MARS: AN ANALYTIC FRAMEWORK OF INTERFACE TRACKING VIA MAPPING AND ADJUSTING REGULAR SEMIALGEBRAIC SETS.

QINGHAI ZHANG * AND AARON FOGELSON †

Abstract. As a sequel to our previous work [Zhang, SIAM J. Numer. Anal., 51:2822, 2013] [Zhang and Fogelson, SIAM J. Sci. Comput., 36: A2369, 2014], this paper presents MARS, a generic framework for analyzing interface tracking (IT) methods via mapping and adjusting regular semialgebraic sets. Our mathematical model for moving material regions is the metric space of bounded regular semianalytic sets, equipped with Boolean algebras and advected by homeomorphic flow maps of a nonautonomous ordinary differential equation. By examining the actions of semidiscrete and discrete flow maps upon this metric space, we pinpoint in Lemma 3.9 a fundamental difficulty in achieving an IT accuracy higher than the second order. We then propose a generic IT method by concatenating three unitary operations on the modeling space, bound its overall IT error by the sum of intuitively defined error terms, and further estimate the individual errors in terms of the time step size and a Lagrangian length scale. The analytic utility of MARS is demonstrated by applying it to analyze a variety of IT methods including volume-of-fluid (VOF) methods and the improved PAM method (iPAM). MARS has a great potential in helping the further development of highly accurate and efficient IT methods. As an example, a cubic iPAM method inspired by MARS resolves two vortex-shear tests to machine precision on a 128-by-128 grid; it could also be vastly superior to VOF methods in terms of efficiency. Fourth-order accuracy in curvature estimation is also achieved under the framework of MARS.

Key words. interface tracking, MARS, polygonal area mapping, iPAM method, front tracking methods, volume-of-fluid methods, moment-of-fluid methods, regular semialgebraic sets, discrete flow maps, curvature estimation.

AMS subject classifications. 76T99, 14P10, 28A60

DOI. 10.1137/140966812 (Published in SIAM J. Numer. Anal.)

1. Introduction.

The study of multiple homogeneous materials separated by sharp and moving phase boundaries appears in various fields of science. The evolution of such a phase boundary generally depends both on the geometry of the interface and on the physics of the distinct phases. If the geometric quantities dominate, the motion of the interface can be described by

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \mathbf{u} = V(X, t, \mathbf{n}, \nabla\mathbf{n}), \tag{1.1}$$

where X is a position on the interface, t the time, $\mathbf{n}$ the normal vector at X, and $\mathbf{u}$ the velocity of the phase boundary at X and time t.

As fundamental difficulties of (1.1), an initially smooth interface may become C^1 discontinuous in a finite time, and topological changes of the interface may develop even in the presence of regularizing effects [25]. One way to resolve these difficulties is through analysis based on level sets [16, 6, 7, 17, 24], which admits the construction of weak solutions of (1.1) that are *unique* both during and after merging and separation of material regions. Over the past two decades, a large number of level set algorithms appeared with justified stability and convergence [5, 8, 10, 51, 11, 18, 21, 19, 20]. For systematic expositions and relatively more complete lists of references, the reader is referred to the books and monographs [9, 43, 23, 22] devoted to level sets.

* Department of Mathematics, University of Utah, 155 S. 1400 E. Rm 233, Salt Lake City UT 84112 USA (qinghai@math.utah.edu).

†Department of Mathematics and Bioengineering, University of Utah, 155 S. 1400 E., JWB 233, Salt Lake City UT 84112 USA (fogelson@math.utah.edu).

With the general context (1.1) in mind, we limit the scope of this work by making two assumptions:

(IT-1) the velocity of the interface is given a priori as $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$, which is independent of any geometric and physical quantities,

(IT-2) $\mathbf{u}(\mathbf{x}, t)$ is sufficiently continuous both in time and in space so that no topological changes of the tracked material can occur.

Level set methods [34, 42] still apply to this more restricted interface tracking (IT) problem and have been very successful in terms of simplicity and robustness. Two other popular families of IT methods are front tracking methods [50] and volume-of-fluid (VOF) methods [29]. Coupled, adaptive, variant versions of these methods have also been proposed, e.g. [45, 26, 48, 15, 41, 46, 31, 33, 28, 52, 30]. The moment-of-fluid (MOF) method [13, 1, 14, 2] reconstructs interfaces more accurately than VOF methods, while at the same time achieving independence from the underlying grid structure. Another relatively new family of IT methods that evolved from VOF methods is the polygonal area mapping (PAM) method [64] and the improved PAM method (iPAM) [62]. There the material regions near the interface are represented as piecewise polygons. Together with Lagrangian characteristic tracing of boundary points and Boolean operations of polygons, this polygonal representation removes the numerical diffusion in VOF and level set methods that smears the neighborhood of interfacial C^1 discontinuities[1]. Furthermore, a relation between the Eulerian grid size h and the Lagrangian length scale h_L of the interface is imposed in the iPAM method, leading to fourth-order convergence rates independent of dynamical kinks. Numerical results in [62] show that the iPAM method is more accurate and efficient than state-of-the-art VOF methods. As a limitation, currently the iPAM method only applies to two dimensions (2D).

This work has been driven by a list of open questions on the above IT methods.

(Q-1) Can we rigorously prove the second-order accuracy of the PAM method and the fourth-order accuracy of the iPAM method?

(Q-2) After answering (Q-1), can we gain any insights or utilities for the extension of the iPAM method to three dimensions (3D)?

(Q-3) For VOF methods, will the combination of second-order advection and second-order reconstruction yield an overall IT accuracy of the second order?

(Q-4) What is the main difficulty associated with improving VOF methods and other IT methods to accuracy higher than the second order?

(Q-5) The MOF method is second-order accurate in reconstructing static material regions [13]. How can we show the overall second-order IT accuracy when combining MOF reconstruction with a second-order advection algorithm?

(Q-6) How is the second-order accuracy of front tracking methods influenced by the operations that handle merges and separations of materials? Is it possible to improve front tracking methods to fourth-order accurate even in the presence of topological changes?

(Q-7) The fourth-order accuracy of the iPAM method is achieved at the price of an $O(1/h^2)$ complexity in 2D. Can we reduce the cost while retaining the accuracy?

(Q-8) Can we further improve the convergence rates of the iPAM method to even higher numbers such as 6 and 8?

[1] These interfacial C^1 discontinuities are sometimes referred to as *kink*s. It is well known that a purely hyperbolic system might generate new kinks from smooth initial conditions and smooth velocity fields. However, Sard's theorem [40] dictates that the kinks on the interface be a set of measure zero for a sufficiently continuous velocity field.

(Q-9) Curvature of a moving interface is one of the most important geometric quantities in many multiphase flow applications, but it is difficult to estimate curvature to high-order accuracy. Is it possible to achieve fourth-order accuracy in curvature estimation under the same framework proposed in this paper?

In dramatic contrast to level set methods, explicit IT methods are rarely analyzed systematically. The authors are unaware of any rigorous analytic framework for front tracking methods. Early characterization of the advection substep in VOF methods relies on the numerical analysis of scalar hyperbolic conservation laws. We believe, however, that the theories based on donating regions [58, 61] are more suitable for analyzing the advection schemes of VOF methods [57]. As for the reconstruction substep, the second-order convergence proof [37, 38] are for *static* material regions. It is an open question whether or not the reconstruction error accumulates to first order over multiple time steps. Given the different nature of the analyses on advection and reconstruction, it is not obvious at all how to answer (Q-3) and (Q-5). In our opinion, further developing VOF-type methods without answering questions (Q-3) and (Q-4) would be more or less a search in the dark. Therefore, a coherent analytic framework is urgently needed for analyzing current IT methods and for developing more accurate and efficient IT methods.

We answer this need by proposing a generic analytic framework called MARS, where, roughly speaking, the material regions are modeled by point sets with piecewise smooth boundaries and passively advected by flow maps determined from the velocity. MARS answers questions (Q-1) to (Q-6) under a single coherent framework. In particular, Lemma 3.9, the answer of (Q-4), pinpoints a fundamental difficulty associated with improving current second-order IT methods to a higher order of accuracy. Applicable to both 2D and 3D, MARS serves as the theoretical foundation for extending the iPAM method to 3D. In addition, MARS can be extended to treat material merges and separations via a coupling to the level set approach, since it is a trivial task to generate level sets from explicit representations of the interface. Hence the two assumptions (IT-1) and (IT-2) are not intrinsic limitations of MARS. In other words, MARS is potentially useful for the numerical analysis of (1.1) in more general contexts such as the geometric evolution equation [12].

As a product of interdisciplinary research, MARS draws from various fields such as solid modeling, Boolean algebra, and ordinary differential equations. However, the employed concepts from these fields are *elementary*, and they are explained with more details in a coherent manner in the supplement of this paper. We therefore hope that MARS could reach out to the wide audience of IT.

The rest of this paper is organized as follows. In Section 2, we briefly summarize our choice of a mathematical space for modeling moving material regions. In Section 3, we study the actions of various homeomorphic flow maps on the metric space of (either open or closed) bounded regular semianalytic sets. In Section 4, we propose a generic MARS method by concatenating three unitary operations on this metric space, define intuitive error terms, bound the overall error of the MARS method by the sum of individual errors, and estimate these errors in terms of the Lagrangian length scale and the time step size. The generality and utility of MARS are demonstrated in Section 5 and the supplement, where questions (Q-1) to (Q-6) are answered by analyzing current IT methods under the proposed framework. In Section 6, questions (Q-7), (Q-8), and (Q-9) are numerically investigated by performing two popular benchmark tests with a cubic iPAM method. Finally, Section 7 concludes this paper with several future research prospects.

2. Preliminaries. The formulation of the MARS framework calls for a mathematical space that appropriately models physically meaningful material regions. This space must meet three requirements: (i) each element in the space is described by a *finite* sequence of symbol structures; (ii) the modeling space is closed under relevant procedures such as Boolean operations; (iii) the employed mathematical entities are amenable to numerical analysis of IT.

As discussed in the supplement of this paper, the space of regular closed semianalytic sets form a Boolean algebra and it satisfies the aforementioned requirements (i) and (ii). The restriction of this space to some bounded elements leads to the *r-sets* [49], a concept widely used in solid modeling and computer-aided geometric design.

Unfortunately, the r-sets do not meet requirement (iii). First, regular open sets are sometimes preferred over regular closed sets for good reasons [3] even inside the discipline of solid modeling. Second, we track the codimension-1 interface by tracking the codimension-0 material region[2], but restricting the material regions to be closed is not amenable to numerical analysis. For example, donating regions, a geometric construction of flux sets of a fixed simple curve [58, 57, 61], can only be based on regular open sets[3]. Third, the IT accuracy in the MARS framework is measured by the volume of the symmetric difference between the true and computed solutions, and this error metric applies to both open and closed regular semianalytic sets. Therefore, it is neither necessary nor appropriate to require the regular sets to be closed.

We denote by $\mathbb{S}^{\mathrm{D}}$ the topological space of either open or closed bounded regular semianalytic sets with Euclidean topology. Similarly, $\mathbb{S}^{\mathrm{D}}_q$ denotes the subspace of $\mathbb{S}^{\mathrm{D}}$ where all generating functions are polynomials of maximum degree q. This nondiscrimination of regular sets being open or closed avoids a loss of generality incurred by the r-sets.

To equip $\mathbb{S}^{\mathrm{D}}$ with a metric, we define the *volume of a regular semianalytic set* $\mathcal{S} \in \mathbb{S}^{\mathrm{D}}$ to be the absolute value of the integral of the constant one over $\mathcal{S}$,

$$\|\mathcal{S}\| := \left| \int_{\mathcal{S}} \mathrm{d}\mathbf{x} \right|, \tag{2.1}$$

which is well defined because the generating functions of $\mathcal{S}$ are analytic. The analysis in Section 4 relies essentially on the fact that

$$\forall \mathcal{S} \in \mathbb{S}^{\mathrm{D}} \qquad \mathcal{S} \oplus \mathcal{S} = \emptyset, \qquad \emptyset \oplus \mathcal{S} = \mathcal{S}, \tag{2.2}$$

where "$\oplus$" denote the *symmetric difference* or *exclusive disjunction* "$\oplus$" defined in equation (1.2) of the supplement. It follows from (2.1) and (2.2) that

$$\forall \mathcal{S}, \mathcal{Q} \in \mathbb{S}^{\mathrm{D}}, \qquad \|\mathcal{S} \oplus \mathcal{Q}\| \leq \|\mathcal{S}\| + \|\mathcal{Q}\|. \tag{2.3}$$

Define a metric $d : \mathbb{S}^{\mathrm{D}} \times \mathbb{S}^{\mathrm{D}} \to \mathbb{R}$ as $d(\mathcal{S}, \mathcal{Q}) := \|\mathcal{S} \oplus \mathcal{Q}\|$. It follows from (2.1), (2.2), and (2.3) that $(\mathbb{S}^{\mathrm{D}}, d)$ is a metric space. Equipped with Boolean algebras, this metric space $(\mathbb{S}^{\mathrm{D}}, d)$ satisfies the three requirements listed in the opening paragraph of this section and is used for modeling physically meaningful material regions in MARS.

[2]The boundary of a bounded subset of $\mathbb{R}^{\mathrm{D}}$ is uniquely determined from the given set, but the converse is not true; cf. the well-known example of the lakes of Wada [53, p. 60]. In other words, two or more bounded subsets of $\mathbb{R}^{\mathrm{D}}$ may share the same boundary. Indeed, unambiguity is a basic principle for representing a solid via its boundary.

[3]Regular closed sets are not appropriate for defining the initial loci of the fluxing Lagrangian particles because the boundary points do not cross the fixed curve *properly* [61]. In other word, the flux sets are always open because the proper-intersection requirement excludes the boundary points.

3. Flow Maps: Unitary Operations on the Modeling Space. So far our mathematical space only models stationary material regions. Their movements are governed by a nonautonomous ordinary differential equation (ODE) of the form

$$\frac{\mathrm{d}\,\mathbf{x}}{\mathrm{d}\,t} = \mathbf{u}(\mathbf{x}, t). \tag{3.1}$$

According to the classic ODE theory, (3.1) admits a unique solution for any given initial time and position as long as $\mathbf{u}$ is continuous in time and Lipschitz continuous in space. This uniqueness furnishes an exact flow map ϕ and we briefly review it in Section 3.1. In Section 3.2, we approximate the exact flow map in time by a semidiscrete flow map, which is in turn approximated by a discrete flow map in space. In Section 3.3, the actions of these flow maps on the metric space $(\mathbb{S}^{\mathrm{D}}, d)$ are examined as a preparation of MARS in Section 4.

3.1. The exact flow map ϕ. The unique solution of (3.1) for a given initial time and position furnishes a *flow map* $\phi : \mathbb{R}^{\mathrm{D}} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{\mathrm{D}}$,

$$\phi_{t_0}^{+\tau}(p(t_0)) := p(t_0 + \tau); \qquad \phi_{t_0}^{-\tau}(p(t_0)) := p(t_0 - \tau), \tag{3.2}$$

where the three independent variables t_0, $p(t_0)$, and τ are the initial time, initial position at time t_0, and time increment, respectively; $p(t_0 + \tau)$ is the value of a solution to (3.1) whose value at time t_0 is $p(t_0)$. The "+" and "−" superscript signs denote forward and backward tracing respectively. When the initial time t_0 is given, each flow map with a fixed τ is a *homeomorphism*, i.e., a continuous bijective function with continuous inverse, and the set of flow maps $\Xi_{t_0} = \left\{\phi_{t_0}^{\sigma} : \sigma \in \mathbb{R}\right\}$ forms a *one-parameter group of diffeomorphisms* [4, p. 6]. For convenience we will also use $\overrightarrow{\bullet}$ and $\overleftarrow{\bullet}$ as synonymous notation for $\phi_{t_0}^{+\tau}$ and $\phi_{t_0}^{-\tau}$,

$$\begin{cases} \overrightarrow{p(t_0, \tau)} := \phi_{t_0}^{+\tau}(p(t_0)) = p(t_0) + \int_{t_0}^{t_0+\tau} \mathbf{u}(p(t), t)\,\mathrm{d}\,t; \\ \overleftarrow{p(t_0, \tau)} := \phi_{t_0}^{-\tau}(p(t_0)) = p(t_0) - \int_{t_0-\tau}^{t_0} \mathbf{u}(p(t), t)\,\mathrm{d}\,t, \end{cases} \tag{3.3}$$

where the time increment τ is always positive. For a fixed initial time t_0 and a fixed initial position $p(t_0)$, (3.3) and the second fundamental theorem of calculus yield

$$\left.\frac{\partial \phi_{t_0}^{\tau}(p(t_0))}{\partial \tau}\right|_{\tau=k} = \mathbf{u}(p(t_0 + k), t_0 + k). \tag{3.4}$$

Then the Taylor expansion of $\phi_{t_0}^{k}(p(t_0))$ in time can be written as

$$\phi_{t_0}^{k}(p(t_0)) = p(t_0) + \sum_{i=0}^{\infty} \frac{k^{i+1}}{i!} \left.\frac{\partial^i \mathbf{u}(p(t_0 + \tau), t_0 + \tau)}{\partial \tau^i}\right|_{\tau=0}, \tag{3.5}$$

which can also be derived by a Taylor expansion of the velocity inside the integral in the first identity of (3.3).

