INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2003; **41**:251–274 (DOI: 10.1002/fld.431)

Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection

Ruben Scardovelli[1,2,*,†] and Stephane Zaleski[3,‡]

[1] *DIENCA-Lab. di Montecuccolino, Via dei Colli 16, Bologna 40136, Italy*
[2] *INFM-BO, Viale Berti Pichat 6, Bologna 40100, Italy*
[3] *Modélisation en Mecanique, CNRS UMR 7607, Université Pierre et Marie Curie, 8 rue du Capitaine Scott, Paris 75015, France*

SUMMARY

Two new volume-of-fluid (VOF) reconstruction algorithms, which are based on a least-square fit technique, are presented. Their performance is tested for several standard shapes and is compared to a few other VOF/PLIC reconstruction techniques, showing in general a better convergence rate. The geometric nature of Lagrangian and Eulerian split advection algorithms is investigated in detail and a new mixed split Eulerian implicit–Lagrangian explicit (EI–LE) scheme is presented. This method conserves the mass to machine error, performs better than split Eulerian and Lagrangian algorithms, and it is only slightly worse than unsplit schemes. However, the combination of the interface reconstruction with the least-square fit and its advection with the EI–LE scheme appears superior to other existing approaches. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: VOF methods; interface reconstruction; Lagrangian and Eulerian advection; mass conservation

1. INTRODUCTION

The volume-of-fluid (VOF) method has become a popular and successful technique to track interfaces in both two-phase and free-surface flows. It is based on a characteristic function χ, with value 1 in one phase and 0 in the other phase or vacuum. Since the fluids are assumed immiscible and incompressible, the characteristic function does not change its value following an elementary fluid parcel, thus it is passively advected by the flow and satisfies

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

The colour function or volume fraction C is the discrete version of χ and represents the fraction of each cell of the mesh occupied by one of the two components, say fluid 1. Cells

* Correspondence to: R. Scardovelli, DIENCA-Lab. di Montecuccolino, Via dei Colli 16, Bologna 40136, Italy.
† E-mail: raus@mail.ing.unibo.it
‡ E-mail: zaleski@lmm.jussieu.fr

Received August 2001
Revised 24 June 2002

Copyright © 2003 John Wiley & Sons, Ltd.

without interfaces will have a value of C either zero or one, while cells cut by the interface will have an intermediate value of C between zero and one. At any time step in the simulation, the interface is not known: its geometry has to be inferred from the knowledge of the scalar function C. Once the interface has been reconstructed, its motion in the velocity field $\mathbf{v}$ is described by an advection equation for C. We assume that Equation (1) is satisfied by the colour function as well, and to integrate it in time we need to calculate the fluxes of C across the cells boundary. This delineates a two-step procedure which is mainly geometric in nature [1]. The reconstruction is not a unique process, given a spatial distribution of C the interface geometry depends on the assumed reconstruction algorithm. We consider a two-dimensional space, then in VOF/SLIC (for simple line interface calculation) methods [2–4] the interface in each cell is a segment parallel to one of the grid coordinate axis, while in VOF/PLIC (for piecewise linear interface calculation) techniques [1, 5–7] the interface is a segment perpendicular to the gradient of the scalar function C. The actual position of this inclined segment in the cell is uniquely determined from volume conservation. However, the overall reconstruction is not in general continuous across the boundary of adjacent cells. VOF/PLIC algorithms are more complex than piecewise constant methods, but they are more accurate and in particular do not produce isolated small fluid bodies (*flotsam*) which appear in VOF/SLIC techniques even for simple flows [3, 4, 7]. One of the methods we describe in this article reconstruct the interface with a sequence of segments, here limited to two, in each cell. This amounts to a sub-grid modelling of the interface in order to decrease the reconstruction error and to approximate the interface as a continuous chain of segments. An alternative approach would be to enhance adaptively the spatial discretization in the neighbourhood of the interface with several mesh levels. This approach has been followed by several authors in Cartesian [8] and unstructured meshes [9]. A combined approach will probably result in a more detailed and correct resolution of the interface and of the physics even for demanding situations such as the formation or break-up of filaments. The colour function C is not continuous across the interface, so a geometric evaluation of the volume fluxes across the cell boundary based on the reconstructed interface is more accurate than standard algebraic numerical schemes for advection equations [4]. More particularly, the calculation of volume fluxes can be done independently along each co-ordinate direction, with multidimensionality obtained via an operator split technique [1, 4, 5, 7, 10, 11].

This approach requires a number of interface reconstructions per time step equal to the dimensionality of the space. Alternatively, multidimensional schemes require only one reconstruction per time step, but are more complex because boundary fluxes depend on fluxes calculated along each co-ordinate direction. Examples of multidimensional algorithms can be found in References [1, 7, 12, 13].

We begin in Section 2 with a quick review of a few VOF/PLIC reconstruction algorithms and the description of two new methods based on a least-square fit technique. In Section 3 we compare the accuracy of these methods with the reconstruction of typical test shapes such as straight lines, ellipses and squares. Then, in Section 4 we review the advection equation for the function C and further investigate the geometrical aspects of two split algorithms, respectively, a Lagrangian [5] and an Eulerian one [1]. We then present a new mixed split method which conserves mass exactly. Finally, in Section 5 we compare the performance of several reconstruction schemes combined with these advection algorithms for standard tests such as the Zalesak slotted disk rotation [14] and the Rider–Kothe reversed single vortex [1].

Copyright © 2003 John Wiley & Sons, Ltd.

2. INTERFACE RECONSTRUCTION

We consider a two-dimensional (2D) rectangular computational domain with square cells. In VOF/PLIC methods the interface is approximated in each cut cell by a portion of a straight line defined by the equation

$$m_x x + m_y y = \alpha \tag{2}$$

The determination of the constants (m_x, m_y, α) of Equation (2) is basically a two-step procedure:

(1) evaluation of the interface normal $\mathbf{m} = (m_x, m_y)$;
(2) determination of the line constant α, so that the fraction of the cell area cut by this line and occupied by the reference phase is equal to C.

When both m_x, m_y are positive, the volume fraction C is a monotonically increasing, non-linear function of α [15, 16]. The determination of α, given the cut volume and the normal direction $\mathbf{m}$ in a computational cell, is then a problem with a unique solution coming from enforcement of volume conservation. It has been solved by many authors, either with a numerical root-finding technique or directly with analytical formulas describing the relation $\alpha = \alpha(C)$. On the contrary, the determination of the normal vector $\mathbf{m}$ is not a unique process. Usually, in VOF/PLIC reconstruction the normal vector is determined by the volume fraction gradient $\mathbf{m} = \nabla C$ and extensive reviews of several methods have been given in References [1, 7]. For completeness and clarity of the presentation, we first quickly describe three of these methods and then discuss with more details our new approach based on a least-square fit technique.

2.1. Parker and Youngs' method

This is our own implementation (CIAM) [5, 15] of what is usually known as the Parker and Youngs' method [6]. We consider the 3×3 block of square cells shown in Figure 1, with $\Delta x = \Delta y = h$. The normal $\mathbf{m}$ is first estimated at the four corners of the central cell (i, j) with a finite difference formula, for example the x-component m_x at the top-right corner is given by

$$m_{x,i+1/2,j+1/2} = \frac{1}{2h}(C_{i+1,j+1} + C_{i+1,j} - C_{i,j+1} - C_{i,j}) \tag{3}$$

Similarly for the y-component m_y and in the other three corners. Then the required cell-centred vector is obtained by averaging the four cell-corner values

$$\mathbf{m}_{i,j} = \tfrac{1}{4}(\mathbf{m}_{i+1/2,j+1/2} + \mathbf{m}_{i+1/2,j-1/2} + \mathbf{m}_{i-1/2,j+1/2} + \mathbf{m}_{i-1/2,j-1/2})$$

2.2. Centred Columns scheme

We consider again the 3×3 block of Figure 1 and sum the volume fractions along the vertical direction. The value obtained in each column can be considered as the height y of the function $y = f(x)$, with $y_i = f(x_i) = \sum_{k=-1}^{1} C_{i,j+k}$, and the abscissa x_i at the centre of the cell. If we approximate the interface as a straight line $y = m_{xc} x + \alpha$, then the slope m_{xc} of this line is

Copyright © 2003 John Wiley & Sons, Ltd.

