Accurate interface normal and curvature estimates on three-dimensional unstructured non-convex polyhedral meshes

By **C. B. Ivey** AND **P. Moin**

1. Motivation and objectives

The volume-of-fluid (VOF) method is one of the most widely used formulations to simulate interfacial and free-surface flows (Scardovelli & Zaleski 1999). In this method, the interface evolution is implicitly tracked using a discrete function, F, whose value represents the volume fraction of the tagged fluid within a cell of the computational mesh. F is a discretized version of the fluid marker function, f, that is constant in each phase, jumps at the interface from 0 to 1, and follows the scalar convection equation

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \nabla f = 0, \tag{1.1}$$

where $\vec{v}$ is the velocity vector.

The piecewise-linear interface calculation (PLIC) has become the standard interface representation within the VOF community (Tryggvason *et al.* 2011). PLIC-VOF methods describe the interface by a series of disconnected planes, each oriented by a unit normal, $\hat{n}$, and positioned by a constant, C, such that $\hat{n} \cdot \vec{x} + C = 0$. Two key steps in any PLIC representation include volume truncation, determination of F given $\hat{n}$ and C, and volume enforcement, determination of C given $\hat{n}$ and F. The importance of the volume truncation and enforcement operations has led researchers to develop analytic and geometric tools to expedite computations for rectangular and hexahedral elements (Scardovelli & Zaleski 2000), for triangular and tetrahedral elements (Yang & James 2006), and for convex polyhedral elements (López & Hernández 2008). We extended the class of geometric tools to non-convex polyhedral meshes in order to implement the PLIC-VOF method in a collocated node-based finite-volume flow solver (Ham *et al.* 2006). As evidenced by the volume enforcement and truncation operations, estimation of $\hat{n}$ is key to the accuracy of any PLIC-VOF method.

The equations governing the motion of an unsteady, viscous, incompressible, immiscible two-fluid system are the Navier-Stokes equations, augmented by a localized surface tension force

$$\begin{aligned} \rho\left(\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v}\right) &= -\nabla p + \nabla \cdot \left(\mu\left[\nabla \vec{v} + \{\nabla \vec{v}\}^T\right]\right) + \sigma\kappa\delta\hat{n}, \\ \nabla \cdot \vec{v} &= 0, \\ \kappa &= -\nabla \cdot \hat{n}, \end{aligned} \tag{1.2}$$

where, ρ is the density, p is the pressure, μ is the viscosity, σ is the surface tension coefficient, κ is the interface curvature, and δ is the Dirac Delta function. As evidenced by Eq. (1.2), in addition to accurately estimating $\hat{n}$, the PLIC-VOF framework also needs to calculate the rate at which $\hat{n}$ turns along the interface, κ.

Determination of $\hat{n}$ and κ in the VOF method is problematic due to the discontinuous nature of F. Nevertheless, various algorithms to calculate $\hat{n}$ and κ have been proposed. The traditional Parker-Youngs (PY) method (Parker & Youngs 1992) uses simple difference formulas to calculate gradients in F for the estimation of $\hat{n}$. The method was extended to nonorthogonal meshes (Ito *et al.* 2009); however, PY is at most a first-order accurate method because $\hat{n}$ for a rectilinear interface is not calculated exactly, a necessary condition for second-order accuracy (Pilliod Jr. & Puckett 2004). The least-squares fit procedure (Scardovelli & Zaleski 2003; Aulisa *et al.* 2007) is more accurate than PY and has been extended to unstructured meshes (Ashgriz *et al.* 2004); however, it too does not satisfy the necessary condition for a second-order method. Several second-order methods for estimating $\hat{n}$ have been proposed, namely the least-squares volume-of-fluid interface reconstruction algorithm (LVIRA) and the efficient least-squares volume-of-fluid method interface reconstruction algorithm (ELVIRA) for structured grids (Pilliod Jr. & Puckett 2004), and the geometric least-squares (GLS) method for unstructured grids (Mosso *et al.* 1996), each able to reconstruct a rectilinear interface exactly. LVIRA orients $\hat{n}$ such that the discrepancy in F from using the calculated linear interface over a neighborhood of cells is minimized in the least-squares sense. The procedure requires costly geometric iterations in which the volume enforcement step must be performed. ELVIRA bypasses the iterations by selecting $\hat{n}$ amongst a set of candidate estimates that uses conventional methods. GLS follows the procedure of LVIRA, requiring geometric iterations within an unstructured framework – a prohibitively costly procedure. A well-known non-iterative method for estimating $\hat{n}$ is the height-function (HF) technique. In the HF method, F is integrated in the cartesian direction closest to $\hat{n}$ (approximated with a simpler method) to calculate a height, H. Slopes of a local H distribution in the other two cartesian directions are used to correct $\hat{n}$ (Cummins *et al.* 2005; Sussman 2003; Francois *et al.* 2006). The HF method was shown to be second-order accurate with proper handling for particular alignments of the interface with respect to the grid lines (Bornia *et al.* 2011; Ferdowsi & Bussmann 2008). In two dimensions, the method was extended to nonuniform rectangular grids (Francois & Swartz 2010) and, by adapting the definition of H and using a least-squares fit, to unstructured rectangular/triangular grids (Ito *et al.* 2014), both exhibiting second-order convergence in $\hat{n}$.

As shown in Eq. (1.2), κ requires higher differentiability than that of $\hat{n}$. To combat the lack of differentiability of F, various methods have been posited to calculate κ. In the continuum surface force (CSF) method, F is convoluted with a kernel function to provide a smoothed-out distribution to calculate second derivates over (Brackbill *et al.* 1992). The reconstructed-distance function (RDF) builds a signed-distance function away from the interface to provide a smooth field from which κ can be calculated (Cummins *et al.* 2005). The RDF technique was extended to unstructured rectangular/triangular meshes (Ito *et al.* 2013). Both CSF and RDF have shown lack of convergence under refinement on structured (Cummins *et al.* 2005) and unstructured (Ito *et al.* 2014) meshes. In addition to the calculation of $\hat{n}$, the HF method has been used to calculate κ (Cummins *et al.* 2005), demonstrating second-order accuracy over a series of canonical test problems on uniform cartesian meshes (López *et al.* 2009; López & Hernández 2010; Bornia *et al.* 2011). In two dimensions, calculation of κ with the HF method was extended to nonuniform rectangular grids without loss of the second-order convergence (Francois & Swartz 2010). The HF technique was extended to two-dimensional unstructured rectangular/triangular grids (Ito *et al.* 2014); however, the reframed definition of H required quadric fitting to calculate κ, and the method was less than first-order accurate. To the best of our knowledge, the HF technique for calculating $\hat{n}$ and κ has not been extended to three-dimensional unstructured meshes.

It is hoped that the superiority of the HF technique for estimating $\hat{n}$ and κ on uniform cartesian meshes, in regard to both cost and accuracy, has been made evident. In this paper, we extend the HF technique for estimating $\hat{n}$ and κ to three-dimensional unstructured non-convex polyhedral meshes. The method embeds structured HF stencils in the unstructured mesh and interpolates the mesh F data to the stencils to construct the H distribution; as such, it will be called the embedded height-function (EHF) technique in the rest of the article. The EHF technique differs from the unstructured HF method (Ito *et al.* 2014) in that it uses the traditional H distribution on cartesian stencils to calculate numerical derivates in an attempt to reclaim the second-order accuracy of the structured HF method in the calculation of $\hat{n}$ and κ. The structure of the report is as follows. Section 2 presents the EHF technique, where Section 2.1 reviews the traditional structured HF technique, Section 2.2 discusses the construction of embedded stencils, and Section 2.3 presents two geometric methods for interpolating F from the mesh to the EHF stencils. Section 3 demonstrates the performance of the EHF technique, as compared to the contemporary methods, on a sphere of radius 2 centered in an 8^3 domain.

