

Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry

E. Aulisa¹, S. Manservigi^{1,2}, R. Scardovelli^{2,*} and S. Zaleski³

November 29, 2006

1. *Dpt. of Mathematics and Statistics, TTU, Lubbock, Texas, USA*
2. *DIENCA-Lab. di Montecuccolino, Via dei Colli 16, 40136 Bologna, Italy*
3. *LMM, UPMC, 4 place Jussieu, 75252 Paris Cedex 05, FRANCE*

Keywords: VOF/PLIC methods, interface reconstruction, Lagrangian and Eulerian advection, mass conservation, incompressible flow

*Corresponding author: Tel.: (+39)-051-6441720; Fax: (+39)-051-6441747; E-mail address: raus@mail.ing.unibo.it

Abstract

We present and analyse two new Volume-of-Fluid (VOF) reconstruction algorithms that approximate the interface separating two immiscible fluids as a linear function in each grid cell (PLIC - Piecewise Linear Interface Calculation). The first one is based on two simple geometrical criteria for the reconstruction of a linear interface, the second one minimizes a distance functional to find the plane coefficients. Their performance is tested for several smooth surfaces. The geometrical nature of operator split advection is quickly reviewed and a new three-dimensional split advection algorithm is presented. It is exactly mass conserving for a divergence-free velocity field and its accuracy rapidly increases as the CFL number is decreased. Second-order convergence is found in the case of velocity fields with not uniform vorticity at high grid resolutions in the asymptotic regime.

1 Introduction

In fluid mechanics, many techniques have been used to track the evolution of a curve or surface. These methods have been successfully applied to industrial and natural flows such as sea waves, jet atomization, splashes, droplet/bubble oscillations and their breakup and coalescence. Among the most popular tracking methods there are the particle tracking or surface-marker method, the level-set method and the Volume-of-Fluid (VOF) method.

In the surface-marker method the interface is approximated by a set of straight lines (or splines) joining marker points on the interface, or a triangulated mesh in three dimensions. Marker particles are advected in a Lagrangian manner following the velocity streamlines and triangulations need to be redefined at regular intervals (see for example [1]).

In the level-set method, a smooth function F is initialized to the signed distance function to the interface, then the interface coincides with the zero level of this function. After advection, the level-set function has to be reset to the distance function after a prescribed number of time steps.

In the Volume-of-Fluid (VOF) method, a color or volume fraction quantity C is initialized in each cell to the fraction of the volume of the cell filled with a reference phase. It is based on the characteristic function χ that has the value 1 in the reference phase and 0 in the other phase or vacuum. If the two fluids are immiscible each elementary fluid parcel does not change its phase in time, therefore the function χ is passively advected by the flow and satisfies

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

The three methods have often been combined to yield hybrid or mixed methods. Markers and level sets have been proposed in [2], level set and VOF in [3] while VOF and markers have been studied in [4]. Each method has drawbacks and advantages. Marker or tracking methods are usually piecewise-linear approximations to the surface in three dimensions, but may be higher-order approximations in two dimensions [5].

This paper is about the design of a practical, three-dimensional, standalone VOF scheme, which is nearly second-order accurate. The main focus will be to improve the algorithms for the computation of the normal in the reconstruction step and for the conservation of mass in the advection step which, so far, remain open problems.

The volume of fluid method has a unique advantage which is its potential to conserve volume. Limits to this potential come from the numerical methods in use and, in practice, it is very hard to find a good VOF method that satisfies this property exactly.

Most popular VOF methods solve the advection equation with a split technique. In a two-dimensional split method, the fluxes along one coordinate direction are first computed and the C data are updated to an intermediate level. The interface is then reconstructed and the fluxes along the other direction are calculated to update the volume fraction field to the next discrete time level. Errors to the exact conservation of mass may come from the fact that adding the fluxes leads to inconsistencies in the C field. In fact, as we update the C data in some locations the *consistency property* ($0 \leq C \leq 1$) of the volume fraction may not be verified. Another form of inconsistency may occur when we get $C < 1$ in the middle of a region where $C = 1$ everywhere at the previous time step. On the other hand in the past few years a lot of effort has been devoted to improve the VOF method by making it *unsplit* [18, 6, 7].

In this paper some effort has been devoted to investigate the consistency and volume conservation of the split method in three dimensions. Recently, the authors have suggested a method in two dimensions which is both consistent and conserves volume exactly [8, 9]. This method is here extended and tested in three dimensions.

In VOF/PLIC (Piecewise Linear Interface Calculation) methods the interface is approximated by a portion of a plane in each cut cell [10, 11, 12, 13]. The other topic discussed in this work is the searching of schemes that make the computation of the interface normal easier and more accurate. The ELVIRA method [14] is second-order accurate, but rather expensive in three-dimensions [15]. In this paper we discuss two simple geometrical criteria to select a preliminary reconstruction among a set of candidates. Points on this reconstructed interface are then selected in order to minimize a distance functional that yields a new planar interface in each cut cell. This new reconstruction algorithm is second-order accurate when applied iteratively, and it is not computationally very intensive.

We begin in Section 2 with the description of reconstruction algorithms and their performance with grid refinement by considering the reconstruction of several smooth surfaces, such as planes, spheres and a sinusoidal surface. In Section 3 we quickly review split advection schemes in one and two dimensions and then discuss the new three-dimensional split algorithm. We first study its performance with velocity fields that do not deform the initial shape of the fluid body, such as solid body translations and rotations, and then with a divergence-free flow that stretches considerably the fluid object. Finally we present our conclusions.

2 Interface reconstruction algorithms

We consider a three-dimensional computational domain with cubic cells of side $\Delta x = \Delta y = \Delta z = h$. The volume fraction C represents the discrete version of the function χ

$$C_{ijk}(t) = \frac{1}{V_{ijk}} \int_{V_{ijk}} \chi(\mathbf{x}, t) d\mathbf{x}, \quad (2)$$

where V_{ijk} is the volume of the computational grid cell (i, j, k) . In VOF/PLIC methods the interface is represented in each cut cell by a portion of a plane, perpendicular to the local gradient ∇C of the volume fraction C , defined by the equation

$$\mathbf{m} \cdot \mathbf{x} = m_x x + m_y y + m_z z = \alpha. \quad (3)$$

The problem is to determine the constants (m_x, m_y, m_z, α) so that the cut volume V under the plane (3) in the cell (i, j, k) is equal to $h^3 C$. The numerical methods to calculate the normal vector are usually based on finite difference approximations of the volume fraction gradient ∇C and may satisfy some other minimizing criteria. In this paper we discuss different algorithms to approximate an interface in the central cell of a $3 \times 3 \times 3$ block of cells by using their volume fraction values. Once the normal vector \mathbf{m} and the cut volume $h^3 C$ are known, then the value of α is computed by enforcing volume conservation. Geometrically this constraint is applied by moving the interface plane (3) along the normal direction, changing in this way the free parameter α , until the volume under the plane is equal to $h^3 C$. In subsection 2.1 we present an expression for the cut volume, area and its center of mass as a function of α and in subsection 2.2 we discuss two new algorithms for the evaluation of the normal vector \mathbf{m} . Results of a few numerical tests are given in subsection 2.3.

2.1 Expressions for the cut volume, cut area and its center of mass

The intersection of the plane (3) with a cube of side h is a polygon with a number of sides varying from three to six, as schematically represented on Fig. 1. In our

notation the vector \mathbf{m} normal to the interface is pointing outside the region where the reference phase is located, towards the secondary phase where $C = 0$. In the case of Fig. 2 the three coefficients m_i are positive and the cut volume $V = h^3 C$ is the volume under the quadrilateral ABCD of area A . The following expression for the cut volume $V = V(\mathbf{m}, \alpha; h)$ was derived in [16]

$$V = h^3 C = \frac{1}{6m_1 m_2 m_3} \left[\alpha^3 - \sum_{i=1}^3 F_3(\alpha - m_i h) + \sum_{i=1}^3 F_3(\alpha - \alpha_{\max} + m_i h) \right], \quad (4)$$

with $m_i \geq 0$, $\alpha_{\max} = h \sum_{i=1}^3 m_i$ and $F_n(z) = z^n$ when $z > 0$ and zero otherwise. A geometrical interpretation of expression (4) for the case depicted in Fig. 2 is that the volume V is computed first by considering the volume of the right tetrahedron under the triangle AEH of area A_0 , then by subtracting the volume under the two triangles CEG and BFH, respectively of area A_1 and A_2 and finally by adding back the volume of the tetrahedron under the triangle DFG of area A_4 , which was subtracted twice in the previous operations. Furthermore, we notice that all the tetrahedra and triangles involved in this computation are similar. The relation (4), given the normal vector \mathbf{m} and the side h , is a one-to-one function connecting C and α , which implies that the inverse relation $\alpha = \alpha(\mathbf{m}, C; h)$ is uniquely defined. Analytical relations for the direct and inverse functions can be found in [17].

