Available online at www.sciencedirect.com
SCIENCE DIRECT
Journal of Computational Physics 214 (2006) 41–54
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Analytic relations for reconstructing piecewise linear interfaces in triangular and tetrahedral grids

Xiaofeng Yang, Ashley J. James *

Department of Aerospace Engineering and Mechanics, University of Minnesota, 107 Akerman Hall, 110 Union Street SE, Minneapolis, MN 55455, USA

Received 3 June 2005; received in revised form 30 August 2005; accepted 7 September 2005
Available online 18 October 2005

Abstract

In volume of fluid methods for interfacial flow simulations, one essential process is the so-called interface reconstruction, in which an approximate interface is reconstructed from a given discrete volume fraction field. In [J. Comput. Phys. 164 (2000) 228–237], Scardovelli and Zaleski presented analytical relations connecting linear interfaces and volume fractions in rectangular grids. Here, we present analytical relations connecting linear interfaces and volume fractions in triangular and tetrahedral grids. For computing the volume of fluid in an arbitrary polygonal or polyhedral fluid element, we also cite some of the most efficient formulas for polygon area and polyhedron volume computations. Simple test cases show that this analytic method of interface reconstruction is about 18 times faster than an iterative method in two dimensions, and four to six times faster in three dimensions. The results can be in general applied to other fields as well.

Keywords: VOF; Interface; Interfacial flow; Unstructured grid

1. Introduction

Interfacial flows are very common in many natural and industrial processes, such as emulsification, polymer blending, and so on. In the past two decades, many methods have been developed for numerical simulations of such processes, of which one of the most widely used is the volume of fluid (VOF) method [1,3,4].

In the VOF method, a volume fraction function f is defined. The value of the volume fraction in each grid cell is equal to the ratio of the volume of one of the fluids in this cell, called fluid 1, to the total volume of the grid cell. Thus, f is unity in a cell if the cell lies completely in fluid 1, and is zero if the cell lies completely in the other fluid, called fluid 2. For cells that include an interface, and thus contain both fluid 1 and fluid 2, f is between zero and unity. These cells with $0 < f < 1$ are sometimes called the interfacial cells. As an example, Fig. 1 shows the fluid distribution and the corresponding volume fractions on a triangular grid, where the interface between the two fluids has been approximated by linear segments.

* Corresponding author. Tel.: +1 612 625 6027; fax: +1 612 626 1558.
E-mail address: ajames@aem.umn.edu (A.J. James).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.09.002

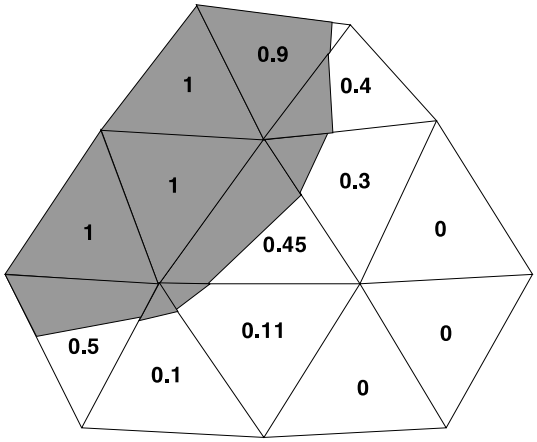

Fig. 1. Fluid distribution and the corresponding volume fractions. The shaded area is occupied by fluid 1. The non-shaded area is occupied by fluid 2. The interface is approximated by linear segments.

Inversely, given the volume fractions in every grid cell, one can reconstruct an interface approximately, which is called "interface reconstruction". Interface reconstruction is essential for a VOF method because, in order to maintain the interface sharp, or prevent excessive numerical diffusion when the interface is being convected, the calculation of the fluid volume fluxes are strongly dependent on the reconstructed interface [1,3,4]. In general, the interface can be approximated by any relatively simple function. For example, in the VOF method of Hirt and Nichols [1], interfaces are effectively approximated using piecewise constant functions which result in a "stair-stepped" interface reconstruction. This is similar to the simple line (piecewise constant) interface calculation (SLIC) of Noh and Woodward [2]. A more modern and currently most widely used method is the piecewise linear interface reconstruction (PLIC) method. The PLIC method is relatively easier to deal with than higher order methods, and it typically produces interfaces of second-order accuracy, if the methods used for volume fraction advection and to compute the interface normal are sufficiently accurate. A PLIC algorithm is at least second order if it reconstructs linear interfaces exactly. A detailed review of some SLIC and PLIC interface reconstruction methods can be found in Rider and Kothe [3].

In a PLIC interface reconstruction method, the interface in a cell is usually approximated as a linear function of the form $\mathbf{n} \cdot \mathbf{x} = \delta$, where $\mathbf{n}$ is the unit normal vector of the interface, which points away from fluid 1, and into fluid 2, $\mathbf{x}$ is the location of a point on the interface and δ is a constant. The unit normal vector $\mathbf{n}$ is often calculated from the gradient of the volume fraction, i.e., $\mathbf{n} = \nabla f/|\nabla f|$, although accurate calculation of ∇f is not trivial because f is in nature discontinuous. A comparison of several normal calculation methods can be found in [3]. The constant δ is determined by enforcing volume conservation, and is usually calculated iteratively. For example, Brent's method is used by Rider and Kothe [3]. During each iterative step, the volume truncated by the linear interface with the most recent estimate of δ is calculated and compared to the given volume of fluid in the cell. A final δ is declared when the discrepancy between these two volumes is within some prescribed tolerance.

In this paper, we are interested in deriving analytic relations connecting linear interfaces (δ) and volume fractions, i.e., given the interface normal vector in a cell, how to find the unique linear segment which also truncates the cell by the given volume fraction. Analytic formulations eliminate the need to iterate, and thus reduce computation time. In addition, the volume of fluid is exactly conserved during interface reconstruction. However, the implementation may not be as straightforward as an iterative method, because of the logic needed to determine which case applies. In [5], Scardovelli and Zaleski presented analytical relations connecting linear interfaces and volume fractions in rectangular grids. In the present paper, we present analytical relations connecting linear interfaces and volume fractions in triangular and tetrahedral grids. The derivations are performed in the context of the VOF method. However, the results can be in general applied to other fields as well. Simple test cases are presented and the CPU time is compared to an iterative method.

2. Analytic PLIC on 2D triangular grids

In 2D, a reconstructed piecewise linear interface is composed of line segments. We represent each line segment by its two end points, which is equivalent to the other forms of line representation, but the former may be more convenient to use than the other forms. Thus, our goal is to find the coordinates of the two end points of each segment.

