Chapter 15

Free Surface Flow Simulation Using VOF Method

Mohammad Javad Ketabdari

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/64161

Abstract

Viscous flow with moving free surface is an important phenomenon in nature which has broad applications in engineering. For these flows, temporal and spatial position of this moving free surface in unsteady or non-uniform conditions is very complicated. In this chapter, free surface simulation methods based on computational grid are presented. Volume of fluid (VOF) is a powerful and the most prevailing method for modeling two immiscible incompressible fluid-fluid interfaces. Herein, the governing equations of fluid flow including Navier-Stokes coupled with VOF equation are discussed and the most prominent VOF schemes hierarchically presented to the readers. Meanwhile, Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM), Higher Resolution Artificial Compressive (HiRAC), High Resolution Interface Capturing (HRIC), Switching Technique for Advection and Capturing of Surfaces (STACS), and some other newly proposed methods are introduced, and the accuracy and time calculation of each method are evaluated. Moreover, surface tension modeling and its discretization as one of the most demanding phenomena in the nature are brought to the readers. Finally, two schemes of parametric study of interfaces are discussed.

Keywords: viscous flow, free surface modeling, Eulerian approach, volume of fluid, interface simulation

1. Introduction

Simulations of free surface flows have progressed rapidly over the last decade, and it is now possible to simulate the motion of complicated waves and their interactions with structures considering even deformable bubbles in turbulent flows. In the continuum mechanics, there are two methods to express the motion in the environment. The first description is the Eulerian approach. In this method, attention is paid to a special volume in the space. A mesh remains fixed in the Eulerian method and fluid regions change in shape and location on the mesh. It uses a fixed grid system which is not transformed during the solution procedure. The fluid is studied while passing this volume and continuously replaced in time. Therefore, this method is not appropriate for formulation of basic equations of fluid movement. The Eulerian method has some limitations. For example, when the portion of the perimeter to the area of a zone of fluid is large, the error of this method is increased. In the Eulerian method, it is not possible to decompose the equation on the boundaries with the same precision of inner region of fluid and accordingly, the finer mesh should be used near the boundaries. Therefore, when the free surface of a discontinuous region is modeled by this method, finer grid should be employed in order to achieve more precise results, specifically if this surface has large deflections. This is crucial when the portion of the area to the perimeter of a zone is low, for example on phase of a multiphase fluid. In this case, using finer mesh could increase the portion of the number of the inner elements to the boundary elements, which in turn, increases the precision of the numerical solution. The main superiority of the Eulerian description is the possibility of modeling of complicated surfaces. For example, the collapse of a column of a fluid could be modeled in the Eulerian grid which is shown in **Figure 1**.

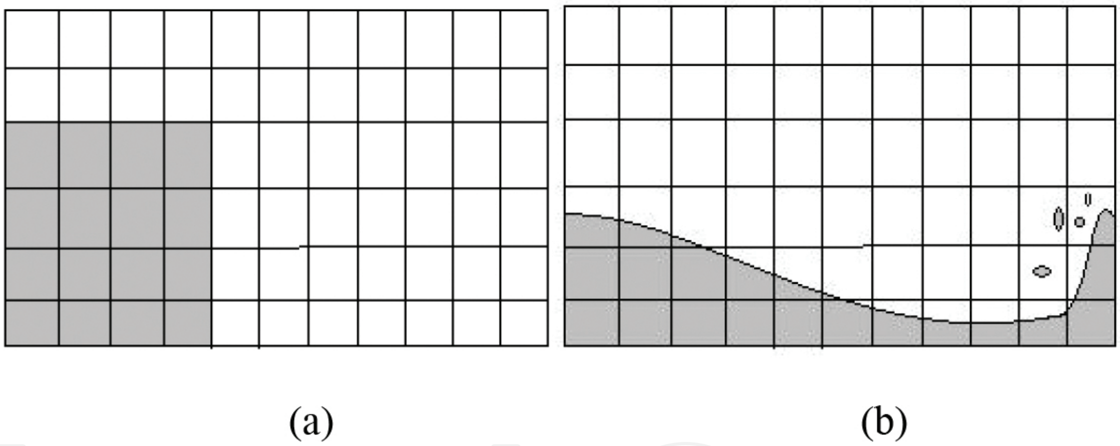

Figure 1. Fluid column in Eulerian grid: (a) before collapse and (b) collapsing flow.

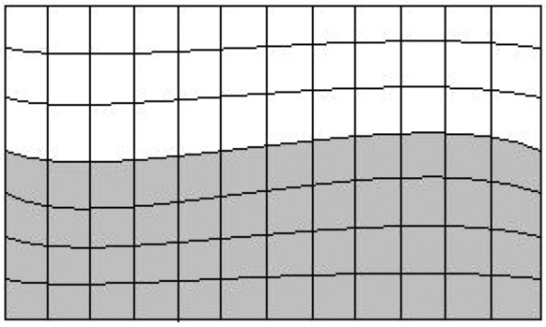

Figure 2. A sample of Lagrangian grid in vertical direction.

In Lagrangian method, the flow field of the considered fluid is covered by a mesh moving with the fluid. The fluid boundaries always coincide with the grid boundaries and the fluid inside each cell of the grid always remains in that computational cell. Although this method is not applicable to flows undergoing large distortions, where meshes can be twisted into unacceptable shapes, but its advantage is the ease with which it handles free surfaces and interfaces, which makes it applicable to a wide variety of problems. For example, the grid shown in **Figure 2** is Lagrangian in the vertical coordinate. For free surface problems, if the free surface movement or the tangential acceleration gradient in the perpendicular direction to its surface is not large, the Lagrangian method can be used to simulate free surfaces. The grid lines are located on the free surface and move with it. Therefore, there is no need for any special boundary condition in this location [1].

2. Governing equations

Governing equations for a compressible viscous fluid flow with no phase change are as follows:

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

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla P = \rho \mathbf{g} + \mathbf{F}_s + \nabla \cdot \left[\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}} \right) \right] \tag{2}$$

In these equations, ρ, $\mathbf{u}$, t, P, μ, $g=(g_x, g_y)$, and $\mathbf{F}_s$ are density, velocity vector, time, total pressure, kinematic viscosity, gravity acceleration, and body forces, respectively. Body forces include forces due to surface tension in the interface. Here, properties of a fluid such as density and viscosity are included in the equations. However, it should be kept in mind that the information changes from one fluid to another. Thus for mesh-based numerical methods, new properties based on fluid properties of both materials should be considered for Eq. (2) in the cell containing the free surface, and the governing equations should be rewritten in the following form:

$$\frac{\partial}{\partial t}(\alpha \rho_1) + \nabla \cdot (\alpha \rho_1 \mathbf{u}_1) = 0 \tag{3}$$

$$\frac{\partial}{\partial t}\left((1-\alpha)\rho_2\right) + \nabla \cdot \left((1-\alpha)\rho_2 \mathbf{u}_2\right) = 0 \tag{4}$$

$$\frac{\partial(\alpha \rho_1 \mathbf{u}_1)}{\partial t} + \nabla \cdot (\alpha \rho_1 \mathbf{u}_1 \mathbf{u}_1) + \nabla P_1 = \alpha \rho_1 \mathbf{g} + \mathbf{F}_s + \nabla \cdot \left[\alpha \mu_1 \left(\nabla \mathbf{u}_1 + \nabla {\mathbf{u}_1}^{\mathrm{T}} \right) \right] \tag{5}$$

$$\frac{\partial\left(\left(1-\alpha\right)\rho_2\mathbf{u}_2\right)}{\partial t}+\nabla\cdot\left(\left(1-\alpha\right)\rho_2\mathbf{u}_2\mathbf{u}_2\right)+\nabla P_2=\left(1-\alpha\right)\rho_2\mathbf{g}+\mathbf{F}_s+ \\ \nabla\cdot\left[\left(1-\alpha\right)\mu_2\left(\nabla\mathbf{u}_2+\nabla\mathbf{u}_2{}^{\mathrm{T}}\right)\right] \tag{6}$$

where indices 1 and 2 show first and second fluids properties, and α is a scalar phase indicator function which is defined as follows:

$$\alpha=\begin{cases}1 & \text{Control volume is filled only with phase 1}\\ 0 & \text{Control volume is filled only with phase 2}\\ 0<\alpha<1 & \text{Interface present}\end{cases} \tag{7}$$