In general, the flow maps also apply to a point set $\mathcal{S}$:

$$\phi_{t_0}^{\pm\tau}(\mathcal{S}) := \left\{\phi_{t_0}^{\pm\tau}(q) : q(t_0) \in \mathcal{S}\right\}, \tag{3.6}$$

where $\phi_{t_0}^{+\tau}(\mathcal{S})$ and $\phi_{t_0}^{-\tau}(\mathcal{S})$ are, respectively, the *image* and *preimage* of $\mathcal{S}$; they are both *homeomorphic* to $\mathcal{S}$,

$$\mathcal{S} \cong \phi_{t_0}^{+\tau}(\mathcal{S}) \cong \phi_{t_0}^{-\tau}(\mathcal{S}), \tag{3.7}$$

because each flow map with fixed τ is a homeomorphism. Consequently, it is also a unitary operation on $\mathbb{S}^{\mathrm{D}}$ and a *homomorphism* with respect to the binary Boolean operations. In other words, the relative structure of two regular semianalytic sets is preserved by the exact flow map, e.g.

$$\phi_{t_0}^{\pm\tau}(\mathcal{P} \oplus \mathcal{Q}) = \phi_{t_0}^{\pm\tau}(\mathcal{P}) \oplus \phi_{t_0}^{\pm\tau}(\mathcal{Q}). \tag{3.8}$$

3.2. The semidiscrete flow map $\mathring{\phi}$ and the discrete flow map $\mathring{\varphi}$. Approximating an exact flow map with a numerical time integrator leads to the following.

DEFINITION 3.1. *A* semidiscrete flow map $\mathring{\phi} : \mathbb{S}^{\mathrm{D}} \to \mathbb{S}^{\mathrm{D}}$ *is an approximation of the exact flow map ϕ with a consistent time integrator on (3.1). $\mathring{\phi}_{t_0}^{nk}$ is* κth-order accurate *up to time T if $\mathring{\phi}_{t_0}^{nk}(p) = \phi_{t_0}^{nk}(p) + O(k^{\kappa})$ for all $p(t_0) \in \mathbb{R}^{\mathrm{D}}$ and $t_0 + nk \le T$.*

Due to the consistency, $\mathring{\phi}$ is also a homeomorphism as $k \to 0$. Hence a semidiscrete flow map commutes with the boundary operation of a point set,

$$\partial(\mathring{\phi}\mathcal{P}) = \mathring{\phi}(\partial\mathcal{P}), \tag{3.9}$$

and it also commutes with Boolean operations.

PROPOSITION 3.2. *For all $\mathcal{P}, \mathcal{Q} \in \mathbb{S}^{\mathrm{D}}$ and sufficiently small time increments, a semidiscrete flow map satisfies*

$$\mathring{\phi}(\mathcal{P} \oplus \mathcal{Q}) = \mathring{\phi}(\mathcal{P}) \oplus \mathring{\phi}(\mathcal{Q}). \tag{3.10}$$

The exact flow maps $\Xi_{t_0} = \left\{\phi_{t_0}^{\sigma} : \sigma \in \mathbb{R}\right\}$ form a group of diffeomorphisms. This also holds for semidiscrete flow maps if we make the following assumption.

ASSUMPTION 3.3 (Semi-discrete flow maps form a group). *For given $t_0, k \in \mathbb{R}$, denote $\mathring{\Phi}_{t_0}(k) = \{\mathring{\phi}_{t_0}^{nk} : n \in \mathbb{Z}\}$. The algebra $(\mathring{\Phi}_{t_0}(k), \circ, ^{-1}, \mathring{\phi}_{t_0}^{0})$ is assumed to be a group, where "$\circ$" denotes function composition, $\left(\mathring{\phi}_{t_0}^{nk}\right)^{-1} = \mathring{\phi}_{t_0}^{-nk}$, and $\mathring{\phi}_{t_0}^{0}$ is an identity map.*

In particular, for all $m, n \in \mathbb{Z}$,

$$\mathring{\phi}_{t_0}^{(m+n)k} = \mathring{\phi}_{t_0+mk}^{nk} \circ \mathring{\phi}_{t_0}^{mk}. \tag{3.11}$$

As an example, let $\mathbf{v}$ be a temporally continuous vector field and ϵ_u a small positive real number. Approximating the exact velocity field $\mathbf{u} = (1 + \epsilon_u)\mathbf{v}$ with $\mathring{\mathbf{u}} = \mathbf{v}$, the semidiscrete flow maps $\mathring{\phi}$ determined by $\mathring{\mathbf{u}}$ clearly satisfy Assumption 3.3. See Figure 3.1 for another illustration of semidiscrete flow maps.

For $\mathcal{P} \in \mathbb{S}_q^{\mathrm{D}}$, a semidiscrete flow map $\mathring{\phi}$ acts on all of its boundary points, hence it might not be possible to exactly represent $\mathring{\phi}(\partial\mathcal{P})$ with a finite number of polynomials. This motivates the notion of discrete flow maps.

DEFINITION 3.4. *A* discrete flow map $\mathring{\varphi} : \mathbb{S}_q^{\mathrm{D}} \to \mathbb{S}_q^{\mathrm{D}}$ *is an approximation of a semidiscrete flow map $\mathring{\phi}$ by two steps:*

(DFM-1) apply $\mathring{\phi}$ to each vertex (0-simplex) of $\mathcal{P}$;

(DFM-2) construct a polynomial approximation of $\mathring{\phi}(\partial\mathcal{P})$ that is homeomorphic to $\partial\mathcal{P}$. For $q = 1$, the adjacent vertices are connected by linear edges/triangles.

As illustrated in Figure 3.2, there are a number of differences between the semidiscrete flow map and the discrete flow map:

- $\mathring{\phi}$ acts on regular semianalytic sets while $\mathring{\varphi}$ acts on regular semialgebraic sets;

- the temporal mapping in $\mathring{\phi}$ is applied to *all* boundary points while that of $\mathring{\varphi}$ to only a *finite* number of boundary points;
- by Definition 3.1 and (3.9), $\mathring{\phi}$ is a unitary operation on $\mathbb{S}^{\mathrm{D}}$, whereas, in order for $\mathring{\varphi}$ to qualify for a unitary operation on $\mathbb{S}^{\mathrm{D}}$, step (DFM-2) in Definition 3.4 is indispensable;
- for $\mathcal{P} \in \mathbb{S}^{\mathrm{D}}_q$, $\mathring{\varphi}(\mathcal{P})$ depends on the particularities of the discrete representation (such as details of the reconstructing polynomials) while $\mathring{\phi}(\mathcal{P})$ does not.

If the time increment is sufficiently small, we have the homeomorphisms

$$\mathcal{P} \cong \mathring{\varphi}^k_{t_0}(\mathcal{P}) \cong \mathring{\phi}^k_{t_0}(\mathcal{P}) \cong \phi^k_{t_0}(\mathcal{P}), \tag{3.12}$$

since ϕ, $\mathring{\phi}$, and $\mathring{\varphi}$ share the same velocity field.

3.3. The actions of flow maps on regular semianalytic sets. Let $\|\partial\mathcal{P}\|$ denote the total length and total area of $\partial\mathcal{P}$ in 2D and 3D, respectively. For each $\mathcal{P} \in \mathbb{S}^{\mathrm{D}}$, the fact that it is bounded and regular implies

$$\|\partial\mathcal{P}\| = O(\|\mathcal{P}\|), \tag{3.13}$$

where $\|\mathcal{P}\|$ is defined in (2.1).

PROPOSITION 3.5. *The flow map $\phi : \mathbb{S}^{\mathrm{D}} \to \mathbb{S}^{\mathrm{D}}$ with fixed initial time t_0 satisfies*

$$\forall \mathcal{P} \in \mathbb{S}^{\mathrm{D}}, \qquad \left\|\phi^k_{t_0}(\mathcal{P})\right\| = \|\mathcal{P}\|(1 + O(k)). \tag{3.14}$$

In particular, for incompressible flows,

$$\nabla \cdot \mathbf{u} = 0 \;\Rightarrow\; \|\phi^{\pm k}_{t_0}(\mathcal{P})\| = \|\mathcal{P}\|. \tag{3.15}$$

Proof. Define the value of a color function $g(\mathbf{x}, t)$ to be 1 if $\mathbf{x}$ is in the closure of $\mathcal{P}(t)$ and 0 otherwise; here $\mathcal{P}(t)$ is the image of $\mathcal{P} = \mathcal{P}(t_0)$ under $\phi^{t-t_0}_{t_0}$. Clearly, the function $v_{\mathcal{P}}(t) := \|\mathcal{P}(t)\|$ depends on time only. The Reynolds transport theorem then yields

$$\frac{\mathrm{d}v_{\mathcal{P}}(\tau)}{\mathrm{d}\tau} = \frac{\mathrm{d}}{\mathrm{d}\tau}\int_{\mathcal{P}} g(\mathbf{x},\tau)\mathrm{d}\mathbf{x} = \int_{\mathcal{P}} \frac{\partial g(\mathbf{x},\tau)}{\partial\tau}\mathrm{d}\mathbf{x} + \int_{\partial\mathcal{P}} g(s,\tau)\mathbf{u}\cdot\mathbf{n}(s,\tau)\mathrm{d}s, \tag{3.16}$$

where $\mathcal{P}$ depends on τ and $\mathbf{n}$ is the normal vector of $\partial\mathcal{P}$. The first right-hand-side term of (3.16) is clearly zero. The divergence theorem and the solenoidal velocity then yield (3.15). For both incompressible and compressible flows, we have

$$v_{\mathcal{P}}(t_0 + k) - v_{\mathcal{P}}(t_0) = \int_{t_0}^{t_0+k}\int_{\partial\mathcal{P}(\tau)} \mathbf{u}\cdot\mathbf{n}(s,\tau)\mathrm{d}s\mathrm{d}\tau = O(k)\|\partial\mathcal{P}\|, \tag{3.17}$$

where the first step follows from integrating (3.16) over $[t_0, t_0 + k]$ and the second step from the mean value theorem for integrals. Then (3.13) yields (3.14). ∎

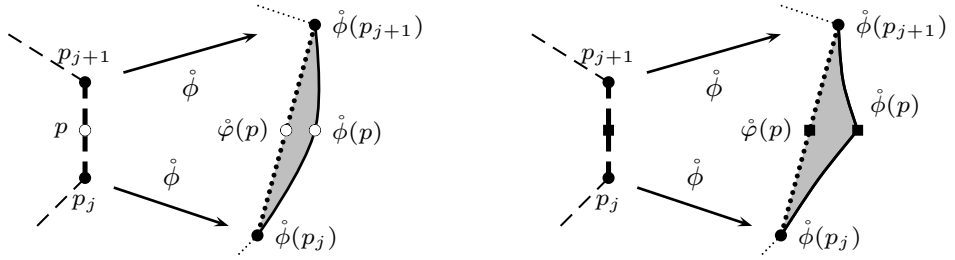
PROPOSITION 3.6. *A κth-order accurate semidiscrete flow map $\mathring{\phi}$ satisfies*

$$\forall \mathcal{P} \in \mathbb{S}^{\mathrm{D}}, \qquad \left\|\mathring{\phi}^k_{t_n}(\mathcal{P}) \oplus \phi^k_{t_n}(\mathcal{P})\right\| \le O\left(k^{\kappa+1}\right). \tag{3.18}$$

Proof. The truncation error of $\mathring{\phi}$ is $O(k^{\kappa+1})$ for mapping any point $p \in \partial\mathcal{P}$. By (3.12), there is a one-to-one correspondence between the boundary points of $\mathring{\phi}^k_{t_n}(\mathcal{P})$

(a) A smooth image (b) A image with kinks

FIG. 3.1. *The actions of exact and semidiscrete flow maps on a semianalytic set. The dashed curve represents the boundary of a semianalytic set $\mathcal{P}$, the thick solid curve $\phi(\partial\mathcal{P})$, and the thin solid curve $\mathring{\phi}(\partial\mathcal{P})$. The one-to-one correspondence between points of $\phi(\partial\mathcal{P})$ and those of $\mathring{\phi}(\partial\mathcal{P})$ is represented by the two "$\circ$" dots. In subplot (b), the solid squares represent dynamically generated kinks on the mapped boundaries. For sufficiently small time step sizes, the semidiscrete flow map preserves the feature of the exact flow map concerning the generation of dynamic kinks.*



(a) A smooth image (b) A image with kinks

FIG. 3.2. *The difference between the actions of semidiscrete and discrete flow maps on linear semialgebraic sets. The dashed lines represent part of the boundary of a linear semialgebraic set $\mathcal{P}$, the thick dashed line an edge $\overline{p_j p_{j+1}} \subset \partial\mathcal{P}$. The discrete flow map $\mathring{\varphi}$ first uses the semidiscrete flow map $\mathring{\phi}$ to send p_j to $\mathring{\phi}(p_j)$ for all j's, then connect adjacent vertices with linear segments (dotted lines). In contrast, the semidiscrete flow map $\mathring{\phi}$ acts on all points of $\partial\mathcal{P}$. In particular, the thick dotted line represents $\mathring{\varphi}(\overline{p_j p_{j+1}})$, the solid curve $\mathring{\phi}(\overline{p_j p_{j+1}})$, and the shaded area the symmetric difference between $\mathring{\varphi}(\mathcal{P})$ and $\mathring{\phi}(\mathcal{P})$ contributed by the edge $\overline{p_j p_{j+1}}$.*

and those of $\phi_{t_n}^k(\mathcal{P})$. As shown in Figure 3.1, the left-hand side of (3.18) can be written as an integral similar to that in (3.17). Applying the integral mean value theorem and (3.13) completes the proof. ∎

COROLLARY 3.7. *A semidiscrete flow map $\mathring{\phi}$ satisfies*

$$\forall \mathcal{P} \in \mathbb{S}^{\mathrm{D}}, \qquad \left\| \mathring{\phi}_{t_n}^k(\mathcal{P}) \right\| \le \|\mathcal{P}\|(1 + O(k)). \tag{3.19}$$

Proof. By (2.2) and (2.3), we have

$$\left\| \mathring{\phi}_{t_n}^k(\mathcal{P}) \right\| = \left\| \mathring{\phi}_{t_n}^k(\mathcal{P}) \oplus \phi_{t_n}^k(\mathcal{P}) \oplus \phi_{t_n}^k(\mathcal{P}) \right\| \le \left\| \mathring{\phi}_{t_n}^k(\mathcal{P}) \oplus \phi_{t_n}^k(\mathcal{P}) \right\| + \left\| \phi_{t_n}^k(\mathcal{P}) \right\|.$$

Then Propositions 3.5 and 3.6 complete the proof. ∎

Proposition 3.5 and Corollary 3.7 state that the volume change of a point set under either a flow map or a semidiscrete flow map is proportional to the time increment;

the difference of these two maps is quantified in Proposition 3.6. After characterizing regular semialgebraic sets with a length scale in Definition 3.8, we quantify the mapping difference between a semidiscrete flow map and a discrete flow map in Lemma 3.9.

DEFINITION 3.8. *The* Lagrangian length scale of a semialgebraic set $\mathcal{P} \in \mathbb{S}_q^{\mathrm{D}}$ *is the maximum distance between the two endpoints of all edges, i.e.*

$$h_L(\mathcal{P}) := \max_{\widetilde{\mathbf{xy}} \subset \partial \mathcal{P}} d_2(\mathbf{x}, \mathbf{y}), \tag{3.20}$$

where $\mathbf{x}, \mathbf{y}$ *are vertices (0-simplices) of* $\mathcal{P}$, $\widetilde{\mathbf{xy}}$ *an edge (1-simplex) of* $\mathcal{P}$, *and* $d_2(\mathbf{x}, \mathbf{y})$ *the Euclidean distance between* $\mathbf{x}$ *and* $\mathbf{y}$.