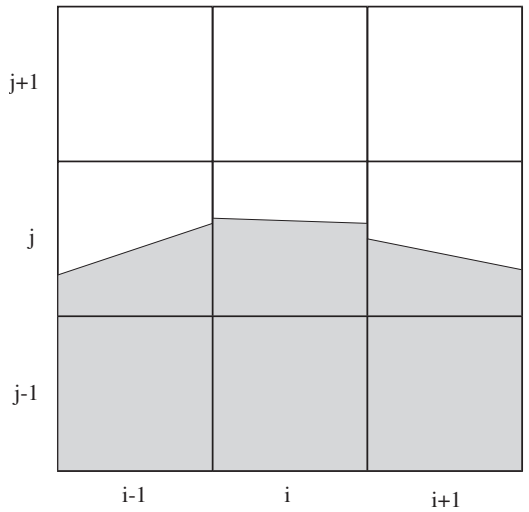

Figure 1. The 3×3 block of cells used in the calculation of the normal vector **m** with a typical VOF/PLIC reconstruction.

given by

$$m_{xc} = \frac{1}{2h}(y_{i+1} - y_{i-1}) = \frac{1}{2h} \sum_{k=-1}^{1} (C_{i+1,j+k} - C_{i-1,j+k})$$

We can also sum the volume fractions along the horizontal direction, and consider an approximating straight line of the type $x = m_{yc}\, y + \alpha$, with

$$m_{yc} = \frac{1}{2h} \sum_{k=-1}^{1} (C_{i+k,j+1} - C_{i+k,j-1})$$

As discussed in Reference [7], a proper way to choose between m_{xc} and m_{yc} is to take the smaller one in absolute value

$$m = \min(\mathrm{abs}(m_{xc}), \mathrm{abs}(m_{yc}))$$

2.3. ELVIRA

The name stands for efficient least-squares volume-of-fluid interface reconstruction algorithm and is due to Pilliod [17]. With respect to the previous case, now we consider also the backward and forward schemes, respectively,

$$m_{xb} = \frac{1}{h} \sum_{k=-1}^{1} (C_{i,j+k} - C_{i-1,j+k})$$

$$m_{xf} = \frac{1}{h} \sum_{k=-1}^{1} (C_{i+1,j+k} - C_{i,j+k})$$

Copyright © 2003 John Wiley & Sons, Ltd.

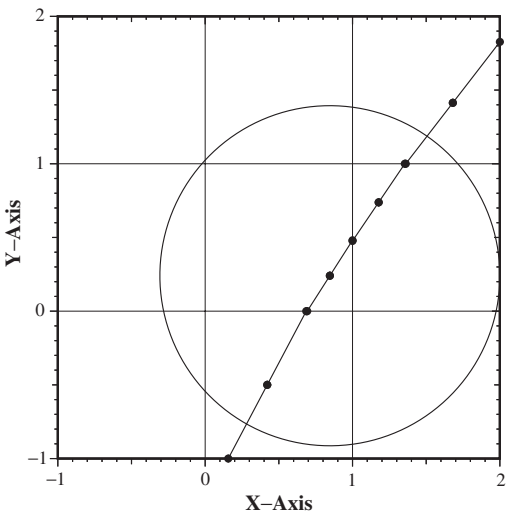

Figure 2. A linear approximation to the interface in a 3×3 block, with the end and middle points of each segments. Only those inside the 'radius of influence' will enter the least-square fit. The segments at the cells boundary do not connect with each other, so there are actually two very close points on the boundary lines.

and analogously for m_{yb} and m_{yf}. Then we have to consider six different cases and for each of them we determine a line constant α to have the correct cut volume in the central cell (i, j). This straight line is then drawn across the 3×3 block defining a different cut volume $\tilde{C}$ in each of the surrounding 8 cells. We then consider the discrete error E in L_2 between the true value C and the approximated value $\tilde{C}$ given by the expression

$$E(\tilde{m}) = \left(\sum_{k,l=-1}^{1} (\tilde{C}_{i+k,j+l}(\tilde{m}) - C_{i+k,j+l})^2 \right)^{1/2} \tag{4}$$

where $\tilde{m}$ is one of the six previously defined angular coefficients. The value of $\tilde{m}$ leading to the smallest error E is the winner. This method reconstructs a linear interface correctly, whatever value of the angular coefficient.

2.4. *Least-square fits*

The first step in the procedure is to gather a convenient set of points inside the 3×3 block of Figure 2. To accomplish this, we need to start from a linear reconstruction of the interface. To this aim the centred columns scheme is our choice, because it is the computationally less expensive of the previous three methods and it has an intermediate convergence rate between CIAM and ELVIRA, as it will be shown in the results section. For each segment of the reconstruction we consider both its end points and the middle one. The reason of this choice is that a good reconstruction intersects once or more the real interface, because of the constant volume constraint, so that there are points on both sides of the plane cut by the interface. For a good and efficient fit we need to consider points on both sides, and this is the simplest

Copyright © 2003 John Wiley & Sons, Ltd.

scheme we have devised to this aim. Given all points inside the 3×3 block we need to select a number of them to enter the least-squares procedure. A simple idea we have implemented is that of a 'radius of influence'. From the midpoint of the segment in the central cell we determine the minimum distance to the boundary of the block. All points inside this circle will enter the least-square fit, as shown in Figure 2. This simple idea makes it easy to extend the procedure to rectangular grids with slowly varying grid size and to unstructured grids. We also notice that when the number of cut cells is greater than a prescribed value (say 5) the number of points increases, but usually this also means a highly convoluted interface (a radius of curvature smaller than the grid size), or the possibility of more than one interface in the block, like for example in the case of two colliding droplets. In this case, we reduce artificially the radius of influence by multiplying it by a number smaller than one.

2.4.1. Linear least-square fit. We present here two different fits. In the first one the line in the central cell is approximated as a portion of a straight line. To this aim we minimize the functional H_1

$$H_1 = \sum_i (y_i + mx_i + \alpha)^2 \tag{5}$$

more precisely we also check if the line should be written as $y+mx+\alpha=0$ or as $x+my+\alpha=0$. In the previous expression each couple (x_i, y_i) represents the co-ordinates of the generic point inside the circle. We solve the resulting 2×2 linear system only for the angular coefficient m, so basically this procedure is another recipe to find the normal vector $\mathbf{m}$. Then the line constant α is determined in the way previously described.

2.4.2. Quadratic least-square fit. In the second approach, we approximate the interface as a portion of a circle. It is simple to extend the procedure to a portion of a parabola. This can be considered a higher-order correction, in the sense that the results depend heavily on the linear approximation used to determine the points inside the circle of influence. So we use this procedure on top of a second-order accurate algorithm, either ELVIRA or the linear least-square fit. The points entering the fit are determined in the same way, but now we minimize the functional H_2

$$H_2 = \sum_i (x_i^2 + y_i^2 + ax_i + by_i + c)^2 \tag{6}$$

and solve the resulting 3×3 linear system for the coefficients a, b, c. The constraint of a given cut volume is not in general satisfied, therefore there is the need to move the arc of circle within the central cell with some 'ad hoc' rule. We have chosen another solution, which makes the idea more prone to be implemented in the three-dimensional (3D) case. We first calculate the intersections of the circle with the boundary of the central cell and then move the central point of the segment connecting these two points in the perpendicular direction, to get the exact volume. In this way, we approximate the curved line with two consecutive segments. A schematic picture of this procedure is shown in Figure 3. Simple extensions to three or more segments can be devised, but then the calculation of the fluxes across the boundary becomes more involved and lengthy. If the procedure does not succeed, for example if there are more than 2 intersections or if the central point moves outside the cell, we slowly move the intersections towards those of the linear approximation until the solution is satisfactory.

Copyright © 2003 John Wiley & Sons, Ltd.

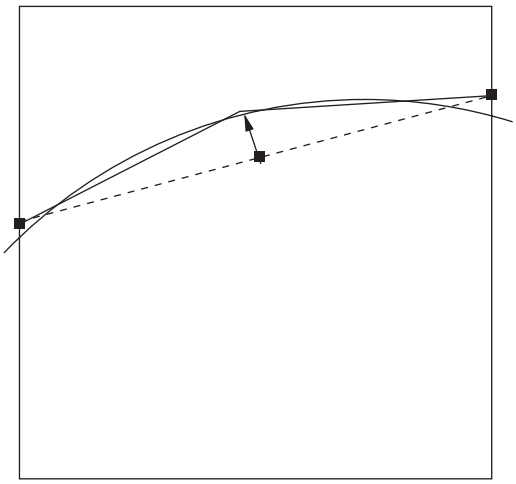

Figure 3. A schematic picture of the three-point reconstruction. The thick solid line is the original line, the dashed line is the segment connecting the two intersections of the reconstructed circle with the boundary of the cell, while the thin solid line is the final reconstruction made up of two consecutive segments.

As a byproduct of this procedure, we have also a local approximation of the curvature κ of the line. We have performed some simple tests by checking the curvature of circles with a radius whose length in terms of number of cells ranges from 8 to 20. The maximum error in the curvature is oscillating between 1 and 3%, without showing a monotonic behaviour, in line with the results of [18].

3. RECONSTRUCTION ACCURACY TESTS

We now assess the accuracy of the reconstruction schemes previously described with their application to a few configurations: a straight line, an ellipse and a square. In all cases, we consider the E_1 norm error, between the real line $l_r(x, y)$ and the approximation $\tilde{l}(x, y)$, defined as

$$E_1 = \int\int |\, l_r(x, y) - \tilde{l}(x, y) \,|\, \mathrm{d}x\, \mathrm{d}y \tag{7}$$

The first check is the reconstruction of a straight line. In the central cell of the computational domain we randomly position a point of the line and determine the angle with the horizontal coordinate line. We average the results over 1000 different cases in order to stabilize the error. This is necessary because all methods, for some subset of the set of all possible lines, are able to reconstruct a straight line exactly. The results are presented in Table I. The ELVIRA scheme reconstructs exactly a straight line, whatever its orientation, while the other schemes only in some case.

