



A geometrical predictor–corrector advection scheme and its application to the volume fraction function

A. Cervone^a, S. Manservigi^a, R. Scardovelli^{a,*}, S. Zaleski^b

^a DIENCA-Lab. di Montecuccolino, Università degli Studi di Bologna, Via dei Colli 16, 40136 Bologna, Italy

^b Institut Jean Le Rond d'Alembert (IJLRdA), UMR 7190, Université Pierre et Marie Curie – Paris 6 and CNRS, 4 Place Jussieu, 75252 Paris Cedex 05, France

ARTICLE INFO

Article history:

Received 9 November 2007

Received in revised form 8 August 2008

Accepted 18 September 2008

Available online 30 September 2008

Keywords:

VOF/PLIC method

Multidimensional Eulerian advection

Mass conservation

Incompressible flow

Flux polygons

ABSTRACT

We present a multidimensional Eulerian advection method for interfacial and incompressible flows in two-dimensional Cartesian geometry. In the scheme we advect the grid nodes backwards along the streamlines to compute the pre-images of the grid lines. These pre-images are approximated by continuous, piecewise-linear lines. The enforcement of the discrete version of the incompressibility constraint is a very important issue to determine correctly the flux polygons and to reduce considerably the integration, discretization and interpolation numerical errors. The proposed method compares favorably with other previous multidimensional advection methods as long as the initial interface line is well reconstructed. Conversely, we show that when the interface is very fragmented the total numerical error is completely dominated by the reconstruction error and in these conditions it is very difficult to assess which advection scheme is the most reliable one.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

Two-phase and free-surface flows are very common in natural processes and industrial applications. Direct numerical simulations of such flows present severe difficulties related to the discontinuity of physical properties, such as density and viscosity, across the interface and to the singularity of the capillary force concentrated on the interface. Many methods have been developed to deal directly with interfaces, such as the front-tracking method [1,2], the volume-of-fluid (VOF) method [3,4], the level-set (LS) method [5,6] and the phase-field method [7,8]. More recently, a number of methods that couple two different schemes have been proposed with the aim to combine the strengths of each method. Among others, we can mention the coupled level-set and VOF method [9,10], the hybrid particle-level-set method [11] and the mixed markers and volume-of-fluid method [12].

The VOF method is based on a characteristic function $\chi(\mathbf{x}, t)$ with value 1 in the region occupied by the reference phase and 0 where the other phase or vacuum is present. If the two fluids are assumed to be immiscible, the function χ does not change its value as it is passively advected by the flow and satisfies a standard advection equation

$$\frac{D\chi}{Dt} \equiv \frac{\partial\chi}{\partial t} + (\mathbf{v} \cdot \nabla)\chi = 0. \quad (1)$$

The discrete version of χ is the so-called color or volume fraction function C representing the fraction of each cell of the computational grid which is occupied by the reference phase. The numerical solution of (1) is far from being simple because the

* Corresponding author. Tel.: +39 051 64417 20; fax: +39 051 64417 47.

E-mail address: ruben.scardovelli@unibo.it (R. Scardovelli).

function χ is discontinuous across the interface. Standard algebraic schemes, such as upwind finite differences, tend to diffuse the interface, therefore geometrical methods are usually adopted to keep the interface sharp.

From the C data the interface is first reconstructed, i.e. an approximated $\tilde{\chi}(\mathbf{x}, t)$ distribution is found which satisfies

$$C_i = \int_{V_i} \tilde{\chi}(\mathbf{x}, t) dV, \tag{2}$$

where V_i and C_i are respectively the volume and the volume fraction of the cell i . In the piecewise-linear interface calculation (VOF/PLIC) the boundary of the $\tilde{\chi}$ distribution in each cell is defined by the linear equation $\mathbf{m} \cdot \mathbf{x} = \alpha$, where \mathbf{m} is the local interface normal vector and α a constant. The reconstructed interface is then piecewise-linear and in general not continuous across the boundary of adjacent cells. Reconstruction techniques mainly differ from each other in the way the normal vector \mathbf{m} is computed. In this paper we do not present a new reconstruction method, but use the two-dimensional Puckett's algorithm [13] in order to compare our results with those obtained using different advection algorithms but the same reconstruction method. Puckett's algorithm is based on a second-order iterative method, which minimizes a measure of the error between the actual volume fractions and those obtained by extending the approximate linear interface from the cell under investigation to the surrounding 3×3 block of cells. Detailed reviews and tests of other reconstruction procedures can be found in [3,14–16].

After the reconstruction, the interface can be advected in either an Eulerian or a Lagrangian fashion. The first approach has been used mainly for Cartesian grids and it requires a geometrical evaluation of the fluid-volume fluxes across the cell boundary, which in two dimensions are usually approximated as triangular, rectangular or trapezoidal flux polygons. This can be accomplished independently along each coordinate direction, with multidimensionality obtained with an operator-split technique [17,3,18,15], or with unsplit schemes that are more accurate but also geometrically characterized by more complex fluid fluxes [17,3,19–21]. In the coupled level-set and VOF method (CLSVOF) [9] the level-set function $\phi(\mathbf{x}, t)$ is given by the signed normal distance from the interface and follows the same advection equation for χ

$$\frac{\partial \phi}{\partial t} + (\mathbf{v} \cdot \nabla) \phi = 0. \tag{3}$$

Strictly speaking this equation is satisfied only by the interface line where $\phi = 0$. An Eulerian advection is performed with a split scheme, to update the values at the next discrete time of the level-set and volume fraction scalar functions, by computing the fluxes of χ and ϕ across the cell edges. Fluid-volume fluxes are rather involved in unstructured triangular cells, but Lagrangian–Eulerian advection methods, consisting in a Lagrangian advection of the interface and a projection onto the fixed grid, could be a promising alternative [22,23]. This technique has been coupled to a level-set method and extended to adaptive unstructured grids (ACLSVOF) [10]. In the hybrid particle-level-set method [11] an Eulerian level-set method has been combined with Lagrangian particles to perform interface reconstruction in under-resolved regions of the interface, while in the mixed markers and volume-of-fluid method [12] interfacial markers are used together with a VOF scheme. We should also mention the polygonal area mapping method, an interface tracking method, where the cell boundary is traced back along the streamlines to update the reference phase boundary via polygon-clippings [24].

In this paper we present an unsplit advection scheme in two-dimensional domains partitioned with a Cartesian mesh, where we combine both the Eulerian and Lagrangian approaches [25]. We compute the fluid fluxes across the cell boundary by enforcing the incompressibility condition as in a standard Eulerian advection, but the flux polygons are defined by advecting backwards in time the grid nodes along the streamlines [26]. The pre-image of the grid nodes and the flux polygons are computed with a geometrical predictor–corrector technique in order to define continuous piecewise-linear lines that in the time step Δt will be mapped to the grid lines. The technique is applied in this paper to the advection of the characteristic function χ , but it can be used to advect the signed distance function ϕ as well.

We begin in Section 2 with a quick review of a few unsplit advection algorithms for the volume fraction function and then discuss in Section 3 the geometrical predictor–corrector advection (GPCA) scheme. In Section 4 we present a few results and compare the performance of several advection schemes. Finally we present our conclusions.

