INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 24, 671–691 (1997)

VOLUME-TRACKING METHODS FOR INTERFACIAL FLOW CALCULATIONS

MURRAY RUDMAN

Commonwealth Scientific and Industrial Research Organisation (CSIRO), Division of Building, Construction and Engineering, PO Box 56, Highett, Vic. 3190, Australia

SUMMARY

A new algorithm for volume tracking which is based on the concept of flux-corrected transport (FCT) is introduced. It is applicable to incompressible 2D flow simulations on finite volume and difference meshes. The method requires no explicit interface reconstruction, is direction-split and can be extended to 3D and orthogonal curvilinear meshes in a straightforward manner. A comparison of the new scheme against well-known existing 2D finite volume techniques is undertaken. A series of progressively more difficult advection tests is used to test the accuracy of each scheme and it is seen that simple advection tests are inadequate indicators of the performance of volume-tracking methods. A straightforward methodology is presented that allows more rigorous estimates to be made of the error in volume advection and coupled volume and momentum advection in real flow situations. The volume advection schemes are put to a final test in the case of Rayleigh–Taylor instability.

KEY WORDS: volume tracking; interfacial flow; flux-corrected transport

1. INTRODUCTION

In the numerical computation of multifluid problems such as density currents or Rayleigh–Taylor instability there is a need for an accurate representation of the interface separating two immiscible fluids. Free surface flows such as water waves and splashing droplets are an approximation to the multifluid problem in which one of the fluids (usually a gas) is neglected as having an insignificant influence on the dynamics of the system. In a general free surface flow problem, fluid coalescence and detachment may occur and deforming meshes cannot be used. In this case the need of an accurate and sharp interface is even greater than in true multifluid computations. Although a slightly diffuse interface may be acceptable in a problem where the continuity, momentum and energy equations are solved throughout the entire mesh, in a free surface simulation the location of the interface determines the size and shape of the computational domain and specifies where boundary conditions must be applied. In this case a diffuse interface cannot be tolerated.

On finite volume (or difference) meshes, standard advection techniques can be used in multifluid problems to advect either the density or a material indicator function, however these methods are either diffusive (e.g. first order upwinding) or unstable (higher order schemes in which unphysical oscillations appear in the vicinity of the interface). Numerous techniques have been devised to limit the diffusiveness of low order schemes and to minimize the instability of high order schemes (see e.g.

CCC 0271–2091/97/070671–21 $17.50

Received January 1996
Revised June 1996

References 1–5). However, none of these techniques can guarantee the sharp, non-oscillatory interfaces that are desirable in immiscible multifluid flow simulations and essential in free surface problems on stationary meshes.

The earliest numerical technique designed specifically for simulating complicated free surface flows was the well-known marker and cell (MAC) method.[6] In MAC, Lagrangian marker particles are advected with the local fluid velocity and their distribution determines the instantaneous fluid configuration. Although the MAC method allows arbitrary free surface problems to be tackled, difficulties arise in the particle representation of the fluid. Because the number density of particles is finite (and typically fairly low), false regions of void can be created in flows with high shear (and hence high fluid extension). It is also difficult to obtain good quantitative information on interface orientation or partial cell volumes from the marker particles, and free surface boundary conditions (especially pressure) are problematic. They are applied in an approximate way that often leads to instability at the free surface.[7] Although some of these problems can be ameliorated by ensuring that each cell contains a large number of markers, significant computational overheads subsequently result.

The inclusion of a special set of surface marker particles[7] can also overcome some of these problems in two dimensions at the expense of making the application of the method less general. Extending this concept to 3D flows is possible, but not easily undertaken, and the logical difficulties that arise when two surfaces collide (enclosing surface fluid in the fluid interior) are algorithmically difficult to resolve.

For interfaces that remain single-valued with respect to one of the co-ordinate directions, the use of a height function (see e.g. Reference 8) in which the distance of the interface above a reference level is calculated offers a simple and robust method for simulating interfacial flow in both two and three dimensions. However, the restriction to single valued interfaces rules out a wide class of interesting and important problems.

Several volume advection techniques for finite volume and difference meshes have been developed with the aim of maintaining very sharp interfaces. The better known are the simplified line interface calculation (SLIC) method of Noh and Woodward,[9] the volume of fluid (VOF) method due to Hirt and Nichols[10] and the method of Youngs.[11] These methods are part of the focus of this paper and their discussion is left to Section 3.

There are other methods for calculating interfacial flows. A recent interface tracking scheme is that of Unverdi and Tryggvason,[12] in which the Lagrangian interface (represented by a set of connected line segments) is used to reconstruct a representation of the density field on a Eulerian mesh. However, the interface is spread over three or four mesh cells and the method is less straightforward than volume tracking with the necessity of solving an elliptic equation for C, and considerable logical overhead required in remeshing the interface as it becomes distorted. Another approach is the level set approach of Sussman *et al.*[13] However, the level set methodology does not guarantee volume conservation[14] in highly distorted flows and this can give rise to unacceptable errors in the method. A comparison of some of these other techniques with VOF methods was recently undertaken by Rider and co-workers.[14,15]

In this paper, only volume-tracking methods are considered and no further mention will be made of interface tracking or level set methodologies.

2. FCT-VOF

In volume tracking schemes a fractional volume or 'colour' function C is defined that indicates the fraction of a mesh cell that is filled with fluid of a particular type. Problems involving M fluids require

$M-1$ colour functions. Algorithms for volume tracking have been designed to solve the advection equation

$$\frac{\partial C}{\partial t}+\nabla\cdot(\mathbf{U}C)=0 \tag{1}$$

in a way that keeps interfaces sharp. Regular advection techniques (even those using flux-corrected transport) rapidly smear the interface over at least three or four mesh cells.

In the donor–acceptor scheme of Hirt and Nichols[10] a combination of first order up and down-wind fluxes is used to advect C. Following Hirt,[16] first-order upwind fluxes in 1D on a uniform staggered mesh can be shown to have an effective diffusion error term with coefficient $\kappa_{\mathrm{up}}=0{\cdot}5(\delta x|U|-\delta tU^2)$ (from which a Courant number of unity is seen to be the maximum for stability). In contrast, first-order downwinding is unstable, having an effective diffusion term with coefficient $\kappa_{\mathrm{dn}}=-0{\cdot}5(\delta x|U|-\delta tU^2)=-\kappa_{\mathrm{up}}$. Although unstable, the downwind scheme has the advantage of maintaining sharp interfaces,[17] which is one requirement needed to maintain a good interface representation. Thus, if a suitable combination of up and downwind fluxes can be formulated that eliminates both the diffusiveness of the upwind scheme and the instability of the downwind scheme, a volume tracking algorithm can be designed. One algorithm that does both is Zalesak's flux-corrected transport.[4]

The idea of adjusting fluxes calculated with a high order (non-monotonic) advection scheme to improve the monotonicity of the final result was introduced by Boris and Book[3] and was generalized and extended to multidimensions by Zalesak.[4] The basic idea involves several stages of calculation. First, an intermediate value of C (C^*) is determined using a monotonic (and hence diffusive) advection scheme. The scheme for solving the one-dimensional version of equation (1) (for mesh cell i) is written symbolically as

$$C_i^*=C_i^n-\frac{1}{\delta x}(F^{\mathrm{L}}_{i+1/2}-F^{\mathrm{L}}_{i-1/2}), \tag{2}$$

where F^{L} represents the low-order flux. An anti-diffusive flux is then defined that attempts to correct the numerical diffusion resulting from the low order scheme. An initial estimate of the antidiffusive fluxes ($F^{\mathrm{A}}_{i+1/2}$) is given by the difference between the high and low order flux approximations:

$$F^{\mathrm{A}}_{i+1/2}=F^{\mathrm{H}}_{i+1/2}-F^{\mathrm{L}}_{i+1/2}. \tag{3}$$

Application of the entire antidiffusive flux would simply result in the high order (unstable) flux being used, thus correction factors q are introduced that limit the anti-diffusive fluxes. The correction factors are calculated to ensure that no new extrema are introduced into the solution after application of the anti-diffusive fluxes. The minimum and maximum values allowed for a mesh cell i are based on the C^n and C^* in cell i and its two neighbours, $i-1$ and $i+1$. Details of the procedure used to limit the fluxes are described in Reference 4 and are not discussed here. The final step of Zalesak's FCT is to apply the corrected anti-diffusive fluxes and obtain the values of C at the new time:

$$C_i^{n+1}=C_i^*-\frac{(q_{i+1/2}F^A_{i+1/2}-q_{i-1/2}F^A_{i-1/2})}{\delta x}. \tag{4}$$

2.1. 1.D FCT-VOF

In FCT-VOF the F^{L} are calculated using first-order upwinding. For the flux at $i+\frac{1}{2}$,

$$F^{\mathrm{L}}_{i+1/2}=\begin{cases}U_{i+1/2}\delta tC_i & \text{if } U_{i+1/2}\geqslant 0,\\ U_{i+1/2}\delta tC_{i+1} & \text{if } U_{i+1/2}<0.\end{cases} \tag{5}$$

Instead of using a high order method to calculate the F^{H}, the flux calculation used in the donor-acceptor scheme[10] and the stability analysis in the manner of Hirt[16] suggest that the F^{H} could instead be calculated using a first-order downwind scheme:

$$F^{\mathrm{H}}_{i+1/2} = \begin{cases} U_{i+1/2}\delta t C_{i+1} & \text{if } U_{i+1/2} \geqslant 0, \\ U_{i+1/2}\delta t C_i & \text{if } U_{i+1/2} < 0. \end{cases} \tag{6}$$

The initial guess for the antidiffusive flux is given by $F^{\mathrm{A}} = F^{\mathrm{H}} - F^{\mathrm{L}}$. The F^{A} are limited as described in Reference 4 and then applied to the intermediate values of C. (The one difference in the limiting procedure in FCT-VOF is that the min/max values are also constrained to lie in the range [0,1] to ensure that round-off does not introduce boundedness problems.) The upwind scheme (with an effective diffusion coefficient κ_{up}) will introduce sufficient numerical diffusion to counteract the instability of the downwind fluxes (with effective diffusion coefficient $-\kappa_{\mathrm{up}}$).

