Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 208 (2005) 51–74
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows

J. López [a], J. Hernández [b,*], P. Gómez [b], F. Faura [a]

[a] Departamento de Ingeniería de Materiales y Fabricación, ETSII, Universidad Politécnica de Cartagena, E-30202 Cartagena, Spain
[b] Departamento de Mecánica, ETSII, UNED, C/Juan del Rosal 12, E-28040 Madrid, Spain

Received 29 July 2004; received in revised form 24 January 2005; accepted 26 January 2005
Available online 13 April 2005

Abstract

A drawback common to PLIC schemes is the difficulty of reconstructing the interface when filaments of a thickness smaller than the cell size are present. In this work we present an improvement of the reconstruction method proposed by López et al. [A volume of fluid method based on multidimensional advection and spline interface reconstruction. J. Comput. Phys. 195 (2004) 718] for two-dimensional flows, which allows tracking fluid structures thinner than the cell size by allowing the interface to be represented in each cell by two non-contiguous linear segments. The method is based on using markers that are placed every time step at the mid-points of the reconstructed cell interface segments. The markers are used to improve the orientation and location of the interface segments, although the method essentially remains of a VOF type. Numerical tests show that a substantial improvement in accuracy is achieved over previous volume of fluid methods, and that the proposed method compares well even with a recent hybrid markers and volume of fluid method.

Keywords: Volume tracking; Volume of fluid; Interface reconstruction and advection methods; Thin fluid structures

1. Introduction

Volume of fluid methods, along with front tracking [29,40,42,43], level set [13,27,38,39] and phase field methods [8,21,20,44], are the most frequently used direct numerical simulation methods for tracking interfaces using a fixed-grid. Reviews of the historical development of volume tracking methods, which can be distinguished from each other by the features of the interface reconstruction algorithm and of the method

* Corresponding author. Tel.: +34 91 398 6424; fax: +34 91 398 6536.
E-mail address: jhernandez@ind.uned.es (J. Hernández).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.01.031

used for time integration of the volume fraction advection equation, can be found in [7,32–34,36]. Different improvements have been introduced recently in VOF methods to keep them competitive with the more recent methods mentioned above. Rider and Kithe [33] proposed a piecewise linear interface calculation (PLIC) method and a multidimensional unsplit time integration scheme, in which material volume fluxes are computed systematically using a set of simple geometric tasks. Harvie and Fletcher [18] proposed a new VOF advection algorithm (termed the Stream scheme) that uses a PLIC method coupled to a fully multidimensional cell face flux integration technique. Later, these authors [19] proposed another new multidimensional advection algorithm (the defined donating region (DDR) scheme), which conserves fluid volume rigorously without the need for a local redistribution algorithm, although with an accuracy comparable to that of the PLIC direction-split advection scheme of Youngs [45]. More recently, Aulisa et al. [2] presented a new advection method (termed the Π method), which exactly preserves mass for incompressible flows on a Cartesian mesh and shows a performance comparable to VOF multidimensional advection methods such as those of Rider and Kothe [33] or Harvie and Fletcher [18]. Another example of a multidimensional advection algorithm can be found in [28]. Recent enhancements of the second-order reconstruction method proposed by Chorin [9] (in which the normal and the local curvature of the interface are obtained from the osculating circle of a curve defined by a 3×3 array of volume fractions, and which is only applicable to orthogonal grids), which perform well on unstructured meshes, were proposed by Mosso et al. [26] and Garrioch and Baliga [14]. In a more recent work, Scardovelli and Zaleski [37] presented two new reconstruction algorithms based on a least-square fit technique (one of them with a variant that maintains the continuity of the interface at the boundary of adjacent cells), and a new mixed split Eulerian implicit–Lagrangian explicit (EI–LE) advection algorithm. López et al. [24] proposed an improved VOF method based on multidimensional advection using edge-matched flux polygons, and spline-based interface reconstruction (EMFPA-SIR). Recently, different hybrid methods that combine the best characteristics of VOF, level set or front tracking methods have been proposed [1,11,41].

In this paper, we present a further improvement of the PLIC-VOF method proposed by López et al. [24], which allows for filaments of a thickness smaller than the grid cell size to be resolved. It is well known that commonly used reconstruction algorithms, such as Youngs' algorithm [45], which calculate the interface normal from the volume fraction gradient, tend to produce ''blobby'' filaments when these become too thin to be resolved on the grid. In the EMFPA-SIR method proposed in [24], the orientation of the interface segment in each element is initially determined using Youngs' algorithm, and then corrected with the aid of a cubic spline interpolation through the center points of the interface segments. Although this correction was found to reduce 'flotsam' substantially, the errors in the calculation of the interface normal still lead the tracking method, in an attempt to locally enforce mass conservation, to incorrectly locate the fluid volume when the grid is not fine enough to resolve small fluid flow features. The method proposed in the present work reduces these errors by allowing the interface to be represented in each cell by two linear segments when fluid structures thinner than the cell size are present, as in Fig. 1, and by introducing a special procedure in cells where the interface curvature is too high. These special reconstruction procedures are based on using markers that are placed every time step at the mid-points of the reconstructed cell interface segments. The new locations of markers after advection are used to calculate the orientation and location of tentative interface segments at the new time step (following a procedure described in Section 2.1), whose mid-points are interpolated using splines in order to obtain the final interface segments (as described in Section 2.2). Despite the use of markers, we will see below that the method remains essentially a VOF-type method. A second improvement of the proposed method over previous methods lies in the use of a new simple analytical algorithm to locate the interface segments by enforcing local volume conservation (presented in Section 2.3), which can handle interface cells with two non-contiguous segments. The use of the interface advection method in cells with thin filaments is described in Section 3, coupling with the Navier–Stokes equations is treated in Section 4, and some of the results obtained in different tests are discussed in Section 5.

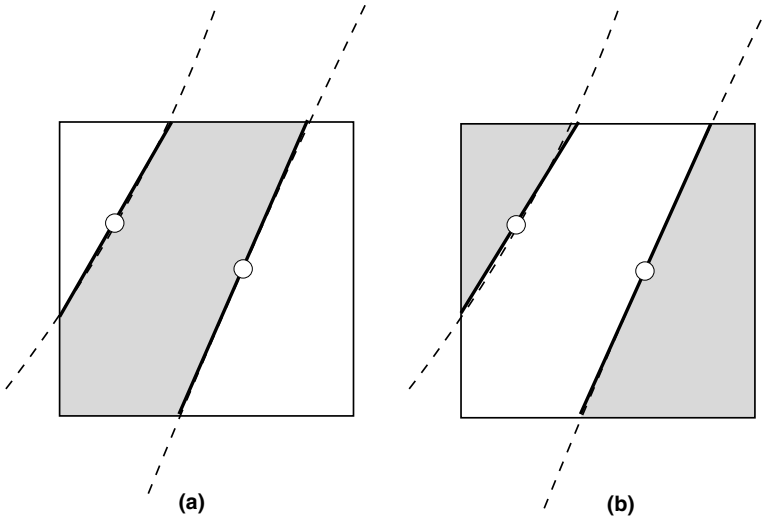

Fig. 1. Interface reconstruction using two linear segments in a cell with (a) only one portion of fluid and (b) two portions of fluid.

2. Interface reconstruction

The interface is approximated in each cell by a straight segment (as in any classical PLIC-VOF scheme), except in some cells where two non-contiguous segments will be used. Each segment is defined by the line $\boldsymbol{n} \cdot \boldsymbol{x} = C$, where $\boldsymbol{n}$ and $\boldsymbol{x}$ are the unit vector normal to the interface (pointing inward from the fluid region) and the position vector of a generic point on the line, respectively, and C is a constant. At every time step, the advection of the volume fraction distribution is first solved following the procedure described in Section 3, and then the interface is reconstructed at the interface cells using linear segments, according to a procedure that basically consists of two steps:

(1) Constructing tentative segments and making a list of ordered points along the interface consisting of the mid-points of these segments.
(2) Readjusting the tentative segments by using a spline interpolation through the ordered points in order to determine final segment orientations.

In both steps, the constant C of every tentative (step 1) or final (step 2) segment is obtained following the procedure described in Section 2.3 to enforce the local volume conservation constraint imposed by the volume fraction distribution.

2.1. Construction of tentative segments and a list of interpolation points

We propose two different procedures, one for the initial instant (Fig. 2(a)) and another for subsequent time steps (Fig. 2(b)).

2.1.1. Initial instant

Although any of the algorithms proposed by Ginzburg and Wittum [15] or López et al. [24], for example, could be used, we propose here a simpler procedure, similar to that used by Shin and Juric [40] for interface reconstruction in a front tracking method, which can easily be extended to three dimensions. The tentative

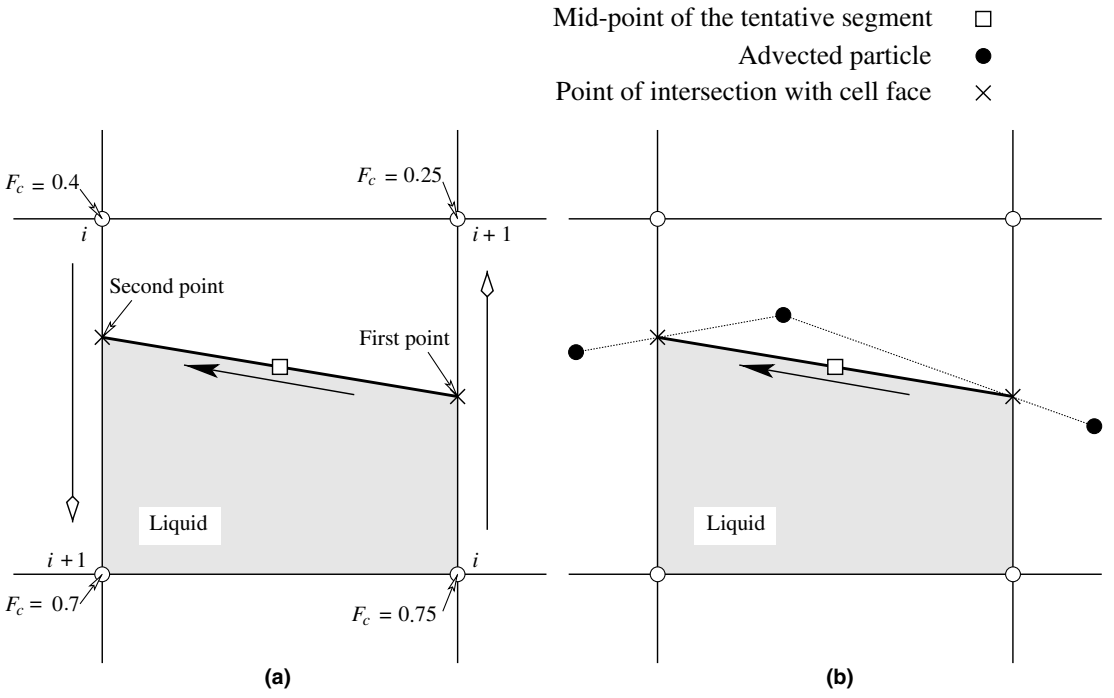

Fig. 2. Schematic representation of the construction of a tentative segment (before local volume enforcement) at (a) the initial instant and (b) subsequent time steps.

