INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **58**:923–944
Published online 17 March 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1775

A new volume of fluid method in three dimensions—Part II: Piecewise-planar interface reconstruction with cubic-Bézier fit

J. López[1], C. Zanzi[2], P. Gómez[2], F. Faura[1] and J. Hernández[2,*,†]

[1]*Dept. de Ingeniería de Materiales y Fabricación, ETSII, Universidad Politécnica de Cartagena, E-30202 Cartagena, Spain*
[2]*Dept. de Mecánica, ETSII, UNED, E-28040 Madrid, Spain*

SUMMARY

A new interface reconstruction method in 3D is presented. The method involves a conservative level-contour reconstruction coupled to a cubic-Bézier interpolation. The use of the proposed piecewise linear interface calculation (PLIC) reconstruction scheme coupled to a multidimensional time integration provides solutions of second-order spatial and temporal accuracy. The accuracy and efficiency of the proposed reconstruction algorithm are demonstrated through several tests, whose results are compared with those obtained with other recently proposed methods. An overall improvement in accuracy with respect to other recent methods has been achieved, along with a substantial reduction in the central processing unit time required. Copyright © 2008 John Wiley & Sons, Ltd.

Received 3 August 2007; Revised 30 December 2007; Accepted 2 January 2008

KEY WORDS: volume tracking; volume of fluid method; 3D multidimensional advection; PLIC interface reconstruction; cubic-Bézier interpolation; level contour

1. INTRODUCTION

Among the several methods developed to simulate free surface and interfacial flows [1] (each of them with its own advantages and disadvantages), the volume of fluid (VOF) method is one of the most widely used. Their high efficiency in conserving mass and the simple way in which the topological changes of the interface are treated are two of the most relevant advantages of VOF methods.

[*]Correspondence to: J. Hernández, Dept. de Mecánica, ETSII, UNED, C/Juan del Rosal 12, E-28040 Madrid, Spain.
[†]E-mail: jhernandez@ind.uned.es

Contract/grant sponsor: Spanish Ministerio de Educación y Ciencia; contract/grant numbers: DPI2004-08198, PR2005-0296

Piecewise linear interface calculation (PLIC) methods are commonly used to reconstruct the interface in VOF methods [1–3]. At each cell, the interface orientation is usually obtained from an approximation of the gradient of a discrete function, F, whose value in each cell is the fraction of the cell occupied by the reference fluid, being equal to one in cells full of fluid, zero in empty cells and a value between zero and one in mixed cells containing the interface. The interface position is then determined from the value of the discretized F function in the cell. Among the most widely used PLIC methods, Youngs' explicit method [4] and Puckett's iterative method [5] may be mentioned. The former has an accuracy between first and second order, and the latter, which is computationally more expensive (the computational time might become prohibitive in 3D problems), is second-order accurate. Another second-order method, only applicable to orthogonal grids, was proposed by Chorin [6]. In this method, the interface orientation is calculated from a curve defined by a 3×3 array of volume fractions. An improvement to this method, which performs well on unstructured grids, was proposed by Swartz [7], and first described in algorithmic form by Mosso *et al.* [8]. In this improved version, the interface curve is constructed to pass through the midpoints of a tentative segment and its two neighbors. More recently, López *et al.* [9, 10] proposed similar approaches whereby a spline-interpolated curve passing through midpoints of tentative segments was used to obtain the interface orientation. A more detailed description of these and other PLIC reconstruction methods in 2D can be found in [9, 10].

To date, relatively few second-order reconstruction methods for 3D have been proposed. We might mention the extension of the non-iterative ELVIRA method [11, 12] to 3D proposed by Miller and Colella [13], which requires high computational effort because it involves between 72 and 144 candidate orientations and a 125-cell stencil for each cell interface to be reconstructed. Liovic *et al.* [14] proposed the iterative second-order CVTNA reconstruction method. This 3D method, which is based on the 2D method proposed by Swartz [7], achieves high-order accuracy within a 27-cell stencil and seems competitive with respect to other second-order methods. However, when used with a direction-split advection algorithm, it is still computationally expensive at high resolution, requiring about 50 times the central processing unit (CPU) time needed by Youngs' method. More recently, Aulisa *et al.* [15] proposed the least-squares fit (LSF) reconstruction method in 3D, which is second-order accurate when applied iteratively.

The aim of this work was to develop a fast, efficient and accurate reconstruction method in 3D which involves a non-iterative PLIC method using a conservative level-contour interface reconstruction coupled to a cubic-Bézier interpolation. The proposed reconstruction method is described in the following section. The results are compared in Section 3 with those obtained by several authors for the following tests: the reconstruction of interfaces with two different shapes, the advection of a spherical fluid body in simple translational and rotational flows and the interface deformation in two different flows with non-uniform vorticity.

2. INTERFACE RECONSTRUCTION

The interface is reconstructed in each cell as a plane defined by $\mathbf{n}\cdot\mathbf{x}+C=0$, where C is a constant, and $\mathbf{n}$ and $\mathbf{x}$ are, respectively, the inward (pointing to the fluid) unit vector normal to the interface and the location of a generic point on the plane. The aim of the proposed interface reconstruction method is to improve the accuracy of methods such as the well-known reconstruction method of Youngs, while keeping the additional required computational effort as low as possible. In order to assess the accuracy and efficiency of the proposed method, its results will be compared in

Section 3 with those of Youngs' method and of two more recent methods that have demonstrated their accuracy: the extension of the ELVIRA method [11, 12] to 3D proposed by Miller and Colella [13] and the iterative second-order CVTNA reconstruction method proposed by Liovic *et al.* [14]. The details of our implementation of Youngs' and the ELVIRA methods are briefly described in Appendix A.

The proposed PLIC method, which is an extension to 3D of the strictly conservative level-contour reconstruction method for 2D described in [10], basically consists of the following steps:

Step 1: *Construction of a 0.5-level contour*. The interface is first reconstructed at each time step from a 0.5-level contour of the volume fraction function. This procedure is similar to that described by Shin and Juric [16] for interface reconstruction in a front tracking method.

The interface is represented in each cell by sequentially joining the points at the edges of the six cell faces where the linearly interpolated volume fraction, F_p, is equal to 0.5. The location of each of these points, $\mathbf{x}_p$, is determined by linear interpolation from the values of the volume fraction at the ends of the corresponding cell edge, F_{e_i} and $F_{e_{i+1}}$ (Figure 1):

$$\mathbf{x}_p = \mathbf{x}_{e_{i+1}} - \frac{F_{e_{i+1}} - 0.5}{F_{e_{i+1}} - F_{e_i}} (\mathbf{x}_{e_{i+1}} - \mathbf{x}_{e_i}) \tag{1}$$

The volume fraction at end i of a given cell edge is calculated by averaging the volume fractions of the eight surrounding cells:

$$F_{e_i} = \frac{\sum_{k=1}^{8} F_k V_k}{\sum_{k=1}^{8} V_k} \tag{2}$$

where V_k is the volume of cell k. All the $\mathbf{x}_p$ points related to a single cell are connected by line segments to form a (warped) polygon; this 'interface polygon' represents the restriction of the 0.5-level contour to a single cell. The first point of the interface polygon will be at the edge of the first face found to contain an $\mathbf{x}_p$ point (the search starts from the face with the lowest index, i.e. face 1 in the case of Figure 1) between corners i and $i+1$ where $F_{e_i} \geqslant 0.5$ and $F_{e_{i+1}} < 0.5$, provided these are ordered counterclockwise. The next point will be at the edge of the same face between corners i and $i+1$ where $F_{e_i} < 0.5$ and $F_{e_{i+1}} \geqslant 0.5$. Then, the next face of the cell that shares the last two corners with the previous face is identified (face 2) and a new point of the interface is obtained similarly. This process is repeated until the interface polygon is closed.

The position vector, $\mathbf{x}_c$, of the centroid of the polygon formed by the interface segments is then calculated as

$$\mathbf{x}_c = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_{p_i} \tag{3}$$

where n is the number of cell edges with $F_p = 0.5$. Note that the closed polygon obtained from the 0.5-level contour is not usually planar.