Consider an arbitrary triangular grid cell of a given volume fraction f and a line segment with a given unit normal vector $\mathbf{n}$. Let us denote the vertices of the triangle by A', B', and C' in an arbitrary order. We can draw a line l' with normal vector $\mathbf{n}$ through vertex A', as illustrated in Fig. 2. Note that in all figures the triangles have been rotated so that the linear segment appears horizontal. The relationship between line l' and the triangle can be divided into three categories: (i) the whole triangle is on one side of the line (see Fig. 2(a)), (ii) the triangle is divided by the line into two parts, one on each side (see Fig. 2(b)), and (iii) the line coincides with one of the edges of the triangle (see Fig. 2(c)). Mathematically, the applicable category can be determined by comparing the signed distances of vertices B' and C' to the line, which are defined as $p_{B'} = \overrightarrow{A'B'} \cdot \mathbf{n}$ and $p_{C'} = \overrightarrow{A'C'} \cdot \mathbf{n}$, where the over-right arrows are used to denote a vector pointing from the first vertex to the second vertex. The distance of vertex A' to the line is, of course, zero, i.e., $p_{A'} = 0$. If $p_{B'}$ and $p_{C'}$ are of the same sign, then B' and C' are on the same side of the line, and category (i) applies. If they are of different signs, then category (ii) applies. If one of them is zero, then category (iii) applies. We assume that the triangle does not degenerate to a straight line, a requirement of grid generation, so $p_{B'}$ and $p_{C'}$ cannot be zero at the same time. One can also transform category (ii) to category (i) simply by exchanging the indices of the three vertices. Category (iii) is in fact a special case of category (i). Transforming different cases to one single case can reduce the number of the cases, and simplify the programming. This is especially helpful in 3D. Here, we show how to perform the transformation, and give the results for category (i).

Let us rename the vertices by A, B, and C such that after renaming the signed distances of the vertices A, B and C to the line l, which is drawn through vertex A and has unit normal vector $\mathbf{n}$ (see Fig. 3), satisfy the following relation:

$$0 = p_A \leqslant p_B \leqslant p_C. \tag{1}$$

Notice that the equalities do not hold simultaneously if we assume that the triangle does not degenerate to a one dimensional line or point. The "renaming" function can be defined as

$$\Pi : \{A', B', C'\} \to \{A, B, C\}$$

by requiring that

$$p_{\Pi^{-1}(A)} \leqslant p_{\Pi^{-1}(B)} \leqslant p_{\Pi^{-1}(C)},$$

where Π^{-1} is the inverse of Π. Thus, after performing the "renaming" process, the new indices A, B, and C satisfy relation (1), and

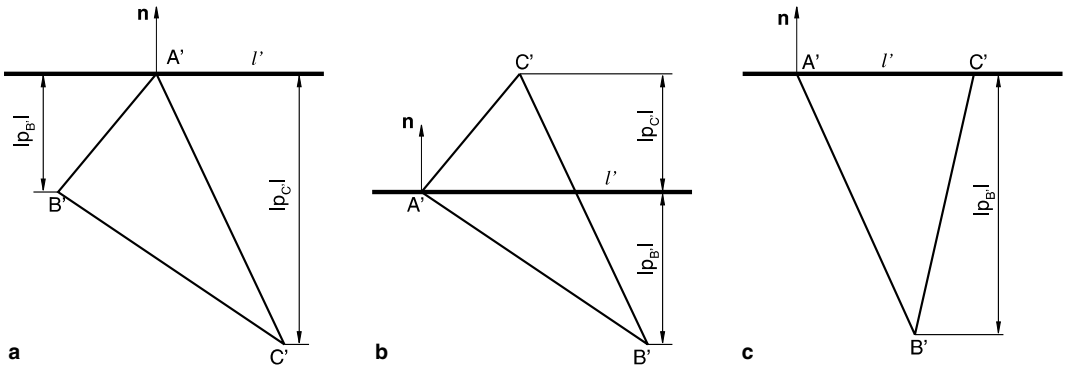

Fig. 2. Drawing a linear segment through a vertex of a triangle. Distances from vertices to the linear segment are denoted. (a) $p_{B'}p_{C'} > 0$; (b) $p_{B'}p_{C'} < 0$; and (c) $p_{B'}p_{C'} = 0$.

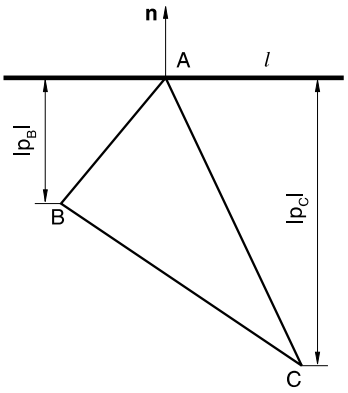

Fig. 3. Indices of the triangle after the transformation.

$$p_A = 0,$$
$$p_B = p_{\Pi^{-1}(B)} - p_{\Pi^{-1}(A)},$$
$$p_C = p_{\Pi^{-1}(C)} - p_{\Pi^{-1}(A)}.$$

With relation (1) satisfied, the intersection of a linear interface with a triangular grid cell can be categorized into two regimes, as illustrated in Fig. 4, where EF is the linear interface segment. Because the normal vector points away from fluid 1 and into fluid 2, the transformation Π ensures that vertex A must be in fluid 1. Let us draw BD with the unit normal vector $\mathbf{n}$, and define the area ratio of the triangle ABD to ABC as $f^* = S_{ABD}/S_{ABC}$. It is easy to show that

$$f^* = \frac{S_{ABD}}{S_{ABC}} = \frac{p_B}{p_C}. \tag{2}$$

Thus, Fig. 4(a) and (b) correspond to $f \leqslant f^*$ and $f > f^*$, respectively.