2. Multidimensional Eulerian advection schemes

We consider the advection Eq. (1) for the characteristic function χ and add to both sides the term $\chi \nabla \cdot \mathbf{v}$, which is identically equal to zero if the flow is incompressible, to get

$$\frac{\partial \chi}{\partial t} + \nabla \cdot (\chi \mathbf{v}) = \chi \nabla \cdot \mathbf{v} = 0. \tag{4}$$

Furthermore, we consider a standard two-dimensional MAC grid with square cells of side h [27], where the discrete divergence-free condition becomes $u^+ - u^- + v^+ - v^- = 0$, as shown in Fig. 1(a). We multiply this expression by the time step $\Delta t = t^{n+1} - t^n$ and the cell side h in order to obtain the following relation among the fluid fluxes F that leave and enter the cell through its boundary

$$F_x^+ - F_x^- + F_y^+ - F_y^- = u^+ \Delta t h - u^- \Delta t h + v^+ \Delta t h - v^- \Delta t h = 0. \tag{5}$$

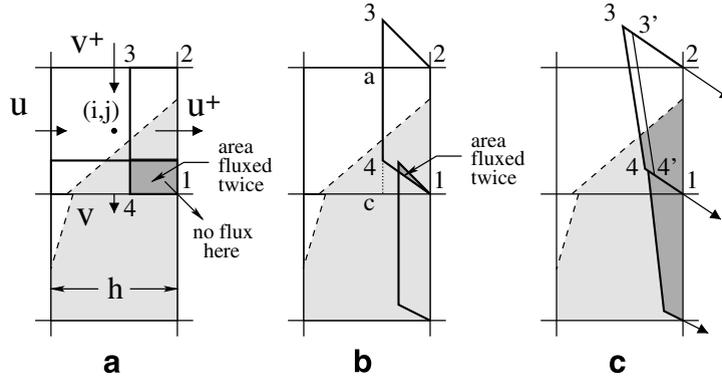


Fig. 1. Unsplit Eulerian advection: (a) concurrent split advection on a staggered MAC cell; (b) the velocity components determine directly the flux polygon 1–2–3–4 [3]; (c) the interpolated velocity at vertices 1 and 2 determines the slope of the sides 2–3, 1–4, while edge 3′–4′ is moved to 3–4 by enforcing area conservation [21].

As long as the fluid fluxes F satisfy expression (5) the incompressibility relation is also satisfied and the total fluid area, or equivalently its mass, is conserved. We now integrate (4) in space over the area of the square cell (i, j) and in time with a forward scheme and, by assuming that the discrete incompressibility condition is exactly satisfied, we get

$$h^2(C_{ij}^{n+1} - C_{ij}^n) = -(\Phi_x^+ - \Phi_x^- + \Phi_y^+ - \Phi_y^-), \quad (6)$$

where Φ is the reference-phase flux across the cell boundary. For example in Fig. 1(a), the total fluid area, including both phase 1 and phase 2, that in the time Δt will flow through the right side 1–2 is $F_x^+ = u^+ \Delta t h$ and is represented by the rectangle 1–2–3–4, where the length of side 2–3 is $u^+ \Delta t$, while Φ_x^+ is given by the grey part of this area which is occupied by the reference phase.

We now discuss briefly three fluxing schemes of increasing accuracy to show how the Eulerian unsplit advection has evolved over the last several years. The first (naive) method shown in Fig. 1(a) satisfies the discrete incompressibility condition, but it ignores completely the multidimensionality of the flow by approximating F through a cell side with a rectangular region as in a split monodimensional advection. As a result some area is fluxed twice, as the darker grey area in Fig. 1(a). In these conditions it is straightforward to show that the volume fraction data may develop overshoots, $C > 1$, or undershoots, $C < 0$. Furthermore, there is no flux in the diagonal direction through vertex 1 and the interface line rapidly develops numerical oscillations.

In the scheme proposed by Rider and Kothe (RK) [3] the rectangular flux $F_x^+ = u^+ \Delta t h$ of the naive method is deformed into a trapezoid by including two right-angled triangles that take into account the multidimensionality of the flow. In the reconstruction of the fluxing area the velocity field components, which are defined on a MAC cell, are used directly, as in Fig. 1(b) where the height of the triangle added on the top side of the cell is $|3 - a| = |v^+| \Delta t$, while that of the triangle removed from the bottom side is $|4 - c| = |v^-| \Delta t$, since both v^+ and v^- are negative. However, since the velocity field varies from point to point in general $v^+ \neq v^-$ and $F_x^+ \neq u^+ \Delta t h$. As a result the relation (5) is not satisfied and the total fluid flux entering the cell from its boundary may be smaller or bigger than the outgoing one. Another problem associated with this scheme is that some area may be fluxed twice, as depicted in Fig. 1(b). To reduce the negative effects of these issues the authors have proposed an algebraic approach by retaining the divergence term in the r.h.s. of (4), thus solving the discretized form

$$h^2(C_{ij}^{n+1} - C_{ij}^n) = -(\Phi_x^+ - \Phi_x^- + \Phi_y^+ - \Phi_y^-) + \tilde{C}_{ij}(F_x^+ - F_x^- + F_y^+ - F_y^-), \quad (7)$$

where \tilde{C}_{ij} may be an intermediate value of the volume fraction, say $\tilde{C}_{ij} = (C_{ij}^{n+1} + C_{ij}^n)/2$. Finally, the volume fraction may be locally redistributed whenever $C > 1$ or $C < 0$. Alternatively, it is possible to change the area of trapezoid 1–2–3–4, by moving the side 3–4 parallel to itself until the spanned area is equal to F_x^+ of expression (5). Then the divergence correction on the r.h.s. of the discrete equation (7) is identically zero. This second scheme has been implemented by Lopez et al. [21], however they show that for standard kinematic tests one approach is not always better than the other one. We will come back to this issue in the results section.

In the edge-matched flux polygon advection (EMFPA) [21] the velocity field is first linearly interpolated to the cell vertices. For example in the MAC cell (i, j) of Fig. 1, if we denote with the indices $(i + 1/2, j)$ the velocity u^+ in the midpoint of the right side of the cell, then the interpolated horizontal component of the velocity at vertex 2 is simply given by $u_2 = u_{i+1/2, j+1/2} = (u_{i+1/2, j} + u_{i+1/2, j+1})/2$, the vertical component v_2 is computed in an analogous way. The straight line parallel to the velocity vector $\mathbf{v}_2 = (u_2, v_2)$ approximates the characteristic curve (in other words the fluid–particle trajectory) through vertex 2. This point is linearly advected backwards along the streamline for a distance $|v_2| \Delta t$ to determine point 3′ of Fig. 1(c). Similarly, point 4′ is computed by moving backwards vertex 1 along the local linearly-approximated streamline for the distance $|v_1| \Delta t$. The slope of the segment 3′–4′ is then clearly

$$\left(\frac{dx}{dy}\right)_{3'4'} = \frac{x_2 - x_1 - (u_2 - u_1)\Delta t}{y_2 - y_1 - (v_2 - v_1)\Delta t}, \tag{8}$$

but the area of the resulting trapezoid 1–2–3'–4' in general is not equal to F_x^+ , therefore the segment 3'–4' is moved parallel to itself to 3–4 to get the correct area of the flux polygon. By construction no area is fluxed twice and in each cell the incoming fluid area is always equal to the outgoing one as long as the discrete velocity field is divergence-free. There are a few minor problems with this scheme as well. Notice that in Fig. 1(c) the reconstruction of two consecutive flux polygons is not continuous at point 4. Furthermore, as the one-dimensional CFL number, $|u|\Delta t/h$ or $|v|\Delta t/h$, gets close to its limiting value which is equal to 1, point 4 may move outside the cell (i, j) or point 3 out of $(i, j + 1)$: both occurrences lead to inconsistencies. To overcome these problems we have developed a new advection scheme that will be discussed in the next section.

3. The geometrical predictor–corrector advection (GPCA) scheme

In this new scheme we require the flux polygons to satisfy the incompressibility relation (5) and their boundary line to be continuous between two consecutive polygons. To this aim we first determine with a geometrical predictor–corrector technique the points that in the time step Δt will be advected along the streamlines to the cell vertices. Then we connect two consecutive points by enforcing area conservation and by keeping the shape of the resulting polygon rather simple, in particular the union of a triangle and of a trapezoid, as the shapes 2–3–a and 2–a–4–1 of Fig. 1(b).

