Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 195 (2004) 718–742
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A volume of fluid method based on multidimensional advection and spline interface reconstruction

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

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

Received 23 May 2003; received in revised form 14 October 2003; accepted 15 October 2003

Abstract

A new volume of fluid method for tracking two-dimensional interfaces is presented. The method involves a multidimensional advection algorithm based on the use of edge-matched flux polygons to integrate the volume fraction evolution equation, and a spline-based reconstruction algorithm. The accuracy and efficiency of the proposed method are analyzed using different tests, and the results are compared with those obtained recently by other authors. Despite its simplicity, the proposed method represents a significant improvement, and compares favorably with other volume of fluid methods as regards the accuracy and efficiency of both the advection and reconstruction steps.

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

1. Introduction

Among the different approaches developed during the last three decades to simulate free surface and interfacial flows (see, for example, the review by Scardovelli and Zaleski [1]), volume tracking is one of the most widely used. A review of the historical development of volume tracking methods since the earliest works of Debar [2], Noh and Woodward [3], Chorin [4], Hirt and Nichols [5] or Youngs [6], can be found in [7–10]. In the volume of fluid (VOF) method the interface evolution is described using a discrete function, F, whose value in each cell of the computational mesh, in single fluid problems, is the fraction of the cell occupied by the fluid, so that it is 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. This volume fraction is a discretized version of a function, f, which is continuous everywhere except at the interface, where it jumps from zero to one, and satisfies a standard advection equation,

* Corresponding author.
E-mail addresses: joaquin.lopez@upct.es (J. López), jhernandez@ind.uned.es (J. Hernández).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2003.10.030

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla f = 0. \tag{1}$$

The initial distribution of the discrete volume fraction is determined from the initial interface geometry. At any time step, the interface is first "reconstructed" at each cell from the F distribution, and is then advected by solving Eq. (1), using geometric considerations to compute volume fluxes through cell boundaries. The different VOF methods can be distinguished by the features of the interface reconstruction algorithm and the method used for time integration of the volume fraction equation. Successive improvements in the VOF method have kept this method competitive with more recent methods such as the front tracking [11] or level set [12] methods. Rider and Kothe [9] proposed a linearity-preserving, piecewise linear interface geometry approximation and a multidimensional unsplit time integration scheme, in which material volume fluxes are computed systematically with a set of simple geometric tasks. These authors use trapezoidal flux polygons at cell faces such as those represented in Fig. 1(a), in which edge 1–2 is parallel to edge 3–4. Harvie and Fletcher [13] proposed a new VOF advection algorithm (termed the Stream scheme) that also uses a piecewise linear interface calculation (PLIC) method coupled to a fully multidimensional cell face flux integration technique. This technique is based on splitting each cell face flux into a discrete number of streamtubes corresponding to equally-spaced face sections (see Fig. 1(b)). The volume of fluid in each streamtube which is fluxed through a particular cell face during a time step is approximately determined by integrating backwards in time through the computational time step along the central streamline of the tube. More recently, Harvie and Fletcher [14] 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

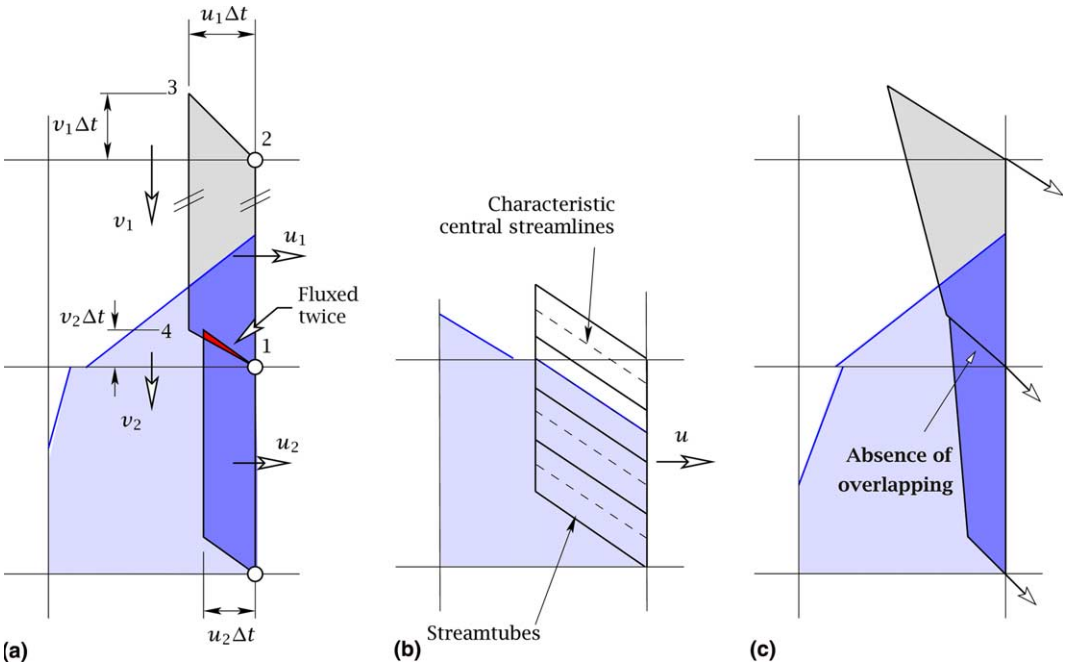

Fig. 1. Examples of flux polygon construction at cell faces: (a) Rider and Kothe [9]; (b) Harive and Fletcher [13] (worst-case scenario); (c) present work.

direction-split advection scheme of Youngs [6]. Another example of a multidimensional advection algorithm can be found in [15].

In PLIC methods, the interface is approximated in each cell by a straight line defined by $\boldsymbol{n}\cdot\boldsymbol{x}+C=0$, where $\boldsymbol{n}$ and $\boldsymbol{x}$ are the outward unit vector normal to the interface and the position vector of a generic point on the line, respectively, and C is a constant. Methods, such as those of Youngs [16] and Puckett [17], or that proposed in the present work, mainly differ in the algorithm used to determine the orientation of the interface. Youngs' method is an explicit method with an accuracy between first and second order, in which a finite-difference gradient approximation of the volume fraction is used to obtain the interface normal as $\boldsymbol{n}=\nabla f/|\nabla f|$. Puckett's method is a second-order iterative method, which is based on minimizing an error function defined by the difference between the actual volume fractions and those obtained by extending the approximate interface over 3×3 cells. Both Rider and Kothe [9] and Harvie and Fletcher [13] use these PLIC reconstruction methods. A more efficient second-order algorithm, named ELVIRA, which reduces the computational effort of the iterative method of Puckett [17], was proposed by Pilliod [18]. In this algorithm, the interface orientation is selected from six different candidates for which the above mentioned error function is determined using backward, central and forward finite-difference approximations for the volume fraction gradient. Another second-order method, only applicable to orthogonal grids, was proposed by Chorin [19], in which the normal $\boldsymbol{n}$ 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. Recent enhancements of this method, which perform well on unstructured meshes, were proposed by Mosso et al. [20] and Garrioch and Baliga [21]. In a more recent work, Scardovelli and Zaleski [22] 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.

The main objective of this work was to develop an improved VOF method based on multidimensional advection using edge-matched flux polygons, and spline-based interface reconstruction, whose accuracy is compared with that of other existing approaches, in particular the above mentioned methods recently proposed by Rider and Kothe [9], Harvie and Fletcher [13] and Scardovelli and Zaleski [22]. We will refer to the coupled Edge-Matched Flux Polygon Advection and Spline-based Interface Reconstruction algorithms proposed in this work as the EMFPA–SIR method.

2. Multidimensional advection algorithm

For two-dimensional (2D) incompressible flows, Eq. (1) can be integrated over a given cell Ω of area A_Ω and the time interval $\Delta t=t_{n+1}-t_n$, to obtain, at each time step,

$$F^{n+1}=F^n-\frac{1}{A_\Omega}\int_{t_n}^{t_{n+1}}\int_\Omega\left[\frac{\partial}{\partial x}(uf)+\frac{\partial}{\partial y}(vf)\right]\mathrm{d}\Omega\,\mathrm{d}t, \tag{2}$$

where the divergence constraint $\nabla\cdot\boldsymbol{v}=0$ is taken into account. The integral in Eq. (2), which represents the net area of fluid advected out of the cell, A_{F_T}, is solved geometrically using the method described hereafter.

In each face of a given cell, a flux polygon such as that represented in Figs. 1(c) and 2 is constructed. The volumetric flux (per unit length) through any face Γ can be expressed as

$$q=\int_\Gamma(u\,\mathrm{d}y-v\,\mathrm{d}x), \tag{3}$$

and, to ensure volume conservation, the area of the flux polygon must be determined from

$$A_d=\int_{t_n}^{t_{n+1}}q\,\mathrm{d}t. \tag{4}$$

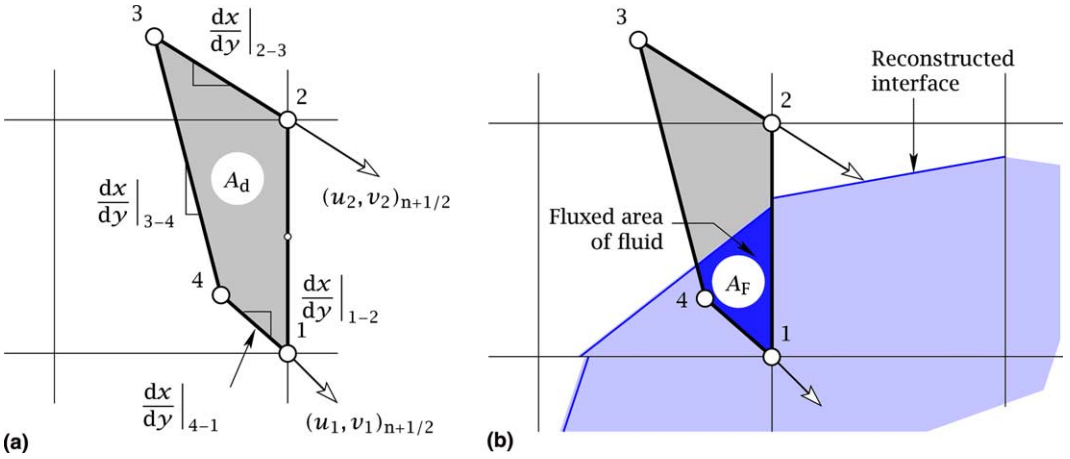

Fig. 2. (a) Edge-matched flux polygon. (b) Area of fluid advected through a cell face.