segments are constructed by sequentially joining the points in cell faces where the interpolated volume fraction is $F_m = 0.5$. The position vector, $\boldsymbol{x}_m$, of each of these points is determined by linear interpolation from the values of the volume fraction at the ends of the corresponding cell face, $F_{c_{i+1}}$ and F_{c_i}, which, in turn, are calculated by averaging the volume fractions of the corresponding surrounding cells (Fig. 2(a)):

$$\boldsymbol{x}_m = \boldsymbol{x}_{c_{i+1}} - \frac{F_{c_{i+1}} - 0.5}{F_{c_{i+1}} - F_{c_i}} \left(\boldsymbol{x}_{c_{i+1}} - \boldsymbol{x}_{c_i}\right), \tag{1}$$

$$F_c = \frac{\sum F_k V_k}{\sum V_k}, \tag{2}$$

where V_k is the volume (area) of cell k. The normal vector of each tentative segment can be obtained from

$$\boldsymbol{n} = \frac{(\boldsymbol{x}_{m_{j+1}} - \boldsymbol{x}_{m_j})^{\perp}}{|(\boldsymbol{x}_{m_{j+1}} - \boldsymbol{x}_{m_j})^{\perp}|}, \tag{3}$$

where $\boldsymbol{x}_{m_j}$ and $\boldsymbol{x}_{m_{j+1}}$ are the position vectors of the points at cell faces, ordered as indicated below, and $\perp$ denotes the perpendicular vector rotated anticlockwise (i.e., $(\boldsymbol{x}_{m_{j+1}} - \boldsymbol{x}_{m_j})^{\perp} = [-(y_{m_{j+1}} - y_{m_j}), (x_{m_{j+1}} - x_{m_j})]$).

Note that local volume conservation has not been enforced yet, which tends to produce interface location errors in regions of high curvature. Fig. 3(a) shows the locations of the mid-points of the tentative segments obtained in the reconstruction of a circle with a low dimensionless diameter, $d/\Delta x = 3.84$ (Δx is the cell size), using the procedure described above. It can be observed that the mid-points of the tentative segments significantly deviate from the exact interface location. In order to remedy this, Shin and Juric [40] reconstruct the interface to an optimum value of F_m (that approaches 0.5 with increasing resolution), for

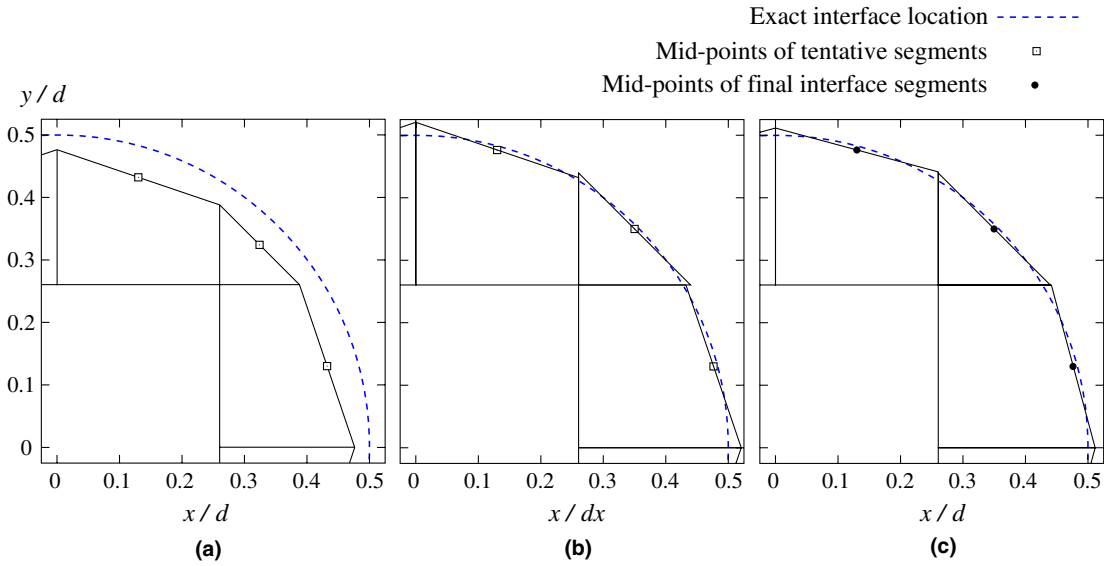

Fig. 3. Location of the interface segments obtained in the different steps of the reconstruction procedure of a circle with dimensionless diameter $d/\Delta x = 3.84$. Tentative reconstruction before (a) and after (b) enforcing local volume conservation, and (c) final spline-based reconstruction.

which the volume enclosed by the interface is the same as the volume before reconstruction. The main benefit of this approach is that the reconstruction procedure alone does not affect the global volume. The effect on local volume conservation was found to be negligible so long as high curvature regions of the interface are reasonably resolved. In this work, we follow a different approach, strictly conservative, based on modifying the position of every tentative segment following the procedure described in Section 2.3, although a method such as Brent's (see for example [30]) could also be used.

The interpolation points will be the mid-points of the tentative segments obtained after volume conservation enforcement, and their order in the list is chosen so that cells with $F = 1$ remain on the left side as we move forward along the interface (Fig. 2(a)). When possible, the sequence starts in a cell adjacent to a boundary of the domain that allows this order to be followed. Otherwise, an arbitrary interface cell is selected to start. In the first cell, the first point of the segment will be that at the edge between corners i and $i + 1$ where $F_{c_i} \geqslant 0.5$ and $F_{c_{i+1}} < 0.5$, provided that these are ordered anticlockwise. The next cell where the interface segment is to be constructed will include the edge of the previous cell that contains the second point of its interface segment (obviously, the first point of the new interface segment will coincide with the second point of the segment of the previous cell). This procedure is successively applied to other interface cells until the first cell of the sequence or a domain boundary is reached. The construction of a new sequence will begin if more interface cells still remain.

2.1.2. Subsequent time steps

At every time step $n + 1$, the procedure begins (except for certain situations discussed below) by placing marker particles at the mid-points of the interface segments constructed in the previous time step, whose position vectors are denoted by $\boldsymbol{x}_{\mathrm{p}}^{i}$. These particles are advected with the local fluid velocity, $\boldsymbol{v}_{\mathrm{p}}$, which is obtained from a weighted average of the velocities at the centers of cell faces, $\boldsymbol{v}_i$:

$$\boldsymbol{v}_{\mathrm{p}} = \frac{\sum_i \frac{1}{d_i}\boldsymbol{v}_i}{\sum_i \frac{1}{d_i}}, \tag{4}$$

where d_i is the distance from the particle to the face center i. The position vector of the new location of each particle is computed as

$$\boldsymbol{x}_{\mathrm{p}} = \boldsymbol{x}_{\mathrm{p}}^{i} + \boldsymbol{v}_{\mathrm{p}}^{*}\Delta t. \tag{5}$$

In tests with a prescribed velocity field, $\boldsymbol{v}_{\mathrm{p}}^{*} = (\boldsymbol{v}_{\mathrm{p}}^{n} + \boldsymbol{v}_{\mathrm{p}}^{n+1})/2$, where $\boldsymbol{v}_{\mathrm{p}}^{n}$ and $\boldsymbol{v}_{\mathrm{p}}^{n+1}$ are the velocity vectors at the initial location of the particle at instants t^n and $t^{n+1} = t^n + \Delta t$, whereas when the motion equations are being solved, $\boldsymbol{v}_{\mathrm{p}}^{*}$ will correspond to the velocity vector obtained after the MAC projection described in Section 4 at the initial location of the particle. It should be mentioned that particles will be used only in the procedure to compute the orientations of the interface segments, which are later readjusted to enforce volume conservation, and so a higher order of accuracy is not crucial. As in front tracking and marker methods, the use of particles allows us to maintain the interface integrity in regions with fluid filaments of thickness smaller than cell size, as in Fig. 4. The locations of the advected particles could be used directly to obtain the spline

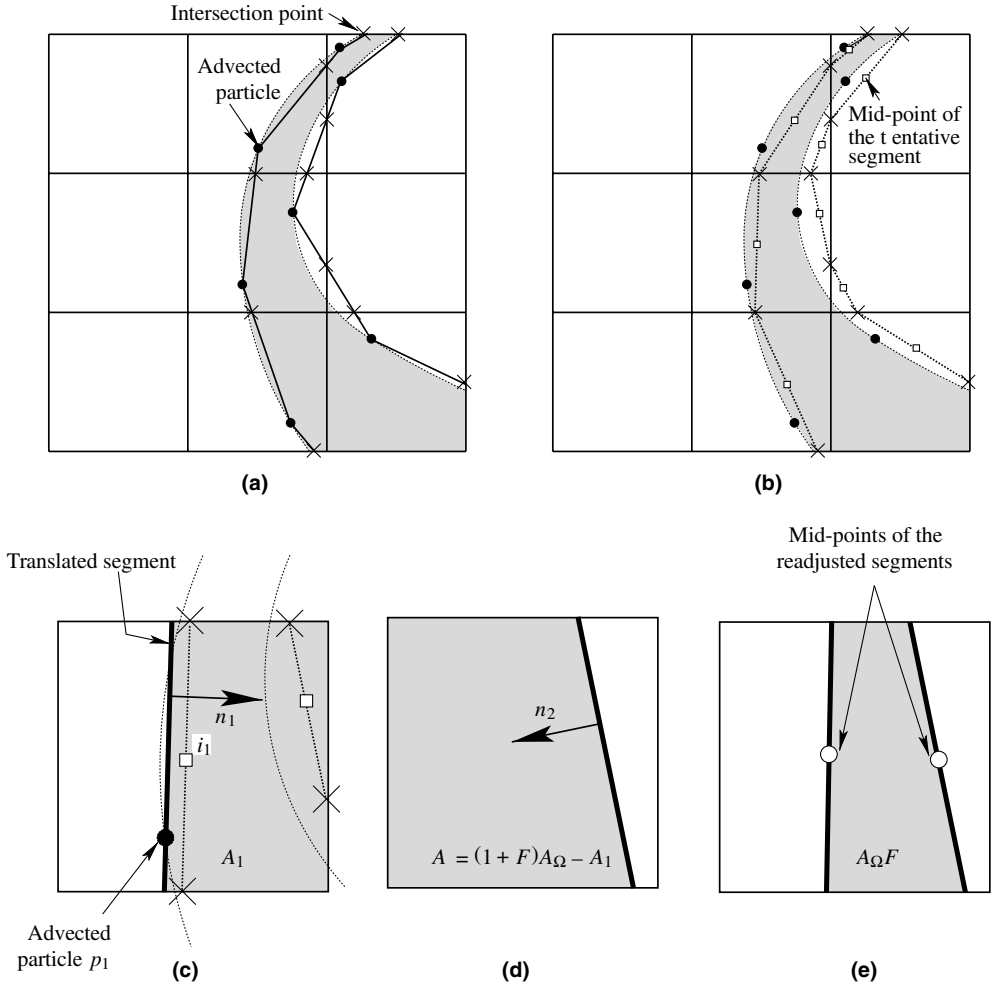

Fig. 4. Schematic representation of a thin filament of fluid. (a) Advected particles and intersection points between lines linking particles and cell faces. (b) Mid-points of the tentative segments linking the intersection points. (c) Readjustment of the first tentative segment. (d) Readjusted second segment after enforcing local volume conservation. (e) Final reconstructed interface segments.

interpolation curve, from which the orientation of the interface segment at every interface cell could be determined. However, in order to prevent particles from becoming too close or too far from each other when interface deformation is strong, and to obtain a better estimation of the interface segment slope (as is shown in the example of Fig. 5, when the advected particle lies far from the mid-point of the interface segment, the normal of the spline curve at this particle may produce an incorrect orientation of the cell segment), we propose instead the following procedure.