The somewhat non-constant behaviour of the order of convergence is due to the fact that as we double the resolution portions of the line which were previously reconstructed exactly

Copyright © 2003 John Wiley & Sons, Ltd.

Table I. The E_1 error in approximating a straight line.

	CIAM	Central	ELVIRA	Linear fit
Grid	Error/order	Error/order	Error/order	Error/order
10^2	8.21e − 4	3.29e − 5	< 1.e − 17	1.78e − 6
	0.97	0.45	****	1.21
20^2	4.18e − 4	2.40e − 5	< 1.e − 17	7.67e − 7
	1.01	0.65	****	0.81
40^2	2.07e − 4	1.52e − 5	< 1.e − 17	4.36e − 7
	1.00	0.81	****	0.85
80^2	1.03e − 4	8.62e − 6	< 1.e − 17	2.42e − 7
	1.00	1.05	****	1.08
160^2	5.16e − 5	4.15e − 6	< 1.e − 17	1.14e − 7
	1.01	0.96	****	0.96
320^2	2.56e − 5	2.13e − 6	< 1.e − 17	5.88e − 8

can now be not and viceversa, and this happen in a rather unpredictable way. However, the results show that the error with the linear fit approximation is a few orders of magnitude less than the error with the CIAM and Centred Columns (or Central) schemes. We can also iterate the linear fit scheme by using the previous approximation, and this approach converges, when the solution is not exact from the beginning, with a number of iterations between 2 and 4. However, in the following of this section, we will show that for practical cases, there is no need to iterate the linear least-square fit.

The second line is an ellipse described by the equation: $x^2/a^2 + y^2/b^2 = 1$, with $a^2 = 0.12$ and $b^2 = 0.02$, with a ratio of 14.697 between the maximum and minimum radius of curvature. We have chosen an ellipse because its curvature is a smooth, continuously varying function. However, we anticipate that the convergence results in the case of this ellipse and of a circle with radius $r = a$ are almost the same. The third line is a square with side $c = 0.512$ and its reconstruction mixes the ability of the methods to reproduce a straight line and a singularity such as a corner. These two test shapes are shown in Figure 4 together with the coarsest mesh. For both tests we have randomly positioned the centre of the test shapes in a cell near the central point of the computational domain, as well as the angle between one axis of symmetry of the figure and the horizontal co-ordinate line. We average the results over several tens of reconstructions to remove fluctuations due for example to particular alignments with the mesh. The coarsest grid has 10^2 cells, as shown in Figure 4, the most refined one has 320^2 cells for the square and the straight lines, and 640^2 for the ellipse, to analyse the convergence rate at very high resolution in the case of a smooth curve. The error is calculated analytically, by using the equation of the real line and of the approximating straight lines, and by considering consecutive intervals in each cell defined by the end points and the intersections of the lines. We have also carried out a numerical integration, and we usually need a few tens of equally spaced points in each cell to reach a convergence up to the fourth digit, which is what we use to calculate the converge rate of each method. In Table II, we present the results for the ellipse. They are in agreement with those presented in References [1, 7]. It is apparent from the data that CIAM is asymptotically a first-order method, but at low resolution, or equivalently when the radius of curvature is not much bigger than the grid size, its performance is comparable,

Copyright © 2003 John Wiley & Sons, Ltd.

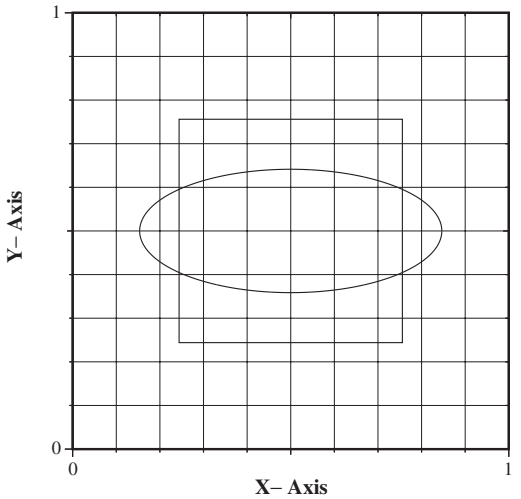

Figure 4. The ellipse and the square considered in our reconstruction tests together with the coarsest mesh.

Table II. The E_1 error in approximating an ellipse.

	CIAM	Central	ELVIRA	Linear fit	Quadratic fit	Continuity
Grid	Error/order	Error/order	Error/order	Error/order	Error/order	Error/order
10^2	3.90e − 3	4.25e − 3	4.76e − 3	3.30e − 3	2.32e − 3	1.79e − 3
	2.34	2.47	2.42	2.41	2.88	2.87
20^2	7.68e − 4	7.65e − 4	8.87e − 4	6.23e − 4	3.16e − 4	2.45e − 4
	1.84	2.27	2.33	2.15	2.63	2.58
40^2	2.14e − 4	1.59e − 4	1.77e − 4	1.41e − 4	5.10e − 5	4.09e − 5
	1.38	2.02	2.10	2.07	2.34	2.17
80^2	8.23e − 5	3.93e − 5	4.12e − 5	3.41e − 5	1.00e − 5	9.06e − 6
	1.16	1.89	2.04	2.02	2.14	2.03
160^2	3.69e − 5	1.06e − 5	1.00e − 5	8.36e − 6	2.27e − 6	2.22e − 6
	1.06	1.73	2.00	2.01	2.03	2.01
320^2	1.77e − 5	3.20e − 6	2.49e − 6	2.07e − 6	5.54e − 7	5.50e − 7
	1.03	1.57	1.99	1.97	2.00	1.99
640^2	8.66e − 6	1.08e − 6	6.26e − 7	5.28e − 7	1.38e − 7	1.38e − 7

if not better, than the Central scheme and ELVIRA. This last one is truly second-order, while the Central scheme exhibits an intermediate behaviour. Our linear least-square fit is almost second-order even at very high resolutions where it remains still slightly better than ELVIRA, moreover at low resolutions it outperforms all previous methods. The quadratic fit produces a nearly third-order converge rate at very low resolutions, this is because a portion of a line with high curvature is clearly better approximated by two consecutive segments than by a straight line. It also exhibits the behaviour of a higher-order correction, in the sense that if

Copyright © 2003 John Wiley & Sons, Ltd.

Table III. The average discontinuity at the cell boundary for the ellipse.

	CIAM	Central	ELVIRA	Linear fit	Quadratic fit	Continuity
Grid	Δ/order	Δ/order	Δ/order	Δ/order	Δ/order	Δ/order
10^2	9.00e − 3	9.41e − 3	9.39e − 3	7.07e − 3	4.75e − 3	1.91e − 3
	2.28	2.65	2.54	2.52	2.84	3.31
20^2	1.85e − 3	1.50e − 3	1.62e − 3	1.23e − 3	6.62e − 4	1.92e − 4
	1.64	2.47	2.51	2.22	2.43	2.75
40^2	5.93e − 4	2.71e − 4	2.84e − 4	2.64e − 4	1.23e − 4	2.85e − 5
	1.16	2.06	2.31	2.05	2.33	2.12
80^2	2.65e − 4	6.50e − 5	5.74e − 5	6.38e − 5	2.44e − 5	6.53e − 6
	1.02	1.67	2.19	2.08	2.24	2.08
160^2	1.31e − 4	2.04e − 5	1.26e − 5	1.51e − 5	5.15e − 6	1.54e − 6
	1.00	1.31	2.08	2.00	2.04	1.91
320^2	6.55e − 5	8.25e − 6	2.97e − 6	3.78e − 6	1.25e − 6	4.09e − 7
	1.00	1.14	2.02	2.01	2.04	2.05
640^2	3.27e − 5	3.73e − 3	7.30e − 7	9.38e − 7	3.03e − 7	9.89e − 8

we apply this scheme on top of the CIAM reconstruction the method is not able to bring to second-order the convergence rate, so in this respect it relies heavily on the fact that the initial linear approximation is a good one. The reconstruction is in general not continuous at the cell boundary since the only constraint between the original line and the approximation is volume conservation in each cut cell. The interface is then approximated in two dimensions by a sequence of segments with discontinuities at the cell boundary that approach zero as the resolution is increased, in particular we already pointed out that ELVIRA reconstructs a straight line exactly. The three-points reconstruction gives us an extra degree of freedom that can be exploited to decrease the discontinuity level at the cell boundary. We have implemented a simple scheme that brings the points on the boundary towards each other. The procedure is not always successful, this may happen when in the process the sign of the local curvature of the line is changed, fact that we do not allow, or when the initial three-points reconstruction was actually a straight line. The error after this procedure is given in the last column of Table II. We see that the overall reconstruction benefits from this scheme at low resolution or equivalently when the radius of curvature is comparable with the grid size. At very high resolution there is no improvement in the results. Moreover, it is of some interest to check how the discontinuity Δ at the cell boundary tends to zero on average as the resolution is increased. From Table III, it is clear that the order of convergence of the method in the case of a smooth line is more or less that of the discontinuity. In Table IV we present the results for the square. Since most methods reconstruct a straight line correctly for some particular inclination of the line with respect to the mesh, we need more cases to extract the correct convergence rate. The error near the singularity plays a major role both at low and high grid resolution. In the first case, its contribution to the total error is significant for all schemes, so that the difference among them is somewhat smeared out. However, for the second-order schemes, the error is basically coming from the region near the singularity which scales as Δx^2, so that they are still second-order. The differences among them are mainly due to the different accuracy in the corner reconstruction. With CIAM, which is asymptotically a first-order scheme, as the resolution is increased the contribution from the error away from the

Copyright © 2003 John Wiley & Sons, Ltd.