3.1. Discrete velocity field and pre-image of the grid lines

To ensure that the discrete incompressibility condition is satisfied to machine accuracy, we compute the velocity field components u and v on the staggered MAC grid of Fig. 1(a) from an analytical stream function $\psi(x, y)$, with $u = -\partial\psi/\partial y$ and $v = \partial\psi/\partial x$ and use centered finite differences to approximate the derivatives. For example, the horizontal component of the velocity placed in the midpoint of the right side of the cell (i, j) is given by $u_{i+1/2, j} = -(\psi_{i+1/2, j+1/2} - \psi_{i+1/2, j-1/2})/h$. We then interpolate linearly the velocity field from the sides to the vertices of each cell [28], in particular in the top-right corner of the same cell we have $u_{i+1/2, j+1/2} = (u_{i+1/2, j+1} + u_{i+1/2, j})/2 = -(\psi_{i+1/2, j+3/2} - \psi_{i+1/2, j-1/2})/(2h)$, therefore the velocity components on the cell vertices are obtained from ψ with centered finite differences with a grid spacing of $2h$.

In order to describe the main features of the geometrical advection scheme we consider the velocity field from the single-vortex test of Section 4.1. In particular the bottom-left cell of Figs. 2 and 4 correspond to the cell of Fig. 6 that includes the point with coordinates (0.3, 0.6). In Fig. 2 we use nondimensional variables, so that the side of the square cells is equal to 1 and the local maximum CFL number is about 0.85. The dashed lines are the exact pre-images of the grid lines and the points where these lines intersect each other will be advected in time Δt to the grid nodes by flowing along the streamlines, which are shown in the figure as dotted lines. In practice, all these lines have been computed numerically from a given analytical velocity field with a fourth-order Runge–Kutta integration scheme, where the time step Δt has been subdivided several times until we reach convergence to a degree that cannot be distinguished in the figure (for example six decimal digits).

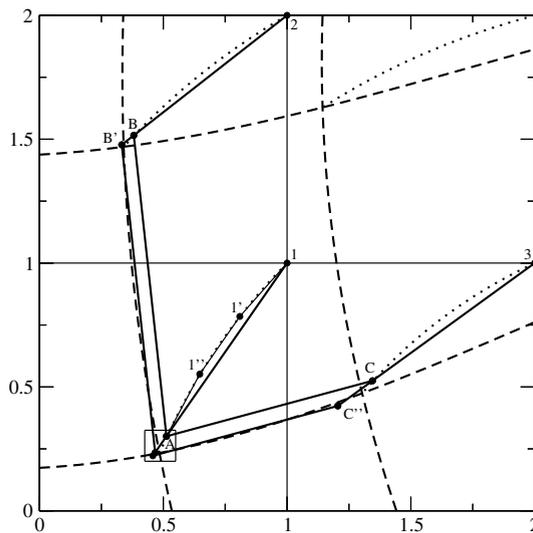


Fig. 2. Pre-image of the grid lines (dashed lines) and streamline through the cell vertices (dotted lines) zoomed from the boxed area in Fig. 6 (top-left); backward advection of vertex 1 to A through intermediate points 1' and 1''; flux polygon 1–2–B–A is corrected to 1–2–B'–A' and 1–3–C–A to 1–3–C'–A', by enforcing area conservation. The square area near point A is zoomed in Fig. 3(a).

3.2. Position of the points mapped onto the cell vertices

On a staggered MAC grid we first interpolate linearly the velocity components from the cell sides to the cell vertices. We then trace backwards along the streamline for the time Δt to compute the pre-image of each cell vertex with a geometrical predictor–corrector method.

3.2.1. Predictor step

Vertex 1 of Fig. 2 is traced back to point A with a simple predictor–corrector integration scheme but with three subdivisions of Δt , thus going through points 1' and 1'' before reaching A. The local value of the velocity is computed with a bilinear interpolation from the velocity at the cell vertices. At the end of the predictor step we have then computed the position of the points, such as A, B, C of Fig. 2, that will be mapped to the cell vertices.

3.2.2. Corrector step

We adjust the position of the computed points with the following scheme. The area of the trapezoid 1–2–B–A in general is not equal to the total fluid flux F_x^+ leaving the top-left cell from the side 1–2. We then move the side B–A to B'–A' in a parallel way until this condition is satisfied. Similarly, we move the side C–A of the trapezoid 1–3–C–A to C''–A'' so that the area of 1–3–C''–A'' is equal to the fluid flux F_y^+ leaving the bottom-right cell through side 1–3. The position of the two points A' and A'', which are aligned with A and vertex 1, can be seen in the zoomed area of Fig. 3(a). We conclude the corrector step by selecting one of the two points A' and A''. Our aim is to distribute the area correction homogeneously along the boundary of the flux polygon, therefore we select A* between A' and A'' so that

$$|A-A^*| = \text{MIN}(|A-A'|, |A-A''|). \tag{9}$$

In this way, if an area correction is still necessary, it will be distributed along the boundary line connecting two consecutive pre-image points, such as A* and B*.

Finally, if we consider an integration scheme backwards along the streamline with a fractional time step, we can also save the position of the last point before A, such as 1'' of Fig. 2 and add the correction $|A-A^*|$ along the segment 1''–A rather than 1–A. The final result does not change in a dramatic way, as shown in Fig. 3(a), but it is a bit smoother with little computational effort. At the end of this corrector step we have defined a set of pre-image points, such as A*, B*, C* of Fig. 4, each of them corresponding in time Δt to a different cell vertex.

3.3. Flux polygons

In the final step of the proposed scheme we determine the flux polygons, again with a predictor–corrector technique.

3.3.1. Predictor step

We compute the area of the trapezoid based on two consecutive pre-image points, such as A* and B* of Fig. 4 and the corresponding vertices 1 and 2.

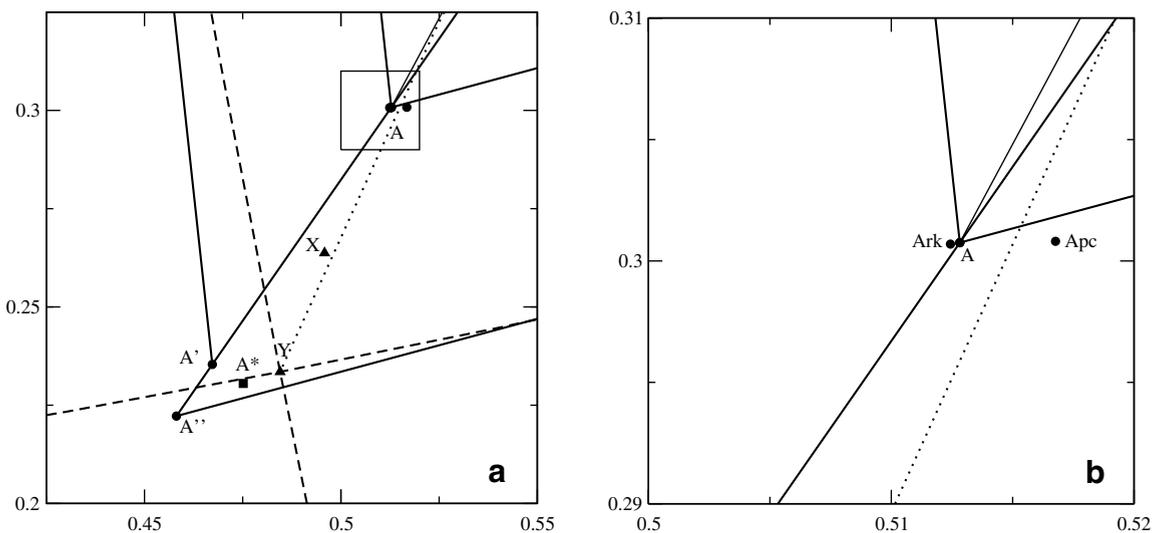


Fig. 3. (a) Zoom of the square area near point A of Fig. 2: the correction is added along the segment 1''–A to define the pre-image A* of vertex 1; the points near A and points X and Y are used to geometrically quantify the integration, discretization and interpolation errors. (b) Zoom of the square area in (a) to show the endpoints A, A_{pc} and A_{rk} of different integration schemes along the same streamline.