2. Interface curvature calculation

2.1. *Overview of the height-function technique on a structured grid*

This section provides an overview of the HF method for calculating curvature on uniform cartesian meshes with a focus on 3×3 stencils. Consider an interfacial cell (i,j,k), where $0<F<1$, in which the absolute value of the z component of the $\hat{n}$, n_z, is largest (assume $\hat{n}$ is known and points outward). Using a 3×3 stencil centered at the cell (i,j,k) on the xy plane, a local distribution of H is calculated as follows

$$H_{r,s}=\sum_{t=-t_{down}}^{t_{up}} F^*_{i+r,j+s,k+t}\Delta z,\ \text{ for } r=-1,0-1 \text{ and } s=-1,0,1, \tag{2.1}$$

where t_{down} and t_{up} are adaptively adjusted from 0 to 3, Δz is the cell size in the z direction, and F^* is a modified distribution of F that is forced to follow a local monotonic variation along the z direction. The stencil is adapted as suggested by Hernández *et al.* (2008). t_{up} is gradually increased up to a maximum of 3 cells as long as the following conditions are satisfied

$$\operatorname{sign}(n_z)\left(\alpha_{k+t_{up}}-\alpha_{k+t_{up}-1}\right)<0 \text{ and } 0<\alpha_{k+t_{up}}<9, \tag{2.2}$$

where $\alpha_k=\sum_{r=-1}^{1}\sum_{s=-1}^{1}F_{i+r,j+s,k}$. t_{down} is adjusted similarly. The stencil adaptation guarantees that the entire H distribution is monotonic in the HF direction. F^* satisfies local monotonicity across the HF direction by the following modification proposed by López *et al.* (2009)

$$\begin{aligned}
&F^*_{i+r,j+s,k+t}=\frac{1}{2}\left(1-\operatorname{sign}(t)\operatorname{sign}(n_z)\right),\\
&\quad\text{if } \operatorname{sign}(n_z)\left(F_{i+r,j+s,k+t}-F_{i+r,j+s,k+t-1}\right)<0 \text{ for } t=-1,\ldots,-t_{down}\\
&\quad\text{or } \operatorname{sign}(n_z)\left(F_{i+r,j+s,k+t}-F_{i+r,j+s,k+t-1}\right)>0 \text{ for } t=1,\ldots,t_{up}.\\
&F^*_{i+r,j+s,k+t}=F_{i+r,j+s,k+t} \text{ otherwise.}
\end{aligned} \tag{2.3}$$

From the local H distribution, $\hat{n}$ and κ of the interface at cell (i,j,k) are determined as

$$
\begin{aligned}
\hat{n} &= -\frac{\left\langle H_x \; H_y \; \mathrm{sign}\left(n_z\right)\right\rangle^{\top}}{\sqrt{1+H_x^2+H_y^2}}, \\
\kappa &= -\frac{H_{xx}\left(1+H_y^2\right)+H_{yy}\left(1+H_y^2\right)-2H_{xy}H_xH_y}{\left(1+H_x^2+H_y^2\right)^{3/2}},
\end{aligned}
\tag{2.4}
$$

where H_x and H_y are the partial derivatives of H in the x and y directions, respectively. The $\hat{n}$ and κ of cell (i,j,k) are determined using Eq. (2.4) if the following condition is satisfied: $z_{k-1/2} < H_{0,0} < z_{k+1/2}$, where $z_{k-1/2}$ and $z_{k+1/2}$ are the heights of the lower and upper faces of cell (i,j,k) from the bottom face of the lowest HF stencil cell $(i,j,k-t_{down})$. When this constraint is not satisfied, the interface crosses multiple cells along the middle HF column, degrading the accuracy of this method (López *et al.* 2009; Ferdowsi & Bussmann 2008; Bornia *et al.* 2011). For the interface cells, $0 < F < 1$, that do not satisfy this criterion, $\hat{n}$ and κ are determined through an interpolation of the neighboring interface cells that do satisfy this criterion. If the interface cell is isolated, meaning that there are no neighbors satisfying the HF interface criterion, $\hat{n}$ and κ have to be determined through an auxiliary method that does not involve HF.

The partial derivatives of H over the 3×3 stencil can be calculated, to second-order accuracy, with the smoothed central difference formulas (López *et al.* 2009; López & Hernández 2010) written as

$$
\begin{aligned}
H_x(\gamma) &= \left[\gamma\left(H_{1,1}-H_{-1,1}\right)+\left(H_{1,0}-H_{-1,0}\right)\right. \\
&\quad \left.+\,\gamma\left(H_{1,-1}-H_{-1,-1}\right)\right] / \left(2\Delta x\left(1+2\gamma\right)\right), \\
H_{xx}(\gamma) &= \left[\gamma\left(H_{1,1}-2H_{0,1}+H_{-1,1}\right)+\left(H_{1,0}-2H_{0,0}+H_{-1,0}\right)\right. \\
&\quad \left.+\,\gamma\left(H_{1,-1}-2H_{0,-1}+H_{-1,-1}\right)\right] / \left(\Delta x^2\left(1+2\gamma\right)\right), \\
H_{xy} &= \left[\left(H_{1,1}-H_{1,-1}\right)-\left(H_{-1,1}-H_{-1,-1}\right)\right] / \left(4\Delta x\Delta y\right),
\end{aligned}
\tag{2.5}
$$

where γ is a smoothing coefficient, and Δx and Δy are the grid spacings in the x and y directions, respectively. The partial derivatives in the y direction are calculated similarly. As shown in Eq. (2.5), the partial derivatives in a direction are potentially smoothed along the other direction in the plane. γ is prescribed to improve the $\hat{n}$ and κ computation by averaging out the errors in H (see discussion in Section 3.2). Local osculating spheres have also been implemented to further correct H (López & Hernández 2010); however, this correction was not invoked here to agglomerate all secondary corrections into γ, simplifying the analysis.

2.2. *Construction of height-function stencils on an unstructured grid*

The HF definition proposed in Section 2.1 has been shown to be second-order convergent in $\hat{n}$ and κ numerically (Cummins *et al.* 2005; Ferdowsi & Bussmann 2008; López *et al.* 2009). The second-order accuracy was corroborated analytically provided that the function describing the interface could be approximated to second order with H (Bornia *et al.* 2011). The constraint requires that the fourth derivative of the function is bounded and, for the mean value theorem to hold, that the interface crosses the HF column. To reconstruct H from the F data, the mean value theorem requires that the series of cells in the HF column are bounded by a full cell, $F = 1$, and an empty cell, $F = 0$. The algorithm described in Section 2.1 was designed to meet such constraints, and to leverage such a method, we propose embedding local uniform cartesian stencils in the unstructured mesh.