The locations of the advected particles (denoted by black circles in Fig. 4(a)) are linked with straight lines, represented by $\boldsymbol{n}_{\mathrm{p}} \cdot x = C_{\mathrm{p}}$. The intersection points between these lines and the grid cell faces determine one or more tentative linear segments in each interface cell (their mid-points are denoted by open squares in Fig. 4(b)). The position vector of each intersection point, which is denoted by a cross symbol in Fig. 4, can be obtained from

$$\boldsymbol{x}_m = \frac{\boldsymbol{n}_{\mathrm{f}}^{\perp} C_{\mathrm{p}} - \boldsymbol{n}_{\mathrm{p}}^{\perp} C_{\mathrm{f}}}{\boldsymbol{n}_{\mathrm{f}}^{\perp} \cdot \boldsymbol{n}_{\mathrm{p}}}, \tag{6}$$

where $\boldsymbol{n}_{\mathrm{f}}$ and C_{f} define the corresponding cell face, $\boldsymbol{n}_{\mathrm{f}} \cdot x = C_{\mathrm{f}}$. The normal vector of every tentative segment is then determined, as in the previous section, from Eq. (3), and its location is finally readjusted by enforcing volume conservation using the method described in Section 2.3. The mid-points of these readjusted tentative segments will be used, instead of the advected particles, as interpolation points, which will be ordered in accordance with the order of the mid-points of the segments in the previous time step where the corresponding particles were placed.

Pinch-off and reconnection are assumed to occur when the distance between two interface segments becomes lower than a given value. When this situation is detected, the particles corresponding to these segments are reassigned to the appropriate interpolation list. Fig. 6 shows an example of the coalescence of two

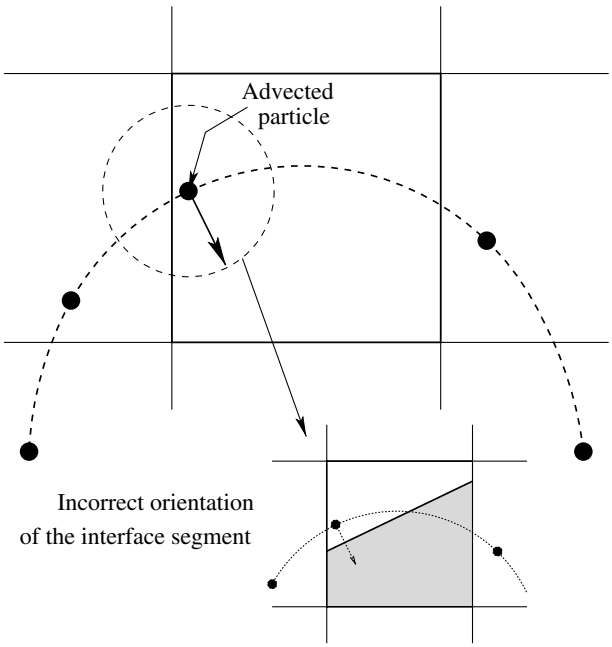

Fig. 5. Example of a case in which the use of an advected particle as spline interpolation point to obtain the normal of the interface in the cell produces an incorrect segment orientation.

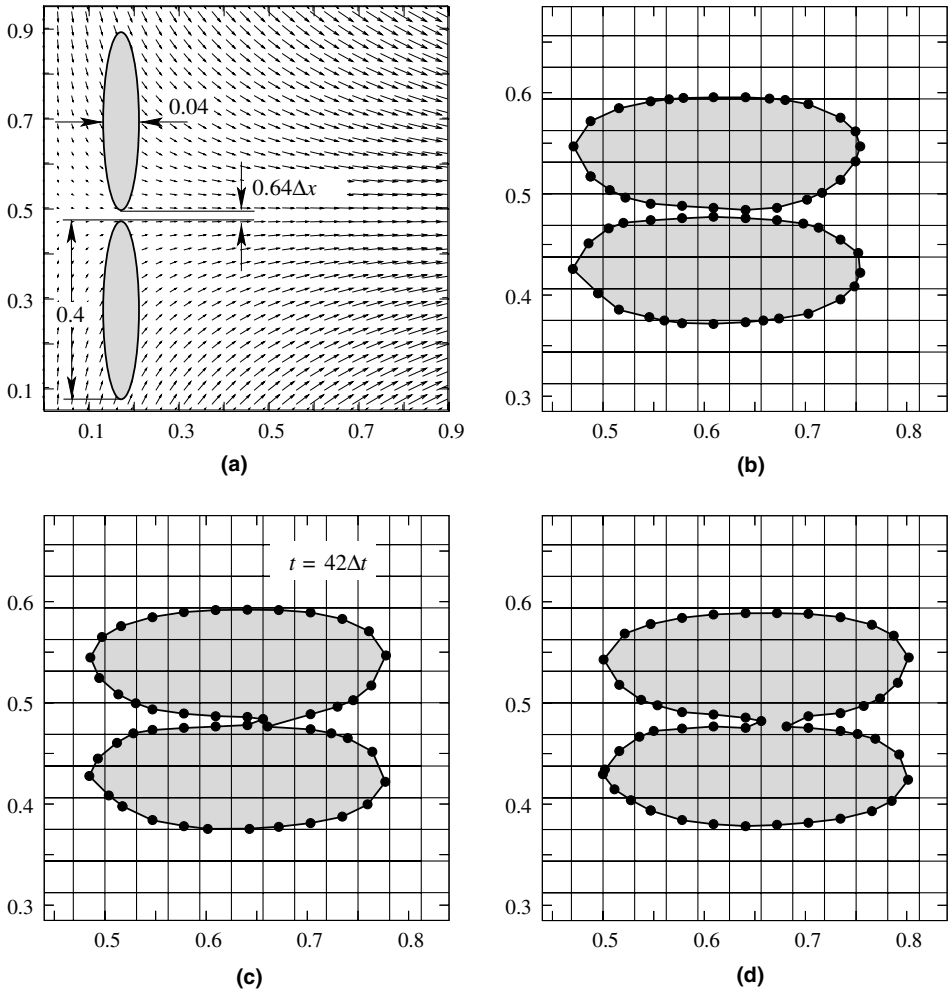

Fig. 6. Example of the coalescence of two fluid blobs of elliptical shape, initially separated by a distance of $0.64\Delta x$ ($\Delta x = \Delta t = 1/32$), in a velocity field given by $v_x = x$, $v_y = 0.485 - y$. (a) Initial instant; (b) last instant before coalescence; (c) coalescence instant ($t = 42\Delta t$); (d) first instant after coalescence.

fluid blobs of elliptical shape, initially separated by a distance of $0.64\Delta x$ ($\Delta x = \Delta t = 1/32$) in a domain of 1×1, which are advected with the velocity field $v_x = x$, $v_y = 0.485 - y$. The interface was reconnected when the distance between two particles in a cell reached a value lower than $0.25\Delta x$, which occurred at the instant $42\Delta t$ (Fig. 6(c)), for which the exact solution gives a shortest distance between the two deformed ellipses of $0.172\Delta x$.

In cells that are not crossed by any straight line linking the advected particles, but in which $0 < F < 1$, the interface is reconstructed with a single segment using Youngs' method [45]. This segment is not further adjusted, its mid-point is not included in the list of points to be interpolated, and no particle to be advected in the next time step is placed within the cell.

In cells where $F = 1$ or $F = 0$, and which are crossed by any line linking advected particles (regardless of whether the cell contains any particle or not), the interface is not reconstructed, but the mid-points of the segments determined by the crossing points of the lines with the cell faces are included in the list of points to be interpolated. A particle to be advected in the next time step will be placed on each of these points.

In cells crossed by more than one straight line linking advected particles, for which the interface will be reconstructed using two linear segments, the method is applied in a similar way as explained above, although the adjustment of the tentative segments to enforce local volume conservation requires a special procedure that will be explained in Section 2.3.

2.2. Spline interpolation

As already mentioned, the final orientation of every interface segment will be given by the slope of the spline interpolation curve at the mid-point of the tentative segment (with some exceptions discussed below). To determine this curve, the coordinates of the interpolation points are parametrized as $[x_j(s_j), y_j(s_j)]$, where s is an approximation to the length of the curve from the initial point, which, for the point j, is defined as

$$s_j = \sum_{i=1}^{j-1} |\boldsymbol{x}_{i+1} - \boldsymbol{x}_i|. \tag{7}$$

The parametric curve $x = x(s)$, $y = y(s)$ is constructed using a natural cubic spline interpolation (see for example [30,12]), and the normal to each segment of the interface is obtained as

$$\boldsymbol{n}_j = \frac{[-y'(s_j), x'(s_j)]}{[x'^2(s_j) + y'^2(s_j)]^{1/2}}, \tag{8}$$

where central differences are used for the derivatives. At the extreme cells we take

$$\boldsymbol{n} = \frac{(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)^{\perp}}{|(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i)^{\perp}|}, \tag{9}$$

where i denotes the first or the last but one segment mid-point of the sequence. If the curve is closed, the first and last points of the curve coincide in the same cell, in which case we take the normalized average of the two extreme values obtained from Eq. (9).

The constant C for every final interface segment is determined by enforcing volume conservation, to which aim we can use, among other methods, the iterative method of Brent, the analytical method proposed by Gueyffier et al. [17] for rectangular grids, or the new method presented in Section 2.3, which can be applied to irregular grids with quadrilateral cells.

There are some particular situations in which special procedures must be followed: for example, when the two intersection points between lines linking particles and the faces of a given cell lie at the same face, as shown in Fig. 7(a), and the volume fraction of the cell is very large or small. In this case, spline interpolation correction followed by volume conservation enforcement may give rise to an inappropriate location of the mid-point of the final segment, as shown in Fig. 7(b), as a result of a small difference between the slopes of the tentative segment (parallel to the cell face) and the tangent to the interpolation curve at the mid-point of the tentative segment. In this and other similar cases (which we define using the criterion that the distance between the mid-points of the tentative segment and of the readjusted segment exceeds a given value, typically $\Delta x/4$) the tentative segment is not readjusted after spline interpolation, and only the volume conservation constraint is enforced.