The integral in Eq. (3) is determined from the velocity components at the face centers, using a midpoint rule, and the integral in Eq. (4) is solved using a trapezoidal quadrature rule. For example, the area of the flux polygon corresponding to the right face (1–2) of the quadrilateral cell shown in Fig. 2 reduces to $A_d = \frac{1}{2}(u^n_{1-2} + u^{n+1}_{1-2})\Delta y \Delta t$, with u^n_{1-2} and u^{n+1}_{1-2} being the horizontal velocity components at the face center at instants t_n and t_{n+1}, respectively, and Δy the cell size. Note that the volume in each cell is conserved provided that the velocity field at the centers of the cell faces satisfies the discrete continuity condition. The velocity at the cell corners, $\boldsymbol{v}_j$, are obtained by interpolation using a weighted average of the velocities at the center of the four surrounding faces, $\boldsymbol{v}_i$:

$$\boldsymbol{v}_j = \frac{\sum_{i=1}^{4} \frac{1}{d_i}\, \boldsymbol{v}_i}{\sum_{i=1}^{4} \frac{1}{d_i}},$$

where d_i is the distance from the cell corner j to the face center i. The slopes of the flux polygon edges are determined from these velocity components at the cell corners as follows (see Fig. 2):

$$\left.\frac{dx}{dy}\right|_{2-3} = \frac{u_2^{n+1/2}}{v_2^{n+1/2}}, \tag{5}$$

$$\left.\frac{dx}{dy}\right|_{3-4} = \frac{x_1 - x_2 - (u_1^{n+1/2} - u_2^{n+1/2})\Delta t}{y_1 - y_2 - (v_1^{n+1/2} - v_2^{n+1/2})\Delta t}, \tag{6}$$

$$\left.\frac{dx}{dy}\right|_{4-1} = \frac{u_1^{n+1/2}}{v_1^{n+1/2}}, \tag{7}$$

$$\left.\frac{dx}{dy}\right|_{1-2} = \frac{x_2 - x_1}{y_2 - y_1}. \tag{8}$$

The locations of points 3 and 4 are obtained from Eqs. (5)–(8) and the condition expressed by Eq. (4). It should be pointed out that the corner velocities are not used to calculate the area of the flux polygon, but

only the slopes of its edges. Note that these slopes are calculated from the corner velocities at the intermediate time $t_n + \frac{1}{2}\Delta t$, which are obtained from linear interpolation between the values corresponding to the instants t_n and t_{n+1}. Also note that, although the mesh represented in Fig. 2 is Cartesian, the method can also be applied, without lack of generality, to unstructured meshes.

The area of fluid advected through each face of a given cell, A_F, which is taken to be positive when the fluid leaves the cell, will depend on the shape of the face flux polygon and the location of the reconstructed interface. Obviously, if the flux polygon is full of fluid or empty, the value of A_F will be equal to the value of A_d obtained from Eq. (4) or zero, respectively. Otherwise, the intersected area between the flux polygon and the area occupied by the fluid (Fig. 2(b)) must be calculated. The problem turns out to be purely geometric, and can be solved using techniques such as those discussed by O'Rourke [23].

Once the interface has been reconstructed (using, for example, any of the methods mentioned in the Introduction), and its intersection with the face flux polygons has been determined, the total net area A_{F_Γ} of fluid that leaves each cell is calculated from

$$A_{F_\Gamma} = \sum_{i=1}^{4} A_{F_i}, \tag{9}$$

and Eq. (2) can be expressed as

$$F^{n+1} = F^n - \frac{1}{A_\Omega} \sum_{i=1}^{4} A_{F_i}. \tag{10}$$

Notice that, in the proposed advection algorithm, the flux polygons of cell faces with a common node have an edge with a common orientation (although, in general, the length of this edge may not be the same in all these polygons). This, in the presence of spatially-varying velocity fields, avoids the over/underlapping of flux polygons, which, as discussed by Rider and Kothe [9], tends to produce overshoots or undershoots in the distribution of F (volume fractions greater than 1 or lower than 0) and make the use of a local redistribution algorithm necessary. Fig. 3 shows a comparison between the flux polygons constructed using the method proposed by Rider and Kothe [9] (Fig. 3(b)) and the edge-matched flux polygons proposed in this work (Fig. 3(c)), for the flows corresponding to three different advection tests that will be considered below: diagonal translation, pure rotation and the "vortex-in-a-box" test. It can be observed that both methods produce the same flux polygons for the simple translation case. For the other two cases, particularly for the "vortex-in-a-box" test, where the vorticity is not uniform, the differences are appreciable. Notice the absence of over/underlapping between the polygons constructed in accordance with the proposed method.

For the rotation and the "vortex-in-a-box" cases, we also represent in Fig. 3, with dashed lines, the "almost exact" limits of the donating flux regions corresponding to each cell face, which are obtained by integrating very accurately the equations that determine the location of the fluid particles that cross the face during the integration time interval Δt. It can be observed that the shape of the flux polygons generated with the method proposed in this work follows these "exact" limits very closely. The suppression of polygon over/underlapping situations, together with the increase in accuracy which results from the reduction of discretization errors, obviously tend to reduce interface advection errors and the need for a redistribution algorithm. Also note that the proposed advection algorithm ensures overall conservation of fluid volume (area), and so this is also a highly desirable property of any redistribution algorithm, which is the case with that proposed by Harvie and Fletcher [13].

It should also be pointed out that the procedure described for constructing the flux polygons at cell faces entails a discrete area-conservation constraint, meaning that a divergence correction such as that introduced by Rider and Kothe [9] is unnecessary. Fig. 4 shows, for the single vortex test, typical error distri-

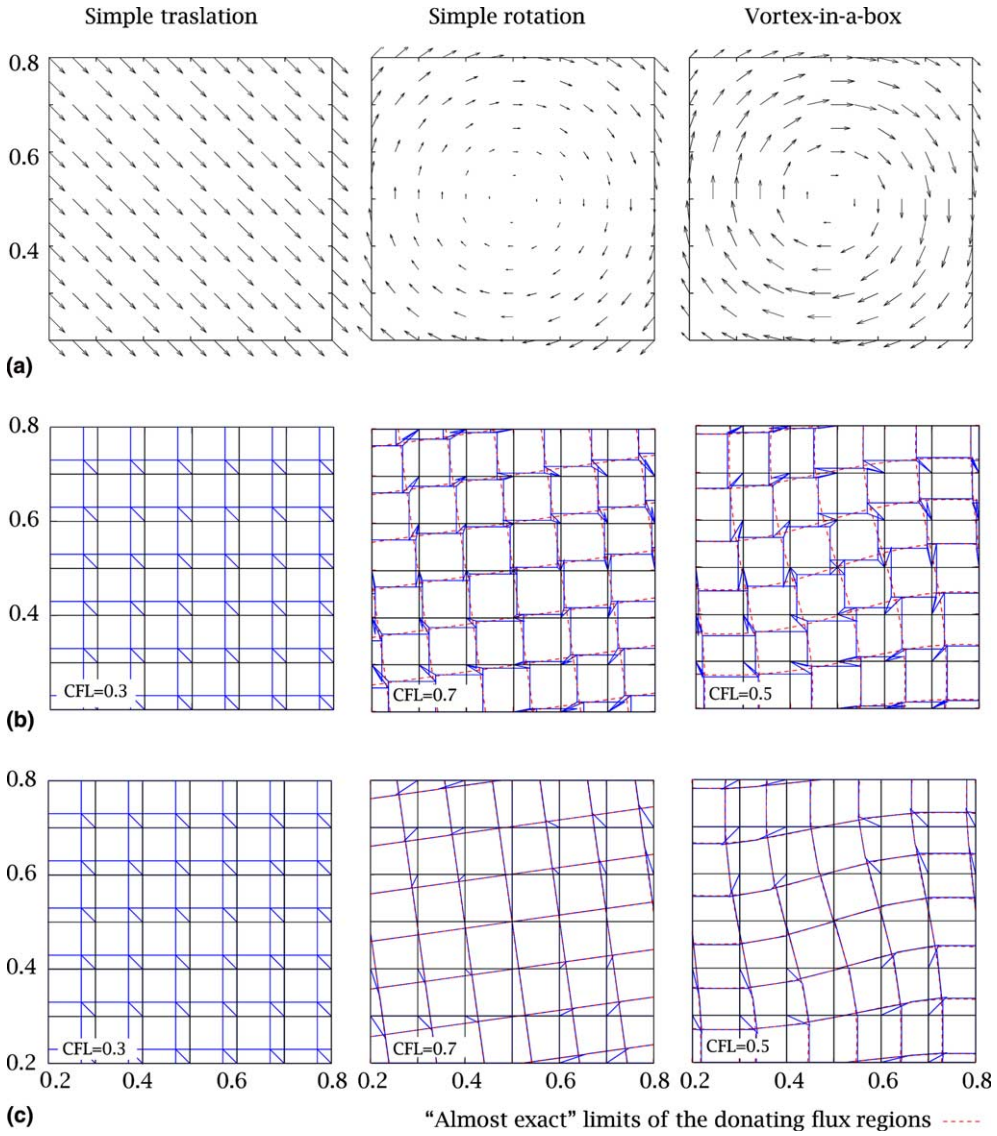

Fig. 3. Flux polygons at cell faces corresponding to three different flows: (a) velocity vectors; (b) flux polygons constructed using the method proposed by Rider and Kothe [9]; (c) edge-matched flux polygons proposed in this work.

butions obtained in the calculation of the flux through the cell faces, defined as $E_{\text{flux}} = \sum_{i=1}^{4} |A_d^i - A_d^{\text{exact}}|$, and of the divergence, $E_{\text{div}} = \sum_{i=1}^{4} A_d^i$, in each cell, using our own implementation of the Rider and Kothe [9] advection method. The upper and lower results in this figure were obtained, respectively, with and without taking into account the correction of the flux polygons at the cell corners proposed by Rider and Kothe [9], and show that a subsequent divergence correction is necessary in the first case. Notice, however, that the accuracy of the algorithm deteriorates unless some type of flux correction, such as that mentioned above, is introduced. In the new algorithm that we propose here, such divergence correction is not necessary, and the need for systematic corner flux polygon correction is avoided with the polygon construction procedure explained above.