Our collocated node-based finite-volume flow solver (Ham *et al.* 2006) creates node-based volumes by tessellating each cell of the primal mesh into sub-tetrahedra, each composed of a node, a face center, an edge center, and a cell center, and agglomerating the sub-tetrahedra associated with the node together to form, in general, a non-convex polyhedron. The node volumes comprise the median-dual mesh. An example of a two-dimensional median-dual mesh, composed of non-convex polygons, formed from a primal mesh composed of convex polygons, is shown in Figure 1(a).

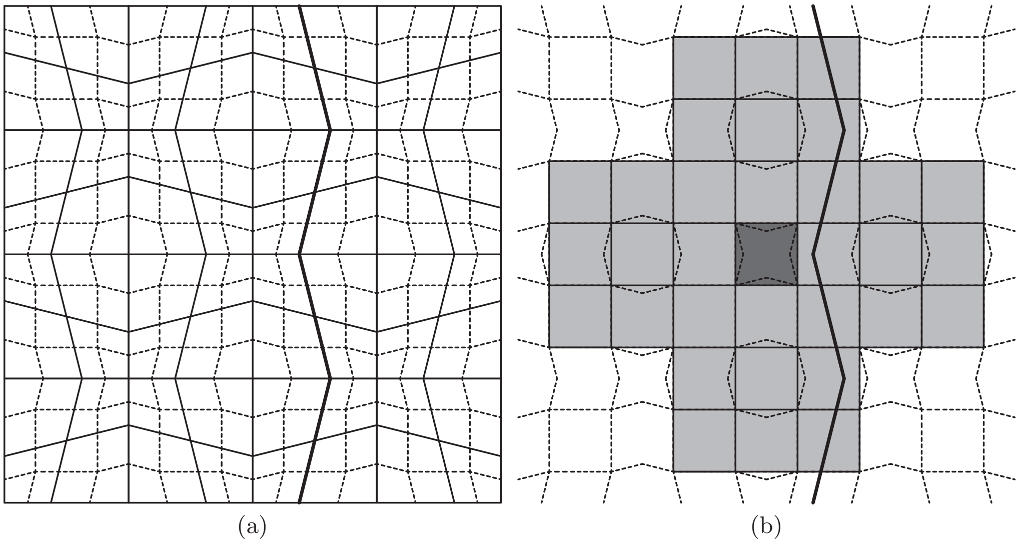
For each node, the HF stencil is constructed by forming the bounding box of the node volume and shifting it in each coordinate direction in increments of its respective length to fill a $(3 \times 7 \times 7) \cup (7 \times 3 \times 7) \cup (7 \times 7 \times 3)$ stencil. To construct the mapping from the mesh to the stencils at the beginning of the simulation, each stencil encompasses the x, y, and z stencils of maximum column height ($t_{up} = t_{down} = 3$), ensuring the availability of any possible subset of the stencil required by Eq. (2.1) to construct the local H distribution. Each node has its own HF stencil to ensure that the $\hat{n}$ and κ estimates for the center bounding box of the HF stencil directly correspond to the node, i.e., if H in the central column resides in the node bounding box, it also resides in the node volume. Under this construction, by performing the algorithm in Section 2.1 on the stencil, the mean value theorem will hold for the node volume. Unfortunately, a single auxiliary uniform cartesian mesh, similar to that used by the balanced-force refined level-set grid method (Herrmann 2008), could not satisfy the constraint placed by the mean value theorem. If stored properly, each HF stencil needs only 135 units in three dimensions (33 in two dimensions), where a given stencil unit, u, can be indexed by the node, p, and the local local cartesian indices, (i, j, k): $u \leftarrow [p, (i, j, k)]$. Each stencil unit needs to store the associated minimum and maximum vertex coordinates of the bounding box, a list of nodes whose volumes intersect with the bounding box, and the intersection volumes of each of these nodes with the bounding box.

In order to efficiently find all the intersections, the alternating digital tree (ADT) search algorithm of Bonet & Peraire (1991) is employed. Bounding boxes of the node volumes are used to build the tree structure. Given the ADT and an HF stencil-unit bounding box, the node volumes whose bounding boxes intersect with the HF stencil-unit bounding box can be efficiently found. For the node-volume bounding boxes that intersect with the HF stencil-unit bounding box, the node volumes are checked for intersection. If the node volume and the HF stencil unit are found to intersect, the node index and intersection volume are added to the list of nodes.

Figure 1(b) shows an example of a two-dimensional stencil created from the bounding box of a node volume of the median-dual mesh constructed in Figure 1(a). In a parallel setting, the embedded stencil can reside on multiple processors, which need not be the same as the unstructured stencils used for differentiating the properties on the median-dual mesh. An example partition is shown to the right of Figures 1(a,b), where the processor boundary separates cells in the primal mesh, cutting through the node volumes.

2.3. *Interpolation of volume fraction data from unstructured grid to stencils*

To perform the calculation in Section 2.1, the HF stencils need to be populated with F data interpolated from the median-dual mesh. As described in Section 2.2, each stencil unit has an associated bounding box and a list of nodes that intersect the bounding box with their respective intersection volumes. From the information available, two simple geometric interpolation schemes can be constructed.



(a) (b)

FIGURE 1. HF stencil embedded in median-dual mesh. The primal mesh is outlined with a solid black line in (a) and the associated median-dual mesh is outlined with dashed black line in (a,b). The processor boundary is indicated by a thick black line in (a,b). The central bounding box of the node volume of interest is indicated by a closed solid black line filled with dark gray, with the associated HF stencil indicated by a closed solid black line filled with light gray in (b).

The simplest method to interpolate F to a single HF stencil unit is to take a weighted average of the F data from the nodes it intersects, where the weight is determined by the intersection volume of the node. The intersection volume-weighted (IVW) interpolation follows the update rule

$$F_u = \frac{\sum_{p\ \epsilon\ \text{intersecting nodes}} V^{\cap}_{u,p} F_p}{V_u}, \tag{2.6}$$

where F_u is the volume fraction of the stencil unit, F_p is the volume fraction of a node, V_u is the stencil-unit bounding box volume, and $V^{\cap}_{u,p}$ is the intersection volume of p's node volume and u's stencil-unit bounding box. Figure 2 illustrates the interpolation scheme for a single stencil unit. IVW provides a first-order approximation to F_u; however, except for F_p, all the required information needs to be calculated once only at the start of the calculation, making it quite efficient over the course of the simulation (note that V_u is the same for all units in a given HF stencil). Once F_u is determined for every HF stencil unit, $\hat{n}$ and κ can be calculated using Eqs. (2.1)-(2.5).