The transport of a 1D step function by a uniform 1D velocity field using FCT-VOF is shown in Figure 1. Clearly, the integrity of the step function shape is maintained. In addition, the velocity of the interface is exact. For this simple test problem (one-dimensional volume tracking) FCT-VOF delivers an exact solution down to mesh resolution.

2.2. Multi-dimensional FCT-VOF

There are two ways in which the one-dimensional scheme can be extended to multidimensions.The first is by use of Zalesak's fully multidimensional FCT algorithm and the second is a direction split implementation.

In Zalesak's multidimensional scheme the 'advected and diffused' estimate of C (C^*) is calculated by multidimensional fluxing using the low order scheme. The antidiffusive fluxes are estimated as usual and are then limited using the C^n and C^* in the five-neighbourhood of the cell. This scheme was implemented here to give a two-dimensional FCT-VOF, however, the results proved to be unsatisfactory. An example is the advection of a 2D step function in a uniform velocity field as shown in Figure 2. Even though the interface remains extremely thin, the interface shape is not maintained. The primary source of error in the multidimensional scheme arises from the flux limiting, in which the direction of the major component of the diffusive error cannot be determined. Possible ways of improving the multidimensional algorithm are currently under investigation.

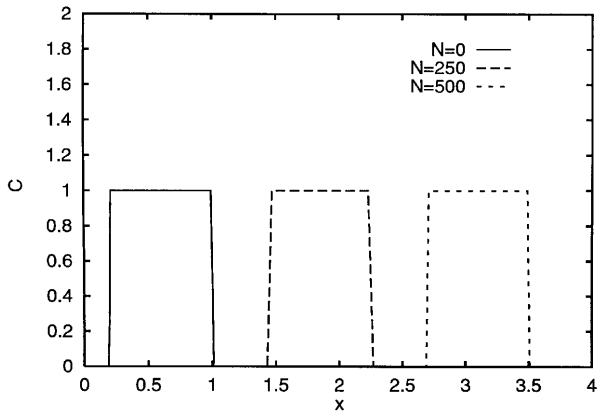

Figure 1. Advection of a step function in one dimension using FCT-VOF. From left to right: initial conditions and solutions after 250 and 500 time steps

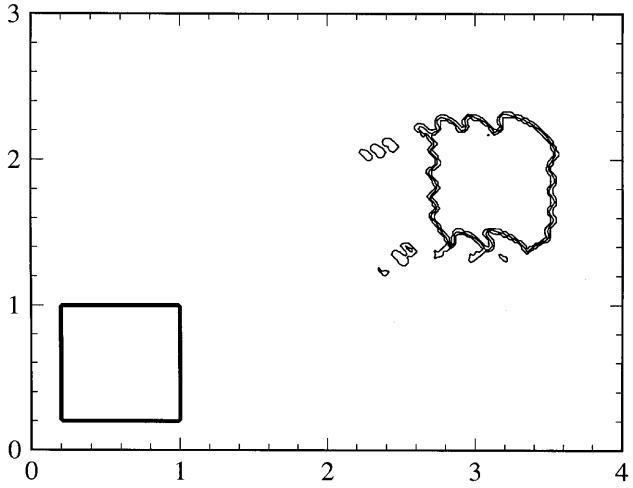

Figure 2. Advection of a square step function using fully multidimensional FCT-VOF

In practice, direction split FCT-VOF gives superior results. In two dimensions this involves sweeping the entire mesh in the x-direction with the 1D algorithm, updating C and following with a sweep in the y-direction. The order of the x- and y-direction sweeps is interchanged each time step to avoid the introduction of systematic error. Direction splitting introduces a problem not found in the multidimensional algorithm. After the first sweep it is possible that void fractions greater than unity may result. Such cells create a problem in the second half of the sweep, because the new fluxes will be calculated using these C values. Cells in the interior of the fluid may then attain values less than unity, which is clearly incorrect. There are several ways of overcoming this problem. In the approach used here, allowance is made for the effective change in volume of a cell during each one-dimensional sweep of the mesh. Defining $\delta V_{i,j}$ to be the volume of cell (i,j), at the beginning of each time-step set $\delta V_{i,j} = \delta x \delta y$. Then for an x-sweep of the mesh calculate

$$\begin{aligned} \tilde{C}_{i,j} &= C^n_{i,j}\delta V^n_{i,j} - (F^x_{i+1/2,j} - F^x_{i-1/2,j}), \\ \delta V^{n+1/2}_{i,j} &= \delta V^n_{i,j} - \delta t \delta y (U_{i+1/2,j} - U_{i-1/2,j}), \\ C^{n+1/2}_{i,j} &= \frac{\tilde{C}_{i,j}}{\delta V^{n+1/2}_{i,j}}. \end{aligned} \tag{7}$$

An equivalent set of steps is then performed in the y-sweep. After the two direction sweeps, $\delta V_{i,j}$ should again be equal to $\delta x \delta y$ and any discrepancy is equal to the error in satisfying the divergence condition, $\nabla \cdot U = 0$. To second order in time this procedure is equivalent to solving

$$\frac{\partial C}{\partial t} + \frac{\partial UC}{\delta x} = C\frac{\partial U}{\partial x}$$

in the x-direction, with a similar equation holding in the y-direction.

It is also extremely important to account for the effective cell volume change that occurs in each one-directional sweep when calculating the low-order diffused solution C^*. If this is not done, the integrity of the method is severely degraded. There is one final correction that is used in FCT-VOF. It is observed that small round-off errors can accumulate and affect the boundedness of the solution after a few hundred time steps. If, after every time step, negative C-values are set to 0·0 and any greater than 1·0 are set to unity, this problem does not arise. However, if such aprocedure is not

implemented, values of C used to calculate both up and down wind fluxes must be limited to lie in the range [0,1].

3. A SUMMARY OF OTHER TECHNIQUES

There are many techniques for advecting scalars (and hence fluid volume). The ones considered here are those that have been designed specifically for use in multifluid problems and that allow interfaces to be kept extremely sharp. Although the choice of schemes is limited below to three well-known ones, there are other possibilities (see e.g. References 18 and 19). Because all the schemes examined here are direction split, the correction for cell volume change, equation (7), is applied in each of them. A brief discussion of the three other techniques used in this paper is now given.

3.1. SLIC

In the SLIC method of Noh and Woodward[9] the interface in a cell is reconstructed using a straight line parallel to one of the co-ordinate directions. It is a direction-split algorithm and during each direction sweep, only cell neighbours in the sweep direction are used to determine the interface reconstruction. For the case of only two fluids there are nine possible interface cell configurations, which reduce to three basic cases as far as flux determination is concerned. (For details see Reference 9, the three basic cases here correspond to types I, II and III in that paper.) Because the interface reconstruction only looks at neighbours in the flux direction, an interface cell can (and often does) have a different representation for each direction sweep. This is illustrated in Figures 3(b) and 3(c), which show the reconstructions used for an x-sweep and y-sweep of the interface configuration shown in Figure 3(a). Once the approximate interface reconstruction has been made, fluxes of volume for each material are calculated geometrically.

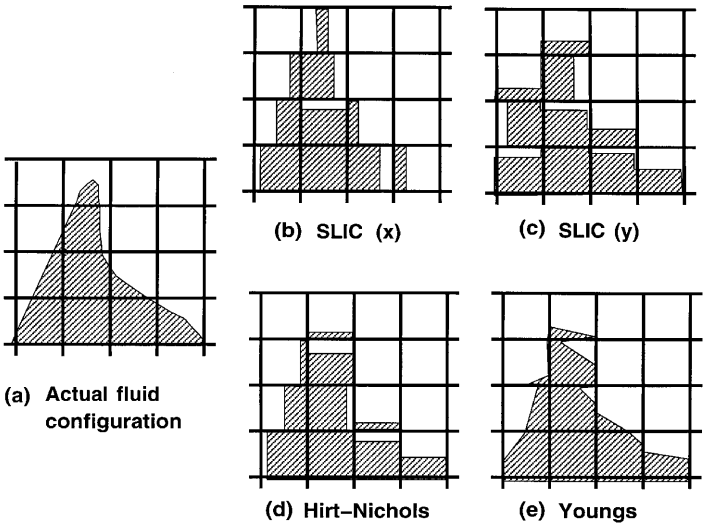

Figure 3. Interface reconstructions of actual fluid configuration shown in (a): (b,c) SLIC (x- and y-sweep respectively); (d) Hirt–Nichols' VOF; (e) Young's method

3.2. Hirt–Nichols' VOF

The original VOF method was described by Hirt and Nichols.[10] It uses an approximate interface reconstruction that, like SLIC, is parallel to one of the co-ordinate axes. Unlike SLIC, cells in a nine-neighbourhood are used to estimate the surface normal, and the interface is specified as horizontal or vertical depending on the relative magnitudes of the surface normal components. Figure 3(d) shows the VOF reconstruction of the fluid configuration seen in Figure 3(a).

