Conservative and bounded volume-of-fluid advection on unstructured grids

By C. Ivey AND P. Moin

1. Motivation and objectives

Immiscible two-phase flows are distinguished by the existence of discontinuities at the fluid interface. Therefore, accurate representation of the interface evolution, while undergoing large topology changes, is integral to performing high-fidelity simulations of two-phase flow problems. The volume-of-fluid (VOF) methods (Hirt & Nichols 1981) track the interface by storing the ratio of the liquid volume to cell volume for each computational cell, which is referred to as the liquid volume fraction. VOF methods have the potential to provide mass conservation, which is important for many two-phase engineering applications with large density and/or breadth of length scales (Bravo *et al.* 2014), and they are the basis of this work. In the VOF method, the phase marker function $f(\vec{x},t)$ is an indicator of the phase present at position $\vec{x}$ and time t. The sharp interfaces are defined at the jump discontinuities of the phase marker function. The evolution of the phase marker function in an incompressible flow is governed by the conservative scalar advection partial differential equation,

$$f_t + \nabla \cdot (\vec{u} f) = 0, \tag{1.1}$$

where $\vec{u}(\vec{x},t)$ is the solenoidal velocity at position $\vec{x}$ and time t. The average of the continuous phase marker function over control volume Ω gives the liquid volume fraction of the control volume. The computational cells with $0 < F < 1$ contain an interface and they will be referred to as interfacial cells. VOF advection methods discretize Eq. (1.1) to evolve the F, which is used to reconstruct the interface.The standard interface reconstruction algorithm is the piecewise-linear interface calculation (PLIC) of Youngs (1982 Time-dependent multi-material flow with large fluid distortion.), in which the interface is locally approximated in the cell with a line or plane in two or three dimensions, respectively. As such, the interface in each cell is parameterized as

$$\hat{n} \cdot \vec{x} + C = 0, \tag{1.2}$$

where $\hat{n}$ is the normal vector of the interface, pointing outward with respect to the liquid, and C is the plane constant, which enforces the volume fraction in the cell. In this brief, the PLIC interface reconstruction is used in conjunction with the embedded-height function (EHF) estimate of the normal vector to provide a second-order accurate geometrical reconstruction of the local interface (see Ivey & Moin 2014, 2015); however, the advection scheme can readily be applied with other interface reconstructions. VOF advection schemes differ in how the liquid volume fraction is transported. Contemporary VOF advection schemes focus on unsplit advection, which avoids splitting errors and intermediate interface reconstructions of directionally-split schemes, and can be readily implemented on unstructured meshes. Unsplit VOF advection schemes are geometric in nature; the reconstructed flux regions strive for consistency with the space-time characteristics of the problem. The flux regions produced by the unsplit VOF advection community are

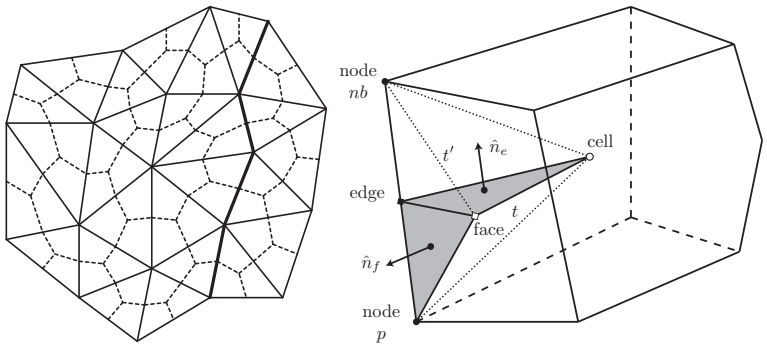


FIGURE 1: Two-dimensional illustration of a median-dual mesh constructed from a triangular primal mesh (a). Geometric details for three-dimensional node-based meshes (b).

becoming progressively more complicated, requiring ever-increasing efforts of the developer to robustly implement them (see survey by Comminal *et al.* 2015). These methods have become more conservative; however, the accuracy of the methods has not drastically improved. As such, the flux region impacts the accuracy less than the conservation and boundedness of the scheme. With this line of thought, a class of advection schemes have been developed that are conservative and bounded regardless of the accuracy of the flux region, the non-intersecting flux polyhedron advection (NIFPA) schemes. Unlike previously proposed unsplit VOF advection schemes (e.g, see Ivey & Moin 2012; Owkes & Desjardins 2014), the NIFPA algorithm is agnostic to the geometric definition of the flux region and is guaranteed to be conservative by ensuring that only the available volume is fluxed. NIFPA is formulated on a collocated node-centered three-dimensional unstructured mesh, which produces non-convex polyhedral elements (see Figure 1(a)). As such, the NIFPA relies on the non-convex polyhedral library described in Ivey & Moin (2015). Various NIFPA flux regions that fit in this class of schemes will be tested on different unstructured meshes. The accuracy and cost of each flux region definition will be compared to our adaptation of the method of Owkes & Desjardins (2014) to a collocated node-centered unstructured discretization, EMFPA-V^{cor}. EMFPA-V^{cor} uses the vertex velocities to produce the bounding edges of the flux region. The edges are forced to be conformal, and the tetrahedral flux volume correction is added onto the flux region capping face. The flux region faces are discretized by adding a face-centered vertex, which replaces the non-planar face with four triangular faces. The flux region is partitioned into signed tetrahedra, thereby simplifying the geometric computations and and easily handling flux regions that intersected with the flux face. There is potential for the flux volume correction to overlap and for the method to still produce non-admissable flux regions, such as hour-glass modes; however, this second-order accurate method can be considered the state-of-the-art VOF advection scheme. The implementation details are not included in this brief because of space limitations.

2. Numerical method

2.1. *Collocated node-based VOF advection with geometric fluxes*

This section discretizes Eq. (1.1) using the collocated node-based finite-volume flow solver developed in Ham *et al.* (2006). The median-dual node-based volumes, which are the control volumes for the collocated node-based finite-volume solver, are formed by tessellating each cell into sub-tetrahedra, each bounded by a node, an edge center, a face center, and a cell center. Figure 1(b) provides some geometric details on the construction of the median-dual mesh. $t \in T_p$ is a sub-tetrahedron of node p, where T_p is the set of all sub-tetrahedra associated to node p. $e \in E_p$ is the sub-edge of node p that corresponds to sub-tetrahedron t, where E_p is the set of all sub-edges associated to node p. Note that a sub-edge is a line segment in two dimensions and a triangle in three dimensions. $f \in F_p$ is a boundary sub-face of node p, where F_p is the set of all boundary sub-faces. Let Ω_p and $\delta\Omega_p$ be the node-based volume and surface associated to node p. Node nb, which lies on the same cell, face and edge as p, shares sub-edge e with p; i.e., $e = E_p \cap E_{nb}$. $t' \in T_{nb}$ is the sub-tetrahedron of node nb that shares sub-edge e with sub-tetrahedron t of node p. Unknowns, F and $\vec{v}$, and the grid coordinates, $\vec{x}$, are collocated at the nodes. Figure 1b provides some geometric details to be used in the VOF advection operator description. Following the derivation of Owkes & Desjardins (2014), Eq. (1.1) can be recast to form the evolution equation for f due to geometric fluxes (in the absence of boundaries),

$$\left(F_p^{n+1} - F_p^n\right) V_p + \sum_{e \in E_p} \left(\int_{\Omega_e^+(t^n)} f\left(\vec{x}, t^n\right) dV - \int_{\Omega_e^-(t^n)} f\left(\vec{x}, t^n\right) dV \right) = 0, \tag{2.1}$$

where $V_p = \int_{\Omega_p} dV$, $F_p^n = 1/V_p \int_{\Omega_p} f\left(\vec{x}, t^n\right) dV$, $\Omega_e^+\left(t^n\right)$ is the flux volume exiting the nodal volume p through sub-edge e, and $\Omega_e^-\left(t^n\right)$ is the flux volume entering the nodal volume p through sub-edge e. Note that Eq. (2.1) is still exact. We approximate flux volumes $\Omega_e^{+/-}(t^n)$ with flux polyhedra $\mathcal{P}_e^{+/-}(t^n)$. Various definitions can be applied to the sub-edge flux polyhedra, which differentiate the unsplit VOF schemes; explicit forms will be discussed later in the brief. To compute the integral of the marker function at time t^n, $f\left(\vec{x}, t^n\right)$, over sub-edge e's flux polyhedron, $\mathcal{P}_e^{+/-}(t^n)$, we need to approximate the interface in each nodal volume, $p \in \{no\}$, with a planar geometry. Define $\mathcal{H}_p^{\mathrm{PLIC}}(t^n)$ to be the half-space behind the plane defined by Eq. (1.2). Then calculate sub-edge e's flux polyhedron liquid volume at time step n, $F_{\mathcal{P}_e^+}^n V_{\mathcal{P}_e^+}^n$, using