In the case shown in Figure 1, there are four faces in which the interface segments have to be constructed, but up to six interface segments could in general be constructed in a cell of six faces. In certain circumstances, such as that described by the cell on the right of Figure 2, more than one 0.5-level-contour polygons could be constructed. However, in order to simplify the triangulation procedure described in the next step, we impose the restriction of constructing a single polygon per interfacial cell. This polygon is constructed by following simple rules: an arbitrarily predefined sequence of cell faces and vertices is used to select the polygon edges; no more than one polygon

DOI: 10.1002/fld

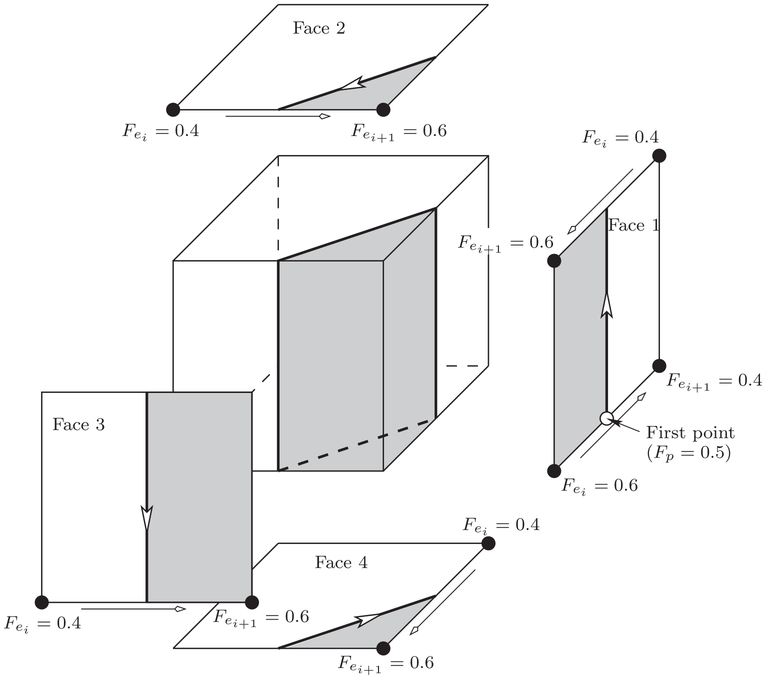

Figure 1. Construction of the 0.5-level contour (Step 1).

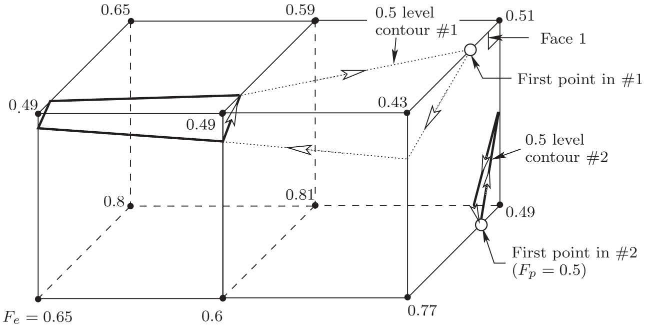

Figure 2. Situation in which the level-contour interfaces of adjacent cells may not be connected through their common face. The dotted line represents the alternative interface that would result if the level-contour construction began at the top edge of face 1.

edge per cell face is allowed, and the polygon must be closed (the first and last points must be the same). The extension of the proposed method to allow more than one interfacial polygon per cell will be the subject of a future work.

Step 2: *Surface triangulation and initial interface normal estimation*. The idea in this step is to obtain an initial estimation of the unit length normal of the interface plane equation at each cell. To this end, triangles are first formed by linking the centroid of the polygon obtained from the 0.5-level

DOI: 10.1002/fld

contour in the cell with the centroids of polygons at adjacent cells. Then, the initial estimation of the normal is calculated from a weighted average of the normals of the triangles. This approach is preferable to the local triangulation used by Shin and Juric [16], which is based on linking the edges and centroid of the 0.5-level-contour polygon at each cell, which would provide only first-order accuracy. Note that the interface constructed at a given cell from the 0.5-level contour will be connected with those of adjacent cells through their common face. This connectivity, which is also useful in the context of front tracking methods, as discussed by Shin and Juric [16], allows us to easily obtain a triangulated interface.

The details of the triangulation procedure are explained in the following with the aid of the example represented in Figure 3. The process starts by connecting the centroid of the 0.5-level-contour polygon in the cell to the left of the figure (point '*a*') with the centroids of the polygons in the adjacent cells (connection points $1, 2, 3$ and 4). At the cell faces adjacent to a domain boundary, the connection points are the corresponding centers of the 0.5-level-contour interface segments (point 5 of the bottom face in the example of Figure 3). Triangles are finally formed by linking the connection points. Once the local triangulated surface is obtained around point '*a*', the process continues around the centroid of the interface polygon in the cell to the right (point '*b*'). Note that if triangulation around this new point were accomplished using only points 6, 7, 8 and '*a*', implicitly connected with it (as made with point '*a*'), the resulting triangles would overlap with those formed around centroid '*a*'. To avoid this, points 2 and 5 (which, respectively, were

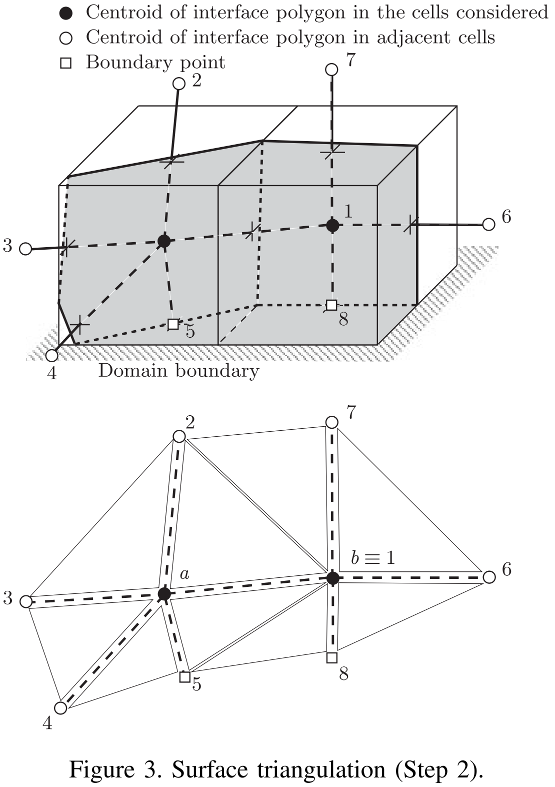
● Centroid of interface polygon in the cells considered
○ Centroid of interface polygon in adjacent cells
□ Boundary point
2
7
1
3
6
5
8
4
Domain boundary
2
7
a
b ≡ 1
3
6
5
8
4

Figure 3. Surface triangulation (Step 2).

DOI: 10.1002/fld

connected to point '*a*' just after and before point '*b*'), which are not implicitly connected with point '*b*', are used to obtain the triangulated surface by linking them with point '*b*'. This procedure is repeated until the whole interface is completely triangulated. Note that the resulting triangulated surface depends on the starting point of the triangulation procedure (for example, if it starts with point '*b*' instead of '*a*' the result will be different). In general, the algorithm produces a more regular triangulated surface when points with a higher connectivity between adjacent cells are considered first (see the example of Figure 4).

The initial estimation of the unit normal vector of the interface plane equation is obtained from the weighted average:

$$\mathbf{n} = \sum_{i=1}^{m} \mathbf{n}_{t_i} \alpha_i \Bigg/ \sum_{i=1}^{m} \alpha_i \tag{4}$$

where m ($\leqslant 12$) is the number of triangles surrounding the centroid $\mathbf{x}_c$, $\mathbf{n}_{t_i}$ is the inward unit normal vector of triangle i and α_i is the spanning angle of triangle i at the centroid (see Figure 5). The normal $\mathbf{n}_{t_i}$ can be obtained as

$$\mathbf{n}_{t_i} = \frac{\mathbf{l}_i \times \mathbf{l}_{i+1}}{|\mathbf{l}_i \times \mathbf{l}_{i+1}|} \tag{5}$$

where $\mathbf{l}_i = (\mathbf{x}_{c_i} - \mathbf{x}_c)/|\mathbf{x}_{c_i} - \mathbf{x}_c|$ and $\mathbf{l}_{i+1} = (\mathbf{x}_{c_{i+1}} - \mathbf{x}_c)/|\mathbf{x}_{c_{i+1}} - \mathbf{x}_c|$ ($\mathbf{x}_{c_i}$ and $\mathbf{x}_{c_{i+1}}$ are the other two vertices of triangle i), and the angle $\alpha_i = \arccos(\mathbf{l}_i \cdot \mathbf{l}_{i+1})$. Note that the normal $\mathbf{n}_{t_i}$ obtained from Equation (5) will point into the reference fluid, provided the triangle vertices are ordered counterclockwise as viewed from inside the fluid.