For fluxes in a direction parallel to the approximate interface reconstruction, upwind fluxes are used. For fluxes in a direction perpendicular to the interface, donor–acceptor fluxes are used. As an example of the donor–acceptor scheme, consider the fluid configuration shown in Figure 4, with a positive x-velocity at $i+\frac{1}{2}$. The interface reconstruction in cell (i,j) is vertical; cell (i,j) is partially filled, as is the downwind cell $(i+1,j)$ (and $C_{i,j} > C_{i+1,j}$). The donor–acceptor scheme calculates the flux of volume across $(i+\frac{1}{2},jk)$ as

$$F_{i+1/2,j} = \delta y\{\mathrm{MIN}[C_{i,j}\delta x, U_{i+1/2,j}C_{i+1,j}\delta t + \mathrm{MAX}(0{\cdot}0, U_{i+1/2,j}(1{\cdot}0 - C_{i+1,j})\delta t - (1{\cdot}0 - C_{i,j})\delta x)]\}. \tag{8}$$

The four components of the right-hand side of equation (8) are:

(a) $C_{i,j}\delta x$, the maximum amount of fluid available for fluxing out of cell (i,j)
(b) $U_{i+1/2,j}C_{i+1,j}\delta t$, the downwind estimate of the C flux
(c) $U_{i+1/2,j}(1{\cdot}0 - C_{i+1,j})\delta t$, the downwind estimate of the void flux
(d) $(1{\cdot}0 - C_{i,j})\delta x$, the maximum amount of void that can be fluxed out of cell (i,j).

The MIN function ensures that no more fluid is fluxed out of cell (i,j) through the side at $(i+\frac{1}{2},j)$ than exists inside the cell. When fluid is fluxed out of the cell, implicitly void is also fluxed out and the MAX function ensures that no more void is fluxed out of cell (i,j) than exists there.

The scheme uses first-order up and downwind fluxes combined in such a way as to ensure stability at the same time as minimizing diffusion. As such, the method can be interpreted as a flux-corrected transport algorithm. In terms of Zalesak's FCT the downwind flux of fluid is the non-monotonic flux that must be limited to ensure that no new extrema are created in the cell. The MIN and MAX functions perform the flux-limiting role.

In the original VOF paper[10] no discussion about direction splitting the algorithm was given. Both a multidimensional and a direction-split algorithm were initially implemented here. The multi-dimensional algorithm resulted in appreciable volume error and the 'shedding' of many isolated blobs of 'flotsam' and 'jetsam'—it is not considered further and the direction-split algorithm is used below.

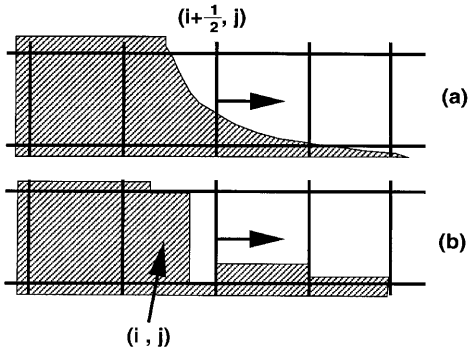

Figure 4. (a) True interface configuration and (b) interface reconstruction for Hirt–Nichols VOF

3.3. Youngs' method

Youngs' VOF (Y-VOF) method uses a more accurate interface reconstruction than either Hirt–Nichols' VOF or SLIC. An estimate is first made of the interface orientation β. The interface within a cell is then approximated by a straight line segment with orientation β and which cuts the cell in such a way that the fractional fluid volume is given by $C_{i,j}$. The geometry of the fluid 'polygon' resulting from this reconstruction is then used to determine the fluxes through any side on which the velocity is directed *out* of the cell. The method is also direction split. Very few details of the interface reconstruction and flux calculation were given in the original paper[11] and the implementation used here is discussed in the Appendix.

The method is interesting in that it cannot be written as a functional combination of up and downwind (or higher-order) fluxes in a simple way. There is a Lagrangian aspect to the method with advection of geometrical 'chunks' of fluid through cell boundaries, although these chunks lose their identity once they leave one cell and enter another.

4. SIMPLE ADVECTION TESTS

In this section no attempt is made to couple the advection of C to solutions of the momentum equations. Analytic velocity fields are chosen, which allow a comparison of the volume-tracking schemes alone to be undertaken. Each of the methods was rigourously tested to ensure that the implementation used here was performing correctly. All the methods have slight failings such as the generation of jetsam or slight broadening of interfaces and the generation of void fractions outside the range of (0,1) due to round-off errors. *Ad hoc* fixes can be proposed to fix problems specific to each method, although none is implemented here except to ensure that values of C less than zero or greater than unity are set to zero or unity respectively. This procedure is applied to all methods and results in a volume change that is readily quantified and always found to be negligible.

Because all algorithms are direction split, equation (7) is applied to all methods to ensure that cells cannot overfill during one direction sweep of the mesh.

4.1. Constant, unidirectional velocity field

The simplest two-dimensional volume advection problem is that of a unidirectional velocity field. Three different scalar fields are considered as shown in Figures (5a) amd 5(f). They are

(a) a hollow square aligned with the co-ordinate axes
(b) a hollow square at an angle of $26{\cdot}57^\circ$ to the x-axis
(c) a hollow circle.

In these C contour plots (and all others below), contour levels of 0·025, 0·5 and 0·975 are displayed. Each of the three scalar fields is separately advected with two velocity fields ((0,1) and (2,1)) and the six combinations together encompass most combinations of interface alignment, curvature and flux direction that may arise in such a simple problem.The mesh size is 200×200, the exterior extent of each of the shapes is initially 40 mesh cells and the distance between the outer and inner interface is 10 mesh cells. The mesh Courant number is 0·25 and advection proceeds for approximately 500 time steps in each case.

The results of advecting these three scalar fields with the two velocity fields are shown in Figures 5(b) and 5(g) for SLIC, Figures 5(c) and 5(h) for Hirt–Nichols' VOF, Figures 5(d) and 5(i) for FCT-VOF and Figures 5(e) and 5(i) for Youngs' method. (For each method and velocity field the results for each of the three scalar fields are displayed on the same figure for compactness, although they were advected separately.)

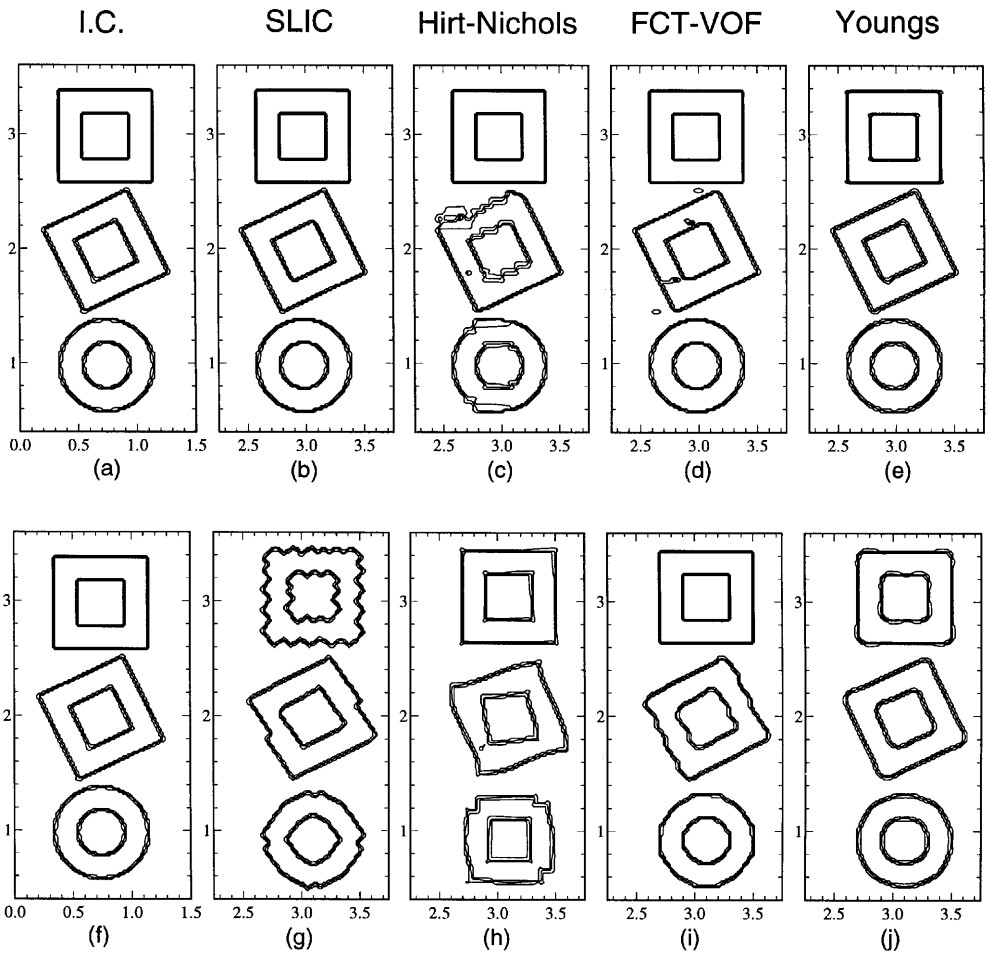

Figure 5. Advection with unidirectional velocity fields (1,0) (top) and (2,1) (bottom). At the left are the initial conditions (I.C.) followed by the results for SLIC, Hirt–Nichols' VOF, FCT-VOF and Youngs' method

SLIC maintains good interface shape in all cases in which advection is aligned with a co-ordinate direction (Figure 5(b)) but is unacceptably poor when advection is inclined to the co-ordinate axes (Figure 5(g)). This is particularly obvious in advection of a co-ordinate-aligned square, which develops large interfacial ripples with an amplitude of several mesh cells. Interface disturbances start at the upwind corner of the square and propagate along the sides.