Another particular case considered, which is depicted in Fig. 8, occurs when the two straight lines linking three consecutive particles form too small an angle. This type of situation, which is found, for example, at the tail of the fluid filament formed in the single vortex test presented in Section 5, tends to increase the interface location error, particularly when particle b in Fig. 8 is far from the mid-points of the contiguous segments, a circumstance which will depend on the location of particles relative to grid cells. The only differences in handling these situations are the following: (1) The point where particle b is located is included in the list of points to be interpolated by splines; (2) the tentative segment in the cell is not readjusted after

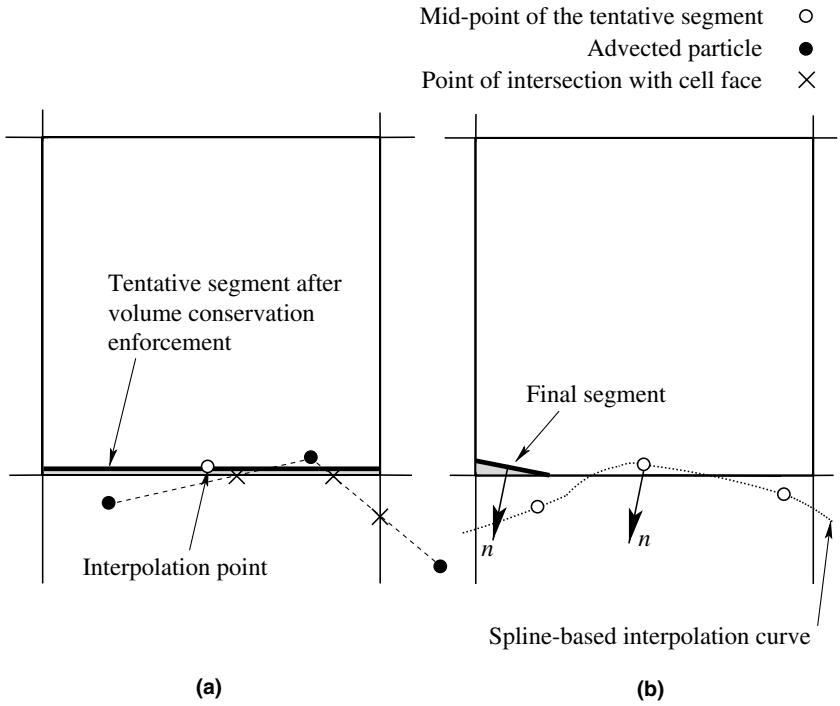

Fig. 7. Example of a cell with a low F value, in which the lines linking the advected particles intersect the same cell face. (a) Tentative segment reconstruction. (b) Inappropriate location of the final interface segment and of the particle to be tracked in the next time step, after spline interpolation.

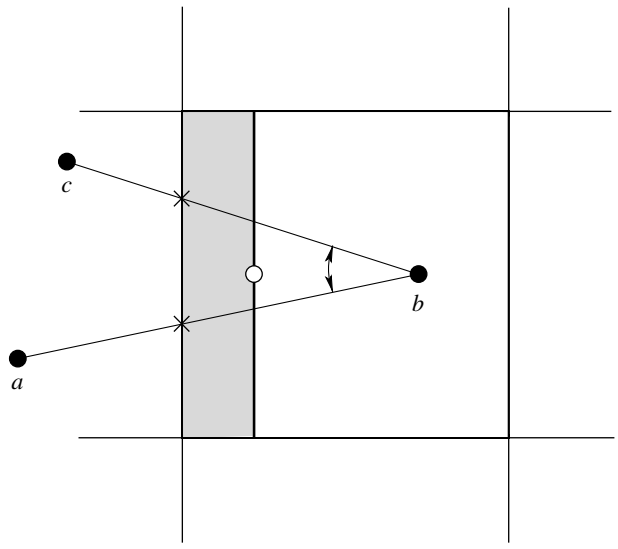

Fig. 8. Case in which two straight lines linking consecutive particles form too small an angle.

spline interpolation, and (3) the particle b, instead of the mid-point of the reconstructed interface segment, will be tracked in the next time step.

2.3. Enforcement of volume conservation

Local volume conservation, which is the main reconstruction constraint, must be enforced taking into account the new volume fraction distribution, which in this work is obtained following the edge-matched flux polygon advection (EMFPA) procedure described in Section 3. One method commonly used in PLIC-VOF schemes to determine the interface segment position at each cell is the iterative method of Brent, which is a root-finding algorithm based on a combination of a root bracketing, interval bisection, and inverse quadratic interpolation (see for example [30]). In this section, a new analytical method is proposed, which can be applied to irregular grids of quadrilateral cells. The CPU time required for this method typically varies between 50% and 80% of the time taken by Brent's method, depending on the orientation of the segment in the cell, the volume fraction and the number of iterations used in Brent's method. We will first consider the case in which only a single segment is required to represent the interface in a cell.

2.3.1. Cells with only one interface segment

The most general case is shown in Fig. 9(a). The position vector of the intersection point between the cell face i, represented by $\boldsymbol{n}_{\mathrm{f}i} \cdot \boldsymbol{x} = C_{\mathrm{f}i}$, and the approximated interface, $\boldsymbol{n} \cdot \boldsymbol{x} = C$, can be expressed as

$$\boldsymbol{x}_{\mathrm{I}i} = \frac{\boldsymbol{n}_{\mathrm{f}i}^{\perp} C - \boldsymbol{n}^{\perp} C_{\mathrm{f}i}}{\boldsymbol{n}_{\mathrm{f}i}^{\perp} \cdot \boldsymbol{n}}, \tag{10}$$

where $\boldsymbol{n}_{\mathrm{f}i}^{\perp}$ and $\boldsymbol{n}^{\perp}$ are, respectively, the vectors perpendicular (rotated anticlockwise) to the normals $\boldsymbol{n}_{\mathrm{f}i}$ and $\boldsymbol{n}$ (we recall that the normal $\boldsymbol{n}$ was obtained from Eq. (3) in the tentative reconstruction described in Section 2.1, and from Eq. (8) in the final spline-based reconstruction of Section 2.2). Taking into account that the area of a polygon of k vertices can be expressed as

$$A = \frac{1}{2}\left(\boldsymbol{x}_k^{\perp} \cdot \boldsymbol{x}_1 + \sum_{i=1}^{k-1} \boldsymbol{x}_i^{\perp} \cdot \boldsymbol{x}_{i+1}\right), \tag{11}$$

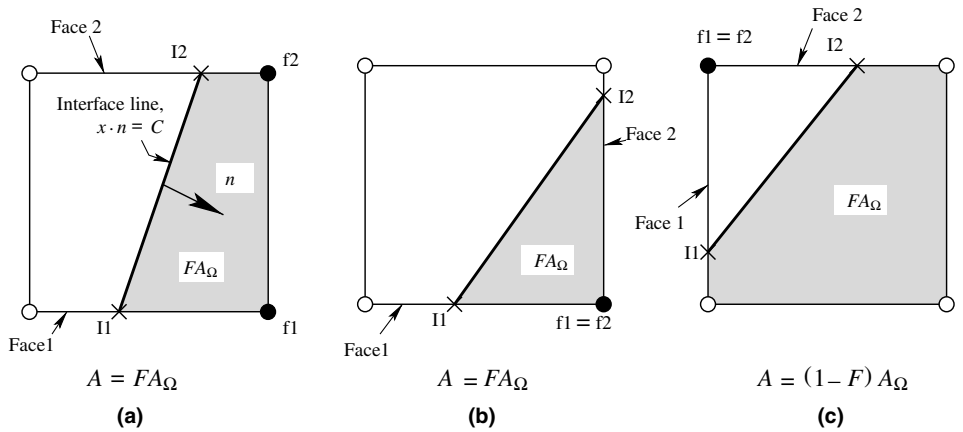

Fig. 9. Different situations considered in the analytical method proposed to enforce local volume conservation.

imposing the condition that $A = FA_\Omega$, where A_Ω is the cell area and A is the area of the polygon f1-f2-I2-I1 in Fig. 9(a), and rearranging terms, the constant C may be obtained from

$$\alpha C^2 + \beta C + \gamma = 2A, \tag{12}$$

where

$$\alpha = \frac{\boldsymbol{n}_{f2}^{\perp} \cdot \boldsymbol{n}_{f1}}{(\boldsymbol{n}_{f1}^{\perp} \cdot \boldsymbol{n})(\boldsymbol{n}_{f2}^{\perp} \cdot \boldsymbol{n})}, \tag{13}$$

$$\beta = \frac{2C_{f2}}{\boldsymbol{n}_{f2}^{\perp} \cdot \boldsymbol{n}} - \frac{2C_{f1}}{\boldsymbol{n}_{f1}^{\perp} \cdot \boldsymbol{n}}, \tag{14}$$

and

$$\gamma = \boldsymbol{x}_{f1}^{\perp} \cdot \boldsymbol{x}_{f2} + \frac{\boldsymbol{n} \cdot \boldsymbol{x}_{f1}}{\boldsymbol{n}_{f1}^{\perp} \cdot \boldsymbol{n}} C_{f1} - \frac{\boldsymbol{n} \cdot \boldsymbol{x}_{f2}}{\boldsymbol{n}_{f2}^{\perp} \cdot \boldsymbol{n}} C_{f2}. \tag{15}$$

Subscripts 1 and 2 refer to the cell faces intersected by the interface segment, and $\boldsymbol{x}_{f1}$ and $\boldsymbol{x}_{f2}$ are the position vectors of the nodes of the intersected faces that are within the fluid.

In the case of Fig. 9(b), we only need to make $\boldsymbol{x}_{f1} = \boldsymbol{x}_{f2}$ in Eq. (15). The case of Fig. 9(c), in which three nodes of the cell are within the fluid, becomes equivalent to that of Fig. 9(b) if the nodes f1 and f2 are assumed to coincide with the cell node that is outside the fluid and Eq. (12) is solved making $A = (1 - F)A_\Omega$.

As an example, Figs. 3(b) and (c) show the tentative and final segments, respectively, obtained in the reconstruction of a circle after enforcement of local volume conservation, and allow one to observe the qualitative improvements introduced by the correction of the tentative segments and the readjustment based on spline interpolation and subsequent volume conservation enforcement.

2.3.2. Cells with two interface segments

In cells in which the interface is represented by two (non-contiguous) segments, local volume conservation must be enforced ensuring the integrity and accuracy in the location of the fluid filament. To determine the first segment in one of these cells, step 1 (see the beginning of Section 2) simply involves the construction of the initial tentative segment (according to the procedures described in Section 2.1) and its subsequent translation parallel to itself, so as to pass through the corresponding advected particle, p_1, as indicated in Fig. 4(c). The constant in the equation of this segment is thus obtained as $C_1 = \boldsymbol{n}_1 \cdot \boldsymbol{x}_{p_1}$, where $\boldsymbol{n}_1$ is calculated from Eq. (3). The mid-point of this segment is added to the corresponding list of interpolation points. In step 2, the orientation of the interface segment is readjusted after spline interpolation, obtaining $\boldsymbol{n}_1$ from Eq. (8) (this readjustment is omitted in Fig. 4), but volume conservation is not enforced in this step either. If no advected particle lies within the cell, the corresponding mid-point of the original tentative segment, i_1, is used to obtain the position of the segment ($C_1 = \boldsymbol{n}_1 \cdot \boldsymbol{x}_{i_1}$).