This phase indicator function is the fluid property or volume fraction, which moves with it and can be derived as follows:

$$\frac{\partial\alpha}{\partial t}+\frac{\partial U_i\alpha}{\partial x_i}=0 \tag{8}$$

This function can be used to calculate the fluid properties in each phase as a weight function. In order to use a set of governing equations using the weight function, each fluid property should be calculated based on the volume occupied by this fluid in the surface cell as expressed in Eqs. (9) and (10) [2]:

$$\rho=\alpha\cdot\rho_1+\left(1-\alpha\right)\cdot\rho_2 \tag{9}$$

$$\mu=\alpha\cdot\mu_1+\left(1-\alpha\right)\cdot\mu_2 \tag{10}$$

Free surfaces considered here are those on which discontinuities exist in one or more variables. This has been the challenge for researchers to omit or reduce this problem as much as possible. The transient state as well as phenomena such as surface tension, changing of fluid phase and Kelvin-Helmholtz instability makes numerical simulation of such problems cumbersome. It is expected that methods used to simulate interface of fluids have a number of characteristics. These include mass conservation, simulating the interface as thin as possible, being able to reproduce complicated topologies, generalization of expansion to 3D problems, and being able to model surface phenomena and be computationally efficient.

3. Free surface modeling methods

There are different methods to simulate free surface flow, each of which has its own advantages and disadvantages:

3.1. Donor-acceptor method

The main idea of donor-acceptor approach is that the value of volume fraction in downwind cell, the acceptor cell, is used for anticipation of transferring fluid in each time step. The problem in this approach is that using downwind cell in calculations may lead to unreal situations which are values out of zero and unity domain in surface cells. **Figure 3a** shows this method with the first fluid with gray color and volume of fluid equals to unity. It could be seen that using donor-acceptor approach with downwind differencing scheme results in values greater than unity in donor cell. It is because the second fluid in the acceptor cell is greater than the value needed in the donor cell. Similarly in **Figure 3b**, using downwind differencing scheme leads to negative values for volume of fluid, which is because the needed fluid in acceptor cell is more than what is in the donor cell [3].

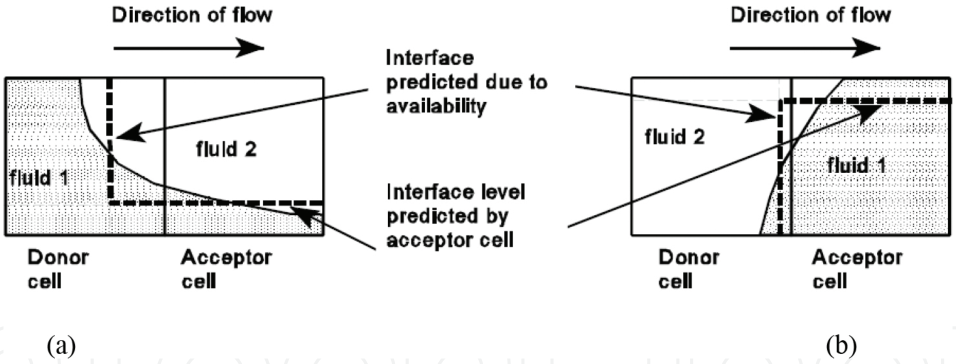

Figure 3. Schematic view of donor-acceptor approach [4].

In order to be assured that volume of fluid is between zero and one, the amount of fluid or volume of fluid in donor cell should be used to regulate the estimated fluid transferring between two adjacent cells [5].

One drawback of donor-acceptor method is that this method changes any finite gradient into step, and consequently increases the slope of the surface model in the direction of flow. This problem was alleviated by proposing a method to consider the slope of interface for flux transferring in adjacent cells by Hirt and Nichols [6]. For this purpose, a donor-acceptor equation was proposed so that it could detect the direction of the flow in interface and then define the upwind and downwind cells accordingly. Thereafter, this model was expanded for 3D domains by Torrey et al. [7]. The Surfer method is one version of volume of fluid which deals with merging and fragmenting of interfaces in multiphase flows [8].

The volume of fluid method is one of the most popular methods for anticipation of interfaces, and many researches have been conducted based on this method including dam break, Rayleigh-Taylor instability, wave generation and bubble movement [6, 9–12]. This method was modified in 2008 to get more accurate results by considering diagonal changes in fluxes of adjacent cells for structured grid domains [13, 14].

3.2. The Hirt-Nichols method

The volume of fluid (VOF) method was first proposed by Hirt and Nichols [6]. In this method, similar to the SLIC method, free surfaces can be reconstructed based on parallel lines with respect to one of the principal coordinates of the system. However, nine neighboring cells are considered for flux changes and defining the normal vector in a desired cell. Then, free surface is considered as either a horizontal or a vertical line in cell with respect to the relative normal vector components. **Figure 4** shows the actual free surface and what was simulated by Hirt-Nichols method.

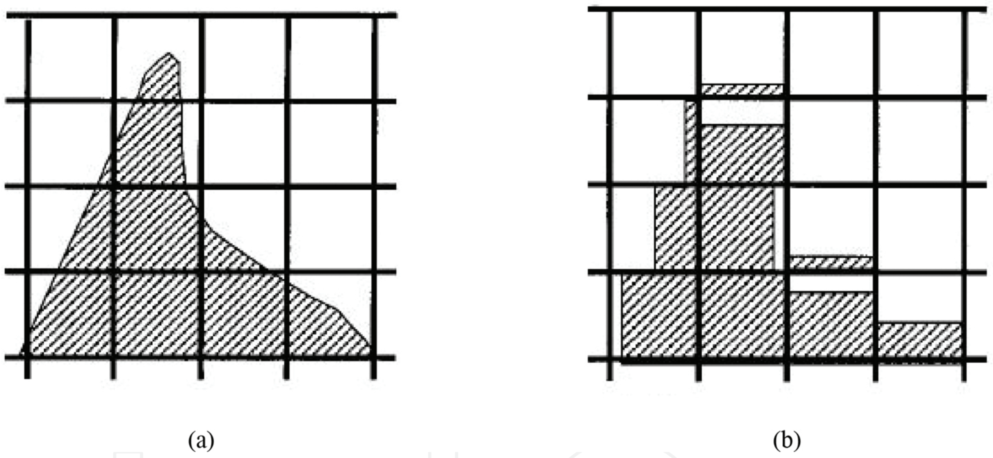

Figure 4. Free surface (a) actual surface and (b) reconstructed surface based on Hirt-Nichols method [6].

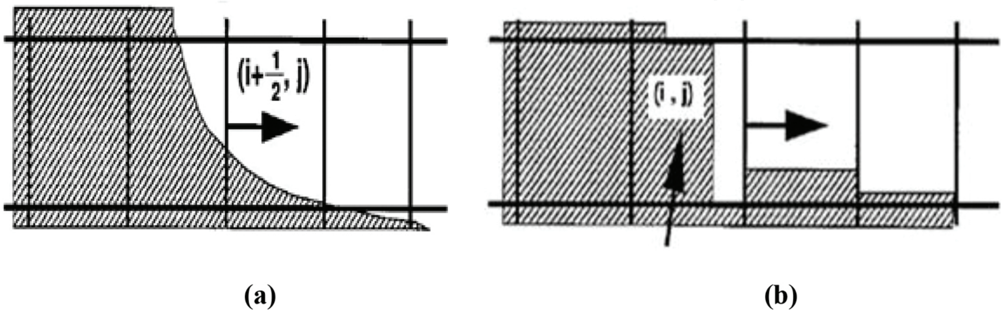

Figure 5. Hirt-Nichols scheme (a) actual surface and (b) reconstructed surface.