Hirt–Nichols' VOF (Figures 5(c) and 5(h)) only produced good results for the co-ordinate-aligned square advected in a co-ordinate direction, in all other cases there was significant interface distortion and some interface spreading. (The spreading could be alleviated with *ad-hoc* corrections such as those suggested by Hirt and Nichols.[10]) The somewhat poorer performance of Hirt–Nichols' VOF (compared with SLIC) is surprising considering that the interface reconstruction is multidimensional (even though the flux calculation is direction-split).

FCT-VOF performs well in most cases, the exceptions being the angled square for both velocity fields (Figures 5(d) and 5(i)). The appearance of jetsam and voids seen in Figure 5(d) highlights the inability of FCT-VOF to handle thin regions of fluid—in this case the corners of the square which are initially only one mesh cell wide. The use of downwind fluxes and the limiting procedure in FCT-

VOF always ensures that regions of fluid only one cell thick in a particular direction will not be advected in that direction and will thus remain stranded on the mesh. *Ad hoc* corrections for this problem could probably be implemented, however it is not reasonable to expect regions of fluid equivalent in size to the mesh resolution to be handled in a physically meaningful way by any numerical scheme. Advection of the co-ordinate-aligned square and circle with both velocity fields results in acceptably good interface shape and thickness.

Youngs' method gives the most consistent results as far is interface shape is concerned (Figures 5(e) and 5(g)), although the case handled best by the other three methods (the co-ordinate-aligned square advected in a co-ordinate direction, Figure 5(e)) is not handled quite as well by Youngs' method (although the error is very small). The method also gives rise to rounding of sharp corners (seen particularly in Figure 5g), but once again this is not unexpected, as the singularity existing at sharp corners cannot be satisfactorily resolved by any Eulerian numerical scheme.

The solution error is quantified by defining

$$E = \frac{\sum_{i,j} \|C^n_{i,j} - C^{\mathrm{e}}_{i,j}\|}{\sum_{i,j} C^0_{i,j}}, \tag{9}$$

where C^n is the calculated solution after n time-steps, C^{e} is the exact solution after n time-steps (which is easily determined with such simple velocity fields) and C^0 is the initial solution.

Solution errors for each of the three scalar fields, two advecting velocities and four methods are shown in Table I. Although there is no consistent pattern to the errors, the table shows that Y-VOF is superior to SLIC and Hirt–Nichols' VOF in all cases but one, and superior to FCT-VOF in all cases but two. In general the error in FCT-VOF is about one-third that of SLIC and Hirt–Nichols' VOF, but about twice as poor as Y-VOF. Thus these tests indicate that the piecewise linear interface reconstruction (sometimes termed PLIC[14]) used in Youngs' method in conjunction with geometrically calculated fluxes gives the best volume tracking performance in simple unidirectional advection problems.

4.2. Solid body rotation

Zalesak's problem,[4] in which a slotted circle is rotated through one or more revolutions, is widely used as a test of scalar advection methods. In the version used here, the mesh size is 200×200 (physical dimensions $4{\cdot}0 \times 4{\cdot}0$), the diameter of the slotted circle is 50 mesh cells and the slot width is six mesh cells. The axis of rotation is the centre of the computational domain (2·0, 2·0) and the circle centre is initially at (2·0, 2·75). The initial conditions are shown in Figure 6. The velocity field has a magnitude of 1·0 at the middle of the domain edges, the Courant number is 0·25 and one rotation corresponds to 2524 time steps.

Table I. Errors for unidirectional velocity fields

Velocity	Scalar field	SLIC	Hirt–Nichols	FCT-VOF	Youngs
(1,0)	Square (0°)	$8{\cdot}42 \times 10^{-8}$	$1{\cdot}03 \times 10^{-8}$	$3{\cdot}89 \times 10^{-8}$	$1{\cdot}08 \times 10^{-3}$
	Square (26·75°)	$1{\cdot}46 \times 10^{-2}$	$6{\cdot}91 \times 10^{-2}$	$2{\cdot}32 \times 10^{-2}$	$5{\cdot}35 \times 10^{-3}$
	Circle	$1{\cdot}30 \times 10^{-2}$	$4{\cdot}55 \times 10^{-2}$	$1{\cdot}28 \times 10^{-2}$	$3{\cdot}08 \times 10^{-3}$
(2,1)	Square (0°)	$1{\cdot}32 \times 10^{-1}$	$6{\cdot}86 \times 10^{-3}$	$1{\cdot}63 \times 10^{-8}$	$2{\cdot}58 \times 10^{-2}$
	Square (26·57°)	$1{\cdot}08 \times 10^{-1}$	$1{\cdot}60 \times 10^{-1}$	$8{\cdot}15 \times 10^{-2}$	$3{\cdot}16 \times 10^{-2}$
	Circle	$9{\cdot}18 \times 10^{-2}$	$1{\cdot}90 \times 10^{-1}$	$3{\cdot}99 \times 10^{-2}$	$2{\cdot}98 \times 10^{-2}$

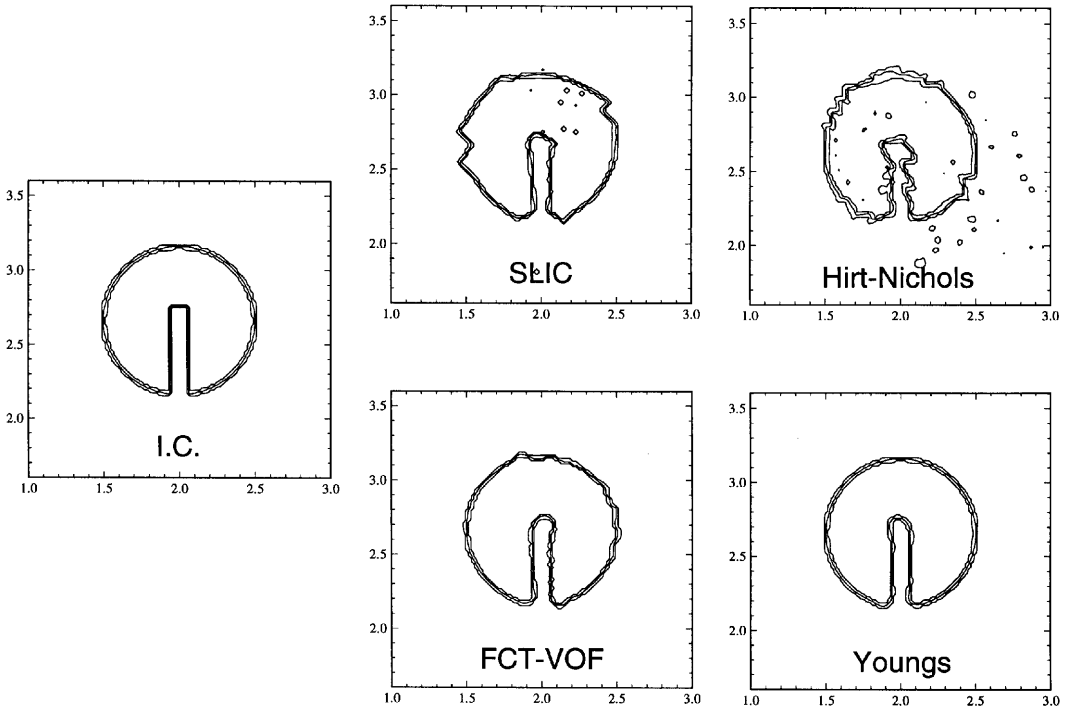

Figure 6. Zalesak's test problem for solid body rotation: C contours for initial conditions (I.C.) and results after one rotation for each of four schemes

Results after one rotation are shown in Figure 6 for each of the four methods, and errors are presented in Table II. Both SLIC and Hirt–Nichols' VOF generate quantities of jetsam and both give quite poor interface shapes after one rotation. Hirt–Nichols' VOF also shows some slight spreading of the interface. In contrast, FCT-VOF and Youngs' method give sharp interfaces without jetsam and both give rise to acceptable interface shapes, although the sharp corners at both ends of the slot become rounded. The relative magnitudes of the errors are similar to those in the previous example.