In situations such as that shown in Figure 2, the level-contour interfaces of adjacent cells (thick lines) may not be connected through their common face (note that if the construction of the level contour in the cell on the right of the figure began at the top edge of face 1, the level contour #1, instead of #2, would be obtained). A triangulated surface formed by connecting the centroid at the left cell and the centroid of the 0.5 level contour #2 would produce a poor estimation of the interface orientation. In this type of cell, or in cells that are not crossed by any 0.5-level contour but where $0 < F < 1$, the interface is reconstructed using Youngs' method, as described in Appendix A.

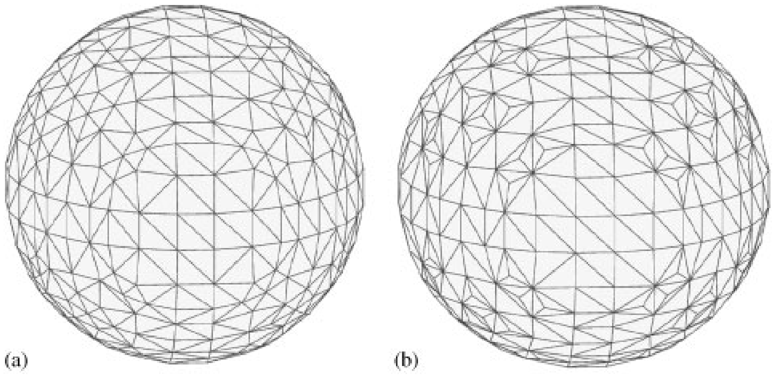

Figure 4. Surface triangulation of a sphere. The triangulation is carried out from (a) higher to lower connectivity points and from (b) lower to higher connectivity points.

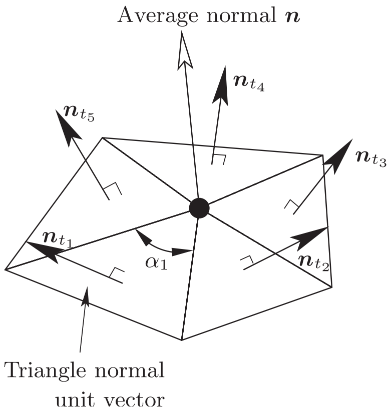

Figure 5. Interface orientation obtained in Step 2 from the triangulated surface.

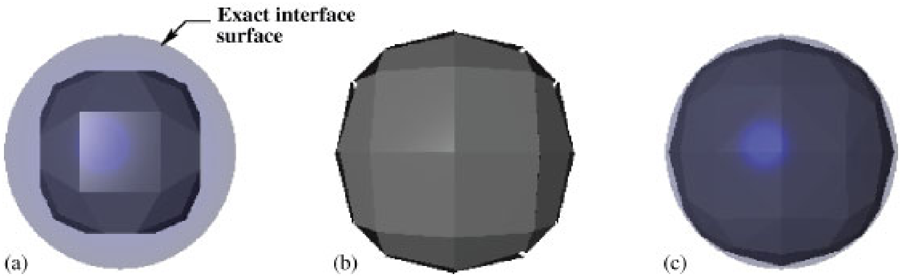

Figure 6. Example of reconstruction of a sphere in a coarse grid (sphere radius of $1.875\Delta x$): (a) triangulated surface obtained after step 2; (b) conservative planar interface (step 3); and (c) triangulated surface obtained after step 4.

Step 3: *Enforcement of volume conservation*. A plane interface surface with the normal **n** obtained in Step 2 is positioned in every cell by enforcing volume conservation, using the analytical method of Scardovelli and Zaleski [17] to calculate the constant C (an iterative method, such as Brent's method [18], could also be employed).

Step 4: *Interface normal adjustment*. In this step, point $\mathbf{x}_c$ of the triangulated surface in every interfacial cell obtained from Equation (3) (see the example of Figure 6(a)) is relocated at the centroid of the interface surface obtained in Step 3 (Figure 6(b)). Once the triangulated surface is readjusted in this way (Figure 6(c)), the normal vector calculation described in Step 2 is applied again. The improved accuracy obtained with this interface normal readjustment can be observed, for sufficiently fine grids, from the results of Tables I and II, which are discussed in Section 3.1.

This type of adjustment, as discussed below, was first proposed by Swartz [7] for 2D problems.

Step 5: *Additional adjustment of the interface normal by cubic-Bézier interpolation*. To further increase the accuracy of the interface orientation in each cell before calculating the final value of constant C of the interface plane equation, the vector normal to the interface, **n**, can be readjusted

DOI: 10.1002/fld

Table I. Reconstruction errors obtained for a sphere using different reconstruction algorithms and grid sizes.

Grid size	Youngs	LSF [15]	LLCIR	ELCIR	CLCIR	CLC-CBIR	CLCIR with 2 iter. in Step 4	CLCIR-T2
10^3	1.89×10^{-3}	1.92×10^{-3}	2.14×10^{-3}	2.23×10^{-3}	2.38×10^{-3}	2.43×10^{-3}	2.33×10^{-3}	2.87×10^{-3}
	(1.84)	(2.01)	(2.03)	(1.99)	(2.11)	(2.11)	(2.09)	(1.95)
20^3	5.28×10^{-4}	4.77×10^{-4}	5.23×10^{-4}	5.62×10^{-4}	5.50×10^{-4}	5.64×10^{-4}	5.46×10^{-4}	7.45×10^{-4}
	(1.45)	(2.00)	(1.89)	(1.98)	(2.08)	(2.12)	(2.07)	(2.15)
40^3	1.93×10^{-4}	1.19×10^{-4}	1.41×10^{-4}	1.42×10^{-4}	1.30×10^{-4}	1.30×10^{-4}	1.30×10^{-4}	1.68×10^{-4}
	(1.17)	(2.00)	(1.61)	(1.88)	(2.01)	(2.03)	(2.01)	(1.96)
80^3	8.60×10^{-5}	2.98×10^{-5}	4.61×10^{-5}	3.86×10^{-5}	3.23×10^{-5}	3.18×10^{-5}	3.22×10^{-5}	4.33×10^{-5}
	(1.06)	(2.00)	(1.32)	(1.69)	(2.01)	(2.02)	(2.01)	(2.07)
160^3	4.12×10^{-5}	7.46×10^{-6}	1.85×10^{-5}	1.20×10^{-5}	8.00×10^{-6}	7.82×10^{-6}	7.99×10^{-6}	1.03×10^{-5}
	(1.02)	—	(1.14)	(1.42)	(2.00)	(2.00)	(2.01)	(2.13)
320^3	2.03×10^{-5}	—	8.40×10^{-6}	4.47×10^{-6}	2.00×10^{-6}	1.95×10^{-6}	1.99×10^{-6}	2.36×10^{-6}

The orders of convergence are in parentheses. The results of the last column (CLCIR-T2) are obtained using the CLCIR method when the triangulation of Step 2 is carried out, as in the case of Figure 4(b), from lower to higher connectivity points.

Table II. Reconstruction errors obtained for a hollow sphere using different reconstruction algorithms and grid sizes.