For $f < f^*$ (see Fig. 4(a)), since ABD and AEF are similar triangles, we have that

$$\frac{|\overrightarrow{AE}|}{|\overrightarrow{AB}|} = \frac{|\overrightarrow{AF}|}{|\overrightarrow{AD}|} = \sqrt{\frac{S_{AEF}}{S_{ABD}}} = \sqrt{\frac{f}{f^*}} \quad \text{and} \quad \overrightarrow{AD} = \frac{p_B}{p_C}\overrightarrow{AC} = f^*\overrightarrow{AC}.$$

Thus, the coordinates of the two end points, E and F, of the interface segment in the triangle ABC are

$$\mathbf{x}_E = \mathbf{x}_A + \sqrt{\frac{f}{f^*}}\overrightarrow{AB} \quad \text{and} \quad \mathbf{x}_F = \mathbf{x}_A + \sqrt{ff^*}\,\overrightarrow{AC}.$$

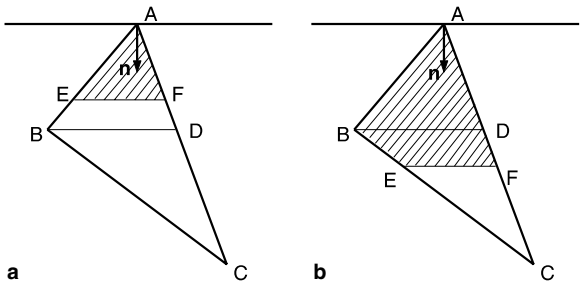

Fig. 4. Fluid configuration for category (i). (a) $f < f^*$ and (b) $f > f^*$.

Similarly, for $f > f^*$ (see Fig. 4(b)), since triangle BCD is similar to ECF, we have that

$$\frac{|\overrightarrow{CE}|}{|\overrightarrow{CB}|} = \frac{|\overrightarrow{CF}|}{|\overrightarrow{CD}|} = \sqrt{\frac{S_{CEF}}{S_{CBD}}} = \sqrt{\frac{1-f}{1-f^*}} \quad \text{and} \quad \overrightarrow{CD} = (1-f^*)\overrightarrow{AC}.$$

Thus, the coordinates of the two end points are

$$\mathbf{x}_E = \mathbf{x}_C + \sqrt{\frac{1-f}{1-f^*}}\overrightarrow{CB} \quad \text{and} \quad \mathbf{x}_F = \mathbf{x}_C + \sqrt{(1-f)(1-f^*)}\,\overrightarrow{CA}.$$

These formulas are also valid for special cases when $p_B = 0$, or $p_B = p_C$. For $p_B = 0$, that is when l coincides with edge AB, we have that $f^* = 0$. Because $1 > f > 0$ in a cell containing an interface, the formulas for $f > f^*$ apply, and are valid. For $p_B = p_C$, that is when l is parallel to edge BC, but not coincide, we have that $f^* = 1$. Thus, the formulas for $f < f^*$ apply, and are valid.

3. Analytic PLIC on 3D tetrahedral grids

In 3D, a reconstructed linear interface in each tetrahedral grid cell is a polygonal segment of a plane, which can be represented by the vertices of the polygon (see Fig. 6).

Consider an arbitrary tetrahedral cell of a given volume fraction f and a linear interface with a given unit normal vector $\mathbf{n}$. The vertices of the tetrahedron are denoted by A', B', C', and D' in an arbitrary order. Let us draw a plane μ' with normal $\mathbf{n}$ through vertex A' as shown in Fig. 5(a). The signed distances of the other vertices B', C', and D' to plane μ' can then be computed as $p_{B'} = \overrightarrow{A'B'} \cdot \mathbf{n}$, $p_{C'} = \overrightarrow{A'C'} \cdot \mathbf{n}$, and $p_{D'} = \overrightarrow{A'D'} \cdot \mathbf{n}$. The distance of the vertex A' to the plane, $p_{A'}$, is of course 0. To simplify the formulation, we want to rename the vertices by A, B, C, and D such that after renaming the signed distances of the vertices A B, C, and D to the plane μ, which is drawn through vertex A and has normal $\mathbf{n}$ (see Fig. 5(b)), satisfy the following relation:

$$0 = p_A \leqslant p_B \leqslant p_C \leqslant p_D. \tag{3}$$

Notice that the equalities do not hold simultaneously if we assume that the tetrahedron does not degenerate to a planar geometry.

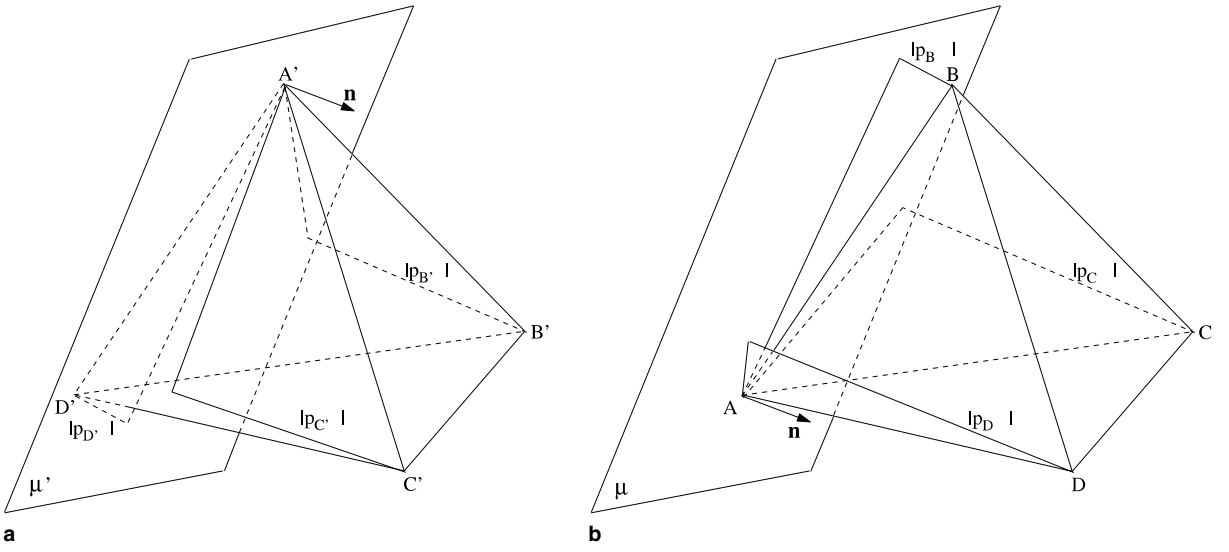

Fig. 5. Exchanging the indices of a tetrahedron. Distances from vertices to the plane are denoted. (a) Before index exchange and (b) after index exchange.

If we define the "renaming" function as

$$\Pi : \{A', B', C', D'\} \to \{A, B, C, D\}$$

by requiring that

$$p_{\Pi^{-1}(A)} \leqslant p_{\Pi^{-1}(B)} \leqslant p_{\Pi^{-1}(C)} \leqslant p_{\Pi^{-1}(D)},$$

where Π^{-1} is the inverse of Π, then, after performing the "renaming" process, the new indices A, B, C, and D satisfy relation (3), and

$$\begin{aligned} p_A &= 0, \\ p_B &= p_{\Pi^{-1}(B)} - p_{\Pi^{-1}(A)}, \\ p_C &= p_{\Pi^{-1}(C)} - p_{\Pi^{-1}(A)}, \\ p_D &= p_{\Pi^{-1}(D)} - p_{\Pi^{-1}(A)}. \end{aligned}$$

As is in the 2D case, this transformation ensures that vertex A is in fluid 1.