A higher-order interpolation scheme for F_u can be constructed by utilizing the PLIC information from the list of intersected nodes. To interpolate using the PLIC reconstruction, $\hat{n}$ needs to be predetermined: e.g., the PY method. Then the planar interface constant, C, needs to be determined through the volume-enforcement operation using $\hat{n}$ and F. From the PLIC information a more accurate approximation to F_u can be found,

$$F_u = \frac{\sum_{p\ \epsilon\ \text{intersecting nodes}} V\left(\mathcal{H}^{\text{PLIC}}_p \cap \mathcal{P}_p \cap \mathcal{P}_u\right)}{V_u}, \tag{2.7}$$

where $V(\cdot)$ is the volume operator, $\mathcal{H}^{\text{PLIC}}_p$ is the half-space formed behind the PLIC plane (in the $-\hat{n}$ direction), $\mathcal{P}_p$ is the polyhedral representation of the node volume, and $\mathcal{P}_u$ is the polyhedral representation of the bounding box. $\mathcal{H}^{\text{PLIC}}_p \cap \mathcal{P}_p$ is the polyhedron

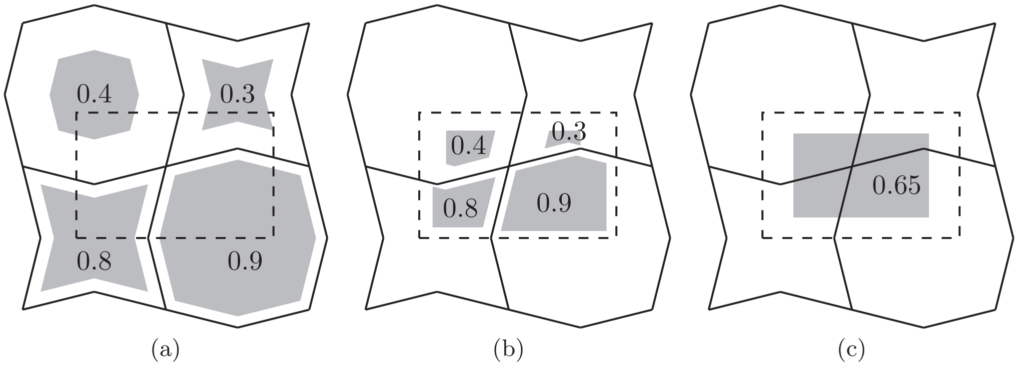


FIGURE 2. IVW interpolation of F_u of the stencil-unit bounding box u from the F_p of the intersected node volumes p. The median-dual mesh is outlined by a solid black line in (a,b,c). The stencil-unit bounding box is outlined by a dashed black line in (a,b,c). The volume-fraction is visualized using a light gray filling in (a,b,c).

formed by truncating $\mathcal{P}_p$ by the PLIC plane. PLIC, along with an accurate estimate of $\hat{n}$, provides a second-order approximation to the fluid interface (Ferdowsi & Bussmann 2008), yielding a second-order accurate F_u. The added accuracy is at the cost of an added volume-enforcement step to reconstruct the PLIC interface. Furthermore, interpolation requires a polyhedron-plane truncation step to find the polyhedron bounding the tagged fluid and a polyhedron-polyhedron intersection step to determine the intersection volume between the bounding box and the tagged fluid. Figure 3 schematically shows the PLIC-based interpolation procedure in two dimensions. Once F_u is determined for the entire HF stencil, $\hat{n}$ can be updated using Eqs. (2.1)-(2.5) and the approximation to F_u can be ameliorated by iterating. IVW, along with Eqs. (2.1)-(2.5), can be used instead of PY to get a first estimate of $\hat{n}$ to perform the PLIC-based interpolation scheme; however, both PY and IVW will provide only a first-order approximation to F_u, so at least two iterations will be required to get a second-order accurate estimation to $\hat{n}$ and κ. The accuracy of the interpolation derives from the accuracy of $\hat{n}$ used in the PLIC reconstruction, so different methods to estimate $\hat{n}$ and iteration counts are surveyed in Section 3.1.

3. Surface normal and curvature tests

Three primal mesh elements are used to test the EHF method, a hexahedron, a wedge, and a tetrahedron. The wedge and tetrahedron mesh elements' uniform edge lengths, Δ, are prescribed such that their respective cell volumes matched that of the hexahedral mesh of the same cell count. Specifically, $\Delta^3_{\text{wedge}} \approx 4/\sqrt{3}\Delta^3_{\text{hexahedron}}$ and $\Delta^3_{\text{tetrahedron}} \approx 6\sqrt{2}\Delta^3_{\text{hexahedron}}$. Unfortunately, the edge constraints could only be approximately satisfied, so the wedge and tetrahedron elements are only approximately uniform in shape. Furthermore, the uniformity constraint on the edge length inadvertently resulted in fewer cells in the wedge- and tetrahedron-packed primal meshes (see Figure 4). The median-dual meshes will have an even larger disparity in node count. Although the primal mesh elements are simple convex polyhedra, the node volumes of the median-dual mesh can be complicated in structure.

A sphere of radius 2 is used to test the EHF technique. The sphere is centered in a

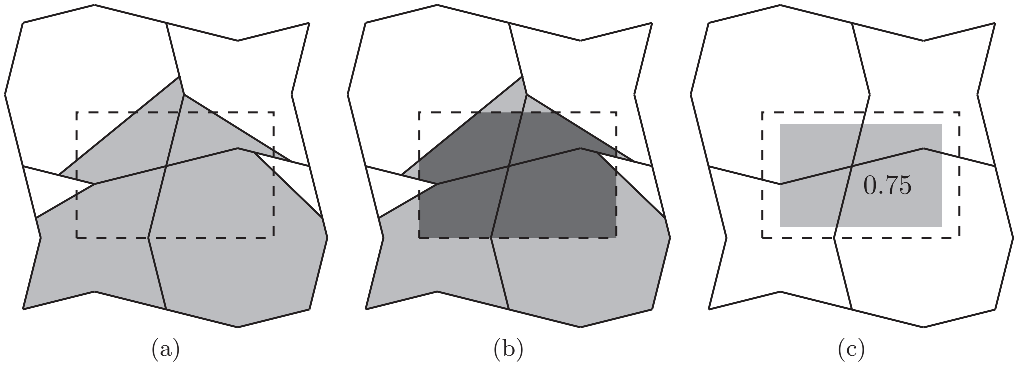


FIGURE 3. PLIC-based interpolation of F_u of the stencil-unit bounding box u from the $\mathcal{H}_p^{\mathrm{PLIC}}$ of the intersected node volumes p. The median-dual mesh is outline by a solid black line in (a,b,c). The stencil-unit bounding box is outlined by a dashed black line in (a,b,c). The PLIC reconstruction is visualized with a closed black line filled with light gray in (a,b). The PLIC region in the stencil-unit bounding box is visualized with a closed black/dashed line filled with dark gray in (b).The volume-fraction is visualized using a light gray filling in (c).

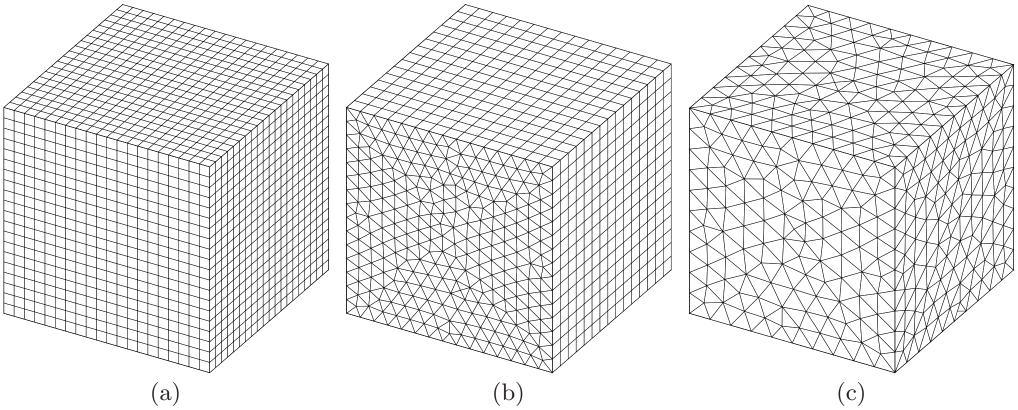


FIGURE 4. Example of primal meshes used in EHF tests of different topology. (a) A mesh composed of 8000 hexahedral elements. (b) A mesh of 7620 wedge elements is shown. (c) A mesh composed of 7302 tetrahedral elements is shown. The wedge and tetrahedron elements in (a) and (b) were created such that their volumes approximately matched a hexahedron element from (a).