Grid size	Youngs	Original ELVIRA-3D	Improved ELVIRA-3D	Liovic *et al.* [14] CVTNA	CLCIR	CLC-CBIR
10^3	3.63×10^{-3}	2.89×10^{-2}	6.89×10^{-3}	2.06×10^{-2}	4.68×10^{-3}	4.91×10^{-3}
	(1.85)	(3.53)	(2.56)	(3.78)	(2.20)	(2.25)
20^3	1.01×10^{-3}	2.50×10^{-3}	1.17×10^{-3}	1.50×10^{-3}	1.02×10^{-3}	1.03×10^{-3}
	(1.45)	(1.96)	(2.05)	(2.57)	(2.09)	(2.10)
40^3	3.70×10^{-4}	6.43×10^{-4}	2.82×10^{-4}	2.52×10^{-4}	2.42×10^{-4}	2.41×10^{-4}
	(1.18)	(1.96)	(2.02)	(1.99)	(2.02)	(2.04)
80^3	1.63×10^{-4}	1.65×10^{-4}	6.93×10^{-5}	6.36×10^{-5}	5.95×10^{-5}	5.85×10^{-5}
	(1.06)	(1.96)	(2.01)	(1.70)	(2.01)	(2.02)
160^3	7.81×10^{-5}	4.24×10^{-5}	1.72×10^{-5}	1.95×10^{-5}	1.48×10^{-5}	1.44×10^{-5}

The orders of convergence are in parentheses.

with the aid of a cubic-Bézier interpolation. The improvement arises from the increase of the interpolation order provided by the use of Bézier patches to regularize the triangulated interface.

The idea is to replace the triangulated interface by a set of Bézier patches, which, in most cases, will provide a more accurate estimation of the interface normal at every cell.

A cubic-Bézier patch (see, for example, [19]) is constructed over each triangle of the triangulated surface obtained in Step 4. The barycentric coordinates of the patch will be defined as $u=i/3$, $v=j/3$ and $w=k/3$, where $i+j+k=3$ and $i, j, k\geqslant 0$. The locations of the three patch vertices, $\mathbf{b}_{i=3,j=0,k=0}$, $\mathbf{b}_{030}$ and $\mathbf{b}_{003}$, are made coincident with those of the three vertices of the triangle, $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ (see Figure 7). To complete the patch definition, six additional control points are needed on the boundaries (two points on each one of the boundaries $u=0$, $v=0$ and $w=0$) and another one inside the patch (see Figure 7). The locations of the control points on the boundaries are obtained using the following cubic Hermite interpolation:

$$
\begin{aligned}
\mathbf{b}_{210} &= (2\mathbf{x}_1+\mathbf{x}_2-\gamma_{12}\mathbf{n}_1)/3\\
\mathbf{b}_{120} &= (2\mathbf{x}_2+\mathbf{x}_1-\gamma_{21}\mathbf{n}_2)/3\\
\mathbf{b}_{021} &= (2\mathbf{x}_2+\mathbf{x}_3-\gamma_{23}\mathbf{n}_2)/3\\
\mathbf{b}_{012} &= (2\mathbf{x}_3+\mathbf{x}_2-\gamma_{32}\mathbf{n}_3)/3\\
\mathbf{b}_{102} &= (2\mathbf{x}_3+\mathbf{x}_1-\gamma_{31}\mathbf{n}_3)/3\\
\mathbf{b}_{201} &= (2\mathbf{x}_1+\mathbf{x}_3-\gamma_{13}\mathbf{n}_1)/3
\end{aligned}
\tag{6}
$$

where $\gamma_{ij}=(\mathbf{x}_j-\mathbf{x}_i)\cdot\mathbf{n}_i$ and $\mathbf{n}_i$ is the normal obtained at $\mathbf{x}_i$ in Step 4. The location of the control point in the interior is obtained from the following quadratic interpolation:

$$
\mathbf{b}_{111}=(\mathbf{b}_{210}+\mathbf{b}_{120}+\mathbf{b}_{021}+\mathbf{b}_{012}+\mathbf{b}_{102}+\mathbf{b}_{201})\tfrac{1}{4}-(\mathbf{b}_{300}+\mathbf{b}_{030}+\mathbf{b}_{003})\tfrac{1}{6} \tag{7}
$$

The resulting cubic surface can be expressed as

$$
\mathbf{S}(u,v,w)=\sum_{i+j+k=3;\, i,j,k\geqslant 0}\frac{3!}{i!j!k!}u^i v^j w^k \mathbf{b}_{ijk}
$$

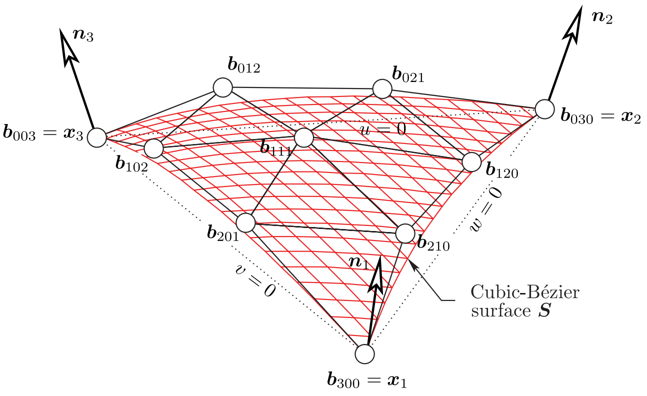

Figure 7. Cubic-Bézier patch used for interpolation in each triangle.

from which a new normal vector $\mathbf{n}_t^*$ is analytically obtained for each triangle at the barycenter point $u=1/3$, $v=1/3$ and $w=1/3$ as, for example,

$$\mathbf{n}_t^* = \frac{\partial \mathbf{S}}{\partial u} \times \frac{\partial \mathbf{S}}{\partial v} \bigg/ \left| \frac{\partial \mathbf{S}}{\partial u} \times \frac{\partial \mathbf{S}}{\partial v} \right|$$

The normal unit vector defining the interface orientation is finally obtained as the weighted average:

$$\mathbf{n} = \sum_{i=1}^{m} \mathbf{n}_{t_i}^* \alpha_i \bigg/ \sum_{i=1}^{m} \alpha_i$$

where α_i is calculated as indicated in Step 2.

Step 6: *Final enforcement of volume conservation*. The constant C of the interface plane (with the adjusted normal unit vector $\mathbf{n}$) is finally obtained from volume enforcement using the analytical method of Scardovelli and Zaleski [17].

The conservative level-contour interface reconstruction procedure consisting of Steps 1–4 and 6 will henceforth be referred to as the CLCIR method, and the coupled conservative level-contour and cubic-Bézier-based interface reconstruction procedure (including Step 5), as the CLC-CBIR method.

Two additional simplified versions of the proposed reconstruction algorithm have been implemented in order to assess the advantages of using Step 2 (construction of an extended triangulated surface over adjacent cells) and Step 4 (interface normal adjustment). The first version consists of Step 1, a local triangulation step and Step 6, and will be referred to as the local level-contour interface reconstruction (LLCIR) method. The local triangulation is based on linking the vertices and centroid of the 0.5-level-contour polygon in each cell, and the unit normal vector of the interface plane equation is calculated using the weighted average expression of Equation (4) (obviously, in this case $3\leqslant m\leqslant 6$). The second version consists of Steps 1, 2 and 6 and will be referred to as the extended level-contour interface reconstruction (ELCIR) method.

It should be pointed out that the adjustment described in Step 4, along with the enforcement of volume conservation of Step 6, can be considered as a 3D version of a single step of the iterative Mosso–Swartz interface reconstruction algorithm [7, 8]. Swartz [7] showed that this step, repeated iteratively, makes PLIC interfaces converge to a second-order approximation quadratically, which may explain why a single iteration is so good in dumping the linear component of the reconstruction error. In Section 3, we show that, for curved interfaces, a single application of this adjustment makes the proposed method to be, even with very fine grids, effectively second-order accurate in 3D.

It should also be mentioned that, when the proposed reconstruction method is used, the number of interfacial cells reconstructed with Youngs' method is very small (typically, a fraction lower than 1% of the total), all these cells having values of the volume fraction very close to 0.0 or 1.0. This is the reason why the overall second-order convergence rate provided by the proposed method in the tests presented in Section 3 does not deteriorate. These situations are similar to those of the near-homogeneous cell cases (that is, with a volume fraction close to 0.0 or 1.0) found by Liovic *et al.* [14] in their extension to 3D of the method of Swartz [7], which are handled using filtering procedures to exclude 'unsuitable triplets' and do not allow for the planarity preservation criterion to be satisfied.

