ELSEVIER

Chemical Engineering Journal 141 (2008) 204–221

Chemical Engineering Journal

www.elsevier.com/locate/cej

Volume of fluid methods for immiscible-fluid and free-surface flows

Vinay R. Gopala*, Berend G.M. van Wachem

Department of Applied Mechanics, Chalmers University of Technology, SE-41296 Gothenburg, Sweden

Received 14 May 2007; received in revised form 10 December 2007; accepted 28 December 2007

Abstract

This article reviews and analyzes a number of numerical methods to track interfaces in multiphase flows. Several interface tracking methods can be found in literature: the level-set method, the marker particle method, the front tracking method and the volume of fluid method (VOF) to name a few. The volume of fluid method has an advantage of being conceptually simple, reasonably accurate and phenomena such as interface breakup and coalescence are inherently included. Over the years a number of different techniques to implement the VOF method have been devised.

This article gives a basic introduction to the VOF method and focuses on four VOF methods: flux-corrected transport (FCT) by Boris et al. [J.P. Boris, D.L. Book, Flux-corrected transport. I: SHASTA, a fluid transport algorithm that works, J. Comput. Phys. 11 (1973) 38–69], Lagrangian piecewise linear interface construction (L-PLIC) by van Wachem and Schouten [B.G.M. van Wachem, J.C. Schouten, Experimental validation of 3-d Lagrangian VOF model: bubble shape and rise velocity, AIChE 48 (12) (2002) 2744–2753], Compressive interface capturing scheme for arbitrary meshes (CICSAM) by Ubbink [O. Ubbink, Numerical prediction of two fluid systems with sharp interfaces, Ph.D. Thesis, Imperial College of Science, Technology and Medicine, 1997] and inter-gamma scheme by Jasak and Weller [H. Jasak, H.G. Weller, Interface-tracking capabilities of the InterGamma differencing scheme, Technical Report, Imperial College, University of London, 1995]. A detailed description of these schemes is given and implemented into an in-house fully coupled solver. Further, the performance of these schemes is examined employing a number of tests to analyze their strengths and weaknesses. Their advantages and limitations are discussed.

Keywords: Volume of fluid method (VOF); Interfaces; Free surfaces; Advection schemes

1. Introduction

Interfacial multiphase flows are frequently encountered both in nature and industries. Processes such as extraction, chemical reaction, mass-transfer, separation, etc., involve interfacial flows. To understand the basic hydrodynamic phenomena associated with such processes requires a proper and sharp definition of the interface between two phases. These phenomena include bubble transport, formation, breakup, coalescence, etc.

A detailed computation of immiscible-fluid and free-surface flows requires an accurate representation of the interface separating the two fluids. Immiscible-fluid flows are commonly encountered in nature as well as in industries. The applications include processes involving separation, extraction, mixing and chemical reactions. Free-surface flows such as water waves and splashing droplets are encountered in nature and industrial processes. These flow problems include phenomena like fluid coalescence and breakup which further increases the need for an accurate and sharp interface definition. A number of techniques to track the interface have been developed in the last few decades. The most important techniques are shown in Table 1. The existing methods for the computation of free surfaces and fluid interfaces can be classified into two groups (Fig. 1): (a) surface methods (surface fitting) and (b) volume methods (surface capturing).

1.1. Surface methods

With surface methods, the interface is represented by special marker points. Interpolation is used to approximate the points between these points, usually using a piecewise polynomial. The advantage of this approach is that the interface position is known throughout the flow field and remains sharp as it is advected across the domain. This enables the accurate calculation of the interface curvature which is needed for the inclusion of the surface tension force. Limitations arise while simulating coalescence and breakup of the interface surface, as the particles might tend to either move apart or very close to each other

* Corresponding author. Tel.: +46 317723589; fax: +46 31180976.
E-mail address: vinay@chalmers.se (V.R. Gopala).

1385-8947/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cej.2007.12.035

Table 1
Overview of interface tracking techniques

Method	Advantages	Disadvantages
Front tracking	Extremely accurate Robust Accounts for substantial topology changes in interface	Mapping of interface mesh onto Eulerian mesh Dynamic re-meshing required Merging and breakage of interfaces requires sub-grid model
Level set	Conceptually simple Easy to implement	Limited accuracy Loss of mass (volume)
Shock capturing	Straightforward implementation Abundance of advection schemes is available	Numerically diffusive Fine grids required Limited or small discontinuities
Marker particle	Extremely accurate Robust Accounts for substantial topology changes in interface	Computationally expensive Re-distribution of marker particles required
SLIC VOF	Conceptually simple Straightforward extension to three dimensions Merging and breakage of interfaces occurs automatically	Numerically diffusive Limited accuracy
PLIC VOF	Relatively simple and accurate Merging and breakage of interface occurs automatically	Difficult to implement in three dimensions Extension to boundary fitted grids very difficult
Compressive VOF	Relatively simple and accurate Easily adaptable to boundary fitted grids Merging and breakage of interface occurs automatically	Requires very low Courant numbers else becomes inaccurate or unstable

leading to lower resolution of the interface. Several surface methods exist, two such methods are explained below:

Front tracking method: In front tracking methods (Unverdi and Tryggvason [15]), the interface is tracked explicitly on a fixed Eulerian mesh by marking the interface with a set of connected massless marker particles. The local velocities are used to advect these massless particles in a Lagrangian manner. The method is sensitive to the spacing between the marker particles, i.e. when the particles are far apart, the interface is not well resolved and when they are too close, the curvature is over-estimated. Therefore it is necessary to add or delete marker particles dynamically. Also, difficulties arise when multiple interfaces interact with each other as in coalescence and breakup requiring a proper sub-grid model.

Level-set method: In level-set methods (Osher and Sethian [10]), the interface is defined as a zero level set of a distance function from the interface. To distinguish between the two fluids on either side of the interface a negative sign is attached to the distance function for one of the fluids. The distance function γ is a scalar property and is advected with the local fluid velocity by solving the scalar advection equation:

$$\frac{\partial \gamma}{\partial t} + U_i \frac{\partial \gamma}{\partial x_i} = 0 \qquad (1)$$

The level-set methods are conceptually simple and relatively easy to implement yielding accurate results when the interface is advected parallel to one of the co-ordinate axis. However, in flow fields with appreciable vorticity or in cases where the interface is significantly deformed, level-set methods suffer from loss of mass.

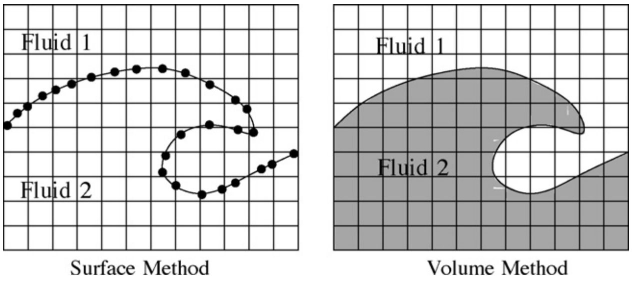


Fig. 1. Different methods representing interface.

1.2. Volume methods

In volume methods, the fluids on either side of the interface are marked by either massless particles or an indicator function. Thus the exact position of the interface is not known explicitly and special techniques are needed to reconstruct the well-defined interface, which is one of the main drawbacks of this technique. A number of volume methods exist, two such volume methods are explained here.

Marker particle methods: In the marker and cell (MAC) method of Harlow and Welch [3] marker particles are scattered initially to identify each material region in the calculation. These particles are transported in a Lagrangian manner along with the materials. Their presence in a computational cell indicates the presence of the marked material. The material boundary is reconstructed using the marker particle densities in the mixed cells with marker particles of two or more materials. Marker particle methods are extremely accurate and robust and can be used successfully to predict the topology of an interface subjected to considerable shear and vorticity in the fluids sharing the interface. However, this method is computationally expensive due to the requirement of many particles, especially in three dimensions. Moreover, difficulties arise when the interface stretches considerably which requires the addition of fresh marker particles during the flow simulation.

Volume of fluid method (*VOF*): A scalar indicator function between zero and one, known as volume fraction is used to distinguish between two different fluids. In this study four different volume of fluid schemes are implemented into a fully coupled in-house finite volume, boundary fitted code. The next section gives an introduction to the volume of fluid method and the schemes implemented are explained in detail. Further, the performance of these schemes is reviewed with the help of various test cases.

2. Volume of fluid method

The volume of fluid method was first proposed by Hirt and Nichols [4]. In the volume of fluid method, the flow equations are volume averaged directly to obtain single set of equations and the interface is tracked using a phase indicator function γ (also known as color function or volume fraction) which is defined as:

- $\gamma = 1 \Rightarrow$ control volume is filled only with phase 1
- $\gamma = 0 \Rightarrow$ control volume is filled only with phase 2
- $0 < \gamma < 1 \Rightarrow$ interface present

The flow equations are volume averaged using an averaging volume smaller than the bubbles/drops used in the simulations. Considering only two phases, without mass exchange, and volume averaging the mass and momentum equations, following three cases are encountered as shown in Fig. 2:

Case 1: The averaging picks out phase 1.

- Mass conservation

$$\frac{\partial\langle\rho_1\rangle}{\partial t} + \nabla\cdot\langle\rho_1\boldsymbol{u}_1\rangle = 0. \tag{2}$$

- Momentum balance

$$\frac{\partial}{\partial t}\langle\rho_1\boldsymbol{u}_1\rangle + \nabla\cdot\langle\rho_1\boldsymbol{u}_1\boldsymbol{u}_1\rangle = \nabla\cdot\langle\boldsymbol{T}_1\rangle + \langle\rho_1\boldsymbol{g}\rangle. \tag{3}$$

Case 2: The averaging picks out phase 2.

- Mass conservation

$$\frac{\partial\langle\rho_2\rangle}{\partial t} + \nabla\cdot\langle\rho_2\boldsymbol{u}_2\rangle = 0. \tag{4}$$

- Momentum balance

$$\frac{\partial}{\partial t}\langle\rho_2\boldsymbol{u}_2\rangle + \nabla\cdot\langle\rho_2\boldsymbol{u}_2\boldsymbol{u}_2\rangle = \nabla\cdot\langle\boldsymbol{T}_2\rangle + \langle\rho_2\boldsymbol{g}\rangle. \tag{5}$$

Case 3: The averaging picks out a piece of the interface and both the phases.

- Mass conservation

$$\frac{\partial}{\partial t}\langle\gamma\rho_1\rangle + \nabla\cdot\langle\gamma\rho_1\boldsymbol{u}_1\rangle = 0, \tag{6}$$

$$\frac{\partial}{\partial t}\langle(1-\gamma)\rho_2\rangle + \nabla\cdot\langle(1-\gamma)\rho_2\boldsymbol{u}_2\rangle = 0. \tag{7}$$

- Momentum balances

$$\begin{aligned}&\frac{\partial}{\partial t}\langle\gamma\rho_1\boldsymbol{u}_1\rangle + \nabla\cdot\langle\gamma\rho_1\boldsymbol{u}_1\boldsymbol{u}_1\rangle\\ &\quad = \nabla\cdot\langle\gamma\boldsymbol{T}_1\rangle + \langle\gamma\rho_1\boldsymbol{g}\rangle + \frac{1}{V}\int_{A_I}\boldsymbol{T}_1\cdot\boldsymbol{n}_{I_{12}}\,\mathrm{d}A,\end{aligned} \tag{8}$$

$$\begin{aligned}&\frac{\partial}{\partial t}\langle(1-\gamma)\rho_2\boldsymbol{u}_2\rangle + \nabla\cdot\langle(1-\gamma)\rho_2\boldsymbol{u}_2\boldsymbol{u}_2\rangle\\ &\quad = \nabla\cdot\langle(1-\gamma)\boldsymbol{T}_2\rangle + \langle(1-\gamma)\rho_2\boldsymbol{g}\rangle\\ &\qquad + \frac{1}{V}\int_{A_I}\boldsymbol{T}_2\cdot\boldsymbol{n}_{I_{21}}\,\mathrm{d}A.\end{aligned} \tag{9}$$

- Jump condition

$$\begin{aligned}&\frac{1}{V}\int_{A_I}\left[-\boldsymbol{T}_1\cdot\boldsymbol{n}_{I_{12}} - \boldsymbol{T}_2\cdot\boldsymbol{n}_{I_{21}}\right]\mathrm{d}A\\ &\quad = -\frac{1}{V}\int_{A_I} m_{12}^{\sigma}\,\mathrm{d}A.\end{aligned} \tag{10}$$

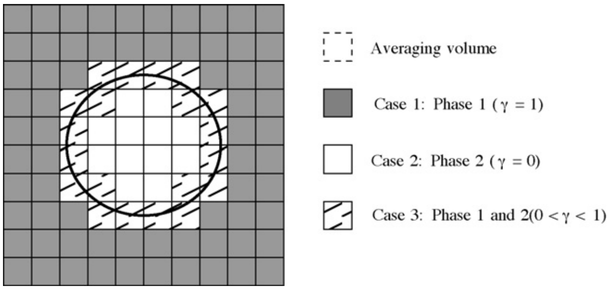


Fig. 2. Averaging volume compared with the bubble/drop volume.