It is also possible to derive an expression for the area A of the cut polygon. The procedure is exactly the same, but instead of adding and removing the volume of similar right tetrahedra we now operate with the area of similar triangles. With the previous definitions and with reference to Fig. 2 we have $A = A_0 - A_1 - A_2 + A_4$, while the more general expression for the area of the cut polygon is

$$A = M_{123} \left[\alpha^2 - \sum_{i=1}^3 F_2(\alpha - m_i h) + \sum_{i=1}^3 F_2(\alpha - \alpha_{\max} + m_i h) \right], \quad (5)$$

where

$$M_{123} = \frac{\sqrt{m_1^2 + m_2^2 + m_3^2}}{2m_1 m_2 m_3}.$$

Finally, we can easily compute the local coordinates of the center of mass \mathbf{x}_g of the area A . For the geometry in Fig. 2 we have $A \mathbf{x}_g = A_0 \mathbf{x}_0 - A_1 \mathbf{x}_1 - A_2 \mathbf{x}_2 + A_4 \mathbf{x}_4$, which is a particular case of

$$A \mathbf{x}_g = M_{123} \left[\alpha^2 \mathbf{x}_0 - \sum_{i=1}^3 \left(F_2(\alpha - m_i h) \mathbf{x}_i - F_2(\alpha - \alpha_{\max} + m_i h) \mathbf{x}_{3+i} \right) \right], \quad (6)$$

where the position of the center of mass \mathbf{x}_i of the triangle A_i is easily determined. Alternatively, one can determine the intersections of the plane with the sides of the central cell, order consecutively this set of points, a variable number between three and six, subdivide the area A in simpler geometrical figures whose center of mass can be easily computed, and then find \mathbf{x}_g . We have developed such an algorithm and even if this procedure is more suited for unstructured grids, the geometrical approach previously described is about three times faster.

2.2 Evaluation of the interface normal

We now describe two three-dimensional algorithms of increasing complexity and accuracy to compute the normal vector. The first one is based on some heuristic criteria discussed in the next section, while the second one minimizes a distance functional to compute the constants m_i of the plane equation (3).

2.2.1 The height function and two geometrical criteria for the reconstruction of linear interfaces

Consider, as in Fig. 3, a linear interface in the 3×3 block of square cells of side h . The volume fraction values can be added columnwise to define a local height function $y = f(x)$ or rowwise for the width function $x = g(y)$. For example, the height y_{i-1} at the abscissa x_{i-1} , placed in the center of the column, is given by the expression $h y_{i-1} = h^2 \sum_{k=-1}^1 C_{i-1,j+k}$. In the case of Fig. 3a we write the equation of the straight line as $\text{sgn}(m_y) y = m_x x + \alpha'$, where $m_y = -\partial C / \partial y$. The sign of m_y is needed because we lose track of what phase is on the top or on the bottom when we integrate the C data to get the height function; we compute the sign with centered finite differences. The angular coefficient m_x can be approximated with backward, centered and forward finite differences, $m_{xb} = (y_i - y_{i-1})/h$, $m_{xc} = (y_{i+1} - y_{i-1})/2h$ and $m_{xf} = (y_{i+1} - y_i)/h$, respectively. The coefficient m_{xb} is equal to m_x , since the two local heights y_i and y_{i-1} are on the interface. This happens when the straight line cuts two opposite sides of the column. In the case of the centered and forward estimates, the line intersects two adjacent sides of the column ($j+1$) and the height y_{i+1} is not on the interface. A small area outside the block of cells is missing and the last two schemes underestimate the value of m_x . Therefore, if two estimates, say m_{x1} and m_{x2} , are available for the same angular coefficient m_x , we select one of them with the following criterion based on the reconstruction of a linear interface

$$|m^*| = \max(|m_{x1}|, |m_{x2}|), \quad (7)$$

where the absolute value is required since the angular coefficients may also be negative. We can also use the width function by considering the line equation $\text{sgn}(m_x) x = m_y y + \alpha''$. For the almost vertical line of Fig. 3b any finite difference scheme calculates the correct value m_y , as the line cuts two opposite sides of each row. On the other hand if we use the height function $y = f(x)$, any discrete estimate of m_x satisfies $|m_x| \leq 3$, a rather crude approximation for a value that should go to infinity as the line becomes vertical. Therefore, between any two approximations m_{x1} and m_{y1} for the angular coefficients m_x and m_y , we now choose

$$|m^*| = \min(|m_{x1}|, |m_{y1}|). \quad (8)$$

In this way we use the form $y = f(x)$ if the interface line is almost horizontal and $x = f(y)$ if the interface is about vertical. The above criteria optimize the

reconstruction of a linear interface, but they may not select the best guess in the case of a curved interface. In Fig. 3c the interface line $y = f(x)$ has a vertical axis of symmetry in the middle of the central column, where $f'(x) = 0$. In this case the best approximation is m_{yc} , since $m_{yc} = 0$, but this would require the minimum value in Eq. (7). However, numerical tests show that on average the two relations (7-8) are the best ones.

A different approach is considered in the ELVIRA algorithm that computes backward, forward and central finite differences for both m_x and m_y , for a total of 6 angular coefficients m_i . For each m_i the intercept α is computed and the straight line is extended to the 3×3 block of cells of Fig. 3, defining in this way a tentative volume fraction \tilde{C} in the neighboring cells. The selected coefficient m_i minimizes the sum of squares error between the actual C and the tentative \tilde{C} data [14]. In the extension to the three-dimensional space a $5 \times 5 \times 5$ block of cells is considered with a total of 72 to 144 candidate normal vectors and associated α . The minimization involves a local height function, defined over columns of 5 cells along one of the three coordinate directions, and not directly the C data [15]. The two-dimensional and three-dimensional algorithms reconstruct any linear interface exactly.

In the three-dimensional space we define the local height function $z = f(x, y)$ by summing the C data along the vertical direction, $h^2 z_{i,j} = h^3 \sum_{l=-1}^1 C_{i,j,k+l}$. This height will be correctly located on the linear interface only if the plane cuts the four vertical sides of the square column. This is not the case for the plane $x + y + z = \alpha$ in Fig. 4. An extra layer of cells must be considered on the top and the bottom of the block. It is straightforward to show that for any $0 \leq \alpha \leq 3$, corresponding to $0 \leq C_{ijk} \leq 1$, the plane now cuts two consecutive columns in the x and y directions so that an exact evaluation of m_x and m_y is possible with finite differences. When the three coefficients m_i are not equal, the 5 cells are taken along the maximum component of the numerically computed volume fraction gradient ∇C . However, a stencil with 45 cells is still quite large and we consider only blocks of $3 \times 3 \times 3$ cells. With the local height z we write the plane equation as $\text{sgn}(m_z) z = m_x x + m_y y + \alpha$. Furthermore, the triplet $(\text{sgn}(m_z), m_x, m_y)$ is normalized to (m_z^0, m_x', m_y') , with the sum of the absolute value of the three components equal to 1. Because of this constraint, if we have two competing planes written as $z = f(x, y)$, instead of taking the maximum of the sum $|m_x'| + |m_y'|$ as required by relation (7), we consider

$$|m^*| = \min(|m_{z1}^0|, |m_{z2}^0|). \quad (9)$$

Similarly, if we have two different planes, written as $z = f(x, y)$ and $y = g(x, z)$ respectively, we select one of the two according to relation (8), now restated as

$$|m^*| = \max(|m_{z1}^0|, |m_{y1}^0|). \quad (10)$$

The two criteria (9-10) do not involve the line constant α which is computed only once when the selection process has been completed.

2.2.2 Mixed Youngs-Centered (MYC) method

To apply these criteria we need to consider a set of normal vectors. We compute them with Youngs' method, for its good behaviour at low resolution, and with the Centered Columns scheme, better at higher resolution, and call the scheme the Mixed Youngs-Centered method. This method provides the preliminary reconstruction for the least-squares fit procedure discussed in (2.2.3).

Youngs' method

The interface normal is evaluated as the gradient of the C function, $\mathbf{m} = -\nabla_h C$, with finite differences. The normal is first estimated at the eight corners of the central cell (i, j, k) , as shown in Fig. 5. In particular, the normal components (m_x, m_y, m_z) in the vertex of coordinates $(x_{i+1/2}, y_{j+1/2}, z_{k+1/2})$ are

$$m_x = \frac{1}{h}(\bar{C}_i - \bar{C}_{i+1}); \quad m_y = \frac{1}{h}(\bar{C}_j - \bar{C}_{j+1}); \quad m_z = \frac{1}{h}(\bar{C}_k - \bar{C}_{k+1}),$$

where, for example, $\bar{C}_i = (C_{i,j,k} + C_{i,j+1,k} + C_{i,j,k+1} + C_{i,j+1,k+1})/4$. A similar finite difference scheme is applied in the other seven vertices. The cell-centered normal vector is finally obtained by averaging the eight cell-corner values.

Centered Columns (CC) method

We approximate the function $z = f(x, y)$ in the central cell of the block with the linear equation $\text{sgn}(m_z) z = m_x x + m_y y + \alpha$. The sign of the coefficient $m_z = -\partial C / \partial z$ is computed with finite differences by using the C data in the top and bottom layers of Fig. 5. For the other two components of the normal vector we consider a centered scheme based on the height function z

$$m_x = \frac{z_{i+1,j} - z_{i-1,j}}{2h} = \frac{1}{2} \left(\sum_{l=-1}^1 C_{i+1,j,k+l} - \sum_{l=-1}^1 C_{i-1,j,k+l} \right)$$

and

$$m_y = \frac{z_{i,j+1} - z_{i,j-1}}{2h} = \frac{1}{2} \left(\sum_{l=-1}^1 C_{i,j+1,k+l} - \sum_{l=-1}^1 C_{i,j-1,k+l} \right).$$

We also consider the height function along the x and y directions, define the two linear functions $x = h(y, z)$ and $y = g(x, z)$ and calculate the corresponding normal vectors.