3. RESULTS AND DISCUSSION

In this section, the accuracy and efficiency of the proposed reconstruction method are analyzed using different tests. The accuracy of the proposed reconstruction algorithm is first assessed in a test that analyzes its ability to reconstruct interfaces of known geometries. The next two tests correspond to the advection of a spherical fluid body in simple translational and rotational flows. Then, the ability of the proposed method to handle interfaces that undergo large deformation is analyzed in two different flows with non-uniform vorticity. All the results are obtained using the multidimensional advection algorithm FMFPA-3D proposed by us in a companion paper [20]. The results are compared with those obtained with different reconstruction methods implemented in this work and with recent results available in the literature. The volume fractions are initialized as in [21, 22], using a recursive local mesh refinement technique. This technique converges to second order and is very accurate, which is very important for obtaining good estimations of the reconstruction errors and convergence orders presented in the next subsection.

3.1. Accuracy of the reconstruction algorithm

The accuracy, convergence order and efficiency of the proposed reconstruction algorithm are analyzed in this subsection. First, the ability of the proposed algorithm to reconstruct planar interfaces is assessed. Then, the convergence order and computational efficiency are discussed for different spherical shapes.

3.1.1. Plane reconstruction. Several reconstructions of planes with different orientations were carried out. As a conclusion, it should be mentioned that the proposed CLCIR method cannot reproduce a planar interface exactly unless Step 4 of the proposed algorithm is applied iteratively. However, we have found that the criterion for planarity preservation used by Liovic *et al.* [14] ($|n_x - n_x^{\text{exact}}|, |n_y - n_y^{\text{exact}}|, |n_z - n_z^{\text{exact}}| < 10^{-4}$) is nearly satisfied with only one additional application of Step 4 (a similar behavior has been observed by Aulisa *et al.* [15] in their recently proposed LSF reconstruction method). On the other hand, for curved interfaces, only a single application of Step 4 is necessary to provide second-order accuracy, even with very fine grids; hence, Step 4 will be applied only once in the rest of the tests presented in this section.

3.1.2. Sphere reconstruction. This test consists of reconstructing a sphere of radius 0.325, which, in order to avoid artificial regularity of the results, is centered at (0.525, 0.464, 0.516) in a unit domain.

Table I shows the reconstruction errors and the order of convergence. The reconstruction error is defined as the volume between the exact interface and its approximate representation. This volume has been determined using an error measurement procedure similar to that described by Aulisa *et al.* [15] (note that the results of the first column of Table I are very close to those obtained by Aulisa *et al.* [15] using Youngs' method). The order of convergence can be determined from

$$\mathcal{O} = \frac{\ln(E_{2n}/E_n)}{\ln(1/2)} \tag{8}$$

where E_n and E_{2n} are the errors obtained using two different grids with n^3 and $(2n)^3$ cells, respectively.

DOI: 10.1002/fld

It can be observed that the use of the extended triangulated surface (Step 2) in the ELCIR method considerably increases the convergence order for the finest grid and that the interface normal adjustment (Step 4) used in the CLCIR and CLC-CBIR methods increases still further the order of convergence and the accuracy of the results. For the finest grid (which provides high resolution in 3D, with a ratio between the radius of the sphere and the cell size of 104), it can be observed that the proposed CLCIR method keeps the second-order convergence and the reconstruction errors are one order of magnitude lower than those obtained with Youngs' method. Also, note that the increase of the interpolation order provided by the use of cubic-Bézier patches only slightly improves the accuracy of the results for sufficiently fine grids. In practical situations in which the fluid body may reach complex geometries, such as those presented in the deformation tests considered in the following sections, this improvement will become more evident (the improvement in accuracy may be as high as 30%).

The sixth column of Table I, corresponding to results obtained with the CLCIR method using two iterations in Step 4, shows that the improvement obtained with an additional iteration for fine grids is lower than that obtained by the use of cubic-Bézier patches. The improvement in accuracy obtained when Step 4 is applied iteratively in the deformation tests is also very low. Similar findings were obtained by Aulisa *et al.* [15] when applying their recently proposed 3D LSF method iteratively. The results obtained by these authors with only one application of the LSF method are included in the second column of the table. Note that the accuracy of the proposed method is very close to that of the LSF method, especially for fine grids.

The last column of Table I (CLCIR-T2) shows results obtained with the CLCIR method when the triangulation of Step 2 is carried out from lower to higher connectivity points (as in the case of Figure 4(b)). These results show that the use of a more regular triangulated interface surface may significantly increase the accuracy. A detailed analysis of this issue will be the subject of future work.

3.1.3. Hollow sphere reconstruction. The accuracy and efficiency of the proposed reconstruction algorithm (with and without the cubic-Bézier interpolation step) in reproducing the hollow sphere considered by Liovic *et al.* [14] (sphere of radius 0.4 with a spherical core of radius 0.2) are compared in this section with those obtained with our implementations of Youngs' and ELVIRA algorithms, which are described in Appendix A, and with the CVTNA reconstruction method proposed by Liovic *et al.* [14]. As in the sphere test, the hollow sphere is centered at (0.525, 0.464, 0.516) in a unit domain. The results of Table II show that the accuracy of the proposed reconstruction algorithm for the coarser grids is only slightly lower than that of Youngs' method and considerably higher than that of the original ELVIRA-3D and CVTNA methods. Obviously, for finer grids, second-order methods provide much better accuracy and the proposed method represents an improvement over the other methods depicted in the table. The good performance of Youngs' algorithm for coarser grids may appear to be more evident than in the 2D version because the grid refinement permitted (for given computational resources) in 2D is usually substantially higher than in 3D.

Table III shows a comparison between the CPU times (relative to those required by Youngs' method) consumed by the second-order reconstruction methods of Table II in the reconstruction of the hollowed sphere. All test cases were run on a workstation with dual 2 GHz Intel T7200 processors. Although the improved version of the ELVIRA method implemented in 3D considerably reduces the time required by the original version (both the improved and original versions of the method are described in Appendix A), it can be observed that its computational cost is still very

DOI: 10.1002/fld

Table III. Relative CPU times consumed by different second-order reconstruction algorithms in the hollowed sphere test, using different grid sizes.

Grid size	Original ELVIRA-3D	Improved ELVIRA-3D	Liovic *et al.* [14] CVTNA	Present work CLCIR	Present work CLC-CBIR
10^3	2547.5	827.4	65.7	2.6	3.6
20^3	3272.6	899.0	57.7	3.2	4.3
40^3	2975.3	835.5	46.7	3.6	4.8
80^3	2634.6	797.6	43.7	4.0	5.1

Values are relative to the CPU time required by Youngs' method using the same grid. The values corresponding to the CVTNA method are taken from Table 2 in [14].

high (about a thousand times higher than that for Youngs' method). Also, note that the CVTNA method of Liovic *et al.* [14] is about 50 times more time consuming than Youngs' method, which represents a considerable improvement with respect to the ELVIRA-3D method. On the other hand, the relative CPU time consumed by the proposed reconstruction algorithm is around one order of magnitude lower than that of the CVTNA method, which makes it very efficient even in combination with a split advection scheme. Note that the much better efficiency of the proposed reconstruction method is accompanied by a slight increase in overall accuracy with respect to ELVIRA-3D and CVTNA second-order methods.

3.2. *Simple translation test*

The test consists of translating a sphere of fluid of radius 0.2, initially centered at point $(0.25, 0.25, 0.25)$, by means of a steady and uniform flow with velocity components $(1, 1, 1)$ for a time $T = 0.5$, in a domain of $1\times 1\times 1$. The error is estimated with an L_1 error norm defined as

$$E = \sum_{i,j,k} V_{\Omega}^{(i,j,k)} |F^{(i,j,k)} - F_{\mathrm{e}}^{(i,j,k)}| \tag{9}$$

where $V_{\Omega}^{(i,j,k)}$ is the volume of cell (i, j, k), and $F^{(i,j,k)}$ and $F_{\mathrm{e}}^{(i,j,k)}$ are, respectively, the calculated and exact volume fractions at the instant $t = T$.