Table IV. The E_1 error in approximating a square.

	CIAM	Central	ELVIRA	Linear fit	Quadratic fit	Continuity
Grid	Error/order	Error/order	Error/order	Error/order	Error/order	Error/order
10^2	8.97e − 3	9.40e − 3	9.14e − 3	7.77e − 3	7.13e − 3	5.61e − 3
	1.89	1.99	1.98	2.01	2.01	2.00
20^2	2.42e − 3	2.36e − 3	2.32e − 3	1.93e − 3	1.77e − 3	1.40e − 3
	1.83	2.01	2.03	2.00	2.00	2.02
40^2	6.80e − 4	5.84e − 4	5.67e − 4	4.81e − 4	4.42e − 4	3.45e − 4
	1.65	1.94	2.00	1.95	1.95	1.96
80^2	2.16e − 4	1.52e − 4	1.42e − 4	1.24e − 4	1.14e − 4	8.86e − 5
	1.54	2.00	2.02	2.04	2.02	2.01
160^2	7.40e − 5	3.78e − 5	3.51e − 5	3.02e − 5	2.81e − 5	2.20e − 5
	1.35	1.92	1.99	2.00	2.02	2.00
320^2	2.91e − 5	1.00e − 5	8.83e − 6	7.69e − 6	6.92e − 6	5.50e − 6

singularity becomes more and more important, and the method converges towards first-order, but at a slower rate, compared to the ellipse case, because of the important contribution to the error coming from the singularity. We conclude this section with a few words on the 3D extension of these techniques and on timing. We have implemented a 3D version of the linear least-square fit. Preliminary results [19] are similar to the 2D case. A second-order convergence rate is maintained for sphere and trigonometric surfaces up to at least a 160^3 grid, while the geometric error for planes reconstruction is still a few orders of magnitude better with the linear fit than with the 3D version of CIAM and of the Central scheme. In terms of CPU time, CIAM and the Central scheme are comparable, ELVIRA and the linear fit are slightly more time consuming, the quadratic fit and the one with added continuity, are, respectively, 1.5 and 2 times more expensive.

4. INTERFACE ADVECTION

Once the interface has been reconstructed, its motion by the underlying velocity field must be modelled by a suitable advection algorithm. Since each fluid component of the flow conserves its identity, the volume fraction function C is passively advected by the flow and we assume it satisfies a standard advection equation

$$\frac{\mathrm{d}C}{\mathrm{d}t} = \frac{\partial C}{\partial t} + \mathbf{v} \cdot \nabla C = 0 \tag{8}$$

Moreover, if the flow is incompressible, $\nabla \cdot \mathbf{v} = \partial u/\partial x + \partial v/\partial y = 0$, the previous equation can be recast in a conservative form

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{v}C) = 0 \tag{9}$$

Equation (9) states the fact that the conservation of mass for an incompressible flow is equivalent to the conservation of the volume of each single phase. VOF advection techniques are mainly divided in two categories: multidimensional or unsplit schemes and one-dimensional

Copyright © 2003 John Wiley & Sons, Ltd.

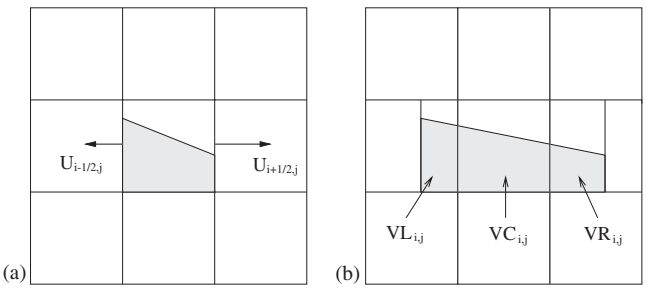

Figure 5. Lagrangian advection of the interface along the x-direction: (a) interface reconstruction in the central cell before advection; (b) contributions along the x-direction of the central cell after advection.

or operator split schemes. In this paper, we limit our presentation to operator split schemes, where boundary fluxes at every time step are calculated independently and consecutively along each co-ordinate direction so that the area fluxing each cell boundary is a simple rectangle, if all points on a side of the square cell move with the same velocity. If this is not the case, the area fluxing through the cell edge is a more complex geometrical figure, for example a trapezoid if a linear velocity profile is assumed. We first review a Lagrangian and an Eulerian scheme and then describe a new mixed Lagrangian–Eulerian method.

4.1. Lagrangian scheme

This method was originally developed by Li [5] and its 3D version was discussed in detail in Reference [15]. The interface advection is performed separately along each spatial direction via operator splitting. The basic features of the algorithm are depicted in Figure 5: the horizontal velocity is assumed constant on the two vertical sides of the central cell that are advanced with a simple forward scheme. After advection, if the one-dimensional (1D) expansion term $\partial u/\partial x$ is different from zero, the interface segment has changed not only its position, but also its length and orientation, in particular there are two new constants (m_x, α) in Equation (2) for a motion along the x-direction, while m_y remains unchanged [15]. However, it is straightforward [16] to calculate the fluxes to the left and to the right, VL and VR, respectively, and what it is left inside, VC. The updated value of C is then given by

$$C_{i,j} = VL_{i+1,j} + VC_{i,j} + VR_{i-1,j} \tag{10}$$

With this formulation, we calculate the fluxes across the vertical boundaries of the cell and at the same time we take into account the contribution of the 1D expansion term $\partial u/\partial x$, which in general is not zero, even if $\nabla \cdot \mathbf{v} = 0$. The interface is then advected in the y-direction in a similar way. The starting sweep direction is alternated every time step.

4.2. Eulerian scheme

We consider the Eulerian scheme introduced by Rider and Kothe [1] where a 'divergence correction' term $C\nabla \cdot \mathbf{v}$ is added to both sides of Equation (8) to give

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{v}C) = C\nabla \cdot \mathbf{v} \tag{11}$$

Copyright © 2003 John Wiley & Sons, Ltd.

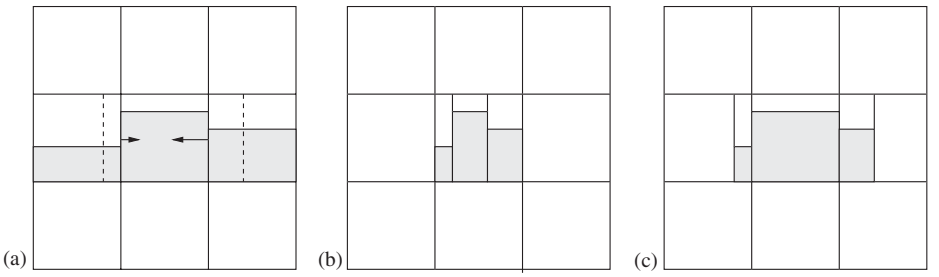

Figure 6. Eulerian advection of the interface along the x-direction: (a) SLIC reconstruction of the interface; (b) explicit scheme; (c) implicit scheme.

If we denote with F the net horizontal flux through a vertical side of the cell and with G the net vertical flux through a horizontal side, the integration from time n to time $n+1$ of the previous equation with an operator split scheme is a two-steps procedure given by

$$C^*_{i,j} = C^n_{i,j} + (F^n_{i-1/2,j} - F^n_{i+1/2,j}) + C^n_{i,j}(u_{i+1/2,j} - u_{i-1/2,j}) \tag{12}$$

for the advection in the x-direction, and

$$C^{n+1}_{i,j} = C^*_{i,j} + (G^*_{i,j-1/2} - G^*_{i,j+1/2}) + C^{n+1}_{i,j}(v_{i,j+1/2} - v_{i,j-1/2}) \tag{13}$$

for the advection in the y-direction. In these two expressions, the superscript * represents an intermediate value for volume fractions and fluxes and the velocity $\mathbf{v}=(u,v)$ is actually the CFL number, i.e. $u = u\,\Delta t/h$ and similarly for v. The generalization to rectangular cells is straightforward. As discussed by the authors [1], key points of this scheme are: (a) the divergence correction term, which appears in the RHS of (12) and (13), contains the volume fraction C at different time levels, in particular the term is explicit (time level n) in the first equation and implicit (time level $n+1$) in the second sweep. As a result, the net divergence correction employs a volume fraction at an intermediate time level; (b) the starting sweep direction is alternated every time step; (c) in 2D cartesian geometry for an incompressible flow, the direct use of the identity $\partial u/\partial x = -\partial v/\partial y$ may improve the discrete mass conservation properties of the operator-split time integration scheme.