We have now a set of four different planes. First we select the best of the three computed with the CC scheme by using (10), then we select between this best one and the plane obtained with Youngs' method with (9).

2.2.3 Least-Squares Fit (LSF) method

In this method we extend to the three-dimensional space a two-dimensional technique described in [8]. The procedure is divided in three steps: 1) choice of a convenient set of points inside the $3 \times 3 \times 3$ block of cells; 2) construction of a weight function and weight assignment to each point in the set; 3) minimization of a suitable distance functional in order to find the plane coefficients. To define the set of points we consider the interface reconstructed with the MYC method and select one point in each cell crossed by the interface. This point is the center of mass \mathbf{x}_g of the area of the polygon cut by the reconstructed plane and the grid cell. We then consider a Gaussian function to weight the point contribution to the distance functional. In particular, we denote with $\mathbf{x}_c = (x_C, y_C, z_C)$ the position of the center of mass of the central cell and define

$$\begin{aligned} d_i &= \sqrt{(x_{gi} - x_C)^2 + (y_{gi} - y_C)^2 + (z_{gi} - z_C)^2}, \\ d_m &= \sum_{i=1}^{n_p} d_i / n_p, \\ \sigma^2 &= \sum_{i=1}^{n_p} (d_i - d_m)^2 / (n_p (n_p - 1)), \end{aligned}$$

where $n_p \leq 27$ is the total number of points, d_i is the distance of each point from \mathbf{x}_c , d_m the average distance and σ^2 the variance. The weights ω_i are first defined as $\omega_i = \exp(-d_i^2 / (a \sigma^2))$, where a is a free parameter (here $a = 0.75$), and then their sum is normalized to one, $\omega_T = \sum_{i=1}^{n_p} \omega_i$ and $\omega_i = \omega_i / \omega_T$. From the preliminary reconstruction based on the the two relations (9-10), we know if the plane equation is written as $z = f(x, y)$, rather than $x = h(y, z)$ or $y = g(x, z)$. In the first case we store the sign of m_z and move the origin of the local coordinate system to the center of mass of the n_p points with coordinates (X_G, Y_G, Z_G) , where, for example, $X_G = \sum_{i=1}^{n_p} (\omega_i x_{gi})$. The functional H for this case is defined by

$$H = \sum_{i=1}^{n_p} \left(\omega_i (Z_{gi} - (m_x X_{gi} + m_y Y_{gi}))^2 \right), \quad (11)$$

and is minimized by taking $\partial H / \partial m_x = \partial H / \partial m_y = 0$. The solution of this simple linear system provides the values of m_x and m_y and then we complete the interface reconstruction in the central cell by computing the intercept α [17].

2.3 Reconstruction tests

We now examine the accuracy and convergence properties of the methods previously described for well-behaved interfaces. The choice of the surfaces is motivated by their local mean curvature, which is zero for a plane, constant for a sphere and variable, in both magnitude and sign, for a sinusoidal surface.

2.3.1 Error measure, convergence rate and numerical integration

Let $\chi(\mathbf{x})$ be the characteristic function associated to a fluid body and $\tilde{\chi}_h(\mathbf{x})$ be its approximation obtained with a VOF reconstruction method in a grid with cubic cells of side h . A natural measure of the difference between the exact interface and the reconstructed one is the geometrical error E_h in L_1

$$E_h = \int \int \int |\chi(\mathbf{x}) - \tilde{\chi}_h(\mathbf{x})| dx dy dz. \quad (12)$$

The order of convergence \mathcal{O} of a reconstruction method can be numerically calculated by considering the errors E_h , obtained with grid spacing h , and $E_{h/2}$ with

$$\mathcal{O} = \frac{\ln(E_h/E_{h/2})}{\ln(h/(h/2))} = \frac{\ln(E_h/E_{h/2})}{\ln(2)}. \quad (13)$$

In general, this expression is a function of the fluid body shape, its position and orientation with respect to the grid lines, and of the grid spacing h . The error E_h should decrease with h , however at very low resolution by halving h it decreases much more than at very high resolution. Therefore, the expression (13) is usually a decreasing function of the resolution h converging towards the asymptotic order of convergence in the limit as $h \rightarrow 0$. We initialize the volume fraction C and calculate the error E_h numerically, in a unit cube partitioned in n^3 cells, with $n = 1/h$. As an example, suppose that around the cell (i, j, k) we can write the surface equation as $f(x, y) - z = 0$, then the value of C in the cell (i, j, k) is defined by

$$h^3 C = \int_{(i-1)h}^{ih} \int_{(j-1)h}^{jh} \int_{(k-1)h}^{kh} \chi(\mathbf{x}) dx dy dz = \int_{(i-1)h}^{ih} \int_{(j-1)h}^{jh} z^* dy dx, \quad (14)$$

where $z^* = \min(h, \max(f(x, y) - (k-1)h, 0))$ is in the range $[0, h]$. We subdivide the base of the cell in l^2 squares, compute the z^* values in the four corners of each square and calculate (14) numerically with a two-dimensional Simpson's rule. For reliable results, the error E_h defined in (12) should be several orders of magnitude bigger than the error between the exact value of the C function and the numerical data (14). On the other hand, the CPU time increases linearly with the number l^2 of squares, and we have found a good compromise for $l = 30$. Furthermore, for each geometry we consider 100 different cases with randomly generated coefficients in order to stabilize the error E_h . This is necessary because the error could be particularly small due to favorable alignments between the interface and the grid lines. Finally, we consider five different partitions of the unit cube in n^3 cells, with n in the set $(10, 20, 40, 80, 160)$.

2.3.2 Reconstruction of a plane

We consider the plane equation $m_x x + m_y y + m_z z = \alpha$, where the four coefficients (m_x, m_y, m_z, α) are randomly generated. We see in Tab. 1 that Youngs' scheme has

the worst performance, the CC and MYC methods are intermediate, while the error of LSF is at least two orders of magnitude smaller than the other ones. As expected from the discussion in section (2.2.1) none of the proposed reconstruction methods can reproduce exactly all linear interfaces. However, notice that the two criteria (9-10) select efficiently between Youngs' and the CC schemes. Moreover, the LSF method can be used as a starting guess for another LSF procedure. We observe that the errors decrease by at least 5 orders of magnitude with the first iteration and after the second one are below machine accuracy. The use of a weight function in the distance functional accelerates the convergence towards the exact linear interface. A second-order method is expected to reconstruct all linear interfaces exactly, but this would require at least one further application of the LSF method. However, the next tests demonstrate that with a curved interface we only need the very first LSF reconstruction and still the algorithm performs as a second-order accurate method even at high grid resolutions. Therefore, in all the following tests we reconstruct the interface only once with the least-squares fit.

2.3.3 Reconstruction of a sphere

We consider the sphere equation $(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = R^2$, where $R = 0.325$ and the center coordinates (x_c, y_c, z_c) are randomly generated near the point $(0.5, 0.5, 0.5)$ in order to keep the whole sphere inside the unit cube. The results are shown in Tab. 2 and are consistent with the results obtained in two dimensions with a circle. Youngs' scheme is competitive only at a very low resolution, the MYC algorithm performs better than Youngs' and the CC methods in all but one case, while the LSF scheme provides the best results. More particularly, the convergence rates of Fig. 6 show that Youngs' scheme is quickly degrading to first-order, while the CC and MYC methods are not yet close to an asymptotic regime. Nevertheless, since the resolution $n = 160$ is very high, the convergence rate of these two methods can be considered between 1 and 2 in most applications. The LSF method performs as a second-order algorithm and in Fig. 7 the reconstructed spherical interface is rather good even at the lowest resolution with $n = 10$. Thus on average, in three dimensions we need a local radius of curvature at least three to four times bigger than the grid spacing for an accurate interface reconstruction. The thickness of a filament should be similar, to avoid the presence in the same stencil of interface regions with opposite normal vectors.

2.3.4 Reconstruction of a sinusoidal surface

Here we consider the sinusoidal function $z = 0.5 + \sin(\pi ax) \sin(\pi by)$, where the two parameters a, b are in the range $[0, 1]$ and are randomly generated. In Fig. 8 the convergence rates are similar to those of the previous test, while the computed reconstruction errors are about two to three times bigger. The interface has a more

complex structure and the absolute value of the radius of curvature is locally smaller than in the sphere test. In Fig. 9 at the lowest resolution $n = 10$ the sinusoidal surface is not well reproduced even by the LSF reconstruction, but at $n = 20$ the interface is much better resolved. As previously stated, when the local radius of curvature is comparable with the grid spacing h the surface is poorly approximated by a plane and the interface discontinuity at the cell boundary is of order h [8].