4.3. Shearing flow

As discussed by Rider and Kothe,[14] an incomplete assessment of the integrity of volume tracking methods is given using simple translation and rotation tests because of a lack of topological change in the solution. The velocity fields used in Sections 4·1 and 4·2 satisfy not only $\nabla \cdot U = 0{\cdot}0$ but also $\partial U/\partial x = 0{\cdot}0$ and $\partial V/\partial y = 0{\cdot}0$. Thus the shape of the fluid region does not deform during its advection. Even with such restrictive constraints on the velocity field, it is seen that achievement of good translation and rotation of a step function on a Eulerian mesh is still a difficult task. In realistic problems the situation is far more complicated, with stretching, shearing, fluid merging and break-up all possible in a flow. The key element missing from translation and rotation tests is the presence of fluid shear; in this subsection, shear is introduced into the velocity field.

Table II. Errors for Zalesak's solid body rotation test after one roation

SLIC	Hirt–Nichols	FCT-VOF	Youngs
$8{\cdot}38 \times 10^{-2}$	$9{\cdot}62 \times 10^{-2}$	$3{\cdot}29 \times 10^{-2}$	$1{\cdot}09 \times 10^{-2}$

The following simple velocity field is chosen:

$$U(x, y) = \cos(x)\sin(y), \qquad V(x, y) = -\sin(x)\cos(y),$$

with $x, y, \in [0, \pi]$. The mesh size is 100×100. The simulation is integrated forward in time for N steps (with a Courant number of 0·25) before reversing the sign of the velocity field and integrating for an additional N steps—a perfect advection scheme would return the initial C configuration. The initial condition is a circle of radius $\pi/5$. The shapes after forward and reverse integration for $N = 1000$ and 2000 time steps are shown in Figures 7(a)–7(d). Errors for $N = 250$, 500, 1000 and 2000 are presented in Table III.

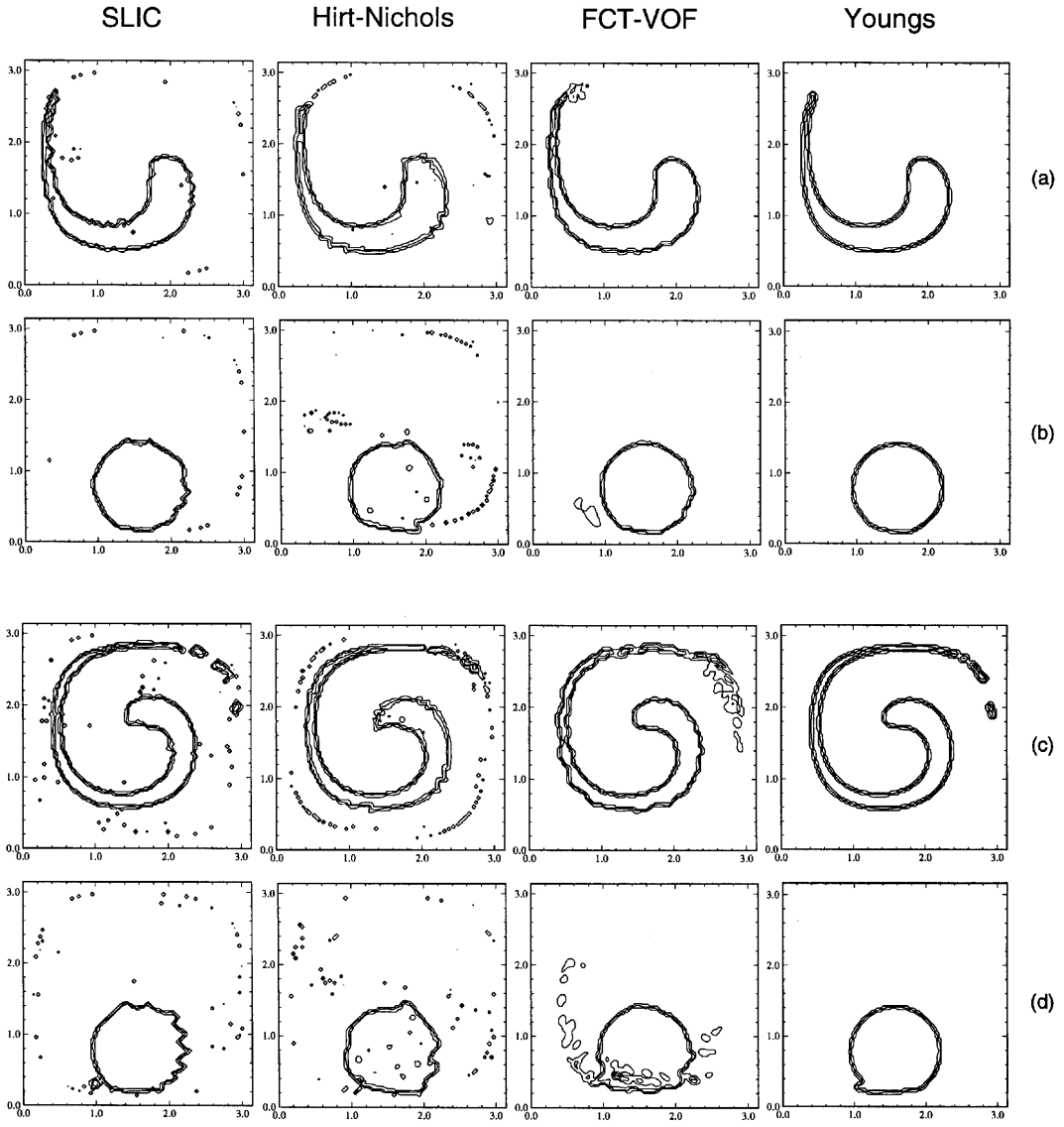

Figure 7. Results for shearing field: (a) after 1000 steps forward; (b) after 1000 steps forward followed by 1000 steps backward; (c) after 2000 steps forward; (d) after 2000 steps forward followed by 2000 steps backward

Table III. Errors for shearing flow

N	SLIC	Hirt–Nichols	FCT-VOF	Y-VOF
250	$2{\cdot}72 \times 10^{-2}$	$3{\cdot}24 \times 10^{-2}$	$1{\cdot}94 \times 10^{-2}$	$2{\cdot}61 \times 10^{-3}$
500	$3{\cdot}30 \times 10^{-2}$	$4{\cdot}00 \times 10^{-2}$	$2{\cdot}35 \times 10^{-2}$	$5{\cdot}12 \times 10^{-3}$
1000	$4{\cdot}59 \times 10^{-2}$	$6{\cdot}66 \times 10^{-2}$	$3{\cdot}14 \times 10^{-2}$	$8{\cdot}60 \times 10^{-3}$
2000	$9{\cdot}02 \times 10^{-2}$	$1{\cdot}09 \times 10^{-1}$	$1{\cdot}44 \times 10^{-1}$	$3{\cdot}85 \times 10^{-2}$

For $N \leqslant 10000$, FCT-VOF maintains a slight advantage over both SLIC and Hirt–Nichols' VOF, although by a smaller margin than in the simple advection tests in sections 4.1 and 4.2. The clear winner is Youngs' method which is up to an order of magnitude more accurate for all N. For $N = 2000$ all methods have begun to break down because the shearing flow has started to stretch the fluid circle into thin filaments only one or two cells wide. Because the mesh is unable to represent the C configuration accurately by this time, it would be fortuitous if any method were able to satisfactorily integrate backward in time and obtain the original solution. Notably, Youngs'method does a fairly good job nevertheless.

On the basis of the simple advection problems discussed previously, FCT-VOF appeared to be a suitable alternative to both SLIC and Hirt–Nichols' VOF, with better interface shapes and errors typically three times smaller. Although FCT-VOF still outperforms both here, the advantages are less clear, with all methods except Youngs' having severe problems returning the initial interface for $N > 1000$. It is seen in this example that after the application of a significant period of shearing, all methods break down to an unacceptable level. Although this is due in part to the inability of the mesh to resolve the fine features of the interface, it reinforces the point that simple advection tests are inadequate when determining the performance of volume-tracking methods.

4.4. Efficiency and order of accuracy

Because the advection of C typically consumes only a small percentage of a fully coupled Navier–Stokes solution for interfacial flow problems, the efficiency of C-advection is often not a primary concern. For completeness the CPU time taken for each of the cases above was compared. The average for each case (normalized to the time for SLIC) is shown in Table IV.

The shearing flow example considered in section was run with several different spatial resolutions and several different time steps. As expected, each of the volume advection schemes considered here is first order in both time and space.

5. A REAL PROBLEM: RAYLEIGH–TAYLOR INSTABILITY

Advection tests such as the ones described above give an indication of the robustness and accuracy of a volume tracking method, however it is seen that considering just one or two of the tests in isolation does not give a good picture of the strengths and, more importantly, the weaknesses of a method—

Table IV. Relative CPU time for each method in simple advection test

SLIC	Hirt–Nichols	FCT-VOF	Youngs
1·0	1·5	4·0	3·3

many different combinations of interface shape, orientation and advection velocity field should be considered.

Because of the difficulty of ensuring that an analytic velocity field will include the degree of complexity encountered in practical situations, a real flow is now considered. An additional problem with the advection tests above is that they may give an indication of the error of passive scalar advection, but when the scalar is dynamically coupled to the momentum solution, errors in scalar advection will result in errors in momentum advection. The feedback of these two errors may totally destroy a solution. The difficulty in testing in real flow situations is that generally there are no analytic solutions available and it is difficult to estimate the error arising from use of a scheme. A methodology is suggested here that will give both an error associated with the coupled integration of a non-passive C and momentum, and also an error associated with the advection of C alone.