The basic features of the explicit and implicit schemes are depicted in Figure 6 [20]. For simplicity and clarity, the initial reconstruction has been done with a VOF/SLIC method, with the interface in each cell approximated by a segment aligned with one of the two co-ordinate axis, as shown in Figure 6(a). With the previous normalization of the velocity field, every grid cell has been mapped to the square with side equal to one, while the horizontal velocity u represents also the rectangular area crossing the vertical edges of the central cell. For example in Figure 6(a), $u_{i-1/2,j}$ is the rectangular area comprised within the left boundary of the central cell and the vertical dashed line on the left, while $F^n_{i-1/2,j}$ is the portion of this area occupied by fluid 1. We can now rewrite (12) in the following way

$$C^*_{i,j} = C^n_{i,j}(1 + u_{i+1/2,j} - u_{i-1/2,j}) + (F^n_{i-1/2,j} - F^n_{i+1/2,j}) \tag{14}$$

It is then evident that $C^*_{i,j}$, as shown in Figure 6(b), represents the volume fraction of the central cell and it is the sum of three contributions: the term $C^n_{i,j}(1 + u_{i+1/2,j} - u_{i-1/2,j})$ which

Copyright © 2003 John Wiley & Sons, Ltd.

represents the area of fluid 1 left inside from the initial configuration after the effects of the compression term $\partial u/\partial x$ have been accounted for (in other terms this contribution represents the actual area occupied by fluid 1 of cell (i,j) after an homogeneous compression or expansion along the x-direction), the flux $F^n_{i-1/2,j}$ coming from the left cell and $F^n_{i+1/2,j}$ from the right one. On the other hand, the implicit scheme of (13) can be reformulated for the advection in the x-direction as

$$C^*_{i,j} = \frac{C^n_{i,j} + F^n_{i-1/2,j} - F^n_{i+1/2,j}}{1 - u_{i+1/2,j} + u_{i-1/2,j}} \tag{15}$$

In this case, the final volume fraction $C^*_{i,j}$ is given by the sum of the initial area occupied by fluid 1 in the reference cell and the two flux contributions from the lateral cells, divided by the total area $1 - u_{i+1/2,j} + u_{i-1/2,j}$ of the composite rectangle, as shown in Figure 6(c).

4.3. Mixed scheme

First we notice that if we apply the Lagrangian scheme, as depicted in Figure 5, to the SLIC reconstruction shown in Figure 6(a), then it is evident that the Lagrangian scheme and the Eulerian explicit one are identical. However, this is just the result of the particular reconstruction of the interface with segments either vertical or horizontal. If a PLIC reconstruction is considered, then the interface segments are in general not parallel to a cartesian axis. After advection, the three rectangles involved in the calculation of the updated value of the volume fraction are always those of Figure 6(b): they fill in exactly the central cell. Moreover, the central contribution is the same, because both the Lagrangian and the Eulerian explicit schemes take into account the compression/expansion of the central cell (this is the first term on the r.h.s. of (14)). However, they differ in the way the fluxes across the cell boundary are calculated. In the Eulerian explicit scheme the fluxes are calculated before advection, while in the Lagrangian scheme the interface is first advected and then the fluxes are determined. As a result, in the latter the 1D compression/expansion of the two lateral cells is more correctly taken into account. The rectangular areas involved in the procedure are the same, thus preserving the volume as a whole, but the two fluxes of fluid 1 are different, because after a Lagrangian propagation the orientation of the segment in general changes. Therefore, the final value of $C^*_{i,j}$ will not be the same for the two schemes. In the next section, we will show that the Lagrangian advection of the interface in one direction combined with an Eulerian implicit step in the other one is optimal in terms of performance. Note that we have given a geometrical interpretation of the two schemes, but we also need to decide in which order to perform them. The two indices usually considered to estimate quantitatively the validity of a method are the conservation of mass and the error in some norm between an initial and a final configuration. We now show why one sequence is better than the other one in terms of mass conservation, while it will be shown in the next section that they are similar from the point of view of the final geometric error. Let us consider the 3×3 block of cells of Figure 7 with the following volume fraction distribution [21]:

$$C^n_{i,j} = \begin{cases} 1 & \text{for } i = j = 1 \\ 0 & \text{elsewhere} \end{cases}$$

We simplify the problem by considering a SLIC reconstruction and by performing first the compression along the x-direction. For the central cell, the kinematic condition $\nabla \cdot \mathbf{v} = 0$

Copyright © 2003 John Wiley & Sons, Ltd.

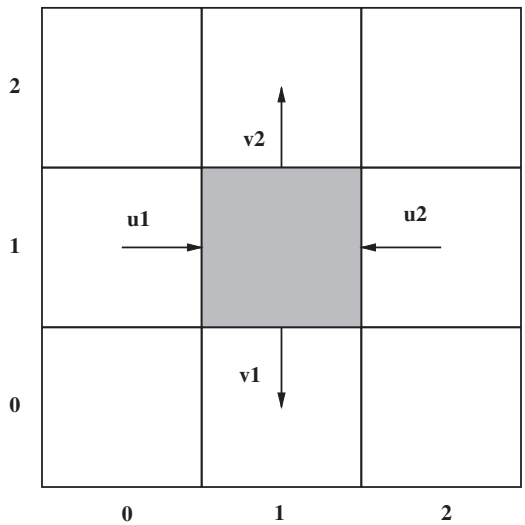

Figure 7. A 3×3 block of empty cells, but the central one which has unitary volume fraction.

implies that $\Delta u_{1,1} = u_2 - u_1 = -\Delta v_{1,1} = v_1 - v_2$, where we have set $h = 1$ for simplicity. We consider first the combination Eulerian Implicit–Lagrangian Explicit (EI–LE), and after the 1D x-compression we have

$$C^*_{1,1} = \frac{1}{1 - u_2 + u_1} = \frac{1}{1 - \Delta u_{1,1}} < 1$$

all other $C^*_{i,j}$ being zero. We consider a SLIC reconstruction with a vertical interface, then the Lagrangian step in the y-direction gives

$$C^{n+1}_{1,1} = C^*_{1,1}, \quad C^{n+1}_{1,0} = -C^*_{1,1} v_1, \quad C^{n+1}_{1,2} = C^*_{1,1} v_2$$

all other $C^{n+1}_{i,j}$ are zero. Since $h = 1$, volume fractions and actual volumes coincide, then by summing over all C's we get

$$\sum_{i,j} C^{n+1}_{i,j} = C^{n+1}_{1,1} + C^{n+1}_{1,0} + C^{n+1}_{1,2} = C^*_{1,1}(1 - v_1 + v_2) = \frac{1 + \Delta v_{1,1}}{1 - \Delta u_{1,1}} = 1$$

since $\Delta u_{1,1} = -\Delta v_{1,1}$. Next we consider the combination LE–EI. After the 1D compression in the x-direction

$$C^*_{1,1} = 1 + u_2 - u_1 = 1 + \Delta u_{1,1} < 1$$

all other $C^*_{i,j}$ are zero. After the expansion along the y-direction the terms different from zero are

$$C^{n+1}_{1,1} = \frac{C^*_{1,1} + C^*_{1,1} v_1 - C^*_{1,1} v_2}{1 - v_2 + v_1} = C^*_{1,1}, \quad C^{n+1}_{1,0} = \frac{-C^*_{1,1} v_1}{1 - \Delta v_{1,0}}, \quad C^{n+1}_{1,2} = \frac{C^*_{1,1} v_2}{1 - \Delta v_{1,2}}$$

Copyright © 2003 John Wiley & Sons, Ltd.

Clearly if $\Delta v_{1,2} = \Delta v_{1,0} = \Delta v_{1,1}$, we have

$$\sum_{i,j} C_{i,j}^{n+1} = 1$$

but in general this is not the case. We conclude that the EI–LE combination is better in terms of mass conservation and this fact will be further supported by the results in the next section.

5. ADVECTION ACCURACY TESTS

Typical advection tests involve simple translations and solid body rotations of geometrical figures such as squares and circles. They are useful for debugging the reconstruction and advection algorithms, as they test the capability of these methods to move around in the computational domain portions of straight lines, circles, corners, with the constraint given by the applied flow that these lines have to preserve their form. The results of different reconstruction algorithms (SLIC, PLIC and others) and advection schemes (either operator split or multidimensional) have been studied in great detail by several authors in the last years [1, 3, 4, 7, 12, 22]. A good algorithm should translate and rotate fluid bodies, with an interface delimited by a continuous line with continuous derivatives, without appreciable distortion or deterioration of the interface (as opposite to the fragmentation of a smooth front determined by a first-order accurate scheme, such as SLIC [3, 4, 7]), while conserving mass within machine error. Moreover, these solid body motions can test the capability of the reconstruction algorithm to approximate regions with corners or with very high curvature relative to the grid resolution. In a dynamical simulation, this situation may happen when an interface is stretched by the flow and ultimately breaks up. Capillarity effects due to surface tension quickly remove regions with infinite curvature such as corners, but they can still be present for some short time, right before and after a reconnection which changes the topology of the interface. For this reason, in the next section we present results for the Zalesak slotted disk rotation [14]. More demanding tests involve flows with a non-uniform vorticity field. Each fluid component deforms and shears as it moves through the computational domain with eventual reconnection or break-up of the interface. The single vortex test [1] is here performed to compare the performance of several reconstruction and advection algorithms. In particular, for both the solid body rotation and the single vortex test we consider only the reconstruction methods, among those described in Section 2, which show a second-order convergence rate together with all three advection schemes, i.e. the Lagrangian, the Eulerian and the mixed one.