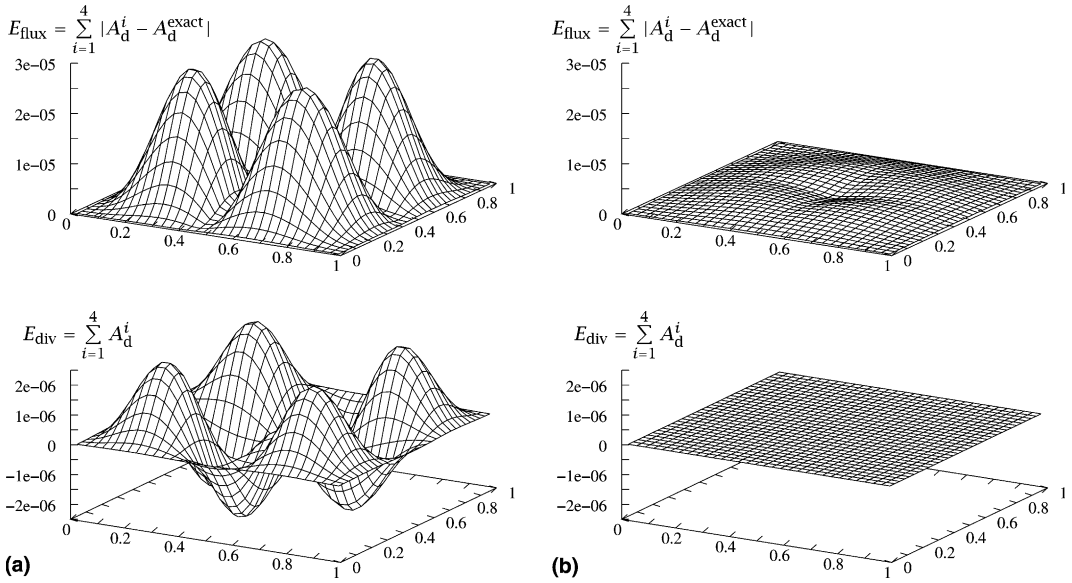

Fig. 4. Flux and divergence errors in the "vortex-in-a-box" test with a 32×32 grid and a maximum CFL number of 0.5: (a) our implementation of the Rider and Kothe [9] advection algorithm, before divergence correction; (b) without flux polygon correction at cell corners.

It should also be mentioned that the constraint imposed by Eq. (4) to construct the flux polygons may require, in certain circumstances, that the algorithm operate on a more extensive region around the cell face (where the corresponding flux polygon may be located) when the velocity is nearly parallel to the cell face and the CFL number is close to unity. Fig. 5 shows a case in which vertices 3 and 4 of the flux polygon lie outside the area occupied by the cells surrounding nodes 2 and 1, respectively. In such a case, an equivalent flux polygon is constructed in which the area constraint given by Eq. (4) is not imposed, and the lengths of edges 2–3′ and 1–4′ are determined as $|\boldsymbol{v}_2|\Delta t$ and $|\boldsymbol{v}_1|\Delta t$. The area of fluid advected through edge 1–2 is taken to be equal to the area of fluid intersected by the flux polygon and corrected by a factor equal to the ratio between the area given by Eq. (4) and that of polygon 1–2–3′–4′. This modification of the algorithm may cause small overshoots or undershoots in the distribution of F. When necessary (in few cases when the CFL number is close to one), a local redistribution algorithm similar to that of Harvie and Fletcher [13] is used. Note that the problem just described is not of a fundamental character, and that other alternatives that could have been chosen to solve it would probably have a minor effect on the results.

It is well known that VOF schemes tend to produce "blobby" filaments when these become too thin to be adequately resolved on the grid. The volume tracking algorithm induces errors in the calculation of the interface normal from the volume fraction gradients and, in the attempt to locally enforce mass conservation, incorrectly locates the fluid volume when the grid is not fine enough to resolve small fluid flow features. Černe et al. [24] proposed an expression based on the gradient of the volume fractions to assess the accuracy of reconstruction and to activate, when necessary, an adaptive grid refinement algorithm or other alternative reconstruction algorithms [25]. More recently, Aulisa et al. [26] proposed a hybrid marker-VOF method to overcome these difficulties, which are inherent in VOF methods. On the other hand, as mentioned by Harvie and Fletcher [13], integration inaccuracies can also cause the artificial generation of "wisps", that is, small areas of fluid in void regions or of void in fluid regions. To alleviate this difficulty, Harvie and Fletcher [13] used an algorithm similar to the above mentioned redistribution algorithms used to overcome the over/undershoots in the distribution of F. Note, however, that these algorithms are

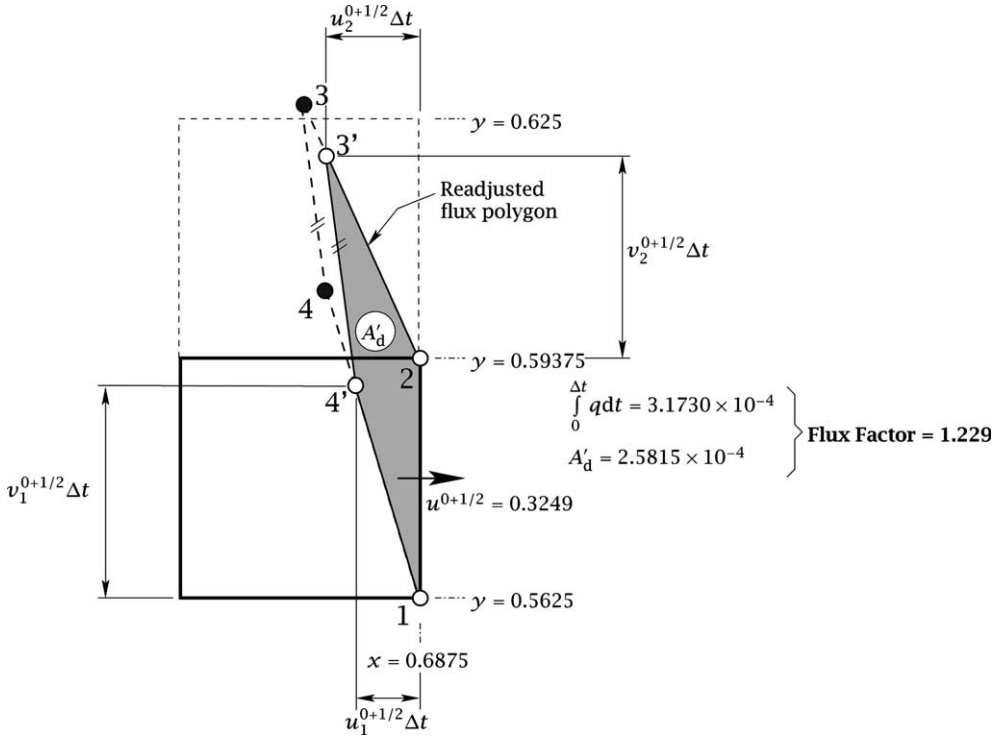

Fig. 5. Flux polygon adjustment at the time step from $t = 0$ to $t = \Delta t = 1/32$, in the cell whose coordinates are given in the figure, in the time-reversed single vortex test with $T = 2$ and a 32×32 grid.

heuristic and do not completely remove the effect of these errors. In this work, in order to better assess the intrinsic features of the new advection algorithm, we have not used any algorithm of this kind (although one could easily have been implemented) to eliminate wisps. Anyway, in the proposed advection algorithm, only the flux polygon adjustments may produce, in very few cases, small "wisp" errors that do not adversely affect the computational efficiency of the algorithm. As an example, for the time-reversed single vortex test described in Section 4.5, with $T = 2$ and a grid with 128×128 cells, the proposed EMFPA–SIR method produced (very small) wisps only when the CFL number based on the maximum velocity in the domain was greater than about 0.95. In all tests with CFL > 0.95, a maximum of 18 cells in the whole computational domain with a maximum void volume fraction of 3.4×10^{-6} were found in fluid regions. For a grid of 32×32 cells and the same test conditions, no wisps were produced with the proposed algorithm, whereas the Stream scheme of Harvie and Fletcher [13] required the use of an algorithm to detect and correct wisps of void or fluid volume fractions of up to 10^{-2} in fluid and void regions, respectively.

Although in the following section we propose a new interface reconstruction method, the improvement in accuracy produced specifically by the new advection method proposed here will be assessed and demonstrated in Section 4 by comparing its results with those obtained by other authors using the same reconstruction methods of Youngs and Puckett.

The extension of the proposed advection algorithm to three dimensions will be the subject of future work. As generally occurs with VOF methods, this extension is not straightforward, and the geometric operations involved are much more complicated than in two dimensions. However, VOF methods can be efficient, robust and relatively simple to implement in 3D if the geometrical operations required to compute

fluid volume fluxes are performed systematically, using a set of simple geometric tasks, as discussed by Kothe et al. [27]. These authors extended a PLIC–VOF method to general 3D hexahedral meshes using this strategy. They introduced exact analytic solutions to calculate the truncation volumes formed by an arbitrary plane cutting through a general hexahedron, which can be applied to extract intersections between flux polyhedra and the volume occupied by the fluid. An extension to generalized unstructured meshes was discussed by Kothe et al. [28]. The shapes of the flux polyhedra that may arise in the extension to 3D of the proposed algorithm would be only slightly more complex than in previous multidimensional VOF advection algorithms (the only geometric difference being that the polyhedron face opposite the corresponding cell face is in general not parallel to this latter), but do not increase substantially the complexity of the geometric operations that are needed. However, we must recognize that the extension of unsplit advection schemes to generalized unstructured 3D grids is a difficult and challenging task, to which much research effort is currently being devoted [28].

3. Spline-based interface reconstruction algorithm

A new PLIC interface reconstruction method based on spline interpolation is presented in this section. The basic idea is to readjust the orientation of the segments representing the interface in each cell with the aid of a cubic spline interpolation through the center points of these segments. The first step in the algorithm is to make an ordered list of the points that are to be interpolated. To this end, we use the following procedure, which is similar to that used by Ginzburg and Wittum [29] to obtain an aligned grid by irregular refining of the cells intersected by a spline-interpolated curve that follows the VOF distribution and represents the interface:

1. The orientation of the interface segment in each cell is initially determined using the method of Youngs [16], and its position is obtained by imposing the condition that the area of fluid in the cell is equal to $F^n A_\Omega$, using the iterative method of Brent (see, for example [30]). For rectangular grids, analytical relations such as those proposed by Gueyffier et al. [31] or Scardovelli and Zaleski [32] could also be used.
2. Two arbitrary adjacent cells, such as cells H and I in Fig. 6, which satisfy the condition that the interface segment in each cuts their common face, are selected (note that the VOF representation of the interface is not in general continuous between cells, and so the interface segments of adjacent cells may cut non-common faces). If it is not possible to find two adjacent cells that satisfy this condition along the whole interface (or any closed portion of it, such as that corresponding to a drop or a bubble), the first two selected cells will be those whose interface segment centers are closest to each other. The center points of the interface segments of these cells (points h and i in Fig. 6) will be the first two points of the interpolation curve. In Fig. 6, the order of the points in the curve is chosen so that the cells with $F = 1$ remain on the left side as we move forward in the curve.
3. The next point in the list corresponds to a cell selected from among the neighbors of the current cell I (including cells with only one common node). The selection is made as follows. Straight lines (dashed lines in Fig. 6) are traced between the center point (i) of the interface segment in cell I and those in the neighboring cells (h, j and k). The new point which is to be interpolated is that (j) for which the corresponding straight line forms the minimum angle with the line through the last two selected points (h and i).
4. In some cases, selecting points corresponding to cells with an F value close to 0 or 1 may produce local distortions of the interpolation curve. To avoid this, we exclude cells with $F \leqslant 0.02$ or $F \geqslant 0.98$, in which the readjusted interface segment orientation is determined by the normal to the spline-interpolation curve at the point closest to the center of the initial interface segment.
5. When it is not possible to find a new valid point in the surrounding cells (because $F \leqslant 0.02$ or $F \geqslant 0.98$ in all of them), we seek a new point within a radius of approximately twice the cell size. If it is still not pos-