With relation (3) satisfied, the intersection of a linear interface with a tetrahedral grid cell can be categorized into three regimes. Let us draw a plane with the given normal vector $\mathbf{n}$ through vertex B, as shown in Fig. 6. The intersections of the plane with AC and AD are F and E, respectively. Then, the volume ratio of the tetrahedron $ABEF$ to $ABCD$ is

$$f_B = \frac{V_{ABEF}}{V_{ABCD}} = \frac{S_{AEF}}{S_{ACD}} = \left(\frac{|\overrightarrow{AF}|}{|\overrightarrow{AC}|}\right)^2 \frac{|\overrightarrow{AL}|}{|\overrightarrow{AD}|} = \left(\frac{p_B}{p_C}\right)^2 \frac{p_C}{p_D}.$$

The second equality holds because $ABEF$ and $ABCD$ have the same height, i.e., the distance of vertex B to AEF or ACD. The third equality holds because $S_{AEF}/S_{ACD} = (S_{AEF}/S_{ALC})(S_{ALC}/S_{ACD})$, and AEF and ALC are similar, and ALC and ACD have the same height, i.e., the distance from C to AD. Similarly, we draw a plane with the given normal vector $\mathbf{n}$ through vertex C. The intersections of the plane with DA and DB are L and K, respectively. Then, the volume ratio of the tetrahedron $DCKL$ to $ABCD$ is

$$f_C = \frac{V_{DCKL}}{V_{ABCD}} = \frac{S_{DKL}}{S_{ABD}} = \left(\frac{|\overrightarrow{DK}|}{|\overrightarrow{DB}|}\right)^2 \frac{|\overrightarrow{DE}|}{|\overrightarrow{AD}|} = \left(\frac{p_D - p_C}{p_D - p_B}\right)^2 \frac{p_D - p_B}{p_D}.$$

If $f \leqslant f_B$, then the linear interface must cut the tetrahedron $ABEF$, as shown in Fig. 6(a). We call this regime I. If $f \geqslant 1 - f_C$, then the linear interface must cut the tetrahedron $DCKL$, as shown in Fig. 6(c). We call this regime III. Otherwise, the linear interface must be between the two planes through B and C, as shown in Fig. 6(b). We call this regime II.

As is in the 2D case, special cases can occur. For example, if $p_B = 0$, then $f_B = 0$ and points E and F collapse to vertex A. The formulas we derive below are valid for these special cases. The only caution is that, to avoid dividing by zero, in practical programming one need explicitly classify the case when $p_A = p_B = p_C = 0$ into regime III, and set $f_C = 1$, and the case when $p_B = p_C = p_D$ into regime II and set $f_B = 1$.

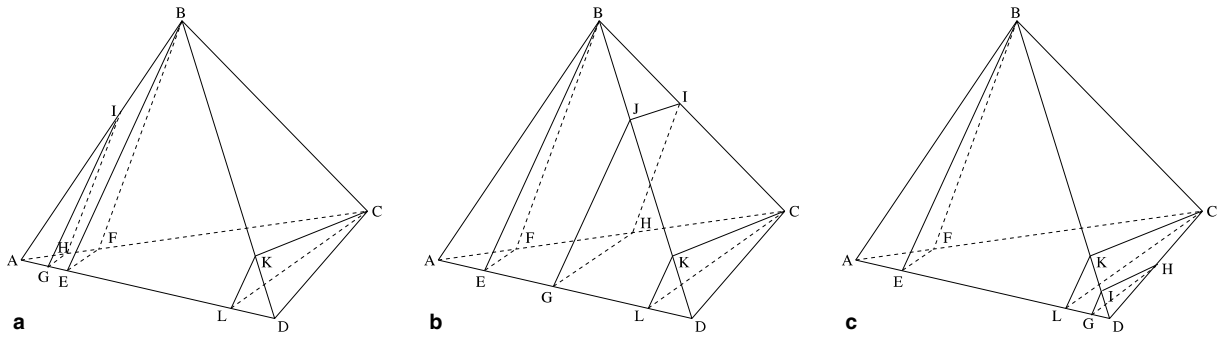

Fig. 6. Intersection of a planar interface with a tetrahedral grid cell. (a) Regime I; (b) regime II; and (c) regime III.

For regime I (see Fig. 6(a)), the interface polygon is a triangle with vertices G, H, and I. We have that

$$\frac{|\overrightarrow{AG}|}{|\overrightarrow{AE}|} = \frac{|\overrightarrow{AH}|}{|\overrightarrow{AF}|} = \frac{|\overrightarrow{AI}|}{|\overrightarrow{AB}|} = \left(\frac{f}{f_B}\right)^{1/3}.$$

Thus, the coordinates of the three vertices of the linear interface piece are:

$$\begin{aligned} \mathbf{x}_G &= \mathbf{x}_A + \left(\frac{f}{f_B}\right)^{1/3}\overrightarrow{AE}, \\ \mathbf{x}_H &= \mathbf{x}_A + \left(\frac{f}{f_B}\right)^{1/3}\overrightarrow{AF}, \\ \mathbf{x}_I &= \mathbf{x}_A + \left(\frac{f}{f_B}\right)^{1/3}\overrightarrow{AB}. \end{aligned} \tag{4}$$

Notice that $\overrightarrow{AE}$ and $\overrightarrow{AF}$ are related to the edges of the original tetrahedra cell by

$$\overrightarrow{AE} = \frac{p_B}{p_D}\overrightarrow{AD} \quad \text{and} \quad \overrightarrow{AF} = \frac{p_B}{p_C}\overrightarrow{AC}.$$

Similarly, for regime III (see Fig. 6(c)), the interface polygon is another triangle with vertices G, H, and I. We have that

$$\frac{|\overrightarrow{DG}|}{|\overrightarrow{DL}|} = \frac{|\overrightarrow{DH}|}{|\overrightarrow{DC}|} = \frac{|\overrightarrow{DI}|}{|\overrightarrow{DK}|} = \left(\frac{1-f}{f_C}\right)^{1/3}.$$

