FOURTH-ORDER INTERFACE TRACKING IN TWO DIMENSIONS VIA AN IMPROVED POLYGONAL AREA MAPPING METHOD

QINGHAI ZHANG† AND AARON FOGELSON‡

Abstract. We present an improved PAM (iPAM) method as the first fourth-order interface tracking method whose convergence rates are independent of $C^1$ (derivative) discontinuities of the interface. As an improved version of the polygonal area mapping (PAM) method [Q. Zhang and P. L.-F. Liu, J. Comput. Phys., 227 (2008), pp. 4063–4088], the accuracy of the iPAM method is achieved via (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, (iii) adjusting the volume of represented cell material regions via polygon ear removal, and (iv) maintaining a relation ($h_L = r_L h^\alpha$) between the Eulerian grid size $h$ and the Lagrangian length scale that measures the distance between adjacent interface markers. Unlike volume-of-fluid methods and level-set methods, merging and separation of the tracked material is not automatically handled by the iPAM method. Instead, flexible subgrid resolutions are provided so that algorithmic behaviors of interface merging and separation can be determined from the specific physics of the problem under study. The iPAM method is a product of interdisciplinary research of different fields such as ordinary differential equations, computational geometry, and general topology. Its superior accuracy, efficiency, and versatility over previous interface tracking methods are demonstrated by a set of benchmark tests. In particular, for the vortex shear test on the $128 \times 128$ grid, the fourth-order iPAM method could be hundreds of times more efficient than state-of-the-art volume-of-fluid methods.

Key words. fourth-order interface tracking, PAM, the improved polygonal area mapping method, polygon ear removal, regular closed sets

AMS subject classifications. 76T99, 35L03, 34A26

DOI. 10.1137/140951886

1. Introduction. Interface tracking is a fundamental problem in numerically simulating multiphase flows. Although the interface is only a set of codimension one, the errors of the interface locus could propagate into the numerical solver from spatial discretization and deteriorate the overall accuracy. In addition, derived quantities of the interface such as its curvature often play an important role in the physics of multiphase flows. Hence the accuracy of numerical models for multiphase flows depends heavily on that of interface tracking.

Currently, there are three main families of interface tracking methods, namely, the front tracking method [26], the level-set method [21], and the volume-of-fluid (VOF) method [12]. Front tracking methods explicitly represent the interface as a set of connected markers while level-set methods implicitly represent the interface as the zero isocontour of a signed distance function. In comparison, VOF methods represent the interface inside a control volume both explicitly as a piecewise linear function and implicitly as the volume fraction of the tracked material. Instead of tracking the interface itself, VOF methods track volume fractions and material regions near the
interface by two substeps within each time step: in the first reconstruction substep
the new material regions are determined from the known volume fractions and in the
second advection substep the new volume fractions are calculated from the known
material regions.

A relatively new interface tracking method that evolved from VOF methods is
the polygonal area mapping (PAM) method [35], which represents material regions
explicitly as piecewise polygons, traces characteristic points on polygon boundaries
along pathlines, and calculates new material regions inside interface cells via polygon-
clipping algorithms in a discrete manner. As such, the PAM method combines the
best features of VOF methods and front tracking methods:

• the \( C^1 \) discontinuities of the interface are tracked naturally and accurately;
• the algorithmic steps can be directly applied to either structured rectangular
  grids or unstructured grids without any modifications;
• mass conservation can be enforced by adjusting the volume of material re-
  gions.

Numerical results in [35] suggest that the PAM method is superior to VOF meth-
ods in terms of accuracy. In a later paper [29], the PAM method is connected to VOF
methods via the donating region analysis [30, 27].

Despite their success, all current interface tracking methods are at best second-
order accurate. In addition, level-set methods are often criticized for lack of volume
conservation: both VOF methods and level-set methods tend to round \( C^1 \) disconti-
nuities of the interface, incurring accuracy deterioration. Other subtle issues such as
enforcing exact mass conservation might also cause the loss of one order of accuracy;
see [29] for a discussion in the context of VOF methods. Furthermore, calculated
curvatures are often of lower accuracy as curvature involves a second derivative of
the interface locus: sometimes the computed curvature does not even converge [23]!
Therefore, it is desirable to have a higher-order interface tracking method whose
accuracy does not degrade if there are \( C^1 \) discontinuities. A fourth-order VOF advect-
ion algorithm called DRACS is proposed in [29]; unfortunately, a fourth-order VOF
method is still unavailable due to the nonexistence of a fourth-order VOF reconstruc-
tion algorithm.

To answer the need of high-order interface tracking, we propose an improved PAM
(IPAM) method that is fourth-order accurate. This fourth-order accuracy is achieved
via

(i) augmenting the representation of material regions by allowing multiple simple
  polygons for a single cell;
(ii) removing the well-shapeness assumptions on simple polygons;
(iii) generalizing the volume-adjusting algorithms via a triangulation algorithm based
  on polygon ear removal;
(iv) actively maintaining a relation between the size of the Eulerian grid and the
  maximum distance between two adjacent markers on the interface.

The rest of this paper is organized as follows. Section 2 contains preliminaries
and notation from different fields. In section 3, we briefly review the PAM method
to establish the foundation for discussing the iPAM method. Then the specific im-
provements necessary to achieve the fourth-order accuracy are explained in detail in

---

\(^1\) “Volume conservation” is the technically correct term for level-set methods as they apply to both
incompressible and compressible flows. In contrast, it is customary to use “mass conservation” for
VOF methods since almost all of them are designed for incompressible fluids with constant density.
Since this work is restricted to incompressible flows, we use mass conservation, hereafter, to follow
the VOF convention.
section 4. The high-order convergence rates of the iPAM method and its efficiency are demonstrated by a set of popular benchmark tests in section 5. Section 6 concludes with several prospects for future research.

2. Preliminaries and notation. This section contains elementary definitions and notation from different fields that are necessary for a precise and self-contained exposition. For clarity, the following notational conventions are adopted in this work:

- \( p(t) \) denotes the location of a moving particle at time \( t \);
- lowercase letters \( p, q \) without the function forms are used when the dependence on time is irrelevant or unimportant;
- uppercase calligraphic letters such as \( \mathcal{P}, \mathcal{Q}, \) and \( \mathcal{M} \) are reserved for point sets, which default to regular closed sets unless otherwise specified;
- “\( \cup \)”, “\( \cap \)”, and “\( \setminus \)” respectively, denote the conventional set-theoretic binary operations of union, intersection, and difference.

2.1. Regular closed sets. In the \( d \)-dimensional Euclidean space \( \mathbb{R}^D \), a point set \( \mathcal{P} \) is open if for each point \( x \in \mathcal{P} \) there exists an open ball \( \mathcal{B}_r(x) := \{ y : \|x - y\|_2 < r \} \) with radius \( r > 0 \) such that \( \mathcal{B}_r(x) \subseteq \mathcal{P} \). A point set \( \mathcal{P} \) is bounded if there exists some open ball with finite radius that covers \( \mathcal{P} \). The complement of \( \mathcal{P} \subseteq \mathbb{R}^D \), written \( \mathcal{P}' \), is the set \( \mathbb{R}^D \setminus \mathcal{P} \). \( \mathcal{P} \) is closed if its complement is open. The closure of \( \mathcal{P} \), written \( \mathcal{P}^- \), is the intersection of all closed supersets of \( \mathcal{P} \). The interior of \( \mathcal{P} \), written \( \mathcal{P}^o \), is the union of all open subsets of \( \mathcal{P} \). The exterior of \( \mathcal{P} \), written \( \mathcal{P}^+ := \mathcal{P}^\circ := (\mathcal{P}')^o \), is the interior of its complement. A point \( x \in \mathbb{R}^D \) is a boundary point of \( \mathcal{P} \) if \( x \notin \mathcal{P}^\circ \) and \( x \notin \mathcal{P}^- \). The boundary of \( \mathcal{P} \), written \( \partial \mathcal{P} \), is the set of boundary points of \( \mathcal{P} \). It can be shown that \( \mathcal{P}^\circ = \mathcal{P} \setminus \partial \mathcal{P} \) and \( \mathcal{P}^- = \mathcal{P} \cup \partial \mathcal{P} \).

A regular closed set \( \mathcal{P} \) is a closed set which coincides with the closure of its own interior (\( \mathcal{P} = \mathcal{P}^\circ \)). It can be shown that \( \mathcal{Q}^\circ \) is a regular closed set for any \( \mathcal{Q} \subseteq \mathbb{R}^D \). Define regularized Boolean operations \( \mathcal{P}' := \mathcal{P}^\circ \), \( \mathcal{P} \cap' \mathcal{Q} := (\mathcal{P} \cap \mathcal{Q})^* \), \( \mathcal{P} \cup' \mathcal{Q} := (\mathcal{P} \cup \mathcal{Q})^* \), and \( \mathcal{P} \setminus' \mathcal{Q} := (\mathcal{P} \setminus \mathcal{Q})^* \) for all \( \mathcal{P}, \mathcal{Q} \subseteq \mathbb{R}^D \). Then the class of all regular closed sets together with \( \cup' \), \( \cap' \), \( \setminus' \), \( \mathcal{P}^o \), and \( \mathbb{R}^D \) form a Boolean algebra [15]. This well-known result serves as the theoretical foundation for Boolean algorithms on regular closed sets in solid modeling [24]. A useful Boolean operation on regular closed sets is the symmetric difference or exclusive disjunction:

\[
(2.1) \quad \mathcal{P} \oplus \mathcal{Q} := (\mathcal{P} \cap' \mathcal{Q}') \cup' (\mathcal{P}' \cap \mathcal{Q}).
\]

2.2. Flow maps. The nonautonomous ordinary differential equation (ODE)

\[
(2.2) \quad \frac{dp(t)}{dt} = u(p(t), t)
\]

has a unique solution for any given initial time and position so long as \( u \) is continuous in time and Lipschitz continuous in space. This uniqueness admits a flow map \( \phi : \mathbb{R}^D \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^D \),

\[
(2.3) \quad \phi^+_\tau(p(t_0)) := p(t_0 + \tau), \quad \phi^-_\tau(p(t_0)) := p(t_0 - \tau),
\]

where the three independent variables \( t_0, p(t_0), \) and \( \tau \) 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 (2.2) 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 fixed \( \tau \) is a homeomorphism, i.e., a continuous bijective function that preserves topological properties. For convenience we will also use \( \overrightarrow{\cdot} \) and \( \overleftarrow{\cdot} \) as synonym notation for \( \phi^+_{t_0} \) and \( \phi^-_{t_0} \),

\[
\begin{align*}
\left\{ \begin{array}{l}
 p(t_0, \tau) := & p(t_0 + \tau) = p(t_0) + \int_{t_0}^{t_0 + \tau} u(p(t), t) \, dt, \\
 q(t_0, \tau) := & p(t_0 - \tau) = p(t_0) - \int_{t_0}^{t_0 - \tau} u(p(t), t) \, dt,
\end{array} \right.
\end{align*}
\]

where the time increment \( \tau \) is always positive. This flow map also applies to a point set \( S \):

\[
(2.5) \quad \phi^\pm_{t_0} (S) := \left\{ \phi^\pm_{t_0}(q) : q(t_0) \in S \right\},
\]

where \( \phi^\pm_{t_0} (S) \) and \( \phi^-_{t_0} (S) \) are the image and preimage of \( S \) at time \( t_0 + \tau \) and \( t_0 - \tau \), respectively; they are both homeomorphic to \( S \),

\[
(2.6) \quad S \cong \phi^+_{t_0} (S) \cong \phi^-_{t_0} (S),
\]

because each flow map with fixed \( \tau \) is a homeomorphism.