Because Youngs' method was the most accurate method in all the test problems above, it was used to advect C in a high resolution solution of the full set of (scaled) momentum equations

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{U}C) = 0, \tag{10}$$

$$\rho = \rho_1 C + \rho_2 (1{\cdot}0 - C), \tag{11}$$

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U}\mathbf{U}) = -\nabla P + \frac{1}{Fr^2}\rho \hat{\mathbf{g}} + \frac{1}{Re}\nabla \cdot \tau, \tag{12}$$

$$\nabla \cdot \mathbf{U} = 0 \tag{13}$$

The high-resolution solution is then considered to be the 'exact' solution (C^{e}) for the problem. To obtain an estimate of the error in the forward integration of the coupled scalar and momentum advection equations, equations (10)–(13) were solved using each of the four volume advection schemes at a coarse resolution. The high resolution solution is then averaged to obtain an estimate of the exact solution at the coarse resolution, and the error is calculated as in equation (9). This provides an error estimate of the coupled solution.

To obtain an estimate of the error in just C advection using each of the schemes, the calculated velocity fields used to advect C at each time step are saved to form a time series. The final C field is then used as an initial condition for a reverse time integration of C. Each of the velocity fields is read in turn, reversed in direction and used to integrate C back one time step toward the beginning of the fully coupled simulation. In the absence of any C advection errors, the C field that results after advecting backward should be identical with the initial condition. Any difference is an indication of the error in C advection alone. Although this requires a large number of velocity fields to be stored, it is a good test of a volume advection scheme in a real flow situation.

The example chosen here is that of Rayleigh–Taylor instability. This is a good example to test the effect of coupled C and momentum transport, because the flow is density-driven and errors in the C field will lead to errors in the momentum solution via equation (11). The domain is a rectangular box of physical dimensions $1{\cdot}0 \times 3{\cdot}0$. The coarse solutions are obtained with a mesh resolution of 64×192 and the 'exact' solution is obtained on a 192×576 mesh. Initially the top one-third of the domain is filled with fluid of density $\rho_1 = 1{\cdot}2$ ($C = 1{\cdot}0$) and the lower two-thirds is filled with fluid of density $\rho_2 = 1 (C = 0{\cdot}0)$. The Froude number ($Fr = U/\sqrt{(gL)}$) is $0{\cdot}5$ and Reynolds number is 500. A perturbation is applied to the initial C distribution with amplitude $0{\cdot}02$ and half wavelength $1{\cdot}0$ (see Figure 10(e)). The scheme used to solve equations (10)–(13) is based on the MAC method,[6] uses flux-limited third-order differences for the advective operator, centred differences for diffusion and a multigrid pressure correction projection, and is second order in time everywhere except for C advection which is first order in time. A maximum mesh Courant number of $0{\cdot}25$ was used in the time integration and the solution was integrated forward in all cases to a dimensionless time of $8{\cdot}0$, which

corresponds to approximately 1340 time steps. The integration time was primarily determined by disk storage constraints for the velocity fields needed in the time series. By time $t = 8{\cdot}0$, the solution was sufficiently distorted to present a severe test of volume advection schemes. Boundary conditions are no-slip except for $x = 0$ which is an axis of symmetry.

The coarse solutions at times $t = 4.0, 6.0$ and 8.0 for SLIC, Hirt-Nichols' VOF, FCT-VOF and Youngs' method are compared with the exact solution (contoured at the fine resolution) in Figures 8(a)–8(e). After an initial linear growth period a plume forms which descends into the lower layer, forming a plume cap that has started to roll up by time $t = 6{\cdot}0$. By time $t = 8{\cdot}0$, the cap has wrapped around itself twice and a secondary sidewise plume has developed at the top right of the flow. Not surprisingly, Youngs' method compares very well with the high resolution 'exact' solution. A little more surprising perhaps is that the other three methods result in quite similar solutions. The early linear growth rate of the instability is identical in all cases. The later non-linear growth of the dense plume cap as it penetrates the lower layer and rolls up also proceeds very similarly for each case. Apart from features that cannot be resolved on the coarse mesh (these are most clearly seen when comparing Youngs'method with the exact solution), the primary difference between Youngs' and the other coarse solutions is the development of Kelvin–Helmholtz-type instabilities on the edge of the descending plume cap—they are clearly evident at time $t = 8{\cdot}0$ in all three other solutions. These instabilities are all numerical in nature (and are quite different for each of SLIC, Hirt–Nichols' VOF and FCT-VOF). They result from C advection errors that cause spurious oscillations to appear on the interface. (Such oscillations were seen previously in Figures 5 and 6.) Although these oscillations do not destroy the solution here, they do introduce significant error and show that coupled C and momentum advection must be considered when volume tracking schemes are being evaluated.

Although coupled C and momentum advection made some significant differences to the final solution in this example, there are flow situations in which spurious interface oscillations may not adversely affect the solution because they are damped by some other mechanism (for example, in the presence of significant surface tension or in the case of surface waves in which gravity may flatten perturbations).

The error in forward time integration is shown in Figure 9(a) as a function of time. Although the interface position is clearly superior for Youngs' method, this fact is not adequately reflected in the error estimates which show that the error is only two to three times better than the other schemes which, at each time, all have approximately equal error. Because the error arises from both C and momentum advection, it is maybe not surprising that the apparently superior interface reconstruction resulting from Youngs' method is not reflected in the error estimates.

The error resulting from just C advection was estimated by integrating backward in time using the time series of stored velocity fields. Initial conditions for C were the C fields at times $t = 1{\cdot}0$, $2{\cdot}0, \ldots, 8{\cdot}0$. The error for each method as a function of time is shown in Figure 9(b). In this case, Youngs' method is seen to result in an order-of-magnitude less error than Hirt–Nichols' VOF and half-an-order less than SLIC and FCT-VOF which have similar errors. The larger errors of Hirt–Nichols' VOF are consistent with its inferior performance found in Section 4.1. The results of the backward integration from time $t = 8{\cdot}0$ are shown in Figure 10. Youngs' method is the only method which maintains a sharp and well-defined interface.

The results from this test case illustrate that the error arising from just the volume advection scheme is not necessarily a good indicator of the error when C advection is dynamically coupled to the momentum solution. Although SLIC is able to integrate backward reasonably (albeit with the creation of considerable jetsam), it performed worst in the forward time integration. Hirt–Nichols' VOF and FCT-VOF were comparable in the (dynamically coupled) forward integration, yet Hirt–Nichols' VOF resulted in the largest volume advection errors. An indication of the errors resulting from the dynamically coupled solution of C and momentum cannot be reliably obtained by