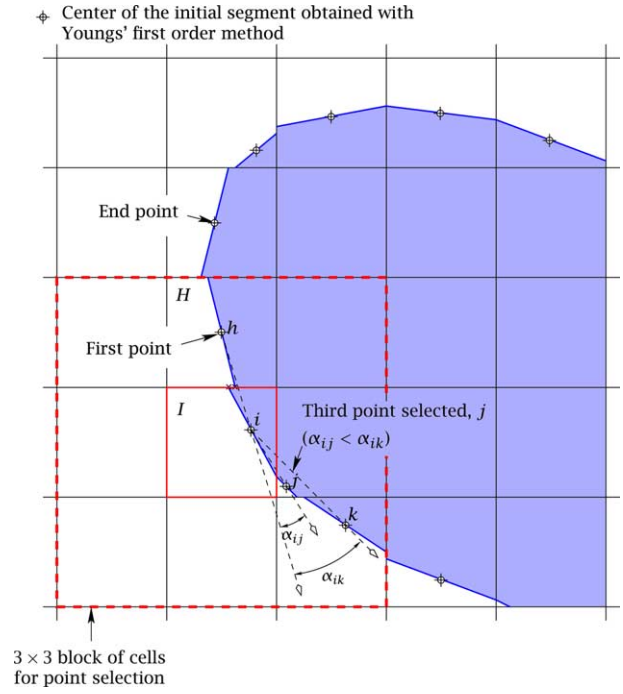

Fig. 6. Schematic representation of the points selection procedure used to construct the spline-based interface.

sible to find a new point because of the presence of a wall or of the initial point is reached, the construction of a new curve begins.

Once the centers of the interface segments have been ordered, their coordinates 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

$$s_j = \sum_{i=1}^{j-1} \left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 \right]^{1/2}. \tag{11}$$

Other possible parametrization methods can be found in Farin [33]. The parametric curve $x = x(s)$, $y = y(s)$ is then constructed using a natural cubic spline interpolation (see, for example, [30]), and the normal to each segment of the interface is obtained as

$$\boldsymbol{n}_j = \frac{[-y'(s_j), x'(s_j)]}{[x'^2(s_j) + y'^2(s_j)]^{1/2}}, \tag{12}$$

except at the extreme points of the interpolation curve, where the normal to the interface segment is obtained from the circle formed by the three starting and end points of the curve, respectively. The equation for each interface segment is finally determined using Brent's method.

The smooth, twice differentiable cubic spline curve constructed as described above, allows an analytic determination of the interface curvature at each cell from

$$k_j = \frac{x''(s_j)y'(s_j) - y''(s_j)x'(s_j)}{(x'^2(s_j) + y'^2(s_j))^{3/2}}. \tag{13}$$

At the extreme points of the interpolation curve, the curvature may be determined, for example, by the circle formed by the three starting or end points of the curve.

In general, standard VOF methods are not very accurate in computing geometric characteristics such as curvature, and so different authors have attempted to improve the accuracy of calculating the curvature from volume fractions (see, for example, [34–37]). On the other hand, in the context of front tracking methods, Popinet and Zaleski [38], among others, found that cubic spline interpolants were sufficiently smooth to ensure an accurate calculation of the interface curvature when modeling the surface tension force. In the present work, in order to assess the accuracy of the methods based on using Eq. (13), a test that involved computing the curvature of a circular interface of a given radius, d, was applied. Fig. 7(a) shows the results for the average curvature as a function of the grid size computed from the spline-based reconstructed interface. These results are compared with those reported by Meier et al. [39], which were obtained with the continuum surface force (CSF) method of Brackbill et al. [40], and with a new method based on using an adjusted estimator function. In the first case, the function F was convolved with a quadratic B-spline kernel to construct a smoothed volume fraction function (a discussion of the merits of different kernels for smoothing the F distribution can be found in Aleinov and Puckett [41]). It can be observed from Fig. 7(a) that the results obtained with the proposed SIR algorithm substantially improve those obtained with the CSF method and are slightly better than those reported by Meier et al. [39]. It can be observed from Fig. 7(b), which shows the curvature distribution along the interface obtained with the SIR algorithm, that the level of noise in the curvature estimates can be further reduced by using wider F ranges to exclude cells with F values that are too high or too low from the list of points to be interpolated (the results obtained excluding points with $F \leqslant 0.05$ and $F \geqslant 0.95$ are indicated in Fig. 7(b) with open circles). However, a better alternative consists of detecting points where the curvature oscillation obtained from Eq. (13) is too high, and relocating them by successive point-removal and interpolation operations (smoothing algorithms of this type are discussed, for example, in [33,42]). The improvement achieved with this smoothing procedure is shown in Fig. 7, where the corresponding results are represented with thick lines. The method has second order accuracy, as deduced from the average errors obtained using different grid sizes. For example, $|\bar{k} - k_{\text{exact}}|/k_{\text{exact}} = 8.04 \times 10^{-3}$, 2.23×10^{-3} and 3.07×10^{-4} for $d/\Delta x = 12.8$, 25.6 and 51.2, respectively.

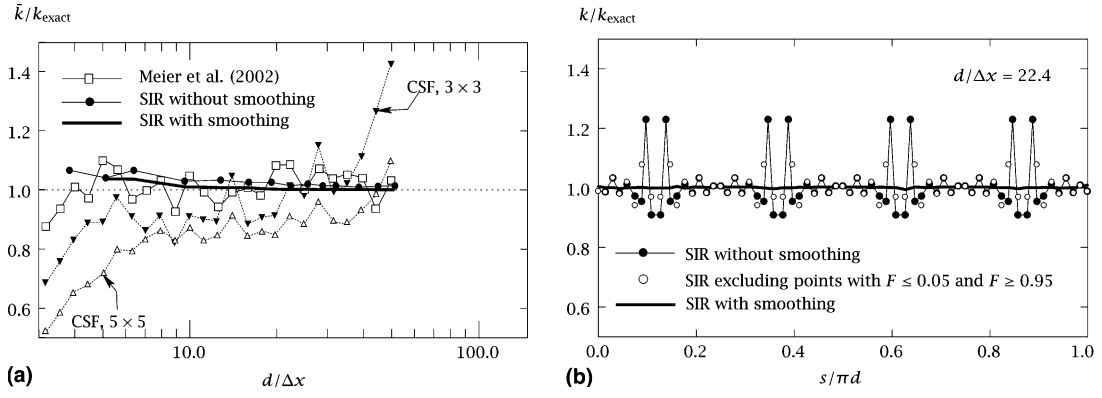

Fig. 7. Comparison between results for the curvature of a circular interface obtained with the proposed SIR algorithm (with and without smoothing) and those obtained by Meier et al. [39] using an adjusted estimator function and the CSF method of Brackbill et al. [40] with kernel matrices of sizes 3×3 and 5×5 to smooth the F distribution: (a) normalized mean curvature as a function of the interface diameter made dimensionless with the grid size, $d/\Delta x$; (b) normalized curvature as a function of the dimensionless arc length along the interface, $s/\pi d$, for a case with $d/\Delta x = 22.4$.

The extension of the proposed spline-based reconstruction algorithm to 3D is not straightforward and requires reformulation of the algorithm. As with the advection algorithm, this will be the subject of future work. However, it is worth making a few comments on the outlook of the method for 3D. Although the parametrization proposed in Eq. (11) cannot be applied directly to the extension to 3D, the algorithm proposed for determining the interface normal is amenable to the use of efficient geometric tools in 3D. As discussed by Ginzburg and Wittum [29], the geometric tasks needed to reconstruct the interface as spline interpolant can be facilitated by using parametric non-uniform rational B-spline (NURBS) surfaces (see, for example, [33,43]). One of the main advantages of the proposed VOF method is that it does not require special procedures to handle topological changes of the interface and so, in this respect, its extension to 3D is easier than in the case of front tracking or marker methods, which involve greater algorithmic complexity and which are not so robust when dealing with interface merging and breakup. On the other hand, among other advantages, front tracking methods generally need lower resolution to maintain the accuracy of front capturing methods. Three-dimensional (3D) interface representation has been successfully implemented by Tryggvason and others (see, for example, [44,45]) in front tracking methods, and some of the ideas used with these methods in three dimensions could also be useful in the implementation of the proposed reconstruction algorithm in 3D. Compared with level set methods, the extension of the proposed method to 3D is much more complicated, but, on the other hand, it is superior as regards local volume conservation and compact interface width. It should also be mentioned that the algorithm proposed to determine the interface normal is non-iterative, which makes it more efficient in 2D than other iterative and less accurate VOF algorithms, which, as mentioned by Rider and Kothe [9], can be prohibitively expensive in 3D.

4. Discussion of results

In this section, the accuracy and order of convergence of the proposed advection and reconstruction algorithms are analyzed using different numerical tests, for which many results obtained with different methods are available in the literature. 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 tests correspond to the advection of a circular fluid body in simple translational and rotational flows. Then, the capacity of the method to handle interfaces with high curvature is analyzed with the Zalesak slotted disk test, and, finally, the time-reversed single vortex test of Rider and Kothe [9] is used to investigate the efficiency of the method in handling thin filaments in a flow of non-uniform vorticity. Since advection errors are almost always accompanied by reconstruction errors, in order to better assess the performance of the advection algorithm proposed in this work and its relative merits compared with other algorithms, some of the tests presented in this section have been carried out using the same reconstruction methods as used in the compared simulations. The results obtained in the different tests are compared with those of other authors, in particular Rider and Kothe [9], Harvie and Fletcher [13] and Scardovelli and Zaleski [22].