A pathline is the curve generated by following a single particle in a time period, written \( \Phi^{\pm k}_{t_0}(p) := \{ \phi^\pm_{t_0}(p) : \tau \in (0,k) \} \).

### 2.3. Jordan curves

We define an open curve as the image of a continuous map \( \gamma : (0,1) \to \mathbb{R}^2 \); it is simple if the map \( \gamma \) is injective. A closed curve is the image of a continuous map \( \gamma : [0,1] \to \mathbb{R}^2 \) satisfying \( \gamma(0) = \gamma(1) \). Although \( \gamma \) denotes the map of a curve, we will also use it to refer to the curve itself.

A closed curve \( \gamma \) is simple closed or Jordan if the restriction of \( \gamma \) to \( [0,1] \) is injective. The Jordan curve theorem states that the complement of a Jordan curve \( \gamma \) in the plane \( \mathbb{R}^2 \) consists of two components, each of which has \( \gamma \) as its boundary; one component is bounded and the other is unbounded. The bounded component is also simply connected in that any Jordan curve inside it can be continuously shrunk to a point in the set.

The orientation of a curve \( \gamma \) is the assignment of a direction in which \( \gamma \) is traversed. A Jordan curve \( \gamma \) is positively oriented if the bounded component of \( \mathbb{R}^2 \setminus \gamma \) always lies to the left of an observer who traverses the curve according to its orientation; otherwise it is negatively oriented.

### 2.4. Simple polygons

Let \( p_0, p_1, \ldots, p_{n_P-1} \) be \( n_P \) points in the plane, connect them by \( n_P \) line segments (or more precisely \( n_P \) open linear curves) \( p_0p_1, p_1p_2, \ldots, p_{n_P-1}p_0 \) to form a closed curve \( \gamma_P \). The polygon \( P \) generated by such a closed curve \( \gamma_P \) is the region enclosed by \( \gamma_P \); this closed curve is called the generating curve of \( P \). The points \( p_i \) are called the vertices of \( P \) and the open linear curves \( p_ip_{i+1} \) the edges of \( P \); the number of vertices and edges is the same and is denoted by \( n_P \). A polygon is a simple polygon if its generating curve is Jordan.

A vertex and the two adjacent edges in a polygon define two angles, the one that intersects the interior of the polygon in a local neighborhood of the vertex is called the internal angle. A vertex is concave if its internal angle is greater than \( \pi \), otherwise it is convex. A polygon \( P \) is convex if \( p, q \in P \) and \( \alpha \in [0,1] \) imply \( \alpha p + (1-\alpha)q \in P \). The convex hull of a point set \( P \) is the smallest convex polygon that contains \( P \).

In this work, the vertices of a polygon \( P \) are indexed by the cyclic convention as \( i \mod n_P \), i.e., \( p_{n_P} = p_0, p_{n_P+1} = p_1 \), and so on.
2.5. **Diagonals and ears of polygons.** The diagonal of a polygon is a fundamental concept of computational geometry and is used in solving many practical problems such as polygon triangulation [18] and the classical art gallery problem [3,9].

**Definition 2.1.** Let $p_i$ and $p_j$ be two different vertices of a polygon $P$, the open line segment $p_ip_j$ is an internal diagonal (or simply a diagonal) of $P$ if $p_ip_j \cap P = p_ip_j$, or an external diagonal if $p_ip_j \cap P = \emptyset$.

See Figure 1 for examples of polygon diagonals. For a positively oriented polygon $P$, $p_ip_j$ is an internal diagonal of $P$ if the first and at least one of the other of the following conditions is met:

- **(Diag-1)** $p_ip_j \cap p_{\ell-1}p_{\ell} = \emptyset$ for all $\ell = 1, 2, \ldots, np$;
- **(Diag-2)** ray $\overrightarrow{p_ip_j}$ is inside the cone swept counterclockwise from $\overrightarrow{p_ip_{i+1}}$ to $\overrightarrow{p_ip_{i-1}}$;
- **(Diag-3)** ray $\overrightarrow{p_jp_i}$ is inside the cone swept counterclockwise from $\overrightarrow{p_jp_{j+1}}$ to $\overrightarrow{p_jp_{j-1}}$.

**(Diag-1)** makes $p_ip_j$ a diagonal, internal or external; **(Diag-2)** or **(Diag-3)** further restrict the diagonal to be internal, as shown in Figure 2. Only one of **(Diag-2)** and **(Diag-3)**, together with **(Diag-1)**, is sufficient to ensure that $p_ip_j$ is an internal diagonal. In an implementation, however, usually both **(Diag-2)** and **(Diag-3)** are checked before checking **(Diag-1)** since checking each costs only $O(1)$ time whereas checking **(Diag-1)** costs $O(np)$ time. Let $\text{Left}(p_ip_j, q)$ be the predicate that $q$ is on the left side of $\overrightarrow{p_ip_j}$. It can be shown [20, p. 35] that **(Diag-2)** is true if two or more of
the predicates of \(\text{Left}(\vec{p}_i p_{i+1}, p_j)\), \(\text{Left}(\vec{p}_i p_{i-1}, p_{i-1})\), and \(\text{Left}(\vec{p}_i p_{i-1}, p_{i+1})\) are true;² see Figure 2 for an illustration.

**Definition 2.2.** The triangle formed by three consecutive vertices \(p_{i-1}, p_i, \) and \(p_{i+1}\) is called an ear at \(p_i\) if \(p_{i-1} p_{i+1}\) is an internal diagonal. Two ears are said to be nonoverlapping if their interiors are disjoint.

For the existence of ears we quote the following theorem.

**Theorem 2.3 (two ears theorem [18]).** Except for triangles, every simple polygon has at least two nonoverlapping ears.

### 2.6. The interface tracking problem.

In VOF methods and the PAM method, a color function is defined for the tracked material \(M\) as

\[
f(x, t) := \begin{cases} 1 & \text{if there is } M \text{ at } (x, t), \\ 0 & \text{otherwise,} \end{cases}
\]

which naturally leads to a point-set formulation of the interface tracking problem.

**Definition 2.4 (the interface tracking problem).** Denote the region occupied by the tracked material \(M\) at time \(t\) by

\[
M(t) := \{x : f(x, t) = 1\}.
\]

The interface tracking problem is the determination of \(M(T)\) from the initial condition \(M(t_0)\) and the given velocity field \(u(x, t)\), \(t \in [t_0, T]\).

To ensure that Definition 2.4 is well-posed, we assume that the velocity is continuous in \([t_0, T]\), Lipschitz continuous in space, and that the color function is constant for any Lagrangian particle passively advected by the unsteady flow (2.2):

\[
\frac{\partial f}{\partial t} + u \cdot \nabla f = 0.
\]

In addition, we also assume that \(M(t)\) is bounded and \(\partial M(t)\) is piecewise smooth so that the volume of \(M(t)\) can be defined as

\[
\|M(t)\| := \left| \int_{M(t)} dx \right|.
\]

A measure of interface tracking errors proposed in [29] is

\[
E_1(t_n) := \|M(t_n) \oplus M^n\|,
\]

where \(M(t_n)\) denotes the exact material region at \(t_n\) and \(M^n\) its approximation by an interface tracking method. It can be shown that the class of regular closed sets together with \(\|P \oplus Q\|\) forms a metric space.

**Definition 2.5 (accuracy of an interface tracking method [29]).** In numerically solving the interface tracking problem posed in Definition 2.4, an interface tracking method \(\Sigma_{\text{IT}}\),

\[
M^{n+1} = \Sigma_{\text{IT}}(M(t_n), u) = M(t_n + k),
\]

is \(\beta\)-th-order accurate in the \(1\)-norm if, as \(k \to 0\), \(E_1(T) = O(k^\beta)\). \(\Sigma_{\text{IT}}\) is said to be consistent if \(\beta > 0\).

²Calculate a cross product vector \(v = \vec{p}_i p_{i+1} \times \vec{p}_i p_{j}\). The predicate \(\text{Left}(\vec{p}_i p_{i+1}, p_j)\) is true if \(v\) points out of the paper and false otherwise.
3. The PAM method. For ease of exposition the computational domain $\Omega$ is hereafter assumed to be a rectangle partitioned by a uniform grid into square control volumes or cells. Let $h$ denote the uniform grid size. The time step size $k$ is assumed to be uniform ($t_n = t_0 + nk$) and it is chosen so that the Courant number $C_r := \frac{k\|u\|_{\infty}}{h}$ is a constant.\footnote{The PAM method can be regarded as a semi-Lagrangian method. As discussed in [28, p. 201], the Courant number can be bigger than 1 so long as the preimage of each cell is simply connected.}

A cell material region is the intersection of the cell $C$ and the material region, written $\mathcal{M}_C(t_n) := \mathcal{M}(t_n) \cap C$. A cell satisfying $\mathcal{M}_C(t_n) = \emptyset$ or $\mathcal{M}_C(t_n) = C$ 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 $t_n + k$. If the Courant number does not exceed 1 and the cells are rectangles, the set of interface candidate cells consists of all interface cells and their edge/corner neighbors. The set of interface candidate cells for unstructured grids can be determined in a straightforward manner from Definition 4.1. The volume fraction at time $t_n$ of the tracked material inside a cell $C$ is the ratio of the volume of $\mathcal{M}_C(t_n)$ to that of $C$, i.e.,

$$\langle f \rangle_C(t_n) := \frac{||\mathcal{M}_C(t_n)||}{||C||}.$$ 

3.1. Representation of material regions. In the PAM method, material regions are modeled as compact sets (or more precisely bounded regular closed sets) since physically meaningful material regions do not contain components of a lower dimension such as isolated points and curves.

An abstract data structure called a set of simple polygons (SSP),

$$\mathcal{S}_{\mathcal{M}_C} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{s_p}, \mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_{s_h}\},$$

is used to approximate the material region inside an interface cell as

$$\mathcal{M}_C(t) = \left( \bigcup_{i=1}^{s_p} \mathcal{P}_i \right) \setminus \left( \bigcup_{j=1}^{s_h} \mathcal{H}_j \right),$$

where $\mathcal{M}_C(t) \approx \mathcal{M}_C(t_n)$, $s_p$ and $s_h$ are the numbers of simple polygons $\mathcal{P}_i$ and $\mathcal{H}_j$ with counterclockwise and clockwise orientations, respectively. The simple polygons are chosen so that

$$\begin{cases} 
  i \neq j \Rightarrow \mathcal{P}_i \cap \mathcal{P}_j = \emptyset; \\
  i \neq j \Rightarrow \mathcal{H}_i \cap \mathcal{H}_j = \emptyset; \\
  \forall \mathcal{H}_j \in \mathcal{S}_{\mathcal{M}_C}, \exists \mathcal{P}_j \in \mathcal{S}_{\mathcal{M}_C} \text{ s.t. } \\
  \mathcal{H}_i \subset \mathcal{P}_j \text{ and } \mathcal{P}_j \setminus \mathcal{H}_i \text{ is not a simple polygon.}
\end{cases}$$

(3.4)

In other words, simple polygons of the same orientation are pairwise disjoint and any "hole" polygon is properly contained in some "positive" polygon. The construction of an SSP from the exact cell material region is illustrated in Figure 3.

As a benefit of (3.4), the total volume of $\mathcal{M}_C(t)$ is easily calculated by

$$\|\mathcal{M}_C(t)\| = \sum_{i=1}^{s_p} \|\mathcal{P}_i\| - \sum_{j=1}^{s_h} \|\mathcal{H}_j\|,$$

and the volume of a simple polygon with vertices $p_0, p_1, \ldots, p_{n_p-1}$ is

$$\|\mathcal{P}\| = \frac{1}{2} \left| \sum_{j=0}^{n_p-3} (p_{j+1} - p_0) \times (p_{j+2} - p_{j+1}) \right|,$$

where "\(\times\)" denotes the cross product of two vectors and "\(|\cdot|\)" the length of a vector.
Approximating cell material regions $M_C(t_0)$ with an SSP $M_0^C$. The dotted lines represent the exact interface $\partial M(t_0)$ inside a cell $C$. After decomposing $\partial M(t_0)$ into the union of a set of Jordan curves, a number of points (solid dots) are selected on each Jordan curve as the vertices of the corresponding simple polygons in the SSP $\{P_1, P_2, H_1\}$. The shaded areas represent the approximated material region $P_1 \cup P_2 \setminus H_1$ by the SSP.