$$F_{\mathcal{P}_e^{+/-}}^n V_{\mathcal{P}_e^{+/-}}^n = \int_{\Omega_e^{+/-}(t^n)} f\left(\vec{x}, t^n\right) dV = \sum_{p \in \{no\} \cap \mathcal{P}_e^{+/-}(t^n)} V\left(\mathcal{H}_p^{\mathrm{PLIC}}\left(t^n\right) \cap \mathcal{P}_e^{+/-}\left(t^n\right) \cap \mathcal{P}_p\right), \tag{2.2}$$

where $F_{\mathcal{P}_e^{+/-}}^n$ and $V_{\mathcal{P}_e^{+/-}}^n$ are sub-edge e's flux polyhedron's liquid volume fraction and total volume at time step n, $\{no\} \cap \mathcal{P}_e^{+/-}\left(t^n\right)$ is the set of all nodes that intersect $\mathcal{P}_e^{+/-}\left(t^n\right)$, $\mathcal{P}_p$ is the polyhedral representation of Ω_p, and $V(\cdot)$ is the volume operator (the same definitions as Ivey & Moin 2015). The flux polyhedron can potentially overlap with multiple nodal volumes, so the rightmost term in Eq. (2.2) requires a sum over all the nodes whose volume overlaps with the flux polyhedron. Substituting Eq. (2.2) into Eq. (2.1), we get an evolution equation for F_p,

$$F_p^{n+1} V_p = F_p^n V_p - \sum_{e \in E_p} \left(F_{\mathcal{P}_e^+}^n V_{\mathcal{P}_e^+}^n - F_{\mathcal{P}_e^-}^n V_{\mathcal{P}_e^-}^n \right). \tag{2.3}$$

Equation (2.3) states that the net change in the nodal liquid volume between time t^n and t^{n+1} is equal to the net amount of liquid volume in the entering sub-edge flux polyhedra minus the net amount of liquid volume in the exiting sub-edge flux polyhedra at time t^n.

2.2. *Non-intersecting flux polyhedron advection*

This subsection introduces novel definitions for the flux polyhedra utilized in Eq. (2.3) to complete the VOF advection scheme. The completed Eulerian-Lagrangian transport method is three-dimensional, unsplit, discretely conservative and bounded in F, and applicable to non-convex polyhedral meshes. The flux polyhedron reconstruction algorithm developed in this section satisfies conservation and boundedness of F irrespective of the flux polyhedron geometry. This is a different approach than current unsplit VOF schemes that prescribe topologically complicated flux polyhedron geometries in an effort to satisfy conservation (e.g., see Ivey & Moin 2012; Owkes & Desjardins 2014); these complicated flux polyhedron geometries require a large number of vertices, edges and faces. Hence, current VOF advection schemes' flux polyhedra have a large memory-footprint and require a large number of floating-point operations to perform a single polyhedron operation. Furthermore, they are prone to topological failures in the polyhedron definition (e.g., hour-glass modes). Lastly, these schemes are still not discretely conservative and bounded in F; although small, their flux corrections still potentially have overlapping volume. In lieu of prescribing complicated flux-polyhedron geometries, this VOF advection scheme, the non-intersecting flux polyhedron advection (NIFPA) method, builds the flux polyhedron iteratively using Brent's method (Brent 1973) such that its intersection with neighboring flux polyhedra, and any other unavailable volume, is empty and its total volume matches the calculated flux volume. For sub-edge e between nodes p and nb, define $\mathcal{P}^a_{p/nb}$ and $\mathcal{P}^u_{p/nb}$ to be the polyhedral representations of the additional and unavailable volume in the node/neighbor, respectively. $\mathcal{P}^a_p$ and $\mathcal{P}^u_p$ store all previously reconstructed flux polyhedra that have entered and exited node p. At the start of the time step, t^n, initialize $\mathcal{P}^{a/u}_p = \emptyset$ for all nodes p. Each iteration, k, consists of three steps:

(a) forming nominal flux polyhedron, $\tilde{\mathcal{P}}_e$,which is parameterized by the current iteration extrusion pseudo-time, $\Delta\tau^k$,

$$\tilde{\mathcal{P}}_e\left(\Delta\tau^k\right) = \tilde{\mathcal{P}}^+_e\left(\Delta\tau^k\right) \cup \tilde{\mathcal{P}}^-_e\left(\Delta\tau^k\right), \tag{2.4}$$

where $\tilde{\mathcal{P}}^+_e$ and $\tilde{\mathcal{P}}^-_e$ are the nominal exiting and entering flux polyhedra in node p (or equivalently the entering and exiting flux polyhedra in node nb), respectively;

(b) forming the actual exiting/entering flux polyhedron, $\mathcal{P}^{+/-}_e$, by intersecting the nominal exiting/entering flux polyhedron, $\tilde{\mathcal{P}}^{+/-}_e$, with the node/neighbor polyhedron, $\mathcal{P}_{p/nb}$, adding the intersection of the nominal exiting/entering flux polyhedron, $\tilde{\mathcal{P}}^{+/-}_e$, with previously constructed flux polyhedra that have entered the node/neighbor, $\mathcal{P}^a_{p/nb}$, and subtracting the intersection of the nominal exiting/entering flux polyhedron, $\tilde{\mathcal{P}}^{+/-}_e$, with previously constructed flux polyhedra that have exited the node/neighbor, $\mathcal{P}^u_{p/nb}$, which can be compactly written as

$$\mathcal{P}^{+/-}_e\left(\Delta\tau^k\right) = \tilde{\mathcal{P}}^{+/-}_e\left(\Delta\tau^k\right) \cup \left(\left(\mathcal{P}_{p/nb} \cup \mathcal{P}^a_{p/nb}\right) \setminus \mathcal{P}^u_{p/nb}\right), \tag{2.5}$$

where $\left(\mathcal{P}_p \cup \mathcal{P}^a_p\right) \setminus \mathcal{P}^u_p$ is the total available volume to node p; and

(c) comparing the signed volume of the resultant flux polyhedron, $V(\mathcal{P}^+_e\left(\Delta\tau^k\right)) -$

$V(\mathcal{P}_e^-\left(\Delta\tau^k\right))$, to the flux volume,

$$V_e^{\text{flux}} = |\vec{U}_e^{n+1/2} \cdot \hat{n}_e| A_e \Delta t, \tag{2.6}$$

to determine the extrusion pseudo-time for the next iteration, $\Delta\tau^{k+1}$, where

$$\begin{aligned} V\left(\mathcal{P}_e^{+/-}\left(\Delta\tau^k\right)\right) = V\left(\tilde{\mathcal{P}}_e^{+/-}\left(\Delta\tau^k\right) \cap \mathcal{P}_{p/nb}\right) + V\left(\tilde{\mathcal{P}}_e^{+/-}\left(\Delta\tau^k\right) \cap \mathcal{P}_{p/nb}^a\right) \\ - V\left(\tilde{\mathcal{P}}_e^{+/-}\left(\Delta\tau^k\right) \cap \mathcal{P}_{p/nb}^u\right). \end{aligned} \tag{2.7}$$

The iteration procedure continues until

$$V_e^{\text{flux}} = V\left(\mathcal{P}_e^+\left(\Delta\tau^{\text{final}}\right)\right) - V\left(\mathcal{P}_e^-\left(\Delta\tau^{\text{final}}\right)\right), \tag{2.8}$$

where $k = \text{final}$. Once the iteration is complete, the node's and sub-edge neighbor's additional and unavailable polyhedra are updated as

$$\mathcal{P}_p^a = \left\{\mathcal{P}_p^a, \mathcal{P}_e^-\right\}, \mathcal{P}_p^u = \left\{\mathcal{P}_p^u, \mathcal{P}_e^+\right\}, \mathcal{P}_{nb}^a = \left\{\mathcal{P}_{nb}^a, \mathcal{P}_e^+\right\}, \text{ and } \mathcal{P}_{nb}^u = \left\{\mathcal{P}_{nb}^u, \mathcal{P}_e^-\right\}, \tag{2.9}$$