Once the location of the first segment has been obtained, local volume conservation is enforced by determining the location of the second segment in the cell in accordance with the following procedure, taking into account that the area of fluid in the cell delimited by the two segments must obviously be equal to FA_Ω. If there is only one portion of fluid in the cell, as in Fig. 1(a), the constant C_2 in the segment equation is obtained from Eq. (12), making $A = (1 + F)A_\Omega - A_1$ (colored area in Fig. 4(d)), with A_1 being the area of the cell region delimited by the first segment to which the normal $\boldsymbol{n}_1$ points (colored area in Fig. 4(c)). The normal $\boldsymbol{n} = \boldsymbol{n}_2$ of the segment in Eqs. (13)–(15) is obtained from Eq. (3) in the tentative reconstruction (step 1) or from Eq. (8) in the spline-based adjustment (step 2). If there are two portions of fluid in the cell, as in Fig. 1(b), the procedure is similar, but making $A = FA_\Omega - A_1$ in Eq. (12).

When $A_1/A_\Omega < F$ or $A_1/A_\Omega > F$ in cells with one or two portions of fluid, respectively, there is no position of the second segment that allows local volume conservation to be enforced. In such cases, the interface is reconstructed with a single segment.

Although thin fluid structures are in practice created during the simulation, and are not usually present at the initial instant, the construction of the interface segments in cells containing a filament at this instant could be achieved by linking the intersection points between the initial interface shape and cell faces, and then modifying the location of one of the segments by enforcing volume conservation, as explained above for subsequent time steps. A particle to be advected in the new time step would be placed at the mid-point of each segment.

3. Interface advection in cells with thin filaments

Once the interface has been reconstructed in all the interface cells at the previous time step, the advection of fluid is solved for the next time step by geometric considerations, as in conventional volume of fluid methods. In this section, we briefly describe how to use advection algorithms in cells with two interface segments.

The first step is to construct flux polygons at the cell faces such as those shown in Figs. 10(a) and (b). Although the type of polygons in these figures is that proposed by López et al. [24], which was found to behave well in reducing advection errors, the method described here can be applied to other types of polygons, such as that proposed by Rider and Kothe [33]. The area of fluid advected through the cell faces considered in Figs. 10(a) and (b) is determined from the areas delimited by the flux polygon and the interface segments. If the flux polygon is cut by two segments within the cell, the fluxed area will be $A_{\omega 1} + A_{\omega 2} - A_\omega$ (where A_ω is the area of the flux polygon, and $A_{\omega 1}$ and $A_{\omega 2}$ are the areas of the regions in the flux polygon delimited by the first and second interface segments, respectively, from where the normal to the segment points) or $A_{\omega 1} + A_{\omega 2}$, depending on whether there are one or two portions of fluid within the polygon.

In [24], we discussed that the suppression of polygon over/underlapping situations and the increase in accuracy derived from the reduction of discretization errors in the EMFPA algorithm reduce interface advection errors and the need for a redistribution algorithm to suppress F values outside the range $0 \leqslant F \leqslant 1$. As shown in [24], the fact that the EMFPA algorithm does not require correction of the flux polygons at the cell corners, as proposed by Rider and Kothe [33], makes any subsequent divergence correction unnecessary in tests with a prescribed solenoidal velocity field. In tests where the velocity field is calculated from the Navier–Stokes equations (as in the Rayleigh–Taylor test of Section 5.2), an advection equation of the form

$$\frac{\partial f}{\partial t} + \nabla \cdot (\boldsymbol{v} f) - f \nabla \cdot \boldsymbol{v} = 0 \tag{16}$$

(F is the discretized version of the function f) is considered in order to take into account the divergence error present in the velocity field. Rider and Kothe [33] showed that by taking into account the last term of Eq. (16) a more precise local and global conservation of F can be attained, along with a reduction (although not complete) in the appearance of cells with volume fractions greater than 1 or lower than 0.

It is well known that the extension to three-dimensions of PLIC-VOF methods is not a simple task, especially because of the complex geometrical operations involved. Good examples of successful implementations of previous PLIC-VOF methods in 3D can be found in [17,22,23]. The extension of the method presented here does not represent an appreciable increase in complexity with respect to that proposed in [24], where the outlook of the method for 3D was briefly discussed.

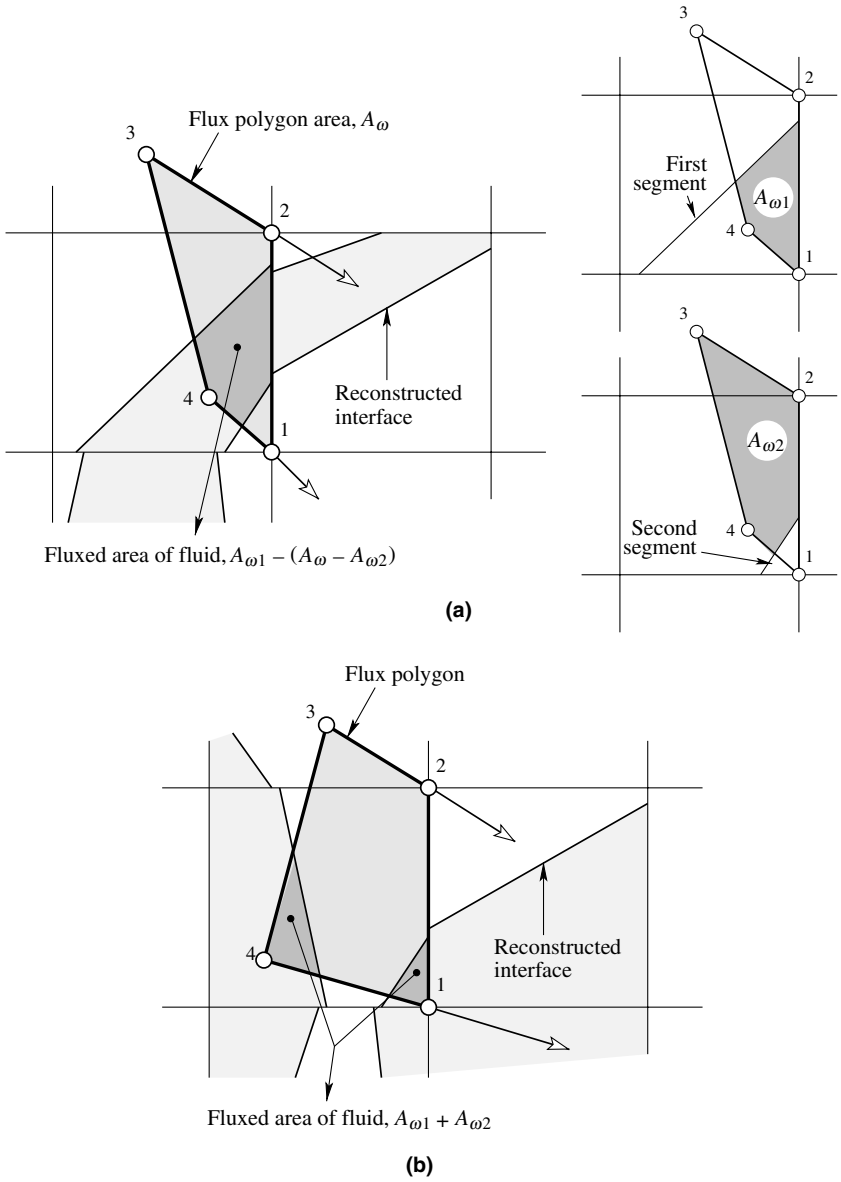

Fig. 10. Volume of fluid advected through a cell face in cells with two interface segments and one (a) and two (b) portions of fluid.

4. Coupling with the Navier–Stokes equations

The interface reconstruction and advection algorithms described above were solved in combination with the incompressible Navier–Stokes equations to obtain some of the results presented in the next section. For this purpose, the proposed algorithms were implemented into a code that solves the Navier–Stokes equations on both sides of the interface. The code, which we described in detail in [16], is based on the approximate projection method proposed by Almgren et al. [4], and generalized to variable density flows by

Puckett et al. [32]. In the first step, an intermediate velocity field $\boldsymbol{v}^{*,n+1}$ is computed from the momentum equation, ignoring the incompressibility constraint, approximating the convective term at $t^{n+1/2}$ and treating the pressure gradient as a source term, which is evaluated at $t^{n-1/2}$. In the projection step, the intermediate vector field $\boldsymbol{v}^{*,n+1}/\Delta t + (\boldsymbol{G}p^{n-1/2})/\rho^{n+1/2}$, where $\boldsymbol{G}$ is a discrete gradient operator, is projected to the space of divergence-free vector fields to obtain the velocity $\boldsymbol{v}^{n+1}$ and pressure $p^{n+1/2}$.

The density and viscosity in each computational cell are calculated from

$$\rho = \rho_1 + (\rho_2 - \rho_1)\widetilde{F}, \quad \mu = \mu_1 + (\mu_2 - \mu_1)\widetilde{F}, \tag{17}$$

where subscripts 1 and 2 denote fluids at both sides of the interface and $\widetilde{F}$ is a continuous and smoothed indicator function obtained from the volume fraction distribution by using a cubic kernel [25,35].

The discretization procedures applied to the equations derived from the projection algorithm are similar to those described by Puckett et al. [32]. A Cartesian computational grid is used, in which the discrete values of the velocity components, indicator function and fluid properties are located at the cell centers and the pressure values at the cell corners. For discretization of the convective term in the momentum equation, we used a method similar to the predictor–corrector methods used by Puckett et al. [32], among others, which is based on the unsplit Godunov method introduced by Colella [10]. In the predictor step, the velocity is extrapolated to the cell faces at $t^{n+1/2}$ using a second-order Taylor series expansion in space and time. Before constructing the convective derivatives, a MAC projection [5] is applied to make the normal velocity components at the cell faces divergence-free. The algorithms for advection of the volume fraction distribution and interface reconstruction are applied every time step after the MAC projection, using the extrapolated velocities to the cell faces at the intermediate time $t^{n+1/2}$.

5. Discussion of results

5.1. Tests with prescribed velocity field

The efficiency and accuracy of the proposed method were assessed by solving different test problems. In this section, we present some of the results obtained in two tests that allow us to investigate the ability of the proposed reconstruction method to track fluid filaments with a thickness smaller than the cell size. As we mention below, for fluid structures sufficiently thicker than the cell size, the proposed method tends to produce similar results to those of the method proposed in [24], where several numerical tests were presented for these cases.