Upwind fluxes are used for fluxes parallel to the reconstructed interface, while donor-acceptor fluxes are used for those fluxes normal to it. For instance according to **Figure 5a**, the interface in the cell $(i,\ j)$ is considered to have positive celerity direction with respect to x coordinate in the face $i+{}^{1}\!/_{2}$ in donor-acceptor method. Therefore, the reconstructed surface in the $(i,\ j)$ cell is vertical (**Figure 5b**), and this cell is considered a downwind cell for the cell $(i+1,\ j)$, $(\alpha_{i,\,j}>\alpha_{i+1,\,j})$. According to donor-acceptor method, transferred flux from the face $\left(i+{}^{1}\!/_{2},\ j\right)$ can be calculated as follows:

$$F_{i+\frac{1}{2},j}=\delta y\left\{\min\left[\alpha_{i,j}\delta x,U_{i+\frac{1}{2},j}\alpha_{i+1,j}\delta t+\max\left(0,U_{i+\frac{1}{2},j}\left(1-\alpha_{i+1,j}\right)\delta t-\left(1-\alpha_{i,j}\right)\delta x\right)\right]\right\} \tag{11}$$

where $\alpha_{i,\,j}\delta x$ is the maximum fluid available for exiting the cell $(i,\ j)$; $U_{i+{}^{1}\!/_{2},\,j}\alpha_{i+1,\,j}\delta t$ is the estimation of downwind flux for the volume of the fluid, α; $U_{i+{}^{1}\!/_{2},\,j}(1-\alpha_{i+1,\,j})\delta t$ is the estimation of downwind flux resulted from the convey of void portion of the cell $(i,\ j)$; and $(1-\alpha_{i,\,j})\delta x$ is the maximum void portion which can exit from the cell $(i,\ j)$.

The "min" operator has been designed to ensure the fluid leaving the cell $(i,\ j)$ is not more than the calculated available fluid in it from the previous time step. As the fluid in a cell transfers, so does the whole void space in the cell. Thus, the "max" operator has been designed in order to assure that amount of void exits the cell is bounded by what was in it calculated from the previous step. In this scheme, the combination of downwind and upwind fluxes has been considered in such a way that not only the solution stability is guaranteed, but also avoids the numerical diffusion.

3.3. Flux Corrected Transport (FCT) method

The FCT method is based on the idea to present a formulation which combines the upwind and downwind fluxes. This formulation aimed to leave out upwind numerical diffusion and instability of downwind scheme [15]. Idea of neighboring fluxes based on higher order translate scheme was first proposed by Boris and Book [16] and then developed by Zalesak [17] to multidimensional.

In this method calculations consist of some steps. First, an intermediate value of volume of fluid, α^{*}, must be defined based on a first-order scheme. **Figure 6** shows schematically the solution for a 1D governing equation of fluid volume fraction for cell i as:

$$\alpha_{i}^{*}=\alpha_{i}^{n}-\frac{1}{\delta x}\left(F_{i+\frac{1}{2}}^{L}-F_{i-\frac{1}{2}}^{L}\right) \tag{12}$$

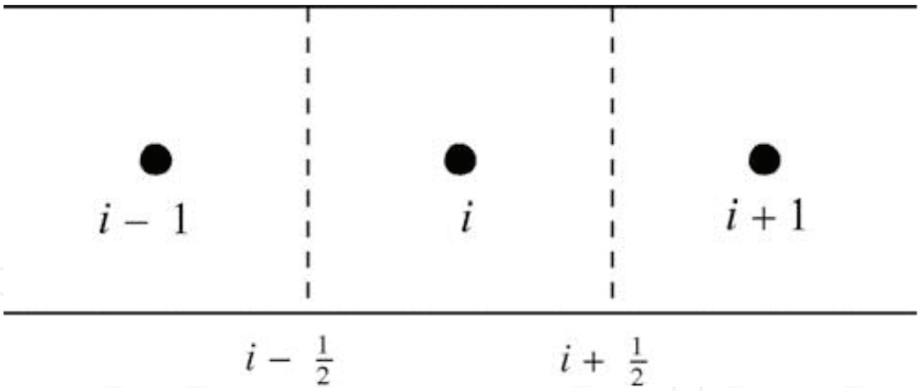

Figure 6. Three adjacent typical cells in FCT method.

where F^L is the flux in the downwind cell, and F_f is defined in the face f as:

$$F_f = U_f \delta t \alpha_f \tag{13}$$

Thereafter, an anti-diffusive flux is needed to be defined (F^L) in order to correct the diffusion of the previous step. This is the difference between upwind and downwind fluxes as:

$$F^A_{i+\frac{1}{2}} = F^H_{i+\frac{1}{2}} - F^L_{i+\frac{1}{2}} \tag{14}$$

To make this stable, a correction factor, q, is needed to modify the fluxes values. Finally, a value for fluid fraction in next time step is defined as:

$$\alpha_i^{n+1} = \alpha_i^* - \frac{\left(q_{i+\frac{1}{2}} F^A_{i+\frac{1}{2}} - q_{i-\frac{1}{2}} F^A_{i-\frac{1}{2}}\right)}{\delta x} \tag{15}$$

3.4. Youngs' method

This method was first proposed by Youngs in 1982 [18]. It was then developed by Rudman [19] with more details. In this method, at first the slope of the interface position is estimated. Then, the free surface is defined as a straight line with the slope of β in each cell of the numerical domain. The position of this line segment in each cell is defined such that the area reconstructed from the line and the perimeter of the cell is equal to the amount of volume of fluid, α. The geometry of the polygon from this reconstruction is used to calculate the flux transferred from the cell faces.

Assuming that $\alpha_{i,j}$ is predefined in every cell, the first step is to calculate first-order upwind fluxes. Then, the Youngs exiting fluxes of every cell can be calculated by considering the values in each cell. To do so, the angle β, between free surface and x-coordinate, must be calculated. Different methods can be used for calculation of β. One of them is first using the gradient function of fluid volume fraction for defining unit normal vector of the free surface, and then calculating β [20]. The method of defining the normal vector, however, can affect the accuracy of the final results. This formulation for uniform grid is as follows:

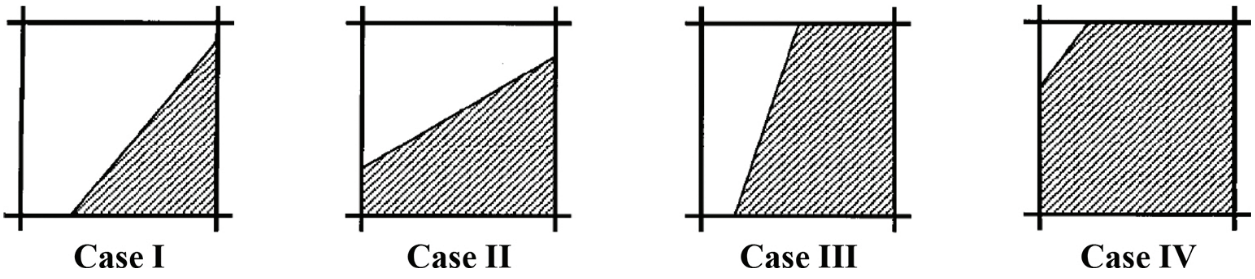

Figure 7. Four possible positions for free surface in Youngs' method [19].

$$n_{i,j}^{x}=\frac{\alpha_{i+1,j+1}+2\alpha_{i+1,j}+\alpha_{i+1,j-1}-\alpha_{i-1,j+1}-2\alpha_{i-1,j}-\alpha_{i-1,j-1}}{\delta x} \tag{16}$$

$$n_{i,j}^{y}=\frac{\alpha_{i+1,j+1}+2\alpha_{i,j+1}+\alpha_{i-1,j+1}-\alpha_{i+1,j-1}-2\alpha_{i,j-1}-\alpha_{i-1,j-1}}{\delta y} \tag{17}$$

Using components of the normal unit vector, the angle β can be calculated as:

$$\beta=\tan^{-1}\left(\frac{-n_x}{n_y}\right) \quad \left(-\pi<\beta\leq\pi\right) \tag{18}$$

The angle γ is also defined as:

$$\gamma=\tan^{-1}\left(\frac{\delta x}{\delta y}\tan\beta\right) \quad \left(0\leq\gamma\leq{}^{\pi}\!/_{2}\right) \tag{19}$$

It is possible to set $0 \leq \gamma \leq 90$ by rotating the cell. Therefore, there are only four possible positions for the free surface, which are depicted in **Figure 7**.