ensuring that subsequently constructed flux polyhedron only donates available volume. Repeat the iteration procedure for each edge of each face of each cell in the domain to construct all of the sub-edge flux polyhedra, $\mathcal{P}_e^{+/-}$. Once all of the $\mathcal{P}_e^{+/-}$ have been formed, the nodal volume fractions, F_p^n, can be advanced to F_p^{n+1} using Eq. (2.3) (in the absence of boundaries). The NIFPA algorithm, described above, is conservative and bounded, regardless of the definition of the nominal flux polyhedra, $\tilde{\mathcal{P}}_e^{+/-}$, because the resultant sub-edge flux polyhedra, $\mathcal{P}_e^{+/-}$, only overlap with volume available to the node/neighbor at the time of the flux calculation, $\left(\mathcal{P}_{p/nb} \cup \mathcal{P}_{p/nb}^a\right) \setminus \mathcal{P}_{p/nb}^u$. The algorithm utilizes the polyhedral operations of the EHF method described in Ivey & Moin (2015). The cost of the method depends on the topological complexity of the nominal flux polyhedron and the number iterations required by Brent's method to satisfy Eq. (2.8). The accuracy of the approach depends on how accurately the nominal flux polyhedra, $\mathcal{P}_e^{+/-}$, approximate the corresponding flux volumes, $\Omega_e^{+/-}$. The NIFPA method avoids conservation and boundedness errors by preventing flux polyhedra from overlapping and preventing the donation of volume before it is received, respectively. The construction of $\mathcal{P}_e^{+/-}$ from Eq. (2.5) allows for en-route volume from previously reconstructed flux polyhedra, but it cannot handle en-route volume for the subsequently reconstructed flux polyhedron. Furthermore, NIFPA does not prevent the diffusive errors that come from gaps between adjacent flux polyhedra. The impact of the diffusive error and the error incurred by the incomplete accommodation of the en-route volume on the NIFPA's accuracy will be assessed by comparing its total shape error, using various nominal flux polyhedron definitions, to that of the EMFPA-V^{cor} scheme. Figure 2 illustrates three sequential two-dimensional sub-edge non-intersecting flux polyhedron reconstructions: $\mathcal{P}_e^+$, $\mathcal{P}_{e'}^+$, and $\mathcal{P}_{e''}^+$. The corresponding nominal flux polyhedra, $\tilde{\mathcal{P}}_e^+$, $\tilde{\mathcal{P}}_{e'}^+$, and $\tilde{\mathcal{P}}_{e''}^+$, are constructed by extruding the sub-edge face in the direction opposite to the sub-edge velocity, $\vec{U}_e^{n+1/2}$, a distance $d = \|\vec{U}_e^{n+1/2}\| \Delta\tau^{\text{final}}$. For prescribed continuous velocity fields, $\vec{u}_p^{n+1/2}$ are sampled from the continuous velocity field at position $\vec{x} = \vec{x}_p$ and time $t = t^{n+1/2}$, $\vec{u}_p^{n+1/2} = \vec{u}\left(\vec{x}_p, t^{n+1/2}\right)$. These velocities are not constrained to be conservative. The conservative sub-edge velocities, $\vec{U}_e^{n+1/2}$, are computed by projecting the $\vec{u}_p^{n+1/2}$ to be divergence-free over p's nodal volume using a Poisson equation. For velocity fields which are not prescribed, as is the case for the two-phase Navier-Stokes equations, $t^{n+1/2}$ val-

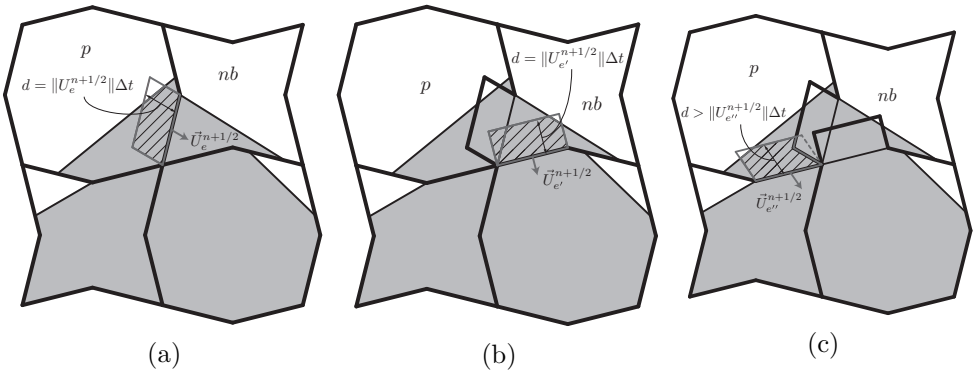


FIGURE 2: Three sequentially reconstructed non-intersecting flux polyhedra: (a) $\mathcal{P}_e^+$, (b) $\mathcal{P}_{e'}^+$, and (c) $\mathcal{P}_{e''}^+$.

ues are obtained from values at t^n and t^{n-1} using Adams-Bashforth extrapolation. The sub-edge flux polyhedron's volume matches the flux volume calculated using Eq. (2.6). In Figure 2(a), the flux polyhedron equals the final nominal flux polyhedron intersected with the node polyhedron. In Figure 2(b), the flux polyhedron equals the final nominal flux polyhedron intersected with the node polyhedron plus the intersection of the final nominal flux polyhedron with the flux polyhedron calculated in Figure 2(a) (additional volume). In Figure 2(c), the flux polyhedron equals the final nominal flux polyhedron intersected with donor cell polyhedron minus the intersection of the final nominal flux polyhedron with the flux polyhedron calculated in Figure 2(c) (unavailable volume). Unfortunately, as evidenced by Figure 2, the NIFPA scheme is order dependent. For example, reconstructing sub-edge e'' first, e' second, and e third will result in $\mathcal{P}_{e''}^+$, $\mathcal{P}_{e'}^+$ $\mathcal{P}_e^+$ having shapes different from those displayed in Figures 2(c,b,a), respectively. To avoid bias, at each time step, we randomize the order in which we traverse the cells, faces of a cell, and nodes of a face. The definition for the nominal flux polyhedron used here is an example, and it is not in any way set. The next section will review a few possible nominal flux polyhedron descriptions.

2.2.1. *Nominal flux polyhedra*

The NIFPA algorithm conserves volume regardless of the nominal flux polyhedron; however, the accuracy of the method depends on how accurately the nominal flux polyhedron approximates the associated streak tube. The computational cost of the method depends on the total number polyhedron intersections required by Eq. (2.3) for all the iterations during the construction of the final actual flux polyhedron. In order to follow the procedure described in Section 2.2, the nominal flux polyhedron geometry needs to be parameterized by a single scalar. Following the ideology of Eulerian-Lagrangian transport schemes (Comminal *et al.* 2015), the class of nominal flux polyhedron used here is formed by tracking the sub-edge face corner vertices backwards in pseudo-time, $\Delta\tau$. $\Delta\tau^k$ parameterizes iteration k's nominal flux polyhedron geometry, $\tilde{\mathcal{P}}_e\left(\Delta\tau^k\right)$, and it is updated by the iteration procedure in Section 2.2 until Eq. (2.8) is satisfied. Note, $\Delta\tau^{\text{final}}$ does not necessarily match Δt as shown in Figure 2. For each of the nodes of sub-edge e's face, $\vec{x}_{e,v}$, the respective nodes of the capping face of the nominal flux polyhedron, $\vec{x}_{c,v}$, are calculated using

$$\vec{x}_{c,v}^k = \vec{x}_{e,v} - \vec{u}_{e,v} \cdot \hat{d}_e \hat{d}_e \Delta\tau^k, \tag{2.10}$$

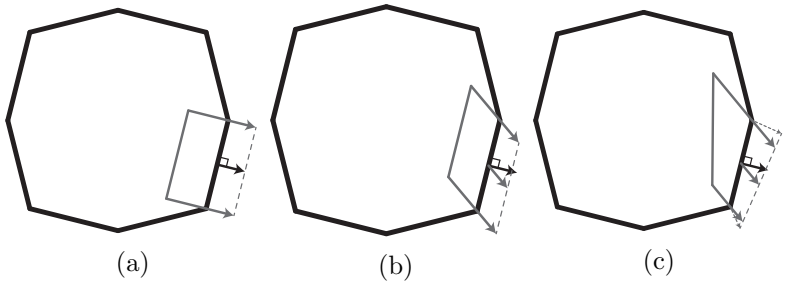


FIGURE 3: Two-dimensional non-intersection flux polyhedron advection (NIFPA) approximations of streak tube using various definitions for $\vec{u}_{e,v}$ and $\hat{d}_e$ from Eq. (2.10): NIFPA-0 (a), NIFPA-1 (b) and NIFPA-2 (c).