In order to obtain single set of equations, the following averaged variables need to be introduced:

$$\rho = \langle\gamma\rho_1 + (1-\gamma)\rho_2\rangle \tag{11}$$

$$\boldsymbol{T} = \langle\gamma\boldsymbol{T}_1 + (1-\gamma)\boldsymbol{T}_2\rangle \tag{12}$$

Using these new variables and assuming that the velocity of the two phases is continuous across the interface, the mass and momentum equations can be written as:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\boldsymbol{u}) = 0, \tag{13}$$

$$\frac{\partial}{\partial t}(\rho\boldsymbol{u}) + \nabla\cdot(\rho\boldsymbol{u}\boldsymbol{u}) = \nabla\cdot\boldsymbol{T} + \rho\boldsymbol{g} + \boldsymbol{f}_I. \tag{14}$$

The last term on the RHS of Eq. (14) represents the surface tension force.

The scalar γ being the property of the fluid (volume fraction) with which it moves, it is evolution is governed by the simple advection equation:

$$\frac{\partial\gamma}{\partial t} + \frac{\partial U_i\gamma}{\partial x_i} = 0. \tag{15}$$

One of the critical issues with the VOF method is the discretization of advection term in Eq. (15). Lower order schemes like the

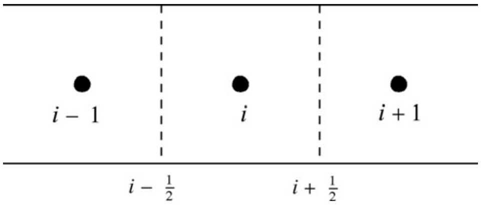


Fig. 3. One-dimensional mesh arrangement for FCT.

first order upwind method smear the interface due to numerical diffusion and higher order schemes are unstable and result in numerical oscillations. Thus, it is necessary to derive advection schemes that can keep the interface sharp and produce monotonic profiles of the color function. Over the years several volume advection techniques for finite volume and finite difference meshes have been proposed by many researchers: Noh and Woodward's [9] SLIC (Simple Line Interface Calculation), Hirt and Nichols' [4] Donor–Acceptor Scheme, Youngs' method [17], flux-corrected transport (FCT) by Boris et al. [1], Ubbink's [14] Compressive interface capturing scheme for arbitrary meshes (CICSAM), etc. This paper focuses on the following volume tracking methods:

- Flux-corrected transport method by Boris et al. [1],
- Lagrangian-PLIC method by van Wachem and Schouten [16],
- CICSAM by Ubbink [14] and
- Inter-gamma compressive scheme by Jasak and Weller [5].

In the following sections, these methods are explained in detail.

2.1. Flux-corrected transport

FCT is based on the idea that 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. The idea of adjusting fluxes calculated with a higher order (non-monotonic) advection scheme to improve the monotonicity of the final result was introduced by Boris et al. [1] and was generalized and extended to multi-dimensions by Zalesak [18].

The method involves several stages of calculation. First, an intermediate value of γ, namely γ^*, is determined using a lower order monotonic (and hence diffusive) advection scheme. The scheme for solving the one-dimensional version of Eq. (15) (for mesh cell i as shown in Fig. 3) is symbolically written as

$$\gamma_i^* = \gamma_i^n - \frac{1}{\delta x}(F^{\mathrm{L}}_{(i+1/2)} - F^{\mathrm{L}}_{(i-1/2)}), \tag{16}$$

where F^{L} represents the lower order flux. The flux F_f at the face f is defined as:

$$F_f = U_f \delta t \gamma_f \tag{17}$$

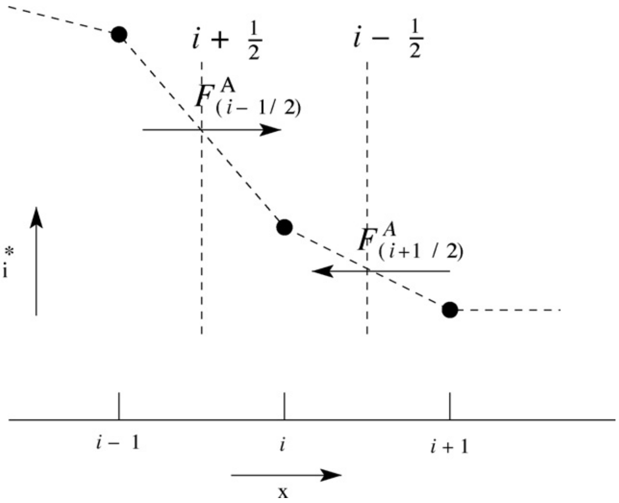


Fig. 4. One-dimensional view of the transported and diffused profile of γ_i^*, showing the two anti-diffusive fluxes.

Further an anti-diffusive flux (F^{A}) is defined that attempts to correct the numerical diffusion resulting from the lower order scheme. An estimate of the anti-diffusive fluxes is given by the difference between the higher and lower order flux approximations:

$$F^{\mathrm{A}}_{(i+1/2)} = F^{\mathrm{H}}_{(i+1/2)} - F^{\mathrm{L}}_{(i+1/2)}. \tag{18}$$

Application of the entire anti-diffusive flux results in the unstable higher order flux being used, thus correction factors (limiters) q are introduced that limit the anti-diffusive fluxes. Detailed procedure used to limit the fluxes as described by Zalesak [18] is presented in the next section. The final step of flux-corrected transport algorithm is to apply the anti-diffusive fluxes with the correction factors and obtain the values of the color function at the new time:

$$\gamma_i^{n+1} = \gamma_i^* - \frac{\left(q_{(i+1/2)}F^{\mathrm{A}}_{(i+1/2)} - q_{(i-1/2)}F^{\mathrm{A}}_{(i-1/2)}\right)}{\delta x}. \tag{19}$$

The above FCT algorithm is extended to multi-dimensions by direction-split implementation. Rudman [12] has shown that the direction-split FCT gives superior results compared to multi-dimensional extension proposed by Ref. [18]. Unfortunately, the method does not work well for flows with vorticity, the method becomes quite diffusive in such cases.

2.1.1. Limiter estimation

As described in the previous section, using the entire anti-diffusive flux results in numerical oscillations, thus limiters q should be introduced that limit the anti-diffusive fluxes. This section presents Zalesak's [18] flux limiting algorithm in one dimension. The limiters 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 γ^n and γ^* in cell i and its two neighbors, $i-1$ and $i+1$. Referring to Fig. 4, one seeks to limit the anti-diffusive flux $F^{\mathrm{A}}_{(i+1/2)}$ such that

$$F^{\mathrm{C}}_{(i+1/2)} = q_{(i+1/2)}F^{\mathrm{A}}_{(i+1/2)}, 0 \leq q_{(i+1/2)} \leq 1 \tag{20}$$

and such that $F^{C}_{(i+1/2)}$ acting in concert with $F^{C}_{(i-1/2)}$ will not allow

$$\gamma_i^{n+1} = \gamma_i^* - \frac{(F^{C}_{(i+1/2)} - F^{C}_{(i-1/2)})}{\delta x} \tag{21}$$

to exceed some maximum value $\gamma_i^{\max}$ nor fall below some minimum value $\gamma_i^{\min}$. Defining three new quantities:

$$\begin{aligned} P_i^+ &= \text{the sum of all anti-diffusive fluxes into grid point } i \\ &= \max(0, F^{A}_{(i-1/2)}) - \min(0, F^{A}_{(i+1/2)}) \end{aligned} \tag{22}$$

$$Q_i^+ = \left(\gamma_i^{\max} - \gamma_i^*\right) \Delta x_i \tag{23}$$

$$R_i^+ = \begin{cases} \min(1, \dfrac{Q_i^+}{P_i^+}) & \text{if} \quad P_i^+ > 0 \\ 0 & \text{if} \quad P_i^+ = 0 \end{cases} \tag{24}$$

Assuming that $\gamma_i^{\max} \geq \gamma_i^*$, all three of the above quantities are positive and R_i^+ represents the least upper bound on the fraction which must multiply all anti-diffusive fluxes into grid point i to guarantee no overshoot at grid point i. Similarly defining three corresponding quantities:

$$\begin{aligned} P_i^- &= \text{the sum of all anti-diffusive fluxes away} \\ &\quad \text{from grid point } i \\ &= \max(0, F^{A}_{(i+1/2)}) - \min(0, F^{A}_{(i-1/2)}) \end{aligned} \tag{25}$$

$$Q_i^- = (\gamma_i^* - \gamma_i^{\min}) \Delta x_i \tag{26}$$

$$R_i^- = \begin{cases} \min\left(1, \dfrac{Q_i^-}{P_i^-}\right) & \text{if} \quad P_i^- > 0 \\ 0 & \text{if} \quad P_i^- = 0 \end{cases} \tag{27}$$

Again assuming that $\gamma_i^{\min} \leq \gamma_i^*$, R_i^- represents the least upper bound on the fraction which must multiply all anti-diffusive fluxes away from grid point i to guarantee no undershoot at grid point i.

All anti-diffusive fluxes are directed away from one grid point and into an adjacent one. Limiting will therefore take place with respect to undershoots for the former and with respect to overshoots for the latter. A guarantee that neither event comes to pass demands taking a minimum:

$$q_{(i+1/2)} = \begin{cases} \min(R^+_{(i+1)}, R_i^-) & \text{if} F^{A}_{(i+1/2)} \geq 0 \\ \min(R_i^+, R^-_{i+1}) & \text{if} F^{A}_{(i+1/2)} < 0 \end{cases} \tag{28}$$

Finally, to determine $\gamma_i^{\max}$ and $\gamma_i^{\min}$ present in Eqs. (23) and (26), Zalesak [18] shows that the following choice performed better:

$$\begin{aligned} \gamma_i^a &= \max(\gamma_i^n, \gamma_i^*) \\ \gamma_i^{\max} &= \max(\gamma_{i-1}^a, \gamma_i^a, \gamma_{i+1}^a) \end{aligned} \tag{29}$$

$$\begin{aligned} \gamma_i^b &= \min(\gamma_i^n, \gamma_i^*) \\ \gamma_i^{\min} &= \min(\gamma_{i-1}^b, \gamma_i^b, \gamma_{i+1}^b). \end{aligned} \tag{30}$$

This choice allows one to look back to the previous time step for upper and lower bounds on γ_i^{n+1}.

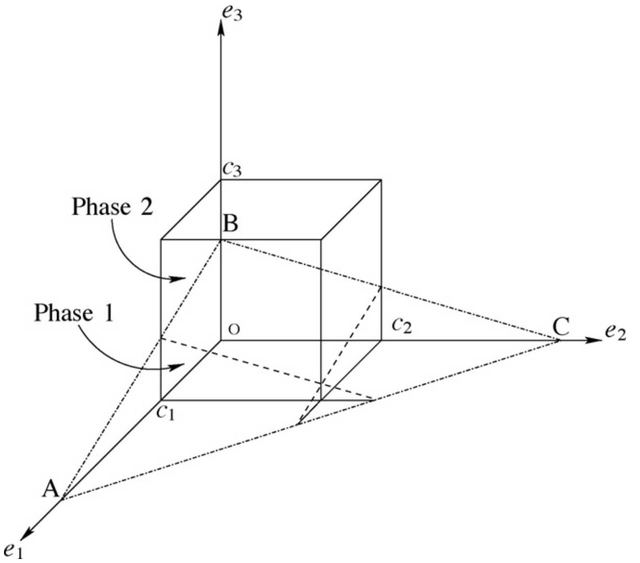


Fig. 5. Volume of the computational cell cut by interface ABC.