5.1. Zalesak slotted disk rotation

This study has become a benchmark for testing the accuracy of various interface tracking methods. In the unit square with 100 grid steps along each co-ordinate direction, a circle with radius equal to 15 cells is centred in (0.5, 0.75). A vertical rectangular cut is produced with width equal to $\frac{1}{3}$ of the radius and the upper bridge, connecting the left and right portions of the circle, has the same maximum width. The velocity field $\mathbf{v} = (u, v)$ can be expressed in terms of the stream function

$$\Psi(x, y) = -\frac{\Omega}{2}((y - y_0)^2 + (x - x_0)^2) \tag{16}$$

Copyright © 2003 John Wiley & Sons, Ltd.

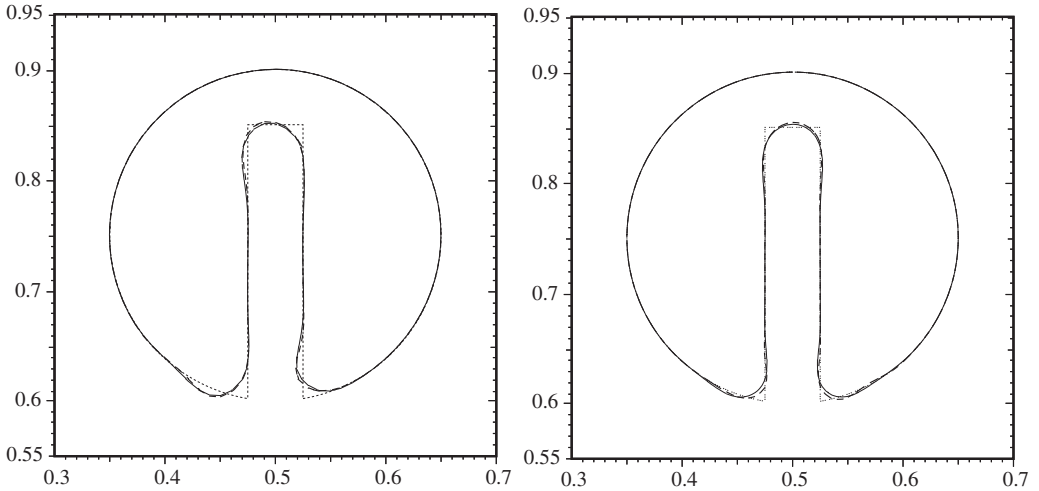

Figure 8. Zalesak's test for one full solid body rotation: on the left with the ELVIRA reconstruction (solid line) and the linear best fit (dashed line), on the right with the quadratic fit alone (solid line) and with the continuity algorithm (dashed line).

with $u=\partial\Psi/\partial y$ and $v=-\partial\Psi/\partial x$. The figure makes a solid body rotation centred at point $(x_0, y_0)=(0.5, 0.5)$ and the constant angular velocity Ω is such that a full revolution is performed in 628 time steps. The CFL number, based on the maximum co-ordinate velocity in the computational domain, is about 0.495. As stated by several authors [1, 12], this study is useful to test the accuracy of the reconstruction method and in particular its ability to represent fluid interfaces with high curvature. With respect to the advection algorithm, we notice that in a solid body rotation both $\partial u/\partial x$ and $\partial v/\partial y$ are zero. The advection methods, described in Section 4, differ among themselves mainly in the way they treat the 1D divergence correction terms, which are equal to zero in this case. Then, the results are basically the same. The only difference we notice is that the Eulerian explicit–implicit scheme gives rise to a more noisy solution. This problem can be easily overcome by introducing a small parameter ε (taken equal to 1×10^{-15} for this case), so that the volume fraction C is set to zero when it is less than ε, and to one when $C>1-\varepsilon$. Such a correction is not necessary for both the Lagrangian and the mixed schemes. Mass is always exactly conserved. In Figure 8 we show the reconstruction after one full solid body rotation: on the left the ELVIRA (solid line) and the linear fit (dashed line) reconstructions, on the right the quadratic fit (solid line) and the quadratic fit with continuity (dashed line) reconstructions. The results with the interface represented by a single segment in each cell are similar to those presented by several authors [1, 12, 7]. At this resolution, the difference between the real line and our four approximations is visible only near the corners. Usually, a VOF/PLIC reconstruction smoothes regions with high curvature, even more drastically near a corner where the discontinuity across the cell boundary is $\mathcal{O}(h)$. When advected by translations or solid body rotations, these discontinuities are progressively smeared out, till the radius of curvature can be correctly reconstructed by the VOF/PLIC method under consideration. With reference to Figure 8, we notice that the discontinuities are smeared out in time, but also advected faster by the counterclockwise flow,

Copyright © 2003 John Wiley & Sons, Ltd.

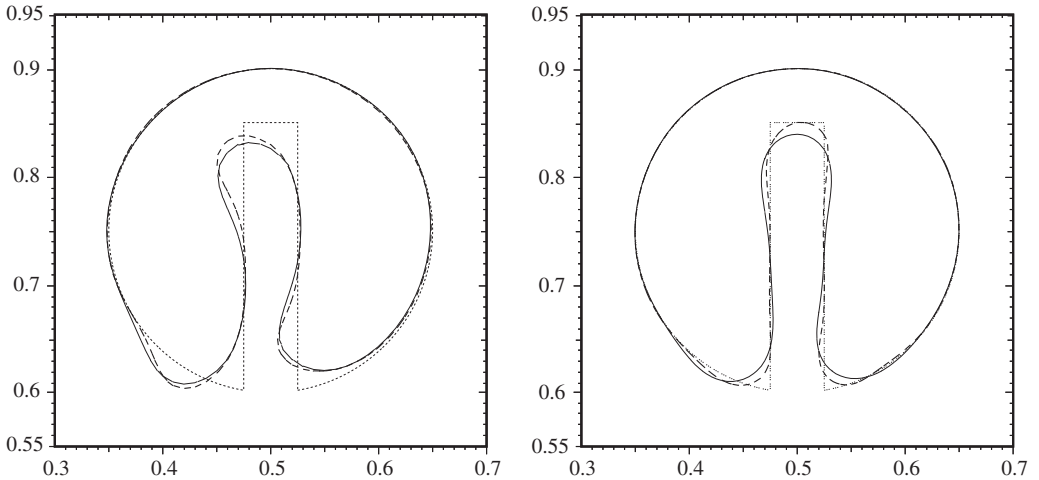

Figure 9. Zalesak's test for ten full solid body rotations: on the left with the ELVIRA reconstruction (solid line) and the linear fit (dashed line), on the right with the quadratic fit alone (solid line) and with the continuity algorithm (dashed line).

thus creating some phase error near the corners. With the two-line reconstructions, at the end of the rotation the approximated interface is more symmetric, closer to the real line in the proximity of the corners and with very little phase error. The fact that smoothing goes on till the radius of curvature is correctly reproduced by the method, is clearly seen in Figure 9, where we present the four reconstructions after ten revolutions, with 6280 time steps. The first two methods are not yet capable to describe correctly the high curvature regions which progressively overshoot the original line. On the contrary, the two-line methods have still little phase shift error near the corner, are symmetric and the degradation of the reconstruction is relatively mild. To compare our results with published work, we consider the Rudman–Zalesak version of this problem [4]. The computational domain is now a square of side 4, with 200 cells along each co-ordinate direction. The circle is centred at (2.,2.75), with a diameter of 50 cells. The width of the slot is 6 cells and 20 cells that of the upper bridge. A full rotation is exactly completed with 2524 steps, corresponding to a Courant number, based on the maximum co-ordinate velocity in the domain, of about 0.25. As in [4, 12], the errors are calculated using

$$E = \frac{\sum_{i,j} |C_{i,j} - \tilde{C}_{i,j}|}{\sum_{i,j} C_{i,j}} \tag{17}$$

where $C_{i,j}$ is the exact volume fraction at the end, or at the beginning, of the test, $\tilde{C}_{i,j}$ is the calculated value with one of the examined combinations of reconstruction and advection algorithms. The error is mainly generated in the reconstruction near the corners, so the results measure more the accuracy of the reconstruction technique in the representation of interfacial lines with regions of high curvature rather than the advection part of the algorithm. The errors are shown in Table V. In this table, the Puckett's reconstruction scheme is similar in spirit to ELVIRA, but it is based on an interative scheme to find a local minimum of expression (4),

Copyright © 2003 John Wiley & Sons, Ltd.