domain of size 8^3. The $\hat{n}$ and κ error norms are, respectively, defined as

$$
\begin{aligned}
\mathcal{L}_1(\hat{n}) &= \frac{\sum_{p\ \epsilon\ \text{interface nodes}} \left|1 - \hat{n}_p^* \cdot \hat{n}_p\right| V_p}{\sum_{p\ \epsilon\ \text{interface nodes}} V_p}, \\
\mathcal{L}_1(\kappa) &= \frac{\sum_{p\ \epsilon\ \text{interface nodes}} \left|\left(\kappa_p^* - \kappa_p\right)/\kappa_p^*\right| V_p}{\sum_{p\ \epsilon\ \text{interface nodes}} V_p},
\end{aligned}
\tag{3.1}
$$

where $\hat{n}_p^*$ and $\hat{n}_p$ are, respectively, the true and calculated surface normals at node p, κ_p^* and κ_p are, respectively, the true and calculated surface curvatures at node p, and a node p is considered an interface node if $0 < F_p < 1$.

3.1. *Impact of interpolation scheme*

Section 2.3 describes methods for interpolating F_p from nodes p that intersect HF stencil unit u to F_u, the stencil-unit volume fraction. The methods are broken down into IVW- and PLIC-based methods. The PLIC-based methods rely on an estimate of $\hat{n}$ to perform the interpolation. Further, $\hat{n}$ can be updated by the EHF method and the interpolation repeated. Methods are compared by checking the convergence of $\mathcal{L}_1(\hat{n})$ and $\mathcal{L}_1(\kappa)$ for a sphere of radius 2 in a domain of size 8^3 over a series of tetrahedral primal meshes. The investigated methods are IVW interpolation, PLIC-based interpolation using the exact $\hat{n}$ for a sphere, PLIC-based interpolation using $\hat{n}$ calculated from the PY method, IVW interpolation followed by PLIC-based interpolation using $\hat{n}$ calculated from the EHF method, PLIC-based interpolation using $\hat{n}$ calculated from the PY method followed by PLIC-based interpolation using $\hat{n}$ calculated from the EHF method, and PLIC-based interpolation using $\hat{n}$ calculated from the PY method followed by 9 iterations of PLIC-based interpolation using $\hat{n}$ calculated from the EHF method. The results of the study are shown in Figure 5. From Figure 5(a), it is evident that the accuracy of $\hat{n}$ calculated using the EHF method is second-order convergent regardless of interpolation method or number of iterations; however, IVW interpolation was the least accurate, whereas PLIC-based interpolation using $\hat{n}$ calculated from the PY method followed by PLIC-based interpolation using $\hat{n}$ calculated from the EHF method was closest in accuracy to the PLIC-based interpolation using exact $\hat{n}$. The differences in the interpolation schemes are more apparent in $\mathcal{L}_1(\kappa)$, shown in Figure 5(b). Both IVW interpolation and PLIC-based interpolation using the PY method begin to diverge for large N_{no}. The methods using at least two iterations continue to converge; however, PLIC-based interpolation for the first iteration was more accurate than IVW interpolation. Following the first iteration of PLIC-based interpolation using the PY method, $\mathcal{L}_1(\kappa)$ is insensitive to the number of subsequent iterations of PLIC-based interpolation using $\hat{n}$ calculated from the EHF method, showing greater than first-order convergence and a nominal error that is slightly larger than interpolation using exact $\hat{n}$. As each iteration requires a volume enforcement operation, and in light of the insensitivity of $\mathcal{L}_1(\kappa)$ to the number of iterations of PLIC-based interpolation after two, we implemented PLIC-based interpolation using $\hat{n}$ calculated from the PY method followed by PLIC-based interpolation using $\hat{n}$ calculated from the EHF method for the remainder of the tests.

3.2. *Impact of smoothing parameter*

The calculations of $\hat{n}$ and κ using Eq. (2.4) have an implicit dependence on γ from the smoothed finite difference formulas in Eq. (2.5). The parameter was proposed as a means to increase the accuracy of κ by addressing H errors due to misalignment of the HF direction and $\hat{n}$ (López *et al.* 2009). γ was prescribed as

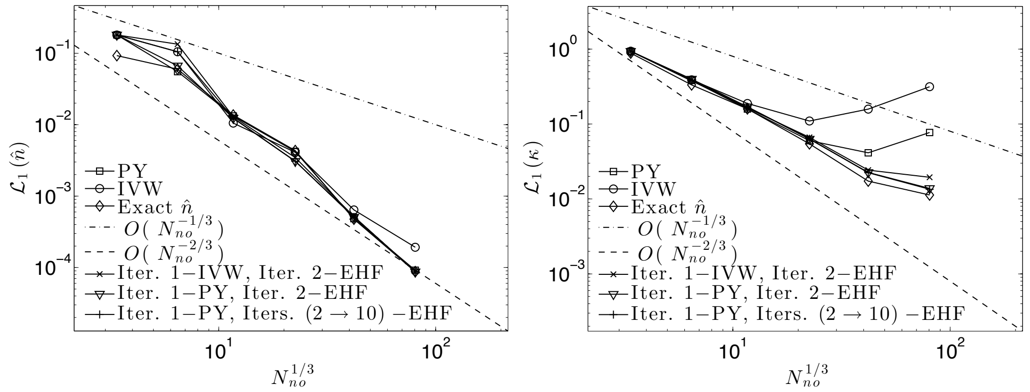
$$\gamma = \begin{cases} 0.2 & \text{if } \theta \geq \theta_{crit} \\ 0 & \text{otherwise} \end{cases}, \tag{3.2}$$

where $\theta = \text{acos}(n_z)$ (z is assumed to be the HF direction), and $\theta_{crit} = 0.8$.