4.1. Accuracy of the reconstruction

In this section, the accuracy of the proposed reconstruction algorithm in reproducing the interface geometries shown in Fig. 8 is compared with that obtained with other algorithms. For this purpose, we implemented some common reconstruction methods employed in VOF schemes. Thus, we considered a version of Youngs' method extended to general grids, as proposed by Kothe et al. [27], in which the interface normal is calculated by solving a weighted least squares problem; a method taken from [22], based on using central difference gradient of the volume fraction to calculate the interface normal; the iterative second-order method proposed by Puckett [17], and the algorithm ELVIRA. The last two were implemented following the indications given in [13,15], respectively. For all these methods, Table 1 shows the

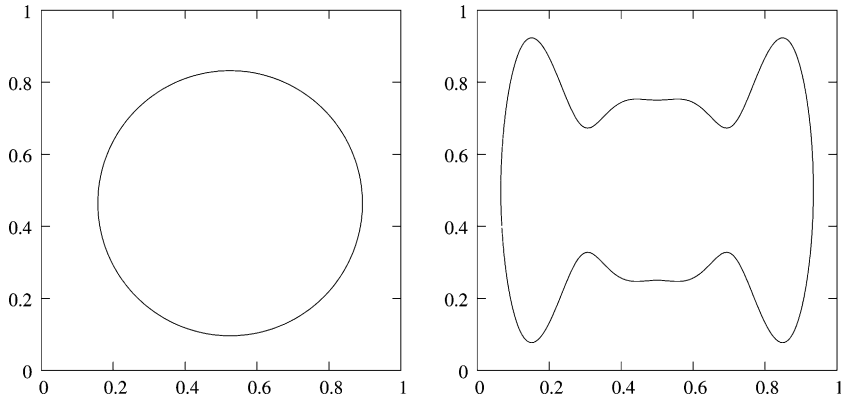

Fig. 8. Interface shapes used to assess the performance of the proposed reconstruction algorithm.

Table 1
Reconstruction errors for a circle and the interface shape of Eq. (14) using different grid sizes and reconstruction algorithms

	Reconstruction algorithm		Youngs	Central difference	Puckett	ELVIRA	SIR
Circle	10×10	E	2.09×10^{-3}	2.34×10^{-3}	3.36×10^{-3}	2.94×10^{-3}	1.94×10^{-3}
		$\mathcal{O}$	1.79	2.06	2.27	2.24	2.03
	20×20	E	6.06×10^{-4}	5.61×10^{-4}	6.96×10^{-4}	6.22×10^{-4}	4.76×10^{-4}
		$\mathcal{O}$	1.18	1.89	2.05	1.98	1.88
	40×40	E	2.67×10^{-4}	1.51×10^{-4}	1.68×10^{-4}	1.58×10^{-4}	1.29×10^{-4}
		$\mathcal{O}$	1.31	1.88	1.94	1.93	2.15
	80×80	E	1.08×10^{-4}	4.10×10^{-5}	4.38×10^{-5}	4.15×10^{-5}	2.91×10^{-5}
		$\mathcal{O}$	1.03	1.81	1.95	2.05	1.89
	160×160	E	5.29×10^{-5}	1.17×10^{-5}	1.13×10^{-5}	1.00×10^{-5}	7.86×10^{-6}
Interface shape of Eq. (14)	10×10	E	2.73×10^{-2}	2.75×10^{-2}	2.57×10^{-2}	2.78×10^{-2}	2.65×10^{-2}
		$\mathcal{O}$	2.23	2.26	2.59	2.97	3.22
	20×20	E	5.81×10^{-3}	5.74×10^{-3}	4.28×10^{-3}	3.54×10^{-3}	2.84×10^{-3}
		$\mathcal{O}$	2.63	2.78	2.19	2.02	2.03
	40×40	E	9.39×10^{-4}	8.36×10^{-4}	9.37×10^{-4}	8.71×10^{-4}	6.97×10^{-4}
		$\mathcal{O}$	2.13	2.47	2.40	2.44	2.69
	80×80	E	2.17×10^{-4}	1.51×10^{-4}	1.77×10^{-4}	1.61×10^{-4}	1.08×10^{-4}
		$\mathcal{O}$	1.41	1.91	2.01	1.98	1.99
	160×160	E	8.12×10^{-5}	4.00×10^{-5}	7.33×10^{-5}	4.08×10^{-5}	2.71×10^{-5}

reconstruction errors and the order of convergence for a circle of radius 0.368, which, in order to avoid artificial regularity of the results, is centered at (0.525,0.464) in a domain of 1×1, as in the work of Rider and Kothe [9], and for a more complex shape which may be represented by the Cartesian equation

$$\cos(6\pi x)(4x-2)^2 + (4y-2)^2 = 1. \tag{14}$$

The reconstruction error is defined as the area between the exact interface and its approximate representation, and the order of convergence can be determined from

$$\mathcal{O} = \frac{\ln\left(E_{2n_x}/E_{n_x}\right)}{\ln(1/2)}, \tag{15}$$

where E_{n_x} and E_{2n_x} are the errors obtained using two different grids with n_x and $2n_x$ cells along a direction. It can be observed that the results obtained with the proposed SIR algorithm are the most

accurate. For the complex shape of Eq. (14) and coarse grids, the convergence rate of all methods is somewhat higher. For the finest grid, all methods reach a nearly second-order convergence rate, except, as expected, Youngs' method. In general, the errors obtained with the ELVIRA algorithm were slightly lower than those obtained with Puckett's iterative method. This was also found in [15] for different translation and rotation tests similar to those presented in the following sections. We shall also show below that the ELVIRA algorithm requires a computational effort lower than that needed by Puckett's method.

Rider and Kothe [9] developed the same type of study for the circular shape, and compared the accuracy of three different methods: Youngs' method, an implementation of Puckett's iterative method based on the work of Pilliod [18], and an iterative second-order method based on the work of Swartz [46]. Although at the cost of a higher computational time, the last method produced results that are very close to those obtained with the SIR algorithm for the simple circular shape. For the other two methods, the differences between the results of Rider and Kothe [9] and those presented in Table 1 are small.

4.2. Simple translation test

Harvie and Fletcher [13] carried out a detailed discussion on the errors resulting from reconstruction and from cell boundary flux calculations. One of the tests they considered for this purpose was the translation of a circle of fluid of radius 0.2, initially located at (0.25, 0.25), by a steady and uniform flow with velocity components (1,1), during a time $T = 0.5$, in a domain of 1×1. The error was estimated with an L_1 norm defined as

$$E = \sum_{i,j} A_{\Omega}^{(i,j)} \left| F_{t=T}^{(i,j)} - F_{\mathrm{e}}^{(i,j)} \right|, \tag{16}$$

where $A_{\Omega}^{(i,j)}$ is the area of the cell (i, j), and $F_{t=T}^{(i,j)}$ and $F_{\mathrm{e}}^{(i,j)}$ are, respectively, the calculated and exact liquid fractions corresponding to the final instant of the test. Notice that, for the simple velocity field considered in the test, the contribution of the advection step to the error is negligible, and so the total error is solely due to reconstruction. On the other hand, the reconstruction error should vanish as the Courant number tends to one and remain bounded as the Courant number approaches zero [13]. This dependence of the error on the Courant number is illustrated in Fig. 9, where the results obtained in the present work using the proposed advection algorithm and both Puckett's and the proposed spline-based reconstruction algorithms are compared with those obtained by Harvie and Fletcher [13] using Puckett's method, for three different grid sizes. We have also represented the results obtained with the coarsest grid using our implementation of the advection algorithm of Rider and Kothe [9], which, as expected (see Fig. 3), in this test gives almost exactly the same result as ours. Note that the agreement between the results obtained with the proposed advection algorithm and those obtained by Harvie and Fletcher [13] (using in both cases Puckett's reconstruction method) is very good. The differences existing in the case of coarse grids may probably be attributed to the limitations pointed out in [13], which arise when the streamlines are aligned with the interface (see Fig. 1(b)). Also note that the spline-based reconstruction method substantially increases the accuracy of the results. For the finer grid and low CFL values, the orders of convergence of the proposed and Puckett's methods are of 1.8 and 1.6, respectively.

4.3. Rotation test

We consider the same problem configuration used by Rider and Kothe [9], in which a circle of fluid of radius 0.15, initially centered at (0.5, 0.75), rotates around the center of a domain of size 1×1 in a uniform

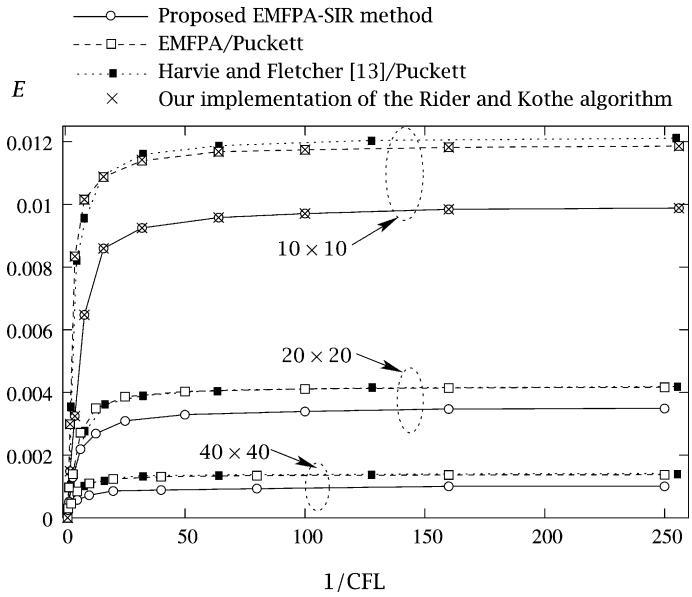

Fig. 9. L_1 error norm as a function of the CFL number for the simple translation of a circular fluid body test, obtained with different advection and reconstruction algorithms.

vorticity field. A constant time step, which corresponds to a CFL parameter based on the maximum velocity in the domain of approximately 0.5, is used. In the exact solution, the circle of fluid obviously keeps its shape, and so this test is especially appropriate for assessing the efficiency of the proposed advection algorithm. Notice that, when the interface undergoes a large distortion, the accuracy of the whole method relies to a large extent on the reconstruction algorithm, which may drastically reduce the order of convergence, especially when the size of small fluid structures approaches the cell size. The error in the numerical solution will be defined as in Eq. (16), with T being the time the circle of fluid takes to complete a revolution (which, for a grid of 32×32 cells, for example, corresponds to 201 time steps). The order of convergence can be determined from Eq. (15).

The values of E and $\mathcal{O}$ obtained with the proposed EMFPA algorithm for grids of 32×32, 64×64 and 128×128 cells are compared in Table 2 with those obtained by Rider and Kothe [9], using in both cases the interface reconstruction method proposed by Puckett [17]. It can be observed that, for this simple advection

Table 2
L_1 error norm and order of convergence, as defined by Eqs. (15) and (16), obtained in the rotation of a circular fluid body test using different advection and reconstruction methods

Advection Reconstruction		Rider and Kothe [9] Puckett	EMFPA Puckett	EMFPA SIR	EMFPA Exact circle center
32×32	E	1.61×10^{-3}	1.42×10^{-3}	8.68×10^{-4}	4.08×10^{-4}
	$\mathcal{O}$	2.19	2.08	1.76	1.78
64×64	E	3.54×10^{-4}	3.37×10^{-4}	2.57×10^{-4}	1.19×10^{-4}
	$\mathcal{O}$	1.98	2.08	2.22	1.84
128×128	E	8.95×10^{-5}	7.96×10^{-5}	5.52×10^{-5}	3.32×10^{-5}

problem, the proposed method exhibits a second-order convergence rate and produces errors appreciably lower than those obtained by Rider and Kothe [9]. These results are also compared with those obtained using the combined EMFPA–SIR algorithms proposed in this work. Note that there is a further and more appreciable reduction in the error when the proposed reconstruction method is used instead of Puckett's method. In the last column of Table 2, we have also included the results obtained with the proposed advection algorithm and a reconstruction algorithm based on determining the interface normal by joining the center of each cell interface segment with the exact location of the circle center at each time step, and using an iterative procedure starting with an initial segment orientation calculated using Youngs' method. Note that, for this test, this reconstruction procedure provides the maximum possible performance of PLIC methods.