where $\vec{u}_{e,v}$ is a sub-edge corner vertex velocity and $\hat{d}_e$ is a sub-edge unit direction. By formulation, Eq. (2.10) constrains the bounding edges of the nominal flux polyhedron to be parallel to a single direction, $\hat{d}_e$. Equation (2.10) enforces planarity of the flux polyhedron surfaces, precluding the need for triangulation, and prevents the edges that approximate the streak lines from crossing, ensuring topological robustness of the flux polyhedron reconstruction. Unlike the NIFPA algorithm, these constraints would result in large conservation errors in contemporary unsplit schemes e.g. (e.g, Hernández *et al.* 2008; Ivey & Moin 2012; Owkes & Desjardins 2014; Comminal *et al.* 2015). Further, this approximation produces one convex polyhedron if the signs of $\vec{u}_{e,v}$ match for all v of e and two convex polyhedra otherwise, in which case the nominal flux polyhedron self-intersects across the sub-edge face. The constraint of at most two convex polyhedra was chosen in order to reduce the total number of intersections required by all iterations of Brent's method in Section 2.2. The EMFPA-V^{cor} scheme requires 16 tetrahedral geometries to represent the flux polyhedron on the median-dual mesh. Each of these tetrahedra will need to intersect with the node volumes, $\mathcal{P}_p$, and respective PLIC half-spaces, $\mathcal{H}_p^{\text{PLIC}}$, for all the nodes p that potentially overlap with the tetrahedron (see Eq. (2.3)). To keep the NIFPA algorithm competitive with the EMFPA-V^{cor} scheme in terms of computational cost, we need to have the number of convex polyhedra making up the flux polyhedron times the number of iterations in Brent's algorithm to be less than 16. This is an inexact heuristic because EMFPA-V^{cor}'s flux polyhedra are composed strictly of tetrahedra, whereas the NIFPA flux polyhedra can be composed of tetrahedra, pyramids, and/or wedges; however, the methods are deemed comparable in terms of computational cost in the verification tests of Section 3. The nominal flux polyhedra tested here differ only in the definition of $\vec{u}_{e,v}$ and $\hat{d}_e$. More specifically, we test the NIFPA algorithm using $\vec{u}_{e,v} \cdot \hat{d}_e \hat{d}_e = \vec{U}_e^{n+1/2} \cdot \hat{n}_e \hat{n}_e$ ($\vec{u}_{e,v} = \vec{U}_e^{n+1/2}$ and $\hat{d}_e = \hat{n}_e$), labeled NIFPA-0, $\vec{u}_{e,v} \cdot \hat{d}_e \hat{d}_e = \vec{U}_e^{n+1/2}$ ($\vec{u}_{e,v} = \vec{U}_e^{n+1/2}$ and $\hat{d}_e = \vec{U}_e^{n+1/2}/\|\vec{U}_e^{n+1/2}\|$), labeled NIFPA-1, and $\vec{u}_{e,v} \cdot \hat{d}_e \hat{d}_e = \vec{u}_{e,v} \cdot \vec{U}_e^{n+1/2} \vec{U}_e^{n+1/2}/\|\vec{U}_e^{n+1/2}\|^2$ ($\vec{u}_{e,v} = \vec{u}_{e,v}^{n+1/2}$ and $\hat{d}_e = \vec{U}_e^{n+1/2}/\|\vec{U}_e^{n+1/2}\|$), labeled NIFPA-2. NIFPA-1 was used in Figure 2 to illustrate the NIFPA algorithm. $\vec{u}_{e,v}^{n+1/2}$ is the velocity of corner vertex v of sub-edge e at time $t^{n+1/2}$, which is calculated from simple averages of $\vec{u}_p^{n+1/2}$. Figures 3(a)-3(c) illustrate three two-dimensional representations of the nominal flux polyhedron definitions of the true flux volume (streak tube).

2.2.2. *Flux correction and time-step limitation*

The nominal flux polyhedra illustrated in Figure 3, $\tilde{\mathcal{P}}_e^{+/-}\left(\Delta\tau^k\right)$, are resized iteratively using Brent's method. For a prescribed maximum number of iterations, $k^{\max}$, it is possible for Eq. (2.8) to not be satisfied. This results in the remaining flux volume, V_e^{rem}, defined as

$$V_e^{\text{rem}} = V_e^{\text{flux}} - \left(V\left(\mathcal{P}_e^{+}\left(\Delta\tau^{\max}\right)\right) - V\left(\mathcal{P}_e^{-}\left(\Delta\tau^{\max}\right)\right)\right). \tag{2.11}$$

This undesirable situation occurs for potentially two reasons. First, $k^{\max}$ could be prescribed to be too small (to reduce computational cost); i.e., there exists a $k^{\text{final}} > k^{\max}$ such that Eq. (2.8) is satisfied to a desired tolerance, $|V_e^{\text{rem}}/V_e^{\text{flux}}| < \epsilon_{\text{flux}}$. This situation can be avoided by either increasing $k^{\max}$ or ϵ_{flux}. Second, it is possible that the region defined by Eq. (2.5) is insufficient regardless of $\Delta\tau$. This situation is illustrated in Figure 4(a), where further iterations result in the same flux polyhedron volume, $V_e^{\text{flux}} - V_e^{\text{rem}}$. To capture the remaining flux volume, we propose adding a small additional flux polyhedron, $\delta\mathcal{P}_e^{+/-}$, with volume V_e^{rem}. Prior to calculating the additional flux polyhedron, the additional and unavailable regions need to be updated, using Eq. (2.9) with $\mathcal{P}_e^{+/-}\left(\Delta\tau^{\max}\right)$. Then, $\delta\mathcal{P}_e^{+/-}$ is calculated by shifting the sub-edge face plane, oriented in $\hat{n}_e$'s direction, such that the volume of the intersection of the half-space ahead/behind the plane, $\mathcal{H}_e^{+/-}$, and p/nb's available volume, $\left(\mathcal{P}_{p/nb} \cup \mathcal{P}^a_{nb/p}\right) \setminus \mathcal{P}^u_{p/nb}$, is V_e^{rem}; i.e.,

$$\begin{aligned} \delta\mathcal{P}_e^{+/-} &= \begin{cases} \mathcal{H}_e^{+}\left(\Delta\tau^k\right) \cup \left(\left(\mathcal{P}_p \cup \mathcal{P}_p^a\right) \setminus \mathcal{P}_p^u\right) / \emptyset & \text{if } \vec{U}_e^{n+1/2} \cdot \hat{n}_e > 0 \\ \emptyset / \mathcal{H}_e^{-}\left(\Delta\tau^k\right) \cup \left(\left(\mathcal{P}_p \cup \mathcal{P}_p^a\right) \setminus \mathcal{P}_p^u\right) & \text{if } \vec{U}_e^{n+1/2} \cdot \hat{n}_e < 0 \end{cases} \\ V_e^{\text{rem}} &= V\left(\delta\mathcal{P}_e^{+}\right) - V\left(\delta\mathcal{P}_e^{-}\right), \end{aligned} \tag{2.12}$$

Once $\delta\mathcal{P}_e^{+,-}$ have been calculated, the additional and unavailable regions are updated using

$$\mathcal{P}_p^a = \left\{\mathcal{P}_p^a, \delta\mathcal{P}_e^{-}\right\}, \mathcal{P}_p^u = \left\{\mathcal{P}_p^u, \delta\mathcal{P}_e^{+}\right\}, \mathcal{P}_{nb}^a = \left\{\mathcal{P}_{nb}^a, \delta\mathcal{P}_e^{+}\right\}, \text{ and } \mathcal{P}_{nb}^u = \left\{\mathcal{P}_{nb}^u, \delta\mathcal{P}_e^{-}\right\}. \tag{2.13}$$

This correction can be seen as repeating the NIFPA algorithm using $\tilde{\mathcal{P}}_e^{+} = \mathcal{H}_e^{+}$ and $\tilde{\mathcal{P}}_e^{-} = \emptyset$. Figure 4(b) illustrates the flux polyhedron correction to the flux polyhedron shown in Figure 4(a). The resultant positively and negatively signed flux polyhedra are then $\mathcal{P}_e^{+}\left(\Delta\tau^{\max}\right) + \delta\mathcal{P}_e^{+}$ and $\mathcal{P}_e^{-}\left(\Delta\tau^{\max}\right) + \delta\mathcal{P}_e^{-}$, respectively. The proposed flux correction definition, as well as that of the nominal flux polyhedron, is not unique. The advantage of using the flux correction defined by Eq. (2.12) is that it can potentially span the entire region defined by $\left(\mathcal{P}_p \cup \mathcal{P}_p^a\right) \setminus \mathcal{P}_p^u$. This allows for a well-defined time-step limit to be placed on the NIFPA scheme for certain nominal flux polyhedron definitions. Using Eq. (2.8), augmented with the flux correction, and continuity, $\sum_{e \in E_p} \vec{U}_e^{n+1/2} \cdot \hat{n}_e A_e \Delta t = 0$, we have

$$\sum_{e \in E_p} \vec{U}_e^{n+1/2} \cdot \hat{n}_e A_e \Delta t = \sum_{e \in E_p} V\left(\mathcal{P}_e^{+}\left(\Delta\tau^{\max}\right) + \delta\mathcal{P}_e^{+}\right) - V\left(\mathcal{P}_e^{-}\left(\Delta\tau^{\max}\right) + \delta\mathcal{P}_e^{-}\right) = 0, \tag{2.14}$$

which states that the volume entering the node must balance the volume leaving the node. Because the flux polyhedra are calculated sequentially, it is possible for all the exiting flux polyhedra to be reconstructed before any entering flux polyhedra are reconstructed. As such, the sum of all the exiting flux polyhedra volume must be less than the nodal volume to ensure only available volume is donated and, from Eq. (2.14), the same must be