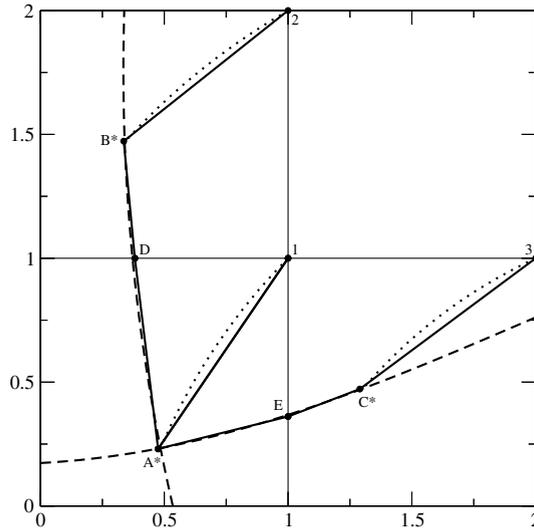


Fig. 4. Pre-image of the grid lines (dashed lines) and streamline through the cell vertices (dotted lines). The area of the polygon 1–2–B*–D–A* is equal to F_x^+ , that of 1–3–C*–E–A* to F_y^+ and the continuous piecewise-linear line B*–D–A*–E–C* approximates the pre-image of the cell sides 1–2 and 1–3.

3.3.2. Corrector step

This area usually is not equal to the total fluid flux F_x^+ through the cell side 1–2, therefore we move the intersection of the segment A*–B* with the horizontal grid line of Fig. 4 to point D, so that the area of the polygon 1–2–B*–D–A* is equal to F_x^+ .

As anticipated, the area of the flux polygon is the union of a triangle, 1–D–A*, and a trapezoid, 1–2–B*–D. Similarly, the area of the flux polygon 1–3–C*–E–A* is equal to the fluid flux F_y^+ through the cell side 1–3. The set of four consecutive segments B*–D, D–A*, A*–E and E–C* is our final approximation to the two curved lines that in the time step Δt will be mapped to the two cell sides 1–2 and 1–3.

The shape of the flux polygons can change considerably if we consider a velocity field with non-uniform vorticity. In Fig. 5 we show a few different cases: on the left the standard flux polygon similar to that of Fig. 4, in the central case the fluxing area is defined across two horizontal cells because of the clockwise rotation of the flow around a point on the right side of the cell, finally in the third case the horizontal component of the velocity changes its sign and the flux polygon cuts three different cells. By considering the intersections of the polygon sides with the grid lines the fluxing area can be easily subdivided in triangles, as shown in Fig. 5. Therefore to compute the reference phase fluxes we use a single routine that determines the area of a triangle that is cut by interface which is approximated in each cell by an oriented segment [10]. Because of this feature, the methodology can be directly extended to unstructured grids.

3.4. A geometrical representation of the numerical errors

If we consider a translational velocity field on a Cartesian grid with square cells, then the fluxing areas are given by polygons that are computed exactly by this and other schemes as well [21]. However, in a vortical flow a few integration errors are always present and in this section we discuss how relevant they are by considering again the single-vortex velocity field of Figs. 2–4. We point out that the local CFL number is about 0.85 and that for this test the maximum variation of the one-

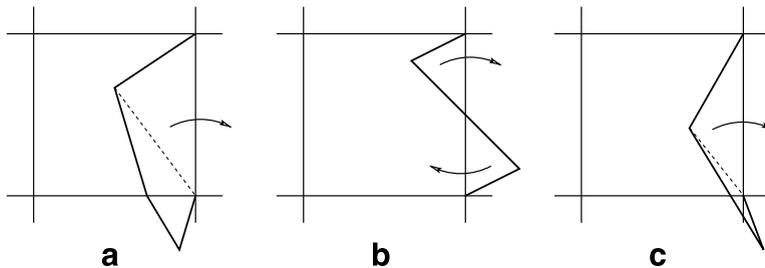


Fig. 5. Three different shapes of the flux polygon through the right side of a cell: (a) in the standard case the flux polygon is inside two consecutive vertical cells; (b) close to an instantaneous center of rotation the flux polygon may be defined across two horizontal cells; (c) when the horizontal component of the velocity changes its sign, the polygon may intersect three different cells. To compute the reference phase fluxes the polygons are subdivided in triangles (dashed lines).

dimensional CFL number is from zero to one in just three cells. As a matter of fact, this is a rather severe test and the errors can be considered as an upper limit for actual dynamical simulations. Furthermore, to simplify the analysis we assume that the total numerical error is made up by several independent terms.

A first contribution comes from the integration scheme to trace back the cell vertices along the streamline for the time Δt . Near point A in Fig. 3(a), which is aligned with A' and A'' , there are actually two more points, as shown in Fig. 3(b). The rightmost one A_{pc} has been obtained with a single predictor–corrector step, the central one is point A of our scheme obtained by subdividing Δt in three equal parts and by using the same predictor–corrector technique in each fractional step. The leftmost one A_{rk} has been obtained with a fourth-order Runge–Kutta method and four equal subdivisions of the time step. The last two positions almost overlap, $|A - A_{rk}|/|1 - A| < 5 \times 10^{-4}$, and the contribution to the total error of the integration along the streamline is very small. We have chosen the predictor–corrector integration with three fractional steps as a satisfactory compromise between accuracy and efficiency. This method also gives us a good estimate near the endpoint A of the tangent along which we add the correction $|A - A^*|$, as discussed at the end of Section 3.2 and shown in Fig. 3(a).

A second contribution is a discretization error. We have computed the velocity field components on the staggered MAC grid with centered finite differences of the stream function in order to satisfy the discrete divergence-free condition and then we have linearly interpolated these values to the cell vertices. If we consider the analytical value of the velocity at the cell vertices, but with the same integration and interpolation schemes, we now end up at point X of Fig. 3(a). The length of the segment A–X is then a measure of the discretization error for this case.

A further contribution is an interpolation error. We compute the velocity values inside the cell from the numerical values at the cell vertices with a bilinear interpolation, that guarantees that the velocity field is continuous everywhere but it does not satisfy locally the divergence-free condition. If we integrate along the streamline as for the pre-image of the grid lines with a fourth-order Runge–Kutta integration scheme, several subdivisions of the time step Δt and with the analytical value of the local velocity we reach point Y of Fig. 3(a) and the length of the segment X–Y is then a measure of the interpolation error. The discretization and interpolation errors are comparable in size and larger than the integration error.

However, at the end of our geometrical predictor–corrector scheme, the three pre-images A^* , B^* and C^* of vertices 1, 2 and 3 of Fig. 4 are very close to the exact values. The fractional increase of area from the initial trapezoid 1–2–B–A to the final one 1–2–B^{*}–A^{*} can be as high as 10%. Therefore, a strict enforcement of the divergence-free condition is indeed very important to compute accurately the pre-images of the grid nodes.

Up to now we have considered a time-independent velocity field. Time dependent errors should not be very important because of the strong limitations on the time step due to numerical stability requirements. They can also be reduced by considering the velocity field at the intermediate time level $n + 1/2$ or by interpolating it linearly between the two consecutive time levels n and $n + 1$.

As a final issue we point out that in the approximation of the flux regions we connect A^* to vertex 1 with a straight line, however the streamline through these two points is curved (this is the dotted line of Fig. 4 connecting A^* and 1). The area comprised between these two lines is an error that enters as an extra positive contribution to the fluxing area through side 1–2. The opposite is true for the area between the straight line and the streamline connecting B^* and 2. Because of the spatial variation of the velocity field these two opposite contributions do not cancel out, in particular in Fig. 4 the first positive contribution is larger than the second negative one. As a result, the segment $B^* - D$ needs to be on the right of the exact line, so that the area between the two consecutive segments $B^* - D$ and $D - A^*$ and the dashed line $B^* - A^*$ cancels out this net positive contribution and the total flux through side 1–2 is equal to F_x^+ . This last error is of the order of 1% at most and it is the main source of the difference between our piecewise-linear reconstruction of the fluxing areas and the pre-images of the grid lines. It is possible to reduce it by approximating the area between the streamline and the segment connecting A^* to 1, or B^* to 2, with a triangle, but we have not implemented such a scheme. Finally, notice that in Fig. 4 the area of these two opposite contributions for the vertical flux F_y^+ is almost the same, and the segment $A^* - E$ is now a rather good approximation as it intersects twice the exact dashed line.