LEMMA 3.9. *Consider the approximation of a semidiscrete flow map* $\mathring{\phi}$ *with a discrete flow map* $\mathring{\varphi} : \mathbb{S}_1^{\mathrm{D}} \to \mathbb{S}_1^{\mathrm{D}}$. *If* $\mathring{\phi}$ *is* κ*th-order accurate with* $\kappa \geq 2$, *then, for each* $\mathcal{P} \in \mathbb{S}_1^{\mathrm{D}}$ *(*$\mathrm{D} = 2, 3$*),*

$$\left\| \mathring{\varphi}_{t_n}^k(\mathcal{P}) \oplus \mathring{\phi}_{t_n}^k(\mathcal{P}) \right\| \leq O\left(k h_{L,\mathcal{P}}^2 + k^{\kappa+1}\right). \tag{3.21}$$

Proof. For $\mathrm{D} = 2$, let $\overline{p_1 p_2}$ denote an edge of $\partial\mathcal{P}$. By Definition 3.8, we have $d_2(p_1, p_2) \leq h_L$. A point $p \in \overline{p_1 p_2}$ can be expressed, for some $a \in [0, 1]$, as

$$p = a p_1 + (1 - a) p_2. \tag{3.22}$$

Write $\mathbf{u}^{(m)} := \frac{\partial^m \mathbf{u}}{\partial t^m}$ with $m \geq 0$ and $\mathbf{u}^{(0)} := \mathbf{u}$. Define a linear combination

$$\Delta_p^{(m)} := a\mathbf{u}^{(m)}(p_1, t) + (1 - a)\mathbf{u}^{(m)}(p_2, t) - \mathbf{u}^{(m)}(p, t). \tag{3.23}$$

Then, for any fixed time, Taylor expansions of $\mathbf{u}^{(m)}(p_1)$ and $\mathbf{u}^{(m)}(p_2)$ at p lead to

$$\Delta_p^{(m)} = O(h_L^2), \tag{3.24}$$

where we have applied (3.22). As shown in Figure 3.2, it follows that

$$\begin{aligned}
\mathring{\varphi}_{t_n}^k(p) - \mathring{\phi}_{t_n}^k(p) &= a\mathring{\phi}_{t_n}^k(p_1) + (1 - a)\mathring{\phi}_{t_n}^k(p_2) - \mathring{\phi}_{t_n}^k(p) \\
&= a\phi_{t_n}^k(p_1) + (1 - a)\phi_{t_n}^k(p_2) - \phi_{t_n}^k(p) + O(k^{\kappa+1}) \\
&= a p_1 + (1 - a) p_2 - p + \textstyle\sum_{i=0}^{\kappa} \frac{k^{i+1}}{i!} \Delta_p^{(i)} + O(k^{\kappa+1}) \\
&= O(k h_L^2) + O(k^{\kappa+1}),
\end{aligned}$$

where the first step follows from (3.22) and Definition 3.4, the second step from the assumption of $\mathring{\phi}$ being κth-order accurate, the third step from (3.5) and (3.23), and the last step from (3.22) and (3.24). By (3.12), there is a one-to-one correspondence between the boundary points of $\mathring{\varphi}_{t_n}^k(\mathcal{P})$ and those of $\mathring{\phi}_{t_n}^k(\mathcal{P})$. Hence the left-hand side of (3.21) can be written as an integral similar to that in (3.17). Applying the integral mean value theorem and (3.13) yields (3.21).

The above arguments also apply to 3D in a straightforward way. ∎

Interestingly, the above proof suggests that Lemma 3.9 holds regardless of dynamically generated kinks. On the other hand, any method that does not explicitly detect the dynamic kinks and account for the discontinuities cannot do better than second-order accuracy. In particular, using a piecewise cubic polynomial representation instead of the piecewise linear one does not give accuracy higher than $O(h_L^2)$ if dynamic kinks develop.

For *any* numerical simulation performed on computers, the underlying data structure has to be *finite* and it must finish in a *finite* number of steps. Hence in practice the semidiscrete flow map must be approximated by a discrete flow map. In answering the question (Q-4) posed in Section 1, Lemma 3.9 pinpoints a fundamental difficulty in achieving an IT accuracy higher than the second order: *the accumulated error of approximating the semidiscrete flow map is always proportional to the square of the Lagrangian length scale h_L; this is independent of the polynomial representation of the boundary of material regions.* This difficulty, due to the dynamically generated kinks on the interface, is the main reason that VOF methods with parabolic and cubic reconstruction are still second-order accurate.

One way to overcome this fundamental difficulty is to set the Lagrangian length scale h_L to be a power function of the Eulerian grid size h. If this can be done without significantly increasing the computational cost, the result is an efficient and effectively higher-order method. The iPAM method is such an example, as shown in Section 6.

4. The MARS Framework. Based on the modeling space $(\mathbb{S}^{\mathrm{D}}, d)$ and the flow maps, we define the IT problem and the accuracy of IT methods in Sections 4.1 and 4.2. Then we propose in Section 4.3 a generic IT method as the concatenation of three unitary operations on $\mathbb{S}^{\mathrm{D}}_q$ and show in Section 4.4 that its overall error can be bounded by the sum of the individual error terms. In Section 4.5, we estimate these individual errors in terms of the time step size k and the Lagrangian length scale h_L. In particular, the adjustment error of a special family of MARS methods is examined in Section 4.6.

4.1. Defining the IT problem. A *color function* can be defined for the tracked material M as

$$f(\mathbf{x},t) := \begin{cases} 1 & \text{if there is } \mathsf{M} \text{ at } (\mathbf{x},\mathrm{t}), \\ 0 & \text{otherwise,} \end{cases} \tag{4.1}$$

and f satisfies the advection equation

$$\frac{\partial f}{\partial t} + \mathbf{u}\cdot\nabla f = 0. \tag{4.2}$$

In terms of the color function, we give the following point-set formulation of the IT problem.

Definition 4.1 (the IT problem). *Denote the area occupied by the tracked material M at time t by*

$$\mathcal{M}(t) := \{\mathbf{x} : f(\mathbf{x},t) = 1\}. \tag{4.3}$$

The IT problem *is the determination of $\mathcal{M}(T)$ from the initial condition $\mathcal{M}(t_0)$ and the given velocity field $\mathbf{u}(\mathbf{x},t)$, $t \in [t_0, T]$.*

The initial condition $\mathcal{M}(t_0)$ in the above definition is modeled as a regular semianalytic set and discretely represented as a regular semialgebraic set; both may be either open or closed. As mentioned in Section 2, this modeling choice does not incur any loss of generality for physically meaningful material regions. Since the flow maps are diffeomorphisms, $\mathcal{M}(t)$ belongs to $\mathbb{S}^{\mathrm{D}}$ for any subsequent time instance. In addition, the definitions of regular semianalytic sets and their volume (2.1) ensure that consistent IT is possible by excluding pathological regular sets such as one whose boundary is nowhere differentiable.

An alternative approach to the IT problem makes use of the tracked material's mass density $\rho(\mathbf{x}, t)$ instead of the color function f. In this approach, the area occupied by the material M is defined as $\mathcal{M}_\rho := \{\mathbf{x} : \rho(\mathbf{x}, t) > 0\}$, where ρ satisfies the scalar conservation law $\frac{\partial \rho}{\partial t} + \nabla \cdot (\mathbf{u}\rho) = 0$. However, in compressible flows the latter approach is susceptible to numerical difficulties when the velocity field dilutes the density of the tracked material to be close to zero. Furthermore, from a physical viewpoint of IT, *the type of the tracked material is a uniform and steady property of each Lagrangian particle and should be independent of any properties of the underlying flow*, a property which is captured by the color function. Another convenience of our modeling choice is that (4.1) yields a natural definition of volume fraction in (4.4). The above discussion justifies our modeling choice of the advection equation (4.2) of the color function (4.1).

4.2. Measuring the accuracy of IT methods. Partition the computational domain Ω into a collection of fixed control volumes. The region occupied by the tracked material inside a cell $\mathcal{C}$ is $\mathcal{M}_\mathcal{C} = \mathcal{M} \cap \mathcal{C}$. The *volume fraction* of M for cell $\mathcal{C}$ is then

$$\langle f(t) \rangle_\mathcal{C} := \frac{1}{\|\mathcal{C}\|} \int_\mathcal{C} f(\mathbf{x}, t)\, \mathrm{d}\mathbf{x} = \frac{\|\mathcal{M}_\mathcal{C}\|}{\|\mathcal{C}\|}. \tag{4.4}$$

In most VOF methods, the IT error at time t_n is measured by the geometric error

$$E_g(t_n) := \sum_{\mathcal{C} \subset \Omega} \|\mathcal{C}\| \left| \langle f(t_n) \rangle_\mathcal{C} - \langle f \rangle_\mathcal{C}^n \right|, \tag{4.5}$$

where $\langle f(t_n) \rangle_\mathcal{C}$ and $\langle f \rangle_\mathcal{C}^n$ are, respectively, the exact and the computed volume fraction of cell $\mathcal{C}$. A different measure is adopted in [13, 2, 57] as

$$E_1(t_n) := \|\mathcal{M}(t_n) \oplus \mathcal{M}^n\| = \sum_{\mathcal{C} \subset \Omega} \|\mathcal{M}_\mathcal{C}(t_n) \oplus \mathcal{M}_\mathcal{C}^n\|, \tag{4.6}$$

where $\mathcal{M}(t_n)$ denotes the exact material region at t_n and $\mathcal{M}^n$ its approximation by an IT method.

The measure (4.6) is more rigorous than (4.5) for three reasons. First, E_1 is a metric on $\mathbb{S}^\mathrm{D}$ as is discussed in Section 2. In contrast, E_g is not a metric: $E_g = 0$ does not necessarily imply exact IT. Second, E_1 does not depend on the partition of Ω while E_g does. Lastly, E_g tends to hide reconstruction errors of VOF methods [57, sect. 3.6].

Definition 4.2 (Accuracy of an IT method). *In numerically solving the IT problem posed in Definition 4.1, an IT method* $\mathcal{L}_\mathrm{IT}$,

$$\mathcal{M}^{n+1} = \mathcal{L}_\mathrm{IT}\big(\mathcal{M}(t_n), \mathbf{u}\big) \approx \mathcal{M}(t_{n+1}), \tag{4.7}$$

is of the β*th-order accuracy in the 1-norm if* $E_1(T) = O(k^\beta)$ *as the time step size* k *approaches zero.* $\mathcal{L}_\mathrm{IT}$ *is said to be* consistent *if* $\beta > 0$.

In practice, the calculation of (4.6) tends to be ill-conditioned. Meanwhile, for any regular semianalytic set $\mathcal{M}$, $E_g \to E_1$ as the cell volume and time step size approach zero. Hence we use (4.6) for theoretical analysis and (4.5) for numerical experiments.

4.3. A generic IT method. Conceptually one could simply apply the discrete flow map in Definition 3.4 to a discrete representation of the initial condition to obtain a result of IT. In practice, this seldom works because two adjacent markers on an interface might become far apart after being stretched by the velocity field, which

deteriorates the IT accuracy. This can be prevented by concatenating the discrete flow map with preprocessing and postprocessing operations that maintain certain desired properties of the interface. As unitary operations on $\mathbb{S}_q^{\mathrm{D}}$, the preprocessing and postprocessing procedures also render the method generic.

DEFINITION 4.3. *A* MARS method *is an IT method of the form*

$$\mathcal{M}^{n+1} = \mathfrak{L}_{\mathrm{Mars}}^n \mathcal{M}^n := \left(\chi_{n+1} \circ \varphi_{t_n}^k \circ \psi_n\right) \mathcal{M}^n, \tag{4.8}$$

where $\mathcal{M}(t_n) \in \mathbb{S}^{\mathrm{D}}$ *is approximated by* $\mathcal{M}^n \in \mathbb{S}_q^{\mathrm{D}}$ *for some* $q \in \mathbb{N}^+$, $\varphi : \mathbb{S}_q^{\mathrm{D}} \to \mathbb{S}_q^{\mathrm{D}}$ *is a* mapping operation *that approximates the exact flow map* ϕ *both in time and in space by sending one semialgebraic set at* t_n *to another at* $t_n + k$, $\psi_n : \mathbb{S}_q^{\mathrm{D}} \to \mathbb{S}_q^{\mathrm{D}}$ *an* augmentation operation *at* t_n *to prepare* $\mathcal{M}^n$ *for the mapping, and* $\chi_{n+1} : \mathbb{S}_q^{\mathrm{D}} \to \mathbb{S}_q^{\mathrm{D}}$ *an* adjustment operation *after the mapping to fulfill some representation invariants (either abstract or concrete) of* $\mathcal{M}^{n+1}$.

For example, an augmentation operation could add more vertices to $\mathcal{M}^n$ so that the Lagrangian length scale $h_L(\widetilde{\mathcal{M}}^{n+1})$ does not exceed a prescribed threshold for

$$\widetilde{\mathcal{M}}^{n+1} := \left(\varphi_{t_n}^k \circ \psi_n\right) \mathcal{M}^n; \tag{4.9}$$

a mapping operation φ could be the discrete flow map in Definition 3.4; an adjustment operation could enforce mass conservation by changing $\widetilde{\mathcal{M}}^{n+1}$ so that the volume of $\mathcal{M}^{n+1}$ equals that of $\mathcal{M}^n$. It follows from Definition 4.3 that

$$\mathcal{M}^{n+1} = \mathfrak{L}_{\mathrm{Mars}}^n \cdots \mathfrak{L}_{\mathrm{Mars}}^0 \mathcal{M}^0 = \left(\chi_{n+1} \circ \varphi_{t_n}^k \circ \psi_n\right) \cdots \left(\chi_1 \circ \varphi_{t_0}^k \circ \psi_0\right) \mathcal{M}^0. \tag{4.10}$$

4.4. Dividing the overall error into individual terms. The error of a MARS method is studied via a "divide-and-conquer" approach. First, the exact and calculated solutions of the IT problem are linked by a number of "stepping-stone" solutions: the expressions of every two adjacent solutions differ only by a single element and the symmetric difference of adjacent pairs leads to error definitions in Definition 4.4. Then the overall IT error of a MARS method is bounded by the sum of the individual error terms in Theorem 4.5.

DEFINITION 4.4. *Consider a MARS method of which the mapping operation* φ *is obtained from the exact flow map* ϕ *by first discretizing* ϕ *in time (leading to a semidiscrete flow map* $\mathring{\phi}$*) and then discretizing* $\mathring{\phi}$ *in space. The* time-integration error $E^{\mathrm{ODE}}(t_n)$, *the* representation error $E^{\mathrm{REP}}(t_n)$, *the* augmentation error $E^{\mathrm{AUG}}(t_n)$, *the* mapping error $E^{\mathrm{MAP}}(t_n)$, *and the* adjustment error $E^{\mathrm{ADJ}}(t_n)$ *of such a MARS method at* $t_n = t_0 + nk$ *are, respectively,*

$$\begin{cases} E^{\mathrm{ODE}}(t_n) & := \left\| \phi_{t_0}^{nk}(\mathcal{M}^0) \oplus \mathring{\phi}_{t_0}^{nk}(\mathcal{M}^0) \right\|; \\ E^{\mathrm{REP}}(t_n) & := \left\| \phi_{t_0}^{nk}(\mathcal{M}(t_0)) \oplus \phi_{t_0}^{nk}(\mathcal{M}^0) \right\|; \\ \mathcal{E}_i^{\mathrm{AUG}} & := \left(\psi_i \mathcal{M}^i\right) \oplus \mathcal{M}^i, \\ E^{\mathrm{AUG}}(t_n) & := \left\| \bigoplus_{j=0}^{n-1} \mathring{\phi}_{t_j}^{(n-j)k} \mathcal{E}_j^{\mathrm{AUG}} \right\|; \\ \mathcal{E}_i^{\mathrm{MAP}} & := \mathring{\phi}_{t_i}^k(\psi_i \mathcal{M}^i) \oplus \varphi_{t_i}^k(\psi_i \mathcal{M}^i), \\ E^{\mathrm{MAP}}(t_n) & := \left\| \bigoplus_{j=1}^{n} \mathring{\phi}_{t_j}^{(n-j)k} \mathcal{E}_{j-1}^{\mathrm{MAP}} \right\|; \\ \mathcal{E}_{i+1}^{\mathrm{ADJ}} & := \left(\varphi_{t_i}^k \psi_i \mathcal{M}^i\right) \oplus \mathcal{M}^{i+1}, \\ E^{\mathrm{ADJ}}(t_n) & := \left\| \bigoplus_{j=1}^{n} \mathring{\phi}_{t_j}^{(n-j)k} \mathcal{E}_j^{\mathrm{ADJ}} \right\|. \end{cases} \tag{4.11}$$

$\mathcal{E}_i^{\mathrm{AUG}}$, $\mathcal{E}_i^{\mathrm{MAP}}$, $\mathcal{E}_{i+1}^{\mathrm{ADJ}}$ *are called the* error region*s of the ith time step,* $i = 0, \ldots, n-1$.