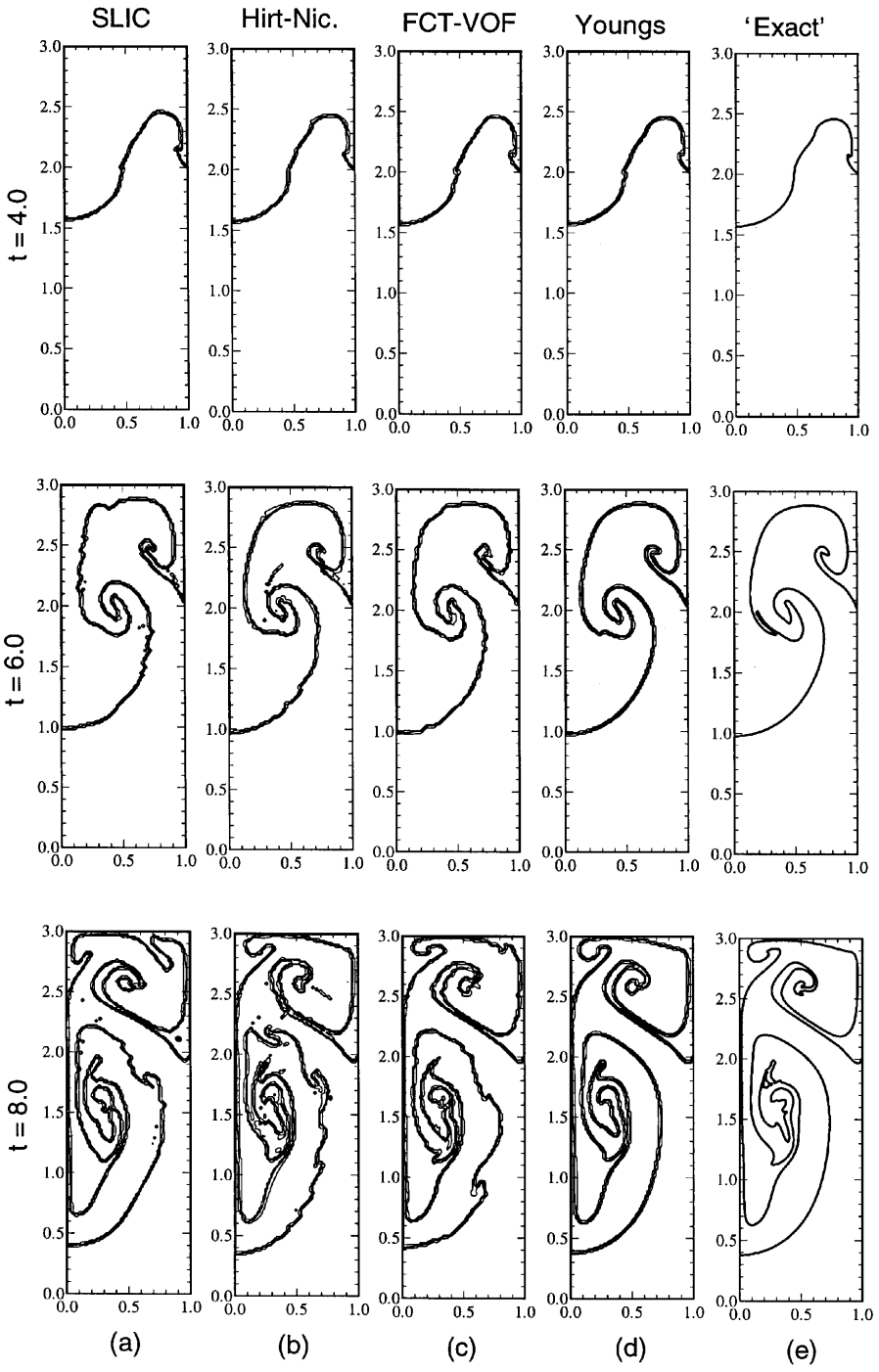

Figure 8. Rayleigh–Taylor instability at times $t = 4{\cdot}0$, $6{\cdot}0$ and $8{\cdot}0$ for each scheme. The high-resolution 'exact' solution is shown at the right

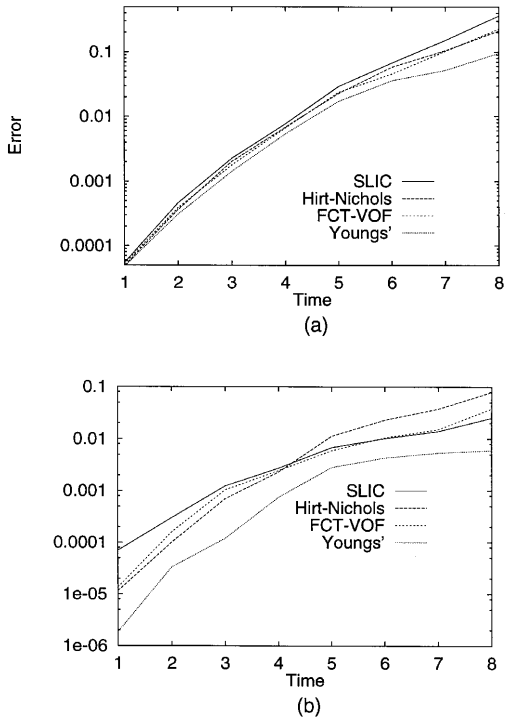

Figure 9. Error in (a) forward and (b) backward integration for Rayleigh–Taylor instability as a function of integration time

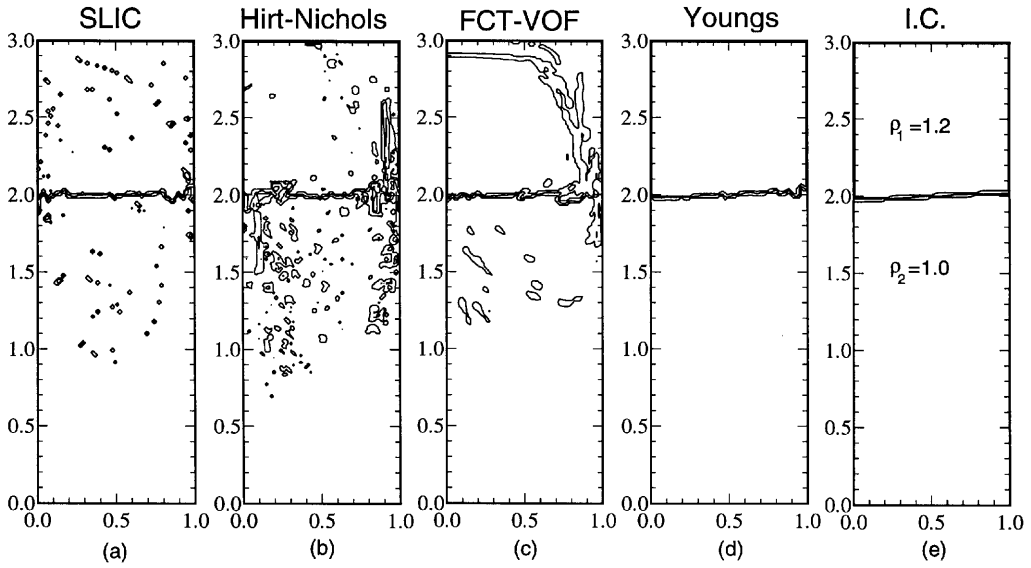

Figure 10. Results of backward integration from time $t = 8{\cdot}0$ for Rayleigh–Taylor instability. The initial condition is shown in (e)

considering volume advection errors alone. Although a scheme that gives large errors in volume advection is unlikely to give a satisfactory solution to a real problem, a scheme that gives small volume advection errors will not necessarily give small errors to the coupled problem. The particular weaknesses of the volume advection scheme coupled with the physical problem will determine the ultimate utility of any scheme.The recommendation is that the application in mind must be used to test the error in the coupled advection of C and momentum (and hence overall solution accuracy). Reliance on C advection tests only is insufficient.

6. CONCLUSIONS

The new technique introduced here for volume tracking (FCT-VOF) is unique in that an approximate interface reconstruction is not required, yet extremely sharp interfaces (of the order one mesh cell) are maintained. Despite considerable effort, the method is still inferior to the method originally proposed by Youngs,[11] although is two to three times more accurate than both the SLIC method of Noh and Woodward[9] and the original VOF method of Hirt and Nichols[10] in simple advection tests. When more realistic advection tests are considered, its utility is less clear cut. One possible advantage of the method is that it is readily extended to three dimensions and curvilinear (orthogonal) co-ordinate systems, which cannot be said for Youngs' method (although both extensions appear possible). FCT-VOF has been implemented in cylindrical coordinates[20] to investigate splashing liquid droplets and has been found to give superior results to Hirt–Nichols' VOF.

Considering a wide range of test problems is essential when assessing the accuracy of volume tracking schemes. As clearly seen from the variety of problems examined above, some schemes will handle one simple test case extremely well, whilst an alternative (but equally simple case) may be handled poorly (contrast the co-ordinate-aligned squares in Figures 5(b) and 5(g). Fluid shear is the single most important factor to include in a flow field for test problems. Without shear the topology of the fluid region does not change and the test is insufficient to determine a scheme's likely usefulness in real flow situations.

The best test is a coupled solution of the C and momentum equations under the flow conditions of the problem being considered. A simple methodology was presented that allows an estimate to be made of the error arising from volume advection in real flow situations. Its use is recommended when volume advection schemes are being assessed.

APPENDIX: YOUNGS' VOF

Although Youngs' idea first appeared in 1982, the original paper had little detail of the methods by which the interface was reconstructed and fluxes calculated. This appendix describes the implementation of Youngs' VOF technique as used in this paper.

Given an Eulerian mesh with values of $C_{i,j}$ specified in each mesh cell, first-order upwind fluxes of C are first calculated for each cell. Each cell is revisited, and if it contains part of the interface ($0 < C_{i,j} < 1$), the *outward* Youngs fluxes must be calculated. First an estimate of β (the angle the interface makes with the x-axis) is made. There are several ways to determine β and here the gradient of C is used to calculate a surface normal from which β is determined in a straightforward manner.

Table V. Flux calculations for Youngs' method. Outward-pointing velocities are defined as being positive and flux calculation is not done for inward-pointing (negative) velocities