Table V. The error for the Rudman–Zalesak slotted disk rotation test. The first two results are taken from [12].

Reconstruction/Advection algorithms	Error
Youngs/Stream (unsplit)	1.07e − 2
Puckett/Stream	1.00e − 2
ELVIRA/Lagrangian (split)	1.00e − 2
Linear fit/Lagrangian	9.42e − 3
Quadratic fit/Lagrangian	5.47e − 3
Quadratic fit+continuity/Lagrangian	4.16e − 3

while the Stream scheme is a multidimensional advection algorithm developed by Harvie and Fletcher [12]. Boundary fluxes are calculated by subdividing each cell side in a number of equal stream sections, by integrating back in time for the computational time step along the fluid streamline passing through each stream section to determine a flux tube, and finally by determining the fluid volumes in each elementary flux tube that will cross the cell boundary during the time step. As previously stated, in a rotation the 1D divergence correction term is zero, so we obtain the same results with the Lagrangian, the Eulerian and the mixed schemes. The results show that the quadratic fit, with or without the added continuity, is superior to the ELVIRA and linear fit reconstructions for simple advection tests. We have also performed several tests with uniform translations and solid body rotations of a regular figure such as a circle, and the results are similar to those presented in the text.

5.2. *Rider–Kothe reversed single vortex flow*

A more precise assessment of reconstruction and advection algorithms is made with a flow containing a non-uniform vorticity field. The test is taken from the work of Rider and Kothe [1], and it has been considered by other authors as well [4, 12]. A circle of radius 0.15 is centred at point (0.5,0.75) in a unit square domain. All boundaries are periodic and the velocity field $\mathbf{v}$ is specified by the stream function

$$\Psi(x, y, t) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos\left(\frac{\pi t}{T}\right) \tag{18}$$

The circle of fluid is advected by this solenoidal velocity field, stretching and spiralling about the centre of the domain. It reaches a maximum deformation at time $t = T/2$, while at time $t = T$ it returns to the initial position because of the temporal component of Equation (18) [23]. Then, a precise indication of the accuracy of the adopted schemes can be deduced by comparing the initial and final position of the fluid. For consistency with previous work [1, 4, 12], we consider a period $T = 2$ and a 32^2 grid, while the error is now calculated as

$$E = \sum_{i,j} h^2 |C^f_{i,j} - C^i_{i,j}| \tag{19}$$

where h is the constant grid size along each co-ordinate direction and $C^f_{i,j}$ and $C^i_{i,j}$ are, respectively, the final and initial volume fraction values at cell (i, j). In Figure 10 we show the results obtained at time $t = T$ on the left with the ELVIRA reconstruction and the Lagrangian and the Eulerian explicit–implicit advection schemes, on the right ELVIRA and the

Copyright © 2003 John Wiley & Sons, Ltd.

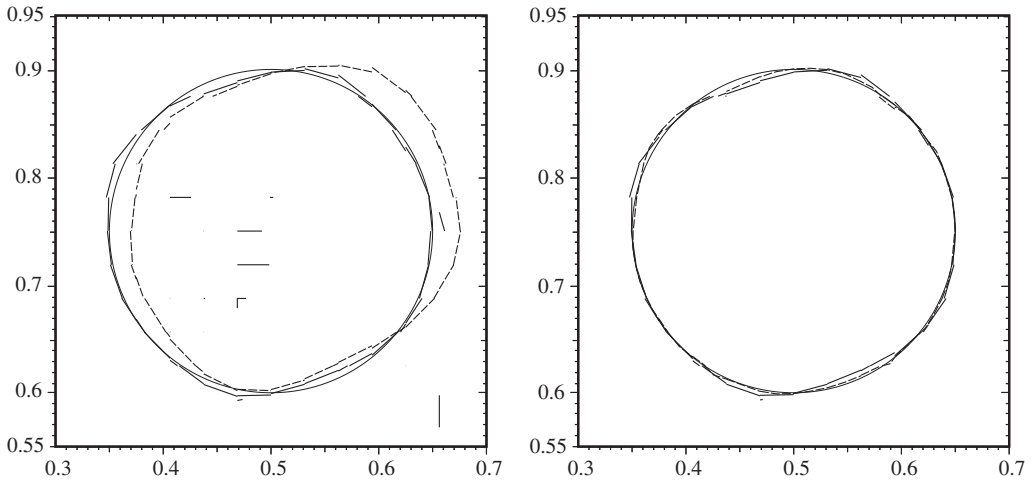

Figure 10. The reversed single vortex test for $T=2$ on a 32^2 grid, with the ELVIRA reconstruction and different advection schemes: on the left the Eulerian explicit–implicit scheme (solid segments) and the Lagrangian method (dashed segments), on the right the ELVIRA/LE–EI (solid segments) and the quadratic fit with continuity/LE–EI (dashed segments) schemes.

Table VI. Mass conservation and errors for the single vortex test, with the ELVIRA Reconstruction ($T=2,\ 32^2$ mesh).

Advection algorithm	Mass error at $t=T/2$ (%)	Mass error at $t=T$ (%)	Error
Lagrangian	3.82	6.4	1.22e − 2
Eulerian exp–imp	7.07e − 1	1.79e − 2	2.82e − 3
LE–EI	7.9e − 1	1.89e − 3	2.50e − 3
EI–LE	3.92e − 14	<1.e − 14	2.52e − 3

quadratic fit with added continuity reconstructions together with the LE–EI split advection. Mass conservation and geometric errors are summarized in Table VI. The Lagrangian algorithm [5] is rather clean, but there is a constant increase in mass, a deformation of the figure and a clear phase error (at the end of the simulation the reconstructed interface is behind the real figure). The mass production is due to the fact that with a CFL ~ 1, the fluid, say at cell (i,j) and moving to the right, feels first the 1D divergence term $(\partial u/\partial x)$ of cell (i,j) and then $(\partial v/\partial y)$ of cell $(i+1,j)$. The sum of these two terms is not zero, so the scheme is constantly changing mass. It is possible to develop a global mass conservation scheme, but the phase error still remains and it is due to the fact that there is not an implicit part in the algorithm. This phase error is clearly also present if one runs the Eulerian explicit scheme in both directions. With the Eulerian explicit–implicit scheme [1] there is no phase error, the deformation is still present and the reconstruction is rather noisy. By following step after step the evolution of the interface one can notice more debris left behind and little holes inside the bulk of the figure. The reason is that this scheme has the tendency to produce cells with $C<0$ and $C>1$, that in the algorithm presented in Figure 10 are simply set to 0 and 1,

Copyright © 2003 John Wiley & Sons, Ltd.

Table VII. Errors for the single vortex test, with several reconstruction algorithms and the EI–LE advection scheme ($T=2$, 32^2 mesh).

Reconstruction algorithm	Error (CFL = 1.0)	Error (CFL = 0.1)
ELVIRA	2.52e − 3	3.11e − 3
linear fit	1.75e − 3	2.22e − 3
quadratic fit	1.88e − 3	2.00e − 3
quadratic fit + continuity	1.09e − 3	1.14e − 3
Puckett/Rider-Kothe (unsplit)	2.36e − 3	—
Puckett/Stream (unsplit)	2.37e − 3	—

The last two results are taken from [1, 12], respectively.

respectively. We have also developed a local redistribution algorithm to take care of this problem and to remove isolated cells with the result that the figure now presents fewer debris and no hole, a better mass conservation and the error of Table VI decreases to 2.77×10^{-3}. With the two mixed schemes there is no phase error, the deformation is a bit less that in the Eulerian scheme and there are no debris or holes during the whole simulation. To remove truncation errors, we simply set the previously defined parameter ε to 10^{-15} (however, there is no difference if we set it to 10^{-12}). The results clearly indicate that the EI–LE scheme conserves mass within machine error, this is not the case for the LE–EI scheme that, together with the Eulerian method, benefits from the back and forth structure of the flow, as seen from the mass conservation errors at $t=T/2$. However, this will not be the case in a real dynamical simulation. So we can conclude that a second-order reconstruction scheme together with a mixed advection algorithm has the feature, within truncation error, to produces no flotsam and no under/overflow of the volume fraction function [3, 12, 1], moreover the EI–LE scheme conserves the mass correctly. We have shown this feature for an operator split advection algorithm in 2D, at the moment we do not know if we can extend this property to a split scheme in 3D or even to an unsplit method. In Table VII we present the results for several reconstruction algorithms with the EI–LE advection scheme, with a CFL number equal to 1 and 0.1, respectively. We notice that at low resolution, the quadratic fit is less performing than the linear fit, because the interface becomes very fragmented during the simulation. By running the simulation with a more realistic CFL number of 0.1 the evolution of the interface is smoother and the quadratic fit outperforms the linear one. For the LE–EI propagation, the mass conservation error decreases with the CFL number (here more than one order of magnitude). The bigger error at the end of the simulation for the smaller CFL number is mainly due to the increased number of reconstructions, here a factor 10, and it is related to the advection of the discontinuity of the reconstructed interface at the boundary of adjacent cells, as pointed out in the discussion of the Zalesak's test. Therefore, if we apply the linear fit procedure two times, the interface becomes a bit smoother and the final error decreases by 10%. The quadratic fit with continuity is always the best, showing an error variation of only a few per cents between the two simulations with different CFL number, again because of the interface is nearly continuous. Moreover, if we run the ELVIRA scheme with a CFL number equal to 0.5 or if we subdivide each temporal step in two halves when the CFL = 1, we have a change in the error of about 0.18. By comparing this result with those of Table VII obtained with two different unsplit algorithms [1, 12], we see that most of the difference for this

Copyright © 2003 John Wiley & Sons, Ltd.