Assumption 3.1 (well-shapeneness of simple polygons by PAM). In the PAM method, a simple polygon is assumed to be well-shaped so that the open triangle $\Delta(p_i, p_{i-1}, p_{i+1})$ formed by any three consecutive vertices $p_i, p_{i-1}, p_{i+1}$ is either a subset of the polygon or completely outside of the polygon. More precisely, we assume

$$\text{(3.7)} \begin{cases} \text{for each convex vertex } p_i \text{ of } P, & \Delta(p_i, p_{i-1}, p_{i+1}) \subseteq P, \\ \text{for each concave vertex } p_i \text{ of } P, & \Delta(p_i, p_{i-1}, p_{i+1}) \cap P = \emptyset. \end{cases}$$

The above assumption holds if the two neighbors of every convex vertex form an internal diagonal and the two neighbors of every concave vertex form an external diagonal. Not all simple polygons are well-shaped; for example, $p_0$ in Figure 1(a) violates the first line of (3.7) and $p_0$ in Figure 1(b) violates the second line of (3.7). Nonetheless, Assumption 3.1 facilitates algorithms that expand and shrink simple polygons (Algorithms 2 and 3 in [35]) in order to adjust the volume of an SSP.

Equations (3.2), (3.3), (3.4), and Assumption 3.1 are collectively known as the PAM representation invariants. For each interface cell, the PAM representation invariants are guaranteed to be true at the end of each time step by the following algorithm, which also adjusts the volume of an SSP to match that of its preimage for incompressible flows.

**Definition 3.2 (volume-adjusting in the PAM (VAPAM)).** Denote by $n_{\text{max}}$ the maximal number of vertices allowed in a single SSP and $v_C$ the target volume of $M_{C}^{n+1}$. The PAM method adjusts $M_{C}^{n+1}$ as follows:

1. **(VAPAM-1)** delete all the hole polygons $H_j$’s from $M_{C}^{n+1}$;
2. **(VAPAM-2)** replace $M_{C}^{n+1}$ by its convex hull if $s_p > 1$ or (3.4) or (3.7) is violated;
3. **(VAPAM-3)** adjust $M_{C}^{n+1}$ by [35, Algorithms 2 and 3] so that $\|M_{C}^{n+1}\| = v_C$;
4. **(VAPAM-4)** if $M_{C}^{n+1}$ has more than $n_{\text{max}}$ vertices, keep deleting the least significant vertex of $M_{C}^{n+1}$ until it only has $n_{\text{max}}$ vertices, then repeat (VAPAM-3) so that $\|M_{C}^{n+1}\| = v_C$.

Here, the significance of a vertex $p_i$ is measured by $\|\Delta(p_i, p_{i+1}, p_{i-1})\|$.

The interface $\partial M$ is a set of codimension one. If the Eulerian grid size $h$ is small relative to the size of material regions, then for most cells $M_C$ is either an empty set or the control volume itself. To save computational resources, the type of each pure cell
is labeled with an integer, with the bookkeeping data structures of $\mathcal{M}_n$ instantiated only for interface cells. As a major difference from VOF methods, the PAM method always deduces the volume fractions of interface cells from cell material regions by (3.1); in other words, the volume fractions are never used in determining the interface.

3.2. Algorithmic steps. At the initial time $t_0$, the type of each cell is determined from $\mathcal{M}(t_0)$ and partition of the domain. The cell material region $\mathcal{M}_C(t_0)$ is approximated by an SSP for each interface cell, as illustrated in Figure 3. The name of the PAM method comes from the fact that the discrete flow map is applied to the polygonal areas of the tracked material in subsequent time steps.

Definition 3.3 (the PAM method). Let $\hat{\varphi}$ denote a discrete flow map approximating $\varphi$ and $\mathcal{M}_C^n$ the SSP representing $\mathcal{M}_C(t_n)$. Within the time step $[t_n, t_n + k]$, the PAM method advances $\mathcal{M}_C^n$ to $\mathcal{M}_C^{n+1}$ in each interface candidate cell $C$ by

- (PAM-1) calculating the cell preimage $\hat{\varphi}^{-k}_{t_n+k}(C)$,
- (PAM-2) computing the intersection $\mathcal{M}_C^{n+1} = \hat{\varphi}^{-k}_{t_n+k}(C) \cap^* \mathcal{M}^n$,
- (PAM-3) tracing the intersection forward to obtain $\mathcal{M}_C^{n+1}$,
- (PAM-4) calling the VAPAM algorithm in Definition 3.2 to fulfill the PAM representation invariants and mass conservation so that

$$\|\mathcal{M}_C^{n+1}\| = \|\mathcal{M}_C^{n+1}\|.$$

As illustrated in Figure 4, the cell preimage $\hat{\varphi}^{-k}_{t_n+k}(C)$ obtained in (PAM-1) contains all the particles to be advected into $C$ at the end of the time step. Note that the preimage may consist entirely, partly, or not at all of material $\mathcal{M}$. On the other hand, $\mathcal{M}_C^n$ contains all the particles of material $\mathcal{M}$ at the beginning of the time step, hence $\hat{\varphi}^{-k}_{t_n+k}(C) \cap^* \mathcal{M}^n$ yields all the particles of material $\mathcal{M}$ that will be inside $C$ at $t_n + k$. The image of the intersection is thus $\mathcal{M}_C^{n+1}$.

A cell preimage is obtained by tracing the preimage of its boundary. Furthermore, the tracings are consistent for neighboring cells in that the preimage of a cell edge is used simultaneously for the two cells whose boundaries contain the same edge. Consequently, the first two steps (PAM-1) and (PAM-2) can be interpreted as a repartition of $\mathcal{M}_C^n$ for global mass conservation. In contrast, the last step (PAM-4) attempts to enforce local mass conservation for each interface candidate cell.

Given input parameters of a convex cell polygon $C$, a simple polygon $\mathcal{P}$, and a proposed volume change, Algorithms 2 and 3 in (VAPAM-3) expand or shrink $\mathcal{P}$ to fulfill the volume change. These algorithms are designed to be “total number of vertices diminishing” (TNVD), an analogy to “total variation diminishing” (TVD). See [35, Figures 7 and 8] for illustrations of these TNVD algorithms and [35, Figure 5] for an example of the VAPAM steps. The benefits of TNVD and VAPAM are twofold: on the one hand, local mass conservation is achieved by the volume adjustment; on the other hand, numerical stability is ensured by reducing the total number of vertices in each SSP. From a physical viewpoint, TNVD and VAPAM are analogous to adding surface tension to the tracked interface.

Upon the completion of (VAPAM-4), only one simple polygon remains in an SSP. However, two or more polygons are allowed in a material SSP during the steps

$^4$In the PAM method, the time integration is approximated by the classical fourth-order Runge–Kutta method; $\hat{\varphi}$ is also discrete in space since it is applied not to all points in the boundary of an SSP, but only to the vertices of the SSP.
Fig. 4. The polygonal area mapping method. Subplots (a), (b), (c), and (d) correspond to the four substeps of the PAM method. Solid dots (●) and hollow dots (○) represent image points and preimage points, respectively. Dashed lines denote pathlines. The solid square dots in subplot (b) represent new interface markers added from intersecting cell preimage with the ambient material region. The images of these markers are \( q_2 \) and \( q_4 \) in subplots (c) and (d). This figure is adapted from [35, Figure 4].

(PAM-2) and (PAM-3) so that merging of two separate material regions is handled satisfactorily; see [35, Figure 9] for an illustration.

The global constant \( n_{\text{max}} \) can be prescribed by the user to control the computational resources consumed by SSPs. This free parameter also affords a cost-effective mechanism for adaptive interface refinement: the resolution of the interface can be improved by increasing \( n_{\text{max}} \) without refining the grids for the main flow. More importantly, the free parameter \( n_{\text{max}} \) yields the flexibility of separating the velocity length scale from the interface length scale. This advantage of the PAM method is exploited by a hybrid continuum-particle model [36] in decomposing a single phase into a continuum zone and a particle zone, aided by the concept of biconnectedness.
in graph theory. Such a decomposition algorithm based on VOF methods would be self-contradictory because of the tight coupling of the grid size to the interface in the reconstruction substep, where the interface normal vector is determined by volume fractions of nearby cells.

4. An improved PAM method. Hereafter an interface marker refers to a vertex of some SSP (set of simple polygons) that is on the interface $\partial M^n$; $n_{\text{mks}}(t)$ denotes the total number of them at time $t$. The Lagrangian length scale of the approximated interface is the maximum distance between two adjacent interface markers:

$$(4.1) \quad h_L(\partial M^n) := \max_{p_j, p_{j+1} \subset \partial M^n} \| p_j - p_{j+1} \|_2.$$ 

Clearly each interface edge $p_jp_{j+1}$ is on the boundary of some simple polygon in an SSP. We will use $h_L$ as a shorthand when $\partial M^n$ is clear from the context. The total length of the interface is

$$(4.2) \quad L_{\partial M^n} := \| \partial M^n \|.$$ 

The error of the PAM method consists mainly of three parts: the representation error $E_{\text{REP}}$ from approximating piecewise smooth curves with linear segments, the integration error $E_{\text{ODE}}$ from replacing the flow map by a numerical integrator, and the volume adjustment error $E_{\text{ADJ}}$ from applying the VAPAM algorithm in Definition 3.2.

Three key observations follow:

- $E_{\text{ODE}} = O(h^4)$ because time integration is approximated by the classical fourth-order Runge–Kutta method;
- the volume difference between $M_{C}^{n+1}$ and $\frac{1}{M_{C}}^{n+1}$ in Definition 3.3 is proportional to the local temporal truncation error of the discrete flow map, hence the volume-adjusting error caused by (VAPAM-3) is $E_{\text{ADJ}} = O(h^4)$;
- a linear representation always has $E_{\text{REP}} = O(h_2^2)$; however, if we enforce a relation $h_L = r_h h^n$ with $\alpha > 1$ and $r_h$ a constant, then $E_{\text{REP}} = O(h^{2\alpha})$.

To improve the PAM method to fourth-order accuracy, we will have to resolve two difficulties. First, the relation $h_L = r_h h^n$ between the Lagrangian length scale and the Eulerian grid size will have to be enforced in the algorithmic steps with $\alpha = 2$. Second, the deletion of the hole polygons in (VAPAM-1) and the invocation of convex hull algorithms in (VAPAM-2) have to be avoided since each of these substeps incurs an error of $O(h^2)$, which would deteriorate the fourth-order accuracy.

In section 4.1, we augment and simplify the representation of material regions by SSPs, then in section 4.2 we discuss a volume-adjusting algorithm that is free of any assumptions on the shape of the simple polygons; the purpose of these two sections is to avoid (VAPAM-1) and (VAPAM-2). In section 4.3, we elaborate the iPAM algorithmic steps, emphasizing how to enforce the relation $h_L = r_h h^n$ efficiently.