What is behind this conclusion is as follows:

```
$if\ \gamma < \pi/4$
    $if\ \alpha \le \frac{1}{2}\tan\gamma$
        $Case\ I$
    $elseif\ \ \alpha \le \left(1-\frac{1}{2}\tan\gamma\right)$
        $Case\ II$
    $else$
        $Case\ IV$
$else$
    $if\ \alpha \le \frac{1}{2}\cot\gamma$
        $Case\ I$
    $elseif\ \alpha \le \left(1-\frac{1}{2}\cot\gamma\right)$
        $Case\ III$
    $else$
        $Case\ IV$
```

	Case I	**Case II**
s_t	0	0
s_r	$\sqrt{2\alpha\tan\gamma}$	$\alpha + \frac{1}{2}\tan\gamma$
s_b	$\sqrt{2\alpha\cot\gamma}$	1
s_l	0	$\alpha - \frac{1}{2}\tan\gamma$
$if\ U_t > 0$	$if\ U_t\delta t \le (1-s_r)\delta y$ $\quad F_t = 0$ $else$ $\quad F_t = \frac{1}{2}[U_t\delta t - (1-s_r)\delta y]^2\cot\gamma$	$if\ U_t\delta t \le (1-s_r)\delta y$ $\quad F_t = 0$ $elseif\ U_t\delta t \le (1-s_l)\delta y$ $\quad F_t = \frac{1}{2}[U_t\delta t - (1-s_r)\delta y]^2\cot\gamma$ $else$ $\quad F_t = U_t\delta t\delta x - (1-\alpha)\delta x\delta y$
$if\ U_r > 0$	$if\ U_r\delta t \ge s_b\delta x$ $\quad F_r = \alpha\delta x\delta y$ $else$ $\quad F_r = \frac{1}{2}U_r\delta t\left(2 - {}^{U_r\delta t}\!/_{s_b\delta x}\right)s_r\delta y$	$F_r = U_r\delta t\left(s_r\delta y - \frac{1}{2}U_r\delta t\tan\gamma\right)$

	Case I	**Case II**

if $U_b>0$	*if* $U_b\delta t\geq s_r\delta y$
	$F_b=\alpha\delta x\delta y$
	else
	$F_b=\frac{1}{2}U_b\delta t\left(2-{}^{U_b\delta t}\!/_{s_r\delta y}\right)s_b\delta x$
		if $U_b\delta t\leq s_l\delta y$
		$F_b=U_b\delta t\delta x$
		elseif $U_b\delta t\leq s_r\delta y$
		$F_b=U_b\delta t\delta x-\frac{1}{2}(U_b\delta t-s_l\delta y)^2\cot\gamma$
		else
		$F_b=\alpha\delta x\delta y$

if $U_l>0$	*if* $U_l\delta t\leq(1-s_b)\delta x$
	$F_l=0$
	else
	$F_l=\frac{1}{2}[U_l\delta t-(1-s_b)\delta x]^2\tan\gamma$
		$F_l=U_l\delta t\left(s_l\delta y+\frac{1}{2}U_l\delta t\tan\gamma\right)$

	Case III	**Case IV**
s_t	$\alpha-\frac{1}{2}cot\gamma$	$1-\sqrt{2\alpha\cot\gamma}$
s_r	1	1
s_b	$\alpha+\frac{1}{2}cot\gamma$	1
s_l	0	$1-\sqrt{2\alpha\tan\gamma}$

if $U_t>0$	$F_t=U_t\delta t\left(s_t\delta x+\frac{1}{2}U_t\delta t\cot\gamma\right)$	*if* $U_t\delta t\geq(1-s_l)\delta y$
		$F_t=U_t\delta t\delta x-(1-\alpha)\delta x\delta y$
		else
		$F_t=U_t\delta t\left(s_t\delta x+\frac{1}{2}U_t\delta t\cot\gamma\right)$

if $U_r>0$	*if* $U_r\delta t\leq s_t\delta x$
	$F_r=U_r\delta t\delta y$
	elseif $U_r\delta t\leq s_b\delta x$
	$F_r=U_r\delta t\delta y-\frac{1}{2}(U_r\delta t-s_t\delta x)^2\tan\gamma$
	else
	$F_r=\alpha\delta x\delta y$
		if $U_r\delta t\leq s_t\delta x$
		$F_r=U_r\delta t\delta y$
		else
		$F_r=U_r\delta t\delta y-\frac{1}{2}\tan\gamma(U_r\delta t-s_t\delta x)^2$

if $U_b>0$	$F_b=U_b\delta t\left(s_b\delta x-\frac{1}{2}U_b\delta t\cot\gamma\right)$	*if* $U_b\delta t\leq s_l\delta y$
		$F_b=U_b\delta t\delta x$
		else
		$F_b=U_b\delta t\delta x-\frac{1}{2}(U_b\delta t-s_l\delta y)^2\cot\gamma$

	Case III	**Case IV**
if $U_l > 0$	*if* $U_l \delta t \le s_b \delta x$	*if* $U_l \delta t \ge (1 - s_t)\delta x$
	$F_l = 0$	$F_l = U_l \delta t \delta y - (1-\alpha)\delta x \delta y$
	elseif $U_l \delta t \le s_t \delta x$	*else*
	$F_l = \frac{1}{2}[U_l \delta t - (1 - s_b)\delta x]^2 \tan\gamma$	$F_l = U_l \delta t \left(s_l \delta y + \frac{1}{2} U_l \delta t \tan\gamma\right)$
	else	
	$F_l = U_l \delta t \delta y - (1-\alpha)\delta x \delta y$	

Table 1. Calculation of exiting flux in Youngs' method.

Four side fractions (s_l, s_b, s_r, s_t) for up, right, down, and left faces can be calculated with the selection of the free surface position in a cell. Thereafter, flow fluxes can be geometrically computed for each face (F_l, F_b, F_r, F_t) based on these side fractions. More details are presented in **Table 1**. In this table, positive value is set for velocities towards the outer edges of a cell, and there is no flux calculation for negative velocities into the cell.