	Case I	Case II	Case III	Case IV
s_t	0	0	$C-\frac{1}{2}\cot\alpha$	$1-(2C\cot\alpha)^{1/2}$
s_r	$(2C\tan\alpha)^{1/2}$	$C+\frac{1}{2}\tan\alpha$	1	1
s_b	$(2C\cot\alpha)^{1/2}$	1	$C+\frac{1}{2}\cot\alpha$	1
s_l	0	$C-\frac{1}{2}\tan\alpha$	0	$1-(2C\tan\alpha)^{1/2}$
if $U_t>0$	if $U_t\delta t\leqslant(1-s_r)\delta y$ $F_t=0$ else $F_t=\frac{1}{2}[U_t\delta t-(1-s_r)\delta y]^2\cot\beta$	if $U_t\delta t\leqslant(1-s_r)\delta y$ $F_t=0$ else if $U_t\delta t\leqslant(1-s_l)\delta y$ $F_t=\frac{1}{2}[U_t\delta t-(1-sr)\delta y]^2\cot\beta$ else $F_t=U_t\delta t\delta x-(1-C)\delta x\delta y$	$F_t=U_t\delta t(s_t\delta x+\frac{1}{2}U_t\delta t\cot\beta)$	if $U_t\delta t\geqslant(1-s_l)\delta y$ $F_t=U_t\delta t\delta x-(1-C)\delta x\delta y$ else $F_t=U_t\delta t(s_t\delta x+\frac{1}{2}U_t\delta t\cot\beta)$
if $U_r>0$	if $U_r\delta t\geqslant s_b\delta x$ $F_r=C\delta x\delta y$ else $F_r=\frac{1}{2}U_r\delta t(2-U_r\delta t/s_b\delta x)s_r\delta y$	$F_r=U_r\delta t(s_r\delta y-\frac{1}{2}U_r\delta t\tan\beta)$	if $U_r\delta t\leqslant s_t\delta x$ $F_r=U_r\delta t\delta y$ else if $U_r\delta t\leqslant s_b\delta x$ $F_r=U_r\delta t\delta y-\frac{1}{2}(U_r\delta t-s_t\delta x)^2\tan\beta$ else $F_r=C\delta x\delta y$	if $U_r\delta t\leqslant s_t\delta x$ $F_r=U_r\delta t\delta y$ else $F_r=U_r\delta t\delta y-\frac{1}{2}\tan\beta(U_r\delta t-s_t\delta x)^2$
if $U_b>0$	if $U_b\delta t\geqslant s_r\delta y$ $F_b=C\delta x\delta y$ else $F_b=\frac{1}{2}U_b\delta t(2-U_b\delta t/s_r\delta y)s_b\delta x$	if $U_b\delta t\leqslant s_l\delta y$ $F_b=U_b\delta t\delta x$ else if $U_b\delta t\leqslant s_r\delta y$ $F_b=U_b\delta t\delta x-\frac{1}{2}(U_b\delta t-s_l\delta y)^2\cot\beta$ else $F_b=C\delta x\delta y$	$F_b=U_b\delta t(s_b\delta x-\frac{1}{2}U_b\delta t\cot\beta)$	if $U_b\delta t\leqslant s_l\delta y$ $F_b=U_b\delta t\delta x$ else $F_b=U_b\delta t\delta x-\frac{1}{2}(U_b\delta t-s_l\delta y)^2\cot\beta$
if $U_l>0$	if $U_l\delta t\leqslant(1-s_b)\delta x$ $F_l=0$ else $F_l=\frac{1}{2}[U_l\delta t-(1-s_b)\delta x]^2\tan\beta$	$F_l=U_l\delta t(\delta_l\delta y+\frac{1}{2}U_l\delta t\tan\beta)$	if $U_l\delta t\leqslant s_b\delta x$ $F_l=0$ else if $U_l\delta t\leqslant s_t\delta x$ $F_l=\frac{1}{2}[U_l\delta t-(1-s_b)\delta x]^2\tan\beta$ else $F_l=U_l\delta t\delta y-(1-C)\delta x\delta y$	if $U_l\delta t\geqslant(1-s_t)\delta x$ $F_l=U_l\delta t\delta y-(1-C)\delta x\delta y$ else $F_l=U_l\delta t(s_l\delta y+\frac{1}{2}U_l\delta t\tan\beta)$

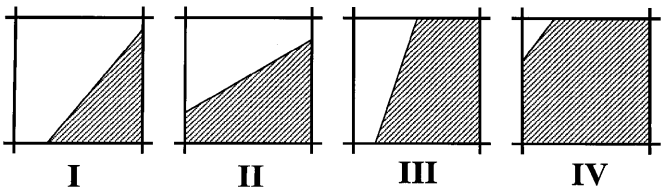

Figure 11. Four possible interface reconstructions for Youngs' VOF

The stencil used to calculate the normal influences the accuracy of the final advection scheme and it was found that the stencil used by Kothe *et al.*[21] gave the best results. For a uniform mesh,

$$n^x_{i,j} = \frac{1}{\delta x}(C_{i+1,j+1} + 2C_{i+1,j} + C_{i+1,j-1} - C_{i-1,j+1} - 2C_{i-1,j} - C_{i-1,j-1}), \tag{14}$$

$$n^y_{i,j} = \frac{1}{\delta y}(C_{i+1,j+1} + 2C_{i,j+1} + C_{i-1,j+1} - C_{i+1,j-1} - 2C_{i,j-1} - C_{i-1,j-1}), \tag{15}$$

from which

$$\beta = \tan^{-1}\left(\frac{-n^x}{n^y}\right) \quad (-\pi < \beta \leqslant \pi).$$

Defining the angle α to be

$$\alpha = \tan^{-1}\left(\frac{\delta x}{\delta y}\tan\beta\right) \quad (0 \leqslant \alpha \leqslant \pi/2),$$

the interface cell can be rotated in such a way that α lies in the range $0^\circ \leqslant \alpha \leqslant 90^\circ$. Once this rotation has been made, there are four possible interface configurations. They are shown in Figure 11 and labelled cases I–IV. The following logic determines the case

```
if α < π/4
  if C ⩽ ½ tan α
    case I
  else if C ⩽ 1·0 − ½ tan α
    case II
  else
    case IV
else
  if C ⩽ ½ cot α
    case I
  else if C ⩽ 1·0 − ½ cot α
    case III
  else
    case IV.
```

Once the case has been determined, the four side fractions can be calculated. The side fractions are the fractions of the top, right, bottom and left sides of the cell that lie within the fluid (s_t, s_r, s_b and s_l). Once these are known, the fluid fluxes (F_t, F_r, F_b, and F_l) can be calculated geometrically. Rather than describe all the possibilities in detail, a summary of the flux calculations is given in Table V.

Note that only outward fluxes from a cell are calculated and that outward velocities are defined as being positive.

REFERENCES

1. R. F. Warming and R. M. Beam, 'Upwind second order difference schemes and applications in aerodynamic flows', *AIAA J.* **14**, 1241–1249 (1976).
2. B. P. Leonard, 'A stable and accurate convective modelling procedure basedon quadratic upstream interpolation', *Comput. Methods Appl. Mech. Eng.*, **57**, 415–438 (1979).
3. J. P. Boris and D. L. Book, 'Flux-corrected transport I. SHASTA, a fluid transport algorithm that works', *J. Comput. Phys.*, **11**, 38–69 (1973).
4. S. T. Zalesak, 'Fully multi–dimensional flux corrected transport algorithms for fluid Flow', *J. Comput. Phys.*, **31**, 335–362 (1979).
5. P. Collela and P. R. Woodward, 'The piecewise parabolic method (PPM) for gas-dynamics simulations', *J. Comput. Phys.*, **54**, 174–201 (1984).
6. J. E. Welch, F. H. Harlow, J. P. Shannon and B. J. Daly, 'The MAC method. A computing technique for solving viscous, incompressible, transient fluid-flow problems involving free surfaces', *LASL Rep. LA-3425*, Los Alamos, NM, 1965.
7. R. K.-C. Chan, R. L. Street and T. Strelkoff, 'Computer study of finite-amplitude water waves', *Tech. Rep. 104*, Department of Civil Engineering, Stanford University, 1969.
8. B. D. Nichols and C. W. Hirt, *J. Comput. Phys.*, **12**, 234 (1973).
9. W. F. Noh and P. Woodward, 'SLIC (simple line interface calculation)',in A. I. van Dooren and P. J. Zandbergen (eds), *Lecture Notes in Physics*, Vol. 59, Springer, New York, 1976, pp. 330–340.
10. C. W. Hirt and B. D. Nichols, 'Volume of fluid (VOF) methods for the dynamicsof free boundaries', *J. Comput. Phys.*, **39**, 201–225 (1981).
11. D. L. Youngs, 'Time-dependent multi-material flow with large fluid distortion', in K. W. Morton and M. J. Baines (eds), *Numerical Methods for Fluid Dynamics*, Academic, New York, 1982, pp. 273–285.
12. S. O. Unverdi and G. Tryggvason, 'A front-tracking method for viscous, incompressible multifluid flow', *J. Comput. Phys.*, **100**, 25–37 (1992).
13. M. Sussman, P. Smereka and S. Osher, 'A level set approach for computingsolutions to incompressible two-phase flow', *J. Comput. Phys.*, **114**, 146–149 (1994).
14. W. J. Rider and D. B. Kothe, 'Stretching and tearing interface tracking methods', *AIAA Paper 95-1717*, 1995.
15. W. J. Rider, D. B. Kothe, S. J. Mosso, J. H. Cerutti and J. I. Hochstein, 'Accurate solution algorithms for incompressible multiphase flows', *AIAA Paper 95–0699*, 1995.
16. C. W. Hirt, 'Heuristic stability theory for finite-difference equations', *J. Comput. Phys.*, **2**, 339–355 (1968).
17. B. LaFaurie, C. Nardone, R. Scardovelli and G. Zanetti, 'Modelling merging and fragmentation in multiphase flows with SURFER', *J. Comput. Phys.*, **113**, 134–147 (1994).
18. A. J. Chorin, 'Flame advection and propagation algorithms', *J. Comput. Phys.*, **35**, 1–11 (1980).
19. N. Ashgriz and J. Y. Poo, 'FLAIR: flux line-segment model for advection and interface reconstruction', *J. Comput. Phys.*, **93**, 449–468 (1991).
20. D. E. Morton, M. J. Rudman and J.-L. Liow, 'Numerical simulation of splashing phenomena', *Proc Ninth Int. Conf. on Numerical Methods in Laminar and Turbulent Flow*, Atlanta, GA, July 1995.
21. D. B. Kothe, R. C. Mjolsness and M. D. Torrey, 'RIPPLE: a computer program for incompressible flows with free surfaces', *Los Alamos National Laboratory Rep. LA-12007–MS*, 1991.