4.1. Representation of material regions. At the end of each time step in the PAM method, the positively oriented polygon with the largest volume is retained and all other polygons are deleted from the SSP of a cell; the volume of the remaining polygon is then adjusted to conserve mass. These simplifications correspond to requiring $s_p = 1$ and $s_h = 0$ in (3.2) at the end of each time step. We have found

$^{5}$In [33], the representation error is defined as $E_{\text{REP}} := \| \phi_{T_{0}}^{T_{t}}(M(t_0)) \oplus \phi_{T_{0}}^{T_{t}}(M^n) \|$, where $\phi$ is the exact flow map. In incompressible flows, the total volume of the material region remains constant. Hence in this work the representation error is solely caused by the initial representation of $M(t_0)$: $E_{\text{REP}} = \| M(t_0) \oplus M^n \|$.
these requirements to be too restrictive. For example, they result in spurious merging of separate material polygons within one cell even if the velocity field does not imply it; see Figure 10 in section 5.2 for two examples. Even in the case of bona fide merging, the accuracy depends largely on the Courant number and the partition of the computational domain [35, Figure 9] [36].

The new iPAM method allows \( s_p \geq 1 \) in (3.2) and requires that \( s_k = 0 \), i.e., there are no negatively oriented polygons. If \( M^n_C \) indeed contains a hole, \( M^n_C \) is represented as the union of multiple positively oriented simple polygons.

To sum up, the iPAM method approximates a cell material region \( M^n_C \) as the union of a set of simple polygons whose interiors are pairwise disjoint:

\[
\begin{align*}
M^n_C & = \bigcup_{i=1}^{s_p} M^n_{C,i}, \\
\text{each } M^n_{C,i} & \text{ is a simple polygon,} \\
s_p & \text{ is the minimum number of simple polygons required for representing } M^n_C, \\
i \neq j & \iff (M^n_{C,i})^o \cap (M^n_{C,j})^o = \emptyset.
\end{align*}
\]

The conditions in (4.3) are called the iPAM representation invariants; it follows that

\[
\|M^n_C\| = \sum_{i=1}^{s_p} \|M^n_{C,i}\|.
\]

With double-precision floating-point arithmetic, we set \( M^n_C = \emptyset \) if \( \langle f \rangle_C < 10^{-12} \). It is emphasized that (4.3) obviates the need for removing hole polygons and invoking convex hull approximations for multiple simple polygons.

4.2. Volume adjusting via polygon ear removal (VAPER). The volume-adjusting algorithms of the PAM method rely heavily on Assumption 3.1, the violation of which triggers the convex hull approximation. The purpose of this subsection is to explain the VAPER algorithm that removes this well-shapeness assumption.

Definitions 2.1 and 2.2, and Theorem 2.3 imply that if an ear at \( p_i \) is removed from a simple polygon \( P \), then the remaining polygon is also simple and has one less vertex than \( P \). This observation naturally leads to an algorithm for converting a polygon to a triangle: keep locating an ear of \( P \) and removing it until the remaining polygon is a triangle.

As the basic building block of VAPER, the procedure\(^7\) ReduceEar shrinks a simple polygon by removing or reducing one of its ears. If the specified volume decrease is larger than the volume of the ear, the ear tip \( p_i \) is removed from \( P \); otherwise \( p_i \) is moved to accommodate the volume decrease. These two cases correspond to Figure 5(b) and (c), respectively.

Shrinking a simple polygon by ReduceEar is made precise in Algorithm 2, where the priority of the \( r \)th vertex of \( P \) with respect to \( C \) is defined as

\[
r_P(C, i) := \begin{cases} 
\|\Delta(p_i, p_{i-1}, p_{i+1})\| & \text{if } p_i \notin \partial C \text{ and } p_i \text{ is an ear tip,} \\
\infty & \text{otherwise.}
\end{cases}
\]

To avoid generating spurious interface or \( C^0 \) interface discontinuities,\(^8\) a vertex of a material polygon is movable only if it is an ear tip and it is also in the interior of the

---

\(^6\)Such an algorithm is known as polygon triangulation via ear clipping in computational geometry.

\(^7\)A procedure differs from an algorithm in that it does not solve a concrete problem, but is rather an important subroutine repeatedly used by an algorithm.

\(^8\)In the PAM method, vertices on cell edges are movable, which might cause discontinuities of the interface across cell boundaries.
FOURTH-ORDER INTERFACE TRACKING BY iPAM

(a) the input polygon.

(b) \(-\delta \geq \|\Delta(p_i, p_{i-1}, p_{i+1})\|\).

(c) \(-\delta < \|\Delta(p_i, p_{i-1}, p_{i+1})\|\).

Fig. 5. Ear removal of a simple polygon: (a) is the input polygon to ReduceEar\((P, i, i+1, \delta)\), (b) and (c) correspond to lines 3, 4 and lines 6, 7 in Procedure ReduceEar, respectively.

**Procedure** Reduc\(e\)Ear\((P, i, j, \delta)\)

**Data:** 
- \(p_i\) is an ear tip of a simple polygon \(P\);
- \(j = i \pm 1\) is the index of the vertex toward which \(p_i\) is moved;
- \(\delta \leq 0\) is the desired volume change of \(P\).

**Result:** \(p_i\) is either removed from \(P\) or changed; the actual volume change of \(P\) is returned.

1. \(a \leftarrow -\delta / ||\Delta(p_i, p_{i+1}, p_{i-1})||\)
2. if \(a > 1\) then
3. remove \(p_i\) from \(P\)
4. return \(-\|\Delta(p_i, p_{i+1}, p_{i-1})\|\)
5. else
6. \(p_i \leftarrow ap_j + (1 - a)p_i\)
7. return \(\delta\)
8. end

**Algorithm 1:** Shrink a simple polygon

**Data:** a simple polygon \(P\), its cell polygon \(C \supset P\), and a volume decrease \(\delta \in [-\|P\|, 0)\)

**Result:** change \(P\) to \(P'\) s.t. \(\|P'\| = \|P\| + \delta\), \(P' \subset P\)

1. initialize \(r_P(C, i) \forall i = 0, 1, \ldots, n_P - 1\) by (4.5)
2. \(r_P(C, i_{\text{min}}) \leftarrow \min r_P(C, i)\)
3. while \(\delta \neq 0\) and \(r_P(C, i_{\text{min}}) < \infty\) do
4. \(j \leftarrow i_{\text{min}} + 1\)
5. if \(r_P(C, i_{\text{min}} - 1) > r_P(C, i_{\text{min}} + 1)\) then \(j \leftarrow i_{\text{min}} - 1\)
6. \(\delta = \delta - \text{ReduceEar}\(P, i_{\text{min}}, j, \delta)\)
7. update \(r_P(C, i_{\text{min}} - 1), r_P(C, i_{\text{min}} + 1),\) and \(i_{\text{min}}\)
8. end
cell. Hence $C$ implicitly determines the priorities of the vertices although it does not appear explicitly in Algorithm 2.

In Algorithm 2, the priorities of the vertices are initialized in line 1 and the vertex with the smallest priority is located in line 2. So long as $\delta$ is negative and the current vertex is movable, the ear at vertex $i_{\text{min}}$ is reduced in line 6. Thanks to the linear property of the triangulation procedure resulting from Theorem 2.3, removing the ear at $p_i$ only affects the priorities of $p_{i-1}$ and $p_{i+1}$, not those of other vertices. Hence we only need to update $i_{\text{min}}$ from the new priorities of these two vertices in line 7. The loop will terminate either when $\delta$ reaches zero or when there are no movable vertices. It is emphasized that Algorithm 2 is more general than its counterpart [35, Algorithm 3] of the PAM method, e.g., $C$ does not have to be convex.

If the cell material region $M^n_C$ contains a hole, $M^n_C$ is partitioned in a manner so that the resulting adjacent simple polygons share vertices with the largest $rP$’s. These common vertices are then marked as not movable to prevent artificial gaps or overlaps between adjacent simple polygons. For such an SSP, Algorithm 2 is first applied to polygons with the smallest volume, then the second smallest, and so on. Note that Algorithm 2 will never incur a violation of the iPAM representation invariants (4.3) because one of its postconditions guarantees that the adjusted polygon is always a subset of the original one.

The iPAM representation invariants (4.3) and Algorithm 2 are sufficiently generic so that if an increase of the volume of $M^n_C$ within $C$ is desired, it can be done by applying Algorithm 2 to $C \setminus ^{+} M^n_C$ and then taking its complement with respect to $C$ as the result.

In essence, VAPER is very similar to polygon triangulation algorithms via ear clipping. It can be shown by the theory of matroids [7] [14, p. 13] that VAPER is an optimal greedy algorithm in terms of seeking the maximal reduction of the number of movable vertices of an input polygon. Similar to the VAPAM algorithm of the PAM method, VAPER also diminishes the total number of interface markers.

### 4.3. Algorithms.

As discussed at the beginning of section 4, a major challenge of improving the accuracy of the PAM method is how to enforce the relation $h_L = r_h h^\alpha$ in an efficient way. With Definition 4.1, the algorithm in Definition 4.2 applies to both structured and unstructured grids.

**Definition 4.1 (interface candidate cell).** A cell $C$ is an interface candidate cell at time $t$ if $C \cap N_M(t) \neq \emptyset$, where

$$N_M(t) := \partial M(t) + B_{hCr}(0)$$

denotes the Minkowski addition of the vector sets of the interface and an open ball centered at the origin and having its radius as $hCr$. Here $Cr$ is the Courant number.

Utilizing the Minkowski addition, we also define a local neighborhood of cell $C$ as

$$N_C := \partial C + B_{hCr}(0).$$

**Definition 4.2 (the iPAM method).** Let $[r_{\text{tiny}} h_L, h_L]$ be given as the range of the distance between adjacent interface markers, where $h_L = r_h h^\alpha$. Within each time step $[t_n, t_n + k]$, the iPAM method advances $M^n$ to $M^{n+1}$ by applying six substeps to each interface candidate cell $C$:

1. **(iPAM-1)** calculate the cell preimage $\hat{T}^{-k}_{t_n+k}(C)$;
FOURTH-ORDER INTERFACE TRACKING BY iPAM

Fig. 6. Enforcing the range of distance between two adjacent interface markers in the iPAM method. Shaded areas represent \( \mathcal{M}_C \), solid dots interface markers, and thick lines an interface edge \( \overline{p_jp_{j+1}} \) and its preimage. Subplots (a) and (b) illustrate (iPAM-4): if the length of \( \overline{p_jp_{j+1}} \) is greater than \( h_L \), its preimage \( \hat{\varphi}_{t_n+k}^{-k}(\overline{p_jp_{j+1}}) \) is divided into equilength subedges by inserting new markers and add the image of the new markers in between \( p_j \) and \( p_{j+1} \) to update \( \partial \mathcal{M}_c^{n+1} \). Repeat this step until no interface edge of \( \partial \mathcal{M}_c^{n+1} \) is longer than \( h_L \); (iPAM-5) if an interface edge \( \overline{p_jp_{j+1}} \subset \partial \mathcal{M}_c^{n+1} \) has its length smaller than \( r_{\text{tiny}} h_L \), replace this edge with its midpoint \( \frac{p_j + p_{j+1}}{2} \); (iPAM-6) (optional) if \( u \) is divergence free, attempt\(^{11}\) to enforce \( \| \mathcal{M}_c^{n+1} \| = \| \overleftarrow{\mathcal{M}_c^{n+1}} \| \) by the VAPER algorithm discussed in section 4.2.