3.5. Piecewise Linear Interface Calculation (PLIC) method

To solve fluid volume transfer equation with FDM or FVM, diffusion error in interface reconstruction occurs. This leads to poor modeling of free surfaces, specifically in the interface of two adjacent fluids with large density difference. PLIC is one of the methods to reconstruct the interface between fluids with second-order accuracy [20]. It can increase the accuracy of transferred flux estimation and geometric fluid distribution in each cell. In this method, unit normal vectors of the surface are calculated based on the volume fraction of fluid using Youngs' least square method as:

$$\vec{n}_{i,j} = -\left(\frac{\nabla \alpha_{i,j}}{\sqrt{\left(\nabla \alpha_{x,i,j}\right)^2 + \left(\nabla \alpha_{y,i,j}\right)^2}} \right) \tag{20}$$

where $(\nabla \alpha)$ is defined as:

$$(\nabla \alpha)_{i,j} = \frac{1}{4}\left[(\nabla \alpha)_{i-½,j-½} + (\nabla \alpha)_{i-½,j+½} + (\nabla \alpha)_{i+½,j+½} + (\nabla \alpha)_{i+½,j-½} \right] \tag{21}$$

$$(\nabla\alpha)_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{2}\left(\frac{\alpha_{i+1,j}-\alpha_{i,j}}{x_{i+1,j}-x_{i,j}} + \frac{\alpha_{i+1,j+1}-\alpha_{i,j+1}}{x_{i+1,j+1}-x_{i,j+1}}\right)\vec{i} + \frac{1}{2}\left(\frac{\alpha_{i,j+1}-\alpha_{i,j}}{y_{i,j+1}-y_{i,j}} + \frac{\alpha_{i+1,j+1}-\alpha_{i+1,j}}{y_{i+1,j+1}-y_{i+1,j}}\right)\vec{j} \tag{22}$$

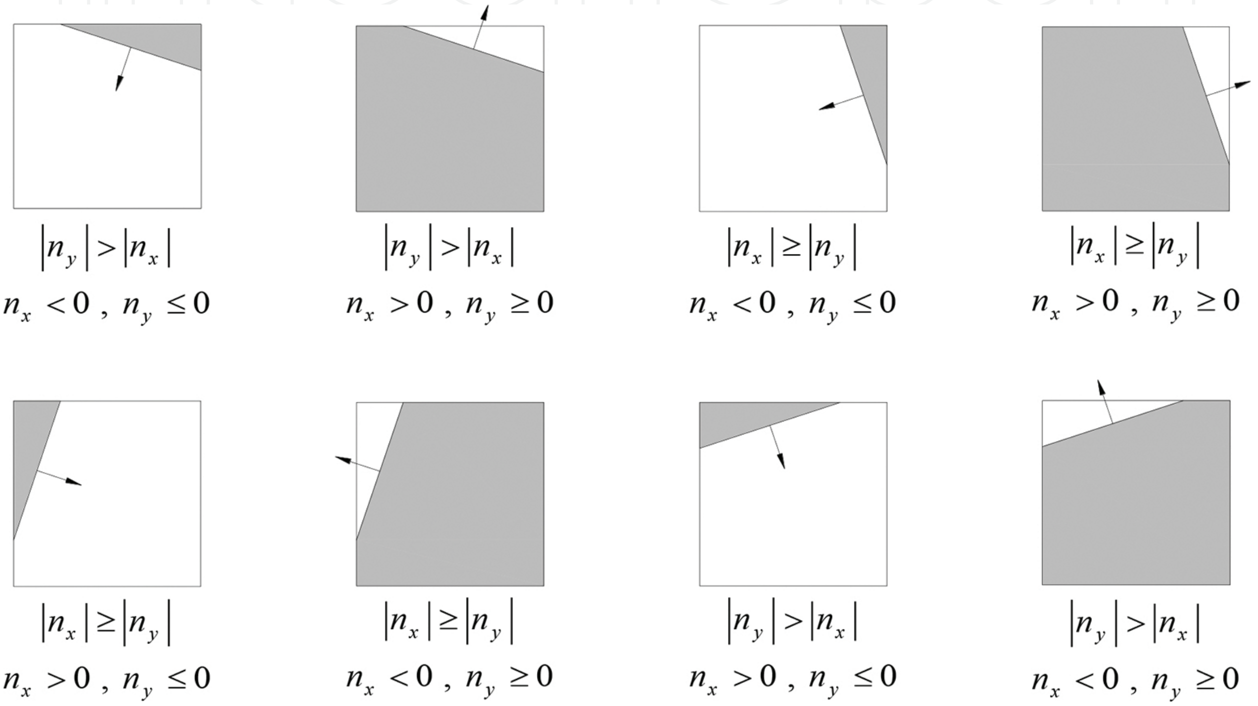

Figure 8. Different positioning of the interface for $(0 \le \theta \le \pi/4)$.

where θ is the angle between the normal vector and the horizontal coordinate varies between zero and 2π. For θ in the first one-eighth space $(0 \le \theta \le \pi/4)$, eight different conditions are possible for the position of free surfaces as illustrated in **Figure 8** [21]. All other situations can be achieved with a mirror reflection of the first quarter with respect to the x and y axes and bisectors between them. The exact position of the free surface is determined defining surface unit normal vector using volume fraction of fluid in each cell. To do this, extreme values of $\alpha_{\lim,1}$ and $\alpha_{\lim,2}$ are determined as:

$$\alpha_{\lim,1} = \frac{n_{\min}}{2n_{\max}} \quad , \quad \alpha_{\lim,2} = 1 - \alpha_{\lim,1} \tag{23}$$

in which $v\left(i+1,\ j-\frac{1}{2}\right)$ and $n_{\max}$ are calculated as:

$$n_{\min} = \min\left(|n_x|,|n_y|\right) \quad , \quad n_{\max} = \max\left(|n_x|,|n_y|\right) \tag{24}$$

$\alpha_{\lim,1}$ and $\alpha_{\lim,2}$ are shown in **Figure 9**.

When unit normal vector of a surface is defined, the true position of the interface can be easily determined using volume of fluid in each cell.

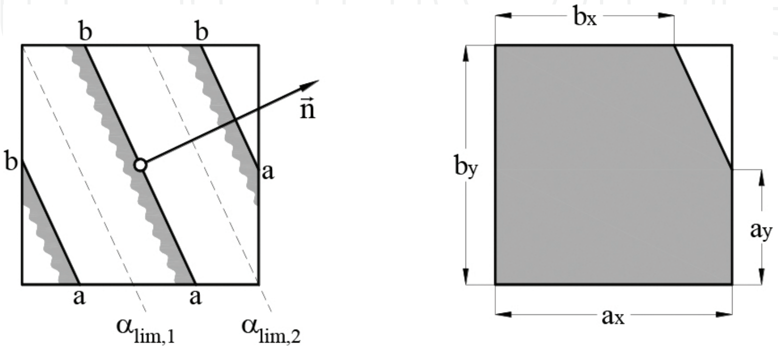

Figure 9. Various positions of an interface in a cell.

3.6. Higher order differencing schemes

Another method to reconstruct the interface between two fluids is to discretize the convection term using higher order differencing schemes or blended differencing scheme. The accuracy of less/non-diffusive schemes and compressive schemes was compared by Davis [22]. Less/non-diffusive schemes prevent the interface profile from being diffused. Compressive schemes not only prevent the interface from being diffused, but also omit any diffusion in the neighboring of the interface. Thus, they are considered as powerful tools for thin interface simulation.