Errors in H, e, are amplified by numerical differentiation. Specifically, the errors in H_x and H_{xx} using second-order finite differences are

$$\begin{aligned} E(H_x) &= (e_1 - e_{-1}) \,/\, (2\Delta x) + O(\Delta x^2) \approx e/\Delta x + O(\Delta x^2), \\ E(H_{xx}) &= (e_1 - 2e_0 + e_{-1}) \,/ \Delta x^2 + O(\Delta x^2) \approx 4e/\Delta x^2 + O(\Delta x^2), \end{aligned} \tag{3.3}$$

so for small Δx, errors in H_x and H_{xx} are dominated by errors in H. From Eqs. (2.4)

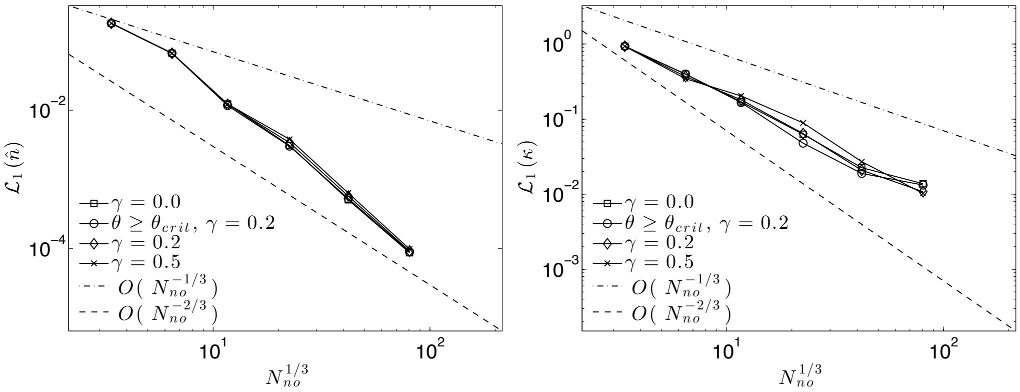


(a) (b)

FIGURE 5. $\mathcal{L}_1$ Errors in $\hat{n}$ (a) and κ (b) for a sphere of radius 2 in a domain of size 8^3 using different interpolation schemes on a tetrahedral primal mesh. IVW indicates IVW interpolation. Exact $\hat{n}$ indicates PLIC-based interpolation using the exact $\hat{n}$ for the sphere. PY indicates PLIC-based interpolation using $\hat{n}$ calculated from the PY method. Iter. 1−IVW, Iter. 2−EHF indicates IVW interpolation followed by PLIC-based interpolation using $\hat{n}$ calculated from the EHF technique. Iter. 1−PY, Iter. 2−EHF indicates PLIC-based interpolation using $\hat{n}$ calculated from the PY method followed by PLIC-based interpolation using $\hat{n}$ calculated from the EHF technique. Iter. 1−PY, Iters. (2 → 10)−EHF indicates PLIC-based interpolation using $\hat{n}$ calculated from the PY method followed by 9 iterations of PLIC-based interpolation.

and (2.5), e is approximately amplified by $1/\Delta x$ in the calculation of $\hat{n}$ and is approximately amplified by $1/\Delta x^2$ in the calculation of κ. The smoothing procedure attempts to minimize the impact of e on $\hat{n}$ and κ. For uniform cartesian meshes, e is mostly due to misalignment of the HF direction from $\hat{n}$, so smoothing based on θ is optimal. In EHF, e is induced from misalignment of the HF direction with $\hat{n}$ and from the interpolation procedure. To address this added source of error, we compared the accuracy of various values of γ to that of Eq. (3.2).

The results of the study are shown in Figure 6. From Figure 6(a), it is evident that $\mathcal{L}_1(\hat{n})$ is insensitive to the choice in γ; this insensitivity is due to the smaller amplification of e in $\hat{n}$, $O(1/\Delta x)$. Although it is visually difficult to discern, $\gamma \neq 0$ slightly increases the error in $\hat{n}$, so we chose not to smooth the numerical derivates in the computation of $\hat{n}$ from the EHF method. The $\hat{n}$ from the EHF method is used for PLIC-based interpolation during the second iteration, so errors in $\hat{n}$ impact the accuracy of κ. The results in Figure 6(b) are constructed without smoothing the derivatives in the computation of $\hat{n}$. From Figure 6(b) it is evident that $\mathcal{L}_1(\kappa)$ is sensitive to the choice of γ, due to the larger amplification of e in κ, $O(1/\Delta x^2)$. Smoothing based on Eq. (3.2) reduced $\mathcal{L}_1(\kappa)$ for all N_{no}; however, the impact diminished at larger N_{no}. Smoothing with a constant γ decreased $\mathcal{L}_1(\kappa)$ for the largest N_{no}; however, for $\gamma = 0.5$, $\mathcal{L}_1(\kappa)$ increased for smaller N_{no}. $\mathcal{L}_1(\kappa)$ for $\gamma = 0.2$ approximately followed that for $\gamma = 0.0$, only decreasing the error for the largest N_{no}. Because a constant $\gamma = 0.2$ aided in the convergence of the EHF method for a larger range of N_{no}, we chose to use it as the smoother for the remainder of the tests (with $\gamma = 0.0$ for the calculation of $\hat{n}$). A more intelligent smoothing procedure that uses Eq. (3.2) for smaller N_{no} and a stronger constant γ for larger N_{no} could be used; however, the transition N_{no} is dependent of the primal mesh, so we did not pursue this approach.



(a) (b)

FIGURE 6. $\mathcal{L}_1$ errors in $\hat{n}$ (a) and κ (b) for a sphere of radius 2 in a domain of size 8^3 using different finite-difference smoothers on a tetrahedral primal mesh.

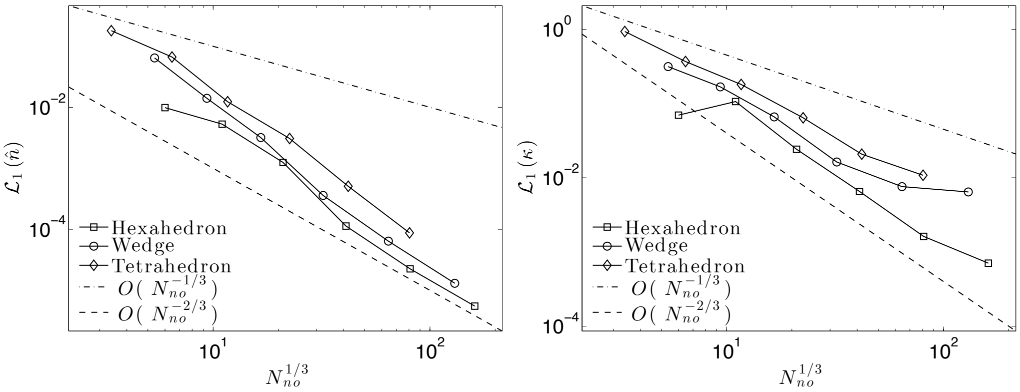
3.3. *Impact of primal mesh*

The EHF technique reduces to standard HF for a uniform hexahedral primal mesh. The median dual of a uniform hexahedral primal mesh is hexahedral (and uniform internally), so an HF stencil-unit bounding box aligns with a single node volume. Because an HF stencil-unit and a node volume are equivalent, the F's match and no interpolation procedure is necessary. We inspect the error induced by embedding stencils and interpolating by comparing $\mathcal{L}_1(\hat{n})$ and $\mathcal{L}_1(\kappa)$ for a wedge and tetrahedral primal mesh to those of the hexahedral primal mesh. The results of the study are shown in Figure 7. As shown in Figure 7(a), $\mathcal{L}_1(\hat{n})$ is at least second-order accurate for all primal meshes; however, there is a loss in precision for the wedge and tetrahedral primal meshes, with the tetrahedral mesh being the least accurate. The convergence behavior of $\mathcal{L}_1(\kappa)$ in Figure 7(b) is impacted more by the primal mesh. $\mathcal{L}_1(\kappa)$ converges at a slightly lower rate for the wedge and tetrahedral primal meshes than for the hexahedral primal mesh, converging at a rate between first and second order. Furthermore, the precision of κ on the wedge primal mesh begins to saturate at large N_{no}, suggesting that the error in κ from truncation begins to balance the errors from e, which are amplified by $O(1/\Delta x^2)$. The increase in $\mathcal{L}_1(\kappa)$ for the hexahedral primal mesh between the two lowest resolutions is attributed to the mesh-based symmetry of a sphere with the HF definition.