Figure 8 shows the results for E, as a function of the Courant number, obtained with three grid sizes and different reconstruction methods implemented in this work. It should be mentioned that, for the steady and uniform flow considered in this test, the contribution of the advection step to the error is negligible; therefore, the error E is only due to reconstruction. On the other hand, as in a similar test carried out by Harvie and Fletcher [23] and López *et al.* [9] for 2D, the reconstruction error tends to reach a minimum value as the Courant number approaches unity and tends to reach a bounded value as the Courant number approaches zero. As discussed by Harvie and Fletcher [23], this asymptotic behavior is important for the viability of the method when small time steps are required. Note that Youngs' method is more accurate for coarser grids, although its convergence order is considerably lower ($\mathcal{O} = 0.75$ for the finest grid and the lowest Courant–Friedrichs–Lewy (CFL) number considered in this test) than that obtained with the reconstruction method proposed in this work ($\mathcal{O} = 1.91$ and 1.82 with and without the CBIR step, respectively). Also note that for this test, in which the interface should keep its spherical shape, the additional CBIR step does not improve the accuracy of the proposed reconstruction method.

DOI: 10.1002/fld

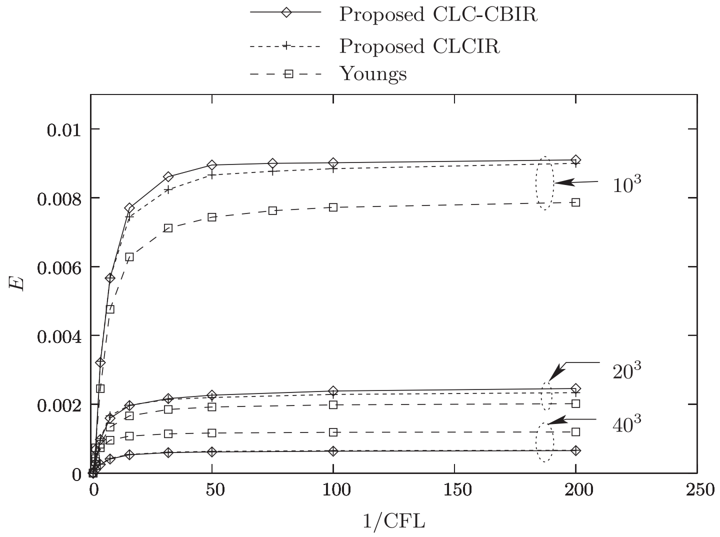

Figure 8. L_1 error norm obtained in the translation test using three different grid sizes, as a function of the CFL number.

3.3. Rotation test

The second test involves the rotation of a sphere of fluid of radius 0.15, initially centered at $(0.5, 0.75, 0.5)$, around an axis parallel to the z-axis and centered in a domain of size $1\times 1\times 1$, in a uniform vorticity field. A CFL number based on the maximum velocity component in the domain approximately equal to 0.5 was used.

Table IV shows the error E and the order of convergence after the sphere of fluid undergoes one complete revolution. The results are obtained with the FMFPA-3D advection method [20] and several reconstruction methods. Note that the shape of the sphere hardly changes during the computation; hence, as in the test in the previous section, the use of an additional CBIR step does not improve the accuracy of the results. It can be observed that the accuracy of the reconstruction method of Youngs for coarser grids is similar to that of the proposed reconstruction method, although the higher convergence order of the latter substantially improves its accuracy for fine grids. Also note that the improved ELVIRA-3D method also reaches nearly second-order accuracy, although at the cost of a considerable increment in CPU time.

3.4. Deformation test

In this test, proposed by LeVeque [24], a sphere of fluid of radius 0.15, initially centered at $(0.35, 0.35, 0.35)$ within a unit computational domain, is deformed in a solenoidal velocity field given by

$$\begin{aligned} u(x,y,z) &= 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\\ v(x,y,z) &= -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\\ w(x,y,z) &= -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z) \end{aligned} \tag{10}$$

which was modulated in time by multiplying by the factor $\cos(\pi t/T)$, using a period $T=3$.

DOI: 10.1002/fld

Table IV. L_1 error norm, E, and order of convergence, $\mathcal{O}$, obtained in the rotation of a spherical fluid body test, using different reconstruction methods and four grid sizes.

Reconstruction algorithm	Grid size	E	$\mathcal{O}$
Youngs	32^3	4.32×10^{-4}	
			1.54
	64^3	1.49×10^{-4}	
			1.23
	128^3	6.36×10^{-5}	
			1.13
	256^3	2.91×10^{-5}	
Improved ELVIRA-3D	32^3	4.92×10^{-4}	
			1.82
	64^3	1.39×10^{-4}	
			1.91
	128^3	3.70×10^{-5}	
			—
	256^3	—	
CLCIR	32^3	4.35×10^{-4}	
			1.87
	64^3	1.19×10^{-4}	
			1.88
	128^3	3.24×10^{-5}	
			1.74
	256^3	9.67×10^{-6}	
CLC-CBIR	32^3	4.96×10^{-4}	
			1.94
	64^3	1.29×10^{-4}	
			1.93
	128^3	3.38×10^{-5}	
			1.73
	256^3	1.02×10^{-5}	

Table V shows the error E and order of convergence at $t=T$ obtained using the different reconstruction methods implemented in this work and the FMFPA-3D advection method [20]. A CFL number (based on the maximum velocity component in the domain) equal to 1.0 was used. Note that, in this case, the additional CBIR step used in the proposed reconstruction method considerably increases the accuracy of the results. As regards the comparison with the improved ELVIRA-3D method, conclusions similar to those deduced from Tables III and IV can be drawn.

The results of the proposed method are also compared with those obtained with the hybrid particle level set method of Enright *et al.* [25]. For this purpose, the results obtained with the FMFPA-3D and CLC-CBIR methods, with a CFL number equal to 1.0 and the same grid size (100^3) used in [25], are presented in Figure 9(a) for different instants ($t=nT/15$, with $n=0,1,2,\ldots,7,9,11,13,15$).

Table V. L_1 error norm, E, and order of convergence, $\mathcal{O}$, obtained in the deformation test of LeVeque [24] with CFL = 1.0, using different reconstruction methods and four grid sizes.

Reconstruction algorithm	Grid size	E	$\mathcal{O}$
Youngs	32^3	7.22×10^{-3}	
			1.46
	64^3	2.62×10^{-3}	
			1.93
	128^3	6.86×10^{-4}	
			1.71
	256^3	2.09×10^{-4}	
Improved ELVIRA-3D	32^3	7.35×10^{-3}	
			1.45
	64^3	2.69×10^{-3}	
			2.05
	128^3	6.51×10^{-4}	
			—
	256^3	—	
CLCIR	32^3	6.85×10^{-3}	
			1.53
	64^3	2.38×10^{-3}	
			2.47
	128^3	4.31×10^{-4}	
			2.47
	256^3	7.78×10^{-5}	
CLC-CBIR	32^3	6.64×10^{-3}	
			1.67
	64^3	2.09×10^{-3}	
			2.57
	128^3	3.52×10^{-4}	
			2.59
	256^3	5.86×10^{-5}	

A comparison by visual inspection with the corresponding results obtained by Enright *et al.* [25] shows that the results obtained with our method compare relatively well with their hybrid particle level set solution and favorably with their standard level set results. It is worth noting that the net change in total volume at the end of the test ($t=T$) predicted by the method proposed here is about 10^{-6}%, whereas with the standard level set method about 80% of the initial volume is lost and about 2.6% with the particle level set method. The corresponding results obtained with a finer grid of size 200^3 are presented in Figure 9(b). Note that, in this case, the grid resolution is sufficiently high to appropriately resolve the thin structures that appear when the body of fluid undergoes the maximum deformation. The values of the error E obtained for the grid sizes of 100^3 and 200^3 cells are 6.60×10^{-4} and 9.94×10^{-5}, respectively, representing a convergence order of 2.73.