Substeps (iPAM-1), (iPAM-2), (iPAM-3), and (iPAM-6) are conceptually the same as the PAM method; see Figure 4 for an illustration. As shown in Figures 6(a) and (b), (iPAM-4) improves the accuracy by subdividing long interface edges with new markers. In (iPAM-2), the intersection points of the cell preimage with

\(^{11}\)This implies that, for some cells, the desired volume change cannot be fulfilled. By (2.11) and Definition 2.5, the mass conservation error is part of the interface tracking error, hence the former is asymptotically zero so long as the interface tracking method is consistent. Considering the fourth-order accuracy of interface tracking, the failure of exact local mass conservation for some interface candidate cells is not a major issue.
ambient material regions are inserted into the preimage of the cell material region as new markers; see Figure 4(b). As such, \( n_{\text{mks}}(t) \), the total number of interface markers, might grow indefinitely over time. To prevent this, (iPAM-5) imposes an upper bound on \( n_{\text{mks}} \) by replacing edges with negligible lengths with their midpoints so that at the end of each time step,

\[
\frac{L_{\partial M}}{h_L} \leq n_{\text{mks}} \leq \frac{L_{\partial M}}{r_{\text{tiny}}h_L},
\]

where \( L_{\partial M} \) is defined in (4.2) and the lower bound follows from (iPAM-4); see Figure 6 for an illustration.

Another way to prevent \( n_{\text{mks}} \) from continuously growing is to adjust volumes of material polygons by the VAPER algorithms. As discussed in section 3.2, the benefits of volume adjusting are twofold: on the one hand, local mass conservation is achieved; on the other hand, the total number of markers is always diminished. The iPAM algorithm handles volume adjusting better than the PAM method by removing the well-shapeness assumption, allowing both convex and concave polygons, and maintaining the \( C^0 \) continuity of the interface. It is these improvements that admit volume adjustment as part of a high-order algorithm. Occasionally, the volume change can only be partially fulfilled with no reductions on vertex numbers of the material polygons. Fortunately, (iPAM-5) resolves this issue by imposing an upper bound on \( n_{\text{mks}} \) in (4.8). Furthermore, this upper bound holds even if the VAPER algorithms are not applied, for this reason the substep (iPAM-6) is marked as optional. Altogether, the substeps (iPAM-4), (iPAM-5), and (iPAM-6) yield an excellent efficiency of marker usage since they tend to drive the lengths of interface edges to an even distribution. Indeed, \( n_{\text{mks}} \) is much less than its upper bound for all numerical tests performed in section 5.

Apart from numerical efficiency, the even distribution of interface markers also precludes potential stability and conditioning problems of inexact floating-point calculations, which is a large benefit as these problems can be catastrophic for computational geometry algorithms [13]. Another natural consequence of diminishing the total number of markers is as follows.

**Proposition 4.3.** Both substep (iPAM-5) in Definition 4.2 and the VAPER algorithm in section 4.2 diminish \( L_{\partial M} \), i.e., the total length of the interface.

**Proof.** This follows directly from Algorithm 2, Definition 4.2, and the triangle inequality. □

From a physical viewpoint, Proposition 4.3 states that (iPAM-5) and VAPER are numerically mimetic to surface tension, the contracting tendency of a liquid surface. Apart from this, the iPAM method is deliberately made neutral from any specific interfacial physics. The reasons for this design principle are discussed in section 4.7.

### 4.4. Convergence

The accuracy of the iPAM method can be analyzed by a generic framework via mapping and adjusting regular semi-algebraic sets. We state the following conclusion and refer the reader to [33] for a proof.

**Proposition 4.4** (convergence of the iPAM method). The iPAM method is \((2\alpha)\)th-order accurate in 1-norm if

(a) its discrete flow map \( \hat{\varphi} \) is at least \((2\alpha)\)th-order accurate in temporal integration,

(b) the Courant number satisfies \( C_r = O(1) \),

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

(d) the accumulated volume adjustment error caused by (iPAM-5) is \( O(h^{2\alpha}) \).
It follows that setting $\alpha = \frac{1}{2}$ yields a third-order method and $\alpha = 2$ a fourth-order method in terms of the Eulerian grid size $h$. This is confirmed by numerical results in section 5.

As discussed at the end of section 3, a user of the PAM method has the option of setting $n_{\text{max}}$ to a larger value, which reduces $h_L$ in magnitude by adding more interface markers\textsuperscript{12} in (PAM-2). In comparison, the iPAM method reduces the representation error $E_{\text{REP}} = O(h_L^2)$ not only in magnitude by setting $r_h$ to a small value but also in order of magnitude by enforcing $h_L = r_h h^\alpha$ with $\alpha > 1$; the latter is the crucial improvement that leads to a higher-order accuracy.

\textbf{4.5. Complexity.} The complexity of the iPAM method depends heavily on that of the Boolean operations of two polygons, on which we quote some well-known results in computational geometry.

\textbf{Proposition 4.5.} Let $P_1$ be a polygon with $n_{P_1}$ vertices and $P_2$ a polygon with $n_{P_2}$ vertices, and let $n := n_{P_1} + n_{P_2}$. Then $P_1 \cap P_2$, $P_1 \cup P_2$, and $P_1 \setminus P_2$ can each be computed in $O(n \log n + m \log n)$ time, where $m$ is the complexity of the output.\textsuperscript{13}

Furthermore, if both $P_1$ and $P_2$ are simply connected, then $P_1 \cap P_2$ can be computed in $O(n)$ time.

\textbf{Proof.} The first paragraph repeats Corollary 2.7 of [6]. As a constructive proof of the second paragraph, an algorithm with linear complexity can be found in [8].

Clearly, the linear complexity for simply connected polygons is optimal.

\textbf{Proposition 4.6.} Algorithm 2 in section 4.2 runs in $O(h^{1-\alpha}\log h^{1-\alpha})$ time for any fixed $h$.

\textbf{Proof.} The number of vertices of a material polygon $P$ is $n_P = O(\frac{h}{h_L}) = O(h^{1-\alpha})$. The first line of Algorithm 2 initializes the priority of each vertex according to whether or not it is an ear tip. To this end, we construct two reduced polygons $P_{\text{odd}}(p_1, p_3, \ldots)$ and $P_{\text{even}}(p_2, p_4, \ldots)$, and intersect each reduced polygon with the original polygon $P(p_1, p_2, \ldots)$. For a fixed grid size, $P_{\text{odd}}(p_1, p_3, \ldots)$ and $P_{\text{even}}(p_2, p_4, \ldots)$ might not be simply connected. By Proposition 4.5, computing the intersections cost $O(n_P \log n_P)$ time since there are at most $O(n_P)$ intersection points.

The second line clearly takes $O(n_P)$ time. The rest of the proof follows from the fact that the loop of lines 3–8 runs also in $O(n_P)$ time since each line takes $O(1)$ time, which is particularly true for updating the least significant vertex after removing each ear tip. \hfill\Box

\textbf{Corollary 4.7} (the complexity of the iPAM method). In the worst case, the iPAM method runs in $O(h^{\alpha-\epsilon}\log h^{1-\epsilon})$ time for each time step.

\textbf{Proof.} The total number of vertices of the polygons in the local neighborhood of an interface candidate cell is $n_P = O(\frac{h}{h_L}) = O(h^{1-\alpha})$. By Definition 4.2, each of the substeps (iPAM-1) and (iPAM-3) costs $O(1)$ time while each of the substeps (iPAM-4) and (iPAM-5) costs $O(n_P)$ time. In the substep (iPAM-2), each polygon in $M^o \cap N_C$ is simply connected, so is the cell preimage; by Proposition 4.6, computing their intersection takes only $O(n_P)$ time. In contrast, the substep (iPAM-6) costs $O(h^{1-\alpha}\log h^{1-\alpha})$ time in the worst case. The proof is completed by noting that there are $O(\frac{1}{h})$ interface candidate cells since the total length of the interface is $O(1)$ in two dimensions. \hfill\Box

\textsuperscript{12}In terms of accuracy and running time, the PAM method is comparable to the special case of the iPAM method with $h_L = \frac{1}{n_{\text{max}}} h$.

\textsuperscript{13}The running time of Boolean operations on polygons is sensitive to the size of the output. As an example, $m$ could be the number of vertices of the output polygon.
In the above proof, the most expensive substep of iPAM is the VAPER algorithms invoked in (iPAM-6). In the proof of Proposition 4.5, the reduced polygons $P_{\text{odd}}(p_1, p_3, \ldots)$ and $P_{\text{even}}(p_2, p_4, \ldots)$ are in general not simply connected, and thus an optimal algorithm [8] with linear complexity cannot be used. However, in the asymptotic case of $h \to 0$, material areas inside a single cell are well resolved and the two reduced polygons may be assumed as simply connected. Consequently, the logarithm factor in Corollary 4.7 can be dropped. The above arguments lead to a better estimate of the complexity of the iPAM method.

**Proposition 4.8.** In the asymptotic case of $h \to 0$, the iPAM method runs in $O(h^{-\alpha})$ time for each time step.

In reality, an interface tracking method is almost always coupled to a flow solver. It is thus reasonable to demand the expense of the iPAM method be less than that of the flow solver. This requirement is indeed satisfied for $\alpha \leq 2$, since the complexity of an optimal fourth-order flow solver in two dimensions is $O(\frac{1}{h^2})$. For $\alpha = 2$, the scaling factor of $\frac{1}{h^2}$ for even an optimal flow solver is usually much larger than that of the iPAM method. Hence sometimes a choice of $\alpha > 2$ may also be acceptable.

### 4.6. Efficiency

VOF methods and the PAM methods are at best second-order accurate [29]; their complexity in two dimensions is $O(\frac{1}{h})$ for each time step because of the $O(\frac{1}{h^2})$ interface candidate cells. To achieve the same second-order accuracy with the iPAM method, Proposition 4.4 implies $\alpha = 1$. By Proposition 4.8, the complexity of the iPAM method is also $O(\frac{1}{h^2})$. Hence the running time of VOF methods, the PAM method, and the iPAM method with $\alpha = 1$ should differ only by a constant factor for any given test and fixed grid size.

The iPAM method achieves fourth-order accuracy by setting $\alpha = 2$. For VOF methods to achieve the same order of accuracy, one could use a much smaller uniform grid size $h_{\text{VOF}} = h^2$, but this might be too expensive because the total number of cells would be $O(\frac{1}{h^2})$. Alternatively, one could use the adaptive mesh refinement moment-of-fluid (AMR-MOF) method [1] to adaptively refine the grids near the interface. If the size of the coarsest grid is $h$, then that of the finest grid has to be $h^2$. As $h$ is reduced, the number of levels of grids also increases, incurring a rapid growth of the expense on bookkeeping the AMR hierarchy. Therefore, the iPAM method is more efficient than adaptive VOF methods in achieving fourth-order accuracy, although in this case they both have $O(\frac{1}{h^2})$ as their formal complexity. In practice, the efficiency advantage of the iPAM method over another method can be measured by the ratio of CPU time in achieving the same accuracy; see section 5.3 for more discussions based on timing results.

The iPAM method decouples the accuracy of interface tracking from that of the flow solver by separating the two length scales $h$ and $h_L$. As discussed in section 4.4, the free parameters $r_h$ and $\alpha$ in the relation $h_L = r_h h^{\alpha}$ furnish a simple, efficient, and flexible way to improve the accuracy of interface tracking without changing the Eulerian grid size $h$. Within a fixed order of accuracy, the flexibility afforded by $r_h$ could be very useful when a high-order velocity field is not available or a certain choice of $h_L$ is too large to be in the asymptotic range; see Table 3 in section 5.2 for an example.

With the development of high-order methods for the incompressible Navier–Stokes equations such as [32, 31], high-order interface tracking methods will eventually be incorporated into these flow solvers as better instruments for the study of multiphase flows. To this end, the better efficiency and flexibility of the iPAM method in coupling with higher-order flow solvers is a significant advantage over previous second-order interface tracking methods.
4.7. **Topological changes.** In reality, different types of interface have different behaviors. For an air-water free surface, a film of chemical or organic surfactants may greatly increase the damping of surface waves on water [5, 25] even if its thickness is less than ten nanometers! As another example, the initially flat surface of a magnetic fluid will move to form a stationary pattern of liquid crests when a certain threshold of the magnetic induction is reached [4]. Even for the same type of interface, coalescence and separation of droplets may exhibit distinct behaviors for different regimes of the Weber number and other impact parameters [22].