The first test consists in the advection of a rectangular area of fluid 0.075 wide by 0.15 high, centered in a 1×1 domain where the velocity is horizontal and varies linearly in the vertical direction: $v_x = 2(y - 0.5)$ (the origin of y is at the bottom of the domain). A CFL number based on the maximum velocity in the domain equal to one was used in all computations. An interesting feature of this test is that, when using advection algorithms such as EMFPA [24], the contribution of the advection step to the error is negligible, and so the total error is solely due to reconstruction. Fig. 11 shows the area occupied by the fluid in every interface cell at four instants, obtained using four different reconstruction algorithms and a relatively coarse grid of 32×32 cells. The thickness of the filament given by the exact solution varies from $\delta = 1.07\Delta x$ at $t = 1$ to $\delta = 0.47\Delta x$ at $t = 2.5$ ($\delta/\Delta x = 2.4$ at the initial instant). It can be observed from Fig. 11 that, particularly when $\delta < \Delta x$, the proposed reconstruction method preserves the integrity of the filament better than the spline reconstruction method of López et al. [24], which in turn substantially improves the results obtained using our implementation of Youngs' and Puckett's methods. The improvement in the interface location accuracy can be observed quantitatively from the values indicated in Fig. 11 of the L_1 error norm, defined as

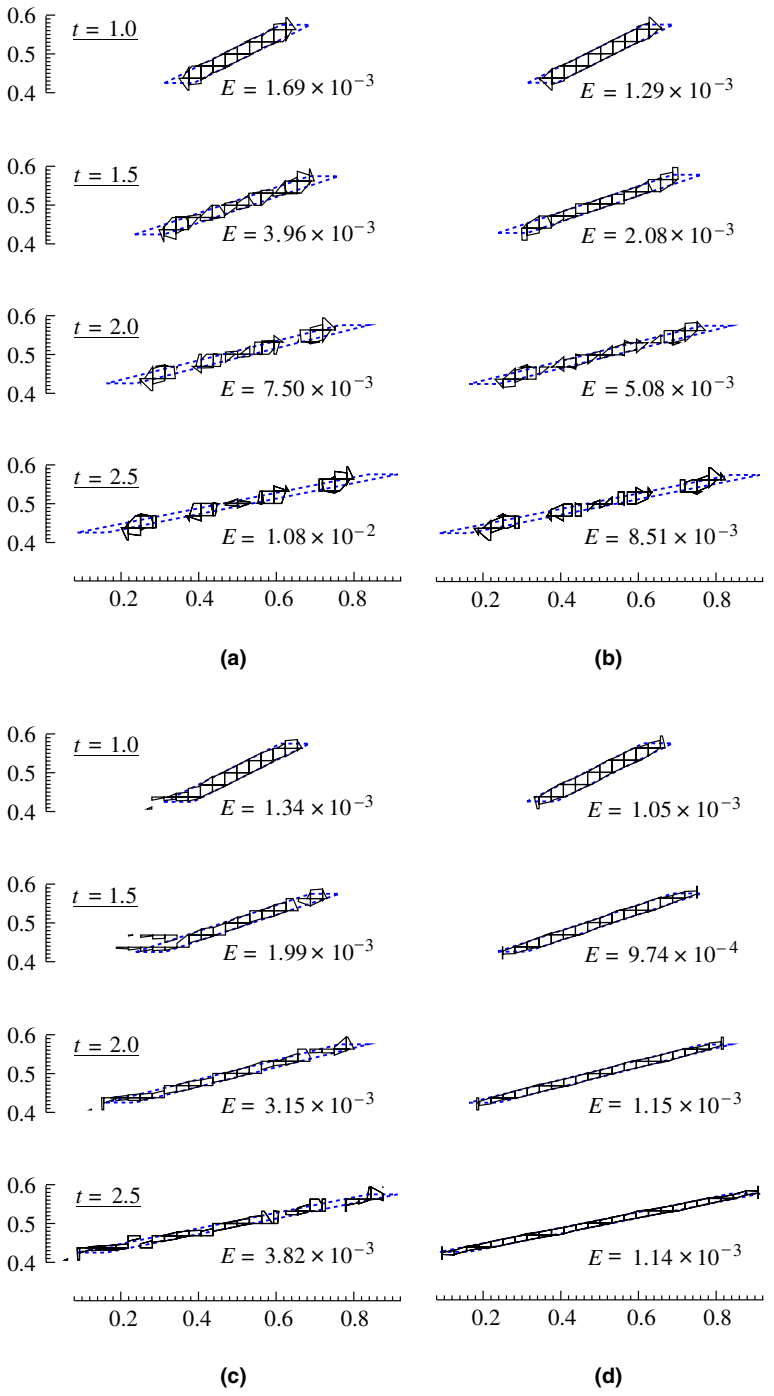

Fig. 11. Results obtained in the deformation of a rectangular area of fluid in a velocity field with uniform strain rate, at four different instants, using a grid of 32×32 cells, the EMFPA advection algorithm and different reconstruction methods. (a) Youngs; (b) Puckett; (c) SIR; (d) proposed method.

$$E = \sum_{i,j} A_{\Omega}^{(i,j)} \left| F^{(i,j)} - F_{\mathrm{e}}^{(i,j)} \right|, \tag{18}$$

where $A_{\Omega}^{(i,j)}$ is the area of the cell (i,j), and $F^{(i,j)}$ and $F_{\mathrm{e}}^{(i,j)}$ are, respectively, the calculated and exact volume fractions corresponding to a given instant.

The second test corresponds to the simulation of the time-reversed single vortex flow for conditions taken from the work of Rider and Kothe [33]. A circle of fluid of radius 0.15, initially centered at $(0.5, 0.75)$ in a unit square computational domain, is deformed in the solenoidal velocity field defined by the stream function $\Psi(x,y) = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y)\cos(\pi t/T)$. The circle of fluid undergoes the largest deformation at $t = T/2$ and returns to its initial state at $t = T$. The interface location error is estimated with the L_1 norm defined in Eq. (18). The order of convergence can be determined from

$$\mathcal{O} = \frac{\ln(E_{2n_x}/E_{n_x})}{\ln(1/2)}, \tag{19}$$

where E_{n_x} and E_{2n_x} are the errors obtained using two different grids with n_x and $2n_x$ cells along one direction. All the results presented here were obtained using a CFL parameter (based on the maximum velocity component in the domain) equal to one. Table 1 shows the L_1 error norm and order of convergence obtained for $T = 8$ and four different grid sizes, using Puckett's reconstruction method and different advection methods, and the EMFPA-SIR algorithm. The iterative method of Puckett [31] has been implemented following the indications given in [18]. Table 2 shows the same type of results, obtained using the proposed method (with and without including the spline-based reconstruction step) and the hybrid markers and volume of fluid method of Aulisa et al. [1]. It can be observed that the proposed method provides the most accurate results. It is of note that the errors obtained with the proposed method are lower than those of the hybrid method, which, in turn, represented a very substantial improvement over previous results. It can also be observed from Table 2 that the spline-based reconstruction step included in the proposed method introduces a substantial improvement in accuracy, especially for fine grids, without appreciably increasing the required CPU time.

The ability of the proposed method to maintain the integrity of thin filaments for coarse grids, avoiding the generation of 'flotsam', can be observed from Figs. 12 and 13, in which the results of the proposed method for the interface shape are compared with those obtained with the previous EMFPA-SIR method proposed by López et al. [24] (which, in turn, substantially improved the results of the methods proposed in [18,33], and were comparable with those of Aulisa et al. [1] only for the finest grid), and with the EMFPA algorithm and the reconstruction method of Puckett. In both figures, the dotted lines correspond to a reference solution obtained with the proposed method and the 256×256 grid. The greater accuracy of the proposed method is visually more evident in the results for the grid of 32×32 shown in Fig. 12, but also for the finer grids shown in Fig. 13. It can also be observed that, although the previous EMFPA-SIR method tends to reduce the generation of 'flotsam' and to maintain the length of the filament slightly longer than when Puckett's reconstruction method is used, the results for the final form of the interface at $t = T$ are still poor. For the grids depicted in Fig. 13, the accuracy provided by the EMFPA-SIR method is clearly greater than that obtained with Puckett's method, although the filament fragmentation is still present for the grid of 64×64.

Tables 1 and 2 also show the relative CPU-time consumed in the simulations carried out in this work, in all of which the analytical procedure described in Section 2.3 to enforce local volume conservation has been used, regardless of the reconstruction method adopted. It can be observed that the reconstruction method of Puckett is always more than twice as time consuming as the SIR algorithm, while the relative CPU-time consumed with the proposed method is very close to that required by the SIR algorithm, especially for fine grids. Note that, as the grid becomes finer, the ratio between the number of cells with two segments and only one segment decreases, and so the proposed algorithm tends to reach a degree of accuracy and

Table 1
L_1 error norm and order of convergence, as defined by Eqs. (18) and (19), obtained for $T = 8$ and different grid sizes in the time-reversed single vortex test, using Puckett's reconstruction method and different advection algorithms, and the EMFPA-SIR algorithm [24]

Reconstruction algorithm	Grid size	E	$\mathcal{O}$	Relative CPU-time
Puckett	*Rider and Kothe [33]*			
	32×32	4.78×10^{-2}		–
			2.78	
	64×64	6.96×10^{-3}		–
			2.27	
	128×128	1.44×10^{-3}		–
			–	
	256×256	–		–
	Harvie and Fletcher [18]			
	32×32	3.72×10^{-2}		–
			2.45	
	64×64	6.79×10^{-3}		–
			2.52	
	128×128	1.18×10^{-3}		–
			–	
	256×256	–		–
	EMFPA [24]			
	32×32	3.77×10^{-2}		3.0
			2.52	
	64×64	6.58×10^{-3}		13.6
			2.62	
	128×128	1.07×10^{-3}		59.4
			2.19	
	256×256	2.35×10^{-4}		248.4
SIR [24]	*EMFPA [24]*			
	32×32	4.64×10^{-2}		1.0
			2.97	
	64×64	5.94×10^{-3}		4.9
			3.46	
	128×128	5.39×10^{-4}		23.9
			2.48	
	256×256	9.63×10^{-5}		114.1

efficiency similar to that of the SIR algorithm. We cannot directly compare the computational resources required by the proposed method and the hybrid method of Aulisa et al. [1]. These authors recognize that marker advection, which is carried out using a fourth-order Runge–Kutta method, is by far the most time consuming part of the algorithm, and that only if the number of subdivisions of the time step is limited to one or two (the results presented in Table 1 were obtained using four substeps) will the CPU-time of the hybrid method be comparable to that of a VOF method using an iterative second-order accurate reconstruction algorithm.

Although the method proposed in this work (essentially of a VOF type) does not reach the level of accuracy of recent front tracking and markers methods, it substantially reduces the differences in the accuracy provided by Lagrangian methods and previous volume of fluid methods in tracking fluid filaments of thickness smaller than the cell size. We can mention, for example, the new algorithm proposed by Aulisa et al. [3], which is based on fixed markers, which are maintained throughout the simulation, and area-preserving markers, which are added at each time step by locally imposing an area conservation constraint. As in a

Table 2
L_1 error norm and order of convergence, as defined by Eqs. (18) and (19), obtained for $T = 8$ and different grid sizes in the time-reversed single vortex test

Reconstruction algorithm	Grid size	E	$\mathcal{O}$	Relative CPU-time
Hybrid markers-VOF [1]	32×32	2.53×10^{-2}		–
			3.19	
	64×64	2.78×10^{-3}		–
			2.54	
	128×128	4.78×10^{-4}		–
			2.04	
	256×256	1.16×10^{-4}		–
Proposed method without spline-based reconstruction step	32×32	7.41×10^{-3}		1.3
			1.83	
	64×64	2.12×10^{-3}		5.4
			2.31	
	128×128	4.27×10^{-4}		26.3
			1.58	
	256×256	1.43×10^{-4}		117.0
Proposed method (including spline-based reconstruction step)	32×32	5.78×10^{-3}		1.6
			1.71	
	64×64	1.77×10^{-3}		6.1
			2.42	
	128×128	3.30×10^{-4}		27.7
			1.93	
	256×256	8.69×10^{-5}		121.3

Comparison between the results of the proposed method with and without including the spline-based reconstruction step and those of the hybrid method of Aulisa et al. [1].

previous work of these authors [1], in order to increase the accuracy of the interface position at each time step, they use a fourth-order Runge–Kutta method to advect the markers along the streamlines, a scheme that introduces a high resolution of the interface and produces very accurate results, although probably at the cost of higher CPU-times consumed. To give an idea of the number of markers used in [3] and in this work, we can say that, for a single-vortex test using a square of fluid of side 9/32, initially centered at $(0.5, 0.75)$, and a grid of 32×32 cells, they used 860 markers to define the deformed interface at $t = T/2$, whereas we use 263 markers but only to improve the orientation and location of the interface segments.