3 Interface advection

We consider again the advection equation (1) for the characteristic function χ recast in conservative form for an incompressible flow

$$\frac{\partial \chi}{\partial t} + \nabla \cdot (\chi \mathbf{v}) = \chi \nabla \cdot \mathbf{v} = 0. \quad (15)$$

We integrate it over the cell volume V to find the evolution equation for the volume fraction C [4]

$$h^3 \frac{\partial C}{\partial t} + \int_{\Gamma} \chi \mathbf{v} \cdot \mathbf{n} d\Gamma = \int_V \chi (\nabla \cdot \mathbf{v}) dV = 0, \quad (16)$$

where Γ is the boundary of the grid cell and \mathbf{n} its outgoing normal. The surface integral represents the net reference phase flux through Γ . In the reconstruction step we approximate the interface with a portion of a plane, defining in this way a function $\tilde{\chi}$ that satisfies

$$h^3 C = \int_V \chi dV = \int_V \tilde{\chi} dV.$$

We then advect the interface in a given velocity field by computing in (16) the boundary fluxes of $\tilde{\chi}$, to update the C data at the next discrete time. Either multidimensional (unsplit) schemes or a sequence of one-dimensional (operator split) schemes have been considered. In the first category the fluid flowing in the time step Δt through a cell face comes from different neighbouring cells and the fluxing volume is a rather complex geometrical figure. A Cartesian geometry is usually adopted in the second category. All the points on a plane perpendicular to the direction of propagation move with the same velocity and the fluxing volume is a right hexahedron. The reference phase flux is then given by the portion of this hexahedron cut by the reconstructed interface and is computed by using analytical expressions [17]. A two-dimensional example is shown in Fig. 10, where the fluid volume flowing into the central cell from the left side is the rectangle $A'ADD'$ and the reference phase flux is represented by the shaded portion of this area. Once the one-dimensional cell boundary fluxes are computed, the volume fraction is updated independently along each coordinate direction. In the next subsections we first review in 3.1 a few advection schemes in one and two dimensions, then in 3.2 we present a new three-dimensional algorithm, while the numerical tests are discussed in 3.3.

3.1 Advection schemes in one and two dimensions

Let us consider first the monodimensional version of (16) along the x coordinate with a cell-averaged approximation of the term $\partial u/\partial x$. This term represents a compression or expansion along x that may differ from zero even if the multidimensional flow is incompressible [18]. We discretize this equation on a staggered MAC grid [19], with cells of side h and use a simple forward scheme in time with $\Delta t = t^{n+1} - t^n$

$$C_{ijk}^{n+1} = C_{ijk}^n + F_{left} - F_{right} + \tilde{C}_{ijk}(u_{right} - u_{left}), \quad (17)$$

where u is from now on the CFL number ($u \Delta t/h \rightarrow u$) and F_{left} and F_{right} denote the volume fraction fluxes across the left and right faces of the cell (i, j, k) . We consider two monodimensional schemes, corresponding to a different value of \tilde{C}_{ijk} . A simple geometrical interpretation shows that both schemes satisfy the consistency property: $0 \leq C_{ijk}^{n+1} \leq 1$.

3.1.1 Eulerian-Implicit (EI) scheme

We set $\tilde{C}_{ijk} = C_{ijk}^{n+1}$, the scheme is implicit and (17) becomes

$$C_{ijk}^{n+1} = \frac{C_{ijk}^n + F_{left} - F_{right}}{1 - (u_{right} - u_{left})}. \quad (18)$$

A two-dimensional geometrical interpretation of this equation is shown in Fig. 10, where the grid cell is the square $ABCD$. The two "Eulerian" fluxes F are computed from the volume fractions at time n and are defined by the gray areas inside the rectangles $A'ADD'$ and $B'BCC'$.

3.1.2 Lagrangian-Explicit (LE) scheme

We now set $\tilde{C}_{ijk} = C_{ijk}^n$ in Eq. (17) and the scheme is explicit

$$C_{ijk}^{n+1} = C_{ijk}^n(1 + (u_{right} - u_{left})) + F_{left} - F_{right}. \quad (19)$$

The fluxes are now the grey areas inside the rectangles $A''ADD''$ and $B''BCC''$ of Fig. 11: the reconstructed interface is first advected and then the fluxes are computed. The Lagrangian advection by a one dimensional flow of a VOF/PLIC reconstruction changes its orientation, as shown in the figure, moreover the areas computed in the LE method are expanded or compressed by the divergence term, hence the EI and LE fluxes are different [8].

3.1.3 The combined EI-LE scheme

We can combine these two schemes into a two-dimensional advection technique which can be either split or unsplit. If the flow is incompressible, the discrete version of $\nabla \cdot \mathbf{v} = 0$ is written as $(u_{right} - u_{left}) + (v_{up} - v_{down}) = 0$. With reference to Fig. 12, the computational domain is meshed with square cells, such as $ABCD$, and the advection scheme corresponds to a mapping from a tessellation of the plane, with rectangles $A'B'C'D'$, into another tessellation of the domain with rectangles $A''B''C''D''$. The mapping is given by the following piecewise linear affine transformation

$$\Pi_{xy} = \begin{cases} x' = a(x + u_{left}) \\ y' = by + v_{down}, \end{cases} \quad (20)$$

where $a = 1/(1 - u_{right} + u_{left})$ and $b = 1 + v_{up} - v_{down}$. The implicit step is in the x direction and maps $A'B'C'D'$ into $ABCD$; it is followed by the Lagrangian step in the y direction mapping $ABCD$ into $A''B''C''D''$. The method is split if an intermediate reconstruction is performed after the first step. The Jacobian J of the linear transformation Π_{xy} is $J = ab$. It is equal to one if the flow is incompressible, then the scheme conserves the area exactly [9]. Finally, the direction of the first implicit advection should be alternated in time in order to avoid a preferential direction of propagation [20].

3.2 Three-dimensional split advection schemes

In this section we consider a series of consecutive advection steps to advance the interface in three dimensions. We recall that a split VOF/PLIC technique requires an interface reconstruction before each advection step, therefore the change in the total error of the method at each time step comes from both the reconstructions and the advection steps. The sequence of one-dimensional propagations along the coordinate directions should be changed at each time step to remove possible asymmetries.

3.2.1 Three-dimensional split LE scheme

We advance the interface with a sequence of consecutive monodimensional LE advectons, each of them described by (19). We have shown that in two dimensions, when the vorticity field is not constant, this method does not conserve the mass [8]. In three dimensions this method was implemented in the SURFER code [16] and it is here considered only for comparison reasons.

3.2.2 Three-dimensional advection with three EI-LE steps

Let the three-dimensional velocity field $\mathbf{v} = (u, v, w)$ be incompressible, then we define three new fields $\mathbf{v}_1 = (u_1, v_1, 0)$, $\mathbf{v}_2 = (u_2, 0, w_2)$, $\mathbf{v}_3 = (0, v_3, w_3)$ and require each field to be incompressible, $\nabla \cdot \mathbf{v}_i = 0$, and their sum to be equal to the given field, $\sum_{i=1}^3 \mathbf{v}_i = \mathbf{v}$. We have six scalar equations for the the six unknowns $(u_1, v_1, u_2, w_2, v_3, w_3)$, but only five of them are independent. For example, if $\mathbf{v}_1, \mathbf{v}_2$ satisfy $\nabla \cdot \mathbf{v}_1 = \nabla \cdot \mathbf{v}_2 = 0$, we can compute $\mathbf{v}_3 = \mathbf{v} - \mathbf{v}_1 - \mathbf{v}_2$, but then $\nabla \cdot \mathbf{v}_3 = \nabla \cdot (\mathbf{v} - \mathbf{v}_1 - \mathbf{v}_2) = 0$ is automatically satisfied. We can use this degree of freedom and let $u_1 = u/2$, then $u_2 = u - u_1 = u/2$. From the boundary, where we set $v_{1:i,-1/2,k} = v_{i,-1/2,k}/2$, we use the discrete form of $\nabla \cdot \mathbf{v}_1 = 0$ on a staggered MAC grid

$$v_{1:i,j+1/2,k} - v_{1:i,j-1/2,k} + u_{1:i+1/2,j,k} - u_{1:i-1/2,j,k} = 0 \quad (21)$$

and solve it for $v_{1:i,j+1/2,k}$. As we move away from the boundary there is only one unknown in the previous equation and the field component v_1 is readily available. Then we compute $v_3 = v - v_1$. In a similar way we calculate w_3 , from $\nabla \cdot \mathbf{v}_3 = 0$, and $w_2 = w - w_3$. To reduce asymmetries, we should alternate at each time step the velocity component that was set arbitrarily at the beginning of the procedure in virtue of the degree of freedom. In the advection tests we have implemented a slightly different approach. We first let $u_1 = u/2$ and calculate $(u_2, v_1, v_3, w_2, w_3)$ as previously described, then repeat the same procedure twice, first by letting $v_3 = v/2$ and then $w_2 = w/2$. The velocity field is then given by one third of the sum of these three sets of incompressible fields and it is called EILE-3D. To illustrate its features we also consider the following simplified decomposition (EILE-3DS)

$$u_1 = u_2 = u/2; \quad v_1 = v_3 = v/2; \quad w_2 = w_3 = w/2,$$

where each velocity field \mathbf{v}_i is not divergence free.

3.3 Advection tests

In this section we analyze the performance of the reconstruction and advection schemes over standard three-dimensional tests. First we consider simple flows such as uniform translations and rotations where a smooth fluid body should be advected without significant distortion and the mass should be conserved exactly. We then study the performance of the proposed algorithms in three-dimensional flows with not uniform vorticity.

3.3.1 Geometrical and mass errors

To quantify the results we consider two widely used discrete error norms in L_1 . The first one is the relative mass error $E_m(t_1)$ between the total volume occupied by the

reference phase at the initial time t_0 and that of the fluid body at time t_1

$$E_m(t_1) = \frac{|\sum_{ijk} C_{ijk}(t_0) - \sum_{ijk} C_{ijk}(t_1)|}{\sum_{ijk} C_{ijk}(t_0)}. \quad (22)$$

The second one is the geometrical error between the position of the reference phase at the two instants (t_0, t_1)

$$E_g(t_1) = \frac{\sum_{ijk} |C_{ijk}(t_0) - C_{ijk}(t_1)|}{\sum_{ijk} C_{ijk}(t_0)}. \quad (23)$$

In both equations the cell volume h^3 has been simplified. The error E_g is meaningful only when the fluid body at time t_1 should be back to its initial configuration.