VOF methods and level-set methods do not have special algorithms for merging and separation of the tracked fluids; this is often considered as an advantage of these methods. However, in the authors’ opinion, this so-called “automatic” merging is more of a disadvantage as far as the interfacial physics is concerned: the various behaviors of the interface in merging and separation are bluntly handled with a single choice that solely depends on the particularities of VOF and level-set methods. As such, this automatic merging deprives a scientist of the opportunity of modeling different physics with different algorithms.

In comparison, the iPAM method provides an application scientist a solid foundation upon which a particular algorithm could be designed to cater to the specific interfacial physics being studied. For example, in dealing with air-water free-surface flows, the separation of water droplets from the bulk of the water phase could be determined by block decomposition of a graph that models the topology of the water phase, and the merging of water droplets back to the bulk flow could be handled by volume-adjusting algorithms based on mass conservation [35, Figure 5] [36]. It is also shown in [36, Figures 17 and 18] that these heuristic algorithms for handling topological changes yield interface loci that agree with experimental results much better than those of the “automatic” approach of VOF methods.

To handle interface merging, a generic procedure should have two parts: detecting a merge and evolving the merge. For the iPAM method, the possibility of multiple simple polygons in a single SSP admits a precise detection of merging after the substep (iPAM-3) and before the substep (iPAM-4): if multiple polygons within the same cell have nonempty intersections, the forward tracing in (iPAM-4) can be rolled back to obtain the time instance of the initial contact of these material regions. Then, for the evolution of the merged interface, different schemes can be designed according to the specific interfacial physics of the fluids under study. It is mainly because of the different physics that the iPAM method does not contain a precise algorithm to handle merging and separation: such algorithms should vary from one regime to another.

As discussed in section 2.2, the flow maps of the velocity field also apply to point sets. So long as the velocity field is continuous in time and Lipschitz continuous in space, the corresponding flow maps are homeomorphisms that preserve the topology of each component of the material regions. Therefore, material regions should neither merge nor break up if the discrete velocity is sufficiently continuous. For many smooth velocity fields, VOF methods and level-set methods break up underresolved material regions or merge separate material components; this can be interpreted as failures of these interface tracking methods in preserving the topological invariants. In practice, merging and separation of material interfaces are usually results of discontinuities in the velocity field at locations of topological change. For better simulations of the interfacial physics, the authors advocate that the problem of how to model topological changes of the tracked material be separated from that of how to track its interface. After all, these problems differ dramatically in terms of their nature and length scales.
5. Tests. In this section, the set of benchmark tests in [35] are performed on a rectangular Eulerian grid with a uniform grid size \( h \) to compare the iPAM method to the PAM method and other highly accurate methods such as the stream scheme by Harvie and Fletcher [11], the EMFPA-SIR method by Lopez et al. [17], the hybrid markers-conservation method by Aulisa, Manservisi, and Scardovelli [2], and the AMR-MOF method by Ahn and Shashkov [1]. For all tests, the ODE (2.2) is integrated in time by Heun’s third-order formula [16, p. 155] and the classic fourth-order Runge–Kutta method for the choices of \( h_L = O(h^{3/2}) \) and \( h_L = O(h^2) \), respectively.

Almost all aforementioned VOF methods measure the error of interface tracking by the so-called “geometric error”

\[
E_g(T) := \sum_{C \subset \Omega} \| C \| \left| \langle f(T) \rangle_C - \langle f \rangle^{T/k}_C \right|,
\]

where \( \langle f(T) \rangle_C \) and \( \langle f \rangle^{T/k}_C \) are the exact and computed volume fraction of cell \( C \) at the ending time \( T \). Hence we present results in terms of \( E_g \) to facilitate the accuracy comparison. A max-norm is also defined to measure the local distance from the computed interface to the exact interface as

\[
E_\infty(T) := h \max_{C \subset \Omega} \left| \langle f(T) \rangle_C - \langle f \rangle^{T/k}_C \right|.
\]

5.1. Translation of rigid bodies. The flow map of a constant velocity field can be approximated to machine precision, thus translation tests measure the most basic feature of an interface tracking method: how well the shape of a material region is represented and maintained. In this subsection, a hollow square, a tilted hollow square, and a hollow circle are shifted by two constant velocities. The initial setup and other test parameters are shown in Figure 7.

VOF methods round all the corners of the squares, as shown in [35, Figure 11]. In contrast, the PAM method maintains the sharpness of some corners in Figure 8(a). However, the upper and lower concave corners of the inner square are still slightly rounded, due to approximating multiple disjoint polygons with their convex hull. Figures 10(e) and (f) show two examples of multiple disjoint polygons within one cell using the iPAM method. As discussed in section 4.1, the iPAM method is more general than the PAM method in that multiple polygons within a single cell are represented faithfully. Consequently, all the corners are preserved very well in Figure 8(b), with the corresponding errors close to machine precision in Table 1.

For all three shapes, the geometric errors of the iPAM method are listed in Table 1 for the finest grid, along with those of the PAM method and the stream scheme [11]. Although the stream scheme is the best VOF method we can find for the translation test, it is much less accurate than both the PAM and iPAM methods because of the rounding of all corners. For all three cases, the iPAM method yields results close to machine precision when the representation errors are excluded.

---

14 The measure \( E_1(t_n) = \| M(t_n) \oplus M^0 \| \) from (2.11) is more rigorous than (5.1) for three reasons. First, \( E_1 \) is a metric on the space of bounded regular closed sets while \( E_g \) is not: \( E_g = 0 \) does not necessarily imply exact interface tracking. Second, \( E_1 \) does not depend on the partition of \( \Omega \) while \( E_g \) does. Last, \( E_g \) tends to hide reconstruction errors of VOF methods [29, section 3.6].

15 We also calculated \( E_1 \) for all tests and we found that \( E_1 \) and \( E_g \) of the iPAM method differ by at most several percent on the finest grids. The convergence rates obtained from \( E_1 \) and \( E_g \) are the same for the first three most significant digits.

16 The representation error is included if \( M(T) \) is used as the exact final solution. In comparison, the representation error is excluded if a translation of \( M^0 \) is used as the exact final solution.
FOURTH-ORDER INTERFACE TRACKING BY iPAM

A2389

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>computational domain</td>
<td>$\Omega = [0, 4] \times [0, 4]$</td>
</tr>
<tr>
<td>simulation time</td>
<td>$t \in [0, 1.25]$</td>
</tr>
<tr>
<td>shape parameters</td>
<td>$D = 0.8$, $d = 0.4$, $\theta = 26.57^\circ$</td>
</tr>
<tr>
<td>velocities</td>
<td>$\mathbf{u} = (1, 0), (2, 1)$</td>
</tr>
<tr>
<td>Courant number</td>
<td>$Cr = 0.25$</td>
</tr>
<tr>
<td>Eulerian grid sizes</td>
<td>$h = 0.08, 0.04, 0.02$</td>
</tr>
<tr>
<td>Lagrangian length scales</td>
<td>$h_L = h^{3/2}, 5h^2$</td>
</tr>
</tbody>
</table>

Fig. 7. Initial setup and other parameters of the translation tests.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Square ($0^\circ$)</th>
<th>Square ($26.57^\circ$)</th>
<th>Circle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stream/Puckett [11]</td>
<td>1.61e-03</td>
<td>4.57e-03</td>
<td>1.42e-03</td>
</tr>
<tr>
<td>iPAM ($h_L = 5h^2$)</td>
<td>4.86e-15</td>
<td>2.31e-14</td>
<td>5.25e-14</td>
</tr>
</tbody>
</table>

$E_y$ for $\mathbf{u} = (1, 0)$

<table>
<thead>
<tr>
<th>Methods</th>
<th>$E_y$ for $\mathbf{u} = (2, 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAM</td>
<td>7.77e-15</td>
</tr>
<tr>
<td>iPAM ($h_L = 5h^2$)</td>
<td>7.23e-14</td>
</tr>
</tbody>
</table>

Fig. 8. Final results of translating the tilted hollow square. The number of markers of the iPAM method for $h = 0.08$ remains constant (290 for the outer square and 137 for the inner square) during the whole simulation period. We show only part of the Eulerian grid inside the material region $\mathcal{M}(t)$.

Table 1

Geometric errors for the translation tests shown in Figure 7. $h = 0.02$. The representation errors are excluded from the results of the PAM and iPAM methods.
Table 2

Errors and convergence rates of the iPAM method for translating the hollow circle in Figure 7. The representation errors are included.

<table>
<thead>
<tr>
<th>Tests of iPAM</th>
<th>$h = 0.08$</th>
<th>$h = 0.04$</th>
<th>$h = 0.02$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u = (1, 0)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_L = h^3, E_g$</td>
<td>1.53e-04</td>
<td>3.00</td>
<td>1.92e-05</td>
</tr>
<tr>
<td>$h_L = h^3, E_\infty$</td>
<td>7.42e-05</td>
<td>3.01</td>
<td>9.23e-06</td>
</tr>
<tr>
<td>$h_L = 5h^2, E_g$</td>
<td>2.86e-04</td>
<td>3.90</td>
<td>1.92e-05</td>
</tr>
<tr>
<td>$h_L = 5h^2, E_\infty$</td>
<td>1.30e-04</td>
<td>3.81</td>
<td>9.23e-06</td>
</tr>
<tr>
<td>$u = (2, 1)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$h_L = h_3^3, E_g$</td>
<td>1.78e-04</td>
<td>3.09</td>
<td>2.09e-05</td>
</tr>
<tr>
<td>$h_L = h_3^3, E_\infty$</td>
<td>8.85e-05</td>
<td>3.10</td>
<td>1.03e-05</td>
</tr>
<tr>
<td>$h_L = 5h^2, E_g$</td>
<td>3.38e-04</td>
<td>4.02</td>
<td>2.09e-05</td>
</tr>
<tr>
<td>$h_L = 5h^2, E_\infty$</td>
<td>2.38e-04</td>
<td>4.53</td>
<td>1.03e-05</td>
</tr>
</tbody>
</table>

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 | $Cr = 0.25$
Eulerian grid sizes | $h = 0.08, 0.04, 0.02$
Lagrangian length scales | $h_L = 5h^2, 0.5h^2$

![Diagram of Zalesak disk rotation test](image)

**Fig. 9. Initial setup and other parameters of the Zalesak disk rotation test.**

The representation errors are zero in all these tests except the hollow circle. After including them, we list the results of the iPAM method for translating the hollow circle in Table 2. As expected, the convergence rates based on the representation errors are three and four for $h_L = h^3$ and $h_L = 5h^2$, respectively. Furthermore, the total number of markers for each test is a constant during the period of simulation. This is not surprising since the maximum distance between adjacent markers is not changed by translation.

5.2. Rotation of the Zalesak disk. This test places a slotted disk in a purely rotational velocity field given by the stream function

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

where $O = (x_O, y_O)$ is the center of the rotation and $\omega$ the constant angular velocity. The initial setup and other parameters of this test are shown in Figure 9. After a full revolution of $2\pi$ rotation, the slotted circle returns to its initial location, hence the exact solution is the same as the initial condition.

In Figure 10, the SSP polygons are plotted on actual grids for a number of time instants including $t = \frac{1}{2}T$ and $t = T$; the possibility of multiple simple polygons...
Fig. 10. Results of Zalesak disk rotation tests. Subplots (a) and (b) are the results of the PAM method ($h = 0.02$) and the other subplots those of the iPAM method ($h = 0.08$). Subplot (e) shows the two simple polygons within the cell at the concave end of the slot for an intermediate time and subplot (f) the three simple polygons for another time. The maximum number of markers of iPAM for $h = 0.08$ is 677 and scales as $O(h^{-1})$ when $h$ is refined.
Errors and convergence rates of the iPAM method for rotating the Zalesak disk in Figure 9. As far as we know, the AMR-MOF method [1] yields the best accuracy of $E_{\infty} = 1.74 \times 10^{-4}$ for $h = 0.02$ among all other interface tracking methods. As for the PAM method with $h = 0.02$, $E_{\infty} = 3.97 \times 10^{-4}$.