In the absence of misalignment of the HF direction with $\hat{n}$, e is a second-order truncation error coming from the PLIC-based interpolation procedure (assuming $\hat{n}$ is estimated accurately enough), $e \approx c_1 \Delta x^2$, so

$$\begin{aligned} E(H_x) &\approx c_1 \Delta x + O(\Delta x^2), \\ E(H_{xx}) &\approx 4c_1 + O(\Delta x^2), \end{aligned} \tag{3.4}$$

where c_1 is small. The actual magnitude of c_1 depends on the PLIC reconstruction, the primal mesh, and the construction of H from Eqs. (2.1)-(2.5), making it difficult to estimate. The important point is that $E(H_x)$ and $E(H_{xx})$ do not grow with N_{no}, so from Eq. (2.4) we know that $\mathcal{L}_1(\hat{n})$ and $\mathcal{L}_1(\kappa)$ monotonically decrease with N_{no}.



(a) (b)

FIGURE 7. $\mathcal{L}_1$ errors in $\hat{n}$ (a) and κ (b) for a sphere of radius 2 in a domain of size 8^3 on different primal meshes.

3.4. *Comparison with other methods*

The calculation of $\hat{n}$ and κ in PLIC-VOF methods on three-dimensional unstructured non-convex polyhedral meshes is underdeveloped. The PY method, because of its simplicity, remains a popular technique to estimate $\hat{n}$, while estimations of κ are traditionally performed with the CSF method or the RDF method. In the latter, a signed-distance field is constructed from the PLIC plane, requiring an estimate of $\hat{n}$ (assumed to come from the PY method here). To have another method to estimate $\hat{n}$, we use gradients of the RDF to update the estimate of $\hat{n}$ calculated from the PY method. The unstructured gradient, divergence, and Laplacian operators of our collocated flow solver (Ham *et al.* 2006) are used for differentiating variables on the median dual. The RDF is generated using the method described by Cummins *et al.* (2005), where the power of the weighting function is taken as 10. Figure 8 plots $\mathcal{L}_1$ errors in $\hat{n}$, computed using the EHF, PY, and RDF methods, and plots $\mathcal{L}_1$ errors in κ, computed using the EHF, CSF, and RDF methods, for hexahedral, wedge, and tetrahedral primal meshes.

Figure 8 plots $\mathcal{L}_1$ errors in $\hat{n}$ and κ for a sphere of radius 2 in a domain of size 8^3. The results for $\hat{n}$ are shown in the left column of the figure. The PY method fails to produce even first-order convergence. The RDF method fails to converge for the hexahedral primal mesh, but provides greater than first-order accurate results for the wedge and tetrahedral primal meshes. The RDF method is shown to be more accurate than PY for all meshes. The EHF method produces normals that are at least second-order accurate. For low resolutions, the precision of the EHF method is lower than that of the PY and RDF methods. The results of the κ calculation are shown in the right column of the figure. The CSF method produced errors that increased with resolution. The RDF method is more accurate than the CSF method; however, at moderate-to-high N_{no}, its numerical errors increased as well. The growth in $\mathcal{L}_1(\kappa)$ for the CSF and the RDF methods has been reported by previous researchers for two-dimensional structured (Cummins *et al.* 2005) and unstructured (Ito *et al.* 2014) meshes. The increase in $\mathcal{L}_1(\kappa)$ for the hexahedral primal mesh between the two lowest resolutions in the EHF technique is attributed to the mesh-based symmetry of a sphere with the HF definition. Ignoring the first point for the hexahedral primal mesh, the EHF method produced monotonically decreasing errors, showing approximately second-order convergence on hexahedral primal meshes

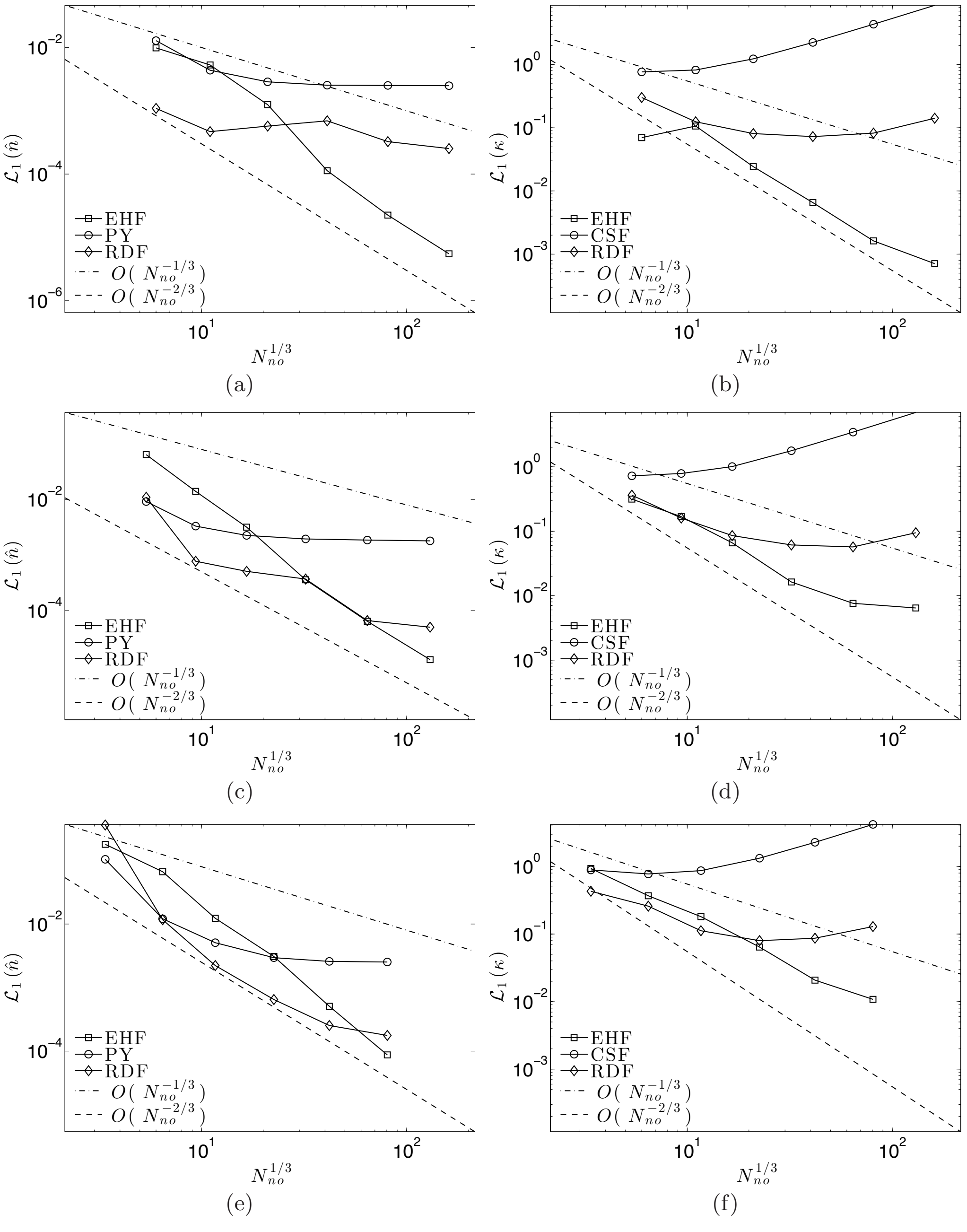


FIGURE 8. $\mathcal{L}_1$ errors in $\hat{n}$ (a,c,e) and κ (b,d,f) for a sphere of radius 2 in a domain of size 8^3 on hexahedral primal meshes (a,b), wedge primal meshes (c,d), and a tetrahedral primal meshes (e,f).

and greater than first-order convergence on tetrahedral primal meshes. For the wedge primal mesh, the precision of the EHF method saturates at the highest resolutions, reducing the convergence rate. Except at low resolution on the tetrahedral mesh, the EHF method produced more precise results than those of the CSF and RDF methods.