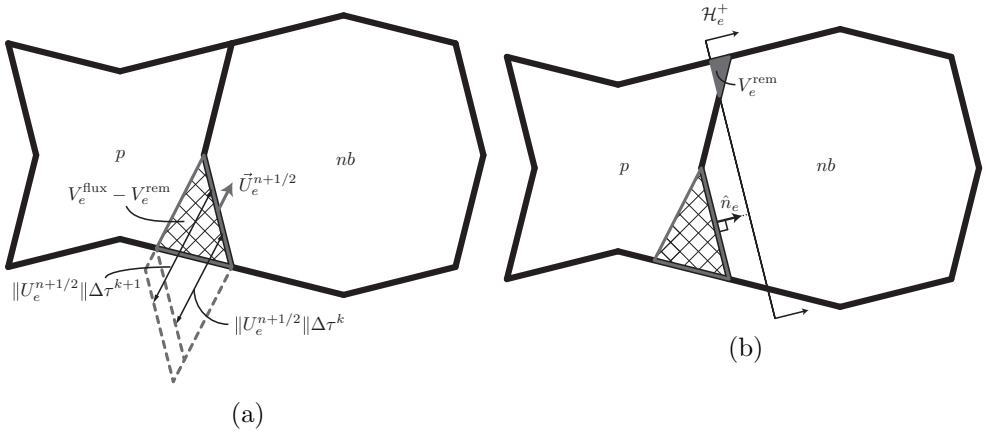


FIGURE 4: Two-dimensional schematic of NIFPA flux correction. (a) Example of a flux polyhedron which is unable to satisfy V_e^{flux} by increasing $\Delta\tau$. (b) Proposed flux correction V_e^{rem}, which ensures that the volume of the net flux polyhedron equals V_e^{flux}.

true for entering flux polyhedra to ensure that only available volume is received; adding these constraints yields

$$\sum_{e\in E_p} V\left(\mathcal{P}_e^+\left(\Delta\tau^{\max}\right)+\delta\mathcal{P}_e^+\right)+V\left(\mathcal{P}_e^-\left(\Delta\tau^{\max}\right)+\delta\mathcal{P}_e^-\right)\le 2V_p. \tag{2.15}$$

In order to get Eq. (2.15) in a form where we can extract a time-step limit, we need to restrict ourselves to NIFPA nominal flux polyhedra where the flux polyhedron is either entering or leaving for a sub-edge, not both. With this requirement, Eq. (2.15) can be expressed as $\sum_{e\in E_p}|\vec{U}_e^{n+1/2}\cdot\hat{n}_e|A_e\Delta t\le 2V_p$. To reconstruct all the flux polyhedra of all the nodes of the mesh with a single time step, we need

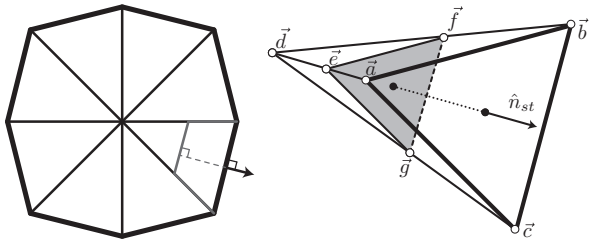
$$\text{CFL}_{\text{NIFPA}*}=\max_{p\in\{no\}}\sum_{e\in E_p}|\vec{U}_e^{n+1/2}\cdot\hat{n}_e|A_e\Delta t/V_p\le 2, \tag{2.16}$$

where $\text{CFL}_{\text{NIFPA}*}$ is the Courant-Friedrichs-Lewy number for the NIFPA algorithms whose nominal flux polyhedra have only entering or leaving volume for each sub-edge. The NIFPA-0 and NIFPA-1 schemes both meet this constraint. Unfortunately, we could not create a well-defined time-step limit for NIFPA-2 because there are sub-edges with both entering and exiting volumes. With both entering and leaving volumes, this flux polyhedron's vertices could extend beyond the available volume even if we limit the flux of each sub-edge using Eq. (2.16).

2.2.3. *Sub-tetrahedron flux polyhedron advection*

Figure 5(a) is a special case of the NIFPA algorithm: the sub-edge e's flux polyhedron is constrained to sub-tetrahedron t/t' of the nodal volume p/nb. As such, the additional and unavailable volumes for the flux polyhedron are fixed, $\mathcal{P}^a_{p/nb}=\emptyset$ and $\mathcal{P}^u_{p/nb}=\mathcal{P}_{p/nb}\setminus\mathcal{S}_{t/t'}$. Because t/t' is a sub-tetrahedron of node p/nb, $\mathcal{S}_{t/t'}\cap\mathcal{P}_{p/nb}=\mathcal{S}_{t/t'}$. With these simplifications, Eq. (2.5) reduces to

$$\mathcal{P}_e^{+/-}\left(\Delta\tau^k\right)=\begin{cases}\tilde{\mathcal{P}}_e^+\left(\Delta\tau^k\right)\cup\mathcal{S}_t/\emptyset & \text{if } \vec{U}_e^{n+1/2}\cdot\hat{n}_e>0\\ \emptyset/\tilde{\mathcal{P}}_e^-\left(\Delta\tau^k\right)\cup\mathcal{S}_{t'} & \text{if } \vec{U}_e^{n+1/2}\cdot\hat{n}_e<0\end{cases}. \tag{2.17}$$



(a) (b)

FIGURE 5: Two-dimensional sub-tetrahedron flux polyhedron approximation of streak tube (a). Geometric details for the STFPA algorithm (b).

By replacing Eq. (2.5) with Eq. (2.17) and defining $v_{e,v}\cdot\hat{d}_e\hat{d}_e = \vec{U}_e^{n+1/2}\cdot\hat{n}_e\hat{n}_e$ in Eq. (2.10), we can utilize the NIFPA algorithm in Section 2.2 to build the iterative sub-tetrahedron flux polyhedron advection (STFPA) algorithm. However, the constraints on the STFPA flux polyhedron are sufficient to avoid iterations and to reconstruct the flux polyhedron analytically, providing a fast algorithm. Figure 5(b) is a pictorial representation of the STFPA flux polyhedron. Vertices $\vec{a}$, $\vec{b}$, and $\vec{c}$ are the corners of the sub-edge associated with the sub-tetrahedron $\left\{\vec{a},\vec{b},\vec{c},\vec{d}\right\}$. $\hat{n}_{st}$ is the outward pointing normal to face $\left\{\vec{a},\vec{b},\vec{c}\right\}$ of sub-tetrahedron st, where

$$\mathcal{S}_{st} = \begin{cases} \mathcal{S}_t & \text{if } \vec{U}_e^{n+1/2}\cdot\hat{n}_e > 0 \\ \mathcal{S}_{t'} & \text{if } \vec{U}_e^{n+1/2}\cdot\hat{n}_e < 0 \end{cases}, \hat{n}_{st} = \begin{cases} \hat{n}_e & \text{if } \vec{U}_e^{n+1/2}\cdot\hat{n}_e > 0 \\ -\hat{n}_e & \text{if } \vec{U}_e^{n+1/2}\cdot\hat{n}_e < 0 \end{cases}. \tag{2.18}$$

Under these definitions, $\mathcal{S}_{st} = \left\{\vec{a},\vec{b},\vec{c},\vec{d}\right\}$ is the donor sub-tetrahedron and $\vec{U}_e^{n+1/2}\cdot\hat{n}_{st} > 0$. Vertices $\vec{e}$, $\vec{f}$ and $\vec{g}$ are the corners of the capping face to the flux polyhedron. The capping face's normal matches $\hat{n}_{st}$. The capping face's plane constant is defined as C_{st}. To get a closed form solution to the flux polyhedron vertices, $\left\{\vec{e},\vec{f},\vec{g}\right\}$ are parameterized in terms of the capping face plane, C_{st} and $\hat{n}_{st}$, as well as the vertices that bound the line segment of the sub-tetrahedron they reside on, $\vec{d}$ and $\left\{\vec{a},\vec{b},\vec{c}\right\}$, respectively:

$$\vec{e} = \vec{d} - \frac{\hat{n}_{st}\cdot\vec{d}+C_{st}}{\hat{n}_{st}\cdot\left(\vec{a}-\vec{d}\right)}\left(\vec{a}-\vec{d}\right), \vec{f} = \vec{d} - \frac{\hat{n}_{st}\cdot\vec{d}+C_{st}}{\hat{n}_{st}\cdot\left(\vec{b}-\vec{d}\right)}\left(\vec{b}-\vec{d}\right), \vec{g} = \vec{d} - \frac{\hat{n}_{st}\cdot\vec{d}+C_{st}}{\hat{n}_{st}\cdot\left(\vec{c}-\vec{d}\right)}\left(\vec{c}-\vec{d}\right). \tag{2.19}$$