3.3.2 Translation and solid body rotation

The velocity field is not only incompressible, but also $\partial u/\partial x = \partial v/\partial y = \partial w/\partial z = 0$. Therefore, the monodimensional compression or expansion term in (17) vanishes and this leads to the same geometrical and mass errors for all methods. For both tests we consider a sphere of radius $R = 0.15$ and center in the unit box partitioned with n^3 cells of side $h = 1/n$ and resolution $n = 16, 32, 64, 128$. In the translation test the sphere center is at $(0.5, 0.5, 0.5)$ and the constant velocity field is $(h/2\Delta t, h/2\Delta t, h/2\Delta t)$ at $t = t_0$. The sphere moves along the main diagonal and it is back to its position after $2n$ steps when we change the sign of the velocity for other $2n$ steps. Periodic boundary conditions are applied on the box boundary. In Fig. 13 we show the fluid body motion after $0, (2n+1)/3, (4n-1)/3$ and $2n$ steps, with resolution $n = 64$. The relative mass error is zero and the geometrical error varies only with the reconstruction method. The results obtained with the MYC and LSF reconstructions are given in Tab. 3. The least-squares fit performs on average better, but its convergence rate is about 1.6. This is a somewhat expected result, since there are three interface reconstructions and monodimensional advections at each time step.

In the rotation test the sphere, with center at $(0.7, 0.5, 0.5)$, performs a rotation in $3n$ steps, then we change the sign of the angular velocity and the sphere goes back to the initial position in the same number of steps. The rotation axis is along the line $x = 0.5; y + z = 2$. The results are given in Tab. 4. At high resolution, the performance of the MYC scheme is close to that of the least-squares fit in terms of errors and convergence rate. The same behaviour is found in the next test, therefore results will be presented only for the LSF method.

3.3.3 Single vortex test with not uniform vorticity

The velocity field $\mathbf{v}(x, y, z; t)$ in the unit cube is defined by

$$\begin{aligned} u(x, y, z; t) &= \sin^2(\pi x) \cos(\pi t_n/2n) [\sin(\pi(y - 0.5)) - \sin(\pi(z - 0.5))], \\ v(x, y, z; t) &= \sin^2(\pi y) \cos(\pi t_n/2n) [\sin(\pi(z - 0.5)) - \sin(\pi(x - 0.5))], \\ w(x, y, z; t) &= \sin^2(\pi z) \cos(\pi t_n/2n) [\sin(\pi(x - 0.5)) - \sin(\pi(y - 0.5))], \end{aligned}$$

where t_n is the discrete time, $1 \leq t_n \leq 2n$. A sphere of radius $R = 0.15$ and center $\mathbf{x}_c = (0.7, 0.5, 0.5)$ is immersed in this flow and stretched around the center of the unit box. It reaches its maximum deformation at $t_n = n$ and then returns to its initial position [21]. To study the convergence in space we consider the four grid resolutions of the previous tests, $n = 16, 32, 64, 128$, and present the results obtained with the LSF reconstruction and the split advection algorithms previously described. The geometrical and mass errors given in Tabs. 5 and 6, where the maximum CFL number is 1. The convergence rate for the EILE-3DS advection remains greater than 2 and it is about one for the EILE-3D scheme, which however conserves mass to machine accuracy. The sequence of three LE steps has the greatest geometrical and mass errors, with a convergence rate of about 1 in this range of grid resolutions and it will not be considered any longer.

In Fig. 14 we show the initial fluid body configuration and after $n/2, n, 2n$ steps, with $n = 64$ and the EILE-3DS advection. The geometrical error is about 1% and can hardly be seen at this resolution. The local radius of curvature and thickness of the deformed fluid body are at least a few cell size h , and the interface is well resolved in the whole simulation. This is not the case at lower n , where the discontinuity in the reconstructed interface at the cell boundary becomes of order h . The interface is more and more poorly resolved as the simulation goes on, eventually inducing a numerical breakup. However, this is a major issue with every VOF method that limits the interface reconstruction to a single linear equation in a cut cell.

To study the temporal behavior of the reconstruction and advection errors we consider a fixed grid resolution and decrease progressively the time step, or equivalently the monodimensional CFL number, $CFL = u\Delta t/h$. The results with $n = 32$ for the geometrical and mass errors are reported in Tab. 7. The geometrical error is converging towards an asymptotic limiting value as the CFL number vanishes [6, 7]. This constant limit is critical for the feasibility of the VOF technique since, in applied dynamical simulations, stability requirements limit the CFL number to rather small values. However, the EILE-3D advection is the only scheme where the geometric error is consistently decreasing towards the asymptotic value, as shown in Tab. 8. We note that at $n = 128$ and $CFL = 0.01$ the geometrical error is not yet in the asymptotic regime, but the convergence rate is at least second-order accurate.

Both the EILE-3D and EILE-3DS schemes require 6 reconstructions and monodimensional interface advectons, but they are based on different decompositions

of the incompressible three-dimensional velocity field \mathbf{v} . Each field component \mathbf{v}_i ($i = 1, 2, 3$) of the EILE-3DS decomposition satisfies the boundary condition $\mathbf{v}_i = 0$, but it is not divergence free, vice versa in the EILE-3D decomposition only the sum of the \mathbf{v}_i satisfies the boundary conditions, but each \mathbf{v}_i is incompressible. In particular, on the boundary where the velocity is found by the discrete divergence-free condition (21), we observe a second-order convergence with grid refinement to the actual boundary value. At high CFL numbers this error is predominant and the convergence rate of the EILE-3D algorithm is only linear, the other one is second-order. To decrease the CFL number we consider smaller time steps, then the total number of steps increases and the error due the fact that $\nabla \cdot \mathbf{v}_i \neq 0$ accumulates and becomes the most important one for the EILE-3DS scheme. The results show that the performance of the EILE-3DS scheme deteriorates and the error converges to the asymptotic value from below. For the EILE-3D algorithm the error decreases with the time step and converges to a much lower value from above.

4 Conclusions

We have discussed two new PLIC algorithms to reconstruct interfaces in two-phase flows with Cartesian three-dimensional grids. The algorithms approximate the interface with a portion of a plane in each cut cell. In one case we select one plane among a set of candidates with two geometrical criteria based on the reconstruction of a linear interface. In the second scheme we consider the previous reconstruction to select points in order to minimize a distance functional and find the plane coefficients. The former does not reconstruct exactly linear interfaces and its convergence rate for curved surfaces is between 1 and 2. The latter reconstructs linear interfaces exactly when applied iteratively and over a curved interface performs as a second-order accurate method. After reviewing the geometric nature of split advection, we have proposed a new split algorithm. It conserves mass exactly and its accuracy increases rapidly as the CFL number is decreased towards an asymptotic value. For flows that stretch and deform considerably the interface, we observe a second-order convergence in the asymptotic regime.

References

- [1] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, 100:25–37, 1992.
- [2] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.*, 183:83–116, 2002.

- [3] Mark Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *J. Comput. Phys.*, 187:110–136, 2003.
- [4] E. Aulisa, S. Manservigi, and 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.*, 188:611–639, 2003.
- [5] Stéphane Popinet and Stéphane Zaleski. A front tracking algorithm for the accurate representation of surface tension. *Int. J. Numer. Meth. Fluids*, 30:775–793, 1999.
- [6] D. J. E. Harvie and D. F. Fletcher. A New Volume of Fluid Advection Algorithm: The Stream Scheme. *J. Comput. Phys.*, 162:1–32, 2000.
- [7] J. López, J. Hernandez, P. Gómez, and F. Faura. A volume of fluid method based on multidimensional advection and spline interface reconstruction. *J. Comput. Phys.*, 195:718–742, 2004.
- [8] R. Scardovelli and S. Zaleski. Interface reconstruction with least-square fit and split Eulerian-Lagrangian advection. *Int. J. Numer. Meth. Fluids*, 41:251–274, 2003.
- [9] E. Aulisa, S. Manservigi, R. Scardovelli, and S. Zaleski. A geometrical area-preserving Volume-of-Fluid advection method. *J. Comput. Phys.*, 192:355–364, 2003.
- [10] D.L. Youngs. Time dependent multimaterial flow with large fluid distortion. In K. M. Morton and M. J. Baines, editors, *Numerical methods for fluid dynamics*, pages 27–39, New York, 1982. Academic Press. Institute for Mathematics and its Applications.
- [11] Jie Li. Calcul d’interface affine par morceaux (piecewise linear interface calculation). *C. R. Acad. Sci. Paris, série Iib, (Paris)*, 320:391–396, 1995.
- [12] W. J. Rider and D. B. Kothe. Reconstructing volume tracking. *J. Comput. Phys.*, 141:112–152, 1998. Euler split and unsplit, alternating implicit/explicit volume expansion factor.
- [13] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech.*, 31:567–603, 1999.
- [14] J. E. Jr. Pilliod and E. G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *J. Comput. Phys.*, 199:465–502, 2004.
- [15] G. H. Miller and P. Colella. A Conservative Three-Dimensional Eulerian Method for Coupled Solid-Fluid Shock Capturing. *J. Comput. Phys.*, 183:26–82, 2002.