Thus, the coordinates of the three vertices of the linear interface piece are:

$$\begin{aligned} \mathbf{x}_G &= \mathbf{x}_D + \left(\frac{1-f}{f_C}\right)^{1/3}\overrightarrow{DL}, \\ \mathbf{x}_H &= \mathbf{x}_D + \left(\frac{1-f}{f_C}\right)^{1/3}\overrightarrow{DC}, \\ \mathbf{x}_I &= \mathbf{x}_D + \left(\frac{1-f}{f_C}\right)^{1/3}\overrightarrow{DK}, \end{aligned} \tag{5}$$

where

$$\overrightarrow{DL} = \frac{p_D - p_C}{p_D}\overrightarrow{DA} \quad \text{and} \quad \overrightarrow{DK} = \frac{p_D - p_C}{p_D - p_B}\overrightarrow{DB}.$$

For regime II (see Fig. 6(b)), the interface polygon is a quadrilateral with vertices G, H, I and J. We want to enforce

$$\frac{V_{ABGHIJ}}{V_{ABCD}} = f. \tag{6}$$

Let us connect B and G, B and H, C and G, and C and J, as shown in Fig. 7. Thus, the volumes of the two complex polyhedron can be decomposed into the volumes of a tetrahedron and a rectangular pyramid, respectively as the following:

$$V_{ABGHIJ} = V_{BAGH} + V_{BGHIJ}, \tag{7}$$

$$V_{DCGHIJ} = V_{CDGJ} + V_{CDGHIJ}. \tag{8}$$

Notice that the interface $GHIJ$, plane BEF and CKL are parallel to each other. So, we have that

$$\frac{|\overrightarrow{BI}|}{|\overrightarrow{BC}|} = \frac{|\overrightarrow{FH}|}{|\overrightarrow{FC}|} = \frac{|\overrightarrow{EG}|}{|\overrightarrow{EL}|} = \frac{|\overrightarrow{BJ}|}{|\overrightarrow{BK}|} = \alpha, \tag{9}$$

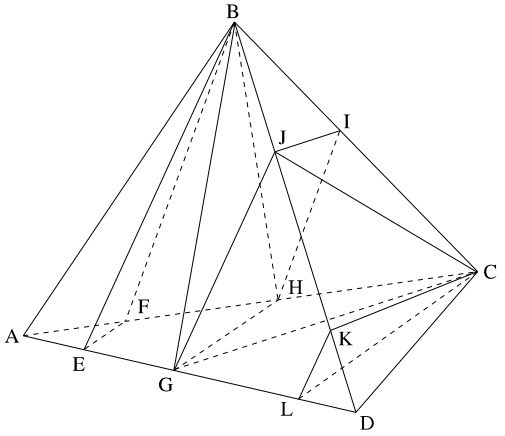

Fig. 7. Decomposition of complex geometries in regime II.

where α is equal to the ratio of the perpendicular distance between plane *BEF* and *GHIJ* to the perpendicular distance between plane *BEF* and *CKL*. Thus, if we find α, we can determine the coordinates of the four vertices of the linear interface from Eq. (9).

To determine α by enforcing Eq. (6), we must find expressions for the ratios of the volumes in Eq. (7) to V_{ABCD}. First, notice that

$$\frac{V_{BAGH}}{V_{ABCD}} = \frac{S_{AGH}}{S_{ACD}} = \left(\frac{|\overrightarrow{AH}|}{|\overrightarrow{AC}|}\right)^2 \frac{|\overrightarrow{AL}|}{|\overrightarrow{AD}|} = \left(\frac{|\overrightarrow{AH}|}{|\overrightarrow{AC}|}\right)^2 \frac{p_C}{p_D}.$$

We decompose $|\overrightarrow{AH}|$ as follows:

$$|\overrightarrow{AH}| = |\overrightarrow{AF}| + |\overrightarrow{FH}| = |\overrightarrow{AF}| + \alpha|\overrightarrow{FC}| = |\overrightarrow{AF}| + \alpha\left(|\overrightarrow{AC}| - |\overrightarrow{AF}|\right),$$

which gives us

$$\frac{|\overrightarrow{AH}|}{|\overrightarrow{AC}|} = \frac{|\overrightarrow{AF}|}{|\overrightarrow{AC}|} + \alpha\left(1 - \frac{|\overrightarrow{AF}|}{|\overrightarrow{AC}|}\right) = \frac{p_B}{p_C} + \alpha\left(1 - \frac{p_B}{p_C}\right).$$

By substitution, we obtain

$$\frac{V_{BAGH}}{V_{ABCD}} = \left(\frac{p_B}{p_C} + \alpha\left(1 - \frac{p_B}{p_C}\right)\right)^2 \frac{p_C}{p_D}. \tag{10}$$

Similarly, we have that

$$\frac{V_{CDGJ}}{V_{ABCD}} = \frac{S_{DGJ}}{S_{ABD}} = \left(\frac{|\overrightarrow{DJ}|}{|\overrightarrow{DB}|}\right)^2 \frac{|\overrightarrow{DE}|}{|\overrightarrow{AD}|} = \left(\frac{|\overrightarrow{DJ}|}{|\overrightarrow{DB}|}\right)^2 \left(1 - \frac{p_{AB}}{p_{AD}}\right),$$

$$|\overrightarrow{DJ}| = |\overrightarrow{DB}| - |\overrightarrow{BJ}| = |\overrightarrow{DB}| - \alpha|\overrightarrow{BK}|,$$

$$\frac{|\overrightarrow{DJ}|}{|\overrightarrow{DB}|} = 1 - \alpha\frac{|\overrightarrow{BK}|}{|\overrightarrow{DB}|} = 1 - \alpha\frac{p_C - p_B}{p_D - p_B}.$$

Then, by substitution, we obtain

$$\frac{V_{CDGJ}}{V_{ABCD}} = \left(1 - \alpha\frac{p_C - p_B}{p_D - p_B}\right)^2 \left(1 - \frac{p_B}{p_D}\right). \tag{11}$$

Now, notice that the two pyramids *BGHIJ* and *CGHIJ* have the same base. So their volume ratio is equal to the ratio of the distance of vertex *B* to *GHIJ* to the distance of vertex *C* to *GHIJ*, i.e.,

$$\frac{V_{BGHIJ}}{V_{CGHIJ}} = \frac{\alpha}{1-\alpha}. \tag{12}$$

In addition, because the decomposed volumes constitute the original tetrahedron *ABCD*, we have that

$$\frac{V_{BAGH}}{V_{ABCD}} + \frac{V_{CDGJ}}{V_{ABCD}} + \frac{V_{BGHIJ}}{V_{ABCD}} + \frac{V_{CGHIJ}}{V_{ABCD}} = 1. \tag{13}$$