While the CSF and RDF techniques employ second-order discretizations, the second derivates of these functions are not bounded, due to an aliasing error, so their associated estimates of κ lose accuracy at higher resolutions (Cummins *et al.* 2005).

4. Conclusions

We developed an accurate method to calculate interface normals and curvatures on a three-dimensional unstructured mesh composed of non-convex polyhedra. The method embeds traditional height-function stencils in the unstructured mesh and geometrically interpolates the volume fraction information from the mesh to the stencil. Two methods for interpolating the volume fraction information were developed, a first-order accurate intersection volume-weighted method and a second-order accurate PLIC-based method.

To perform both PLIC-VOF on non-convex polyhedral meshes and the intersections necessary for interpolation methods, we developed a series of geometric tools that utilize the convex-decomposition description of a non-convex polyhedron.

To validate the embedded height-function method, the interface normal and curvature for a sphere was calculated on the median duals of hexahedral, wedge, and tetrahedral primal meshes. The tests demonstrated that the embedded height-function method calculated interface normals and curvatures with at least second- and first-order accuracy, respectively. At moderate-to-high resolutions, the method was nominally more accurate than contemporary methods. To have an additional method to compare with, we proposed using the reconstructed distance function, traditionally used for curvature, to estimate the normal. This method required an approximation of the normal to reconstruct a planar interface from which the signed distances were constructed. The reconstructed distance function provided a more accurate alternative to the Parker-Young's method.

Note that the convergence of the method in curvature began to saturate on wedge meshes of the highest resolution. The reduction of the convergence rate at high resolutions is due to errors in the computed heights. As a first attempt at reducing the error, we smoothed the finite-difference operators; however, the convergence rate still decreased at the highest resolutions. A more thorough investigation of the errors induced by geometric interpolation of discontinuous data may yield insight in increasing the range over which the method converges. Further, the errors induced by misalignment of the height-function direction with the surface normal could be addressed by this method by embedding a second stencil that is rotated 45° in the xy and xz planes.

Acknowledgments

The authors acknowledge NSF for supporting the Certainty cluster at CTR. CI was supported by the DOE Computational Science Graduate Fellowship.

REFERENCES

Ashgriz, N., Barbat, T. & Wang, G. 2004 A computational Lagrangian–Eulerian advection remap for free surface flows. *Int. J. Numer. Meth. Fl.* **44**, 1–32.

Aulisa, E., Manservisi, S., Scardovelli, R. & Zaleski, S. 2007 Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry. *J. Comput. Phys.* **225**, 2301–2319.

Bonet, J. & Peraire, J. 1991 An alternating digital tree (ADT) algorithm for 3d geometric searching and intersection problems. *Int. J. Numer. Meth. Eng.* **31**, 1–17.

Bornia, G., Cervone, A., Manservisi, S., Scardovelli, R. & Zaleski, S. 2011 On the properties and limitations of the height function method in two-dimensional cartesian geometry. *J. Comput. Phys.* **230**, 851–862.

Brackbill, J. U., Kothe, D. B. & Zemach, C. 1992 A continuum method for modeling surface tension. *J. Comput. Phys.* **100**, 335–354.

Cummins, S. J., Francois, M. M. & Kothe, D. B. 2005 Estimating curvature from volume fractions. *Comput. Struct.* **83**, 425–434.

Ferdowsi, P. A. & Bussmann, M. 2008 Second-order accurate normals from height functions. *J. Comput. Phys.* **227** (22), 9293–9302.

Francois, M. M., Cummins, S. J., Dendy, E. D., Kothe, D. B., Sicilian, J. M. & Williams, M. W. 2006 A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *J. Comput. Phys.* **213**, 141–173.

Francois, M. M. & Swartz, B. K. 2010 Interface curvature via volume fractions, heights, and mean values on nonuniform rectangular grids. *J. Comput. Phys.* **229**, 527–540.

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.

Herrmann, M. 2008 A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *J. Comput. Phys.* **227**, 2674–2706.

Ito, K., Kunugi, T., Ohno, S., Kamide, H. & Ohshima, H. 2014 A high-precision calculation method for interface normal and curvature on an unstructured grid. *J. Comput. Phys.* **273**, 38–53.

Ito, K., Kunugi, T., Ohshima, H. & Kawamura, T. 2013 A volume-conservative PLIC algorithm on three-dimensional fully unstructured meshes. *Comput. Fluids* **88**, 250–261.

Ito, K., Kunugi, T., Oohshima, H. & Kawamura, T. 2009 Formulations and validations of a high-precision volume-of-fluid algorithm on nonorthogonal meshes for numerical simulations of gas entrainment phenomena. *J. Nucl. Sci. Technol.* **46**, 366–373.

López, J. & Hernández, J. 2008 Analytical and geometrical tools for 3d volume of fluid methods in general grids. *J. Comput. Phys.* **227**, 5939–5948.

López, J. & Hernández, J. 2010 On reducing interface curvature computation errors in the height function technique. *J. Comput. Phys.* **229**, 4855–4868.

López, J., Zanzi, C., Gómez, P., Zamora, R., Faura, F. & Hernández, J. 2009 An improved height function technique for computing interface curvature from volume fractions. *Comput. Method. Appl. M.* **198**, 2555–2564.

Mosso, S. J., Schartz, B. K. & Kothe, D. B. 1996 *Recent enhancements of volume tracking algorithms for irregular grids.* Technical Report LA-CP-96-227. Los Alamos National Laboratory.

Parker, B. J. & Youngs, D. L. 1992 *Two and three dimensional Eulerian simulation and fluid flow with material interfaces.* Technical Report 01/92. UK Atomic Weapons Establishment.

Pilliod Jr., J. E. & Puckett, E. G. 2004 Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *J. Comput. Phys.* **199**, 465–502.

Scardovelli, R. & Zaleski, S. 1999 Direct Numerical Simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech.* **31**, 567–603.

Scardovelli, R. & Zaleski, S. 2000 Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *J. Comput. Phys.* **164**, 228–237.

Scardovelli, R. & Zaleski, S. 2003 Interface reconstruction with least-square fit and split Eulerian Lagrangian advection. *Int. J. Numer. Meth. Fl.* **41**, 251–274.

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

Tryggvason, G., Scardovelli, R. & Zaleski, S. 2011 *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge University Press.

Yang, X. & James, A. J. 2006 Analytic relations for reconstructing piecewise linear interfaces in triangular and tetrahedral grids. *J. Comput. Phys.* **214**, 41–54.