4.4. Zalesak slotted disk test

In this test, a slotted circle of fluid rotates in a uniform vorticity field [47]. The same domain as in [8] and a grid with 200 cells along each coordinate were used. The time step was chosen so as to make the circle complete a revolution at the instant $T = 2524\Delta t$, which corresponds to a CFL parameter, based on the maximum velocity in the domain, of approximately 0.25. The test ends when the first (anticlockwise) revolution is completed. The error is defined as

$$E = \frac{\sum_{i,j} \left| F_{\mathrm{T}}^{(i,j)} - F_0^{(i,j)} \right|}{\sum_{i,j} F_0^{(i,j)}}, \tag{17}$$

where $F_0^{(i,j)}$ and $F_{\mathrm{T}}^{(i,j)}$ are the fluid fractions in the cell (i, j) corresponding to the initial and final instants, respectively. The errors in the results obtained using the proposed advection algorithm, EMFPA, and three different reconstruction methods, including the new proposed method, SIR, are compared in Table 3 with those reported by other authors. Note that the proposed advection algorithm produces errors slightly lower than those obtained with the Stream algorithm and the same interface reconstruction method. Only two of the very recent three reconstruction methods proposed by Scardovelli and Zaleski [22] included in Table 3, which are based on a quadratic least-square technique and approximate the interface in each cell using two

Table 3
Errors, as defined in Eq. (17), obtained in the slotted disk rotation test using different advection and reconstruction algorithms

Advection/reconstruction algorithms	E (Eq. (17))
SLIC	8.38×10^{-2}
Hirt–Nichols	9.62×10^{-2}
FCT–VOF	3.29×10^{-2}
Youngs	1.09×10^{-2}
Harvie and Fletcher [13] (Stream)/Youngs	1.07×10^{-2}
Harvie and Fletcher [13] (Stream)/Puckett	1.00×10^{-2}
Harvie and Fletcher [14] (DDR)/Youngs	1.56×10^{-2}
Harvie and Fletcher [14] (DDR)/Puckett	1.50×10^{-2}
Proposed EMFPA/Youngs	1.06×10^{-2}
Proposed EMFPA/Puckett	9.73×10^{-3}
Proposed EMFPA–SIR	8.74×10^{-3}
Scardovelli and Zaleski [22] (linear least-square fit)	9.42×10^{-3}
Scardovelli and Zaleski [22] (quadratic fit)	5.47×10^{-3}
Scardovelli and Zaleski [22] (quadratic fit + continuity)	4.16×10^{-3}

Results for the four first methods are taken from [8].

consecutive segments, produce errors lower than that obtained with the reconstruction algorithm proposed here, although at the cost of a higher computational time, as discussed below. Less accurate reconstruction methods (see, for example, the three first cases of Table 3) produce errors that may be up to one order of magnitude larger. This test is indeed more appropriate for assessing the efficiency of the reconstruction method than that of the advection algorithm. Fig. 10 shows that the interface location error is concentrated at regions of high interface curvature. That the error in these regions largely represents the main contribution to the total error is confirmed by comparing the errors obtained with the proposed algorithm, given in Table 3, with those obtained for a non-slotted circle of fluid using the same algorithm and test conditions, which were found to be equal to 1.95×10^{-3} and 7.33×10^{-4}, respectively, when Youngs' and Puckett's reconstruction methods were used.

4.5. Vortex-in-a-box test

In this test, introduced by Bell et al. [48], a circle of fluid of radius 0.15, initially centered at (0.5, 0.75), is advected in a domain of 1×1, where the solenoidal velocity distribution given by the stream function $\psi = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y)$ is imposed. All the results presented in this section were obtained using a CFL parameter (based on the maximum velocity component in the domain) equal to one.

Figs. 11(a) and (b) show, for different instants and grid sizes of 32×32 and 128×128 cells, the area occupied by the fluid in every cell of the interface (where $0 < F < 1$) obtained using the proposed advection method and, respectively, Puckett's reconstruction method and the one proposed in this work. The typical fragmentation of the fluid filament, which tends to occur when its thickness becomes close to the cell size, can be observed in the last instants represented in Fig. 11, particularly for the coarser grid. The results of Fig. 11(a) are consistent with those obtained by Rider and Kothe [9] and Harvie and Fletcher [13], and slightly improve the accuracy of those obtained by Rider and Kothe [9] for the same test conditions, which can be attributed to the better performance of the new EMFPA advection algorithm. The qualitative improvement introduced by the proposed reconstruction method with respect to Puckett's method can be observed by comparing Figs. 11(a) and (b).

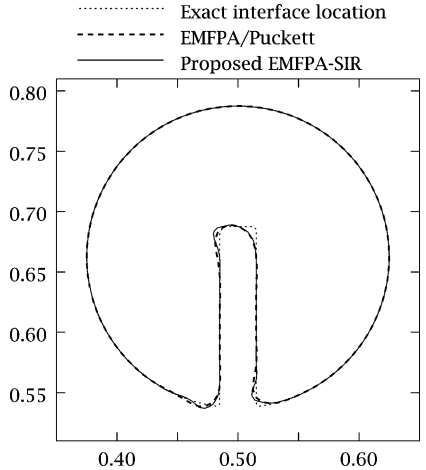

Fig. 10. Interface profiles obtained in the slotted-disk test of Zalesak [47] using the proposed EMFPA advection algorithm in combination with Puckett's and the proposed SIR reconstruction algorithms.

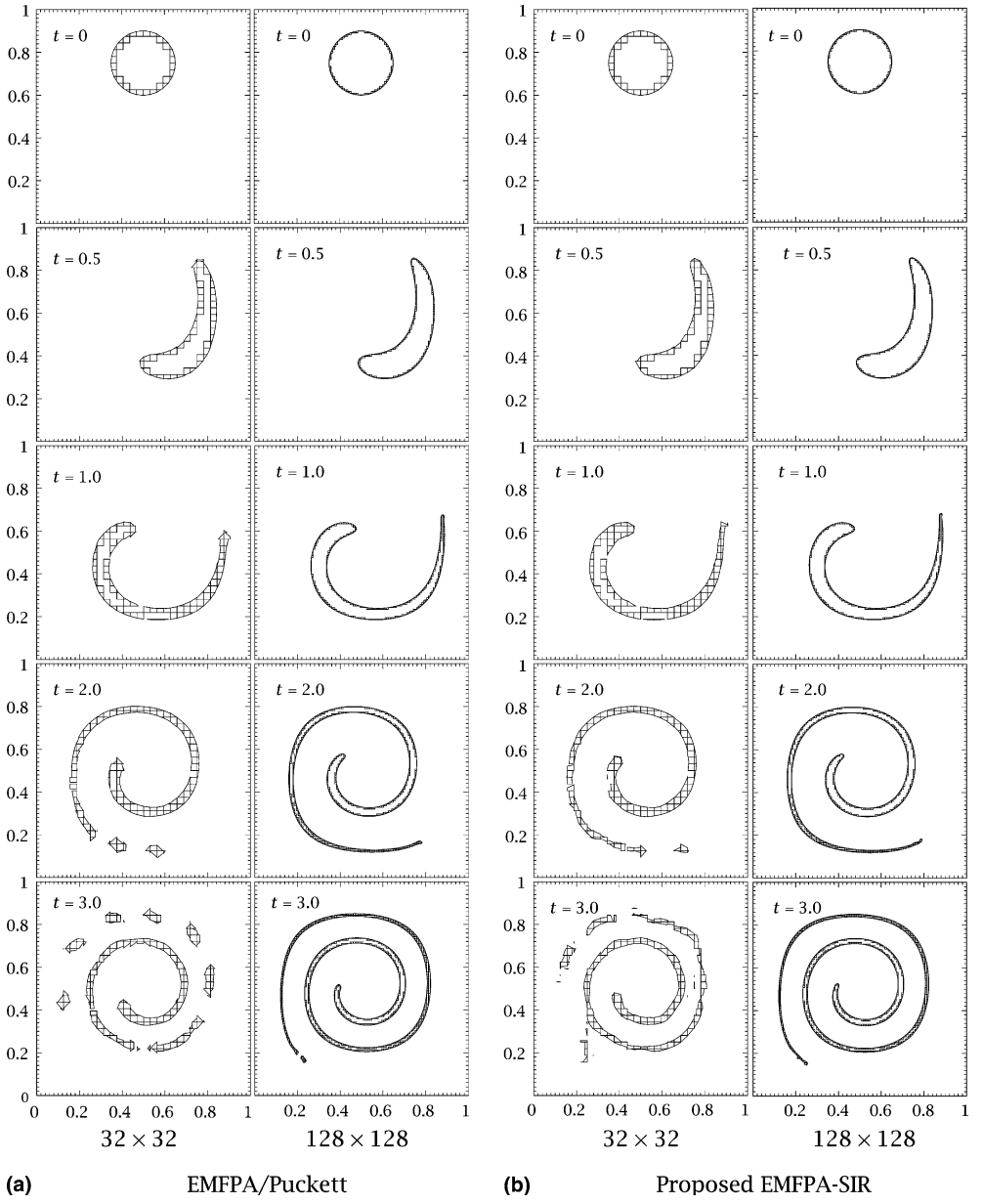

Fig. 11. Results obtained in the single vortex test using the proposed advection algorithm, two different reconstruction algorithms and two grid sizes: (a) Puckett's algorithm; (b) SIR algorithm proposed in this work.