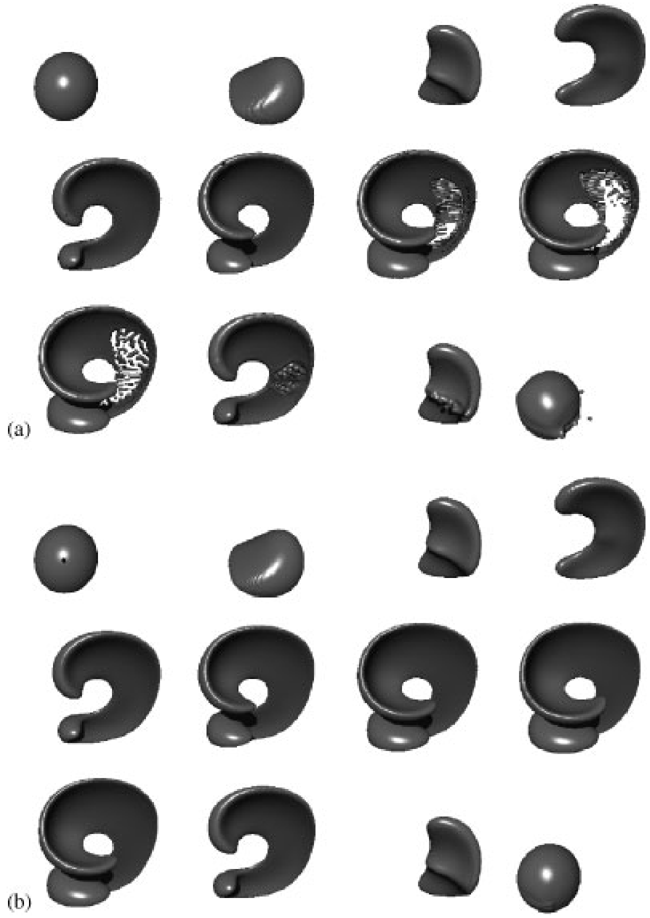

Figure 9. Results for the $F=0.5$ iso-surfaces of the deformation test of LeVeque [24] at instants $t=nT/15$, with $n=0,1,2,\ldots,7,9,11,13,15$, for grid sizes of (a) 100^3 and (b) 200^3.

3.5. 3D shearing flow test

In this test, a sphere of fluid of radius 0.15, initially centered at $(0.5,0.75,0.25)$ in a domain of size $1\times1\times2$, is deformed in the velocity field defined as

$$\begin{aligned} u(x,y,z) &= \sin^2(\pi x)\sin(2\pi y)\cos(\pi t/T) \\ v(x,y,z) &= -\sin^2(\pi y)\sin(2\pi x)\cos(\pi t/T) \\ w(x,y,z) &= \{1-2[(x-0.5)^2+(y-0.5)^2]^{1/2}\}^2\cos(\pi t/T) \end{aligned} \tag{11}$$

which is a combination of the classical vortex-in-a-box test of Rider and Kothe [2] with a laminar pipe flow in the z-direction, as in the work of Liovic *et al.* [14]. Table VI shows, for different grid sizes, the error E and order of convergence at $t=T$ obtained for values of $T=3,6$ and 9, using the reconstruction methods implemented in this work and the FMFPA-3D advection method [20], Table VI. L_1 error norm, E, and order of convergence, $\mathcal{O}$, obtained for three values of T and different grid sizes in the time-reversed single vortex test with a shear flow velocity along the z-axis and CFL$=1.0$, using different reconstruction methods.

Reconstruction algorithm	Grid size	$T=3.0$		$T=6.0$		$T=9.0$	
		E	$\mathcal{O}$	E	$\mathcal{O}$	E	$\mathcal{O}$
Youngs	$32\times32\times64$	3.74×10^{-3}		1.21×10^{-2}		1.60×10^{-2}	
			1.74		1.47		0.79
	$64\times64\times128$	1.12×10^{-3}		4.38×10^{-3}		9.27×10^{-3}	
			1.75		1.61		1.58
	$128\times128\times256$	3.34×10^{-4}		1.43×10^{-3}		3.09×10^{-3}	
CLCIR	$32\times32\times64$	3.45×10^{-3}		1.15×10^{-2}		1.61×10^{-2}	
			1.85		1.55		0.98
	$64\times64\times128$	9.58×10^{-4}		3.92×10^{-3}		8.14×10^{-3}	
			2.00		1.72		1.61
	$128\times128\times256$	2.40×10^{-4}		1.19×10^{-3}		2.67×10^{-3}	
CLC-CBIR	$32\times32\times64$	3.03×10^{-3}		9.95×10^{-3}		1.47×10^{-2}	
			1.99		1.61		1.06
	$64\times64\times128$	7.64×10^{-4}		3.27×10^{-3}		7.03×10^{-3}	
			2.09		1.82		1.70
	$128\times128\times256$	1.80×10^{-4}		9.27×10^{-4}		2.17×10^{-3}	
CVTNA	*PCFSC*, Liovic *et al.* [14]						
	$64\times64\times128$	6.20×10^{-4}		3.64×10^{-3}		8.01×10^{-3}	
Front tracking	*LGB*, Du *et al.* [26]						
	$64\times64\times128$	6.28×10^{-4}		9.07×10^{-4}		1.19×10^{-3}	

The results of the last two rows, for the intermediate grid size, are taken from Table 5 in [26].

with a CFL number equal to 1.0. As in the previous tests, the proposed reconstruction method obviously provides, in all cases, an accuracy and convergence order higher than those of Youngs' method. This improvement can be visually observed from Figure 10, where a comparison between the results obtained with Youngs' (Figure 10(a)) and the CLC-CBIR (Figure 10(b)) methods for the intermediate grid size and $T=9$ is presented. Note the significant reduction in the fragmentation of the fluid achieved when the proposed reconstruction method is used.

The results shown in the last two rows of Table VI correspond to those obtained by Liovic *et al.* [14] using their unsplit advection (PCFSC) and reconstruction (CVTNA) methods and by Du *et al.* [26] using their front tracking method (LGB), for the intermediate grid of $64\times64\times128$. Note that, except when $T=3$, the FMFPA-3D method coupled with the CLC-CBIR method showed a slightly higher accuracy than that obtained by Liovic *et al.* [14] using their more expensive iterative second-order reconstruction method, especially in cases with large deformation. Also, note that the front tracking method again provides a higher accuracy than that obtained with any of the PLIC-VOF methods presented in the table, especially for the cases with greater deformations. However, the relative merits of the different methods should also be assessed taking into account the required computational effort, and, from this point of view, the PLIC-VOF methods have several advantages.

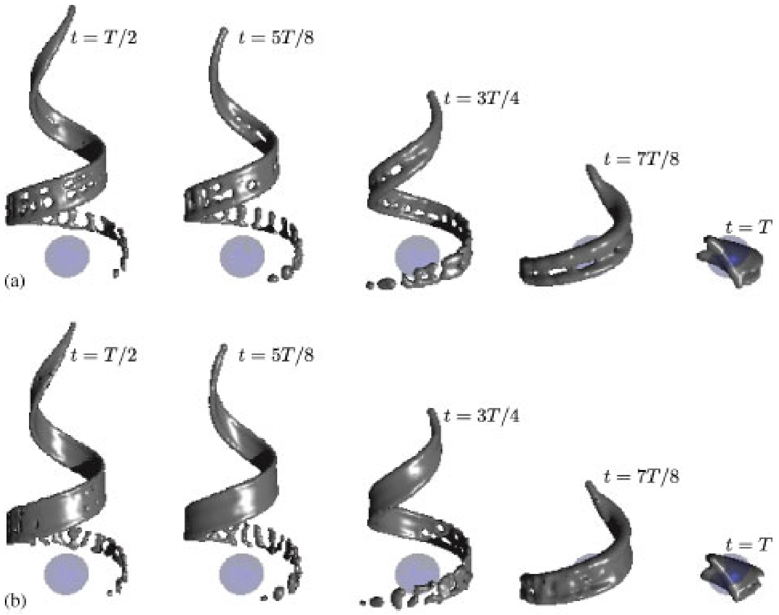

Figure 10. Results for the $F = 0.5$ iso-surfaces of the vortex-in-a-box test with a non-zero velocity component along the z-axis and $T = 9$, obtained using (a) Youngs' method and (b) the proposed CLC-CBIR method, for the intermediate grid size of $64 \times 64 \times 128$.