The volume of sub-tetrahedron $\left\{\vec{a},\vec{b},\vec{c},\vec{d}\right\}$, V_{st}, is calculated using $V_{st} = SV\left(\left\{\vec{a},\vec{b},\vec{c},\vec{d}\right\}\right)$, where

$$SV\left(\left\{\vec{a},\vec{d},\vec{g},\vec{j}\right\}\right) = -\left(\vec{a}-\vec{j}\right)\cdot\left(\left(\vec{d}-\vec{j}\right)\times\left(\vec{g}-\vec{j}\right)\right)/6, \tag{2.20}$$

and $SV(\cdot)$ is the signed-volume operator. The STFPA flux polyhedron is bound by sub-edge and capping-face vertices $\left\{\vec{a},\vec{b},\vec{c}\right\}$ and $\left\{\vec{e},\vec{f},\vec{g}\right\}$, respectively, and its volume must equal the flux volume, V_e^{flux}:

$$V_e^{\text{flux}} = |\vec{U}_e^{n+1/2}\cdot\hat{n}_e|A_e\Delta t = V_{st} - SV\left(\left\{\vec{e},\vec{f},\vec{g},\vec{d}\right\}\right). \tag{2.21}$$

Substituting Eq. (2.19) into Eq. (2.21) yields a definition for C_{st} in terms of knowns,

$$C_{st} = \sqrt[3]{\left(\frac{|\vec{U}_e^{n+1/2} \cdot \hat{n}_e| A_e \Delta t}{V_{st}} - 1\right) \left[\hat{n}_{st} \cdot \left(\vec{a} - \vec{d}\right)\right] \left[\hat{n}_{st} \cdot \left(\vec{b} - \vec{d}\right)\right] \left[\hat{n}_{st} \cdot \left(\vec{c} - \vec{d}\right)\right] - \hat{n}_{st} \cdot \vec{d}}. \tag{2.22}$$

Once C_{st} is determined from Eq. (2.22), vertices $\vec{e}$, $\vec{f}$ and $\vec{g}$ can be calculated using Eq. (2.22), providing a closed-form expression for the STFPA flux polyhedron. In order for the capping face to stay inside the sub-tetrahedron, $\vec{e}$, $\vec{f}$, and $\vec{g}$ must reside on the line segments bound by $\vec{d}$ and $\vec{a}$, $\vec{d}$ and $\vec{b}$, and $\vec{d}$ and $\vec{c}$, respectively. This constraint is satisfied when $0 \leq |\vec{U}_e^{n+1/2} \cdot \hat{n}_e| A_e \Delta t / V_{st} \leq 1$, which prevents conservation and boundedness errors by forcing that the STFPA flux polyhedron remains within the sub-tetrahedron; if the flux polyhedron were allowed a larger time step, the algorithm would create over- and undershoots. To reconstruct all of the flux polyhedra in all the nodes of the mesh with a single time step, we need

$$\mathrm{CFL}_{\mathrm{STFPA}} = \max_{p \in \{no\}} \left(\max_{e \in E_p, t \in T_p} \left(|\vec{U}_e^{n+1/2} \cdot \hat{n}_e| A_e \Delta t_{\mathrm{STFPA}} / V_{st}\right)\right) \leq 1, \tag{2.23}$$

where $\mathrm{CFL}_{\mathrm{STFPA}}$ is the Courant-Friedrichs-Lewy number for the STFPA algorithm. Equation (2.23) is a tighter time-step restriction than Eq. (2.16), so STFPA requires a smaller time step than NIFPA-0 and NIFPA-1; however, it is possible for NIFPA-2 to require a smaller time step than STFPA.

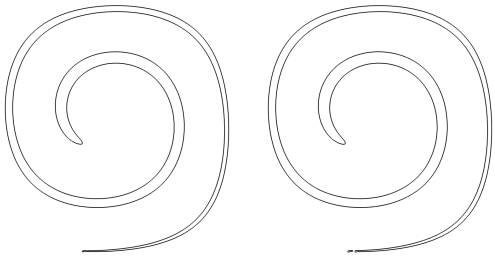
3. Verification tests

The proposed schemes are studied using two solenoidal velocity tests: two-dimensional deformation and three-dimensional deformation. Contemporary unsplit VOF advection methods are suitable for unstructured meshes; however, to the best of our knowledge, the verification tests have concentrated on structured two-dimensional and three-dimensional cartesian meshes. A key contribution of this study to the VOF community is to measure the cost and accuracy of the formulated NIFPA, STFPA, and EMFPA-V^{cor} schemes on collocated node-based unstructured meshes. Specifically, the tests are performed on quadrilateral and triangular primal meshes for the two-dimensional deformation test case and hexahedral and tetrahedral primal meshes for the three-dimensional deformation test case. The two-dimensional primal meshes have a width of a single cartesian cell and are periodic in the z direction. The simplex (triangle and tetrahedron) mesh elements' uniform edge lengths, Δ, are prescribed such that their respective cell areas or volumes match those of the quadrilateral or hexahedral mesh of the same cell count. Specifically, $\Delta^2_{\mathrm{triangle}} \approx 4/\sqrt{3}\Delta^2_{\mathrm{quadrilateral}}$ and $\Delta^3_{\mathrm{tetrahedron}} \approx 6\sqrt{2}\Delta^3_{\mathrm{hexahedron}}$. Unfortunately, the edge constraints could only be approximately satisfied, so the simplex elements are only approximately uniform in edge length. Furthermore, the uniformity constraint on the edge length resulted in fewer cells in the simplex primal meshes. Although the primal mesh elements are simple convex polyhedra, the node volumes of the median-dual mesh can be complicated in structure (see Ivey & Moin 2015). Furthermore, the ratio of the number of nodes in the mesh to the number of cells depended on the primal mesh element. In two dimensions, this ratio is smaller for the triangle than for the quadrilateral elements. While in three dimensions, this ratio is smaller for the tetrahedron than for the hexahedron elements. In order to compare the accuracy of the schemes, all the tests are performed using the time step from Eq. (2.23) with a constant $\mathrm{CFL}_{\mathrm{STFPA}} = 1$. This definition for the time

step is the largest time step which kept all the schemes conservative and bounded, as defined by the prescribed tolerance, $\epsilon = 10^{-12}$. For the EMFPA-V^{cor} scheme, this suggests that the conservation and boundedness errors from the flux correction or non-admissible polyhedra are insignificant. The NIFPA-0 and NIFPA-1 schemes are guaranteed to be conservative and bounded because $\text{CFL}_{\text{SFTPA}}$ is a smaller time-step restriction than $\text{CFL}_{\text{NIFPA}*}$. NIFPA-2 is not guaranteed to be conservative with $\text{CFL}_{\text{NIFPA}*}$, but it too had insignificant conservation and boundedness errors using $\text{CFL}_{\text{SFTPA}}$. For the time-periodic flows tested here, the $\mathcal{L}_1(F) = \sum_{p\in\{no\}} |F_p(t=T) - F_p(t=0)|V_p / \sum_{p\in\{no\}} V_p$ shape error norm is used to measure the accuracy of the VOF schemes, where T is the period. The conservation error, $\sum_{p\in\{no\}} \left(F_p(t^n) - F_p(t=0)\right) V_p / \sum_{p\in\{no\}} V_p$, and the boundedness error, $\max\left(-\min_{p\in\{no\}} F_p(t^n), \max_{p\in\{no\}} \left(F_p(t^n) - 1\right)\right)$, were checked at each time step, n, to ensure each were less than the prescribed tolerance. To assess the computational cost of the geometric VOF advection schemes, we compare the average time required to reconstruct a flux polyhedron using STFPA using the iterative procedure of Section 2.2 (STFPA-it), NIFPA-0, NIFPA-1, NIFPA-2, and EMFPA-V^{cor} with that of STFPA using the analytic relations of Eqs. (2.19) and (2.22) (STFPA-an). Simulations were performed using a single processor of a large memory node that utilizes dual 6-core X5650 2.67 GHz CPUs with 144 GB of RAM. For a given solenoidal velocity, primal mesh element and N_{no}, the average time to reconstruct a flux polyhedron is computed by summing the times to reconstruct all flux polyhedra in the domain at every time step and dividing it by the total number of reconstructed flux polyhedra in the domain at every time step. The ratio of the average flux polyhedron reconstruction time over the average STFPA-an reconstruction time was found no to depend on N_{no}, so this ratio is averaged over all N_{no} to provide a single value as an assessment of the computational cost for a given flux polyhedron advection scheme for a given primal mesh element and prescribed solenoidal velocity field.