Table VIII. Errors and convergence rates for the single vortex test.

	(a)	(b)	(c)	(d)	(e)	(f)
Grid	Error/order	Error/order	Error/order	Error/order	Error/order	Error/order
32^2	2.52e − 3	1.75e − 3	1.88e − 3	1.09e − 3	2.36e − 3	2.37e − 3
	1.95	1.91	2.08	1.96	2.01	2.07
64^2	6.46e − 4	4.66e − 4	4.42e − 4	2.80e − 4	5.85e − 4	5.65e − 4
	2.15	2.19	2.24	2.29	2.16	2.10
128^2	1.45e − 4	1.02e − 4	9.36e − 5	5.72e − 5	1.31e − 4	1.32e − 4

Note: (a) ELVIRA/EI–LE, (b) linear fit/EI–LE, (c) quadratic fit/EI–LE, (d) quadratic fit and continuity/EI–LE, (e) Puckett/Rider and Kothe, (f) Puckett/Stream. The last two results are taken from [1, 12], respectively.

test is coming from the number of reconstructions, rather than from the advection algorithm. Furthermore, we notice that in Figure 10 most of the error is on the top of the circle. In the evolution of the deforming interface, this region is the one which develops the highest curvature. By close inspection, the continuity routine sometimes fails to reconstruct correctly regions of high curvature when there is more than one way to connect segments. The reason is that it is not straightforward to code something that appears to be intuitively obvious by direct visual inspection. We point out that a VOF reconstruction has no memory of the previous one, so an alternative way to circumvent the problem rather than by increasing the complexity of the continuity algorithm, could be the development of a mixed markers and VOF algorithm. In Table VIII, we examine the convergence properties of the EI–LE scheme and four reconstruction algorithms and compare them with the results obtained with two unsplit advection algorithms [1, 2]. All results present roughly a second-order convergence rate. Given a reconstruction scheme, the unsplit schemes have smaller errors than the split EI–LE method, but they remain within 10%. On the other hand, the reconstruction technique plays a major role: the more continuous the interface, the better the results. In particular, we notice that as we increase the mesh from 32^2 to 64^2 cells, the quadratic fit performs better than the linear fit, because the discontinuities in the interface reconstruction become smaller. Finally, in Figure 11 we show the interface reconstruction at the end of the simulation on a 32^2 and on a 128^2 mesh. In the second case the difference between the real line and our reconstruction can be hardly seen at this resolution.

6. CONCLUSIONS

We have presented two new methods to reconstruct interfaces in free-surface and two-phase flows with cartesian two-dimensional grids. They are based on a least-square technique and they approximate the interface in each cell with a segment (as in a standard VOF/PLIC method) or with 2 consecutive segments, respectively. These methods have been tested with several test shapes and their convergence rate is about two. If the reconstruction is made continuous or almost continuous at the boundary between two adjacent cells, the error in the reconstruction further decreases, in particular at low resolution. We have also investigated the geometric nature of split advection algorithms based on Lagrangian and Eulerian approaches. We have developed a new mixed split advection scheme, with first an Eulerian implicit

Copyright © 2003 John Wiley & Sons, Ltd.

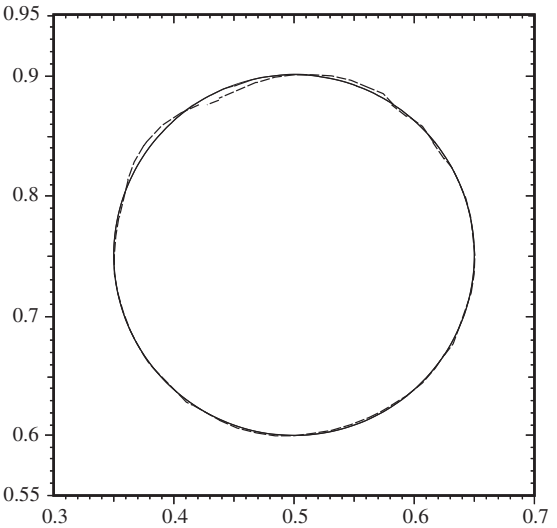

Figure 11. The reversed single vortex test for $T=2$ with the quadratic fit with continuity reconstruction and the EI–LE advection scheme on a 32^2 (dashed segments) and on a 128^2 (solid segments) grid.

step followed by a Lagrangian one, which conserves mass exactly with no flotsam and no undershoots or overshoots of the volume fraction C. Given a reconstruction scheme, this algorithm is slightly less performing than unsplit algorithms, mainly because the number of reconstructions is twice as much. However, when coupled to the new reconstruction methods, the combined algorithm performs consistently better than other combinations of VOF/PLIC and unsplit algorithms. The results point out that by making the interface continuous at the boundary between cells, or with very small discontinuities, the performance of the overall method is greatly improved.

REFERENCES

1. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
2. Noh WF, Woodward PR. Slic (simple line interface calculation). In Lecture Notes in Physics, van der Vooren AI, Zandbergen PJ (eds), vol. 59. Springer: Berlin, 1976; 330–340.
3. Lafaurie B, Nardone C, Scardovelli R, Zaleski S, Zanetti G. Modelling merging and fragmentation in multiphase flows with surfer. *Journal of Computational Physics* 1994; **113**:134–147.
4. Rudman M. Volume-tracking methods for interfacial flows calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**:671–691.
5. Jie Li. Calcul d'interface affine par morceaux (piecewise linear interface calculation). *Comptes Rendus des Seances del' Academie des Sciences Paris*, *Série IIb* 1995; **320**:391–396.
6. Parker BJ, Youngs DL. Two and three dimensional Eulerian simulations of fluid flow with material interfaces, 1992. *Technical Report* 01/92.
7. Pilliod Jr JE, Puckett EG. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Technical report*, Lawrence Berkeley National Laboratory, 1997. No. LBNL-40744.
8. Almgren AS, Bell JB, Colella P, Howell LH, Welcome ML. A conservative adaptive projection method for the variable density incompressible Navier-stokes equations. *Journal of Computational Physics* 1998; **142**:1–46.
9. Tezduyar T, Aliabadi S, Behr M. Enhanced-discretization interface-capturing technique (edict) for computation of unsteady flows with interfaces. *Computer Methods in Applied Mechanics and Engineering* 1998; **155**:235–248.
10. Strang G. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis* 1968; **5**:506.

Copyright © 2003 John Wiley & Sons, Ltd.

11. Youngs DL. Tmie-dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluid Dynamics*, Morton KW, Baines MJ (eds). Academic Press: New York, 1982; 273.
12. Harvie DJE, Fletcher DF. A new volume of fluid advection algorithm: The stream scheme. *Journal of Computational Physics* 2000; **162**:1–32.
13. Harvie DJE, Fletcher DF. A new volume of fluid advection algorithm: The defined donating region scheme. *International Journal for Numerical Methods in Fluids* 2001; **35**:151–172.
14. Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics* 1979; **31**:335.
15. Gueyffier D, Nadim A, Li J, Scardovelli R, Zaleski S. Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics* 1999; **152**:423–456.
16. Scardovelli R, Zaleski S. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *Journal of Computational Physics* 2000; **164**:228–237.
17. Pilliod JE. An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods. *Master's thesis*, U.C. Davis, 1992.
18. Poo JY, Ashgriz N. A computational method for determining curvature. *Journal of Computational Physics* 1989; **84**:483–491.
19. Aulisa E, Tomassini R, Scardovelli R, Manservisi S. Tecniche per la ricostruzione di interfacce con porzioni di superfici piane. In *Proceedings of the 19th UIT National Heat Transfer Conference*, *Modena*, *I*, Tartarini P (ed.), June 25–27, 2001; 183–187.
20. Rieber M. University of Stuttgart, private communication.
21. Aulisa E. Tecniche di ricostruzione e di convezione nello spazio tridimensionale dell'interfaccia di separazione tra sistemi bifase composti da fluidi immiscibili e incomprimibili. *Master's thesis*, University of Bologna, 2001.
22. Ashgriz N, Poo JY. Flair: flux line-segment model for advection and interface reconstruction. *Journal of Computational Physics* 1991; **93**:449–468.
23. Leveque RJ. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis* 1996; **33**:627–665.

Copyright © 2003 John Wiley & Sons, Ltd.