<table>
<thead>
<tr>
<th>Tests</th>
<th>$h = 0.08$</th>
<th>$h = 0.04$</th>
<th>$h = 0.02$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_L = 5h^2$, $E_g$</td>
<td>6.55e-03</td>
<td>4.45</td>
<td>3.00e-04</td>
</tr>
<tr>
<td>$h_L = 5h^2$, $E_{\infty}$</td>
<td>8.83e-03</td>
<td>4.65</td>
<td>3.53e-04</td>
</tr>
<tr>
<td>$h_L = 0.5h^2$, $E_g$</td>
<td>7.76e-06</td>
<td>3.98</td>
<td>4.91e-07</td>
</tr>
<tr>
<td>$h_L = 0.5h^2$, $E_{\infty}$</td>
<td>3.17e-06</td>
<td>3.99</td>
<td>2.00e-07</td>
</tr>
</tbody>
</table>

within one cell is demonstrated in subplots (e) and (f). The much better preservation of these sharp corners is a major improvement of the iPAM method over the PAM method.

In Table 3, the convergence rates of the iPAM method for $h_L = 5h^2$ are well over four, suggesting that $h_L$ is not in the asymptotic range yet. In comparison, a smaller Lagrangian length scale $h_L = 0.5h^2$ yields excellent fourth-order convergence rates both in the $1$-norm and in the max-norm. In both cases, the iPAM method is much more accurate than the AMR-MOF method, although its results are the best we can find in the literature.

One condition rigorously enforced by the iPAM method is that the lengths of all interface edges are within the range $[r_{\text{tiny}}h_L, h_L]$. In this work $r_{\text{tiny}} = 0.1$, hence the averaged distance of adjacent markers for $h_L = 5h^2$ is about 10 to 100 times larger than that for $h_L = 0.5h^2$; it then follows that the errors of $h_L = 5h^2$ should be 100 to 10000 times larger than those of $h_L = 0.5h^2$, which is verified in Table 3 as the ratio of these two errors is around 1000 for all cases. These results support the discussion in section 4.6 that the parameter $r_h$ in $h_L = r_hh^2$ provides a simple and flexible mechanism for tweaking the accuracy of interface tracking. Conceptually, reducing the value of $r_h$ is equivalent to refining the interface. This one-dimensional mesh refinement of the Lagrangian grid is more efficient than that of the two-dimensional Eulerian grid, even if the latter approach adaptively restricts the refinement in a local neighborhood of the interface.

5.3. Vortex shear of a circular disk. In this test, a nonuniform, transient velocity field given by the stream function

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

is imposed on a circular disk. The initial setup and other test parameters are shown in Figure 11. At time $t = \frac{T}{2}$, the velocity field is reversed by the cosinusoidal temporal factor so that the exact solution at $t = T$ is the same as the initial condition.

The simple polygons representing the cell material regions of the iPAM method for $T = 8$, $h = \frac{T}{8}$, and $h_L = h^2$ are plotted on the actual grids for a number of time instants in Figure 12. During the period of simulation, the velocity field stretches the material region into a filament that spirals toward the vortex center, potentially tearing it apart. As shown in Figure 12, the iPAM method preserves very well the thin tail as one piece for all time instants. In contrast, VOF methods tend to break up the tail to “flotsam” if the grids resolution is not fine enough; see [35, Figure 16] for an example.
<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>computational domain</td>
<td>$[0, 1] \times [0, 1]$</td>
</tr>
<tr>
<td>simulation time</td>
<td>$t \in [0, T]$</td>
</tr>
<tr>
<td>shape parameters</td>
<td>$C = (0.5, 0.75), R = 0.15$</td>
</tr>
<tr>
<td>velocity periods</td>
<td>$T = 0.5, 2, 8$</td>
</tr>
<tr>
<td>Courant number</td>
<td>$Cr = 1$</td>
</tr>
<tr>
<td>Eulerian grid sizes</td>
<td>$h = \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$</td>
</tr>
<tr>
<td>Lagrangian length scales</td>
<td>$h_L = 0.1 h, h^{\frac{3}{2}}, h^2$</td>
</tr>
</tbody>
</table>

**Fig. 11.** Initial setup and other parameters of the reversed single vortex test.

At $t = 0$, the circle is initialized by dividing its circumference into edges with the same length $\frac{2\pi}{h_L}$. The total number of markers increases dramatically from 350 to 5455 as the number of interface cells grows in the first half of the simulation, and decreases steadily as the thin spiral merges back to the disk in the latter half of the simulation. Given perfect initial distribution of the interface markers, the function

$$r_{\text{mks}}(t) := \frac{\mid\partial \mathcal{M}(t)\mid n_{\text{mks}}(t_0)}{\mid\partial \mathcal{M}(t_0)\mid n_{\text{mks}}(t)}$$

for $t = \frac{T}{2}$, $T$ is 350, 5455, 670, respectively.
Table 4
Errors and convergence rates based on $E_g$ for the vortex shear tests in Figure 11. Those based on $E_\infty$ are qualitatively the same.

<table>
<thead>
<tr>
<th>Methods</th>
<th>$h = \frac{1}{32}$</th>
<th>$h = \frac{1}{64}$</th>
<th>$h = \frac{1}{128}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_g$ for $T = 0.5$</td>
<td>$E_g$ for $T = 2$</td>
<td>$E_g$ for $T = 8$</td>
</tr>
<tr>
<td>iPAM, $h_L = 0.1h$</td>
<td>4.07e-05</td>
<td>1.70e-05</td>
<td>1.22e-07</td>
</tr>
<tr>
<td>iPAM, $h_L = h^{\frac{5}{2}}$</td>
<td>4.96e-06</td>
<td>2.17e-06</td>
<td>8.45e-09</td>
</tr>
<tr>
<td>iPAM, $h_L = h^2$</td>
<td>2.98e-07</td>
<td>2.95</td>
<td>3.88</td>
</tr>
<tr>
<td>PAM [35] with $n_{\text{max}} = 10$</td>
<td>5.86e-07</td>
<td>5.39e-07</td>
<td>5.73e-10</td>
</tr>
<tr>
<td>EMPPA-SIR [17]</td>
<td>1.41e-05</td>
<td>2.17e-06</td>
<td>2.80e-07</td>
</tr>
<tr>
<td></td>
<td>2.44e-06</td>
<td>3.88</td>
<td>3.88</td>
</tr>
<tr>
<td></td>
<td>2.21</td>
<td>2.80e-07</td>
<td>5.73e-10</td>
</tr>
<tr>
<td></td>
<td>5.15e-07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPAM, $h_L = 0.1h$</td>
<td>8.34e-05</td>
<td>3.42e-05</td>
<td>5.04e-07</td>
</tr>
<tr>
<td>iPAM, $h_L = h^{\frac{5}{2}}$</td>
<td>1.05e-05</td>
<td>4.23e-06</td>
<td>2.32e-08</td>
</tr>
<tr>
<td>iPAM, $h_L = h^2$</td>
<td>2.94</td>
<td>2.97</td>
<td>3.82</td>
</tr>
<tr>
<td>PAM [35] with $n_{\text{max}} = 10$</td>
<td>3.57e-06</td>
<td>5.39e-07</td>
<td>5.73e-10</td>
</tr>
<tr>
<td>AMR-MOF [1]</td>
<td>1.22e-07</td>
<td>4.44</td>
<td>2.32e-08</td>
</tr>
<tr>
<td></td>
<td>2.32e-06</td>
<td>3.82</td>
<td>1.64e-09</td>
</tr>
<tr>
<td></td>
<td>2.80e-07</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.11</td>
<td>3.88</td>
<td>1.64e-09</td>
</tr>
<tr>
<td></td>
<td>1.24e-05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iPAM, $h_L = 0.1h$</td>
<td>6.21e-04</td>
<td>2.07e-04</td>
<td>4.32e-06</td>
</tr>
<tr>
<td>iPAM, $h_L = h^{\frac{5}{2}}$</td>
<td>7.85e-05</td>
<td>2.69e-05</td>
<td>1.74e-07</td>
</tr>
<tr>
<td>iPAM, $h_L = h^2$</td>
<td>2.99</td>
<td>2.92</td>
<td>3.89</td>
</tr>
<tr>
<td>PAM [35] with $n_{\text{max}} = 10$</td>
<td>9.89e-06</td>
<td>3.55e-06</td>
<td>1.17e-08</td>
</tr>
<tr>
<td>AMR-MOF [1]</td>
<td>9.89e-06</td>
<td>3.55e-06</td>
<td>1.17e-08</td>
</tr>
<tr>
<td></td>
<td>9.89e-06</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.89e-06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hybrid markers-conservation [2]</td>
<td>4.24e-05</td>
<td>7.24e-06</td>
<td>2.42</td>
</tr>
<tr>
<td></td>
<td>2.42</td>
<td>1.35e-06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.24e-05</td>
<td>7.24e-06</td>
<td>2.42</td>
</tr>
<tr>
<td></td>
<td>2.42</td>
<td>1.35e-06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.33e-02</td>
<td>2.89</td>
<td>5.04e-04</td>
</tr>
<tr>
<td></td>
<td>3.09e-04</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.09e-04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

measures how evenly the interface markers are distributed at time $t$ and thus the efficiency of the usage of markers. By (4.8), we have $r_{\text{tiny}} \leq r_{\text{mks}} \leq 1$. For the test shown in Figure 12, $r_{\text{mks}}(T) = 350/670 \approx 0.52$, which is much bigger than the lower bound $r_{\text{tiny}} = 0.1$; this supports the discussion in section 4.3 that the last three substeps of the iPAM method tend to distribute the markers evenly along the interface. As also discussed in section 4.3, the even distribution of the markers helps to maintain stability of the computational-geometry algorithms, and diminishing the number of markers is analogous to adding numerical surface tension to the interface.

The errors and convergence rates of the iPAM method, together with those of some previous methods, are listed in Table 4. For the case of $h_L = 0.1h$, its errors on the finest grid are close to that of the PAM method with $n_{\text{max}} = 10$, especially for the cases of $T = 0.5$ and $T = 2$. Although the iPAM method with $h_L = 0.1h$ is conceptually only second-order accurate, we observe convergence rates very close to three; this is probably due to the fact that the interface is very well resolved by the Lagrangian length scale $0.1h$. This third-order accuracy is not achieved by the PAM method, because its underlying representation and volume adjusting algorithms are not as rigorous as those of the iPAM method.

For the choices of $h_L = h^{\frac{5}{2}}$ and $h_L = h^2$, the third- and fourth-order convergence rates of the iPAM method are also confirmed for all cases in Table 4. In particular, the fourth-order iPAM method with $h_L = h^2$ yields errors smaller by orders of magnitudes than those of the best methods we can find in the literature. As discussed in previous sections, the accuracy of the iPAM method can be further improved by reducing $r_h$. On the other hand, if errors of the iPAM method are much smaller than those of a flow solver, one can increase the value of $r_h$ to reduce the computational expense.
FOURTH-ORDER INTERFACE TRACKING BY iPAM

Table 5
CPU time in seconds and the corresponding amplification rates for running the vortex shear tests in Figure 11 with the iPAM method on a personal computer with a single Intel® i7-3930k CPU. By Proposition 4.8, the amplification rate is expected to be, respectively, 2, 2.5, and 3 for $h_L = 0.1h$, $h^2$, and $h^3$ since there are $O(\frac{1}{h})$ number of time steps.