Here E^{ODE} is the error caused by approximating the exact flow map ϕ with a semidiscrete flow map $\mathring{\phi}$, E^{REP} the final IT error caused by representing the material region with a semialgebraic set at the initial time, E^{AUG} the accumulated error of augmenting the semialgebraic sets, E^{MAP} the accumulated error of approximating the semidiscrete flow map with the mapping operation, and E^{ADJ} the accumulated error of adjusting the mapped semialgebraic sets.

It follows from (3.8) and Proposition 3.5 that

$$E^{\mathrm{REP}} = \left\|\phi_{t_0}^{nk}\left(\mathcal{M}(t_0) \oplus \mathcal{M}^0\right)\right\| = \left\|\mathcal{M}(t_0) \oplus \mathcal{M}^0\right\|(1 + O(nk)), \tag{4.12}$$

which implies that *the representation error does not accumulate in time to a larger order of magnitude*. As for E^{ODE}, the temporal accumulation of integration errors caused by a semidiscrete flow map has been formally encapsulated within its definition. The forms of E^{AUG}, E^{MAP}, E^{ADJ} also suggest that they may accumulate over the time steps. However, error accumulations of E^{AUG}, E^{MAP}, E^{ADJ} are intrinsically different from that of E^{ODE}: the error regions $\mathcal{E}_i^{\mathrm{AUG}}$, $\mathcal{E}_i^{\mathrm{MAP}}$, $\mathcal{E}_i^{\mathrm{ADJ}}$ of different time steps are sent to the *same* final time by the semidiscrete flow maps while the accumulated atom errors in E^{ODE} are at *different* time instances. The implication of this difference will be further discussed in Sections 4.5 and 4.6.

The main result of this subsection is the following theorem:

THEOREM 4.5. *The IT error of a MARS method is bounded as*

$$E_1(t_n) \le E^{\mathrm{AUG}} + E^{\mathrm{ADJ}} + E^{\mathrm{MAP}} + E^{\mathrm{REP}} + E^{\mathrm{ODE}}, \tag{4.13}$$

where the individual error terms are defined in Definition 4.4.

Proof. Definition 4.3 and the definition of $E_1(t_n)$ in (4.6) lead to

$$\begin{aligned}
E_1(t_n) = &\ \left\|\left(\phi_{t_0}^{nk}\mathcal{M}(t_0)\right) \oplus \mathcal{M}^n\right\| \\
= &\ \left\|\phi_{t_0}^{nk}\mathcal{M}(t_0) \oplus \phi_{t_0}^{nk}\mathcal{M}^0 \oplus \phi_{t_0}^{nk}\mathcal{M}^0 \oplus \mathring{\phi}_{t_0}^{nk}\mathcal{M}^0 \oplus \mathring{\phi}_{t_0}^{nk}\mathcal{M}^0 \oplus \mathcal{M}^n\right\| \\
\le &\ E^{\mathrm{REP}} + E^{\mathrm{ODE}} + \left\|\left(\mathring{\phi}_{t_0}^{nk}\mathcal{M}^0\right) \oplus \mathcal{M}^n\right\|, \\
\le &\ E^{\mathrm{REP}} + E^{\mathrm{ODE}} + E^{\mathrm{AUG}} + E^{\mathrm{MAP}} + E^{\mathrm{ADJ}},
\end{aligned}$$

where the second step follows from (2.2), the third step from (2.3) and Definition 4.4, and the fourth step from Lemma 4.6 below, (2.3), and Definition 4.4. ∎

Theorem 4.5 is a direct consequence of Definitions 4.3 and 4.4. It depends neither on the representation details of semialgebraic sets nor on the particularities of the three operations in Definition 4.3.

LEMMA 4.6. *If Assumption 3.3 holds, then a MARS method satisfies*

$$\left(\mathring{\phi}_{t_0}^{nk}\mathcal{M}^0\right) \oplus \mathcal{M}^n = \left(\bigoplus_{j=0}^{n-1} \mathring{\phi}_{t_j}^{(n-j)k}\mathcal{E}_j^{\mathrm{AUG}}\right) \oplus \left(\bigoplus_{j=1}^{n} \mathring{\phi}_{t_j}^{(n-j)k}\mathcal{E}_{j-1}^{\mathrm{MAP}}\right) \oplus \left(\bigoplus_{j=1}^{n} \mathring{\phi}_{t_j}^{(n-j)k}\mathcal{E}_j^{\mathrm{ADJ}}\right). \tag{4.14}$$

Proof. Denote $\mathcal{M}_\psi^n = \psi_n\mathcal{M}^n$, then (4.9) and Definition 4.3 yield

$$\begin{cases} \widetilde{\mathcal{M}}^{n+1} &= \varphi_{t_n}^k\mathcal{M}_\psi^n, \\ \mathcal{M}^{n+1} &= \chi_{n+1}\widetilde{\mathcal{M}}^{n+1}. \end{cases} \tag{4.15}$$

We prove (4.14) by an induction on the number of time steps taken. The induction basis of $n = 1$ holds because

$$\begin{aligned}
&\left(\mathring{\phi}_{t_0}^k \mathcal{M}^0\right) \oplus \mathcal{M}^1 \\
= \;& \mathring{\phi}_{t_0}^k \mathcal{M}^0 \oplus \mathring{\phi}_{t_0}^k \mathcal{M}_\psi^0 \oplus \mathring{\phi}_{t_0}^k \mathcal{M}_\psi^0 \oplus \widetilde{\mathcal{M}}^1 \oplus \widetilde{\mathcal{M}}^1 \oplus \mathcal{M}^1 \\
= \;& \left(\mathring{\phi}_{t_0}^k \mathcal{M}^0 \oplus \mathring{\phi}_{t_0}^k \mathcal{M}_\psi^0\right) \oplus \left(\mathring{\phi}_{t_0}^k \mathcal{M}_\psi^0 \oplus \varphi_{t_0}^k \mathcal{M}_\psi^0\right) \oplus \left(\widetilde{\mathcal{M}}^1 \oplus \chi_1 \widetilde{\mathcal{M}}^1\right) \\
= \;& \mathring{\phi}_{t_0}^k \left(\mathcal{E}_0^{\mathrm{AUG}}\right) \oplus \mathcal{E}_0^{\mathrm{MAP}} \oplus \mathcal{E}_1^{\mathrm{ADJ}},
\end{aligned}$$

where the first step follows from (2.2), the second step from (4.15), the last step from (3.10) and Definition 4.4.

Assume (4.14) holds. Then for step $n+1$ we have

$$\begin{aligned}
&\mathring{\phi}_{t_0}^{(n+1)k} \mathcal{M}^0 \oplus \mathcal{M}^{n+1} \\
= \;& \mathring{\phi}_{t_0}^{(n+1)k} \mathcal{M}^0 \oplus \mathring{\phi}_{t_n}^k \mathcal{M}^n \oplus \mathring{\phi}_{t_n}^k \mathcal{M}^n \oplus \mathring{\phi}_{t_n}^k \mathcal{M}_\psi^n \oplus \mathring{\phi}_{t_n}^k \mathcal{M}_\psi^n \oplus \widetilde{\mathcal{M}}^{n+1} \oplus \widetilde{\mathcal{M}}^{n+1} \oplus \mathcal{M}^{n+1} \\
= \;& \mathring{\phi}_{t_n}^k \left(\mathring{\phi}_{t_n}^{nk} \mathcal{M}^0 \oplus \mathcal{M}^n\right) \oplus \mathring{\phi}_{t_n}^k \left(\mathcal{E}_n^{\mathrm{AUG}}\right) \oplus \mathcal{E}_n^{\mathrm{MAP}} \oplus \mathcal{E}_{n+1}^{\mathrm{ADJ}}, \\
= \;& \left(\bigoplus\nolimits_{j=0}^{n-1} \mathring{\phi}_{t_n}^k \mathring{\phi}_{t_j}^{(n-j)k} \mathcal{E}_j^{\mathrm{AUG}}\right) \oplus \left(\bigoplus\nolimits_{j=1}^{n} \mathring{\phi}_{t_n}^k \mathring{\phi}_{t_j}^{(n-j)k} \mathcal{E}_{j-1}^{\mathrm{MAP}}\right) \oplus \left(\bigoplus\nolimits_{j=1}^{n} \mathring{\phi}_{t_n}^k \mathring{\phi}_{t_j}^{(n-j)k} \mathcal{E}_j^{\mathrm{ADJ}}\right) \\
& \oplus \mathring{\phi}_{t_n}^k \mathcal{E}_n^{\mathrm{AUG}} \oplus \mathcal{E}_n^{\mathrm{MAP}} \oplus \mathcal{E}_{n+1}^{\mathrm{ADJ}} \\
= \;& \left(\bigoplus\nolimits_{j=0}^{n-1} \mathring{\phi}_{t_j}^{(n+1-j)k} \mathcal{E}_j^{\mathrm{AUG}}\right) \oplus \left(\bigoplus\nolimits_{j=1}^{n} \mathring{\phi}_{t_j}^{(n+1-j)k} \mathcal{E}_{j-1}^{\mathrm{MAP}}\right) \oplus \left(\bigoplus\nolimits_{j=1}^{n} \mathring{\phi}_{t_j}^{(n+1-j)k} \mathcal{E}_j^{\mathrm{ADJ}}\right) \\
& \oplus \mathring{\phi}_{t_n}^k \mathcal{E}_n^{\mathrm{AUG}} \oplus \mathcal{E}_n^{\mathrm{MAP}} \oplus \mathcal{E}_{n+1}^{\mathrm{ADJ}} \\
= \;& \left(\bigoplus\nolimits_{j=0}^{n+1-1} \mathring{\phi}_{t_j}^{(n+1-j)k} \mathcal{E}_j^{\mathrm{AUG}}\right) \oplus \left(\bigoplus\nolimits_{j=1}^{n+1} \mathring{\phi}_{t_j}^{(n+1-j)k} \mathcal{E}_{j-1}^{\mathrm{MAP}}\right) \oplus \left(\bigoplus\nolimits_{j=1}^{n+1} \mathring{\phi}_{t_j}^{(n+1-j)k} \mathcal{E}_j^{\mathrm{ADJ}}\right)
\end{aligned}$$

where the first step follows from (2.2); the second step from (4.15), (3.10), and Definition 4.4; the third step from the induction hypothesis and (3.10); the fourth step from Assumption 3.3; the last step from the commutativity of the exclusive disjunction operator. □

4.5. Estimating the individual error terms. According to Definition 4.3, a MARS method is unambiguously described by fully specifying the details of its three operations and the representation of $\mathcal{M}(t_n)$ by $\mathcal{M}^n$. After examining the representation error, we use the results in Section 3.3 to make the error bound in Theorem 4.5 more concrete.

PROPOSITION 4.7. *If a MARS method represents the initial material region with a linear semialgebraic set, i.e. $\mathcal{M}^0 \in \mathbb{S}_1^{\mathrm{D}}$ ($\mathrm{D} = 2, 3$), then $E^{\mathrm{REP}}(t_n) = O(h_L^2(\mathcal{M}^0))$.*

Proof. In 2D, all vertices of $\mathcal{M}^0$ clearly belong to $\mathcal{M}(t_0)$. For two adjacent vertices p_j and p_{j+1}, if the arc $\widetilde{p_j p_{j+1}} \subset \partial\mathcal{M}(t_0)$ is C^2, set up a local orthogonal coordinate system (x, y) with its x-axis aligned with the line segment $\overline{p_j p_{j+1}} \subset \partial\mathcal{M}^0$, $|y(x)|$ is then the distance between corresponding points of $\widetilde{p_j p_{j+1}}$ and $\overline{p_j p_{j+1}}$. Since $y_j = 0$, $y_{j+1} = 0$, Rolle's theorem implies that there exists a point $p \in \widetilde{p_j p_{j+1}}$ such that $\left.\frac{\mathrm{d}y}{\mathrm{d}x}\right|_p = 0$. Then it follows from a Taylor expansion of y at p that the maximum distance between $\widetilde{p_j p_{j+1}}$ and $\overline{p_j p_{j+1}}$ is $O(h_L^2)$. In comparison, this distance is $O(h_L)$ if $\widetilde{p_j p_{j+1}}$ contains a kink of $\partial\mathcal{M}(t_0)$. However, by definition of semianalytic sets, the number of kinks on $\partial\mathcal{M}(t_0)$ is $O(1)$, hence

$$\left\|\mathcal{M}(t_0) \oplus \mathcal{M}^0\right\| = O(1) h_L O(h_L) + O\left(\frac{1}{h_L}\right) h_L O(h_L^2) = O(h_L^2),$$

where $O(1)$ and $O\left(\frac{1}{h_L}\right)$ are the numbers of edges of the two cases.

In 3D, similar arguments yield

$$\left\|\mathcal{M}(t_0) \oplus \mathcal{M}^0\right\| = O(\frac{1}{h_L})h_L^2 O(h_L) + O\left(\frac{1}{h_L^2}\right) h_L^2 O(h_L^2) = O(h_L^2),$$

where h_L^2 is the order of the face areas; $O\left(\frac{1}{h_L^2}\right)$ and $O(\frac{1}{h_L})$ are, respectively, the number of smooth faces and that of C^1 discontinuous faces; $O(h_L)$ and $O(h_L^2)$ are the maximum distance between corresponding boundary points respectively for a smooth face and a kinked face. Therefore,

$$\left\|\phi_{t_0}^{nk}(\mathcal{M}(t_0)) \oplus \phi_{t_0}^{nk}(\mathcal{M}^0)\right\| = \left\|\phi_{t_0}^{nk}\left(\mathcal{M}(t_0) \oplus \mathcal{M}^0\right)\right\| = O(h_L^2)(1 + O(nk)) = O(h_L^2),$$

where the first step follows from (3.8) and the second step from Proposition 3.5. □

Clearly, Proposition 4.7 holds independently of whether all kinks of $\mathcal{M}(t_0)$ have been captured by the vertices and edges in the discrete representation $\mathcal{M}^0$.

COROLLARY 4.8. *If the semidiscrete flow map $\mathring{\phi}$ of a MARS method is κth-order accurate, then $E^{\mathrm{ODE}}(t_n) = O(k^\kappa)$ for any $t_n = t_0 + nk$ with $k = O(\frac{1}{n})$.*

Proof. This follows directly from Proposition 3.6 and the fact that there are $O(\frac{1}{k})$ time steps. □

The next corollary is a consequence of Lemma 3.9.

PROPOSITION 4.9. *If a MARS method represents material regions with linear semialgebraic sets for all time steps, has the discrete flow map as its mapping operation ($\varphi = \mathring{\varphi}$), and uses a κth-order accurate semidiscrete flow map $\mathring{\phi}$ with $\kappa \geq 2$, then*

$$E^{\mathrm{MAP}}(t_n) = O(h_L^2 + k^\kappa), \tag{4.16}$$

where $h_L = \max_{j=0}^n h_L(\psi_j \mathcal{M}^j)$.

Proof. There exists $c_k = O(1)$ such that

$$E^{\mathrm{MAP}}(t_n) = \left\|\bigoplus_{j=1}^n \mathring{\phi}_{t_j}^{(n-j)k} \mathcal{E}_{j-1}^{\mathrm{MAP}}\right\| \leq \sum_{j=1}^n \left\|\mathring{\phi}_{t_j}^{(n-j)k} \mathcal{E}_{j-1}^{\mathrm{MAP}}\right\| = c_k \sum_{j=1}^n \left\|\mathcal{E}_{j-1}^{\mathrm{MAP}}\right\|, \tag{4.17}$$

where the three steps follow from the definition of E^{MAP} in (4.11), (2.3), and Proposition 3.5, respectively. The proof is then completed by Lemma 3.9 and the fact that there are $O(\frac{1}{k})$ time steps. □

COROLLARY 4.10. *If a MARS method represents material regions with linear semialgebraic sets for all time steps, has the discrete flow map as its mapping operation ($\varphi = \mathring{\varphi}$), and uses a κth-order accurate semidiscrete flow map $\mathring{\phi}$ with $\kappa \geq 2$, then the overall IT error of the MARS method can be bounded as*

$$E_1(t_n) \leq E^{\mathrm{AUG}} + E^{\mathrm{ADJ}} + O(k^\kappa) + O(h_L^2), \tag{4.18}$$

where $h_L = \max_{j=0}^n h_L(\mathcal{M}^j)$.