4. Flux polygon reconstruction and volume fraction advection tests

We now consider two standard analytical velocity fields with non-uniform vorticity, namely the single-vortex test and the deformation field. The aim is to check how the geometrical method approximates the fluid areas that are flowing through the grid cell boundary and to compare them with the approximations from the two other Eulerian schemes previously mentioned. We then present results for the advection of the volume fraction function only for two standard benchmark tests, namely the Rudman–Zalesak slotted disk rotation test and the Rider–Kothe reversed single-vortex test, in order to point out the main differences with other state-of-the-art VOF methods.

4.1. Flux polygon reconstruction tests

In this first we focus our analysis on the comparison of the flux polygons computed by different advection schemes with the exact fluid fluxes. In the single-vortex the velocity field is determined by the stream function

$$\psi(x, y) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y), \quad (10)$$

and with $u = -\partial\psi/\partial y$ and $v = \partial\psi/\partial x$ the flow rotates in the clockwise direction. The computational domain is a unit square $[0, 1] \times [0, 1]$ partitioned with square cells of side $h = 1/n$, where n is the number of cells along each coordinate direction. In Fig. 6 we show the grid lines of the top-left quadrant with $n = 12$, their pre-images obtained by a backward advection along the streamlines of (10) and the portion of the streamlines covered by the pre-images of the points in their motion towards the cell vertices. In the same figure we depict the flux polygons of the three advection schemes previously described, namely the RK, EMFPA and GPCA algorithms. As seen in Fig. 6 when the vortex is resolved with 12 cells across its diameter the maximum variation of the one-dimensional CFL number is from zero to one in just three cells. The RK algorithm is clearly discontinuous and many flux polygons overlap. The EMFPA scheme presents difficulties in the cells marked by the symbol ‘*’, where the corrected flux polygon 1–2–3–4 of Fig. 1(c) will have either point 3 or 4, or both, out of their cell, corresponding to a CFL greater than 1. There are also a few instances where the two sides 1–4 and 2–3 intersect each other, but the area of the resulting triangle is smaller than the flux F , hence the problem has no geometrical solution. In both cases we draw only the flux polygon which is equivalent to 1–2–3’–4’ of Fig. 1(c). We will discuss in the next section the solution suggested in [21] to go around this problem. The flux polygons computed with the new geometrical predictor–corrector algorithm always satisfy the CFL condition and are very close to the exact solution. Finally, we consider the same test but with the maximum CFL

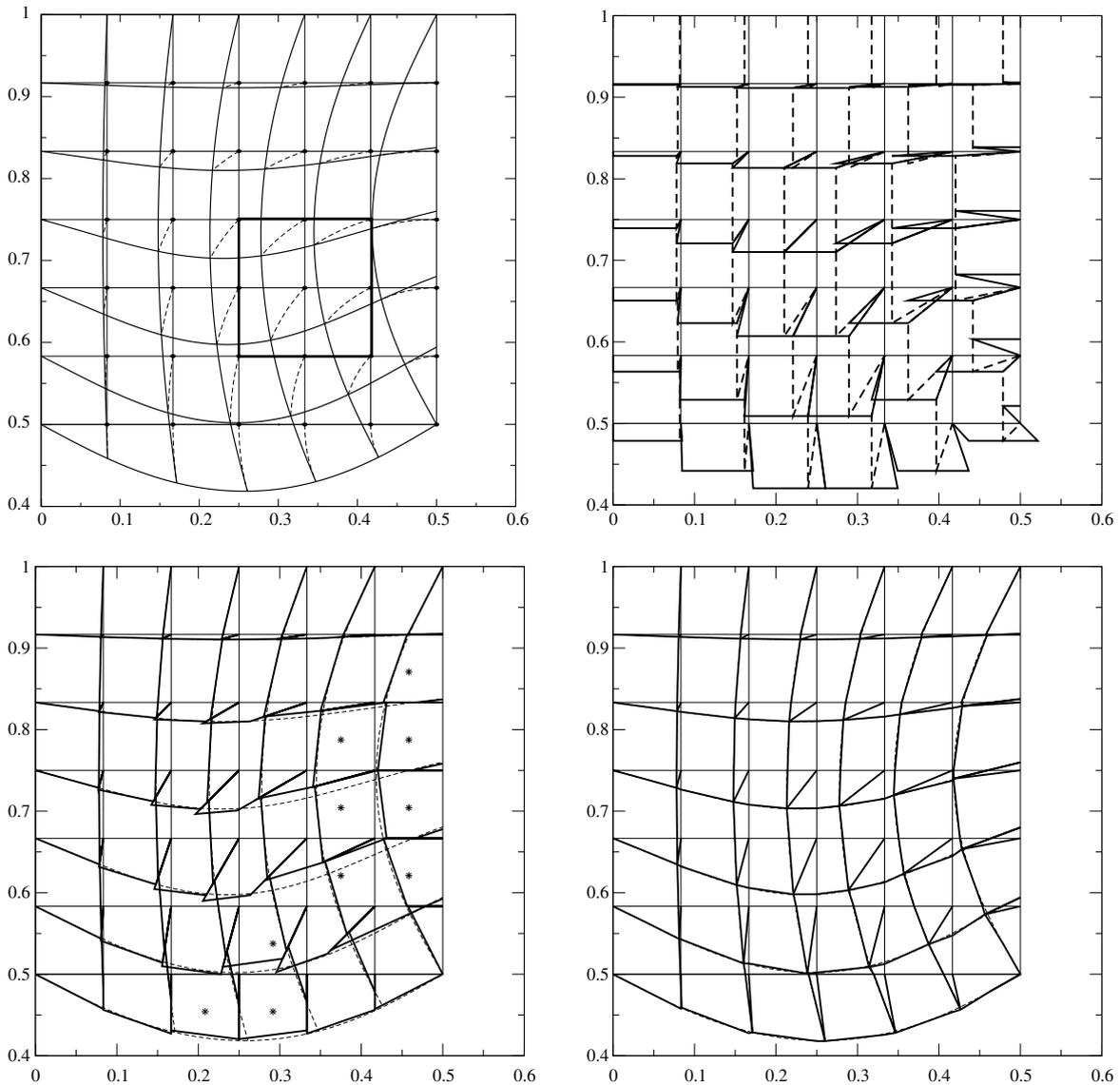


Fig. 6. Single-vortex: grid lines of the top-left quadrant with $n = 12$, their pre-image when $\max(\text{CFL}) = 1$ (thick solid lines) and streamline through grid nodes (dashed lines) (top-left); RK algorithm with the flux polygons through the horizontal (solid lines) and vertical (dashed lines) cell sides (top-right); EMFPA (bottom-left) and GPCA (bottom-right) flux polygons (solid lines) and exact pre-image of the grid lines (dashed lines). The boxed area in the top-left figure is Fig. 2.

number equal to 0.5 to show in Fig. 7 the major differences between the EMFPA and GPCA schemes at a CFL number closer to those found in actual dynamical simulations.

For the deformation field test we consider the stream function

$$\psi(x, y) = \frac{1}{4\pi} \sin(4\pi x) \sin(4\pi y), \quad (11)$$

that describes in the same computational domain as the previous test a matrix of 4×4 square counter-rotating vortices. We now consider $n = 1/h = 32$ so that each vortex is inside a little square with eight cells along each side. In Fig. 8 we show the rectangular computational grid that contains two consecutive vortices in the horizontal direction. In the figure we also show the pre-images of the grid lines computed numerically both with the analytical velocity field and with the GPCA scheme from the discrete velocity values on the MAC grid. Again, the flux polygons approximate rather well the exact fluxing areas. Notice that even if each vortex is defined inside a square with only eight cells per side, the velocity along its boundary is not zero and the maximum variation of the CFL number is now equal to one across 4 cells. However, the deformation test case has counter-rotating vortices which allow for more complex cell acceptance and donor fluxing characteristics than the single-vortex test.