2.2. Lagrangian PLIC

This Lagrangian volume of fluid method based on piecewise linear interface construction (PLIC) is proposed by van Wachem and Schouten [16]. Fig. 5 shows a computational cell with an interface separating the two fluids, i.e. phases 1 and 2. The interface itself is defined by the local volume fraction of one of the two fluids and the estimated normal $\boldsymbol{n}$. This interface between the two phases is propagated by the local fluid flow along the interface. This method involves two steps:

Reconstruction of the interface: As shown in Fig. 5 e_1, e_2 and e_3 are the three Cartesian directions and c_1, c_2 and c_3 are the lengths of the orthogonal computational grid cell. The general equation for a plane in three dimensions is given by the equation:

$$n_1 x_1 + n_2 x_2 + n_3 x_3 = \alpha, \tag{31}$$

where α is the shortest distance from the plane to the origin, n_i are the normals to the surface, and x_i is the coordinate in the direction of e_i. Now, the volume of the large tetrahedron below the interface ABC (Fig. 5) is given by

$$\frac{\alpha^3}{6 n_1 n_2 n_3} \tag{32}$$

To obtain only the volume lying under the interface within the computational cell one needs to subtract the three volumes of tetrahedra that protrude outside of the original tetrahedron. The magnitudes of these volumes is $(1 - n_i x_i/\alpha)^3$, if $\alpha > n_i x_i$. However, this results in the volume of the small tetrahedron in front of the figure being subtracted multiple times, hence this volume should be added again. The magnitude of this volume is $(1 - n_i c_i/\alpha - n_j c_j/\alpha)^3$, where $i \neq j$, and only if $\alpha > (n_i c_i + n_j c_j)$. Thus, the equation for the total volume lying under the interface within the computational cell can be written as

$$V=\frac{\alpha^3}{6n_1n_2n_3}\left\{1-\sum_{i=1}^{3}H(\alpha-n_ic_i)\left(1-\frac{n_ic_i}{\alpha}\right)^3\right.$$
$$\left.+\sum_{i=1}^{3}H(\alpha-\alpha_{\max}+n_ic_i)\left(1-\frac{\alpha_{\max}}{\alpha}+\frac{n_ic_i}{\alpha}\right)\right\},\qquad(33)$$

where H is the Heaviside step function, and

$$\alpha_{\max}=n_1c_1+n_2c_2+n_3c_3.\qquad(34)$$

In this problem, we not only need the forward Eq. (33) but also the inverse, to obtain the value of α with a known volume. Since Eq. (33) is monotonic in α, solving α is straightforward, except in cases for any of the n_i being very close to zero. To solve the inverse problem, Newton–Raphson method is used as the derivative of Eq. (33) can be determined.

Lagrangian propagation of the interface: The Lagrangian propagation of the interface can be best described by how Eq. (31) describing the interface changes due to the movement of the flow. This results in two contributions:

1. Change in the values of α and n_i due to the fluid flow and lead to the movement of the interface within the computational cell.
2. Change in the values of c_i due to the movement of the rectangular sides of the volume, thus shifting the origin to the interface.

The first contribution is found by updating Eq. (31) with the new values of α and n_i which are in turn found by integrating the local fluid velocity over the interface. In one direction, this local velocity is written as

$$u_i(x)=U^{\rm l}\left(1-\frac{x_i}{c_i}\right)+U^{\rm r}\frac{x_i}{c_i},\qquad(35)$$

where $U^{\rm l}$ is the fluid velocity at the left face of the computational cell, $U^{\rm r}$ is the fluid velocity at the right face of the computation cell, and x_i is the coordinate into the computational cell, being zero at the left edge and c_i at the right. Thus, the new coordinate, x_i^n, becomes

$$x_i^{(n)}=x_i+u_i(x_i)\,\Delta t=\left(1+\left\{\frac{U^{\rm l}-U^{\rm r}}{c_i}\right\}\Delta t\right)x_i+U^{\rm r}\,\Delta t.\qquad(36)$$

Now, substituting Eq. (36) into Eq. (31) one can obtain the new values for α and n_i. Hence,

$$n_i^{(n)}=\frac{n_i}{1+(U^{\rm l}-U^{\rm r}/c_i)\,\Delta t}\qquad(37)$$

$$\alpha^{(n)}=\alpha+\frac{n_iU^{\rm r}\,\Delta t}{1+(U^{\rm l}-U^{\rm r}/c_i)\,\Delta t}.\qquad(38)$$

This process is done sequentially for all spatial directions. The second contribution occurs both on the left and right-hand sides of each direction due to protrusion of the interface into neighboring cells. At the left face, the interface protrudes into the left-hand side cell if $U_{\rm left}<0$. The volume shifted from the current cell into the left-hand cell is determined by considering the intersection of a cell with dimensions $-U_{\rm left}\,\Delta t$, c_j and c_k, where $i\neq j\neq k$, with an interface at which α is shifted by $-n_iU_{\rm left}\,\Delta t$. Similarly at the right face, the interface protrudes into the right-hand side cell if $U_{\rm right}>0$ and $\alpha/n_i>c_i$, as the interface has to intersect with the right face of the computational cell. The volume shifted from the current cell into the right-hand side cell is determined by considering the intersection of a cell with dimensions $U_{\rm right}\,\Delta t$, c_j and c_k, where $i\neq j\neq k$, with an interface intersecting this cell of which α is shifted by $-n_ic_i$. Moving these volumes is done sequentially for all spatial directions.

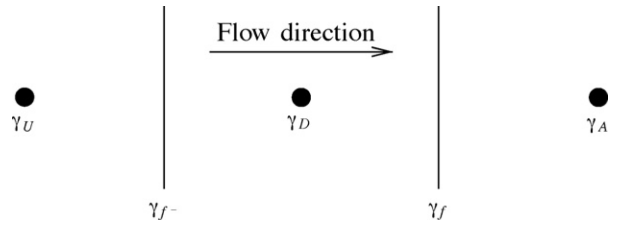


Fig. 6. One-dimensional control volume.

2.3. CICSAM

CICSAM by Ubbink [14] is a high-resolution differencing scheme based on the idea of the donor–acceptor flux approximation using normalized variable diagram (NVD) of Leonard [6]. The schematic representation of a one-dimensional control volume and its neighbors is shown in Fig. 6. The center cell known as the donor cell is represented by subscript D, has two neighbors known as the acceptor cell, represented by subscript A and the upwind cell, represented by the subscript U. The flow direction is used to determine the location of the neighbors. The cell receiving fluid from the donor cell (D) is the acceptor cell (A) and the other neighbor to the donor cell is the upwind cell (U). The face between the donor and the acceptor cells, referred with a subscript f, is the face under consideration.

2.3.1. Normalized variable diagram

Consider the variation of a convected scalar $\gamma(x, y, z)$ along a direction normal to a control volume face as shown in Fig. 6. The normalized variable is defined as [6]:

$$\tilde{\gamma}=\frac{\gamma-\gamma_{\rm U}}{\gamma_{\rm A}-\gamma_{\rm U}}.\qquad(39)$$

Note particularly that, in terms of normalized variables, $\tilde{\gamma}_{\rm A}=1$ and $\tilde{\gamma}_{\rm U}=0$. For example, for QUICK (Quadratic Upwind Interpolation for Convection Kinematics), on an uniform grid, the convected face variable is

$$\gamma_f=\frac{1}{2}(\gamma_{\rm A}+\gamma_{\rm D})-\frac{1}{8}(\gamma_{\rm A}-2\gamma_{\rm D}+\gamma_{\rm U}),\qquad(40)$$

so, in terms of normalized variables,

$$\tilde{\gamma}_f=\frac{1}{2}(1+\tilde{\gamma}_{\rm D})-\frac{1}{8}(1-2\tilde{\gamma}_{\rm D}),\qquad(41)$$

Table 2
Convection schemes in their original and normalized variables

Schemes	Original variables	Normalized variables
First order upwind	$\gamma_f = \gamma_{\text{D}}$	$\tilde{\gamma}_f = \tilde{\gamma}_{\text{D}}$
First order downwind	$\gamma_f = \gamma_{\text{A}}$	$\tilde{\gamma}_f = \tilde{\gamma}_{\text{A}} = 1$
Second order upwind	$\gamma_f = \frac{3}{2}\gamma_{\text{D}} - \frac{1}{2}\gamma_{\text{U}}$	$\tilde{\gamma}_f = \frac{3}{2}\tilde{\gamma}_{\text{D}}$
Central differencing	$\gamma_f = \frac{1}{2}(\gamma_{\text{A}} + \gamma_{\text{D}})$	$\tilde{\gamma}_f = 0.75 + 0.5(\tilde{\gamma}_{\text{D}} - 0.5)$

or, more conveniently,

$$\tilde{\gamma}_f = 0.75 + 0.75(\tilde{\gamma}_{\text{D}} - 0.5). \tag{42}$$

It should be clear that if γ_f is a function of γ_{D}, γ_{A} and γ_{U}, then the normalized variable $\tilde{\gamma}_f$ is only a function of $\tilde{\gamma}_{\text{D}}$ (since $\tilde{\gamma}_{\text{A}} = 1$ and $\tilde{\gamma}_{\text{U}} = 0$). This is the basis of the normalized variable diagram, which is a plot of the functional relationship between the normalized convected face value $\tilde{\gamma}_f$ and the normalized adjacent donor node value $\tilde{\gamma}_{\text{D}}$.

Eq. (42) shows that, for QUICK, the normalized variable diagram is a straight line passing through (0.5, 0.75) with a slope of 0.75. Table 2 shows few well-known schemes in their original and normalized variables. The linear NVD characteristics of various schemes are shown in Fig. 7.

Leonard [6] shows that linear NVD characteristics which pass through the second quadrant may produce unphysical oscillations in steady one-dimensional convection. This is well known for central differencing scheme and may also occur to some extent with QUICK under high-convection conditions. From Fig. 7 one can observe that these two characteristics indeed pass through the second quadrant. It is also found that the characteristics which pass through the fourth quadrant (i.e. below **O**) are artificially diffusive. Numerical experimentation has shown that NVD characteristics that pass above **P** are oscillatory in two dimensions and characteristics that pass below **P** are artificially diffusive. Thus, to avoid oscillations without being artificially diffusive, the desired NVD characteristic should pass through **O** and **P**.

2.3.2. *Convection boundedness criteria (CBC)*

Gaskell and Lau [2] suggested a convection boundedness criteria for an implicit differencing scheme using the normalized variable as:

$$\begin{aligned} &\tilde{\gamma}_f = \tilde{\gamma}_{\text{D}} && \text{for } \tilde{\gamma}_{\text{D}} < 0 \text{ or } \tilde{\gamma}_{\text{D}} > 1 \\ &\tilde{\gamma}_{\text{D}} \le \tilde{\gamma}_f \le 1 && \text{for } 0 \le \tilde{\gamma}_{\text{D}} \le 1 \end{aligned} \tag{43}$$

which present bounds on γ_f for which an implicit differencing scheme always preserves the local boundedness criteria. In Fig. 7, CBC represents the shaded area with the line representing the upwind differencing scheme. Further, Leonard [7] modified CBC for explicit flow calculations as:

$$\begin{aligned} &\tilde{\gamma}_f = \tilde{\gamma}_{\text{D}} && \text{for } \tilde{\gamma}_{\text{D}} < 0 \text{ or } \tilde{\gamma}_{\text{D}} > 1 \\ &\tilde{\gamma}_{\text{D}} \le \tilde{\gamma}_f \le \min\left\{1, \frac{\tilde{\gamma}_{\text{D}}}{c_f}\right\} && \text{for } 0 \le \tilde{\gamma}_{\text{D}} \le 1 \end{aligned} \tag{44}$$

where c_f is an arbitrary Courant number at the face *f*, defined as $c_f = |\boldsymbol{u}_f \delta t|/\delta x$.