Ghobadian [23] applied the higher order scheme proposed by Van Leer [24]. However, his results showed that this scheme has poor ability in terms of removing diffusion. Therefore, he proposed solutions for decreasing numerical diffusion. Other methods for omitting diffusion proposed by Pericleous and Chen [25] proved to be associated with interface diffusion. Although first-order upwind or downwind schemes lead to diffusion, higher order methods result in numerical fluctuations in the interface. There are other methods for reducing the interface as follows:

3.6.1. Compressive Interface Capturing Scheme for Arbitrary Meshes (CICSAM) scheme

The CICSAM scheme, presented by Ubbink, is a combined method to reduce the diffusion problem in interface modeling. This method imposes some limitations on the fluid fraction value. It is obvious that the value of a fluid in a cell should be constant in the absence of a source. The CICSAM approach presents an equation for free surface volume fraction as:

$$\tilde{\alpha}_f = \gamma_f \tilde{\alpha}_{f_{CBC}} + (1-\gamma_f)\tilde{\alpha}_{f_{UQ}} \tag{25}$$

where $\tilde{\alpha}_{f_{CBC}}$ is an index for transport bound which defines a bound as follows:

$$\tilde{\alpha}_{f_{CBC}} = \begin{cases} \min\left\{1, \dfrac{\tilde{\alpha}_D}{c_f}\right\} & when\ 0 \le \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & when\ \tilde{\alpha}_D \le 0,\ \tilde{\alpha}_D > 1 \end{cases} \tag{26}$$

where $\tilde{\alpha}_D$ is defined based on the following equation:

$$\tilde{\alpha}_D = \frac{\alpha_D - \alpha_U}{\alpha_A - \alpha_U} \tag{27}$$

$$\tilde{\alpha}_{f_{UQ}} = \begin{cases} \min\left\{\dfrac{8c_f\tilde{\alpha}_D + (1-c_f)(6\tilde{\alpha}_D + 3)}{8}, \tilde{\alpha}_{f_{CBC}}\right\} & when\ 0 \le \tilde{\alpha}_D \le 1 \\ \tilde{\alpha}_D & when\ \tilde{\alpha}_D < 0,\ \tilde{\alpha}_D > 1 \end{cases} \tag{28}$$

More details on determining $\tilde{\alpha}_f$ can be derived based on the combined schemes such as Normalized Variable Diagram (NVD) and discussed by Ubbink and Issa [26]. Accordingly, a local boundedness is defined for $\tilde{\alpha}_{f_{UQ}}$. In Eq. (25), γ_f is a weight factor which is defined as:

$$\gamma_f = \min\left\{k_\gamma \frac{\cos(2\theta_f)+1}{2}, 1\right\} \tag{29}$$

where k_γ is a constant, usually set to unity, and θ_f is the angle between free surface normal vector, $(\nabla\alpha)_D$, and the vector $\vec{d}_f$ such that it connects the center of the adjacent cells based on **Figure 10**, and defined as:

$$\theta_f = \arccos\left|\frac{(\nabla\alpha)_D . \vec{d}_f}{\left|(\nabla\alpha)_D\right| \left|\vec{d}_f\right|}\right| \tag{30}$$

The CICSAM method well satisfies the bounds defined within it, and can be accurately reconstruct the free surface. The basis of the method, however, is on the 1D equations and linearization, which makes it less accurate for 3D modeling reconstructions.

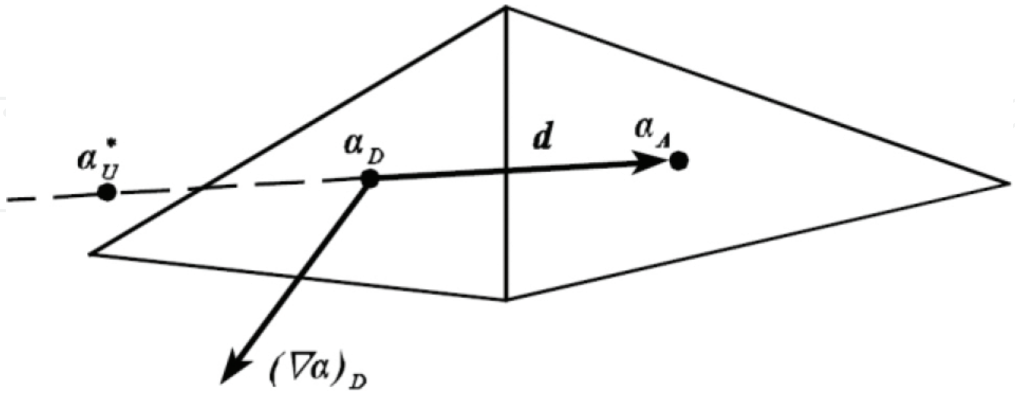

Figure 10. Calculation of the upwind value for an arbitrary mesh.

3.6.2 THOR scheme

This scheme is based on the CICSAM and switches smoothly between the upper bound of the universal limiter and ULTIMATE-QUICK, a combination of the universal limiter and QUICK, considering the angle between the interface and the direction of motion [27].

Analogous to CICSAM, this scheme is an algebraic advection scheme for the interface, which is designed for the implicit time advancing algorithm. In this method α_f is calculated using a weighting factor β_f as follows:

$$\alpha_f = \left(1-\beta_f\right)\alpha_D + \beta_f \alpha_A \tag{31}$$

where

$$\beta_f = \frac{\tilde{\alpha}_f - \tilde{\alpha}_D}{1-\tilde{\alpha}_D} \tag{32}$$

$\tilde{\alpha}_f$ and $\tilde{\alpha}_D$ can be calculated by Eqs. (25) and (27), respectively.

3.6.3. Higher Resolution Artificial Compressive (HiRAC) scheme

HiRAC scheme is another modification of the CICSAM method [28]. This newly proposed method tries to improve the computational efficiency and maintain the accuracy. In this method, the weighing factor, γ_f, of Eq. (29) can be redefined as:

$$\gamma_f = min\left(\left(\theta_f\right)^m, 1\right) \tag{33}$$

where θ_f was previously defined in Eq. (30). For m=2, the new formulation reduces to the weighting function of Ubbink and Issa [26]. As m increases, the interpolation becomes more biased towards the diffusive higher resolution scheme. It is shown that m=2 provides a good balance between the compressive and diffusive higher resolution schemes.

3.6.4. High Resolution Interface Capturing (HRIC) scheme

This method is somehow similar to CICSAM, which benefits from a combined interpolation scheme. In HRIC method, the difference between two upwind schemes is calculated based on the normal vector angle of the free surface as [29]:

$$\tilde{\alpha}_f^{**} = \begin{cases} \tilde{\alpha}_D & \tilde{\alpha}_D < 0 \\ 2\tilde{\alpha}_D & 0 < \tilde{\alpha}_D < 0.5 \\ 1 & 0.5 < \tilde{\alpha}_D < 1 \\ \tilde{\alpha}_D & 1 < \tilde{\alpha}_D \end{cases} \tag{34}$$

$$\tilde{\alpha}_f^{*} = \begin{cases} \tilde{\alpha}_f^{**} & c_f < 0.3 \\ \tilde{\alpha}_D + \left(\tilde{\alpha}_f^{**} - \tilde{\alpha}_D\right)\dfrac{0.7 - c_f}{0.4} & 0.3 < c_f < 0.7 \\ \tilde{\alpha}_D & 0.7 < c_f \end{cases} \tag{35}$$

The portion of each of the two terms in the above equations can be defined as:

$$\tilde{\alpha}_f = \tilde{\alpha}_f^{*}\sqrt{\cos\theta} + \tilde{\alpha}_D\,(1 - \sqrt{\cos\theta}) \tag{36}$$

In this way, α_f is discretized based on the neighboring cells.

It should be noted that an improved scheme of HRIC, called Flux-Blending Interface-Capturing Scheme (FBICS), has been recently proposed. In this method, analogous to CICSAM and HRIC, the difference between two upwind schemes is calculated based on the normal vector angle of the free surface. Based on FBICS, Eq. (33) can be reformulated to obtain a more accurate scheme as:

$$\tilde{\alpha}^{*}_{f,FBICS} = \begin{cases} 3\tilde{\alpha}_D\tilde{\alpha}_D \leq {}^{1}\!/\!_{8} \\ \tilde{\alpha}_D + \frac{1}{4}\,{}^{1}\!/\!_{8} < \tilde{\alpha}_D < {}^{3}\!/\!_{4} \\ 1\,{}^{3}\!/\!_{4} < \tilde{\alpha}_D < 1 \\ \tilde{\alpha}_D\tilde{\alpha}_D \leq 0\, or\, \tilde{\alpha}_D \geq 1 \end{cases} \tag{37}$$

Some other modifications are also proposed by Tsui et al. [30].