We finally set $n = 31$, so that the horizontal line $y = 0.5$ does not coincide with a grid line. In Fig. 9 we show a portion of the computational grid around $y = 0.5$ and the computed pre-image lines. The flow is of course incompressible, but from left to right it is first converging towards the horizontal direction and then it diverges. In this case the central leftmost cell is accepting fluid through its left side from three consecutive vertical cells, the opposite is true for the central rightmost cell

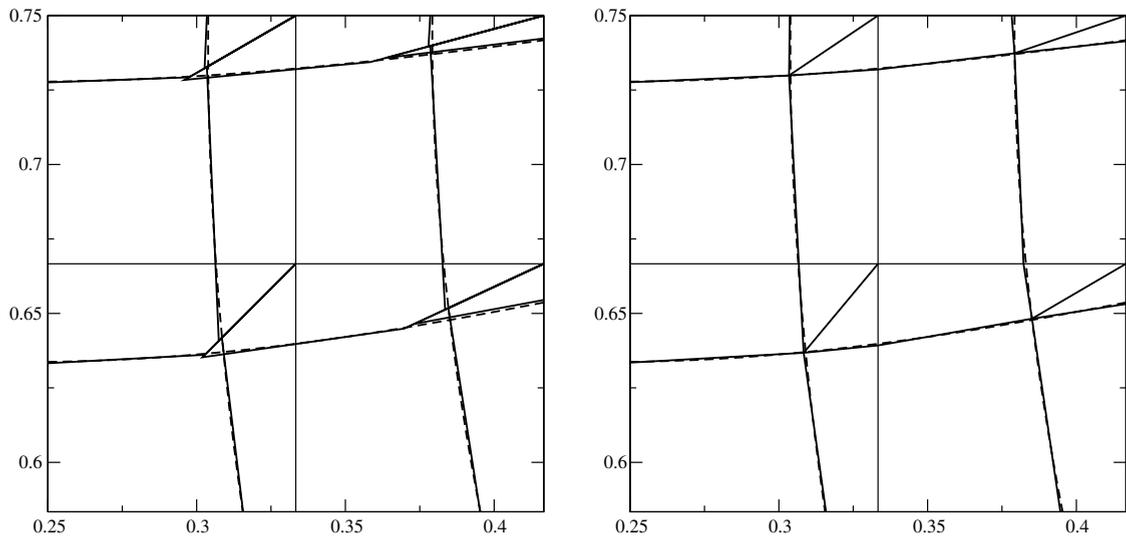


Fig. 7. Single-vortex test: a box with four grid cells of the top-left quadrant with $n = 12$, pre-images inside the box when $\max(\text{CFL}) = 0.5$ (dashed lines) and flux polygons inside the box (solid lines) computed by EMFPA (left) and GPCA (right).

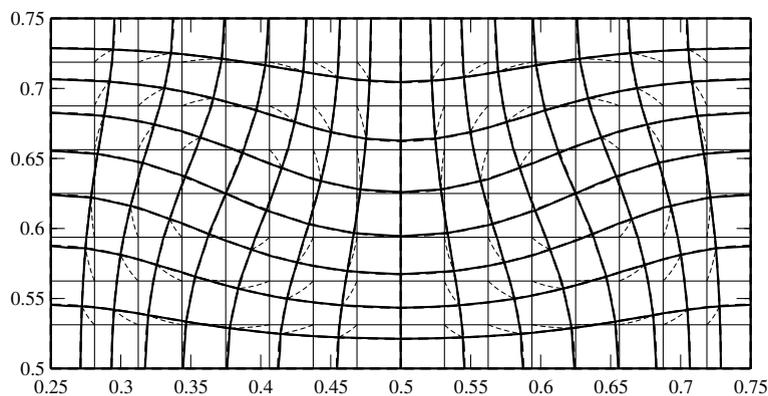


Fig. 8. Two horizontal vortices of the deformation field test (that on the left rotates in the counterclockwise direction) with $n = 32$ and $\max(\text{CFL}) = 1$: exact pre-images (dashed lines), their approximation by GPCA (solid lines) and a part of the streamline through the grid nodes (thin dashed lines).

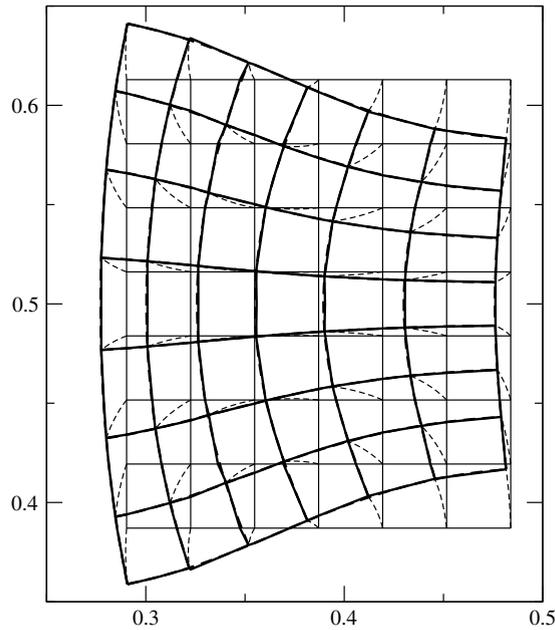


Fig. 9. Deformation field test with $n = 31$ and $\max(\text{CFL}) = 1$: exact pre-images (dashed lines), their approximation by GPCA (solid lines) and a part of the streamline through the grid nodes (thin dashed lines), showing a convergent-divergent behaviour towards the horizontal line $y = 0.5$.

which is donating fluid through its right side to three consecutive vertical cells. These two cases define the most complex flux polygon structures in two dimensions.

4.2. Rudman–Zalesak slotted disk rotation test

In this test a slotted disk is advected by a uniform vorticity field. The test was first introduced by Zalesak [29] and the version described in [30] is often used to check the performance of an interface capturing method. A square of side equal to 4 is divided in 200×200 square grid cells. The diameter of the circle is 50 mesh cells and the slot width is 6 cells. The circle center is initially at $(2.0, 2.75)$ and the velocity components of the solid-body rotation are derived from the following stream function:

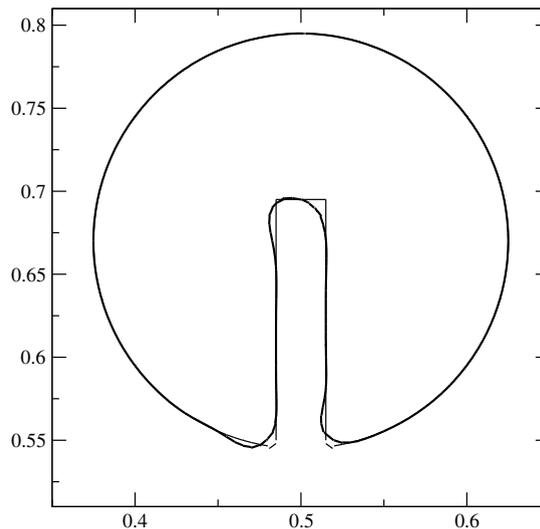


Fig. 10. Rudman–Zalesak slotted disk rotation test: initial (thin solid line) and final (thick solid line) interface profiles.