<table>
<thead>
<tr>
<th>vortex shear test</th>
<th>$h = \frac{1}{10}$</th>
<th>rate</th>
<th>$h = \frac{1}{20}$</th>
<th>rate</th>
<th>$h = \frac{1}{40}$</th>
<th>rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_L = 0.1h, T = 0.5$</td>
<td>9</td>
<td>1.85</td>
<td>32</td>
<td>2.26</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>$h_L = 0.1h, T = 2$</td>
<td>67</td>
<td>2.03</td>
<td>273</td>
<td>2.36</td>
<td>1403</td>
<td></td>
</tr>
<tr>
<td>$h_L = 0.1h, T = 8$</td>
<td>984</td>
<td>2.08</td>
<td>417</td>
<td>2.00</td>
<td>16709</td>
<td></td>
</tr>
<tr>
<td>$h_L = h^2, T = 0.5$</td>
<td>5</td>
<td>2.53</td>
<td>26</td>
<td>2.56</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>$h_L = h^2, T = 2$</td>
<td>52</td>
<td>2.21</td>
<td>239</td>
<td>2.47</td>
<td>1322</td>
<td></td>
</tr>
<tr>
<td>$h_L = h^2, T = 8$</td>
<td>692</td>
<td>2.34</td>
<td>354</td>
<td>2.39</td>
<td>18431</td>
<td></td>
</tr>
<tr>
<td>$h_L = h^3, T = 0.5$</td>
<td>16.1</td>
<td>3.02</td>
<td>131</td>
<td>3.18</td>
<td>1185</td>
<td></td>
</tr>
<tr>
<td>$h_L = h^3, T = 2$</td>
<td>124</td>
<td>2.86</td>
<td>91</td>
<td>2.99</td>
<td>7167</td>
<td></td>
</tr>
<tr>
<td>$h_L = h^3, T = 8$</td>
<td>1911</td>
<td>2.97</td>
<td>15003</td>
<td>3.05</td>
<td>124110</td>
<td></td>
</tr>
</tbody>
</table>

of interface tracking. These examples illustrate the discussion in section 4.6 on the flexibility of the iPAM method for it to be coupled with a flow solver.

CPU time in seconds for performing these tests by the iPAM method are listed in Table 5. By Proposition 4.8, the expected amplification rates of CPU time are, respectively, 2, 2.5, and 3 for $h_L = 0.1h$, $h^2$, and $h^3$. Table 5 confirms this expectation and thus verifies that the complexity of the iPAM method is $O(h^{-\infty})$.

The rest of this subsection contains an efficiency comparison of the fourth-order iPAM method to other methods quoted in Table 4. For this purpose, we utilize the systematic arguments presented in [31, section 7], where the efficiency of different methods are measured by their CPU time in achieving the same accuracy. Let $E_{g,4}$ and $E_{g,\mathcal{O}}$ denote the accuracy of the fourth-order iPAM and another $O$th-order method. For the vortex shear tests, the values of both $E_{g,4}$ and $E_{g,\mathcal{O}}$ for $h = \frac{1}{128}$ are readily available in the last column of Table 4. For the $O$th-order method to achieve the accuracy $E_{g,4}$, the number of times that the grid has to be refined is

$$n_{\mathcal{O} \rightarrow 4} = \log_2 \frac{E_{g,\mathcal{O}}}{E_{g,4}},$$

where the base $2^\mathcal{O}$ of the logarithm is the reduction factor of $E_{g,\mathcal{O}}$ upon each halving of the grid size. Consequently, in improving the accuracy of the $O$th-order method from $E_{g,\mathcal{O}}$ to $E_{g,4}$, its CPU time is increased by a factor of

$$R_{\mathcal{O}} = 4^{n_{\mathcal{O} \rightarrow 4}} = \left(\frac{E_{g,\mathcal{O}}}{E_{g,4}}\right)^\frac{2^\mathcal{O}}{\mathcal{O}},$$

where the base 4 is due to the assumption that the complexity of the $O$th-order method is $O(\frac{1}{h})$ for each time step. Let $R_{\mathcal{O}|4}$ denote the ratio of CPU time of the choice $h_L = h^2$ to that of $h_L = 0.1h$ in Table 5. Then the CPU-time speedup of $h_L = h^2$ over $h_L = 0.1h$ can be calculated by

$$S_{4|\mathcal{O}} = \frac{R_{\mathcal{O}}}{R_{\mathcal{O}|4}} = \frac{1}{R_{\mathcal{O}|4}} \left(\frac{E_{g,\mathcal{O}}}{E_{g,4}}\right)^\frac{2^\mathcal{O}}{\mathcal{O}}.$$
Table 6

<table>
<thead>
<tr>
<th>vortex shear test</th>
<th>the Oth-order method</th>
<th>( S_{4/\Omega} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T = 0.5 )</td>
<td>iPAM with ( h_L = 0.1h )</td>
<td>11.5</td>
</tr>
<tr>
<td></td>
<td>PAM with ( n_{\text{max}} = 10 )</td>
<td>55.6</td>
</tr>
<tr>
<td></td>
<td>EMFPA-SIR [17]</td>
<td>1694.9</td>
</tr>
<tr>
<td>( T = 2 )</td>
<td>iPAM with ( h_L = 0.1h )</td>
<td>19.0</td>
</tr>
<tr>
<td></td>
<td>PAM with ( n_{\text{max}} = 10 )</td>
<td>50.3</td>
</tr>
<tr>
<td></td>
<td>AMR-MOF [1]</td>
<td>156.7</td>
</tr>
<tr>
<td>( T = 8 )</td>
<td>iPAM with ( h_L = 0.1h )</td>
<td>12.2</td>
</tr>
<tr>
<td></td>
<td>PAM with ( n_{\text{max}} = 10 )</td>
<td>6.8</td>
</tr>
<tr>
<td></td>
<td>Hybrid markers-conservation [2]</td>
<td>8.4</td>
</tr>
</tbody>
</table>

If we further assume that, for the grid \( h = \frac{1}{128} \), the CPU time of the iPAM method with \( h_L = 0.1h \) is the same as that of the Oth-order method,\(^17\) then \( S_{4/\Omega} \), the CPU-time speedup of the fourth-order iPAM method to other methods quoted in Table 4, can also be calculated by (5.8). Its values are listed in Table 6.

As shown in Table 6, the fourth-order iPAM method with \((h_L = 0.1h^2)\) is 1650 times more efficient than the state-of-the-art VOF method EMFPA-SIR for \( h = \frac{1}{128} \) and \( T = 0.5! \) For \( T = 2 \), it is more than a hundred times more efficient than the adaptive method AMR-MOF. The value of the speedup is much less for the method of hybrid markers-conservation, probably because this method is essentially Lagrangian and its convergence rate is taken as three. As discussed in [31, section 7], although the values of the speedup might differ for other tests, the pattern remains the same that the CPU-time speedup grows as a power function when the grid size or the targeted accuracy is reduced. On the other hand, when combined with a flow solver, the efficiency advantage of the fourth-order iPAM method might not be obvious since the CPU time spent in interface tracking is usually much less than that of a flow solver.

5.4. Deformation of a circular disk. This test has long been regarded as one of the most difficult interface tracking problems. The velocity field is given by the stream function

\[
\psi(x, y) = \frac{1}{n_{\Omega} \pi} \sin(n_{\Omega} \pi (x + 0.5)) \cos(n_{\Omega} \pi (y + 0.5)) \cos \left( \frac{\pi t}{T} \right),
\]

where \( n_{\Omega} \) is the number of vortices in the computational domain and the temporal factor reverses the flow at \( t = \frac{T}{2} \). The initial setup and other test parameters are shown in Figure 13.

The results of the iPAM method on the coarsest grid are shown in Figure 14, where the simple polygons in the SSPs are plotted on the actual grids. Although the velocity field stretches the disk to an extremely thin width at \( x = 0.5 \), the filament should not break up because, as discussed in section 4.7, the flow map of the smooth

\(^{17}\text{It is reported in [35] that the PAM method is slower than a piecewise linear interface calculation VOF method (DDR/ELVIRA) by about twenty percent. In our tests, the running time of the iPAM method with } h_L = 0.1h \text{ is about the same as that of the PAM method with } n_{\text{max}} = 10. \text{ Hence, the CPU time of the iPAM method with } h_L = 0.1h \text{ is a good estimate of the lower bound on the methods in Table 6 since none of them is simpler than DDR/ELVIRA.}
velocity field is a homeomorphism and any topological change such as splitting and merging would be a failure of the interface tracking method in fulfilling this topological invariant. As shown in all subplots of Figure 14, the iPAM method preserves the topology of the disk for all time instants, even on the coarsest grid. More important, this illustrates an advantage of the iPAM method in decoupling the Eulerian grid size from the Lagrangian length scale since an interface tracking method with only one length scale will have to use a very small grid size to resolve the fine features and prevent the filament from breaking up.

The evolution of $n_{	ext{msk}}(t)$ for this test is qualitatively the same as that for the vortex shear test with $T = 8$: $n_{	ext{msk}}(t)$ attains its minimum at $t = t_0 = 0$ and its maximum at $t = T/2$, and $n_{	ext{msk}}(T) < 2n_{	ext{msk}}(t_0)$. The efficiency measure defined in (5.5) is $r_{	ext{msk}}(T) = 349/650 > 0.5$, which implies that, after all of the stretching and merging, the lengths of the interface edges vary by a factor of about two at the final time compared to their initial spacing.

The errors and convergence rates of the iPAM method are listed in Table 7. As expected from the discussion in section 4, the convergence rates are three and four for the two choices of $h_L = h^2$ and $h_L = h^2$, respectively. The rate of convergence in the max-norm is slightly less than four for the case of $h_L = h^2$; this is probably caused by the sharp corners present in the subplots (c)–(g) of Figure 12. Similar to previous tests, the fourth-order iPAM method is much more accurate than the PAM method.

6. Conclusions. We have proposed the iPAM method for fourth-order interface tracking in incompressible flows by separating the Lagrangian length scale of interface markers from the Eulerian grid size. A range on the Lagrangian length scale is actively enforced by dividing long interface edges, removing short interface edges, and adjusting the material regions via polygon ear removal. In summary, the advantages of the iPAM method are

(I) accuracy: the choice of $h_L = O(h^2)$ yields fourth-order convergence even for interfaces with derivative discontinuities;

(II) generality: the iPAM method can be directly applied to either structured grids or unstructured grids without any algorithmic modifications;

(III) adaptivity: the parameter $r_h$ in $h_L = r_h h^a$ provides a simple mechanism of refining the Lagrangian grids for better accuracy;

(IV) versatility: the decoupling of the resolution of interface tracking from that of the flow field is amenable to truly multiscale simulations.

Prospects for future research follow. First, the iPAM method can be further generalized to tracking an arbitrary number of materials with fourth-order accuracy for each material. Second, the augmentation of the iPAM method to compressible
Fig. 14. Results of the iPAM method for the deformation test with \( h = \frac{1}{32} \) and \( h_L = h^\frac{3}{2} \). The number of markers at \( t = 0, \frac{T}{2}, T \) is 349, 3199, 650, respectively.
flows should be straightforward since the divergence-free condition is not assumed on the velocity field. Third, the iPAM method can be extended to three dimensions via the theory of Nef polyhedra [19] and the algorithms on selective Nef complex [10]. Fourth, although conceptually the iPAM method can be applied to unstructured grids, the actual implementation might require much care. Last, we will utilize the ideas in [28] to couple the iPAM method with other fourth-order finite-volume methods [34, 32, 31] for solving the advection-diffusion-reaction equation and the incompressible Navier–Stokes equations with moving boundaries.

REFERENCES