5.2. Rayleigh–Taylor instability test

In this section, we present the results of a test in which a heavy fluid of density $\rho_1 = 1.225\ \mathrm{kg\,m^{-3}}$ lies above a light fluid of density $\rho_2 = 0.1694\ \mathrm{kg\,m^{-3}}$ in a rectangular domain 1 m wide by 4 m high. The dynamic viscosity of both fluids, which are assumed to be non-miscible, is $3.13 \times 10^{-3}\ \mathrm{kg\,m^{-1}\,s^{-1}}$. This test was performed by, among others, Bell and Marcus [6] and Puckett et al. [32] using a VOF-type method and by Popinet and Zaleski [29] using a front tracking method. Due to the symmetry of the problem, only half of the physical domain was solved. The interface shape is initially a cosine function, $y = -0.05\cos(\pi x/\lambda)$, where λ is the computational domain width. The horizontal coordinate x has its origin at the symmetry axis, where appropriate boundary conditions are imposed. Free-slip boundary conditions were imposed at the upper and bottom walls, and symmetry conditions at the other lateral boundary. A CFL number based on the maximum velocity in the domain of 0.1 was used to obtain the results presented below.

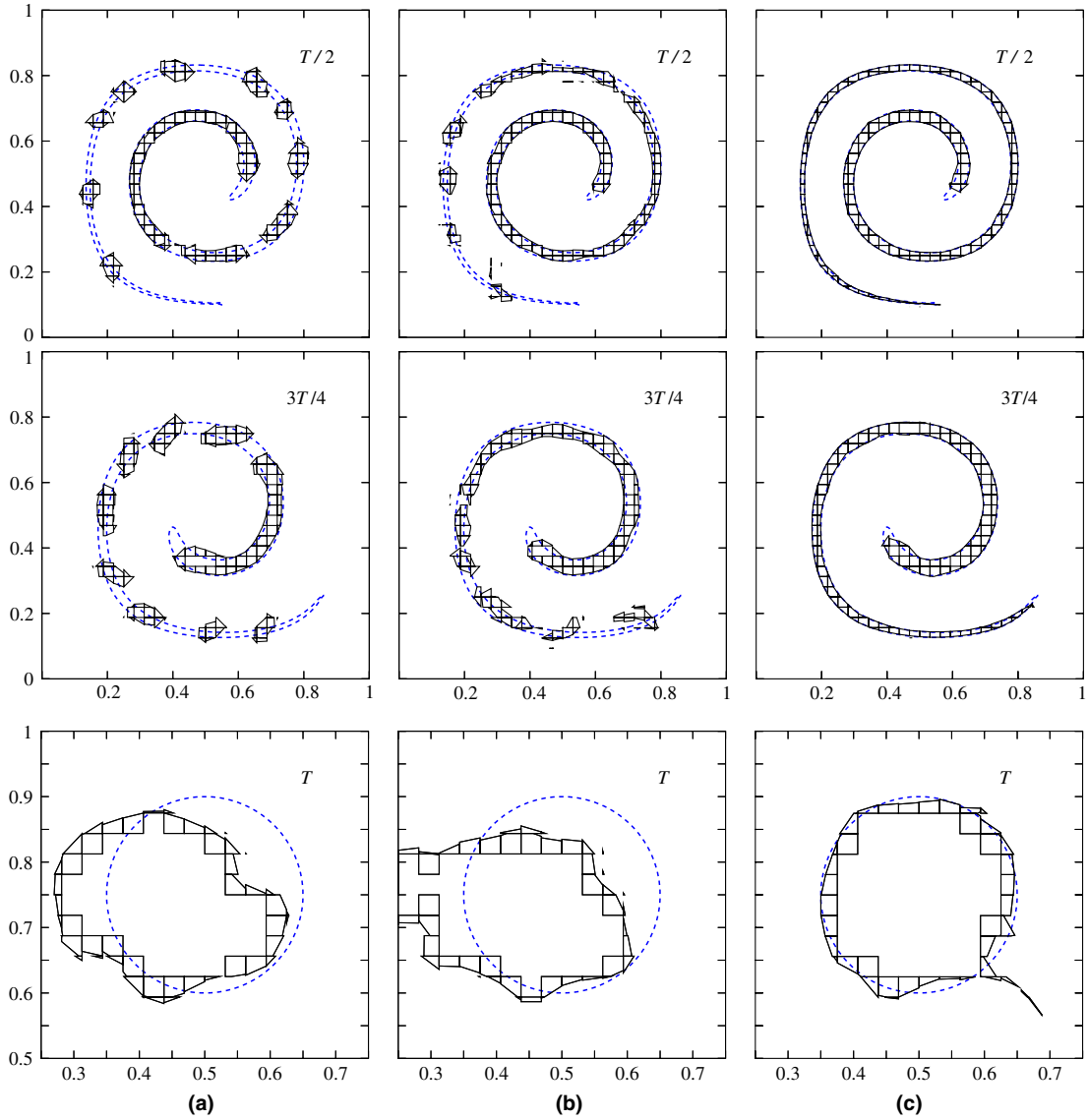

Fig. 12. Results obtained in the time-reversed single vortex test with $T = 8$ at three different instants, using a grid of 32×32 cells and (a) the EMFPA advection algorithm together with Puckett's reconstruction method, (b) the EMFPA-SIR method, and (c) the proposed method.

Figs. 14(a) and (b) show the results for the interface shape (instead of the interface segments, we represent the area occupied by the upper fluid in every cell of the interface, where $0 < F < 1$) obtained at different instants on a grid of 32×256 cells using the EMFPA advection algorithm and two reconstruction algorithms: Puckett's and the proposed algorithm, respectively. The ability of the proposed method to maintain thin filaments using a relatively coarse grid can be observed by comparing the results obtained at $t = 0.95$ s. Notice that the filament has almost disappeared in Fig. 14(a), whereas its integrity is maintained when the proposed reconstruction method is used. A detail of the interface shape at $t = 0.95$ s is shown at the right of

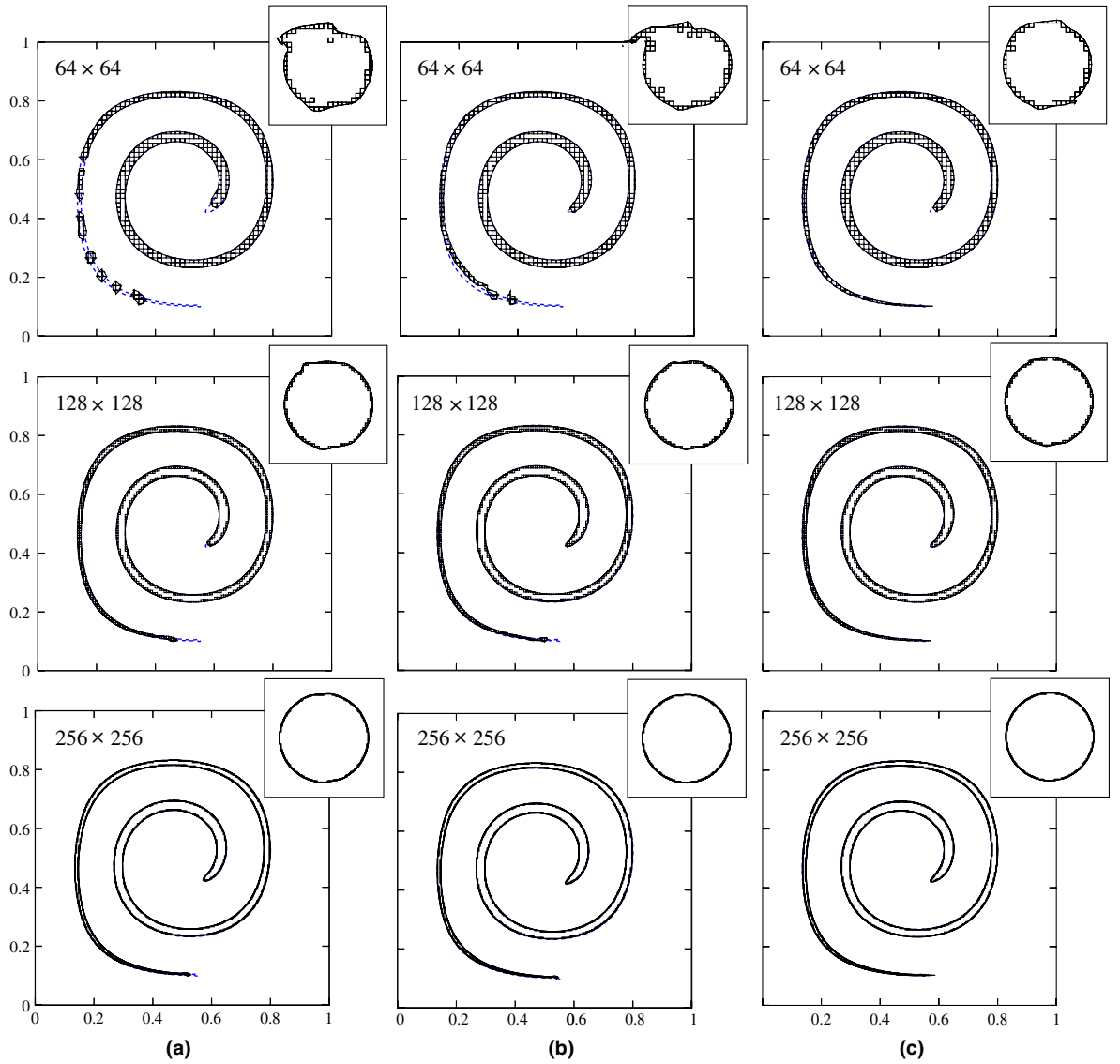

Fig. 13. Results obtained in the time-reversed single vortex test with $T = 8$ at $t = T/2$ and $t = T$ (the latter in the upper right-hand corner), using three grid sizes (64×64, 128×128 and 256×256) and (a) the EMFPA advection algorithm together with Puckett's reconstruction method, (b) the EMFPA-SIR method, and (c) the proposed method.