Fig. 8 shows the CBC for the explicit implementation. Any scheme that falls within the upper left triangle in Fig. 8 will be unbounded. Any scheme that is nearer to the upper bound is more compressive and nearer to the line representing upwind differencing scheme is more diffusive. Thus, normalized variable diagram shown in Fig. 8 can be used to evaluate the degree of boundedness and diffusiveness of the any scheme without actually implementing them. Fig. 8 also shows that when Courant number increases, the bounded area diminishes and when Courant number is greater than one all the schemes become unbounded with an explicit implementation.

Leonard's [7] Hyper-C differencing scheme follows the upper bound of the CBC for explicit flow calculations, thus making it very compressive. The normalized face value for Hyper-C differencing scheme is defined as:

$$\tilde{\gamma}_{f_{\text{CBC}}} = \begin{cases} \min\left\{1, \dfrac{\tilde{\gamma}_{\text{D}}}{c_f}\right\} & \text{when} \quad 0 \le \tilde{\gamma}_{\text{D}} \le 1 \\ \tilde{\gamma}_{\text{D}} & \text{when} \quad \tilde{\gamma}_{\text{D}} < 0, \tilde{\gamma}_{\text{D}} > 1 \end{cases} \tag{45}$$

The Hyper-C scheme on its own is not suitable for modeling interfacial flow, because it tends to wrinkle the interface [4]. This is because downwinding tends to compress any gradient

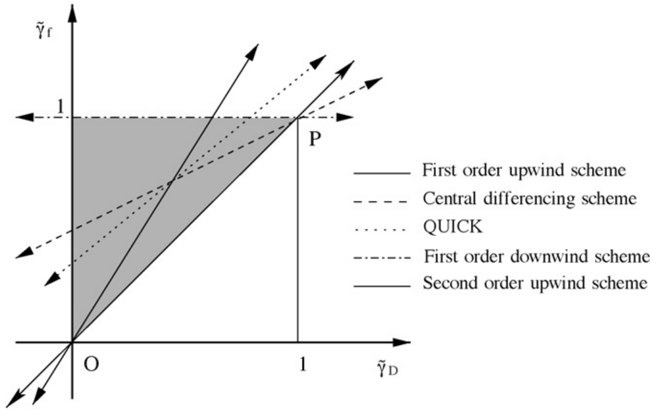

$\tilde{\gamma}_f$
1
P
O
1
$\tilde{\gamma}_D$
First order upwind scheme
Central differencing scheme
QUICK
First order downwind scheme
Second order upwind scheme

Fig. 7. The convection boundedness criteria for explicit flow calculations.

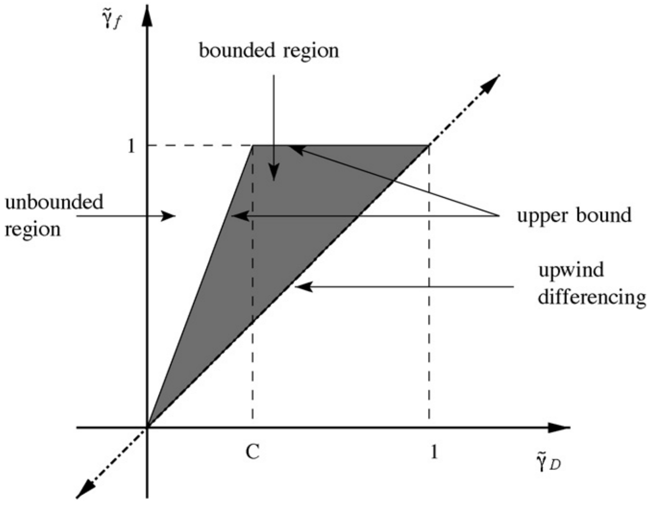
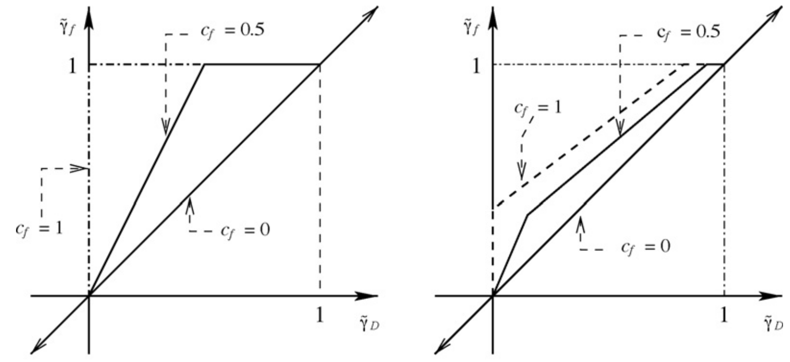


Fig. 8. The convection boundedness criteria for explicit flow calculations.



(a) Hyper-C

(b) ULTIMATE-QUICKEST

Fig. 9. Building blocks of the CICSAM scheme. (a) The upper bound of the CBC; (b) ULTIMATE-QUICKEST.

into a step profile, even if the orientation of the interface is almost tangential to the flow direction. This problem is countered by switching from the controlled downwinding formulation to upwind differencing under certain conditions. Hirt and Nichols' [4] VOF method determines the slope of the interface and switches to upwind differencing if the smallest angle between the interface and the face of the control volume is more than 45°. Numerical experimentation has shown that it is necessary to switch to another high-resolution (HR) scheme which will preserve the interface shape better. The CICSAM scheme of Ubbink [14] switches to ULTIMATE-QUICKEST (UQ) [7], the one-dimensional explicit bounded version of QUICK. The mathematical formulation of UQ is given as (Fig. 9):

$$\tilde{\gamma}_{f_{\mathrm{UQ}}} = \begin{cases} \min\left\{\dfrac{8c_f\tilde{\gamma}_{\mathrm{D}} + (1-c_f)(6\tilde{\gamma}_{\mathrm{D}}+3)}{8}, \tilde{\gamma}_{f_{\mathrm{CBC}}}\right\} & \text{when}0 \leq \tilde{\gamma}_{\mathrm{D}} \leq 1 \\ \tilde{\gamma}_{\mathrm{D}} & \text{when}\tilde{\gamma}_{\mathrm{D}} < 0, \tilde{\gamma}_{\mathrm{D}} > 1 \end{cases} \tag{46}$$

The CICSAM scheme is formulated in a way that it switches between the more compressive Hyper-C scheme and less compressive UQ scheme using a scaling factor $0 \leq \sigma_f \leq 1$ (Fig. 10) and the normalized face value of the color function is calculated using the following definition:

$$\tilde{\gamma}_f = \sigma_f \tilde{\gamma}_{f_{\mathrm{CBC}}} + (1-\sigma_f)\tilde{\gamma}_{f_{\mathrm{UQ}}}, \tag{47}$$

where σ_f is the scaling factor calculated based on the cosine of the angle θ_f between the vector normal to the interface $(\nabla\gamma)_{\mathrm{D}}$ and the vector $\boldsymbol{d}_f$, which connects the donor and acceptor cells. The equations for the angle and the scaling factor are:

$$\theta_f = \arccos\left|\frac{(\nabla\gamma)_{\mathrm{D}} \cdot \boldsymbol{d}_f}{|(\nabla\gamma)_{\mathrm{D}}||\boldsymbol{d}_f|}\right| \tag{48}$$

$$\sigma_f = \min\left\{\frac{\cos(2\theta_f)+1}{2}, 1\right\} \tag{49}$$

Fig. 10 shows the NVD for CICSAM scheme. The value $\sigma_f = 1$ is used when the interface orientation is normal to the direction of motion and $\sigma_f = 0$ is used when interface orientation is tangential to the direction of motion. Further, the extension of the above scheme to multi-dimensions results in the occurrence of non-physical values of the volume fraction as the face values are predicted in isolation of each other. To avoid such non-physical values, a detailed predictor–corrector solution procedure is explained in Ref. [14].

2.4. Inter-gamma differencing scheme

Rusche [13] presents a scheme proposed by Jasak and Weller [5], wherein the necessary compression of the interface is achieved by introducing an extra, artificial compression term into the VOF Eq. (15) instead of just using a compressive differencing scheme:

$$\frac{\partial\gamma}{\partial t} + \nabla\cdot(\boldsymbol{U}\gamma) + \nabla\cdot(\boldsymbol{U}^{\mathrm{r}}\gamma(1-\gamma)) = 0, \tag{50}$$

where $\boldsymbol{U}^{\mathrm{r}}$ is a velocity field suitable to compress the interface. This artificial term is active only in the interface region due to the term $\gamma(1-\gamma)$. The solution to this equation is bounded between zero and one using the inter-gamma differencing scheme which is explained later in the same section. Eq. (50) is re-written in the following form:

$$\frac{\partial\gamma}{\partial t} + \nabla\cdot(\phi[\gamma]_{f(\phi,S)}) + \nabla\cdot(\phi_{r_b}[\gamma]_{f(\phi_{r_b},S)}) = 0, \tag{51}$$

where $\phi = \boldsymbol{S}\cdot\boldsymbol{U}_f$ is the volumetric flux and $\phi_{r_b} = (1-\gamma)_{f(-\phi^{\mathrm{r}},S)}\phi^{\mathrm{r}}$. The following formulation for compression velocity $\boldsymbol{U}^{\mathrm{r}}$ is used:

$$\phi^{\mathrm{r}} = K_c n^* \max\left\{\frac{|n^*\phi|}{|\boldsymbol{S}|^2}\right\} \tag{52}$$

where K_c is an adjustable coefficient (preferred $K_c = 1.5$) which determines the magnitude of the compression and n^* is the unit normal vector of the interface evaluated from the smoothed indicator function. In Eq. (52), the maximum velocity magnitude is multiplied by the normal vector of the interface to achieve compression perpendicular to the interface. The solution to the discretized form of Eq. (51) is obtained by using an inter-gamma

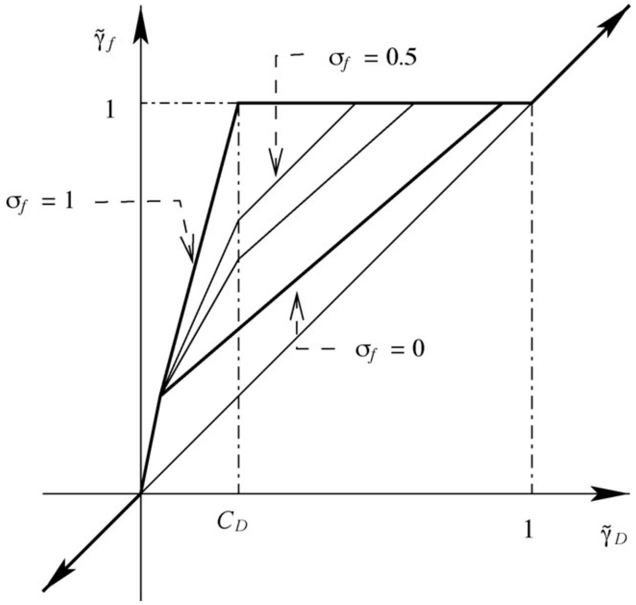


Fig. 10. NVD for CICSAM scheme.

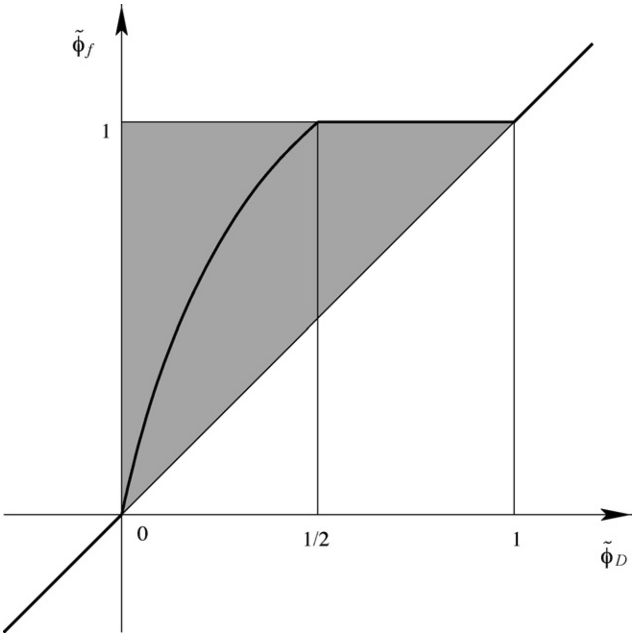


Fig. 11. Inter-gamma scheme in NVD.

differencing scheme by Jasak and Weller [5]. The inter-gamma differencing scheme is also based on the donor–acceptor formulation using Leonard's [6] normalized variable diagram. The variable and control volume arrangement (Fig. 6) is similar to ones used in CICSAM description. The mathematical formulation of the inter-gamma differencing scheme is described as:

- for $\tilde{\gamma}_D < 0$, $\tilde{\gamma}_f > 1 \Rightarrow \tilde{\gamma}_f = \tilde{\gamma}_D$ to obey the NVD criterion.
- for $\frac{1}{2} < \tilde{\gamma}_D < 1 \Rightarrow \tilde{\gamma}_f = 1$ to ensure the compressive behavior of the scheme.
- for $0 < \tilde{\gamma}_D < \frac{1}{2} \Rightarrow \tilde{\gamma}_f = -2\tilde{\gamma}_D^2 + 3\tilde{\gamma}_D$ to ensure a smooth change from upwind to downwind differencing

Fig. 11 shows the NVD for the inter-gamma differencing scheme. The shaded region in Fig. 11 shows the boundedness criterion for convection differencing schemes as described by Leonard [6] and Gaskell and Lau [2].