Table 1

Geometrical error as defined in (13) in the slotted disk rotation test for different reconstruction and advection algorithms

Reconstruction/advection algorithms	E
Youngs [30]	1.09×10^{-2}
Puckett/Stream [19]	1.00×10^{-2}
Puckett/DDR [20]	1.50×10^{-2}
Puckett/EMFPA [21]	9.73×10^{-3}
Puckett/GPCA	9.79×10^{-3}

$$\psi(x, y) = -\frac{\omega}{2}[(x - x_0)^2 + (y - y_0)^2], \quad (12)$$

with $(x_0, y_0) = (2., 2.)$. The angular velocity ω is such that a full revolution is completed in 2524 steps, corresponding to a maximum CFL number of about 0.25. For this test the geometrical error E is usually calculated as

$$E = \frac{\sum_{ij} |C_{ij}^T - C_{ij}^0|}{\sum_{ij} C_{ij}^0}, \quad (13)$$

where C_{ij}^0, C_{ij}^T are the initial and final volume fraction values at cell (i, j) . The mass/area error is of the order of machine accuracy, since the discrete divergence-free condition is always satisfied. For comparison with other advection schemes we have implemented Puckett's reconstruction method [13] which performs a bit better than the ELVIRA reconstruction when the interface has a local radius of curvature smaller than the grid spacing [14]. The geometrical errors of (13) are given in Table 1 for different combinations of reconstruction/advection schemes but for the same grid resolution. We basically recover the same results of EMFPA scheme, since the flux polygons for this test, a solid-body rotation with relatively small CFL numbers, are very similar and the geometrical error is mainly concentrated near the sharp edges of the disk (see Fig. 10).

4.3. Rider–Kothe reversed single-vortex test

In this test a circular body of radius 0.15 and center at point $(0.5, 0.75)$ is advected by the velocity field derived by the stream function $\psi(x, y)$ of (10). The computational domain is a unit square box and the velocity field is zero on its boundary. The stream function is modulated in time by the temporal factor $\cos(\pi t/T)$, where T is the period [31]. The fluid body is stretched by the flow, reaching a maximum deformation at time $t = T/2$ and then it goes back to its initial position at $t = T$. The geometrical error E at grid spacing $h = 1/n$ and the order of convergence ϱ , between the two resolutions n and $2n$, are calculated as

$$E(h) = h^2 \sum_{i=1}^n \sum_{j=1}^n |C_{ij}^T - C_{ij}^0|, \quad \varrho = \frac{\ln(E(h)/E(h/2))}{\ln(2)}, \quad (14)$$

with C_{ij}^0, C_{ij}^T the initial and final volume fraction values at cell (i, j) , while the mass/area error is of the order of machine accuracy, since the discrete divergence-free condition is always satisfied. In Fig. 11 we show the reconstructed interface at times $t = T/2, T$ obtained with two different grid resolutions, $n = 32, 128$. The geometrical errors of (14) are reported in Table 2 for the three advection tests described in the previous sections. With the smallest period $T = 0.5$ the body is deformed by the flow, but the exact local radius of curvature remains always larger than the cell size and the interface is always numerically well resolved. To quantify this statement we have calculated the jump of the interface reconstruction at the boundary of two consecutive cells. This calculation has been performed for all mixed cells at every time step and we have found that the maximum of the computed discontinuity remains always $\mathcal{O}(h^2)$. In these conditions the GPCA scheme performs slightly better than the other two advection algorithms.

With period $T = 2$ and at halftime as seen in Fig. 11, the deformed fluid body is poorly reconstructed near the head and the tail of the spiral, where the local curvature is comparable or even smaller than the grid spacing $1/h$ and the discontinuity at the cell boundary is now $\mathcal{O}(h)$. In these conditions it is not difficult to devise a situation where an approximation in the reference phase flux calculation can actually be more accurate than the GPCA scheme. More particularly, we recall that in Fig. 1(c), when the CFL number is close to 1, point 3 may move out of the cell $(i, j + 1)$. In these conditions the EMFPA algorithm computes the reference phase flux inside the polygon $P = 1-2-3'-4'$ and then corrects it by a factor R , which is equal to the ratio of the exact total fluid area F_x^+ to the area of P , $R = F_x^+/P > 1$. Now suppose that in Fig. 12(a) the segment $3'-4'$ of the flux polygon coincides with the interface line. If the corrected point 3 goes out of its cell, the reference phase flux through the right side of the cell is estimated by multiplying the area of the polygon P by the factor $R > 1$, then its value is overestimated. This is not the case for the GPCA scheme, which is not subject to this problem. Therefore when the reconstruction is close enough to the interface line, GPCA should perform better than the other two schemes, as in the results of Table 2 with $T = 0.5$.

Consider now Fig. 12(b) which is similar to Fig. 12(a), except that the interface line $3'-4'$ is poorly reconstructed by the segment $S-T$, with $A_1 = A_2$ because of area conservation. The two jumps at the cell boundary between the interface line and

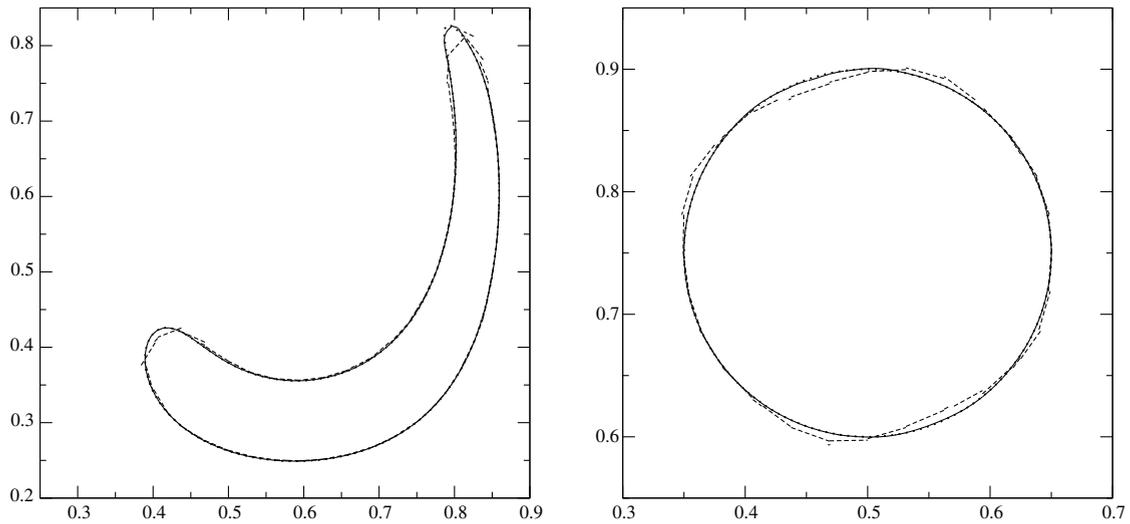


Fig. 11. Single-vortex test with period $T = 2$ and $\max(\text{CFL}) = 1$: the exact and reconstructed interfaces at $t = T/2$ (left) and $t = T$ (right). In both figures the dotted line is the exact interface described by line markers, while the reconstructed interfaces refer to a 32^2 (dashed segments) and a 128^2 (solid segments) grid.

Table 2

Geometrical error E and order of convergence ϱ in the reversed single-vortex test for different periods T , grid resolution n and advection algorithms

n	$T = 0.5$			$T = 2.$		
	GPCA	EMFPA	RK	GPCA	EMFPA	RK
32 E	4.12×10^{-4}	4.45×10^{-4}	7.29×10^{-4}	2.18×10^{-3}	2.14×10^{-3}	2.36×10^{-3}
(ϱ)	(2.41)	(2.48)	(2.36)	(2.05)	(1.99)	(2.01)
64 E	7.32×10^{-5}	7.99×10^{-5}	1.42×10^{-4}	5.32×10^{-4}	5.39×10^{-4}	5.85×10^{-4}
(ϱ)	(1.93)	(1.97)	(1.86)	(2.03)	(2.06)	(2.16)
128 E	1.93×10^{-5}	2.04×10^{-5}	3.90×10^{-5}	1.29×10^{-4}	1.29×10^{-4}	1.31×10^{-4}