Proof. This follows from applying Theorem 4.5 with the last three error terms on the right-hand side of (4.13) replaced with the estimates in Proposition 4.7, Corollary 4.8, and Proposition 4.9. □

The above corollary is the main conclusion on the overall error of the special MARS method using $\mathbb{S}_1^{\mathrm{D}}$, $\varphi = \mathring{\varphi}$, and a semidiscrete flow map $\mathring{\phi}$ with $\kappa \geq 2$.

4.6. The adjustment error of a special family of MARS methods. In (4.11), the error regions $\mathcal{E}_i^{\mathrm{ADJ}}$ at different time steps are mapped to the same time and thus may cancel each other in calculating the overall adjustment error E^{ADJ}. Hence the alternative definition

$$\tilde{E}^{\mathrm{ADJ}} := \sum_{j=1}^{n} \left\| \mathcal{E}_j^{\mathrm{ADJ}} \right\| \tag{4.19}$$

might overestimate the adjustment errors of a MARS method. However, for certain MARS methods the estimation of E^{ADJ} with $\tilde{E}^{\mathrm{ADJ}}$ may be good enough.

LEMMA 4.11. *Consider an adjustment operation $\chi : \mathbb{S}_q^{\mathrm{D}} \to \mathbb{S}_q^{\mathrm{D}}$ satisfying*

$$\forall \mathcal{P} \in \mathbb{S}_q^{\mathrm{D}}, \qquad \|\mathcal{P} \oplus \chi\mathcal{P}\| = O(h_{L,\chi\mathcal{P}}^2), \tag{4.20}$$

where $h_{L,\chi\mathcal{P}}$ is the Lagrangian length scale of $\chi\mathcal{P}$ as in Definition 3.8. Suppose χ is also idempotent, i.e. $\chi\circ\chi = \chi$, and it is used in the MARS method $\mathfrak{L}_{\mathrm{Mars}}^n = \chi\circ\varphi_{t_n}^k\circ\psi_n$ that satisfies

$$\forall n \in \mathbb{N}^+, \qquad \lim_{k\to 0} \psi_n = \mathcal{I}, \qquad \lim_{k\to 0} \varphi_{t_n}^k \circ \psi_n = \mathcal{I}, \tag{4.21}$$

where $\mathcal{I}$ denotes the identity operation and $\varphi_{t_n}^k$ is based on a second-order semidiscrete flow map $\mathring{\phi}_{t_n}^k$. Then the adjustment error of each time step can be sharpened as

$$\left\| \widetilde{\mathcal{M}}^n \oplus \chi\widetilde{\mathcal{M}}^n \right\| = O(kh_L^2), \tag{4.22}$$

where $h_L = \max_n h_L(\mathcal{M}^n)$, $\mathcal{M}^n = \chi\widetilde{\mathcal{M}}^n$, and $\widetilde{\mathcal{M}}^n$ is defined in (4.9).

Proof. Consider the mappings in the MARS method before t_n:

$$\widetilde{\mathcal{M}}^{n-1} \xrightarrow{\chi} \mathcal{M}^{n-1} \xrightarrow{\varphi_{t_{n-1}}^k\circ\psi_{n-1}} \widetilde{\mathcal{M}}^n \xrightarrow{\chi} \mathcal{M}^n. \tag{4.23}$$

The condition (4.20) implies

$$\left\| \widetilde{\mathcal{M}}^n \oplus \mathcal{M}^n \right\| = O(h_L^2). \tag{4.24}$$

The second-order accuracy of $\mathring{\phi}_{t_n}^k$ and Lemma 3.9 yield

$$\left\| \mathring{\phi}_{t_{n-1}}^k \psi_{t_{n-1}} \mathcal{M}^{n-1} \oplus \widetilde{\mathcal{M}}^n \right\| = O(kh_L^2). \tag{4.25}$$

It follows from (4.24), (4.25), and the triangle inequality (2.3) that

$$\left\| \mathring{\phi}_{t_{n-1}}^k \psi_{t_{n-1}} \mathcal{M}^{n-1} \oplus \mathcal{M}^n \right\| \le O(h_L^2) + O(kh_L^2). \tag{4.26}$$

Now suppose $\left\| \mathring{\phi}_{t_{n-1}}^k \psi_{t_{n-1}} \mathcal{M}^{n-1} \oplus \mathcal{M}^n \right\| = O(h_L^2)$. Because $O(h_L^2)$ does not depend on k, there exists some fixed h_L such that $\lim_{k\to 0} \left\| \mathring{\phi}_{t_{n-1}}^k \psi_{t_{n-1}} \mathcal{M}^{n-1} \oplus \mathcal{M}^n \right\| \neq 0$. This is, however, an invalid statement because, in the asymptotic case of $k \to 0$, $\mathring{\phi}_{t_{n-1}}^k \to \mathcal{I}$, the condition (4.21), and the idempotence of χ yield

$$\mathring{\phi}_{t_{n-1}}^k \psi_{t_{n-1}} \mathcal{M}^{n-1} \xrightarrow{k\to 0} \mathcal{M}^{n-1} = \chi\widetilde{\mathcal{M}}^{n-1} = \chi\circ\mathcal{I}\circ\chi\widetilde{\mathcal{M}}^{n-1} = \chi\circ\mathcal{I}\mathcal{M}^{n-1} \xrightarrow{k\to 0} \mathcal{M}^n,$$

which implies $\lim_{k\to 0} \|\mathring{\phi}^k_{t_{n-1}} \psi_{t_{n-1}} \mathcal{M}^{n-1} \oplus \mathcal{M}^n\| = 0$. Hence (4.26) is sharpened as

$$\left\| \mathring{\phi}^k_{t_{n-1}} \psi_{t_{n-1}} \mathcal{M}^{n-1} \oplus \mathcal{M}^n \right\| = O(kh_L^2), \tag{4.27}$$

which, together with (4.25) and the triangle inequality, gives (4.22). ∎

For an idempotent adjustment operation that is second-order accurate for *static* semialgebraic sets, the above lemma states that the adjustment error at each time step should be scaled by the time step size if the MARS method has a second-order time integrator and consistent augmentation and mapping operations.

5. Applications: Analyzing Interface Tracking Methods. In this section, we demonstrate the utilities of MARS by analyzing VOF methods and the iPAM method. Other IT methods, including MOF methods, the PAM method, and front tracking methods are analyzed in the supplement. These discussions answer questions (Q-1) to (Q-6) posed in Section 1.

As mentioned in Section 4.2, a *cell material region* is the intersection of the cell $\mathcal{C}$ with the material region, written $\mathcal{M}_{\mathcal{C}}(t_n) := \mathcal{M}(t_n) \cap \mathcal{C}$. A cell satisfying $\mathcal{M}_{\mathcal{C}}(t_n) = \emptyset$ or $\mathcal{M}_{\mathcal{C}}(t_n) = \mathcal{C}$ is a *pure cell* at time t_n; otherwise it is an *interface cell* at time t_n. An *interface candidate cell* at time t_n is a cell that might become an interface cell at time $t_n + k$.

5.1. VOF methods. VOF methods represent the interface inside an interface cell both explicitly with a piecewise function and implicitly with a volume fraction. Within each time step, the new volume fractions are first updated by calculating edge fluxes of the color function f from the known material regions [57]; then the new material regions are determined from the new volume fractions. These two substeps are called, respectively, the advection substep and the reconstruction substep. State-of-the-art VOF methods for these two substeps are the unsplit advection algorithms [57] and the piecewise linear interface reconstruction (PLIC) [54, 36, 35, 39]. Puckett [37, 38] showed that, given a square grid of size h covering a C^2 simple closed curve in the plane, one can construct a PLIC interface from the volume fractions so that the pointwise distance between the C^2 curve and its linear reconstruction is $O(h^2)$.

Referring to Definition 4.3, VOF methods can be considered as a special class of MARS method with its augmentation operation ψ as the identity operation (thus $E^{\text{AUG}}_{\text{VOF}} = 0$), its mapping operation φ as the advection substep, and its adjustment operation χ as the reconstruction substep:

$$\mathcal{M}^{n+1} = \mathfrak{L}^n_{\text{VOF}}(\mathcal{M}^n) = \left(\chi \circ \varphi^k_{t_n}\right) \mathcal{M}^n. \tag{5.1}$$

VOF advection algorithms output volume fractions. To comply with the signature of $\varphi^k_{t_n}$ as a unitary operation on $\mathbb{S}^{\text{D}}_q$, we define $\varphi^k_{t_n} \mathcal{M}^n$ to be a semialgebraic set that satisfies the following: (i) the volume fraction of $\varphi^k_{t_n} \mathcal{M}^n$ in each cell is the same as that produced by the advection algorithm, and (ii) the error $\|\varphi^k_{t_n} \mathcal{M}^n \oplus \mathring{\phi}^k_{t_n} \mathcal{M}^n\|$ is minimized. To make $\varphi^k_{t_n} \mathcal{M}^n$ unique, we could impose additional conditions. However, the analysis of VOF methods does not depend on the way in which $\varphi^k_{t_n} \mathcal{M}^n$ is made unique, because it is merely an intermediate result and will be promptly changed by the reconstruction substep. In other words, the analysis in the rest of this subsection holds for any definition of $\varphi^k_{t_n} \mathcal{M}^n$ that satisfies (i) and (ii).

The overall second-order accuracy of VOF methods is usually taken for granted when coupling a second-order advection algorithm and a second-order reconstruction scheme. However, the reasons for the overall IT accuracy being second order are not

obvious at all. As the main difficulty, the second-order accuracy of the advection algorithm is with respect to the entire simulation (the temporal error accumulation has already been incorporated), whereas that of the reconstruction scheme is for a fixed time instant. This difficulty is resolved by the following theorem, which states that the idempotence of the reconstruction operation guarantees the overall second-order accuracy.

THEOREM 5.1. *A VOF method of the form (5.1) is second-order accurate in the 1-norm (4.6) if*
(a) its advection algorithm is second-order accurate;
(b) its reconstruction scheme χ is second-order accurate, i.e. χ satisfies (4.20);
(c) χ is idempotent.

Proof. By Theorem 4.5, the overall error of VOF methods is

$$E_1^{\mathrm{VOF}}(t_n) \le E_{\mathrm{VOF}}^{\mathrm{MAP}} + E_{\mathrm{VOF}}^{\mathrm{ADJ}} + E_{\mathrm{VOF}}^{\mathrm{REP}} + E_{\mathrm{VOF}}^{\mathrm{ODE}}. \tag{5.2}$$

Proposition 4.7 gives $E_{\mathrm{VOF}}^{\mathrm{REP}} = O(h^2)$ and Corollary 4.8 yields $E_{\mathrm{VOF}}^{\mathrm{ODE}} = O(k^2)$. Condition (a) and (4.17) imply $E_{\mathrm{VOF}}^{\mathrm{MAP}} = O(h^2)$. Then it suffices to show $E_{\mathrm{VOF}}^{\mathrm{ADJ}} = O(h^2)$.

Since the augmentation operation in VOF methods is the identity operation, condition (4.21) holds. With conditions (b) and (c), we can apply Lemma 4.11 to deduce that $\left\|\mathcal{E}_n^{\mathrm{ADJ}}\right\| = O(kh_L^2)$ holds for each time step. By arguments similar to those for (4.17), there exists $c_k = O(1)$ such that $E^{\mathrm{ADJ}} \le c_k \sum_{j=1}^{n} \left\|\mathcal{E}_j^{\mathrm{ADJ}}\right\| = O(h^2)$, which completes the proof. ∎

PLIC reconstructions are idempotent because they preserve volume fractions, and these volume fractions uniquely determine the reconstructed interface. Thus Theorem 5.1 yields the following.

COROLLARY 5.2. *A PLIC VOF method with second-order advection and second-order reconstruction has overall IT accuracy of the second order in the 1-norm (4.6).*

The above corollary answers question (Q-3) posed in Section 1. As a caveat, some of the so-called second-order advection algorithms are strictly speaking only first-order accurate; this order reduction has recently been demonstrated in [57] for certain cases such as vortical flows.

The second-order accuracy of the MOF method is proved in Section 2.2 of the supplement. For a discussion on improving VOF methods to higher-order accuracy, see Section 2.4 of the supplement.

5.2. The iPAM method. Recently, the authors proposed a fourth-order iPAM method [62] as an improvement of the second-order PAM method (see Section 2.1 in the supplement for a justification of the second-order accuracy). The accuracy of the iPAM method is achieved by (i) augmenting the abstract data structure of PAM to faithfully represent multiple components of material regions within a single cell; (ii) removing restrictive assumptions of PAM to reduce the adjustment errors; (iii) adjusting the volume of represented cell material regions via VAPER [62, sect. 4.2]; a computational-geometry algorithm based on polygon ear removal, and (iv) maintaining a relation ($h_L = r_h h^\alpha$) between the Eulerian grid size h and the Lagrangian length scale h_L (see Definition 3.8). The last improvement (iv) is the key feature of iPAM that distinguishes it from other methods [15, 33, 28] utilizing Lagrangian markers to increase resolutions.

DEFINITION 5.3 (The iPAM method [62]). *Let $[r_{\mathsf{tiny}} h_L,\, h_L]$ be given as the range of the allowed distance between adjacent interface markers. Within each time step $[t_n, t_n + k]$,* the iPAM method *advances $\mathcal{M}^n$ to $\mathcal{M}^{n+1}$ by applying six substeps to each interface candidate cell:*

(iPAM-1) *Calculate the cell preimage* $\mathring{\varphi}_{t_n+k}^{-k}(\mathcal{C})$.

(iPAM-2) *Compute the intersection* $\overleftarrow{\mathcal{M}_{\mathcal{C}}^{n+1}} = \mathring{\varphi}_{t_n+k}^{-k}(\mathcal{C}) \cap \mathcal{M}^n$.

(iPAM-3) *Trace the intersection forward to obtain* $\mathcal{M}_{\mathcal{C}}^{n+1}$.

(iPAM-4) *If* $\|p_j - p_{j+1}\|_2$, *the length of an interface edge* $\overline{p_j p_{j+1}} \subset \partial\mathcal{M}_{\mathcal{C}}^{n+1}$, *is greater than* h_L, *divide* $\mathring{\varphi}_{t_n+k}^{-k}(\overline{p_j p_{j+1}})$ *into* $\left\lceil \frac{\|p_j - p_{j+1}\|_2}{h_L} \right\rceil$ *equilength subedges by inserting new markers and add the image of the new markers in between* p_j *and* p_{j+1} *to update* $\partial\mathcal{M}_{\mathcal{C}}^{n+1}$. *Repeat this step until no interface edge of* $\partial\mathcal{M}_{\mathcal{C}}^{n+1}$ *is longer than* h_L.

(iPAM-5) *If an interface edge* $\overline{p_j p_{j+1}} \subset \partial\mathcal{M}_{\mathcal{C}}^{n+1}$ *has length smaller than* $r_{\mathsf{tiny}} h_L$, *replace this edge with its midpoint* $\frac{p_j + p_{j+1}}{2}$.

(iPAM-6) *(Optional.) If* $\mathbf{u}$ *is divergence-free, attempt to enforce* $\left\|\mathcal{M}_{\mathcal{C}}^{n+1}\right\| = \left\|\overleftarrow{\mathcal{M}_{\mathcal{C}}^{n+1}}\right\|$ *by the VAPER algorithm [62, sect. 4.2].*

In substep (iPAM-1), a cell preimage is obtained from the preimage of its boundary; *the tracings are consistent for neighboring cells* in that the preimage of a cell edge is used simultaneously for the two cells whose boundaries share the same edge. Consequently, the first two substeps, (iPAM-1) and (iPAM-2), can be interpreted as a repartition of the approximated material region $\mathcal{M}^n := \bigcup_{\mathcal{C}\subseteq\Omega} \mathcal{M}_{\mathcal{C}}^n$ for *global* mass conservation. In contrast, the last step, (iPAM-6), attempts to enforce *local* mass conservation for each interface candidate cell. Note that this step of mass conservation should not be invoked for compressible flows. The substep (iPAM-4) enforces an upper bound of the distance between two adjacent markers by subdividing long interface edges with newly added markers. In comparison, (iPAM-5) enforces a lower bound by replacing edges of negligible lengths with their midpoints, which is equivalent to imposing an upper bound on n_{mks}, the maximum number of interface markers. See [62, Fig. 4.2] for illustrations of (iPAM-4) and (iPAM-5).