2.4.1. Courant number limit

The compressiveness of CICSAM and inter-gamma schemes is obtained by using controlled downwinding of the fluxes. Controlled downwinding schemes have compressive characteristics which depend on the CFL number, and therefore require small time steps to achieve sharp interfaces.

When simple upwind or downwind differencing schemes are selected for discretization of the convection term, the fluxes on the faces of the control volume guarantee that the central coefficient will not be zero. The requirement on the Courant number for central differencing comes from the condition of the diagonal equalness of the matrix.

In the case of inter-gamma and CICSAM differencing schemes the situation is a bit more complicated. The choice between upwind and downwind differencing is not made just by checking the direction of the flux. The choice is actually made based both on the flux direction and local shape of the solution. This means that the continuity criterion is not enough to guarantee the existence of the central coefficient. It often happens in one-dimension that one of the faces of the control volume uses upwind and the other uses downwind differencing. The result is that the central coefficient from the convection discretization comes out to be zero and the sum of the neighboring coefficients is twice as much as in the case of central differencing. Thus resulting in the Courant number limit of Co $< 1/2$. In multi-dimensions the situation gets even more severe and it is usually necessary to keep the Courant number to be less than Co $< 1/3$.

The low Courant number constraint posed by the compressive schemes result in high computational costs. One way to over come this problem is by using so called subcycles to solve the VOF equation. For a given Courant number, the flow equations are solved initially and later the VOF equation is fractionally updated n times, where n represents the number of subcycles pre-defined in the code. This provides the flexibility to use larger Courant numbers for the whole simulation as such without losing the accuracy of the VOF scheme.

3. Test cases

To evaluate the performance of the implemented VOF schemes, a number of standard test cases were setup. These test cases can be generally classified into theoretical ones, where comparisons are made against analytical solutions and real cases, where comparisons are made against theoretical/experimental data.

Table 3
Table showing the fractional error for two unidirectional velocity fields

Velocity	Scalar field	FCT	L-PLIC	CICSAM	I-gamma
(1,0)	Hollow square	8.3217e–17	3.3500e–02	5.6292e–04	9.7000e–03
	Hollow circle	9.5261e–17	2.7100e–02	3.0100e–02	1.2300e–02
(2,1)	Hollow square	2.8316e–16	8.9300e–02	2.7000e–02	7.1830e–01
	Hollow circle	3.0200e–02	1.4900e–02	7.4500e–02	7.1300e–01

Error estimation: The fractional error resulting from the simulations is calculated using:

$$E = \frac{\sum_{i,j} \|\gamma_{i,j}^{n} - \gamma_{i,j}^{\mathrm{e}}\|}{\sum_{i,j} \gamma_{i,j}^{\mathrm{o}}}, \tag{53}$$

where γ^n is the calculated solution after n time steps, γ^{e} is the exact solution after n time steps and γ^{o} is the initial solution.

The percentage error for each is scheme is estimated using:

$$\epsilon = 100 \frac{\sum_{i=1}^{N} \gamma_i^n - \sum_{i=1}^{N} \gamma_i^o}{\sum_{i=1}^{N} \gamma_i^o} \tag{54}$$

where N is the total number of cells in the domain, γ^n is the color function at the new time step and γ^{o} is the color function at the initial time step. Eq. (54) gives the percentage addition or deletion of mass by a scheme, i.e. it represents the diffusiveness of a given scheme.

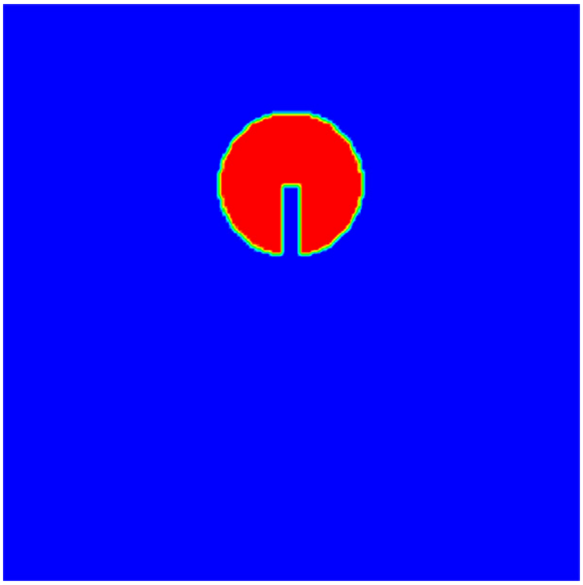


Fig. 12. Figure showing the initial configuration of the solid-body rotation test with the value of the color function is one inside the slotted circle and zero outside.

3.1. Advection tests (2D)

In this section, various shapes of color function are advected in chosen analytical velocity fields completely uncoupled from the solutions of the momentum equations. The two fluids considered are inviscid and of constant density. This allows to compare the methods without the effect of fluid behavior.

3.1.1. Translation test

In this case an oblique unidirectional velocity field is used to advect the following shapes of the color function:

(1) a hollow square aligned with the co-ordinate axes;
(2) a hollow circle.

The computational domain is of size 1 m × 1 m with mesh size 200 × 200. The exterior extent of the shapes initially is 40 mesh cells and the distance between the outer and inner interface is 10 mesh cells. A Courant number of 0.25 is used and the advection proceeds for 500 time steps. Two velocity fields (1,0) and (2,1) are used to advect the scalar fields.

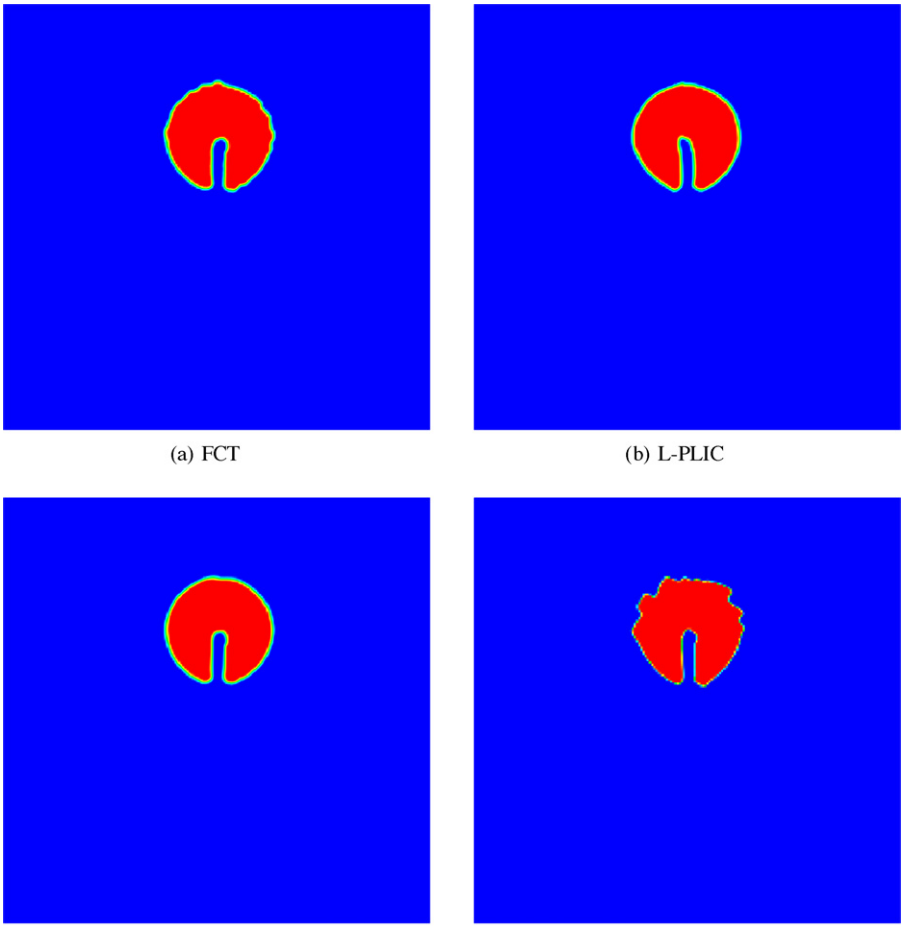
All the four methods give consistently good results for the case when the advection is aligned with a co-ordinate direction. In the case when the advection is inclined, the sharp corners of the square shape are rounded off by all the schemes. Table 3 shows the fractional errors for two advecting velocity fields and four methods. There is no consistent pattern to the errors. In general, FCT has the lowest error for both the velocity fields, compared to other cases.

3.1.2. Rotation of a slotted circle

Zalesak's [18] solid-body rotation of a slotted circle poses a tough test with regard to the advection schemes. In this case, a slotted circle is rotated through one or more revolutions around an external point. The computational domain is a square with dimensions 4 m *times* 4 m with uniform mesh of size 200 × 200. The diameter of the slotted circle is 50 mesh cells with the initial circle center at (2.0, 2.75) and the slot width is six mesh cells. The velocity field given by:

$$\begin{aligned} U &= -\Omega(y - y_0), \\ V &= \Omega(x - x_0), \end{aligned} \tag{55}$$

where $(x_0, y_0) = (2.0, 2.0)$ is the center of rotation and Ω is the angular velocity of 0.5 rad/s. Initially, the Courant number is 0.25 and one rotation corresponds to 2524 time steps. The initial conditions are shown in Fig. 12. Results after one rotation are shown in Fig. 13 for each of the four methods and the frac-


(a) FCT

(b) L-PLIC

(c) CICSAM

(d) Inter-Gamma

Fig. 13. Figures showing the results for solid-body rotation test after one rotation.

tional and percentage errors are shown in Table 4. Fig. 13(d) shows that the inter-gamma scheme results in wrinkling of the interface after one rotation. In contrast, CICSAM, Lagrangian piecewise linear interface construction (L-PLIC) and FCT gives sharp acceptable interface shapes, although the sharp corners at both ends of the slot are rounded off. The relative fractional error for the schemes is of the same magnitude.