- [16] D. Gueyffier, A. Nadim, J. Li, R. Scardovelli, and S. Zaleski. Volume of Fluid Interface Tracking with Smoothed Surface Stress Methods for Three-dimensional Flows. *J. Comput. Phys.*, 152:423–456, 1999.
- [17] R. Scardovelli and S. Zaleski. Analytical Relations Connecting Linear Interfaces and Volume Fractions in Rectangular Grids. *J. Comput. Phys.*, 164:228–237, 2000.
- [18] W. J. Rider and D. B. Kothe. Reconstructing volume tracking. *J. Comput. Phys.*, 141:112–152, 1998.
- [19] F. H. Harlow and J. E. Welch. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Phys. Fluids*, 8:2182–2188, 1965.
- [20] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5:506–517, 1968.
- [21] R. J. Leveque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis*, 33:627–665, 1996.

| | n=10 | n=20 | n=40 | n=80 | n=160 |
|--------|----------|----------|----------|----------|----------|
| Youngs | 5.743e-4 | 2.863e-4 | 1.431e-4 | 7.159e-5 | 3.580e-5 |
| CC | 1.578e-4 | 7.867e-5 | 3.939e-5 | 1.970e-5 | 9.848e-6 |
| MYC | 7.981e-5 | 4.023e-5 | 2.011e-5 | 1.005e-5 | 5.030e-6 |
| LSF | 2.555e-7 | 1.158e-7 | 5.490e-8 | 2.673e-8 | 1.318e-8 |

Table 1: Reconstruction errors for a plane with different meshes with n^3 cells and reconstruction algorithms

| | n=10 | n=20 | n=40 | n=80 | n=160 |
|--------|----------|----------|----------|----------|----------|
| Youngs | 1.883e-3 | 5.300e-4 | 1.948e-4 | 8.635e-5 | 4.132e-5 |
| CC | 2.430e-3 | 5.379e-4 | 1.390e-4 | 3.895e-5 | 1.223e-5 |
| MYC | 2.151e-3 | 5.148e-4 | 1.285e-4 | 3.381e-5 | 9.658e-6 |
| LSF | 1.921e-3 | 4.767e-4 | 1.191e-4 | 2.980e-5 | 7.463e-6 |

Table 2: Reconstruction errors for a sphere with different meshes with n^3 cells and reconstruction algorithms

| | n=16 | n=32 | n=64 | n=128 |
|-------------|----------|----------|----------|----------|
| E_g (MYC) | 3.852e-2 | 4.436e-2 | 1.388e-2 | 2.862e-3 |
| E_g (LSF) | 4.401e-2 | 1.216e-2 | 5.247e-3 | 1.734e-3 |

Table 3: Geometrical error E_g for the translation of a sphere with different meshes with n^3 cells and reconstruction algorithms

| | n=16 | n=32 | n=64 | n=128 |
|-------------|----------|----------|----------|----------|
| E_g (MYC) | 8.538e-2 | 1.291e-2 | 3.796e-3 | 1.889e-3 |
| E_g (LSF) | 8.937e-2 | 1.092e-2 | 3.731e-3 | 1.785e-3 |

Table 4: Geometrical error E_g for the rotation of a sphere with different meshes with n^3 cells and reconstruction algorithms

| | n=16 | n=32 | n=64 | n=128 |
|------------------|----------|----------|----------|----------|
| E_g (LE) | 4.213e-1 | 1.703e-1 | 8.005e-2 | 4.115e-2 |
| E_g (EILE-3D) | 8.855e-2 | 4.351e-2 | 2.150e-2 | 1.085e-2 |
| E_g (EILE-3DS) | 1.835e-1 | 4.379e-2 | 9.532e-3 | 2.010e-3 |

Table 5: Geometrical error E_g for the single vortex test of a sphere with different meshes with n^3 cells and split advection algorithms

| | n=16 | n=32 | n=64 | n=128 |
|------------------|----------|----------|----------|----------|
| E_m (LE) | 3.464e-1 | 1.664e-1 | 7.252e-2 | 3.505e-2 |
| E_m (EILE-3DS) | 8.174e-4 | 1.065e-4 | 9.084e-6 | 1.444e-6 |

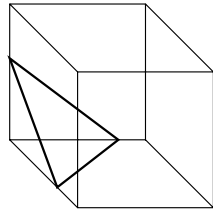
Table 6: Mass error E_m for the single vortex test of a sphere with different meshes with n^3 cells and split advection algorithms

| | CFL=1.0 | CFL=0.1 | CFL=0.01 | CFL=0.001 |
|-------------------|----------|----------|----------|-----------|
| E_g (EILE-3D) | 4.351e-2 | 1.491e-2 | 1.555e-2 | 1.564e-2 |
| E_g (EILE-3DS) | 4.378e-2 | 5.024e-2 | 5.119e-2 | 5.127e-2 |
| E_m (3 EI-LE-S) | 1.065e-4 | 1.323e-6 | 1.433e-8 | 2.247e-9 |

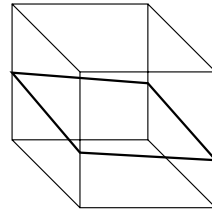
Table 7: Geometrical (E_g) and mass (E_m) errors for the single vortex test of a sphere on a 32^3 mesh with different CFL numbers and split advection algorithms

| | CFL=1.0 | CFL=0.1 | CFL=0.01 |
|---------------|----------|----------|----------|
| E_g (n=64) | 2.150e-2 | 3.658e-3 | 3.294e-3 |
| E_g (n=128) | 1.085e-2 | 1.200e-3 | 6.235e-4 |

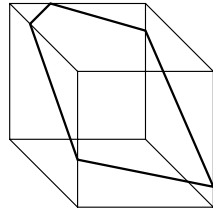
Table 8: Geometrical error for the single vortex test of a sphere on a 32^3 mesh with the 3 EILE-3D split advection, different meshes with n^3 cells and CFL numbers



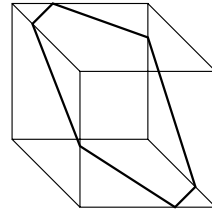
3 intersections



4 intersections



5 intersections



6 intersections

Figure 1: The polygon cut by a plane and a right hexahedron: the number of its sides varies from three to six

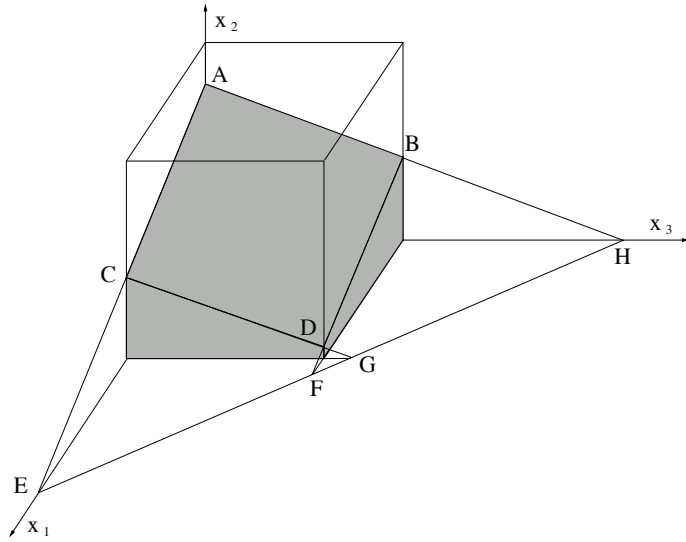


Figure 2: The cut volume is the shaded volume inside the grid cell under the polygon $ABCD$

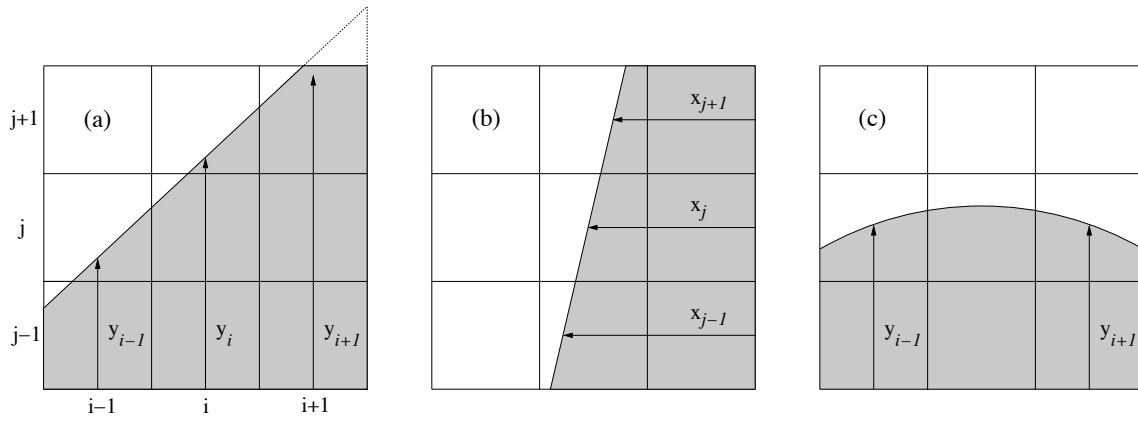


Figure 3: Volume fractions are added columnwise, heights y_{i-1} and y_{i+1} of case (a), or rowwise, widths x_{j-1} x_{j+1} of case (b), to compute the slope of a linear interface