In Definition 5.3, (iPAM-1), (iPAM-2), and (iPAM-4) can be considered as an augmentation operation for inserting additional vertices into the interface at the beginning of the time step. In (iPAM-3), the discrete flow map acts on the material regions. (iPAM-5) and the VAPER algorithm in (iPAM-6) can be regarded as an adjustment operation. Therefore, iPAM is of MARS type.

PROPOSITION 5.4 (Convergence of the iPAM method[4]). *The iPAM method is* (2α)*th-order accurate in the 1-norm (4.6) if*

(a) its discrete flow map $\mathring{\varphi}$ *is at least* (2α)*th-order accurate in temporal integration;*

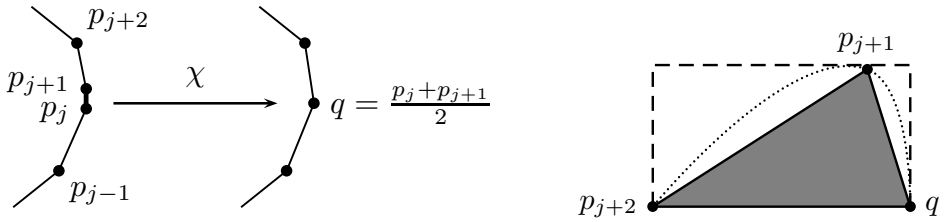
(b) the time step size satisfies $k = O(h)$*;*

(c) $h_L = r_h h^\alpha$ *is fulfilled with the constants* $r_h = O(1)$ *and* $\alpha \geq 1$.

Proof. By (iPAM-4) and assumptions (b) and (c), the maximum distance between two adjacent vertices on the interface is $h_L = O(h^\alpha) = O(k^\alpha)$ for all time steps. Adding new markers to the edges does not incur any error, hence $E^{\mathrm{AUG}} = 0$. By Corollary 4.10, it suffices to show that $E^{\mathrm{ADJ}} = O(k^{2\alpha})$.

The adjustment operation χ in iPAM method consists of (iPAM-5) and the VAPER algorithm in (iPAM-6). When (iPAM-6) is invoked for incompressible flows,

[4]This proposition also appears as Proposition 4.4 in [62], where another assumption (d) on the adjustment error is made. In revising this work, we found that, due to Lemma 4.11, the assumption on the adjustment error can be removed as redundant to strengthen the conclusion on the convergence rate of the iPAM method.



(a) Replacing a short edge with its midpoint (b) The corresponding error

FIG. 5.1. *Adjustment error of (iPAM-5). In subplot (a), an edge of negligible length is replaced with its midpoint by the adjustment operator χ. Subplot (b) is a rotated zoom-in of a triangle near q in subplot (a). The shaded region represents one of the two parts of the adjustment error.*

we have $\|\mathcal{M}_{\mathcal{C}}^{n+1}\| = \|\phi_{t_n+k}^{-k}\mathcal{M}_{\mathcal{C}}^{n+1}\|$. By Proposition 3.6 and Lemma 3.9,

$$\left\|\phi_{t_n+k}^{-k}\mathcal{M}_{\mathcal{C}}^{n+1} \oplus \overleftarrow{\mathcal{M}_{\mathcal{C}}^{n+1}}\right\| = O(k^{2\alpha+1}) + O(kh_L^2 + k^{2\alpha+1}) = O(k^{2\alpha+1}). \tag{5.3}$$

Hence VAPER in (iPAM-6) incurs adjustment errors totaling at most $O(k^{2\alpha})$ over $O(\frac{1}{k})$ time steps.

It is readily verified that χ is idempotent. It is also trivial to verify that (4.21) holds for iPAM. By Lemma 4.11, it suffices to show that (4.20) holds when (iPAM-5) is applied to a static semialgebraic set.

As shown in [62, Fig. 6 (c) and (d)] and Figure 5.1 (a), the error caused by replacing an edge $\overline{p_jp_{j+1}}$ with its midpoint q is the total volume of two triangles, one with vertices p_{j-1}, p_j, q and the other p_{j+1}, p_{j+2}, q. Consider the latter triangle in Figure 5.1 (b). The vertices define a quadratic curve, the tight bounding box of which also contains the triangle. By Rolle's theorem, the maximum distance of the points on the curve to the longest edge of the triangle is $O(h_L^2)$, which is also the height of the bounding box. Hence the volume of the triangle is at most $O(h_L^3)$. Since there are $O(\frac{1}{h_L})$ vertices, the maximum number of such replacing operations per time step is $O(\frac{1}{h_L})$, and thus applying (iPAM-5) to a static semialgebraic set $\mathcal{P}$ causes an error of $O(h_L^2)$. Note that in the case of $\mathcal{P}$ having C^1 discontinuities, Rolle's theorem does not apply at the vicinity of such a discontinuity, and the volume of the aforementioned triangle could be $O(h_L^2)$. Fortunately, Sard's theorem states that there exists at most $O(1)$ such C^1 discontinuities and the above estimation of adjustment errors still holds.

To sum up, the error caused by applying the iPAM adjustment operation to a static semialgebraic set $\mathcal{P}$ is $O(h_L^2)$. Lemma 4.11 implies that $\left\|\mathcal{E}_n^{\mathrm{ADJ}}\right\| = O(kh_L^2)$ for each time step. Then arguments similar to those for (4.17) imply that there exists $c_k = O(1)$ such that $E^{\mathrm{ADJ}} \le c_k \sum_{j=1}^{n} \left\|\mathcal{E}_j^{\mathrm{ADJ}}\right\| = O(h^2)$. The proof is completed by applying Corollary 4.10 with assumption (a). □

Results of the numerical tests in [62] confirm that setting $\alpha = 1.5$ and $\alpha = 2$ leads, respectively, to third- and fourth-order accuracy of the iPAM method.

Since the iPAM method runs in $O(\frac{1}{h_L}) = O(\frac{1}{h^2})$ time for each time step, the choice of $\alpha = 2$ balances the cost of interface tracking with that of the flow solver in 2D. We caution the reader that, in choosing the parameters in the relation $h_L = r_h h^\alpha$, α should not exceed 2 and r_h should not be much smaller than 1; otherwise the expense of IT might dominate that of the flow solver. See [62, sect. 4] for more discussions.

Proposition 5.4 partially answers question (Q-1) posed in Section 1. Since the analysis in Sections 3 and 4 also applies to 3D, the answer to question (Q-2) is positive.

There exist high-order IT methods such as the spectral element method by Sussman and Hussaini [46]. However, the interface there is assumed to be locally smooth, and this smoothness assumption is essential in establishing the spectral convergence rates higher than two. In particular, no numerical experiments are performed in [46] for problems with dynamically generated kinks. As discussed at the end of Section 3.3, Lemma 3.9 dictates that the second-order error in approximating the semidiscrete flow map cannot be improved by using higher-order representations of material boundaries, due to the fundamental difficulty incurred by dynamic C^1 discontinuities. This is confirmed by numerical experiments performed in Section 6.

As a unique feature of iPAM, its fourth-order accuracy in terms of h is achieved by enforcing the relation $h_L = O(h^2)$, and this fourth-order accuracy is independent of dynamic kinks, as shown both by the analysis in Proposition 5.4 and by numerical experiments in [62]. To the best of our knowledge, the iPAM method is the first fourth-order IT method whose convergence rates are independent of C^1 discontinuities dynamically generated on the interface.

To avoid any potential confusion and misinterpretation, we reiterate that *the iPAM method is fourth-order accurate in terms of the Eulerian grid size h, but is only second-order accurate in terms of the Lagrangian length scale h_L.* Inspired by the analysis in MARS, the fourth-order accuracy of the iPAM method is not a contradiction of Lemma 3.9, but rather a circumvention of the fundamental difficulty caused by dynamic kinks.

6. Numerical Experiments. In this section, further improvements of the iPAM method are numerically investigated to answer questions (Q-7), (Q-8), and (Q-9) posed in Section 1.

A *cubic iPAM method* is attained from the original linear iPAM method by turning off (iPAM-6) in Definition 5.3 and representing material region with *cubic splinegon*s, i.e. curvilinear polygons with cubic splines as the edges. When computing the intersection of two such splinegons, we first calculate intersections of the corresponding linear polygons and then use them as the initial guess in an iterative Newton's method to obtain accurate intersection points of the splinegons. This algorithm of splinegon intersection works well because the topology of the intersection in the cubic case is the same as that in the linear case for a sufficiently small Eulerian grid size h. The advected Lagrangian markers are used as the knots of the periodic cubic splines. When dividing a long edge into smaller ones, the new markers are added on the cubic spline so that augmentation errors remain zero.

6.1. Rotation of the Zalesak disk. In this test, a slotted disk is placed in a purely rotational velocity field given by the stream function

$$\psi(x, y) = -\frac{\omega}{2}[(x - x_O)^2 + (y - y_O)^2], \tag{6.1}$$

where $O = (x_O, y_O)$ is the center of the rotation and ω the constant angular velocity. The initial setup and other parameters of this test are shown in Figure 6.1. After a full revolution of 2π rotation, the slotted circle returns to its initial location, hence the exact solution is the same as the initial condition.

We perform this test using the cubic iPAM method with $h_L = 0.1h, h^{3/2}, 5h^2$. At the initial time, the cubic splinegons in the interface cells are computed from a single cubic spline that represents the boundary of the slotted disk. Since the four corners of the initial disk remain C^1 discontinuities during the solid-body rotation, at each time step we only specify C^0 continuity for each of the four corners when constructing

Parameters	Values
computational domain	$\Omega = [0,4] \times [0,4]$
simulation time	$t \in [0, \frac{2\pi}{\omega}]$
slot parameters	$r = 0.4$, $s = 0.06$
circle parameters	$R = 0.5$, $C = (2.0, 2.75)$
velocity parameters	$O = (2.0, 2.0)$, $\omega = 0.5$
Courant number	Cr=1.0
Eulerian grid sizes	h= 0.08, 0.04, 0.02
Lagrangian length scales	$h_L = 0.1h, h^{3/2}, 5h^2$

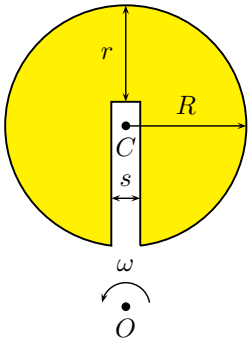
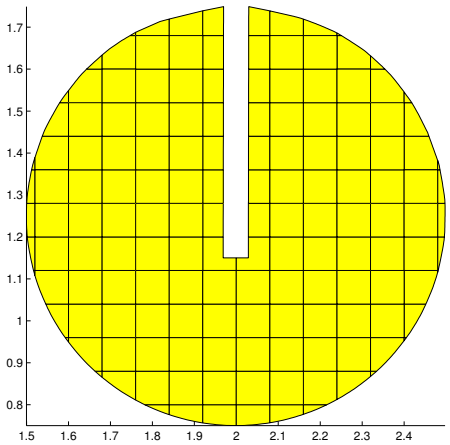
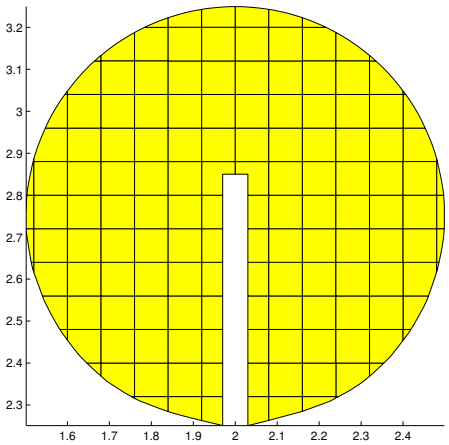


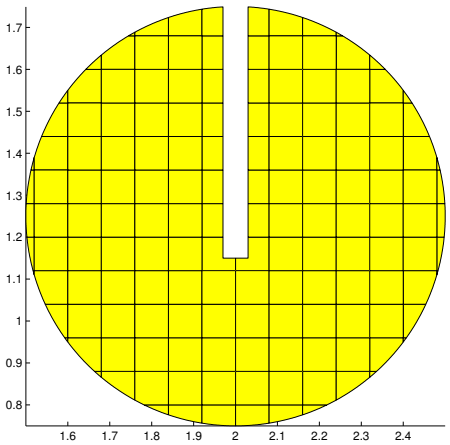
FIG. 6.1. *Initial setup and other parameters of the Zalesak-rotation test*



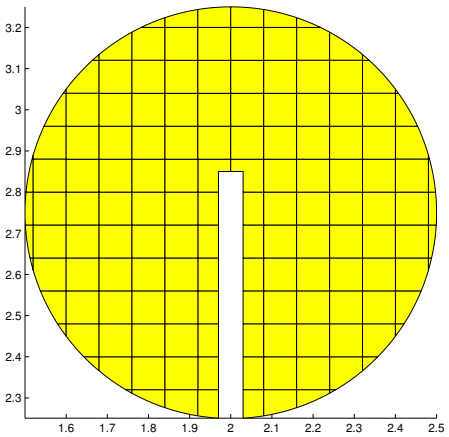
(a) linear iPAM with $h_L = 5h^2$ for $t = \frac{1}{2}T$



(b) linear iPAM with $h_L = 5h^2$ for $t = T$



(c) cubic iPAM with $h_L = 0.1h$ for $t = \frac{1}{2}T$



(d) cubic iPAM with $h_L = 0.1h$ for $t = T$

FIG. 6.2. *Results of the iPAM method for the Zalesak-rotation test on a grid with* $h = 0.08$. *The set of simple polygons within the control volumes are shown for the tracked material. Subplots (a) and (b) are taken from Figure 10 in [62]. Subplots (c) and (d) are the results of this work.*

TABLE 6.1

Errors and convergence rates of the iPAM method for the Zalesak-rotation test in Figure 6.1. Results of the linear iPAM method are taken from [62].

Test case	$E_g(h = 0.08)$	rate	$E_g(h = 0.04)$	rate	$E_g(h = 0.02)$
linear iPAM $h_L = 5h^2$	6.55e-03	4.45	3.00e-04	4.49	1.34e-05
cubic iPAM $h_L = 5h^2$	1.49e-08	7.84	6.50e-11	8.21	2.19e-13
cubic iPAM $h_L = h^{3/2}$	3.20e-09	5.65	6.38e-11	6.96	5.13e-13
cubic iPAM $h_L = 0.1h$	2.87e-07	3.99	1.80e-08	4.00	1.13e-09

TABLE 6.2

An accuracy comparison of state-of-the-art VOF methods with the cubic iPAM method for the Zalesak-rotation test in Figure 6.1 on a grid with $h = 0.02$. For this test, the definition of the geometric error in these VOF methods differs from E_g in (4.5) by a constant factor of about 0.74. Hence the error of the cubic iPAM method is multiplied by a factor of 1.35 to facilitate the comparison.

Method	Geometric error ($1.35E_g$ with $h = 0.02$)
Stream/Puckett [26]	1.00×10^{-2}
EMFPA-SIR (cubic spline) [31]	9.73×10^{-3}
Quadratic fit + continuity [41]	4.16×10^{-3}
cubic iPAM with $h_L = 0.1h$ (this work)	1.53×10^{-9}

the corresponding cubic splinegon within some interface cell. In Figure 6.2, material polygons of the cubic iPAM method with $h_L = 0.1h$ are plotted on actual grids for two time instants $t = \frac{1}{2}T$ and $t = T$, along with the corresponding results of the linear iPAM method [62]. Clearly, the cubic iPAM method preserves the shape of the sharp corners on the periphery of the disk much better than the linear iPAM method.