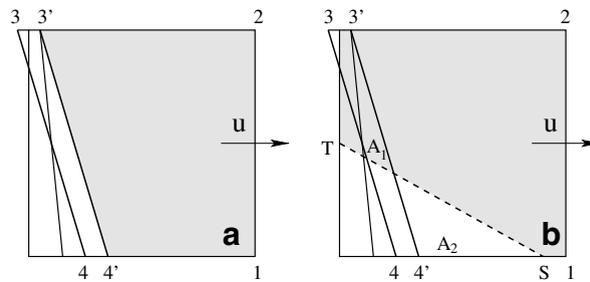


Fig. 12. The performance of a fluxing scheme is strongly affected by the reconstruction: (a) the interface line $3'-4'$ coincides with a side of the fluxing polygon $1-2-3'-4'$ of area P , the corrected point 3 of the EMFPA scheme is out of the cell and the flux PR , with $R > 1$, overestimates the actual reference phase flux; (b) the interface line is poorly reconstructed by the segment $S-T$, in this case the approximation in the EMFPA scheme fluxes more efficiently than the continuous GPCA scheme (thin solid line).

its approximation are clearly $\mathcal{O}(h)$ and they are given by the two distances $3'-T$ and $4'-S$. With the EMFPA scheme the initial reference phase defect is equal to A_2 , however as we multiply it by the factor R , we obtain a final flux which is larger than the one computed by the GPCA scheme and closer to the exact value. The reason for this result is that as we move from point S' towards the bottom-left cell vertex, the fraction of the cell height h which is occupied by the reference phase decreases progressively. Notice that by increasing the spatial resolution n we increase the number of cells where the local CFL number is close to 1, both in space and in time because of the cosinusoidal time modulation of the stream function. Therefore, the number of cells where this situation may occur increases with the resolution. In practice, it is possible to devise situations where because of the poor local interface reconstruction, the EMFPA and RK algorithms may locally flux better than a continuous

Table 3Geometrical error as defined in (14) for different advection algorithms in the reversed single-vortex test for $T = 8$ and grid resolution 128×128

Reconstruction/advection algorithm	E
Puckett/Rider and Kothe [3]	1.44×10^{-3}
Puckett/Stream [19]	1.18×10^{-3}
Puckett/EMFPA [21]	1.17×10^{-3}
Puckett/GPCA	1.17×10^{-3}

fluxing scheme such as GPCA, even if the boundary of two consecutive flux polygons is not continuous or there are overlapping regions.

In view of this discussion, the results of Table 2 at $T = 2$ show that we are more or less at the limit of the accuracy that can be achieved with Puckett's method when the interface is highly deformed, in other terms a much more refined grid is required near the tail of the spiral to appreciate the differences among the different advection algorithms. Smoothing of the interface normal can improve the performance of a combined reconstruction and advection algorithm where the interface is already rather regular [3,21]. However, in regions of very high curvature or in the presence of filaments, whose thickness is comparable to the grid spacing, adaptive mesh refinement and/or multiple segment reconstruction are required to improve the performance of VOF methods.

The period $T = 2$ is still too early to sufficiently test a VOF method. For this reason we perform the single-vortex test for the longer period $T = 8$. The numerical breakup of the interface is completely dominated by the reconstruction algorithm which fails to approximate adequately high curvature regions and filament structures. In Table 3 we show the results on a grid with 128×128 cells. At lower resolutions, the interface line breaks in several pieces because the tail of the spiral becomes thinner than the edge h and the different fluxing schemes may locally perform better or worse with respect to one another because of random favorable alignments of the reconstructed interface and the flux polygons. Front-tracking methods with interfacial points can approximate the interface in each mixed cell by an ordered sequence of segments and can represent thin filaments and high curvature regions very precisely. In this test the interface usually does not break up and the geometric error is one to two orders of magnitude smaller than the corresponding value of VOF methods [32,24]. Finally, we have compared the CPU time of the single-vortex test for $T = 8$ and 128×128 cells consumed by the GPCA method and the unsplit Eulerian–Lagrangian advection Π of [15], the latter being computationally very efficient because it is characterized by simple rectangular boundary fluxes. We have found that the GPCA method requires about twice the CPU time of Π , which we consider a satisfactory result given the complexity of the flux polygons.

5. Conclusions

We have developed a multidimensional Eulerian advection method in two-dimensional Cartesian geometry and we have applied it to follow the evolution of the volume fraction function. The scheme is based on an accurate computation of the pre-images of the grid lines that are approximated by continuous, piecewise-linear lines. The enforcement of the discrete version of the incompressibility constraint is a key issue to reduce considerably the total numerical error. The proposed method compares favorably with other previous multidimensional advection methods as long as the interface line is well reconstructed. We have also found that when the interface is very fragmented the total numerical error is completely dominated by the reconstruction error and in these conditions it is very difficult to assess which advection scheme is the most reliable one.

Acknowledgment

We would like to thank the referees for their helpful comments and suggestions.

References

- [1] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* 100 (1992) 25–37.
- [2] 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.
- [3] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, *J. Comput. Phys.* 141 (1998) 112–152.
- [4] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* 31 (1999) 567–603.
- [5] S. Osher, R.P. Fedkiw, Level set methods: an overview and some recent results, *J. Comput. Phys.* 169 (2001) 463–502.
- [6] J.A. Sethian, P. Smereka, Level set methods for fluid interfaces, *Annu. Rev. Fluid Mech.* 35 (2003) 341–372.
- [7] D. Jacqmin, Calculation of two-phase NavierStokes flows using phase-field modeling, *J. Comput. Phys.* 155 (1999) 96–127.
- [8] V.E. Badalassi, H.D. Ceniceros, S. Banerjee, Computation of multiphase systems with phase field models, *J. Comput. Phys.* 190 (2003) 371–397.
- [9] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, *J. Comput. Phys.* 162 (2000) 301–337.
- [10] X. Yang, A.J. James, J. Lowengrub, X. Zheng, V. Cristini, An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids, *J. Comput. Phys.* 217 (2006) 364–394.
- [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] E. Aulisa, S. Manservigi, 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.

- [13] E.G. Puckett, A volume of fluid interface tracking algorithm with applications to computing shock wave rarefaction, *Proc. 4th Int. Symp. Comput. Fluid Dyn.* (1991) 933–938.
- [14] J.E. Pilliod Jr., E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, *J. Comput. Phys.* 199 (2004) 465–502.
- [15] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection, *Int. J. Numer. Meth. Fluids* 41 (2003) 251–274.
- [16] E. Aulisa, S. Manservigi, R. Scardovelli, S. Zaleski, Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry, *J. Comput. Phys.* 225 (2007) 2301–2319.
- [17] 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.
- [18] M. Rudman, Volume-tracking methods for interfacial flow calculations, *Int. J. Numer. Meth. Fluids* 24 (1997) 671–691.
- [19] D.J.E. Harvie, D.F. Fletcher, A new volume of fluid advection algorithm: the stream scheme, *J. Comput. Phys.* 162 (2000) 1–32.
- [20] 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.
- [21] 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.
- [22] K. Shahbazi, M. Paraschivoiu, J. Mostaghimi, Second order accurate volume tracking based on remapping for triangular meshes, *J. Comput. Phys.* 188 (2003) 100–122.
- [23] N. Ashgriz, T. Barbat, G. Wang, A computational Lagrangian–Eulerian advection remap for free surface flows, *Int. J. Numer. Meth. Fluids* 44 (2004) 1–32.
- [24] Q. Zhang, P.L.F. Liu, A new interface tracking method: the polygonal area mapping method, *J. Comput. Phys.* 227 (2008) 4063–4088.
- [25] J.K. Dukowicz, M.C. Cline, F.L. Addessio, A general topology Godunov method, *J. Comput. Phys.* 82 (1989) 29–63.
- [26] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, *J. Comput. Phys.* 87 (1990) 171–200.
- [27] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids* 8 (1965) 2182–2189.
- [28] M. Shashkov, B. Swartz, B. Wendroff, Local reconstruction of a vector field from its normal components on the faces of grid cells, *J. Comput. Phys.* 139 (1998) 406–409.
- [29] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys.* 31 (1979) 335.
- [30] M. Rudman, Volume-tracking methods for interfacial flows calculations, *Int. J. Numer. Meth. Fluids* 24 (1997) 671–691.
- [31] R.J. Leveque, High-resolution conservative algorithms for advection in incompressible flow, *SIAM J. Numer. Anal.* 33 (1996) 627–665.
- [32] E. Aulisa, S. Manservigi, 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.