3.1.3. Shearing flow test

The unidirectional velocity and rotational velocity fields do not involve any topological changes in the solution thus not providing a complete assessment of the integrity of the volume tracking methods. 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 section, shear is introduced into the velocity field. This test

Table 4
Table indicating the fractional error for solid-body rotation test after one rotation

Error	FCT	PLIC	CICSAM	I-gamma
Fractional	0.0810	0.1761	0.0749	0.1218

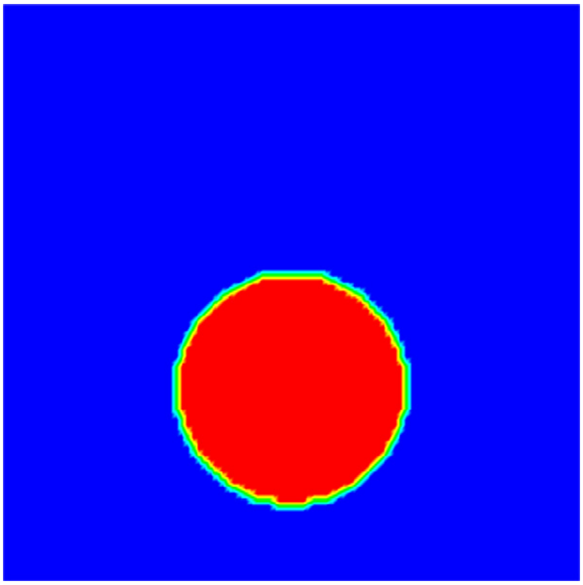
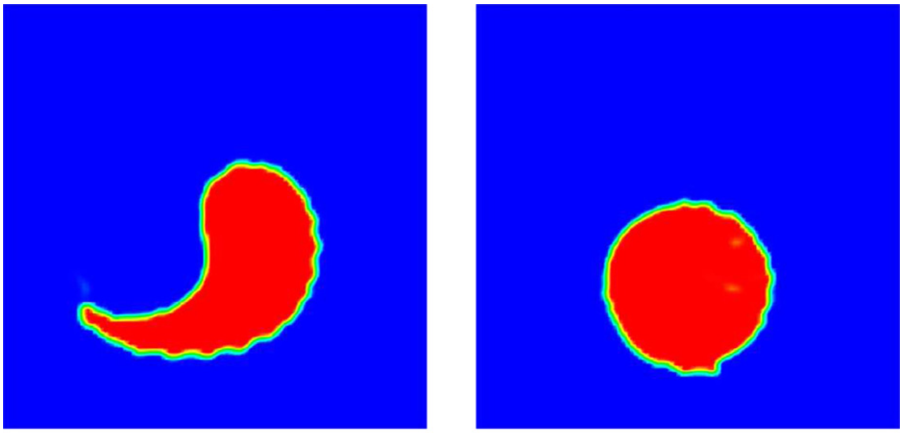


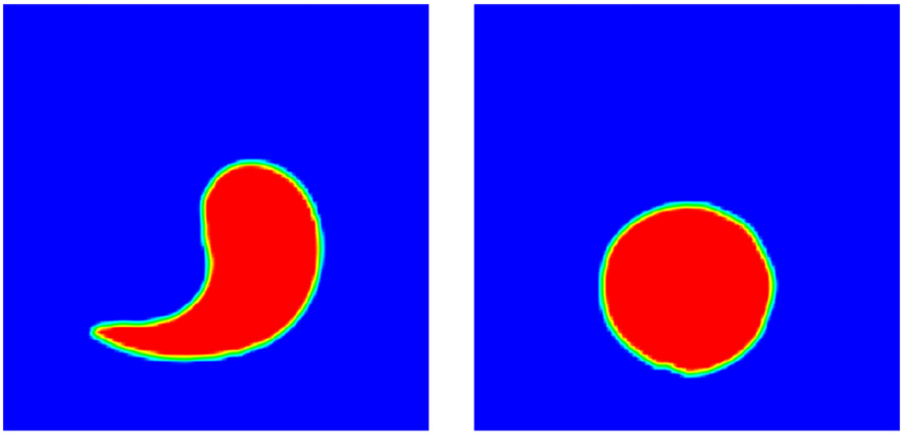
Fig. 14. Figure showing the initial configuration of the shearing flow test with the value of the color function is one inside the circle and zero outside.



(a) FCT - N = 1000

(b) FCT - N = 2000

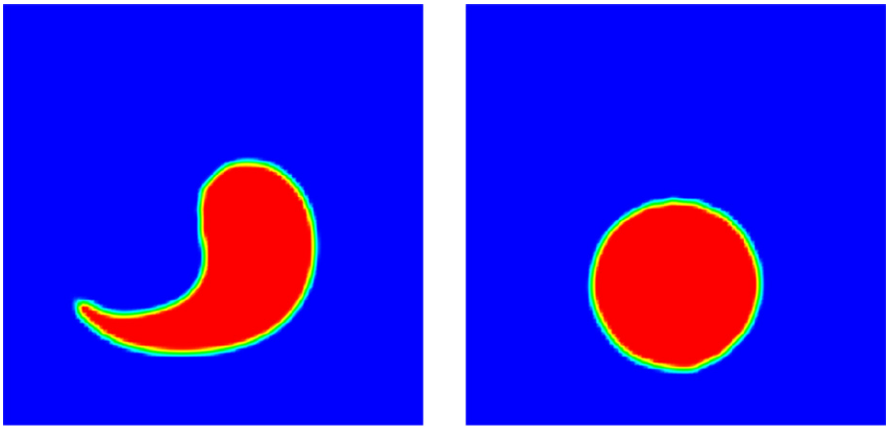
Fig. 15. Figures showing the result of the shearing flow test using FCT for $N = 2000$ time steps. (a) The result after 1000 time steps and (b) the results after 2000 time steps.



(a) L-PLIC - N = 1000

(b) L-PLIC - N = 2000

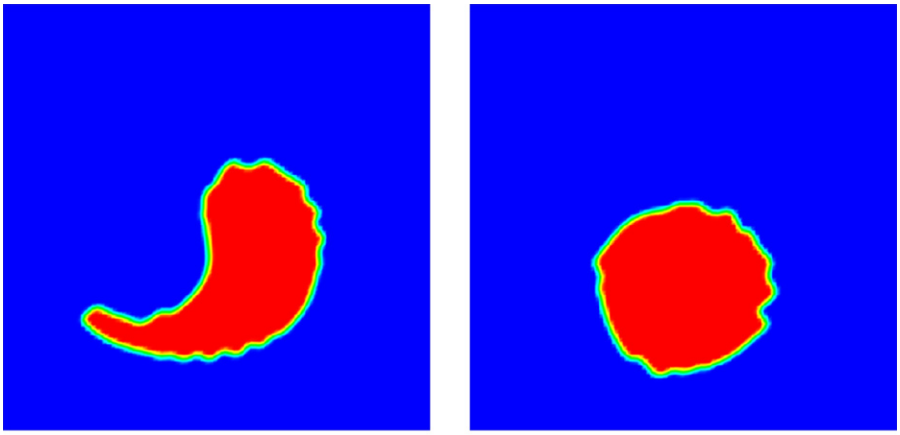
Fig. 16. Figures showing the result of the shearing flow test using L-PLIC for $N = 2000$ time steps. (a) The result after 1000 time steps and (b) The results after 2000 time steps.



(a) CICSAM - N = 1000

(b) CICSAM - N = 2000

Fig. 17. Figures showing the result of the shearing flow test using CICSAM for $N = 2000$ time steps. (a) The result after 1000 time steps and (b) the results after 2000 time steps.


(a) Inter-Gamma N = 1000

(b) Inter-Gamma N = 2000

Fig. 18. Figures showing the result of the shearing flow test using inter-gamma scheme for $N = 2000$ time steps. (a) The result after 1000 time steps and (b) the results after 2000 time steps.

is performed in both two and three dimensions: The following two-dimensional velocity field is chosen:

$$\begin{aligned} U &= \cos(x)\,\sin(y),\\ V &= -\sin(x)\,\cos(y), \end{aligned} \tag{56}$$

with x, $y\epsilon[0, \pi]$. The mesh is of size 100×100 with the initial condition a circle of radius $\pi/5$ with its center at $[\pi/2, \pi/3]$. The simulation is run for N time steps before reversing the sign of the velocity field and running for an additional N time steps. A perfect advection scheme would return the initial shape. Value of N in the range $500 \leq N \leq 2000$ are tested. A time step of $\Delta t = \pi/400$ which corresponds to a Courant number of approximately 0.25 is used.

Fig. 14 shows the initial configuration for the shearing flow test with the color function equal to one inside the circle and zero outside. Three cases with N =500, 1000 and 2000 were investigated. The simulations were run for $N/2$ time steps before changing the signs of the velocity field for the remaining $N/2$ time stamps. Figs. 15–18 shows the results at the two stages during the simulations for $N = 2000$ using FCT, L-PLIC, CICSAM and inter-gamma schemes respectively. Qualitative comparison shows that CICSAM and L-PLIC schemes clearly perform the best, maintaining a sharp interface and also retaining the initial configuration after 2000 time steps. FCT forms small wiggles at $N = 1000$ (Fig. 15(a)), but does seem to retain the initial configuration after 2000 time steps. Inter-gamma scheme clearly performs the worse compared to other schemes. Table 5 shows the progression of fractional errors at three different time steps. The errors for the inter-gamma schemes are clearly higher than other schemes.

Table 5
Table indicating the fractional error for shearing flow test

Time steps	FCT	PLIC	CICSAM	I-gamma
500	0.0258	0.0210	0.0262	0.0193
1000	0.0255	0.0249	0.0269	0.0242
2000	0.0540	0.0286	0.0286	0.0302

3.2. Realistic flow cases

In the following sections tests are performed using a realistic flow field wherein the advection of color function is dynamically coupled to the solution of the momentum equations. The realistic flow field includes the degree of complexities encountered in practical situations which is hard to ensure with an analytical velocity field.

3.3. Sloshing test

The sloshing of a liquid wave with low amplitude under the influence of gravity is used as a test case to evaluate the interface capturing methodology by Raad et al. [11]. The same setup is used here to evaluate the performance of implemented VOF methods. The two-dimensional setup considered is shown in Fig. 19. The computational domain has dimensions of 0.1 m in length and 0.065 m in depth. Initially, the surface of the quiescent fluid is defined by one half of a cosine wave with an amplitude of 0.005 m. The bottom and the sides of the computa-

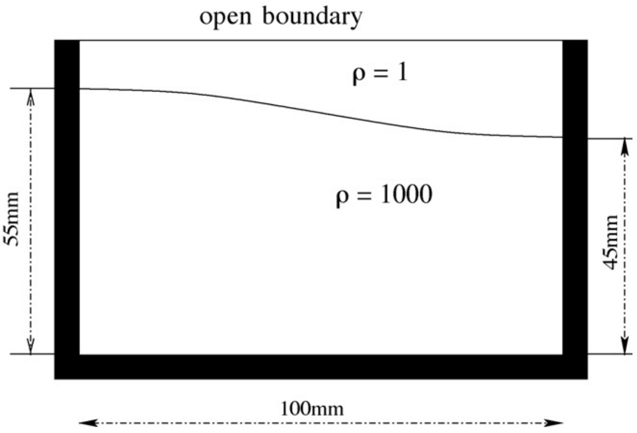
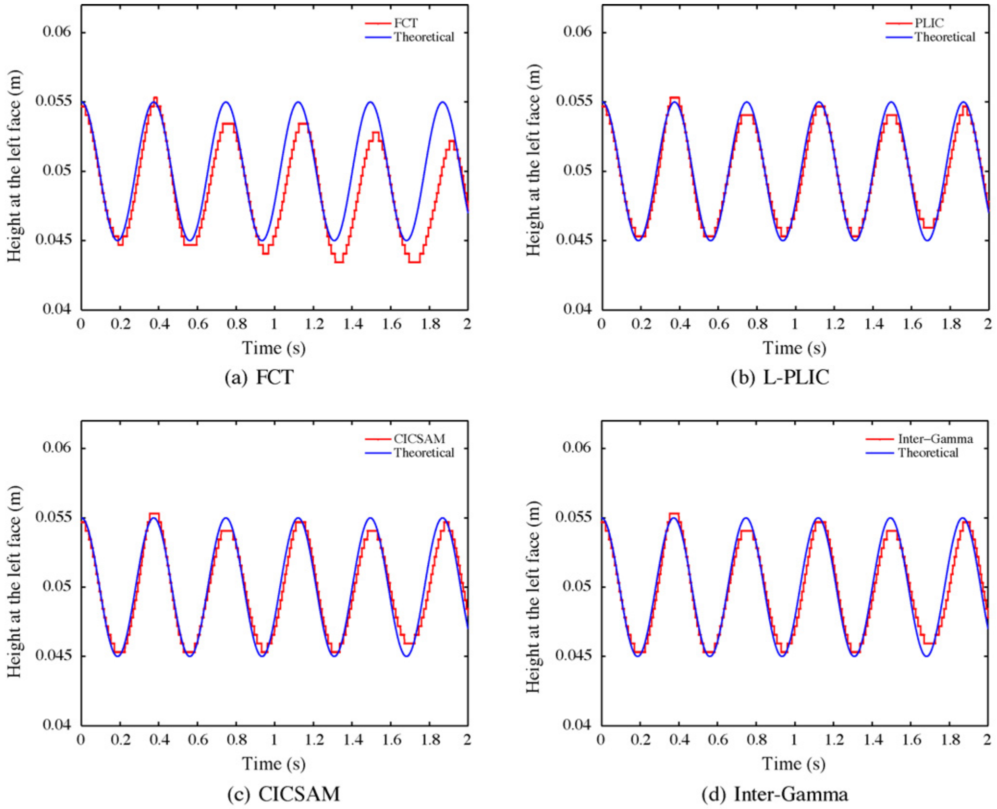