3.1. *Two-dimensional deformation*

The first verification test is the stretching and unstretching of a disk in a vortex (LeVeque 1996).The initial 0.3 diameter disk is centered at $(x, y) = (0, 0.25)$ within the $[-0.5, 0.5]^2$ domain. The disk is deformed in a time-periodic fashion using $u = -\cos^2(\pi x)\sin(2\pi y)\cos(\pi t/T)$ and $v = +\cos^2(\pi y)\sin(2\pi x)\cos(\pi t/T)$, where $T = 8$. The half-period (at $t = T/2$) shapes are shown in Figures 6. Figure 6(a) is for the highest-resolution quadrilateral primal mesh ($N_{no} = 1,050,625$) and Figure 6(b) is for the highest-resolution triangular primal mesh ($N_{no} = 517,169$). In comparing Figures 6(a) and 6(b), it is evident that the NIFPA-1 method performs similarly on quadrilateral and triangular primal meshes; however, there is a small detached volume for the triangular primal mesh case, which is thought to be attributed to the difficulties in reconstructing the interface using EHF on non-convex meshes (see Ivey & Moin 2014, 2015), as well as a smaller N_{no}. In Figure 6(b), the liquid in the tail region reaches the resolution limit of the median-dual mesh, and a small drop detaches from the tail; this drop causes the liquid to move away from the true solution, as shown in Figure 6(d). In addition to reconstruction errors, interface-capturing methods that strive to enforce mass conservation tend to produce detached droplets when there is insufficient resolution (Owkes & Desjardins 2014). Figure 7 compares the shape error of the proposed VOF schemes to that of EMFPA-V^{cor}. For both the quadrilateral and triangular primal meshes, NIFPA-1, NIFPA-2, and EMFPA-V^{cor} exhibit second-order accuracy (see Figures 7(a,b), respectively), with EMFPA-V^{cor} being slightly more accurate on the highest-resolution quadrilateral primal meshes. The small



(a) (b)

FIGURE 6: $F = 0.5$ contour for a two-dimensional deformation test case using NIFPA-1 at $t = T/2$ on the highest-resolution quadrilateral (a) and triangular (b) primal meshes.

increase in precision of the EMFPA-V^{cor} scheme is due to its flux polyhedron's accuracy in approximating the true flux volume; however, the NIFPA-1 and NIFPA-2 schemes's flux polyhedra are sufficiently accurate to maintain second-order accuracy and not to greatly degrade precision. As shown in Figure 7(a,b), the convergence rates STFPA and NIFPA-0 schemes decay to zeroth and first order on the highest resolution quadrilateral and triangular primal meshes, respectively. The authors believe that if a higher-resolution triangular primal mesh were tested, the convergence rates of STFPA and NIFPA-0 would decay to zeroth order as well. The STFPA and NIFPA-0 schemes' flux polyhedra approximations of the streak tube lack information of the velocity field, which results in large diffusive errors. STFPA has larger diffusive errors than those of NIFPA-0 because it artificially forces that flux polyhedron to reside in the sub-tetrahedron associated to the flux face (see Eq. (2.17)), which is a small subset of the total available volume to NIFPA-0 (see Eq. (2.5)).Table 1 provides the average time to reconstruct a flux polyhedron over the time to reconstruct the flux polyhedron using STFPA using the analytic relations in Eqs. (2.19) and (2.22), T. For a given flux polyhedron reconstruction, the relative computational cost of performing the flux polyhedron reconstruction is larger for the triangular primal mesh than for the quadrilateral primal mesh. The non-convex polyhedral decomposition of the quadrilateral primal mesh is composed of fewer constituent convex polyhedra than for triangular primal mesh. Because non-convex polyhedron plane truncation and intersection operations scale with the number of constituent convex polyhedra, the relative computational costs on each primal mesh are as expected. On the quadrilateral primal mesh, $T_{\text{STFPA-it}} < T_{\text{NIFPA-0}} < T_{\text{NIFPA-1}} < T_{\text{NIFPA-2}} < T_{\text{EMFPA-}V^{\text{cor}}}$. Furthermore, on the triangular primal mesh, $T_{\text{STFPA-it}} < T_{\text{NIFPA-0}} < T_{\text{EMFPA-}V^{\text{cor}}} < T_{\text{NIFPA-1}} < T_{\text{NIFPA-2}}$. The EMFPA-$V^{\text{cor}}$ scheme is slower than NIFPA-1 and NIFPA-2 on quadrilateral meshes, but it is faster than those on triangular meshes; however, the computational costs are close to those of the STFPA scheme.

3.2. *Three-dimensional deformation*

The second verification test case is a three-dimensional version of the two-dimensional test case (LeVeque 1996). The 0.3 diameter drop is centered at $(x, y, z) = (0.35, 0.35, 0.35)$ within the $[0, 1]^3$ domain. The droplet is stretched and unstretched using $u = +2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos(\pi t/T)$, $v = -2\sin^2(\pi y)\sin(2\pi x)\sin(2\pi z)\cos(\pi t/T)$, and $w = -2\sin^2(\pi z)\sin(2\pi x)\sin(2\pi y)\cos(\pi t/T)$, where $T = 3$. For the NIFPA-1 scheme, the half-period shapes are shown in Figure 8. Figures 8(a) is for the highest-resolution hex-

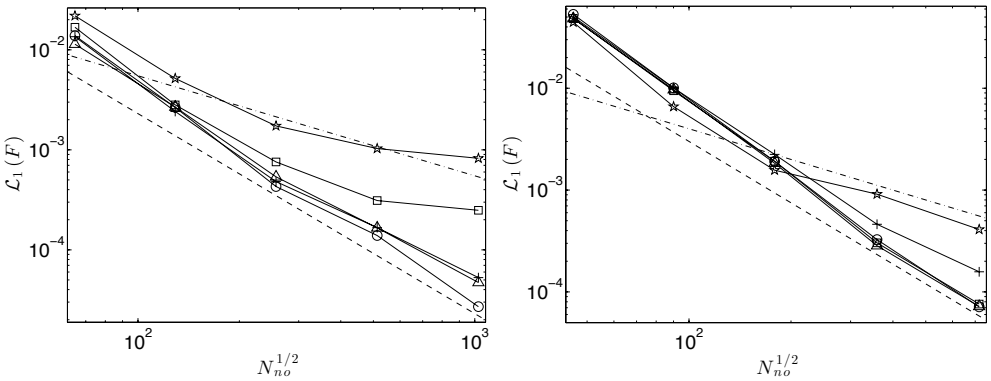
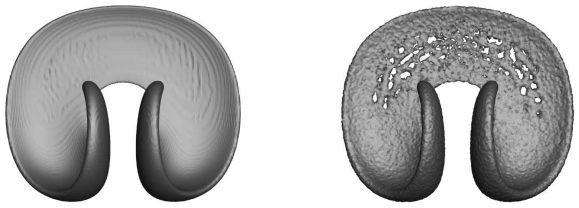


FIGURE 7: Shape error for a two-dimensional deformation test case using EMFPA-V^{cor} and the proposed NIFPA nominal flux polyhedron reconstructions on quadrilateral (a) and triangular (b) primal meshes. The STFPA, NIFPA-0, NIFPA-1, NIFPA-2 and EMFPA-V^{cor} data points are indicated by pentagrams, squares, triangles, plus signs, and circles, respectively.

	STFPA-it	NIFPA-0	NIFPA-1	NIFPA-2	EMFPA-V^{cor}
Quadrilateral	10.17	632.0	1,301	1,335	1,535
Triangle	11.90	2,007	2,572	2,666	2,253

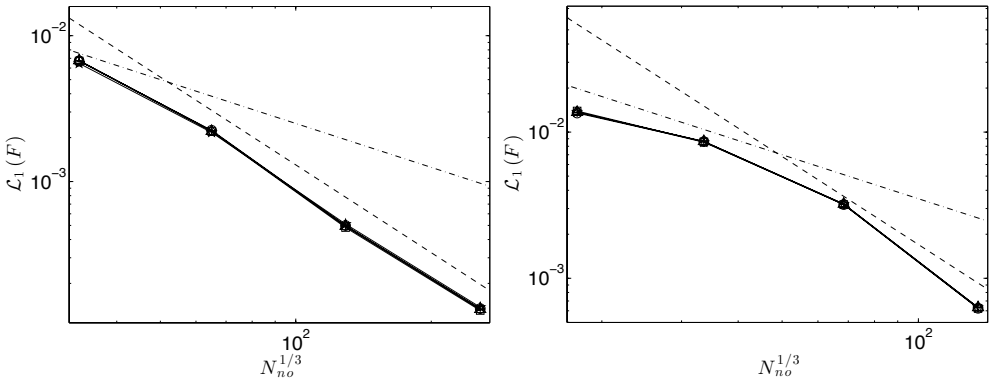
TABLE 1: Average relative time to reconstruct a flux polyhedron for a two-dimensional deformation test case.