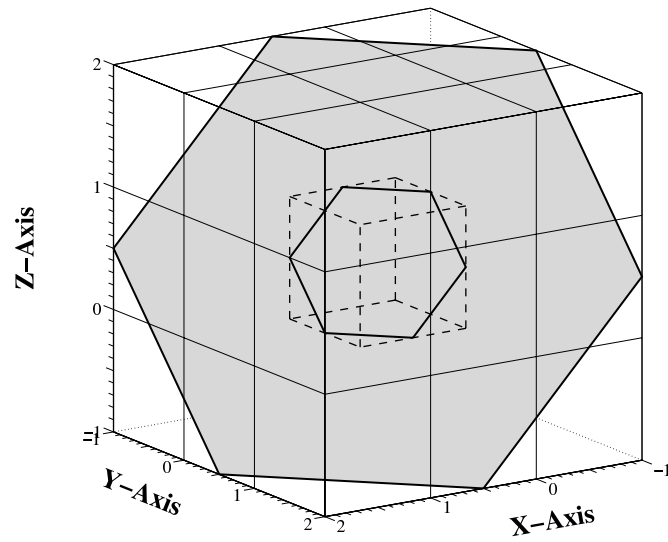


Figure 4: The two shaded hexagons define the intersection of a given plane with a $3 \times 3 \times 3$ block of cubic cells and with its central cell, respectively

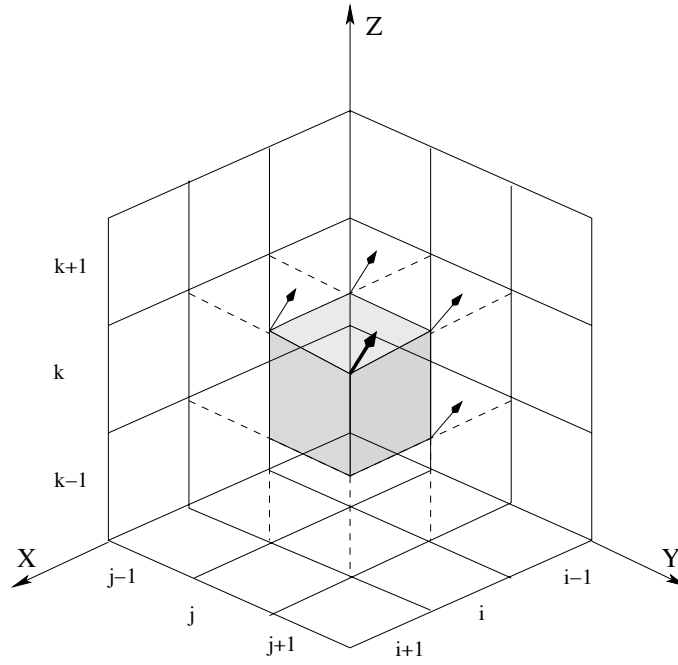


Figure 5: The $3 \times 3 \times 3$ block of cells used in the algorithms to compute the normal vector. A few of the normals in the vertices of the central cell (i, j, k) , that are defined in Youngs' method, are also shown

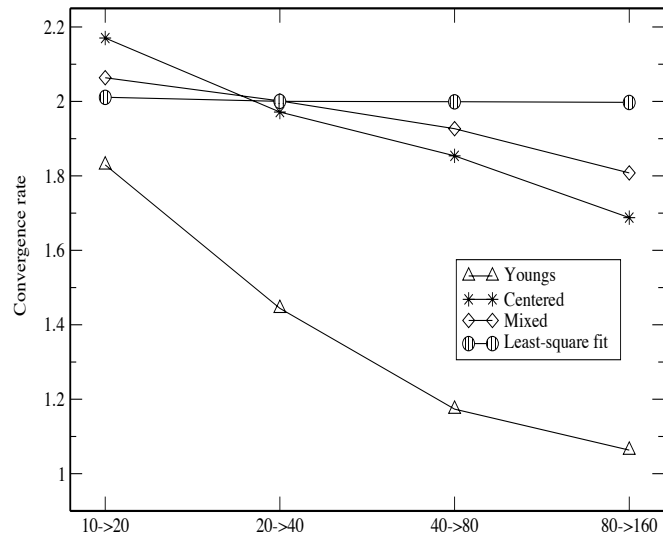


Figure 6: The convergence rates for the reconstruction of a sphere obtained with several reconstruction algorithms

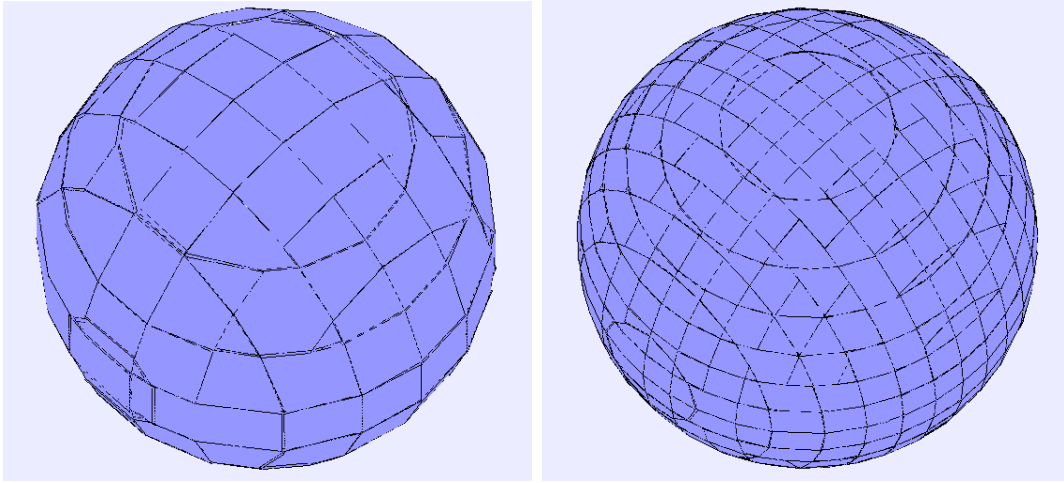


Figure 7: The reconstruction of a sphere in a grid with n^3 cells, $n = 10$ (left) and $n = 20$ (right)

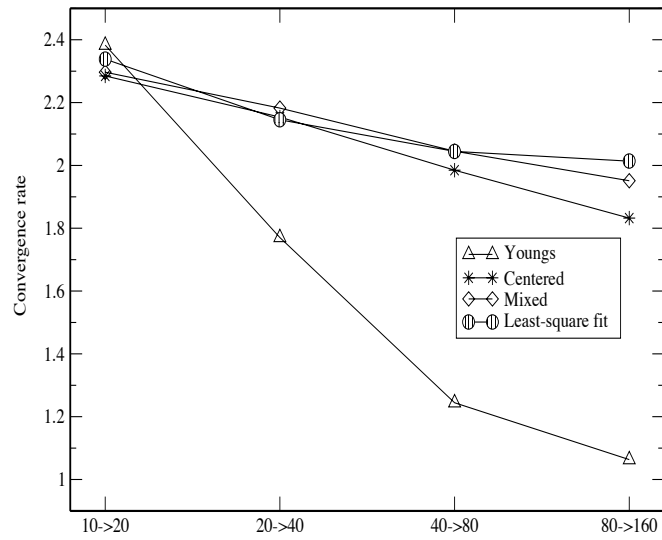


Figure 8: The convergence rates for the reconstruction of a sinusoidal surface obtained with several reconstruction algorithms

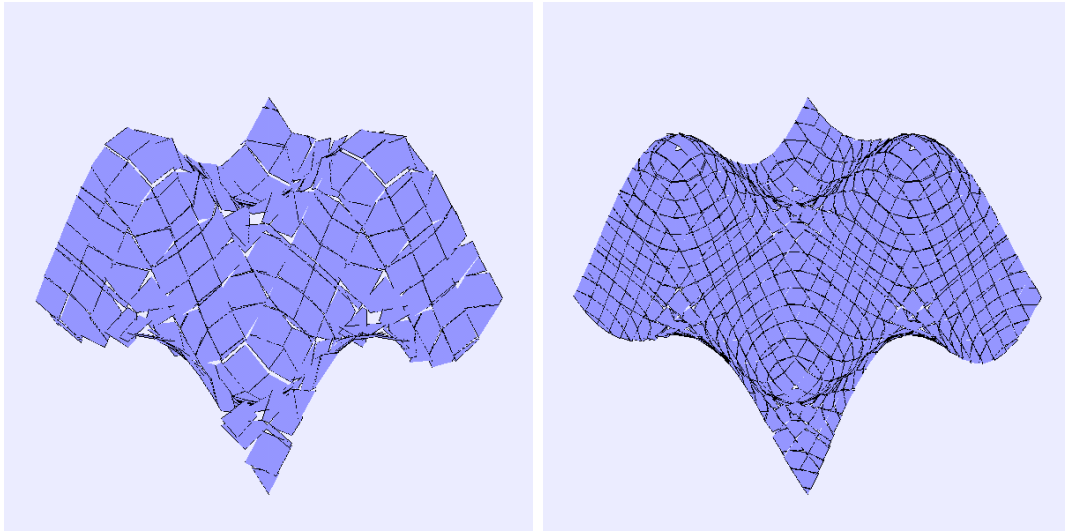


Figure 9: The reconstruction of a sinusoidal surface in a grid with n^3 cells, $n = 10$ (left) and $n = 20$ (right)