Combining Eqs. (10)–(13), we get

$$\frac{V_{BGHIJ}}{V_{ABCD}} = -\alpha^3 \frac{(p_C - p_B)^2}{p_D}\left(\frac{1}{p_C} + \frac{1}{p_D - p_B}\right) + \alpha^2 \frac{2(p_C - p_B)^2}{p_D p_C} + \alpha \frac{p_B(p_C - p_B)}{p_D p_C}. \tag{14}$$

Finally, enforcing the known volume fraction, Eq. (6), in conjuction with Eqs. (7), (10) and (14) gives us an equation for α

$$P(\alpha) = a\alpha^3 + b\alpha^2 + c\alpha + d = 0, \tag{15}$$

where

$$a = -\frac{(p_C - p_B)^2}{p_D}\left(\frac{1}{p_C} + \frac{1}{p_D - p_B}\right),$$
$$b = \frac{3(p_C - p_B)^2}{p_D p_C},$$
$$c = \frac{3p_B(p_C - p_B)}{p_D p_C},$$
$$d = \frac{p_B^2}{p_D p_C} - f$$

are known quantities. Notice that $P(\alpha)$ is the difference between the volume fraction corresponding to the position of *GHIJ* given by α and the given volume fraction f, and thus $P(\alpha)$ should increase as α increases from 0 to 1 or as *GHIJ* moves from *BEF* to *CKL*. On the other hand, because $a < 0$, $P(\mp\infty) = \pm\infty$. Therefore, Eq. (15) must have three real roots, and the root of interest must be the middle one, near which $P(\alpha)$ increases.

Analytic solutions can be found for cubic equations [7]. In our case, the root of interest is

$$\alpha = \sqrt{-\frac{p}{3}}\left(\sqrt{3}\sin\theta - \cos\theta\right) - \frac{b}{3a},$$

where

$$p = \frac{c}{a} - \frac{b^2}{3a^2}, \quad q = \frac{d}{a} + \frac{2b^3}{27a^3} - \frac{bc}{3a^2}, \quad \text{and} \quad \theta = \frac{1}{3}\arccos\left(-\frac{q}{2\sqrt{-(p/3)^3}}\right).$$

It is also possible to find the roots of a cubic equation numerically. Scardovelli and Zaleski [5] showed that directly evaluating this analytic formula is less than two thirds as time consuming as Newton–Raphson iterations.

Now that α is known, the coordinates of the four vertices of the linear interface can then be determined from Eq. (9), i.e.

$$\begin{aligned}
\mathbf{x}_G &= \mathbf{x}_E + \alpha\overrightarrow{EL},\\
\mathbf{x}_H &= \mathbf{x}_C + (1-\alpha)\overrightarrow{CF},\\
\mathbf{x}_I &= \mathbf{x}_B + \alpha\overrightarrow{BC},\\
\mathbf{x}_J &= \mathbf{x}_B + \alpha\overrightarrow{BK},
\end{aligned} \tag{16}$$

where

$$\begin{aligned}
\mathbf{x}_E &= \mathbf{x}_A + \frac{p_B}{p_D}\overrightarrow{AD},\\
\overrightarrow{EL} &= \frac{p_C - p_B}{p_D}\overrightarrow{AD},\\
\overrightarrow{CF} &= \frac{p_C - p_B}{p_C}\overrightarrow{CA},\\
\overrightarrow{BK} &= \frac{p_C - p_B}{p_D - p_B}\overrightarrow{BD}.
\end{aligned}$$

Notice that the above derivations can be inversely used to calculate the volume fraction of a cell when the cell is cut by a given linear interface. For example, when a linear interface intersects a tetrahedron as regime II, then one can easily compute α from the location of one of the verticies of the interfacial polygon, and the cubic polynomial $P(\alpha)$, when evaluating d as $p_B^2/p_D p_C$, gives the volume fraction.

4. Computing the volume of fluid in an arbitrary polygonal or polyhedral fluid element

Another important problem in the VOF method is to compute the volume fluxes of fluid, which in general boils down to computing the area or volume of an arbitrary polygon (in 2D) or polyhedron (in 3D). These polygonal or polyhedral fluid elements are usually obtained by clipping a fluid element against a grid cell (see [3] for example). In this section, we cite some formulas for computing the area/volume of an arbitrary polygon/polyhedron.

Let (x_i, y_i), $i = 0, \ldots, n$, be the coordinates of the vertices of a polygon with $x_n = x_0$ and $y_n = y_0$, then the signed area of the polygon can be computed as the following [8]:

$$S = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i) = \frac{1}{2}\sum_{i=1}^{n} x_i(y_{i+1} - y_{i-1}), \tag{17}$$

where in the second summation it is assumed that $x_{n+1} = x_1$ and $y_{n+1} = y_1$. This signed area is positive when the vertices are ordered counterclockwise, and negative when ordered clockwise. The only restriction on this formula is that the polygon must not be self-intersecting. For a polygon with n vertices, the first summation, which is used in many textbooks, involves $2n$ multiplications and $(2n-1)$ additions, while the second summation due to Sunday [8] only involves n multiplications and $(2n-1)$ additions.

Now, consider a polyhedron with m faces, labeled $F_1, \ldots, F_m$, with their normal vectors pointing away from the polyhedron. Let $\mathbf{P}_j$ be an arbitrary point on face F_j, which is usually chosen to be any vertex of face F_j. Then, the volume of the polyhedron can be computed as [9]

$$V = \frac{1}{6}\sum_{j=1}^{m}\mathbf{P}_j \cdot (2\mathbf{S}_j),$$

where $\mathbf{S}_j$ is the vector area of face F_j. Let $\mathbf{X}_i = (x_i, y_i, z_i)$, $i = 0, \ldots, n$, be the vertices of a planar polygonal face in 3D space, listed in counterclockwise direction when viewed from the half space into which the normal vector points. Then, the vector area of the face can be computed as [9]

$$2\mathbf{S}_j = \sum_{i=1}^{h-1}(\mathbf{X}_{2i} - \mathbf{X}_0) \times (\mathbf{X}_{2i+1} - \mathbf{X}_{2i-1}) + (\mathbf{X}_{2h} - \mathbf{X}_0) \times (\mathbf{X}_l - \mathbf{X}_{2h-1}),$$

where h is the greatest integer less than $(n-1)/2$, and $l = 0$ for n odd, and $l = n-1$ for n even.