ahedral primal mesh ($N_{no} = 16,974,593$) and Figure 8(b) is for the highest-resolution tetrahedral primal mesh ($N_{no} = 2,483,359$). Comparing Figure 8(a) with 8(b), it is evident that the NIFPA-1 performs better on the hexahedral primal mesh; for the tetrahedral primal mesh, there are voids in the stretched sheet due to the difficulties in reconstructing the interface on non-convex polyhedral meshes and a much lower N_{no} (see Ivey & Moin 2014, 2015). Figure 9 compares the posited VOF advection schemes $\mathcal{L}_1(F)$. All tested VOF schemes approach second-order convergence at higher resolutions on the hexahedral and tetrahedral primal meshes. Furthermore, the accuracy of each scheme is approximately equal. The lower convergence rate at coarse resolutions on the tetrahedral primal meshes is due to the lower N_{no} as compared to the hexahedral primal mesh, degrading the convergence in the computed $\hat{n}$ of the PLIC interface (see Ivey & Moin 2014, 2015)). Unlike the results of the two-dimensional test case (see Figure 7), there is no discernible difference between the shape error of the tested VOF advection schemes. At the tested resolutions, the PLIC reconstruction error dominates, making it difficult to compare the accuracies of the VOF advection schemes. From the two-dimensional results, the authors expect that at a higher untested N_{no}, the STFPA and NIFPA-0 schemes' convergence rates will deviate from those of NIFPA-1, NIFPA-2, and EMFPA-V^{cor}, and will eventually decrease to zeroth order. Table 2 assesses the relative computational cost of



(a) (b)

FIGURE 8: $F = 0.5$ contour for a three-dimensional deformation test case using NIFPA-1 at $t = T/2$ on highest-resolution hexahedral (a) and tetrahedral (b) primal meshes.



(a) (b)

FIGURE 9: Shape error for a three-dimensional deformation test case using using EMFPA-V^{cor} and the proposed NIFPA nominal flux polyhedron reconstructions on hexahedral (a) and tetrahedral (b) primal meshes. The STFPA, NIFPA-0, NIFPA-1, NIFPA-2 and EMFPA-V^{cor} data points are indicated by pentagrams, squares, triangles, plus signs, and circles, respectively.

each VOF advection scheme; it provides the average time to reconstruct a flux polyhedron over the time to reconstruct the flux polyhedron using STFPA using the analytic relations in Eqs. (2.19) and (2.22), T. For a given flux polyhedron reconstruction, the relative computational cost of performing the flux polyhedron reconstruction for the tetrahedral primal mesh is larger than for the hexahedral primal mesh. The non-convex polyhedral decomposition of the hexahedral primal mesh is composed of fewer constituent convex polyhedra than for tetrahedral primal mesh. Because non-convex polyhedron plane truncation and intersection operations scale with the number of constituent convex polyhedra, the relative computational costs on each mesh are as expected. On the hexahedral primal mesh, $T_{\text{STFPA-it}} < T_{\text{NIFPA-0}} < T_{\text{EMFPA-}V^{\text{cor}}} < T_{\text{NIFPA-1}} < T_{\text{NIFPA-2}}$. On the tetrahedral primal mesh, $T_{\text{STFPA-it}} < T_{\text{EMFPA-}V^{\text{cor}}} < T_{\text{NIFPA-0}} < T_{\text{NIFPA-1}} < T_{\text{NIFPA-2}}$. The computational costs of performing NIFPA on three-dimensional primal meshes, as compared to STFPA-an and STFPA-it, are greater owing to the greater number constituent convex polyhedra in the nodal volumes and the greater number of intersections between adjacent flux polyhedra. The NIFPA-1 and NIFPA-2 schemes were two to three times more costly than the EMFPA-V^{cor} scheme for all three-dimensional primal meshes.

	STFPA-it	NIFPA-0	NIFPA-1	NIFPA-2	EMFPA-V^{cor}
Hexahedron	10.37	1,792	4,099	4,159	2,006
Tetrahedron	12.89	11,670	12,220	13,280	4,456

TABLE 2: Average relative time to reconstruct a flux polyhedron for a three-dimensional deformation test case.

4. Conclusions

This brief introduced a new three-dimensional unsplit PLIC-VOF advection method, called the non-intersecting flux polyhedron advection (NIFPA) scheme, which is guaranteed to discretely conserve and bound the liquid volume fraction. The NIFPA scheme ensures conservation and boundedness through an iterative algorithm, where each iteration intersects the candidate nominal flux polyhedron with the volume guaranteed to be available to it to yield a candidate flux polyhedron. The accuracy and computational cost of the scheme depend on the choice in nominal flux polyhedron. Unlike contemporary unsplit VOF schemes, the proposed NIFPA nominal flux polyhedron topologies are also guaranteed to be robust and conservative. Three nominal flux polyhedron definitions are proposed; their definitions are described by an extrusion direction and sub-edge vertex-dependent velocity projected onto the extrusion direction. Time-step restrictions are derived for two of the nominal flux polyhedron definitions. Further, a special NIFPA scheme is proposed, the sub-tetrahedron flux polyhedron advection (STFPA) scheme, which constrains the flux polyhedron to the sub-tetrahedron of the associated sub-edge; it has a closed-form solution, does not require iterations, and has a well-defined time-step restriction. The NIFPA algorithm is compared in accuracy and computational cost to the adaption of the method of Owkes & Desjardins (2014) to collocated node-centered unstructured meshes, EMFPA-V^{cor}. Tests are performed on canonical two- and three-dimensional prescribed solenoidal velocity fields that are time periodic. For all tests, each of the VOF advection methods was more precise and less costly on cartesian primal meshes than on simplex primal meshes. Two of the NIFPA definitions were nearly as precise as EMFPA-V^{cor} with second-order accuracy for all tests on all meshes. For the three-dimensional test case, all methods had approximately the same second-order accurate shape error, suggesting that the shape error was dominated by the PLIC reconstruction error (see Ivey & Moin 2014, 2015). Depending on the balance between PLIC reconstruction and advection errors, the STFPA method had accuracies ranging from second to zeroth order, respectively. On two-dimensional tests, the computational cost of NIFPA approximately followed that of EMFPA-V^{cor}, while on three-dimensional meshes, the computational cost was larger. As the STFPA scheme was much faster than all other methods, we suggest using it for cases where PLIC reconstruction errors dominate. However, since we cannot know this *a priori*, we suggest using the NIFPA method that was nearly as accurate as EMFPA-V^{cor} and had a well-defined time-step limit.

Acknowledgments

This investigation was funded by the DOE Computational Science Graduate Fellowship Grant # DE-FG02-97ER25308, the Stanford Graduate Fellowship, and the ONR Grant # 118221.

REFERENCES

Bravo, L., Ivey, C. B., Kim, D. & Bose, S. T. 2014 High-fidelity simulation of atomization in diesel engine sprays. *Proceedings of the Summer Program,* Center for Turbulence Research, Stanford University pp. 89–98.

Brent, R. P. 1973 *Algorithms for Minimization without Derivatives*. Prentice-Hall, Inc., New Jersey.

Comminal, E., Spangenberg, J. & Hattel, J. H. 2015 Cellwise conservative unsplit advection for the volume of fluid method. *J. Comput. Phys.* **283**, 582–608.

Ham, F., Mattson, K. & Iaccarino, G. 2006 Accurate and stable finite volume operators for unstructured flow solvers. *Annual Research Briefs,* Center for Turbulence Research, Stanford University pp. 243–261.

Hernández, J., López, J., Gómez, P., Zanzi, C. & Faura, F. 2008 A new volume of fluid method in three dimensions–Part I: Multidimensional advection method with face-matched flux polyhedra. *Int. J. Numer. Meth. Fl.* **58**, 897–921.

Hirt, C. W. & Nichols, B. D. 1981 Volume of fluid (VOF) method for the dynamics of free boundaries) method for the dynamics of free boundaries. *J. Comput. Phys.* **39**, 201–225.

Ivey, C. B. & Moin, P. 2012 A conservative volume of fluid advection method on unstructured grids in three dimensions. *Annual Research Briefs,* Center for Turbulence Research, Stanford University pp. 179–192.

Ivey, C. B. & Moin, P. 2014 Accurate interface normal and curvature estimates on three-dimensional unstructured non-convex polyhedral meshes. *Annual Research Briefs,* Center for Turbulence Research, Stanford University pp. 53–68.

Ivey, C. B. & Moin, P. 2015 Accurate interface normal and curvature estimates on three-dimensional unstructured non-convex polyhedral meshes. *J. Comput. Phys.* **300**, 365–386.

LeVeque, R. J. 1996 High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.* **33**, 627–665.

Owkes, M. & Desjardins, O. 2014 A computational framework for conservative, three-dimensional, unsplit, geometric transport with application to the volume-of-fluid (VOF) method. *J. Comput. Phys.* **270**, 587–612.

Youngs, D. L. 1982 Time-dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluid Dynamics*. Academic Press, New York.