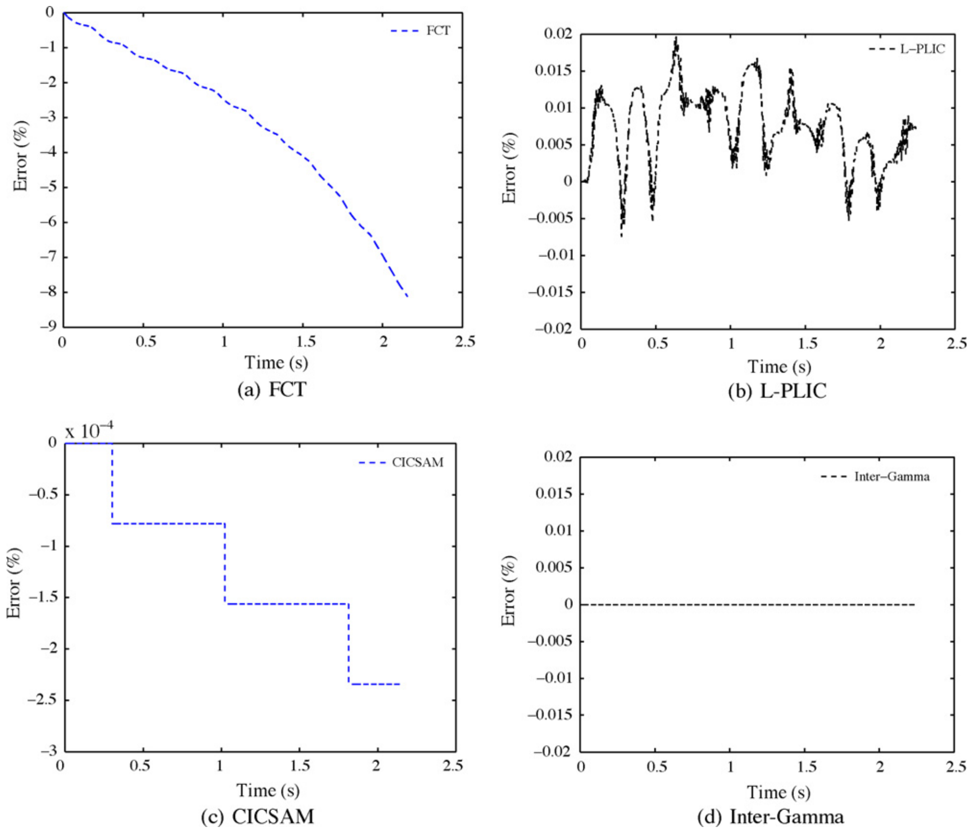
Fig. 19. Figure showing the 2d tank approximately three fourth filled with water forming an interface with air. The interface is a cosine wave of low amplitude as the initial condition.



(a) FCT (b) L-PLIC

(c) CICSAM (d) Inter-Gamma

Fig. 20. Figures comparing the calculated and the theoretical position of the interface at the left wall as a function of time for six periods of oscillation.



(a) FCT (b) L-PLIC

(c) CICSAM (d) Inter-Gamma

Fig. 21. Figures showing percentage error for each scheme as a function of time.

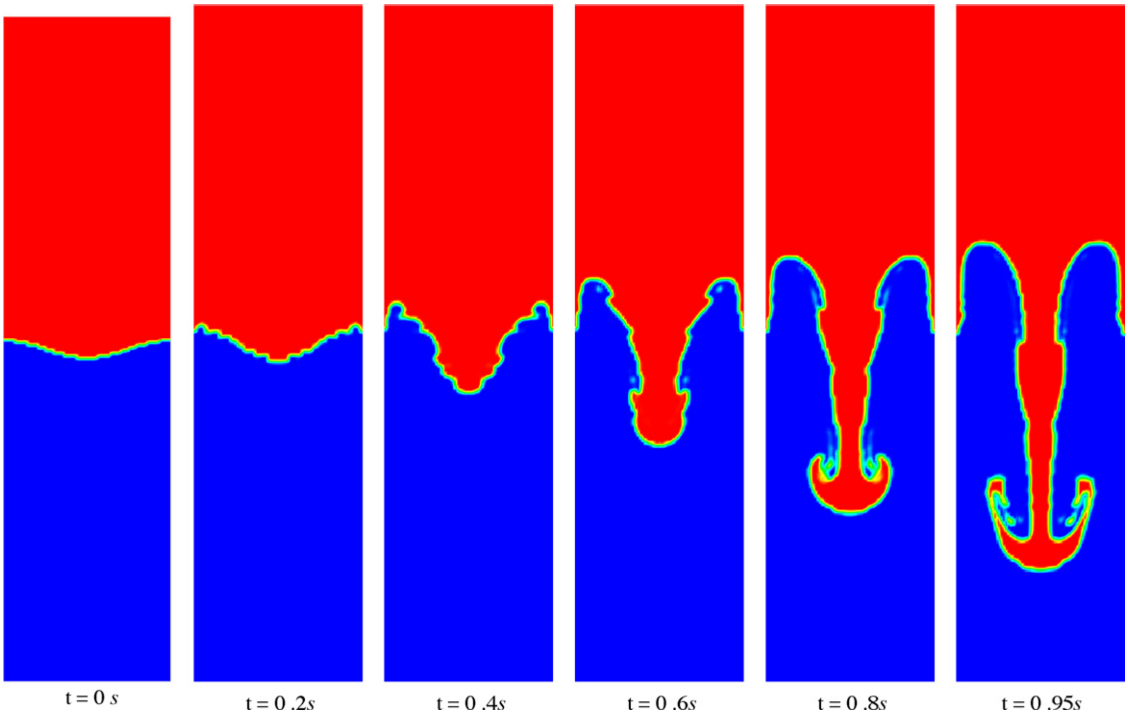


Fig. 22. Figures showing the progression of Rayleigh–Taylor instability in time using FCT scheme.

tional domain are treated as walls and at the top a fixed pressure boundary condition is applied. Both the fluids are considered inviscid with densities $\rho_1 = 1000\,\text{kg/m}^3$and $\rho_2 = 1\,\text{kg/m}^3$. The domain is discretized with 160 cells in the horizontal direction and 104 cells in vertical direction.

The fluid begins to slosh solely under the influence of gravity. The theoretical sloshing of the first mode is (Raad et al. [11]):

$$P = \frac{2\pi}{\sqrt{gk\ \tan h(kh)}} = 0.3737\,s \tag{57}$$

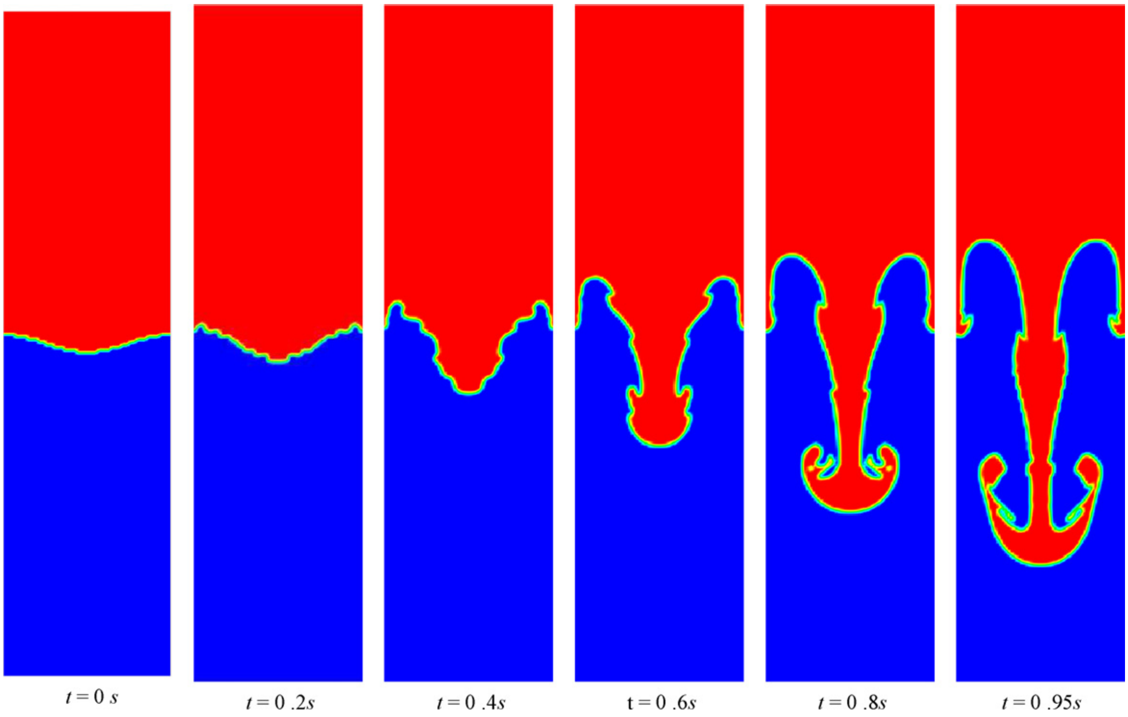


Fig. 23. Figures showing the progression of Rayleigh–Taylor instability in time using L-PLIC scheme.

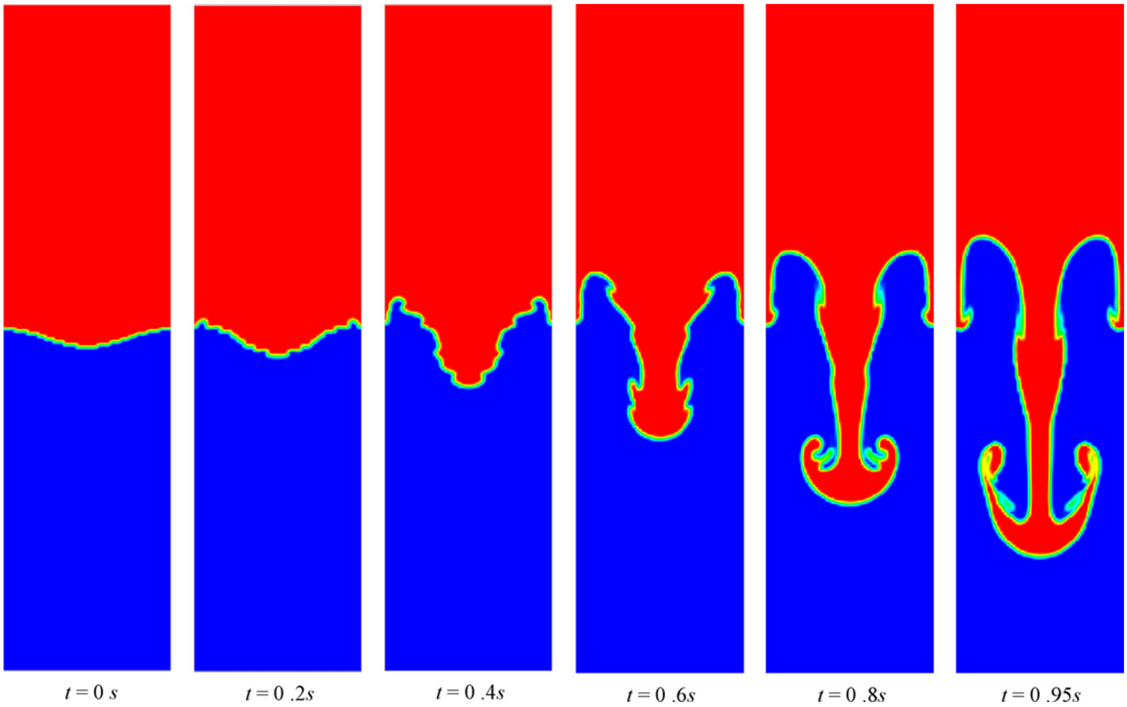


Fig. 24. Figures showing the progression of Rayleigh–Taylor instability in time using CICSAM scheme.

where k is the wave number and h the average fluid depth. Fig. 20 shows the calculated and the theoretical position of the interface at the left boundary against time for the first six periods of oscillation. It is observed that except for FCT, other three schemes closely follow the theoretical calculation. FCT seems to follow the trend initially, but with time it deviates considerably from the theoretical results. This can be explained by the loss of mass in time as shown in Fig. 21(a). Further Fig. 21(d) shows that the inter-gamma compressive scheme is far superior to all other schemes in terms of mass conservation. Lagrangian PLIC and CICSAM also perform well considering the maximum error magnitude being 0.02 and 2.5e −4% respectively.