3.6.5. Switching Technique for Advection and Capturing of Surfaces (STACS) method

One of the drawbacks of HRIC and CICSAM schemes is high Courant numbers. Both methods lack a proper switching strategy to accurately model the interface when Courant number increases. The Courant number, *Cn*, in HRIC method can be written as follows:

$$\tilde{\alpha}_f = \tilde{\alpha}^{*}_f + \left(\tilde{\alpha}^{*}_f - \tilde{\alpha}_D\right)\frac{0.7 - Cn}{0.7 - 0.3} \tag{38}$$

which is suitable for Courant numbers between 0.3 and 0.7. For a *Cn* below 0.3 the scheme is not modified, while for a Courant number above 0.7 the upwind scheme is used. This is true for CICSAM when the *Cn* is equal to unity.

STACS method has been proposed to improve the accuracy and stability of the results specifically in high Courant numbers by Darwish and Moukalled [31]. It uses an implicit transient discretization, i.e. no transient bounding is applied, and in order to minimize the stepping behavior of HRIC scheme, a modification is proposed. In this method, applying $\cos^4\theta$ term is designed instead of $\sqrt{\cos\theta}$ in Eq. (36) as follows:

$$\tilde{\alpha}_f = \tilde{\alpha}^{*}_{f,sup} cos^4\theta + \tilde{\alpha}^{*}_{f,stoic}\left(1 - cos^4\theta\right) \tag{39}$$

where $\tilde{\alpha}^{*}_{f,sup}$ and $\tilde{\alpha}^{*}_{f,stoic}$ are calculated as follows:

$$\tilde{\alpha}^*_{f,sup} = \begin{cases} \tilde{\alpha}_D \tilde{\alpha}_D \leq 0 \\ 10 < \tilde{\alpha}_D < 1 \\ \tilde{\alpha}_D \tilde{\alpha}_D \geq 1 \end{cases} \qquad \tilde{\alpha}^*_{f,stoic} = \begin{cases} \tilde{\alpha}_D \tilde{\alpha}_D \leq 0 \\ \frac{1}{2} + \frac{1}{2}\tilde{\alpha}_D 0 < \tilde{\alpha}_D < \frac{1}{2} \\ \frac{3}{8} + \frac{3}{4}\tilde{\alpha}_D \frac{1}{2} < \tilde{\alpha}_D < \frac{5}{6} \\ 1\frac{5}{6} < \tilde{\alpha}_D < 1 \\ \tilde{\alpha}_D \tilde{\alpha}_D \geq 1 \end{cases} \tag{40}$$

This enables a rapid but smooth switching strategy that works very well, especially where the normal to the free-surface face is not along the grid direction.

3.6.6. Inter-gamma scheme

In this method, presented by Jasak and Weller [32], free surface compression is modeled using additional compressive artificial terms.

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot \left(\mathbf{U}\gamma\right) + \nabla \cdot \left(\mathbf{U}^r \gamma \left(1-\gamma\right)\right) = 0 \tag{41}$$

where $\mathbf{U}^r$ is a velocity field for compressing the free surface. This artificial term is activated only in the presence of free surface because of having the term $\gamma(1-\gamma)$. The solution to this equation is bounded from zero to unity with the inter-gamma scheme. Eq. (40) can be rewritten as:

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot \left(\phi[\gamma]_{f(\phi,S)}\right) + \nabla \cdot \left(\phi_{r_b}[\gamma]_{f\left(\phi_{r_b},S\right)}\right) = 0 \tag{42}$$

where $\phi = \mathbf{S} \cdot \mathbf{U}_f$ is the volume flux and $\phi_{r_b} = (1-\gamma)_{f(-\phi^r,S)}\phi^r$. Eq. (43) is used to calculate $\varnothing^r$ as:

$$\phi^r = K_c n^* \max\left\{\frac{\left|n^*\phi\right|}{|S|^2}\right\} \tag{43}$$

where n^* is the free surface normal unit vector and K_c is an adjustable coefficient, with an appropriate value of 1.5, which defines the compression rate of free surface. The inter-gamma scheme is also similar to CICSAM and is based on a donor-acceptor equation and normalized variable diagram, NVD. The formulation is as:

$$\tilde{\alpha}_f = \begin{cases} \tilde{\alpha}_D & \tilde{\alpha}_D < 0,\ \tilde{\alpha}_D > 1 \\ -2\tilde{\alpha}_D^2 + 3\tilde{\alpha}_D & 0 < \tilde{\alpha}_D < \frac{1}{2} \\ 1 & \frac{1}{2} < \tilde{\alpha}_D < 1 \end{cases} \tag{44}$$

Figure 11 shows the NVD for inter-gamma scheme.

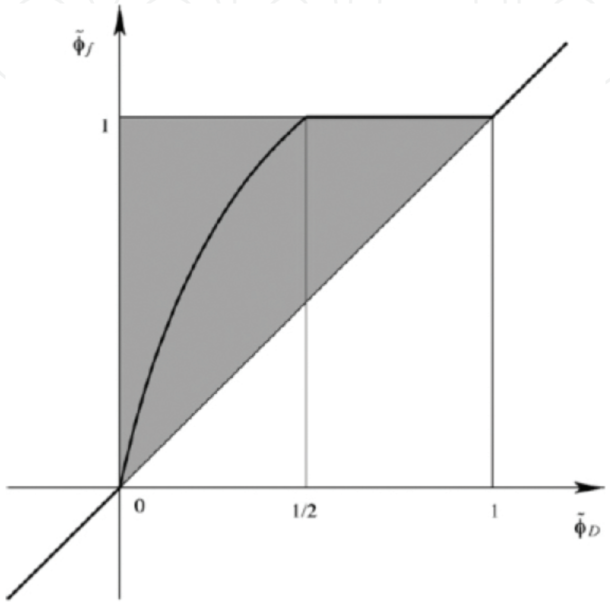

Figure 11. NVD for inter-gamma scheme.

3.7. Integrated methods

As mentioned before, volume of fluid is among the most popular methods in free surface modeling. Having in mind that this method is based on defining a discontinuous function, the color function, there is not a unique form for free surface. Therefore, it is required to reconstruct the free surface using volume of fraction function. In one hand, VOF method satisfies the conservation of mass while it is unable to calculate free surface parameters including curvature radius and normal unit vector directly. On the other hand, in level set methods as the distance function is smooth, the surface geometry can be easily calculated, while satisfying the conservation of mass is very demanding. In order to resolve the problems of level set methods, a number of different researches have been conducted. For example, higher order schemes were proposed to improve the conservation of continuity equation by Peng et al. [33]. Adaptive mesh refinement techniques were also proposed to increase the accuracy of the local mesh consistency. In 2009, an integrated method known as hybrid Particle Level Set (PLS) was proposed to improve the accuracy of the results. However, the problem still remained in relation with mass conservation.

In order to take the advantages of both methods and eliminate their disadvantages, integration of volume of fluid and level set methods was proposed in a new scheme known as coupled level set and volume-of-fluid (CLSVOF) method to model two-phase incompressible flows by Sussman and Pucket [34]. It should be noted that although accurate, this method cannot be easily employed, because these two methods, VOF and level set, should be individually solved and their effects need to be coupled based on the reconstructed interface.

4. Calculating surface tension

Defining the pressure difference inserted on the surface of two fluids with different densities and tension stresses is one of the most demanding problems in fluid mechanics. One method to do this is the Pressure Calculation based on the Interface Location (PCIL) method which is presented here. Surface tension, that changes the value of variables in momentum equations, imposes a discontinuity at the position of the interface between two fluids [21].