We compare errors and convergence rates of the linear iPAM method to those of the cubic iPAM method for the relation $h_L = 5h^2$. From the first two lines of Table 6.1, the maximum numbers of markers for each test are exactly the same for the two methods, yet the accuracy on the finest grid is improved by 8 orders of magnitude! The last three lines of Table 6.1 demonstrate the effect upon the overall IT accuracy from various choices of r_h and α in the relation $h_L = r_h h^\alpha$. The choices of $\alpha = 1, \frac{3}{2}$, and 2 yield convergence rates of 4, 6, and 8, respectively. Sometimes the expensive choice of $\alpha = 2$ may be relaxed if a small r_h with $\alpha = 1, \frac{3}{2}$ has already lead to solutions that are accurate enough. As shown in the second and the third lines of Table 6.1, the choice of $h_L = h^{3/2}$ has an accuracy very close to that of $h_L = 5h^2$ on the finest grid, although the asymptotic expense of the former is much less than that of the latter. The above observations illustrate the flexibility of the iPAM method in striking a balance between accuracy and efficiency; see [62, sect. 5.2] for more discussions. Questions (Q-7) and (Q-8) posed in Section 1 are also partially answered by these numerical results.

We also compare the accuracy of the cubic iPAM method ($h_L = 0.1h$) to that of state-of-the-art VOF methods in Table 6.2. On the finest grid, the iPAM method is more accurate than the VOF methods by roughly 6 orders of magnitude! Although VOF methods can utilize adaptive mesh refinement (AMR) to increase its accuracy, it can be shown that the complexity of such a AMR-VOF method is $O(\frac{1}{h})$ for each time step. By [62, Cor. 4.7], the iPAM method with $h_L = 0.1h$ also runs in $O(\frac{1}{h})$ time

for each time step. Hence the CPU-time ratio of the AMR-VOF method to the iPAM method is a constant for any given test with fixed grid size and Courant number. Our implementation of the EMFPA-SIR [31] method and timing results indicate that its CPU-time ratio to the cubic iPAM method is around 0.25 for this test.

To compare the efficiency of the two methods in achieving the same accuracy, we quantify the speedup of the cubic iPAM method over VOF methods in terms of CPU time. Systematic arguments and simple formulas for this purpose have already been proposed in [60, sect. 7] and [62, sect. 5.3], here we simply apply the same idea to the Zalesak-rotation test. In order to reach the same accuracy of the cubic iPAM method in Table 6.2, the EMFPA-SIR has to refine the grid at least 11 times (by a factor of 2) so that the second-order accuracy yields an error reduction ratio of 4^{11}. Meanwhile, the computational cost in terms of CPU time is increased by a factor of 4.0×10^6, due to the linear complexity of each time step and the fact that the solution has to be advanced over $O(\frac{1}{k})$ time steps. Since the EMFPA-SIR method is about four times faster than the cubic iPAM method on the grid with $h = 0.02$, the speedup of the cubic iPAM method over the EMFPA-SIR method is roughly 10^6. As of June 2015, the fastest computer in the world was the Tianhe-2 computer in China with 3,120,000 cores and a maximum Linpack performance at $33,862,700$ Gigaflops [32]. The personal desktop of the first author has a single Intel® i7-3930 hexa-core CPU and runs at a speed around 138 Gigaflops. Clearly, an upper bound[5] of the speedup in switching from the author's personal desktop to Tianhe-2 is about 2.5×10^5. The following remark is justified by the fact that the speedup of the cubic iPAM method over EMFPA-SIR (10^6) is greater than 2.5×10^5.

REMARK 6.1. *To achieve an IT accuracy of* 1.5×10^{-9} *for the Zalesak-rotation test detailed in Figure 6.1, it is faster to run the cubic iPAM method on a personal computer than to run the EMFPA-SIR method on the fastest supercomputer in the world!*

It is shown in [60, sect. 7] and [62, sect. 5.3] that the CPU-time speedup of a fourth-order method over a second-order method grows as a power function as the targeted accuracy is reduced. This pattern remains the same for different tests, although values of the speedup for a given accuracy may vary largely across these tests.

6.2. Vortex shear of a circular disk. In this test, we numerically solve the popular single-vortex problem, where a nonuniform, transient velocity field given by the stream function

$$\psi(x, y) = -\frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos\left(\frac{\pi t}{T}\right) \tag{6.2}$$

is imposed on a circular disk. The initial setup and other test parameters are shown in Figure 6.3. At time $t = \frac{T}{2}$, the velocity field is reversed by the cosinusoidal temporal factor so that the exact solution at $t = T$ coincides with the initial condition.

For vortex-shear tests with short time periods $T = 0.5, 2$, Figure 6.4 shows the material regions at $t = \frac{T}{2}$ which are generated by the original *linear* iPAM method

[5]With the assumption of perfect scaling, this upper bound is very loose. The load balancing problem for high-performance computing is notoriously difficult. For practical problems, it is often the case that the computing time cannot be reduced any more once the number of processes reaches tens of thousands, let alone millions of them. By using the timing results of EMFPA-SIR method with single levels, we have also assumed that the time spent in AMR bookkeeping for a corresponding AMR-VOF method is zero.

Parameters	Values
computational domain	$[0,1]\times[0,1]$
simulation time	$t\in[0,T]$
shape parameters	$C=(0.5,0.75),\ R=0.15$
velocity periods	$T=0.5,\ 2,\ 8$
Courant number	$\mathrm{Cr}=1$
Eulerian grid sizes	$h=\frac{1}{32},\ \frac{1}{64},\ \frac{1}{128}$
Lagrangian length scales	$h_L=0.2h,\ 0.5h^{3/2},\ 5h^2$

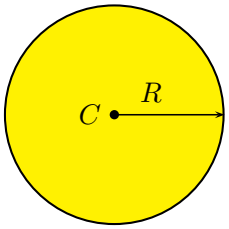
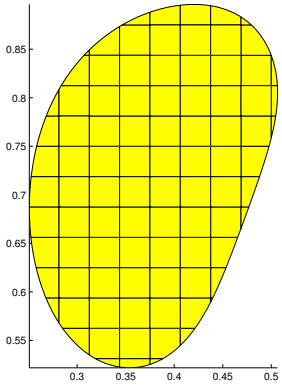
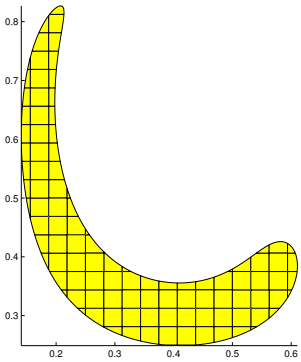


FIG. 6.3. *Initial setup and parameters of single-vortex tests for the cubic iPAM method.*



(a) $T=0.5$



(b) $T=2$

FIG. 6.4. *Results of the original linear iPAM method for the vortex-shear tests at* $t=\frac{T}{2}$ *on a* 32^2 *grid. The numbers of Lagrangian markers are, respectively, 1956 and 2992 for* $T=0.5$ *and* $T=2$.

since cubic splines might round the corner at a true kink. During the entire course of these simulations, we do not observe any kinks. However, as shown in [62, Fig. 5.6], the tail of the spiral develops into a prominent kink for the test with $T=8$.

We list the errors and convergence rates of the cubic iPAM method along with those of the original iPAM method in Table 6.3. A number of observations follow.

(SV-1) For $T=0.5,2$, convergence rates of the cubic iPAM method with $h_L=0.2h$ are well over 4, and its errors are much smaller than those of the linear iPAM method. Meanwhile, the complexity of the cubic iPAM method is $O(\frac{1}{h})$, a cost much cheaper than that of the linear iPAM method.

(SV-2) For $T=0.5,2$, the cubic iPAM method with $h_L=0.5h^{1.5}$ and $h_L=5h^2$ are more accurate than the linear iPAM method by, respectively, 4 and 6 orders of magnitude!

(SV-3) For $T=0.5,2$, errors of the cubic iPAM method with $h_L=5h^2$ on the 128^2 grid are very close to machine precision, indicating good conditioning.

(SV-4) For $T=0.5,2$, the choices of $h_L=O(h),O(h^{1.5}),O(h^2)$ in the cubic iPAM method yield convergence rates of 4, 6, 8, respectively.

(SV-5) In the case of $T=8$, convergence rates of the cubic iPAM method deteriorate to about 2.5, 3.5, 4.5 for $h_L=O(h),O(h^{1.5}),O(h^2)$, respectively. The accuracy of the economy case $h_L=0.2h$ is substantially worse than that of the

TABLE 6.3

Errors and convergence rates of the iPAM method based on E_g for the vortex-shear tests. The results of the linear cases are taken from [62]. The substep (iPAM-6) is turned off.

iPAM tests of vortex shear	$h = \frac{1}{32}$	rate	$h = \frac{1}{64}$	rate	$h = \frac{1}{128}$
	E_g for $T = 0.5$				
linear segments, $h_L = h^2$	1.22e-07	3.85	8.45e-09	3.88	5.73e-10
cubic spline, $h_L = 0.2h$	5.35e-08	4.99	1.69e-09	4.98	5.35e-11
cubic spline, $h_L = 0.5h^{\frac{3}{2}}$	4.03e-10	6.96	3.24e-12	7.38	1.95e-14
cubic spline, $h_L = 5h^2$	3.44e-11	8.14	1.21e-13	9.10	2.22e-16
	E_g for $T = 2$				
linear segments, $h_L = h^2$	5.04e-07	4.44	2.32e-08	3.82	1.64e-09
cubic spline, $h_L = 0.2h$	1.13e-07	4.86	3.87e-09	4.51	1.70e-10
cubic spline, $h_L = 0.5h^{\frac{3}{2}}$	2.97e-10	6.71	2.84e-12	6.60	2.94e-14
cubic spline, $h_L = 5h^2$	8.72e-10	7.97	3.48e-12	8.73	8.21e-15
	E_g for $T = 8$				
linear segments, $h_L = h^2$	4.32e-06	4.64	1.74e-07	3.89	1.17e-08
cubic spline, $h_L = 0.2h$	1.22e-06	1.71	3.72e-07	2.35	7.28e-08
cubic spline, $h_L = 0.5h^{\frac{3}{2}}$	1.48e-07	4.07	8.86e-09	3.65	7.06e-10
cubic spline, $h_L = 5h^2$	5.89e-07	4.90	1.97e-08	4.64	7.92e-10

linear iPAM method for the two finest grids.

In the absence of dynamic kinks, (SV-1) indicates that the cubic iPAM method with $h_L = O(h)$ is an answer to question (Q-7) posed in Section 1. In comparison, (SV-2), (SV-4), and (SV-5) suggest that a *local dynamic* kink may adversely affect the *global* accuracy of IT, especially when quadratic or higher-order curves are employed. The order reduction also confirms the fundamental difficulty of high-order IT revealed by Lemma 3.9 that reduction of the mapping error is limited by the Lagrangian length scales. Despite its eighth-order accuracy in the absence of dynamic C^1 discontinuities, *the current cubic iPAM method with $h_L = O(h^2)$ is still a fourth-order method.* After all, an IT method has to be fourth-order accurate for *all* possible test cases to be called a "fourth-order" IT method. On the other hand, results of the Zalesak-rotation test indicate that the cubic iPAM method may be improved to an accuracy higher than fourth order if we can capture local dynamic kinks with enough accuracy. These discussions partially answer question (Q-8) posed in Section 1.

6.3. Curvature estimation. In simulating multiphase flows, it is notoriously difficult to accurately estimate the curvature of a deforming interface. For VOF methods, curvature can be estimated from volume fractions to second-order accuracy by the height function approach [27, 44]. Fourth-order formulas of this approach also exist for static shapes [47], but the authors are not aware of any demonstrated fourth-order accuracy in estimating curvature from the results of a nontrivial IT test. As the major difficulty, rarely does an IT method have the property that, after a finite time period of deformation, the tracked interface is still accurate enough to support fourth-order curvature estimation.

The iPAM method is an exception in that it is fourth-order accurate not only for IT, but also for curvature estimation. To support this claim, we estimate the interface

curvature at both the initial and the final times of simulations for the vortex-shear tests presented in the previous section. The steps are detailed as follows.

(CE-1) For all cells of a grid with size h, compute at time $t = 0$ the volume fractions resulting from a linear or cubic spline representation of the circle; the Eulerian grid size is set to $h = \frac{1}{64}, \frac{1}{128}$, or $\frac{1}{256}$, while the Lagrangian length scale is chosen as $h_L = h^2$ (linear segments) or $h_L = 0.2h, 0.5h^{\frac{3}{2}}, 5h^2$ (cubic splines).

(CE-2) For each interface cell $\mathcal{C}_{\mathbf{i}}$ with its volume fraction between 0.0001 and 0.9999, estimate its curvature from the volume fractions in (CE-1) using the fourth-order formula proposed by Sussman and Ohta [47, p. 427]:

$$\kappa_{\mathbf{i}} = \frac{\ddot{H}}{(1+\dot{H}^2)^{\frac{3}{2}}}, \tag{6.3}$$

where $\mathbf{i}$ is the multi-index of an interface cell,

$$\ddot{H} = -\frac{1}{8}(F_{-2}+F_2) + \frac{3}{2}(F_{-1}+F_1) - \frac{11}{4}F_0;$$

$$\dot{H} = \frac{5}{48}(F_{-2}-F_2) - \frac{17}{24}(F_{-1}-F_1);$$

$$F_\ell = \begin{cases} \sum_{j=-6}^{+6} \langle f \rangle_{\mathbf{i}+(\ell,j)} & \text{if the local interface is aligned more horizontally,} \\ \sum_{j=-6}^{+6} \langle f \rangle_{\mathbf{i}+(j,\ell)} & \text{if the local interface is aligned more vertically;} \end{cases}$$

and $\langle f \rangle_{\mathbf{i}+(\ell,j)}$ is the volume fraction of cell $\mathbf{i}+(\ell,j)$ as defined in Section 4.2. Note that the local interface at cell $\mathbf{i}$ is aligned more horizontally than vertically if $\sum_{j=-6}^{+6} \left| \langle f \rangle_{\mathbf{i}+(1,j)} - \langle f \rangle_{\mathbf{i}+(-1,j)}) \right| < \sum_{j=-6}^{+6} \left| \langle f \rangle_{\mathbf{i}+(j,1)} - \langle f \rangle_{\mathbf{i}+(j,-1)}) \right|$; otherwise it is aligned more vertically than horizontally.

(CE-3) Deduce curvature errors from the exact curvature $\kappa_E = 1/R = 20/3$ and compute the max-norm and 1-norm of the error vector.

(CE-4) Track the interface to the final time $T = 0.5, 2$, or 8 by the iPAM method with the same h and h_L as those in (CE-1).

(CE-5) Repeat (CE-2) and (CE-3) to estimate curvature from the volume fractions at the final time.

In Table 6.4, we organize results of the 72 vortex-shear tests into six cases as many tests yield the same results. Case I contains all tests of curvature estimation at the initial time. For all choices of h_L and h, either the linear or the cubic spline representation is accurate enough to achieve fourth-order curvature estimation of the circle both in the max-norm and the 1-norm. Note that fourth-order convergence rates of curvature estimation for a *static* unit circle have already been shown by Sussman and Ohta [47]. Since the circle radii in our tests and theirs are, respectively, 0.15 and 1, the error of their curvature estimation at the grid size $\frac{1}{32}$ corresponds to that of ours at $h = \frac{3}{640} \approx \frac{1}{213}$.

Case II contains all tests of curvature estimation at the final time with $T = 0.5, 2$. The results of Case II are identical to those of Case I, implying that if the interface is always smooth, the iPAM method with all the above choices of h_L tracks the interface well enough so that the curvature errors caused by IT are much smaller that those caused by the fourth-order formula (6.3).

TABLE 6.4

Errors and convergence rates of the iPAM method in estimating curvature for the vortex-shear tests both at the initial time $t = 0$ (Case I) and at the final time (all other cases) $T = 0.5, 2, 8$. The relation $h_L = h^2$ implies linear-segment representation of the moving interface while $h_L = 0.2h, 0.5h^{\frac{3}{2}}$, or $5h^2$ implies cubic spline representation.