The above formulas offer a substantial improvement of the efficiency over Goldman's standard formula [10,9]. Further speed-up can be achieved upon specific implementations (see [9]). Concerning the vector area calculation, one more efficient formula can be obtained by using the strategy due to Sunday [8], in which the planar polygon in 3D is projected onto a 2D plane by simply ignoring one of the three coordinates, the area in the 2D plane can be computed using the previous fastest formula, and the area of the 3D planar polygon can be recovered by scaling the 2D area. This strategy achieves about three times speed-up over the above Gelder's formula [8].

5. Iterative method

In the next section, the direct method will be compared to the results of an interactive method. In this section we briefly describe the iterative method used for this comparison. Instead of computing the line segment end points or polygon verticies, the line constant, δ, is computed to define an interface with the given unit normal.

The method is based on Brent's method, as implemented in [11]. First, for each cell vertex δ is determined for a line in 2D or a plane in 3D, with the given unit normal, that passes through that vertex. The minimum and maximum of these values of δ provide bounds on δ for Brent's method, which begins with these bounds and iterates until the absolute difference between the given volume fraction, and the volume fraction corresponding to the current value of δ is less than some tolerance.

In two dimensions, the line defined by δ and the given unit normal splits the grid cell into two polygons. A clipping routine (see [3] for further description) is used to determine the verticies of the polygon that the normal points away from. The polygon area is then computed using Eq. (17), which is also used to compute the cell area. The volume fraction corresponding to δ is then the polygon area divided by the cell area.

In three dimensions, the volume fraction corresponding to δ is computed using the inverse of the method of Section 3. The verticies are reordered so that relation (3) is satisfied. The signs of $p_{A'}$, $p_{B'}$, $p_{C'}$, and $p_{D'}$ are inspected to determine whether regime I, II or III applies. One of the coordinates of one of the verticies of the interfacial polygon is determined by finding the intersection between the interfacial plane defined by δ and the line connecting two of the cell verticies. This coordinate is used in Eq. (4) for regime I, Eq. (16) for regime II, or Eq. (5) for regime III to determine the volume fraction.

6. Verification

Test cases are performed for the two regimes illustrated in Fig. 4 in two dimensions (referred to below as cases 1 and 2) and for regimes I, II, and III in three dimensions. The results are compared to the iterative method outlined in Section 5.

The 2D test cases are shown in Fig. 8. The line segment is the same in the two cases, but fluid 1 lies on opposite sides of the line between the cases. The analytic method determines the locations of the segment

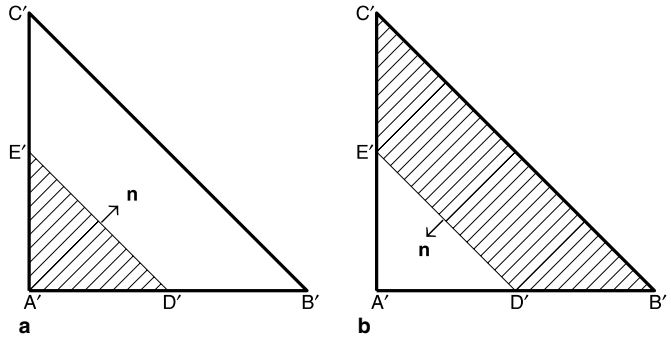

Fig. 8. Two dimensional test cases. The cell verticies are A', B', and C', with coordinates $(0,0)$, $(1,0)$, and $(0,1)$, respectively. The line segment endpoints are D' and E', with coordinates $(1/2,0)$ and $(0,1/2)$, respectively. (a) Case 1: $f<f^*$, the unit normal is $(1/\sqrt{2},1/\sqrt{2})$, $f=1/4$, and $\delta=1/(2\sqrt{2})$. (b) Case 2: $f>f^*$, the unit normal is $(-1/\sqrt{2},-1/\sqrt{2})$, $f=3/4$, and $\delta=-1/(2\sqrt{2})$.

end points, then the line constant is computed as $\delta = \mathbf{n} \cdot \mathbf{x}$ at each end point for comparison to the iterative method, which determines δ directly.

The percent error in δ is presented in Table 1 for the two cases using the analytic and iterative methods. In the analytic method, the end point locations and δ are exact to machine precision. In the iterative method the error decreases as the tolerance decreases. The results of the two cases are almost identical. Nine iterations are required for the solution to fully converge. All test cases were run on a Dell Workstation with dual 933 MHz Pentium III xeon processors to evaluate the CPU time. The CPU time required is reported for 100,000 repetitions, and increases as the tolerance is decreased for the iterative method. The CPU time required by the iterative method when it is fully converged is about 18 times that required by the analytic method. In addition, we scaled down the geometry of cases 1 and 2 to evaluate the effect of round-off error, and found that the results were exact for any scaling factor.

The 3D test cases are illustrated in Fig. 9. The same interfacial plane is used to test regimes I and III, as shown in Fig. 9(a), but fluid 1 lies on opposite sides of the plane between the two cases. The configuration used to test regime II is shown in Fig. 9(b). As in the 2D tests the line constant is computed as $\delta = \mathbf{n} \cdot \mathbf{x}$ at every polygon vertex for comparison to the iterative method. The results are given in Table 2. The analytic results are exact within machine accuracy. As in the 2D case, the results of regimes I and III are almost identical. Nine or 10 iterations are required for full convergence, in which case the iterative method requires about four times the CPU time of the analytic method. For regime II only eight iterations are needed, but 6.6 times the CPU time of the analytic method is required. In 3D the analytic method requires eight times as much CPU time as in 2D, while the iterative method requires less than twice as much time in 3D as in 2D for the same tolerance.

The analytic method is clearly faster in both 2D and 3D, but the complexity of the implementation is comparable. Implementation of the analytic method is quite simple, except perhaps the logic of reordering the verticies from A', B', C', and, in 3D, D' to A, B, C, and D. In 3D, the iterative method, we used also requires reordering of the verticies, but has the benefit that only one coordinate of one polygon vertex need be computed to determine the volume fraction, and hence δ. Further speed-up of the iterative 3D method may be attainable by implementing a clipping routine and computing the polygon volume using the formulas of Section 4, as we did in the 2D iterative method. In 2D, we found this to be slightly faster than using the formulas of Section 2. In 2D, the clipping subroutine is somewhat complex, but in a dynamic simulation it may be needed elsewhere, such as in computation of the flux.

