and by using a conservation correction for each multi-material cell. The sharp interface property is maintained by an efficient reduced interface interaction model to obtain the interface condition which serves to calculate the velocity of the interface network and the exchange flux across different materials, instead of solving the full Riemann problem inside a multi-material-cell. In addition, computational robustness is enhanced by removing non-resolved interface scales with a multi-material interface scale separation model. A range of test cases demonstrate that the proposed method has the intended properties and show that the method is very robust, flexible and efficient for simulations of compressible flows with multiple fluid-like materials. Although we only consider 1D and 2D test cases in this paper and inviscid flows, we emphasize that this method is straightforward to adapt to 3D and without difficulties can be extended to viscous compressible flows with source terms such as surface tension and gravity, by modifying the exchange fluxes in the multi-material cells.

Acknowledgements

This work is supported by China Scholarship Council under No. 201306290030. XYH acknowledges funding from National Natural Science Foundation of China (No. 11628206). NAA acknowledges funding from the European Research Council (ERC) under the Horizon 2020 grant agreement No. 667483. We acknowledge Dr. Tzanio Kolev for useful discussion about the triple-point-shock interaction problem.

References

[1] V.A. Thomas, R.J. Kares, Drive asymmetry and the origin of turbulence in an ICF implosion, Phys. Rev. Lett. 109 (7) (2012) 075004.
[2] E.J. Lentz, S.W. Bruenn, W.R. Hix, A. Mezzacappa, O.B. Messer, E. Endeve, J.M. Blondin, J.A. Harris, P. Marronetti, K.N. Yakunin, Three-dimensional core-collapse supernova simulated using a 15 $M_{\odot}$ progenitor, Astrophys. J. Lett. 807 (2) (2015) L31.
[3] G. Gisler, R. Weaver, C. Mader, M. Gittings, Two- and three-dimensional simulations of asteroid ocean impacts, LA-UR 02-66-30 Sci. Tsunami Hazards 21 (2) (2003) 119.
[4] K.A. Brakke, The surface evolver, Exp. Math. 1 (2) (1992) 141–165.
[5] C. Hirt, A.A. Amsden, J. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253.
[6] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1) (1981) 201–225.
[7] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12–49.
[8] D.J. Benson, Volume of fluid interface reconstruction methods for multi-material problems, Appl. Mech. Rev. 55 (2) (2002) 151–165.
[9] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (11) (2008) 5361–5384.
[10] K. So, X. Hu, N. Adams, Anti-diffusion interface sharpening technique for two-phase compressible flow simulations, J. Comput. Phys. 231 (11) (2012) 4304–4323.
[11] X.Y. Hu, B.C. Khoo, An interface interaction method for compressible multifluids, J. Comput. Phys. 198 (1) (2004) 35–64.
[12] S. Pan, X. Hu, N.A. Adams, High-resolution method for evolving complex interface networks, Comput. Phys. Commun. (2018), https://doi.org/10.1016/j.cpc.2018.01.001, in press.
[13] D.P. Starinshak, S. Karni, P.L. Roe, A new level set model for multimaterial flows, J. Comput. Phys. 262 (2014) 1–16.
[14] W. Zheng, J.-H. Yong, J.-C. Paul, Simulation of bubbles, Graph. Models 71 (6) (2009) 229–239.
[15] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (2) (1999) 457–492.
[16] J. Wang, K. Liu, D. Zhang, An improved CE/SE scheme for multi-material elastic–plastic flows and its applications, Comput. Fluids 38 (3) (2009) 544–551.
[17] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (1) (2005) 225–246.
[18] X. Hu, B. Khoo, N.A. Adams, F. Huang, A conservative interface method for compressible flows, J. Comput. Phys. 219 (2) (2006) 553–578.
[19] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, J. Comput. Phys. 227 (4) (2008) 2674–2706.
[20] M. Herrmann, A parallel Eulerian interface tracking/Lagrangian point particle multi-scale coupling procedure, J. Comput. Phys. 229 (3) (2010) 745–759.
[21] X. Hu, N. Adams, M. Herrmann, G. Iaccarino, Multi-scale modeling of compressible multi-fluid flows with conservative interface method, in: Proceedings of the Summer Program, vol. 301, Center for Turbulence Research, 2010.
[22] L. Han, X. Hu, N.A. Adams, Scale separation for multi-scale modeling of free-surface and two-phase flows with the conservative sharp interface method, J. Comput. Phys. 280 (2015) 387–403.
[23] J. Luo, X. Hu, N. Adams, Efficient formulation of scale separation for multi-scale modeling of interfacial flows, J. Comput. Phys. 308 (2016) 411–420.
[24] X. Hu, N. Adams, G. Iaccarino, On the HLLC Riemann solver for interface interaction in compressible multi-fluid flow, J. Comput. Phys. 228 (17) (2009) 6572–6589.
[25] L. Han, X. Hu, N. Adams, Adaptive multi-resolution method for compressible multi-phase flows with sharp interface model and pyramid data structure, J. Comput. Phys. 262 (2014) 131–152.
[26] B. Müller, F. Kummer, M. Oberlack, Y. Wang, Simple multidimensional integration of discontinuous functions with application to level set methods, Int. J. Numer. Methods Eng. 92 (7) (2012) 637–651.
[27] S. Pan, X. Hu, N. Adams, A consistent analytical formulation for volume-estimation of geometries enclosed by implicitly defined surfaces, arXiv preprint arXiv:1704.05355.
[28] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL-Riemann solver, Shock Waves 4 (1) (1994) 25–34.
[29] K.-M. Shyue, A fluid-mixture type algorithm for compressible multicomponent flow with Mie–Grüneisen equation of state, J. Comput. Phys. 171 (2) (2001) 678–707.
[30] V. Pasquariello, G. Hammerl, F. Örley, S. Hickel, C. Danowski, A. Popp, W.A. Wall, N.A. Adams, A cut-cell finite volume-finite element coupling approach for fluid–structure interaction in compressible flow, J. Comput. Phys. 307 (2016) 670–695.
[31] A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws, Commun. Pure Appl. Math. 48 (12) (1995) 1305–1342.
[32] O. Roussel, K. Schneider, A. Tsigulin, H. Bockhorn, A conservative fully adaptive multiresolution algorithm for parabolic PDEs, J. Comput. Phys. 188 (2) (2003) 493–523.