	$h = \frac{1}{64}$	rate	$h = \frac{1}{128}$	rate	$h = \frac{1}{256}$
Case I: $t = 0$, linear iPAM with $h_L = h^2$ and cubic iPAM with $h_L = 0.2h, 0.5h^{\frac{3}{2}}, 5h^2$					
$L_\infty(E_\kappa)$	4.74e-02	4.29	2.42e-03	3.93	1.59e-04
$L_1(E_\kappa)$	1.12e-02	4.70	4.30e-04	4.08	2.54e-05
Case II: $t = T = 0.5, 2$, linear iPAM with $h_L = h^2$ and cubic iPAM with $h_L = 0.2h, 0.5h^{\frac{3}{2}}, 5h^2$					
$L_\infty(E_\kappa)$	4.74e-02	4.29	2.42e-03	3.93	1.59e-04
$L_1(E_\kappa)$	1.12e-02	4.70	4.30e-04	4.08	2.54e-05
Case III: $t = T = 8$, linear iPAM with $h_L = h^2$					
$L_\infty(E_\kappa)$	4.75e-02	4.28	2.44e-03	3.72	1.85e-04
$L_1(E_\kappa)$	1.12e-02	4.62	4.55e-04	4.09	2.68e-05
Case IV: $t = T = 8$, cubic iPAM with $h_L = 0.2h$					
$L_\infty(E_\kappa)$	1.68e-01	-0.48	2.35e-01	-2.39	1.23e+00
$L_1(E_\kappa)$	1.72e-02	2.10	4.01e-03	-1.05	8.29e-03
Case V: $t = T = 8$, cubic iPAM with $h_L = 0.5h^{\frac{3}{2}}$					
$L_\infty(E_\kappa)$	4.74e-02	4.24	2.51e-03	2.03	6.13e-04
$L_1(E_\kappa)$	1.14e-02	4.58	4.76e-04	4.00	2.98e-05
Case VI: $t = T = 8$, cubic iPAM with $h_L = 5h^2$					
$L_\infty(E_\kappa)$	4.74e-02	3.11	5.48e-03	5.11	1.59e-04
$L_1(E_\kappa)$	1.21e-02	4.52	5.29e-04	4.34	2.61e-05

As detailed in Table 6.4, the last four cases concern curvature estimation at the final time of the longest vortex-shear test, where a C^1 discontinuity is prominent for most of the period [62, Fig. 12]. With the choice $h_L = O(h^2)$ in Cases III and VI, we observe convergence rates close to four for both cases. The slightly better accuracy of Case VI is probably due to its cubic spline representation. Due to the adverse effect of the kink, the nonconvergent curvature estimation of Case IV is expected. In Table 6.3, the IT error for this case with $h = \frac{1}{128}$ is 7.28×10^{-8}, much smaller than 0.235, the max-norm of corresponding curvature errors. This dramatic contrast suggests that *tracked interface loci that appear accurate might lead to very inaccurate curvature estimates*. In addition, a comparison of the errors of Case III to those of Case IV shows that *the curvature estimated from a spline representation might be much less accurate than that from a linear-segment representation*.

With a slight increase of the computational cost in Case V, we were able to obtain second-order and fourth-order convergence rates in the max-norm and in the 1-norm, respectively. The above discussion on the results in Table 6.4 suggests that using cubic splines with $h_L = O(h)$ and $h_L = O(h^{\frac{3}{2}})$ might be efficient choices of accurate curvature estimation for a smooth interface and a C^1 discontinuous interface, respectively.

Our stringent numerical tests strongly suggest that, despite the possible formation

of C^1 discontinuities on a deforming interface, its curvature can *always* be estimated to fourth-order accurate by appropriately choosing the spline type and the Lagrangian length scale under the MARS framework. In summary, this subsection gives a positive answer to question (Q-9) posed in Section 1, supporting our discussions in the previous section on the iPAM method being fourth-order accurate.

7. Conclusions. We have proposed MARS, a generic framework of analyzing IT methods in both compressible and incompressible flows. By examining the coupling of advection algorithms and reconstruction schemes, we prove a theorem on the overall second-order accuracy of VOF methods including the MOF method. The fourth-order accuracy of the iPAM method is also rigorously proved. In addition, MARS applies directly to front tracking methods, thanks to the similarity between the iPAM method and front tracking methods, To the best of our knowledge, MARS is the first generic framework for systematic analysis of explicit IT methods.

We investigate the possibilities of further improving the accuracy and efficiency of the iPAM method by solving standard benchmark tests with a cubic iPAM method. When cubic splines are used to represent the interface, a central issue with practical significance is how to minimize the adverse influence of local dynamic kinks on the global IT accuracy. One possible solution is to capture these dynamic kinks and enforce C^0 continuity on the underlying spline representation.

Numerical results demonstrate the great potential of MARS in directing future development of highly accurate and efficient IT methods. To achieve a geometric accuracy of 1.5×10^{-9} for the Zalesak-rotation test, a cubic iPAM method with optimal linear complexity running on the author's personal desktop is faster than a state-of-the-art VOF method running on the fastest supercomputer in the world! Under the MARS framework, the interface curvature can also be estimated to fourth-order accuracy, regardless of the potential development of C^1 discontinuities.

The next step along this line of research is the extension of the iPAM method to 3D and an arbitrary number of materials under the guidance of MARS. In future research of numerically simulating multiphase flows with moving boundaries, the MARS framework serves as a solid theoretical foundation for coupling the iPAM method to fourth-order finite volume methods [63, 59, 60]. For irregular domains, spatial operators can be discretized as linear combinations of cell-averaged quantities on *structured* control volumes, via combining the representation of material regions by regular semialgebraic sets, the interpolation stencils based on principal lattices [55], and the algebraic formulas for multidimensional quadrature on donating regions [56]. The ultimate goal is a fourth-order semi-Lagrangian method of linear complexity for solving the incompressible Navier-Stokes equations with sharp moving boundaries, of which the convergence rates are dependent neither on dynamic C^1 discontinuities of the interface nor on topological changes of the material regions.

Acknowledgements. This work was supported in part by NSF grant DMS-1160432 and NIH grant 1R01-GM090203. The authors would like to thank two anonymous referees, whose comments led to an improvement on the quality of this paper. The authors also thank Dr. Pavel Bochev for his insightful suggestions on organizing the contents of this paper.

REFERENCES

[1] H. T. Ahn and M. Shashkov, *Multi-material interface reconstruction on generalized polyhedral meshes*, J. Comput. Phys., 226 (2007), pp. 2096–2132.

[2] ———, *Adaptive moment-of-fluid method*, J. Comput. Phys., 228 (2009), pp. 2792–2821.
[3] F. Arbab, *Set models and Boolean operations for solids and assemblies*, IEEE computer graphics & Applications, 10 (1990), pp. 76–86.
[4] V. I. Arnold, *Ordinary Differential Equations*, MIT Press, 1973. ISBN: 0-262-51018-9.
[5] G. Barles and P. E. Souganidis, *Convergence of approximation schemes for fully nonlinear second order equations*, Asymp. Anal., 4 (1991), pp. 271–283.
[6] Y.-G. Chen, Y. Giga, and S. Goto, *Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations*, Proc. Japan Acad. Ser. A. Math. Sci., 65 (1989), pp. 207–210.
[7] ———, *Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations*, J. Diff. Geom., 33 (1991), pp. 749–786.
[8] Y.-G. Chen, Y. Giga, T. Hitaka, and M. Honma, *A stable difference scheme for computing motion of level surfaces by the mean curvature*, in Proceedings of the Global Research Analysis Center Symposium, D. Kim et al., ed., Seoul, Korea, 1994, pp. 1–19.
[9] M. G. Crandall, H. Ishii, and P.-L. Lions, *User's guide to viscosity solutions of second order partial differential equations*, Bull. Amer. Math. Soc., 27 (1992), pp. 1–67.
[10] M. G. Crandall and P.-L. Lions, *Convergent difference schemes for nonlinear parabolic equations and mean curvature motion*, Numerische Mathematik, 75 (1996), pp. 17–41.
[11] K. Deckelnick, *Error bounds for a difference scheme approximating viscosity solutions of mean curvature flow*, Interfaces and Free Boundaries, 2 (2000), pp. 117–142.
[12] K. Deckelnick, G. Dziuk, and C. M. Elliott, *Computation of geometric partial differential equations and mean curvature flow*, Acta Numerica, 14 (2005), pp. 139–232.
[13] V. Dyadechko and M. Shashkov, *Moment-of-fluid interface reconstruction*, Tech. Report LA-UR-05-7571, Los Alamos National Laboratory, 2005. `http://cnls.lanl.gov/~shashkov/`.
[14] ———, *Reconstruction of multi-material interfaces from moment data*, J. Comput. Phys., 227 (2008), pp. 5361–5384.
[15] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, *A hybrid particle level set method for improved interface capturing*, J. Comput. Phys., 183 (2002), pp. 83–116.
[16] L. C. Evans and P. E. Souganidis, *Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaccs equations*, Indiana Univ. Math. J., 33 (1984), pp. 773–797.
[17] L. C. Evans and J. Spruck, *Motion of level sets by mean curvature. I*, J. Diff. Geom., 33 (1991), pp. 635–681.
[18] M. Falcone and R. Ferretti, *Convergence analysis for a class of high-order semi-Lagrangian advection schemes*, SIAM J. Numer. Anal., 35 (1998), pp. 909–940.
[19] ———, *Semi-Lagrangian schemes for Hamilton-Jacobi equations, discrete representation formulae and Godunov methods*, J. Comput. Phys., 175 (2002), pp. 559–575.
[20] ———, *Consistency of a large time-step scheme for mean curvature motion*, in Numerical Mathematics and Advanced Applications, F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, eds., Springer Milan, 2003, pp. 495–502. ISBN:978-88-470-2167-9.
[21] M. Falcone and C. Makridakis, eds., *Numerical Methods for Viscosity Solutions and Applications*, vol. 59 of Advances in Mathematics for Applied Sciences, World Scientific, 2001. ISBN:978-981-02-4717-1.
[22] ———, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*, SIAM, 2014. ISBN: 978-1611973044.
[23] Y. Giga, *Surface Evolution Equations: A Level Set Approach*, vol. 99 of Monographs in Mathematics, Birkhäuser Basel, 2006. ISBN:078-3-7643-7391-7.
[24] Y. Giga and S. Goto, *Motion of hypersurfaces and geometric equations*, J. Math. Soc. Japan, 44 (1992), pp. 99–111.
[25] M. Grayson, *A short note on the evolution of surfaces via mean curvature*, Duke Math. J., 58 (1989), pp. 555–558.
[26] D. J. E. Harvie and D. F. Fletcher, *A new volume of fluid advection algorithm: The stream scheme*, J. Comput. Phys., 162 (2000), pp. 1–32.
[27] J. Helmsen, P. Colella, and E. G. Puckett, *Non-convex profile evolution in two dimensions using volume of fluids*, Tech. Report LBNL-40693, Lawrence Berkeley National Laboratory, 1997.
[28] M. Herrmann, *A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids*, J. Comput. Phys., 227 (2008), pp. 2674–2706.
[29] C. W. Hirt and B. D. Nichols, *Volume of fluid (VOF) method for the dynamics of free boundaries*, J. Comput. Phys., 39 (1981), pp. 201–225.
[30] V. Le Chenadec and H. Pitsch, *A 3D unsplit forward/backward volume-of-fluid approach and coupling to the level set method*, J. Comput. Phys., 233 (2013), pp. 10–33.
[31] J. López, J. Hernández, P. Gómez, and F. Faura, *A volume of fluid method based on*

multidimensional advection and spline interface reconstruction, J. Comput. Phys., 195 (2004), pp. 718–742.

[32] H. Meuer, E. Strohmaier, J. Dongarra, and H. Simon, *Top 500 supercomputer sites*. `http://www.top500.org`, June 2015.

[33] M. B. Nielsen and K. Museth, *Dynamic tubular grid: An efficient data structure and algorithms for high resolution level sets*, J. Sci. Comput., 26 (2006), pp. 261–299.

[34] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49.

[35] J. E. Pilliod, *An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods*, master's thesis, University of California, Davis, December 1992.

[36] E. G. Puckett, *A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction*, in Proceedings of the Fourth International Symposium on Computational Fluid Dynamics, H. Dwyer, ed., 1991, pp. 933–938.

[37] ———, *On the second-order accuracy of volume-of-fluid interface reconstruction algorithms: convergence in the max norm*, Communications in Applied Mathematics and Computational Science, 5 (2010), pp. 99–148.

[38] ———, *Second-order accuracy of volume-of-fluid interface reconstruction algorithms, II: An improved constraint on the cell size.*, Communications in Applied Mathematics and Computational Science, 8 (2013), pp. 123–158.

[39] W. J. Rider and D. B. Kothe, *Reconstructing volume tracking*, J. Comput. Phys., 141 (1998), pp. 112–152.

[40] A. Sard, *The measure of the critical values of differentiable maps*, Bull. Amer. Math. Soc., 48 (1942), pp. 883–890.

[41] R. Scardovelli and S. Zaleski, *Interface reconstruction with least-square fit and split Eulerian-Lagrangian advection*, Inter. J. Numer. Meth. Fluids, 41 (2003), pp. 251–274.

[42] J. A. Sethian, *Recent numerical algorithms for hypersurfaces moving with curvature-dependent speed: Hamilton-Jacobi equations and conservation laws*, J. Diff. Geom., 31 (1990), pp. 131–161.

[43] ———, *Level Set Methods and Fast Marching Methods*, vol. 3 of Cambridge Monograph on Applied and Computational Mathematics, Cambridge University Press, 1999. ISBN: 978-0521645577.

[44] M. Sussman, *A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles*, J. Comput. Phys., 187 (2003), pp. 110–136.

[45] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, *An adaptive level set approach for incompressible two-phase flows*, J. Comput. Phys., 148 (1999), pp. 81–124.

[46] M. Sussman and M. Y. Hussaini, *A discontinuous spectral element method for the level set equation*, J. Sci. Comput., 19 (2003), pp. 479–500.

[47] M. Sussman and M. Ohta, *High-order techniques for calculating surface tension forces*, Int. Ser. Numer. Math., 154 (2006), pp. 425–434.

[48] M. Sussman and E. Puckett, *A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows*, J. Comput. Phys., 162 (2000), pp. 301–337.

[49] R. B. Tilove and A. A. G. Requicha, *Closure of Boolean operations on geometric entities*, Computer-Aided Design, 12 (1980), pp. 219–220.

[50] G. Tryggvason, B. Bunner, D. Juric, W. Tauber, S. Nas, J. Han, N. Al-Rawahi., and Y.-J. Jan, *A front-tracking method for the computations of multiphase flow*, J. Comput. Phys., 169 (2001), pp. 708–759.

[51] N. J. Walkington, *Algorithms for computing motion by mean curvature*, SIAM J. Numer. Anal., 33 (1996), pp. 2215–2238.

[52] Y. Wang, S. Simakhina, and M. Sussman, *A hybrid level set-volume constraint method for incompressible two-phase flow*, J. Comput. Phys., 231 (2012), pp. 6348–6471.

[53] K. Yoneyama, *Theory of continuous set of points*, Tôhoku Mathematical Journal, 12 (1917), pp. 43–158.

[54] D. L. Youngs, *Time-dependent multi-material flow with large fluid distortion*, in Numerical Methods for Fluid Dynamics, K. W. Morton and M. J. Baines, eds., Academic Press, New York, 1982, pp. 273–285.

[55] Q. Zhang, *High-order, multidimensional, and conservative coarse-fine interpolation for adaptive mesh refinement*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 3159–68.

[56] ———, *Highly accurate Lagrangian flux calculation via algebraic quadratures on spline-approximated donating regions*, Comput. Methods Appl. Mech. Engrg., 264 (2013), pp. 191–204.

[57] ———, *On a family of unsplit advection algorithms for volume-of-fluid methods*, SIAM J. Numer. Anal., 51 (2013), pp. 2822–2850.
[58] ———, *On donating regions: Lagrangian flux through a fixed curve*, SIAM Review, 55 (2013), pp. 443–461.
[59] ———, *A fourth-order approximate projection method for the incompressible Navier-Stokes equations on locally-refined periodic domains*, Appl. Numer. Math., 77 (2014), pp. 16–30.
[60] ———, *GePUP: Generic projection and unconstrained PPE for fourth-order solutions of the incompressible Navier-Stokes equations with no-slip boundary conditions*, J. Sci. Comput., in press (2015). http://dx.doi.org/10.1007/s10915-015-0122-4.
[61] ———, *On generalized donating regions: Classifying Lagrangian fluxing particles through a fixed curve in the plane*, J. Math. Anal. Appl., 424 (2015), pp. 861–877.
[62] Q. Zhang and A. Fogelson, *Fourth-order interface tracking in two dimensions via an improved polygonal area mapping method*, SIAM J. Sci. Comput., 36 (2014), pp. A2369–A2400.
[63] Q. Zhang, H. Johansen, and P. Colella, *A fourth-order accurate finite-volume method with structured adaptive mesh refinement for solving the advection-diffusion equation*, SIAM J. Sci. Comput., 34 (2012), pp. B179–B201.
[64] Q. Zhang and P. L.-F. Liu, *A new interface tracking method: The polygonal area mapping method*, J. Comput. Phys., 227 (2008), pp. 4063–4088.