Figs. 14(a) and (b), where the results obtained on grids of 32×256 and 64×512 cells are compared with a reference solution obtained on a grid of 128×1024 cells using the EMFPA algorithm and the proposed reconstruction method, which was found to be nearly grid independent (for clarity, only the interface segment is represented at every interfacial cell in the reference solution). As occurred in tests with prescribed velocity field, the CPU time required by the proposed algorithm is shorter than that consumed with Puckett's algorithm, although the difference in the total CPU time consumed in the present test, in which the Navier–Stokes equations consume the largest fraction of the total CPU time, is much lower (around 4%

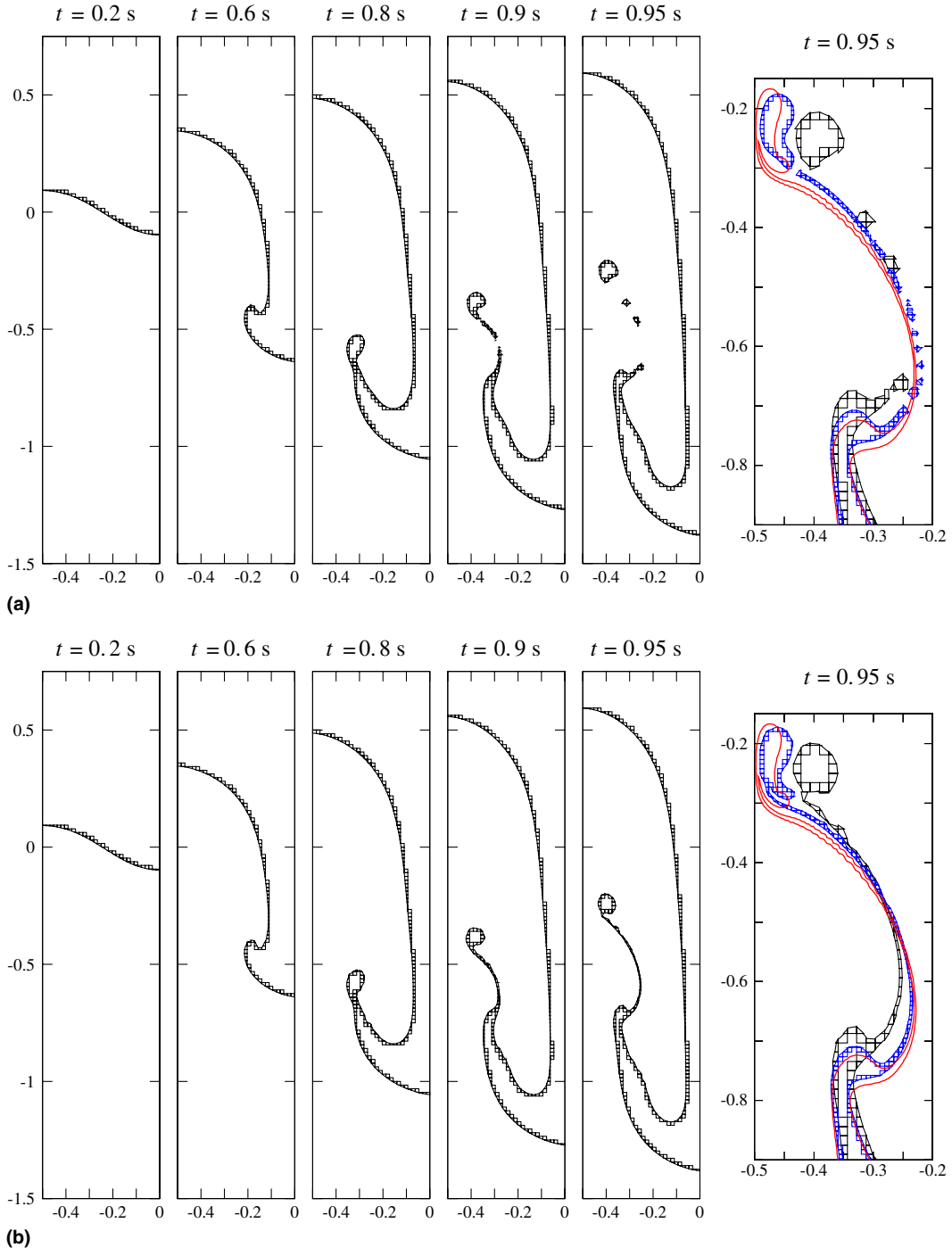

Fig. 14. Interface location at different instants in the Rayleigh–Taylor instability test. Results obtained with (a) Puckett's reconstruction method and (b) the proposed reconstruction method, using a grid of 32×256 cells. The enlarged detail at the right of each picture shows a comparison between the results for the filament shape at $t = 0.95$ s obtained on grids of 32×256 (black) and 64×512 (blue) cells, and a nearly grid independent solution (red line).

at the instant $t = 0.95$ s, for the grid of 32×256 cells). Similarly, the increase in the total CPU time consumed when the proposed algorithm is used instead of that proposed in [24] is very small (lower than 0.1% at the instant $t = 0.95$ s).

6. Conclusions

An improvement of the reconstruction method proposed by López et al. [24], which allows tracking fluid structures thinner than the cell size, has been presented. The method is based on using markers that are placed every time step at the mid-points of the cell interface segments in order to improve the accuracy of the interface reconstruction, although it can essentially be considered as a VOF-type method. The efficiency and accuracy of the new reconstruction method, which reduces drastically the formation of 'flotsam' even for very coarse grids, compare favorably with other recent volume of fluid and hybrid methods.

Acknowledgment

The authors gratefully acknowledge the support of the Spanish Ministerio de Ciencia y Tecnología under Grants DPI2001-1390-C02 and DPI2004-08198.

References

[1] E. Aulisa, S. Manservisi, R. Scardovelli, A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows, J. Comput. Phys. 188 (2003) 611–639.
[2] E. Aulisa, S. Manservisi, R. Scardovelli, S. Zaleski, A geometrical area-preserving volume-of-fluid advection method, J. Comput. Phys. 192 (2004) 355–364.
[3] E. Aulisa, S. Manservisi, R. Scardovelli, A surface marker algorithm coupled to an area-preserving marker redistribution method for three-dimensional interface tracking, J. Comput. Phys. 197 (2004) 555–584.
[4] A.S. Almgren, J.B. Bell, W.G. Szymczak, A numerical method for the incompressible Navier–Stokes equations based on an approximate projection, SIAM J. Sci. Comput. 17 (1996) 358–369.
[5] J.B. Bell, P. Colella, L.H. Howell, An efficient second-order projection method for viscous incompressible flow, in: Proceedings of the AIAA 10th Computational Fluid Dynamics Conference, Honolulu, HI, 1991, pp. 360–367.
[6] J.B. Bell, D.L. Marcus, A second-order projection method for variable density flows, J. Comput. Phys. 101 (1992) 334–348.
[7] D.J. Benson, Volume of fluid interface reconstruction methods for multi-material problems, Appl. Mech. Rev. 55 (2002) 151–165.
[8] W.J. Boettinger, J.A. Warren, C. Beckermann, A. Karma, Phase-field simulation of solidification, Annu. Rev. Mater. Res. 32 (2002) 163–194.
[9] A.J. Chorin, Curvature and solidification, J. Comput. Phys. 58 (1985) 472–490.
[10] P. Colella, Multidimesional upwind methods for hyperbolic conservation laws, J. Comput. Phys. 87 (1990) 171–200.
[11] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, J. Comput. Phys. 183 (2002) 83–116.
[12] G. Farin, Curves and Surfaces for Computer-aided Geometric Design, A Practical Guide, fourth ed., Academic Press, San Diego, 1997.
[13] R.P. Fedkiw, G. Sapiro, C.-W. Shu, Shock capturing, level sets, and PDE based methods in computer vision and image processing: a review of Osher's contributions, J. Comput. Phys. 185 (2003) 309–341.
[14] S.H. Garrioch, B.R. Baliga, A PLIC volume tracking method using circle-fit segment orientation, in: Proceedings of the 4th International Conference on Multiphase Flow, New Orleans, LA, 2001.
[15] I. Ginzburg, G. Wittum, Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants, J. Comput. Phys. 166 (2001) 302–335.
[16] P. Gómez, J. Hernández, J. López, On the reinitialization procedure in a narrow-band locally refined level set method for interfacial flows, Int. J. Numer. Meth. Eng. 63 (2005).
[17] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (1999) 423–456.

[18] D.J.E. Harvie, D.F. Fletcher, A new volume of fluid advection algorithm: the stream scheme, J. Comput. Phys. 162 (2000) 1–32.
[19] D.J.E. Harvie, D.F. Fletcher, A new volume of fluid advection algorithm: The defined donating region scheme, Int. J. Numer. Meth. Fluids 35 (2001) 151–172.
[20] D. Jacqmin, Calculation of two-phase Navier–Stokes flows using phase-field modeling, J. Comput. Phys. 155 (1999) 96–127.
[21] R. Kobayashi, Modeling and numerical simulations of dendritic crystal growth, Physica D 63 (1993) 410–423.
[22] D.B. Kothe, M.W. Williams, K.L. Lam, D.R. Korzekwa, P.K. Tubesing, E.G. Puckett, A second-order accurate, linearity-preserving volume tracking algorithm for free surface flows on 3-D unstructured meshes, in: 3rd ASME/JSME Joint Fluids Engineering Conference, San Francisco, CA, 1999.
[23] D.B. Kothe, W.J. Rider, S.J. Mosso, J.S. Brock, J.I. Hochstein, Volume tracking of interfaces having surface tension in two and three dimensions, Technical Report AIAA 96-0859, AIAA, 1996.
[24] J. López, J. Hernández, P. Gómez, F. Faura, A volume of fluid method based on multidimensional advection and spline interface reconstruction, J. Comput. Phys. 195 (2004) 718–742.
[25] D. Lörstad, L. Fuchs, High-order surface tension VOF-model for 3D bubble flows with high density ratio, J. Comput. Phys. 200 (2004) 153–176.
[26] S.J. Mosso, B.K. Swartz, D.B. Kothe, S.P. Clancy, Recent enhancements of volume tracking algorithms for irregular grids, Technical Report LA-CP-96-227, Los Alamos National Laboratory, 1996.
[27] S. Osher, R. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys. 169 (2001) 463–502.
[28] J.E. Pilliod, E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2004) 465–502.
[29] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, Int. J. Numer. Meth. Fluids 30 (1999) 775–793.
[30] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran, Cambridge University Press, Cambridge, UK, 1986.
[31] E.G. Puckett, A volume of fluid interface tracking algorithm with applications to computing shock wave rarefraction, in: Proceedings of the 4th International Symposium on Computational Fluid Dynamics, 1991, pp. 933–938.
[32] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, W.J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, J. Comput. Phys. 130 (1997) 269–282.
[33] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[34] M. Rudman, Volume-tracking methods for interfacial flow calculations, Int. J. Numer. Meth. Fluids 24 (1997) 671–691.
[35] M. Rudman, A volume-tracking method for incompressible multifluid flows with large density variations, Int. J. Numer. Meth. Fluids 28 (1998) 357–378.
[36] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[37] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split Eulerian–Lagragian advection, Int. J. Numer. Meth. Fluids 41 (2003) 251–274.
[38] J.A. Sethian, Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science, second ed., Cambridge University Press, Cambridge, UK, 1999.
[39] J.A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annu. Rev. Fluid Mech. 35 (2003) 341–372.
[40] S. Shin, D. Juric, Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity, J. Comput. Phys. 180 (2002) 427–470.
[41] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3 D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2000) 301–337.
[42] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.
[43] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.
[44] A.A. Wheeler, W.J. Boettinger, G.B. McFadden, A phase-field model for isothermal phase transitions in binary alloys, Phys. Rev. A 45 (1992) 7424–7439.
[45] D.L. Youngs, Time-dependent multimaterial flow with large fluid distortion, in: K. Morton, M. Baines (Eds.), Numerical Methods for Fluid Dynamics, Academic Press, New York, 1982, pp. 273–285.