4. CONCLUSIONS

A new non-iterative PLIC reconstruction method has been developed for tracking interfaces in 3D. The new reconstruction method uses a conservative level-contour interface reconstruction coupled with a cubic-Bézier interpolation and achieves second-order accuracy even for high grid resolutions. The proposed reconstruction method has been assessed using different tests. The overall accuracy of the results is found to be slightly higher than that obtained with other schemes of similar degree of complexity, especially in flows involving large deformation, and, at the same time, a substantial reduction in the CPU time required by the reconstruction algorithm is achieved.

APPENDIX A: IMPLEMENTATION DETAILS OF THE RECONSTRUCTION METHODS USED FOR COMPARISON

A.1. Youngs' method

In the PLIC reconstruction method of Youngs [4] (see also [27, 28]), the interface is reconstructed in each cell as a plane $\mathbf{n} \cdot \mathbf{x} + C = 0$, in which the vector normal to the interface is obtained from the gradient of the volume fraction function, ∇F. A finite-difference formula is used to obtain the gradient of F at each corner of the cell. For example, the x component of the gradient at the corner $(i+\frac{1}{2}, j+\frac{1}{2}, k+\frac{1}{2})$ can be expressed as

$$(\nabla F)_x = \frac{1}{4\Delta x} \sum_{r=0}^{1} \sum_{s=0}^{1} (F_{i+1,j+r,k+s} - F_{i,j+r,k+s}) \tag{A1}$$

The gradient of F at the center of the cell (i, j, k) is obtained from the average of the gradients at the eight corners, and the normal to the interface at the corresponding cell is determined as $\mathbf{n} = \nabla F/|\nabla F|$. Gueyffier *et al.* [27] demonstrated that this method is only first-order accurate, although for relatively coarse grids the accuracy is comparable to that of a second-order least-squares algorithm.

Finally, the constant C of the interface plane is obtained from the value of F in the cell and enforcement of local volume conservation.

A.2. Extension of the ELVIRA method [*11, 12*] *to 3D*

Two different versions of the ELVIRA method in 3D were implemented in this work. The first version, which we refer to as the original ELVIRA-3D method, was implemented following a procedure similar to that described by Miller and Colella [13]. For each cell interface (i, j, k) to be reconstructed, a list of 24 or 48 candidate orientations, depending on the location of the center of mass of the VOF distribution around the considered cell, is obtained per coordinate direction following the indications given in [13]. This provides a total of $72, 96, 120$ or 144 candidate orientations. For each candidate orientation, a planar interface is reconstructed by enforcing local volume conservation in the considered cell. Then, the reconstructed interface is extended beyond the considered cell, and an associated error function is calculated as

$$E_{i,j,k} = \sum_{r=-p}^{p} \sum_{s=-p}^{p} \sum_{t=-p}^{p} (F_{i+r,j+s,k+t} - F^*_{i+r,j+s,k+t})^2 \tag{A2}$$

where $p=2$ (value that corresponds to a stencil of $5\times5\times5$ cells) and $F^*_{i+r,j+s,k+t}$ is the volume fraction calculated by truncating the boundaries of the cell $(i+r, j+s, k+t)$ with the extended planar interface. The candidate orientation that minimizes the error function defined in Equation (A2) is selected as the final interface orientation in the cell (i, j, k), and the position of the planar interface is finally obtained from volume enforcement.

In order to validate our implementation of the method, the same rotation test involving a notched block as resolved by Miller and Colella [13] was carried out, and a high degree of agreement was found.

In the second version of the ELVIRA method, which will be referred to as the improved ELVIRA-3D method, we have implemented in 3D, the stencil used in Equation (A2) is reduced to $3\times3\times3$ cells ($p=1$). An additional candidate is obtained from the average value of every 24 candidates per coordinate direction obtained as indicated in [13]. Therefore, this approach provides a total of $75, 100, 125$ or 150 candidate orientations. Despite the slight increase in the number of candidate orientations, note that the volume fraction error of Equation (A2) needs to be evaluated in only 27 cells per each candidate *versus* the 125 cells used in the first version of the method, which makes the CPU time consumed substantially lower.

ACKNOWLEDGEMENTS

The authors thank the reviewers for their helpful comments. This work was supported by the Spanish Ministerio de Educación y Ciencia under Grants DPI2004-08198 and PR2005-0296.

REFERENCES

1. Scardovelli R, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics* 1999; **31**:567–603.
2. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
3. Rudman M. Volume tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**:671–691.
4. Youngs DL. An interface tracking method for a 3D Eulerian hydrodynamics code. *Technical Report 44/92/35*, AWRE, 1984.
5. Puckett EG. A volume of fluid interface tracking algorithm with applications to computing shock wave rarefraction. *Proceedings of the 4th International Symposium on Computational Fluid Dynamics*, Davis, CA. University of California at Davis: Davis, 1991; 933–938.
6. Chorin AJ. Curvature and solidification. *Journal of Computational Physics* 1985; **58**:472–490.
7. Swartz B. The second-order sharpening of blurred smooth borders. *Mathematics of Computation* 1989; **52**(186):675–714.
8. Mosso SJ, Swartz BK, Kothe DB, Ferrell RC. A parallel, volume-tracking algorithm for unstructured meshes. *Parallel Computational Fluid Dynamics'96*, Italy, 1996.
9. López J, Hernández J, Gómez P, Faura F. A volume of fluid method based on multidimensional advection and spline interface reconstruction. *Journal of Computational Physics* 2004; **195**:718–742.
10. López J, Hernández J, Gómez P, Faura F. An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows. *Journal of Computational Physics* 2005; **208**:51–74.
11. Pilliod JE, Puckett EG. Second-order volume-of-fluid algorithms for tracking material interfaces. *Technical Report LBNL-40744*, Lawrence Berkeley National Laboratory, 1997.
12. Pilliod JE, Puckett EG. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics* 2004; **199**:465–502.
13. Miller GH, Colella P. A conservative three-dimensional Eulerian method for coupled solid–fluid shock capturing. *Journal of Computational Physics* 2002; **183**:26–82.
14. Liovic P, Rudman M, Liow J-L, Lakehal D, Kothe D. A 3D unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction. *Computers and Fluids* 2006; **35**:1011–1032.
15. Aulisa E, Manservisi S, Scardovelli R, Zaleski S. Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry. *Journal of Computational Physics* 2007; **225**:2301–2319.
16. Shin S, Juric D. Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity. *Journal of Computational Physics* 2002; **180**:427–470.
17. Scardovelli R, Zaleski S. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *Journal of Computational Physics* 2000; **164**:228–237.
18. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in Fortran*. Cambridge University Press: Cambridge, U.K., 1986.
19. Farin G. *Curves and Surfaces for Computer-Aided Geometric Design*, *A Practical Guide* (4th edn). Academic Press: San Diego, 1997.
20. Hernández J, López J, Gómez P, Zanzi C, Faura F. A new volume of fluid method in three dimensions—Part I: Multidimensional advection method with face-matched flux polyhedra. *International Journal for Numerical Methods in Fluids* 2008; DOI: 10.1002/fld.1776.
21. Francois MM, Cummins SJ, Dendy ED, Kothe DB, Sicilian JM, Williams MW. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics* 2006; **213**:141–173.
22. Cummins SJ, Francois MM, Kothe DB. Estimating curvature from volume fractions. *Computers and Structures* 2005; **83**:425–434.
23. Harvie DJE, Fletcher DF. A new volume of fluid advection algorithm: the stream scheme. *Journal of Computational Physics* 2000; **162**:1–32.

DOI: 10.1002/fld

24. LeVeque R. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis* 1996; **33**:627–665.
25. Enright D, Fedkiw R, Ferziger J, Mitchell I. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics* 2002; **183**:83–116.
26. Du J, Fix B, Glimm J, Jia X, Li X, Li Y, Wu L. A simple package for front tracking. *Journal of Computational Physics* 2006; **213**:613–628.
27. Gueyffier D, Li J, Nadim A, Scardovelli R, Zaleski S. Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics* 1999; **152**:423–456.
28. Biausser B, Guignard S, Marcer R, Fraunie P. 3D two phase flows numerical simulations by SL-VOF method. *International Journal for Numerical Methods in Fluids* 2004; **45**:581–604.

DOI: 10.1002/fld