Table 1
Results of the two-dimensional test cases using the direct and iterative methods

Method	Tolerance	% Error	CPU time	$t_{\text{iterative}}/t_{\text{analytic}}$	Iterations
Case 1					
Analytic	–	0	0.047	1.0	–
Iterative	1.E − 2	0.59	0.59	12.7	6
Iterative	1.E − 4	6.51E − 3	0.67	14.3	7
Iterative	1.E − 6	2.68E − 5	0.75	16.0	8
Iterative	1.E − 8	8.71E − 10	0.86	18.3	9
Iterative	1.E − 10	8.71E − 10	0.84	18.0	9
Case 2					
Analytic	–	0	0.047	1.0	–
Iterative	1.E − 2	0.59	0.59	12.7	6
Iterative	1.E − 4	6.51E − 3	0.69	14.7	7
Iterative	1.E − 6	2.68E − 5	0.78	16.7	8
Iterative	1.E − 8	8.71E − 10	0.88	18.7	9
Iterative	1.E − 10	8.71E − 10	0.86	18.3	9

The % error in δ, the CPU time required for 100,000 repetitions, the ratio of the time required by the iterative method to that required by the analytic method, and the number of iterations are reported.

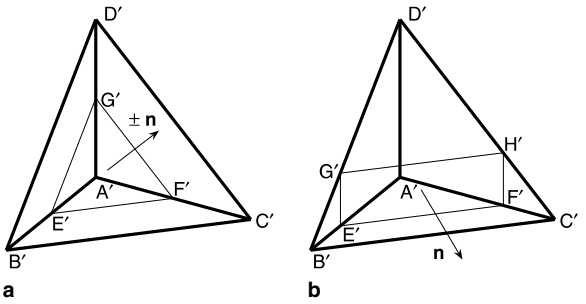

Fig. 9. Three dimensional test cases. The cell verticies are A', B', C', and D' with coordinates $(0,0,0)$, $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$, respectively. (a) Regimes I and III, the polygon verticies are E', F', and G' with coordinates $(1/2,0,0)$, $(0,1/2,0)$, and $(0,0,1/2)$, respectively. For regime I, the unit normal is $(1/\sqrt{3},1/\sqrt{3},1/\sqrt{3})$, $f=1/8$, and $\delta=1/(2\sqrt{3})$. For regime III, the unit normal is $(-1/\sqrt{3},-1/\sqrt{3},-1/\sqrt{3})$, $f=7/8$, and $\delta=-1/(2\sqrt{3})$. (b) Regime II, the polygon verticies are E', F', G', and H' with coordinates $(2/3,0,0)$, $(0,2/3,0)$, $(2/3,0,1/3)$, and $(0,2/3,1/3)$, respectively, the unit normal is $(1/\sqrt{2},1/\sqrt{2},0)$, $f=20/27$, and $\delta=\sqrt{2}/3$.

Table 2
Results of the three-dimensional test cases using the direct and iterative methods

Method	Tolerance	% Error	CPU time	$t_{\text{iterative}}/t_{\text{analytic}}$	Iterations
Regime I					
Analytic	–	0	0.36	1.0	–
Iterative	1.E − 2	5.16E − 2	1.05	2.8	6
Iterative	1.E − 4	9.06E − 4	1.33	3.5	8
Iterative	1.E − 6	4.68E − 7	1.50	4.0	9
Iterative	1.E − 8	4.68E − 7	1.50	4.0	9
Iterative	1.E − 10	0	1.48	4.0	9
Regime II					
Analytic	–	1.18E − 14	0.22	1.0	–
Iterative	1.E − 2	0.48	0.95	4.4	5
Iterative	1.E − 4	6.73E − 6	1.13	5.1	6
Iterative	1.E − 6	6.73E − 6	1.28	5.9	7
Iterative	1.E − 8	5.61E − 10	1.44	6.6	8
Iterative	1.E − 10	5.61E − 10	1.44	6.6	8
Regime III					
Analytic	–	0	0.38	1.0	–
Iterative	1.E − 2	5.16E − 2	1.06	2.8	6
Iterative	1.E − 4	9.06E − 4	1.38	3.7	8
Iterative	1.E − 6	4.68E − 7	1.50	4.0	9
Iterative	1.E − 8	4.68E − 7	1.52	4.0	9
Iterative	1.E − 10	1.92E − 14	1.67	4.5	10

The % error in δ, the CPU time required for 100,000 repetitions, the ratio of the time required by the iterative method to that required by the analytic method, and the number of iterations are reported.

7. Conclusions

We have derived analytic formulas to compute the positions of the endpoints of a linear interface approximation in a 2D, triangular grid cell and the vertices of a planar interface approximation in a 3D, tetrahedral grid cell, given the interface normal and the cell volume fraction. These formulas allow the interface reconstruction step of the volume of fluid method to be performed analytically, rather than iteratively, reducing the computational cost. Formulas to compute the volume of fluid in a given 2D or 3D region are also presented.

Simple test cases show that this analytic method is about 18 times faster than an iterative method in 2D and four to six times faster in 3D. The implementation of the analytic and iterative methods are of comparable complexity. The 2D formulation has been implemented in a coupled level set/volume of fluid method with a Stokes solver on an unstructured, adaptive mesh [6]. In future work, we will extend this work to 3D using the formulas developed here. The use of adaptive meshes is necessary to simulate interfacial flow problems with a wide range of length scales, such as drop coalescence and tip streaming.

Acknowledgments

The authors thank the reviewers for their useful suggestions.

References

[1] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, Phys. Fluids 39 (1981) 201–225.
[2] W.F. Noh, P.R. Woodward, in: A.I. van de Vooren, P.J. Zandbergen (Eds.), SLIC (Simple Line Interface Method), Lecture Notes in Physics, vol. 59, Springer, Berlin, 1976, p. 330.
[3] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[4] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[5] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, J. Comput. Phys. 164 (2000) 228–237.
[6] X. Yang, A.J. James, J. Lowengrub, X. Zheng, V. Cristini, An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids, J. Comput. Phys., in review.
[7] J.J. Tuma, Engineering Mathematics Handbook, second ed., McGraw-Hill, New York, 1979.
[8] D. Sunday, Fast polygon area and Newell normal computation, J. Graphics Tools: JGT 7 (2) (2002) 9–13.
[9] A.V. Gelder, Efficient computation of polygon area and polyhedron volume, in: A.W. Paeth (Ed.), Graphics Gems V, Academic Press, New York, 1995, pp. 35–41.
[10] R. Goldman, Area of planar polygons and volume of polyhedra, in: J. Arvo (Ed.), Graphics Gems II, Academic Press, New York, 1994, pp. 170–171.
[11] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, second ed.Numerical Recipes in Fortran 90, vol. 2, Cambridge University Press, Cambridge, 1996.