To quantify the accuracy of the proposed method, the time-reversed single vortex test used by Rider and Kothe [9] was carried out, in which the above stream function is multiplied by the factor $\cos(\pi t/T)$. The location error at $t = T$ and order of convergence (as defined by Eqs. (15) and (16)) obtained with the proposed advection algorithm are compared in Table 4 with those obtained by Rider and Kothe [9] and Harvie and Fletcher [13] for two values of T, three grid sizes and Youngs' and Puckett's reconstruction

Table 4
L_1 error norm and order of convergence, as defined by Eqs. (15) and (16), obtained for two values of T and different grid sizes in the time-reversed single vortex test, using different advection methods and Youngs' and Puckett's reconstruction methods

Reconstruction algorithm	Grid size	$T = 0.5$		$T = 2.0$	
		E	$\mathcal{O}$	E	$\mathcal{O}$
Youngs	*Harvie and Fletcher* [13]				
	32×32	3.42×10^{-4}		2.49×10^{-3}	
			0.99		1.82
	64×64	1.72×10^{-4}		7.06×10^{-4}	
			0.84		1.66
	128×128	9.60×10^{-5}		2.23×10^{-4}	
	Proposed EMFPA				
	32×32	2.29×10^{-4}		2.32×10^{-3}	
			0.51		1.79
	64×64	1.61×10^{-4}		6.72×10^{-4}	
			0.84		1.58
	128×128	8.99×10^{-5}		2.24×10^{-4}	
Puckett	*Rider and Kothe* [9]				
	32×32	7.29×10^{-4}		2.36×10^{-3}	
			2.36		2.01
	64×64	1.42×10^{-4}		5.85×10^{-4}	
			1.86		2.16
	128×128	3.90×10^{-5}		1.31×10^{-4}	
	Modified Rider and Kothe algorithm				
	32×32	5.32×10^{-4}		2.47×10^{-3}	
			2.21		2.15
	64×64	1.15×10^{-4}		5.57×10^{-4}	
			1.7		2.06
	128×128	3.54×10^{-5}		1.34×10^{-4}	
	Harvie and Fletcher [13]				
	32×32	5.51×10^{-4}		2.37×10^{-3}	
			2.32		2.07
	64×64	1.10×10^{-4}		5.65×10^{-4}	
			1.71		2.10
	128×128	3.38×10^{-5}		1.32×10^{-4}	
	Proposed EMFPA				
	32×32	4.45×10^{-4}		2.14×10^{-3}	
			2.48		1.99
	64×64	7.99×10^{-5}		5.39×10^{-4}	
			1.97		2.06
	128×128	2.04×10^{-5}		1.29×10^{-4}	

methods. Table 5 shows the same type of results obtained by Garrioch and Baliga [21], Scardovelli and Zaleski [22] and Aulisa et al. [26], along with those obtained using the proposed advection and reconstruction algorithms, EMFPA–SIR. The results of Garrioch and Baliga [21] were obtained using a multidimensional advection algorithm similar to that of Pilliod and Puckett [15], with some modifications to ensure strict mass conservation.

It can be observed from Table 4 that, for both Youngs' and Puckett's reconstruction algorithms, the error is always lowest when the proposed advection method is used, except in the case of $T = 2$ and a grid

Table 5
L_1 error norm and order of convergence, as defined by Eqs. (15) and (16), obtained for two values of T and different grid sizes in the time-reversed single vortex test, using different advection and reconstruction methods and the hybrid method of [26]

	Grid size	$T = 0.5$		$T = 2.0$	
		E	$\mathcal{O}$	E	$\mathcal{O}$
Garrioch and Baliga [21]	*Conservative Pilliod and Puckett* [15] *advection/circle-fit*				
	32×32	–		2.23×10^{-3}	
					2.18
	64×64	–		4.93×10^{-4}	
					2.14
	128×128	–		1.12×10^{-4}	
Scardovelli and Zaleski [22]	*EI–LE advection/linear least-square fit*				
	32×32	–		1.75×10^{-3}	
					1.91
	64×64	–		4.66×10^{-4}	
					2.19
	128×128	–		1.02×10^{-4}	
	EI–LE advection/quadratic least-square fit				
	32×32	–		1.88×10^{-3}	
					2.08
	64×64	–		4.42×10^{-4}	
					2.24
	128×128	–		9.36×10^{-5}	
	EI–LE advection/quadratic least-square fit + continuity				
	32×32	–		1.09×10^{-3}	
					1.96
	64×64	–		2.80×10^{-4}	
					2.29
	128×128	–		5.72×10^{-5}	
Aulisa et al. [26]	*Hybrid markers and volume of fluid method*				
	32×32	–		1.00×10^{-3}	
					1.89
	64×64	–		2.69×10^{-4}	
					2.30
	128×128	–		5.47×10^{-5}	
Present work	*Proposed EMFPA–SIR*				
	32×32	2.07×10^{-4}		8.62×10^{-4}	
			1.95		1.86
	64×64	5.36×10^{-5}		2.37×10^{-4}	
			2.11		2.08
	128×128	1.24×10^{-5}		5.62×10^{-5}	

size of 128×128 cells, for which all the advection algorithms, for a given reconstruction method, produce similar results. It should be pointed out that the results of Harvie and Fletcher [13] were obtained using 100 donating regions at each cell face, which significantly increases the computational cost. The advantages of the proposed advection algorithm are more appreciable in the test where $T = 0.5$, for which the reconstruction errors are expected to be lower.

We have also included in Table 4 the results obtained by implementing a modified version of the method used by Rider and Kothe [9] to construct the flux polygons, in which the polygon edge (3–4) parallel to the corresponding cell face (1–2) (see Fig. 1(a)) is appropriately displaced to enforce the area conservation constraint of the Eq. (4) (solved using a trapezoidal rule), which avoids the need for a subsequent divergence correction. It can be observed from Table 4 that this modified version of the Rider and Kothe [9] algorithm produces results similar to those of the original algorithm, although a slight improvement is obtained for cases with $T = 0.5$, particularly for coarse grids. The further improvement in accuracy that is obtained with the EMFPA algorithm, is mainly due to the use of matched flux polygons.

It can be observed from Table 5 that the proposed EMFPA–SIR method gives the lowest error except for the hybrid method of Aulisa et al. [26] with a grid size of 128×128. As shown below, this better behavior of the hybrid method of [26] is more pronounced in cases with higher interface deformation, and can be attributed to the sub-grid resolution provided by the representation of the interface with Lagrangian particles. It should also be pointed out that all the results in Tables 4 and 5 were obtained with CFL number equal to one, and that the accuracy of this hybrid method in general may increase substantially with lower CFL numbers [26]. On the other hand, in the proposed VOF method, volume conservation is better than in the hybrid method (although the results reported in [26] are also very good in this respect, and only for coarse grids and high CFL numbers does the mass error become more appreciable), and topology changes can be handled in an easy way, particularly in 3D flows.

A comparison of the results obtained using the proposed advection algorithm, EMFPA, and Puckett's and SIR reconstruction algorithms (Tables 4 and 5, respectively), shows that the spline-based interface reconstruction algorithm proposed in this work substantially improves the accuracy of Puckett's algorithm, which, in turn, reduces considerably the errors obtained with Youngs' algorithm. This can also be qualitatively observed from Fig. 12, which presents a comparison between the results for the interface location at $t = T$, obtained in the test for $T = 2$ and a grid of 32×32 cells.

Table 6 shows that the relative CPU time required for this test by the proposed SIR algorithm is around 65% lower than that required by the iterative Puckett method, and around 10% higher than that needed for Youngs' method (using in all cases the EMFPA algorithm). Relative CPU times required by the central difference and ELVIRA reconstruction algorithms are also shown in Table 6. According to Scardovelli and Zaleski [22], their quadratic least-square reconstruction algorithm with added continuity

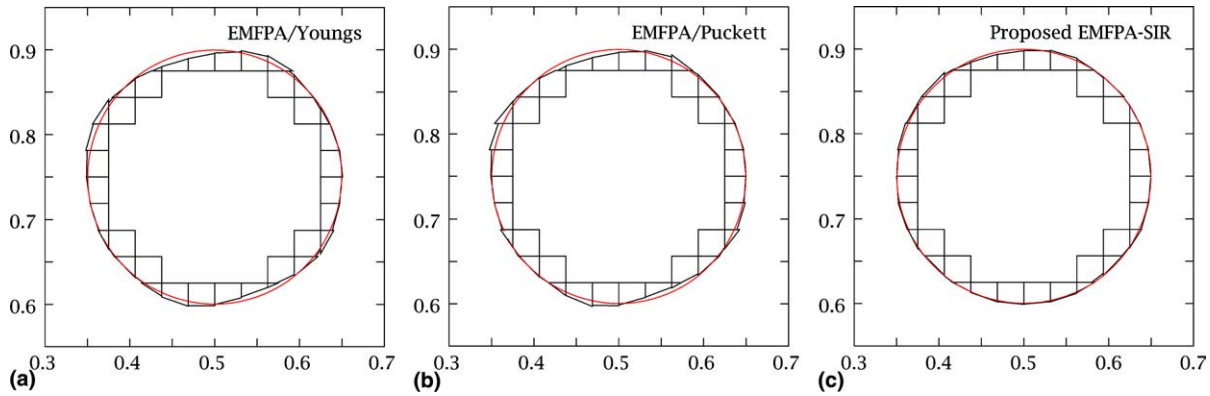

Fig. 12. Interface location obtained at $t = T$ in the time-reversed single vortex test, with $T = 2$ and a grid of 32×32 cells, using the proposed advection algorithm and three reconstruction algorithms: (a) Youngs' algorithm; (b) Puckett's algorithm; (c) proposed SIR.

Table 6
Relative CPU time required by the proposed EMFPA algorithm and different reconstruction methods and the corresponding L_1 error norm, for the time-reversed single vortex test with $T = 2$ and a grid of 32×32 cells

Reconstruction algorithm	Youngs	Central difference	Puckett	ELVIRA	Proposed SIR
Relative CPU time	1.00	1.01	3.27	2.19	1.10
Error, Eq. (16)	2.32×10^{-3}	2.31×10^{-3}	2.14×10^{-3}	2.22×10^{-3}	8.62×10^{-4}

Table 7
L_1 error norm, as defined by Eq. (16), obtained for $T = 8$ and a grid of 128×128 cells in the time-reversed single vortex test, using different advection and reconstruction methods and the hybrid method of [26]

Vortex in a box ($T = 8$)	
Advection/reconstruction algorithms	Error, Eq. (16)
Rider and Kothe [9]/Puckett	1.44×10^{-3}
Harvie and Fletcher [13]/Youngs	2.16×10^{-3}
Harvie and Fletcher [13]/Puckett	1.18×10^{-3}
Proposed EMFPA/Youngs	2.13×10^{-3}
Proposed EMFPA/Puckett	1.17×10^{-3}
Proposed EMFPA–SIR	7.57×10^{-4}
Aulisa et al. [26], hybrid markers/VOF	4.78×10^{-4}

(whose results for this test, using the EI–LE advection algorithm, are slightly less accurate than those obtained with EMFPA–SIR), is twice as time consuming as Youngs' method, whereas for the proposed SIR algorithm the relative CPU time is as low as 1.1. However, despite this good behavior of the proposed algorithm compared with others as regards the required computational effort, and although an efficient method is desirable, the time consumed in tracking the interface may not always be of great importance when solving the Navier–Stokes equations simultaneously. Robustness and accuracy are usually more important than efficiency.