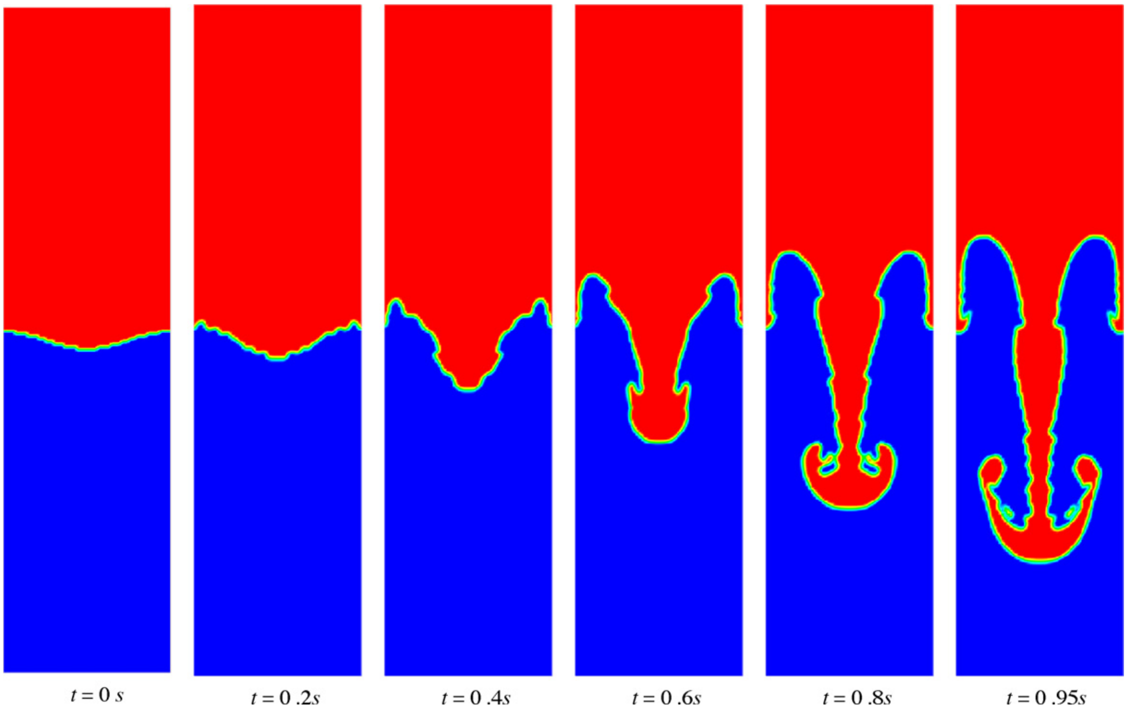


Fig. 25. Figures showing the progression of Rayleigh–Taylor instability in time using inter-gamma scheme.

Table 6
Table showing the percentage error of Rayleigh–Taylor instability simulation at various time steps

Time	FCT	PLIC	CICSAM	I-gamma
0.2 s	−0.0151	−1.225e–5	−8.21e–4	0.0
0.4 s	−0.150	−6.370e–5	−0.0015	0.0
0.6 s	−0.6876	−8.946e–4	−0.0021	0.0
0.8 s	−1.5167	−0.044	−0.0027	0.0
0.95 s	−2.6791	−0.0174	−0.003	0.0

3.4. Rayleigh–Taylor instability

The next test case chosen is the Rayleigh–Taylor instability problem (Refs. [12] and [8]). In this case, a high density liquid is placed over a low density liquid in a rectangular domain of dimension 1 m × 4 m with a mesh resolution of 64 × 256. The top half of the domain is filled with fluid of density $\rho_1 = 1.225\,\text{kg/m}^3$ ($\gamma = 1$) and the bottom half is filled with a fluid of density $\rho_2 = 0.1694\,\text{kg/m}^3$ ($\gamma = 0$). The viscosity of both the fluids is taken as 3.13×10^{-3} kg/m/s. Initially a perturbation given by the function $y = 0.05\,\cos(2\pi x)$ is applied at the interface. This results in a density driven flow and errors in the γ field will lead to errors in the momentum equations via Eq. (11). No-slip boundary conditions are used on all the boundaries. Keeping in mind the Courant number limitations of CICSAM and inter-gamma differencing schemes, a constant time step of 2e −4 is used which approximately corresponds to Courant number of 6e −2.

Figs. 22–25 show progression of the Rayleigh–Taylor instability using FCT, Lagrangian PLIC, CICSAM and inter-gamma schemes at $t = 0, 0.2, 0.4, 0.6, 0.8$ and 0.95 s. A qualitative comparison of the results from different schemes shows similar patterns up to 0.4 s. At 0.6 s, FCT shows the formation of jetsam below the peak on both the sides close to the wall and also close to the filament connecting to the blob at the center. The formation of the blob at the center is seen to be rounded off in the case of FCT compared to other schemes. In time, the interface at the center takes the shape of an inverted mushroom, with smaller secondary filaments forming at the inner circumference of this instability. The formation of these secondary instabilities are more pronounced (sharp) with inter-gamma, CICSAM and PLIC schemes compared to more diffused solution of FCT. Table 6 shows the percentage error of the schemes at different time steps. It can be clearly seen that FCT is the most diffusive scheme with a mass loss corresponding to 2.6% relative to PLIC and CICSAM schemes, which explains the diffusive solution obtained by FCT. inter-gamma scheme is superior with 100% mass conservation.

4. Summary of methods

The FCT method is a simple scheme to implement. It uses a combination of a lower order stable scheme and a higher order monotonic scheme to calculate anti-diffusive fluxes. The fluxes are limited using limiters calculated from the possible theoretical extrema in the solution. Direction splitting is used to extend the scheme into multi-dimensions. The results of the simple advection tests show that the performance of FCT is reasonably good, but, the results of more practical tests, i.e. sloshing of a liquid wave and the Rayleigh–Taylor instability problem indicate that FCT results in a more diffusive interface. This is attributed to the loss of mass, indicated by the percentage error for each test case. Hence, for many practical cases FCT is not mass conserving. This is because the small round-off errors accumulate and affect the boundedness of the solution after a few hundred time steps. Thus, after every time step, negative color function values are set to zero and any value greater than one is set to unity. Also, the method being direction split results in the method being restrained to only structured meshes.

The Lagrangian-PLIC scheme uses a straight line in 2d, or plane in 3d, to reconstruct the interface in a cell and the fluxes leaving the cell faces are calculated geometrically. The method is extended to multi-dimensions using direction-split technique. The results of both simple advection tests and the real test cases show that the scheme is fairly accurate keeping a sharp interface by definition. The accuracy criterion for all test cases shows very small percentage of mass loss, indicating that the scheme is mass conservative. The scheme is direction split, which limits its usage to structured meshes.

The CICSAM scheme is a high-resolution scheme based on the donor–acceptor flux approximation using the normalized variable diagram of Leonard [6]. The CICSAM scheme is formulated in a way that it switches between the more compressive Hyper-C scheme and less compressive Ultimate-QUICKEST scheme using a scaling factor calculated based on the angle made by the normal to interface to the vector connecting the donor and acceptor cells. The results of the tests indicate that the scheme is very accurate keeping the interface sharp. The scheme, as the name indicates, is derived for arbitrary meshes, thus providing flexibility in-terms of boundary fitted grid usage. The limitations of the scheme is that the boundedness of the scheme is dependent on the local Courant number, requiring very small time steps to keep the interface sharp. The error for different test cases indicates that the method is fairly mass conservative.

The inter-gamma scheme uses an extra artificial compression term into the VOF Eq. (15) instead of merely using a compressive differencing scheme. The differencing scheme is also based on Leonard's [6] normalized variable diagram. The simple advection results indicate that the scheme is very dependent on CFL numbers resulting in the usage of very small time steps for keeping the interface sharp. The practical test cases using low CFL numbers show very good resolution of the interface and the percentage error for all the test cases show zero addition or deletion of mass indicating that the method is completely mass conservative.

5. Conclusions

This paper discusses four volume of fluid (VOF) schemes in detail, showing the derivation, implementation, advantages and limitations. The schemes presented are: flux-corrected transport (Boris et al. [1]), Lagrangian piecewise linear interface construction (van Wachem and Schouten [16]), Compressive interface capturing scheme for arbitrary meshes (Ubbink [14]) and inter-gamma scheme (Jasak and Weller [5]). The performance of the four schemes is tested initially using simple analytical velocity fields and further with two practical test cases of sloshing of a liquid wave and the Rayleigh–Taylor instability problem.

The FCT method works well for an analytical steady velocity field, but is non-mass conservative for practical flow cases. Because the method employs direction splitting for higher geometric dimensions, the method is restricted to structured meshes. The Lagrangian-PLIC scheme is found to be fairly accurate and mass conservative for all flow test cases, but can only be employed for structured meshes. It can be employed with quite large time steps, as long as CFL < 1. Both the compressive schemes CICSAM and inter-gamma are found to be accurate for all flow cases, in conserving mass and keeping the interface very sharp. CICSAM and inter-gamma are flexible in terms of arbitrary mesh usage, as they do not require direction splitting for higher geometric dimensions. However, the schemes are very sensitive to the local CFL number, as they require very small time steps, CFL < 0.01, for keeping a sharp interface.

References

[1] J.P. Boris, D.L. Book, Flux-corrected transport. I: SHASTA, a fluid transport algorithm that works, J. Comput. Phys. 11 (1973) 38–69.

[2] P.H. Gaskell, A.K.C. Lau, Curvature-compensated convective transport: SMART, a new boundedness-preserving transport algorithm, Int. J. Numer. Methods Fluids 8 (1988) 617–641.

[3] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.

[4] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.

[5] H. Jasak, H.G. Weller, Interface-tracking capabilities of the InterGamma differencing scheme, Technical Report, Imperial College, University of London, 1995.

[6] B.P. Leonard, Simple high-accuracy resolution program for convective modeling of discontinuities, Int. J. Numer. Methods Fluids 8 (1988) 1291–1318.

[7] B.P. Leonard, The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection, Comput. Methods Appl. Mech. Eng. 88 (1991) 17–74.

[8] J. Lòpez, J. Hernàndez, P. Gòmez, F. Faura, An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows, J. Comput. Phys. 208 (2005) 51–74.

[9] W.F. Noh, P. Woodward, SLIC (Simple Line Interface Calculations), Lecture Notes Phys. 59 (1976) 330–340.

[10] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1988) 234–246.

[11] P.E. Raad, S. Chen, D.B. Johnson, The introduction of micro cells to treat pressure in free surface fluid flow problems, J. Fluids Eng. 117 (1995) 683–690.

[12] M. Rudman, Volume-tracking methods for interfacial flow calculations, Int. J. Numer. Methods Fluids 24 (1997) 671–691.

[13] H. Rusche, Computational fluid dynamics of dispersed two-phase flows at high phase fractions, Ph.D. Thesis, Imperial College of Science, Technology and Medicine, 2002.

[14] O. Ubbink, Numerical prediction of two fluid systems with sharp interfaces, Ph.D. Thesis, Imperial College of Science, Technology and Medicine, 1997.

[15] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.

[16] B.G.M. van Wachem, J.C. Schouten, Experimental validation of 3-d Lagrangian VOF model: Bubble shape and rise velocity, AIChE 48 (12) (2002) 2744–2753.

[17] D.L. Youngs, K.W. Morton, M.J. Baines, Time-dependent multi-material flow with large fluid distortion, in: Numerical Methods for Fluid Dynamics, Academic Press, New York, 1982, pp. 273–285.

[18] S.T. Zalesak, Fully multi-dimensional flux corrected transport algorithm for fluid flow, J. Comput. Phys. 31 (1979) 335–362.