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007

[33] M.O. Domingues, S.M. Gomes, O. Roussel, K. Schneider, An adaptive multiresolution scheme with local time stepping for evolutionary PDEs, J. Comput. Phys. 227 (8) (2008) 3758–3780.
[34] S. Pan, X. Lyu, X.Y. Hu, N.A. Adams, High-order time-marching reinitialization for regional level-set functions, J. Comput. Phys. 354 (2018) 311–319.
[35] G.-S. Jiang, C.-W. Shu, Efficient Implementation of Weighted ENO Schemes, Tech. rep., DTIC Document, 1995.
[36] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (2) (1988) 439–471.
[37] R. Saurel, R. Abgrall, A multiphase Godunov method for compressible multifluid and multiphase flows, J. Comput. Phys. 150 (2) (1999) 425–467.
[38] S. Galera, P.-H. Maire, J. Breil, A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction, J. Comput. Phys. 229 (16) (2010) 5755–5787.
[39] X. Zeng, G. Scovazzi, A frame-invariant vector limiter for flux corrected nodal remap in arbitrary Lagrangian–Eulerian flow computations, J. Comput. Phys. 270 (2014) 753–783.
[40] M. Kucharik, M. Shashkov, Conservative multi-material remap for staggered multi-material arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 258 (2014) 268–304.
[41] D.P. Starinshak, J. Owen, A multimaterial extension to subzonal reconstruction, J. Comput. Phys. 313 (2016) 594–616.
[42] V.A. Dobrev, T.E. Ellis, T.V. Kolev, R.N. Rieben, High-order curvilinear finite elements for axisymmetric Lagrangian hydrodynamics, Comput. Fluids 83 (2013) 58–69.
[43] C. Joggerst, A. Nelson, P. Woodward, C. Lovekin, T. Masser, C.L. Fryer, P. Ramaprabhu, M. Francois, G. Rockefeller, Cross-code comparisons of mixing during the implosion of dense cylindrical and spherical shells, J. Comput. Phys. 275 (2014) 154–173.
[44] D. Ranjan, J. Oakley, R. Bonazza, Shock-bubble interactions, Annu. Rev. Fluid Mech. 43 (2011) 117–140.
[45] A. Bagabir, D. Drikakis, Mach number effects on shock-bubble interaction, Shock Waves 11 (3) (2001) 209–218.
[46] B. Hejazialhosseini, D. Rossinelli, M. Bergdorf, P. Koumoutsakos, High order finite volume methods on wavelet-adapted grids with local time-stepping on multicore architectures for the simulation of shock-bubble interactions, J. Comput. Phys. 229 (22) (2010) 8364–8383.
[47] J.-F. Haas, B. Sturtevant, Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities, J. Fluid Mech. 181 (1987) 41–76.
[48] R.R. Nourgaliev, T.-N. Dinh, T.G. Theofanous, Adaptive characteristics-based matching for compressible multifluid dynamics, J. Comput. Phys. 213 (2) (2006) 500–529.
[49] J.H. Niederhaus, J. Greenough, J. Oakley, D. Ranjan, M. Anderson, R. Bonazza, A computational parameter study for the three-dimensional shock–bubble interaction, J. Fluid Mech. 594 (2008) 85–124.
[50] N. Zabusky, S. Zeng, Shock cavity implosion morphologies and vortical projectile generation in axisymmetric shock–spherical fast/slow bubble interactions, J. Fluid Mech. 362 (1998) 327–346.
[51] C.-H. Chang, X. Deng, T.G. Theofanous, Direct numerical simulation of interfacial instabilities: a consistent, conservative, all-speed, sharp-interface method, J. Comput. Phys. 242 (2013) 946–990.

Please cite this article in press as: S. Pan et al., A conservative interface-interaction method for compressible multi-material flows, J. Comput. Phys. (2018), https://doi.org/10.1016/j.jcp.2018.02.007