In Table 7 we compare the errors obtained in a test with $T = 8$, a grid size of 128×128 and a maximum CFL number equal to 1, using the proposed EMFPA–SIR method and other methods. It can be observed that the proposed method produces the lowest error among the non-hybrid VOF methods. Only the error produced by the hybrid markers and VOF method is lower. It should be pointed out that, as in the other tests presented in this work, it was not necessary to use any redistribution algorithm to alleviate the very small wisps that appeared in the few cases in which the flux polygon area had to be readjusted to satisfy the area-conservation constraint of Eq. (4). The significant improvement in accuracy provided by the proposed reconstruction method can be visually observed from Fig. 13, where the interface locations obtained at $t = T/2$, $3T/4$ and T using the last four methods of Table 7 are represented. Note the significant reduction in numerical surface tension effects obtained with the advection and reconstruction methods proposed in this work.

A comparison of the proposed method with the hybrid particle level set method of Enright et al. [49] cannot be made directly because of the different interface location error metrics used. Enright et al. [49] mentioned that the L_1 errors reported in their paper compare favorably with those reported by Rider and Kothe [9], which, in turn, are improved on by the proposed method. A qualitative comparison shows that the results for the interface location of the hybrid method are slightly better than those obtained with the proposed method, although a considerably larger area is lost (0.7% in the case of a 128×128 grid and $T = 8$).

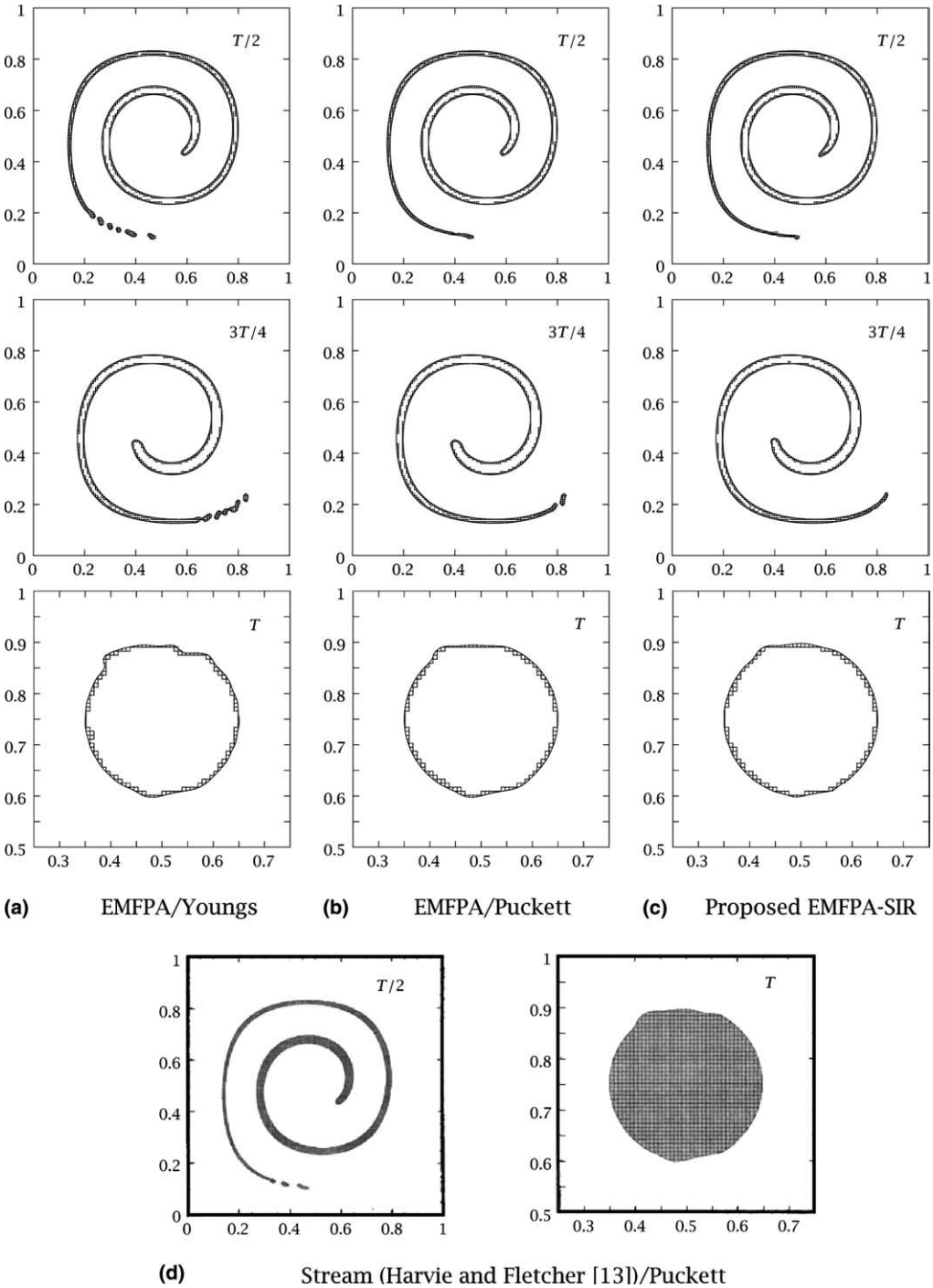

Fig. 13. Results obtained in the time-reversed single vortex test, for $T = 8$ and a grid of 128×128 cells, using the proposed EMFPA advection algorithm and three reconstruction algorithms: (a) Youngs' algorithm; (b) Puckett's algorithm; (c) proposed SIR; (d) the Stream scheme of Harvie and Fletcher [13] and Puckett's algorithm.

5. Conclusions

We proposed a volume of fluid method which introduces a new multidimensional procedure to integrate the volume fraction advection equation and a new spline-based reconstruction algorithm. The proposed advection algorithm, which is based on constructing edge-matched flux polygons at cell faces, locally enforces mass conservation and does not require any posterior divergence correction. The new reconstruction algorithm uses a cubic spline curve to represent the interface, which allows an accurate determination of the interface curvature. Different tests have demonstrated the accuracy and efficiency of the proposed method, which compare very favorably with other existing VOF methods. It has been found that the proposed advection algorithm tends to reduce the formation of over/undershoots and wisps, thus diminishing the need for using local redistribution algorithms, and increases the accuracy of similar algorithms without increasing the computational effort. The new reconstruction algorithm hinders the formation of jetsam, and shows an accuracy which is substantially higher than that of most existing algorithms of comparable complexity, and similar to that of more complex algorithms proposed recently, which require considerably longer computational times.

Acknowledgements

The authors gratefully acknowledge the support of the Spanish Ministerio de Ciencia y Tecnología under grants PB98-0007 and DPI2001-1390-C02.

References

[1] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[2] R. Debar, Fundamentals of the KRAKEN Code, Technical Report UCIR-760, Lawrence Livermore National Laboratory, 1974.
[3] W.F. Noh, P.R. Woodward, SLIC (simple line interface calculation), Lect. Notes Phys. 59 (1976) 330–340.
[4] A.J. Chorin, Flame advection and propagation algorithms, J. Comput. Phys. 35 (1980) 1–11.
[5] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[6] 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.
[7] 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.
[8] M. Rudman, Volume-tracking methods for interfacial flow calculations, Int. J. Numer. Methods Fluids 24 (1997) 671–691.
[9] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112–152.
[10] D.J. Benson, Volume of fluid interface reconstruction methods for multi-material problems, Appl. Mech. Rev. 55 (2002) 151–165.
[11] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.
[12] S. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[13] D.J.E. Harvie, D.F. Fletcher, A new volume of fluid advection algorithm: the stream scheme, J. Comput. Phys. 162 (2000) 1–32.
[14] D.J.E. Harvie, D.F. Fletcher, A new volume of fluid advection algorithm: the defined donating region scheme, Int. J. Numer. Methods Fluids 35 (2001) 151–172.
[15] J.E. Pilliod, E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, Technical Report LBNL-40744, Lawrence Berkeley National Laboratory, 1997.
[16] D.L. Youngs, An interface tracking method for a 3D Eulerian hydrodynamics code, Technical Report 44/92/35, AWRE, 1984.
[17] 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.
[18] J.E. Pilliod Jr., An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods, Master's Thesis, University of California at Davis, 1992.

[19] A.J. Chorin, Curvature and solidification, J. Comput. Phys. 58 (1985) 472–490.
[20] 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.
[21] 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, Louisiana, 2001.
[22] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split Eulerian–Lagragian advection, Int. J. Numer. Methods Fluids 41 (2003) 251–274.
[23] J. O'Rourke, Computational Geometry in C, Cambridge University Press, Cambridge, 1993.
[24] G. Černe, S. Petelin, I. Tiselj, Numerical errors of the volume-of-fluid interface tracking algorithm, Int. J. Numer. Methods Fluids 38 (2002) 329–350.
[25] G. Černe, S. Petelin, I. Tiselj, Coupling of the interface tracking and the two-fluid models for the simulation of incompressible two-phase flow, J. Comput. Phys. 171 (2001) 776–804.
[26] 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.
[27] 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.
[28] 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, California, 1999.
[29] 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.
[30] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran, Cambridge University Press, Cambridge, UK, 1986.
[31] 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.
[32] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, J. Comput. Phys. 164 (2000) 228–237.
[33] G. Farin, Curves and Surfaces for Computer-Aided Geometric Design, A Practical Guide, fourth ed., Academic Press, San Diego, 1997.
[34] M.W. Williams, D.B. Kothe, E.G. Puckett, Convergence and accuracy of kernel-based continuum surface tension models, Technical Report LA-UR-98-2268, Los Alamos National Laboratory, invited paper of the Chia-Shuh Yih Memorial Symposium, in: Proceedings of the Thirteenth U.S. National Congress of Applied Mechanics, Gainesville, FL, 1998.
[35] M. Sussman, A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles, J. Comput. Phys. 187 (2003) 110–136.
[36] J. Helmsen, P. Colella, E.G. Puckett, Non-convex profile evolution in two dimensions using volume of fluids, Technical Report LBNL-40693, Lawrence Berkeley National Laboratory, 1997.
[37] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, J. Comput. Phys. 183 (2002) 400–421.
[38] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, Int. J. Numer. Methods Fluids 30 (1999) 775–793.
[39] M. Meier, G. Yadigaroglu, B.L. Smith, A novel technique for including surface tension in PLIC–VOF methods, Eur. J. Mech. B – Fluids 21 (2002) 61–73.
[40] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335–354.
[41] I. Aleinov, E.G. Puckett, Computing surface tension with high-order kernels, in: Proceedings of the 6th International Symposium on Computational Fluid Dynamics, Lake Tahoe, CA, 1995.
[42] N. Sapidis, G. Farin, Automatic fairing algorithm for B-splines curves, Comput. Aided Des. 22 (1990) 121–129.
[43] L. Piegl, W. Tiller, The NURBS Book, Springer-Verlag, New York, 1997.
[44] 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.
[45] 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.
[46] B. Swartz, The second-order sharpening of blurred smooth borders, Math. Comput. 52 (1989) 675–714.
[47] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (1979) 335–362.
[48] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (1989) 257–283.
[49] 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.