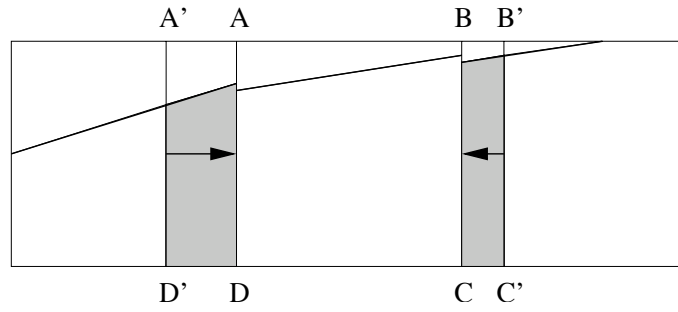


Figure 10: VOF/PLIC reconstruction in three consecutive cells and geometrical interpretation of the Eulerian-Implicit (EI) scheme. The rectangle $A'B'C'D'$, that includes the central cell and the fluxing areas of the two adjacent cells, is mapped into $ABCD$

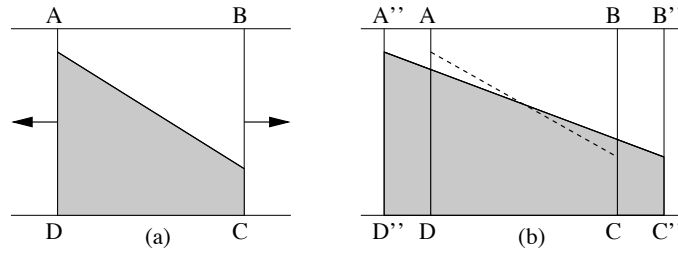


Figure 11: VOF/PLIC reconstruction in a cell (a) and geometrical interpretation of the Lagrangian-Explicit (LE) scheme (b). The cell $ABCD$ is mapped into the rectangle $A''B''C''D''$, that includes both the central cell and the fluxed areas into the two adjacent cells

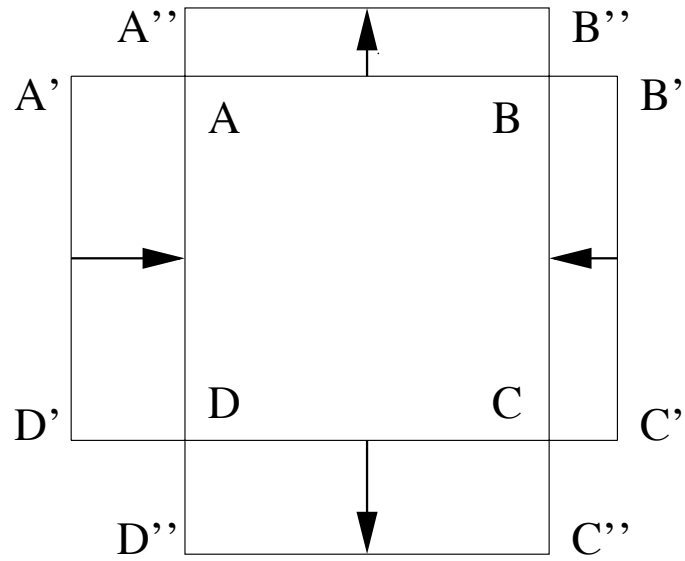


Figure 12: In the EI-LE scheme the rectangle $A'B'C'D'$ of a tessellation of the plane is mapped into $A''B''C''D''$ of a different tessellation. The two rectangles have the same area, therefore the scheme is area-preserving

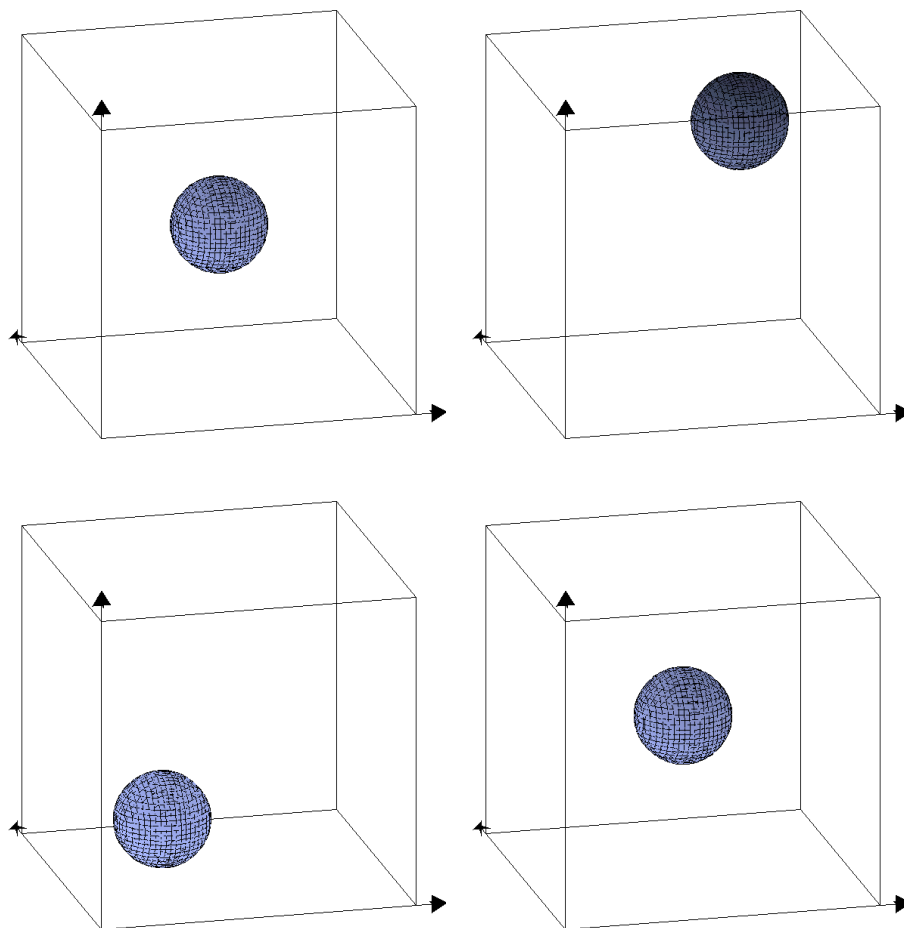


Figure 13: Translation of a sphere along the main diagonal. Initial configuration and after $(2n + 1)/3$, $(4n - 1)/3$, $2n$ steps, with $n = 64$ (left to right, top to bottom)

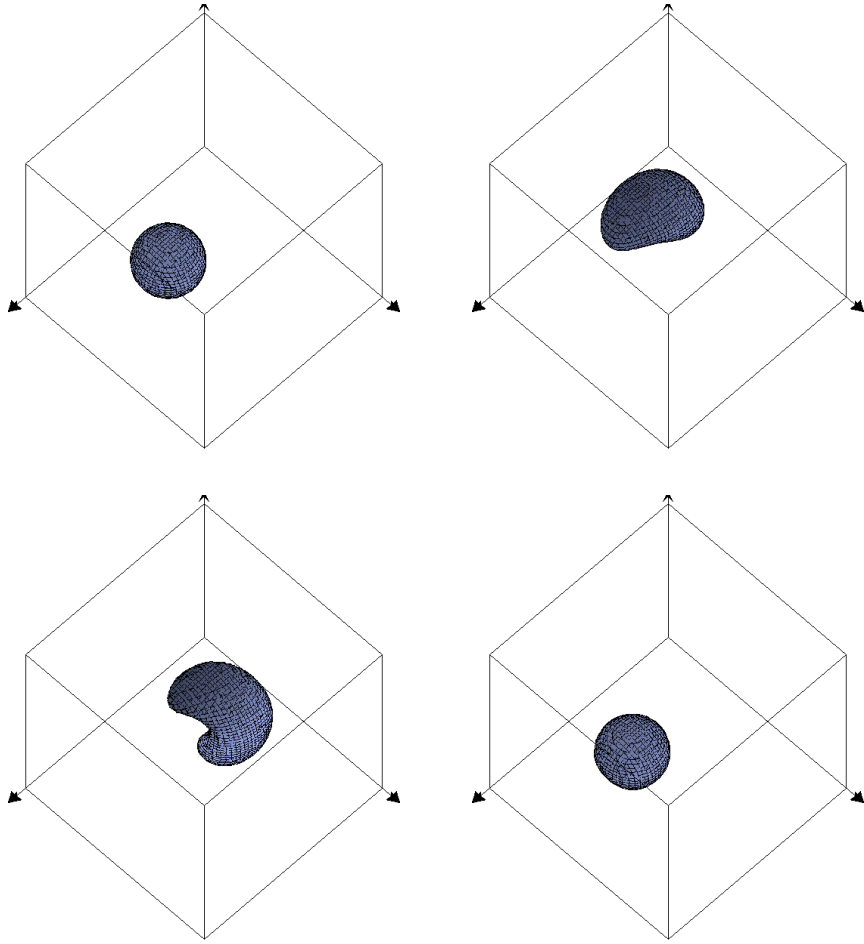


Figure 14: Advection of a sphere in a deforming velocity field. Initial configuration and after $n/2, n, 2n$ steps, with $n = 64$ (left to right, top to bottom)