Stress from surface tension inserts a force upon the interface. The resultant force is perpendicular to the surface and its curvature is dependent on the geometry of the surface. Surface tension can be considered in two ways. In the first approach, it is considered as a boundary condition in the equations for the surface. This needs using an iterative method for true approximation of pressure, which in result, increases the time and cost of calculation and consequently makes it inefficient. In order to address this problem, some other methods have been proposed in which the precise calculation of interface position is not necessary. In these methods, the direct force of surface tension has been replaced with the body force in the momentum equation. The Continuum Surface Force (CSF) method is a base method for calculation of body forces of fluid surface tension [2]. The body forces can be considered to act smoothly on a narrow strip of cells in interface zone. In this method the surface stresses are replaced with the body forces which are calculated as:

$$\mathbf{F}_s = \sigma \int_S \kappa \mathbf{n} \delta\left(\mathbf{x} - \mathbf{x}_s\right) dS \tag{45}$$

where σ, κ, $\mathbf{n}$, $\delta(\mathbf{x}-\mathbf{x}_s)$ and $\mathbf{X}_s$ are surface tension force, curvature of the surface, normal unit vector, Dirac delta function, and the position of a given base point on the interface S, respectively. This equation has been discretized for numerical methods of two-phase fluids.

Another approach based on CSF method was proposed by Torrey et al. [7] called Continuum Surface Stress (CSS), in which body forces of CSF method were replaced by tension tensors of surface tension based on the following equation:

$$\mathbf{F}_s = \sigma \kappa \mathbf{n} \delta_s = -\nabla \cdot \mathbf{T}, \quad \mathbf{T} = -\sigma\left(\mathbf{I} - \mathbf{n} \otimes \mathbf{n}\right)\delta_s \tag{46}$$

where **I** and **T** are unit tensor and tangential tension tensor of the interface respectively.

It should be mentioned that employing CSF or CSS methods has some drawbacks. For instance, spurious velocities of the thinner fluid near the interface is one the reported problems.

A number of researches have been performed in order to resolve the problem of spurious velocities [35, 36]. In one approach, using virtual particles moving along with the surface could improve the results [37]. One of the latest methods presented in this field is PCIL. This method shows that having more precise border cells and calculating their associated pressure based on the momentum equation can lead to significant reduction of bothersome flows near this region. PCIL is a simple and efficient method of calculating free surfaces. The total pressure on the left side of the cell can be calculated as (see **Figure 12**):

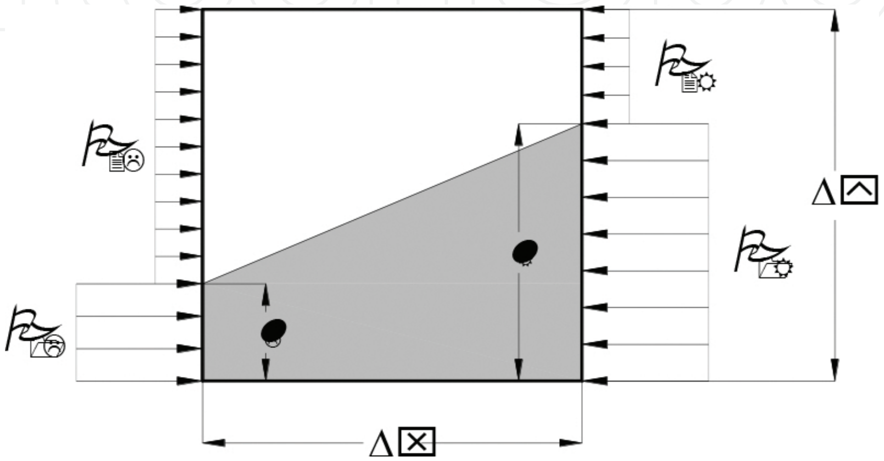

Figure 12. Cells on the interface in contact with left and right cells of the free surface [38].

$$P_L = P_{1L}\frac{l_L}{\Delta y} + P_{2L}\frac{\Delta y - l_L}{\Delta y} = P_{1L}H_L + P_{2L}\left(1 - H_L\right) = P_{2L} + H_L\left(P_{1L} - P_{2L}\right) \tag{47}$$

where P_L is the mean pressure on the left side of the given cell. In the same way, mean pressure can be calculated for other faces of the cell as follows:

$$P_R = P_{2R} + H_R\left(P_{1R} - P_{2R}\right) \tag{48}$$

$$P_T = P_{2T} + H_T\left(P_{1T} - P_{2T}\right) \tag{49}$$

$$P_B = P_{2B} + H_B\left(P_{1B} - P_{2B}\right) \tag{50}$$

where H is a dimensionless number which shows the position of free surface in different directions. For faces completely immersed in the main fluid, H is equal to 1, and for those completely in the secondary fluid, H is zero. For other cases, H varies between zero and unity. Note that for 3D models, it is just needed to replace edge faces with surfaces faces.

On the other hand, the change in pressure P_s in every point of the interface is calculated as:

$$P_s = P_1 - P_2 = \sigma\kappa \tag{51}$$

Accordingly, the above equation can be reformulated for pressure in the *K*th face of every common cell as follows:

$$P_k = P_{2k} + H_k\sigma\kappa \tag{52}$$

where the second term introduces the normal force of the surface tension per unit area of the interface. This can be presented in the vector form as:

$$\mathbf{F}_s = H\sigma\kappa\mathbf{n} \tag{53}$$

where $\mathbf{F}_s$ is the surface tension force vector.

One of the most fundamental steps to perform surface tension calculations is defining the curvature of the interface. Defining this curvature is not so demanding as long as the precise position of the interface is known. However, using volumetric tracing methods and equivalent alternatives representing the interface position make the estimation of the curvature cumbersome.

The method of volume of fluid presented by Hirt and Nichols [6] is one of the earliest methods in this field. In this method, a curve $y(x)$ is fitted to the nine neighboring cells of the interface. In this way, summation of the volume of fluid of three cells located in a column creates a value for y. By fitting parabolic curve to the values of three neighboring columns and then two times differentiating of these values, one can define the curvature of the surface. This method, however, suffers from low accuracy and some limiting conditions. Another method is presented by Chorin [39] in which a circle is defined based on trial and error in order to satisfy the nine-cell set in the best possible way. Thereafter, curvature of the surface can be defined on the basis of the calculated circle. However the main problem of this method is its dependency on the several times of trials and errors in order to calculate the best way of estimating the circle, the thing that makes it practically inefficient. Ashgriz and Poo [40] proposed another method in which a parabolic curve is fitted to the 9 or 25 neighboring cells. This method is more accurate than the previous one; however, its applications are very limited to specific cases.

The value of the surface curvature can be defined based on the distance function as:

$$\kappa = -\nabla\cdot\mathbf{n} = -\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|} \tag{54}$$

To discretize the above equation, it is required to first calculate the normal vector of the surface based on **Figure 13** and the following relations, and consequently estimate the curvature:

$$div_{x,i+\frac{1}{2},j} = 2\frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x_{i+1} + \Delta x_i}, \quad div_{y,i+\frac{1}{2},j} = \frac{\phi_{i+1,j} - \phi_{i,j-1} + \phi_{i+1,j+1} - \phi_{i+1,j-1}}{\Delta y_{j+1} + 2\Delta y_j + \Delta y_{j-1}} \tag{55}$$

$$n_{x,i+\frac{1}{2},j} = \frac{div_{x,i+\frac{1}{2},j}}{\left|\mathbf{n}_{i+\frac{1}{2},j}\right|}, \quad \left|\mathbf{n}_{i+\frac{1}{2},j}\right| = \sqrt{div^2_{x,i+\frac{1}{2},j} + div^2_{y,i+\frac{1}{2},j}} \tag{56}$$

$$\kappa_{i,j} = -\frac{n_{x,i+\frac{1}{2},j} - n_{x,i-\frac{1}{2},j}}{\Delta x_i} - \frac{n_{y,i,j+\frac{1}{2}} - n_{x,i,j-\frac{1}{2}}}{\Delta y_j} \tag{57}$$

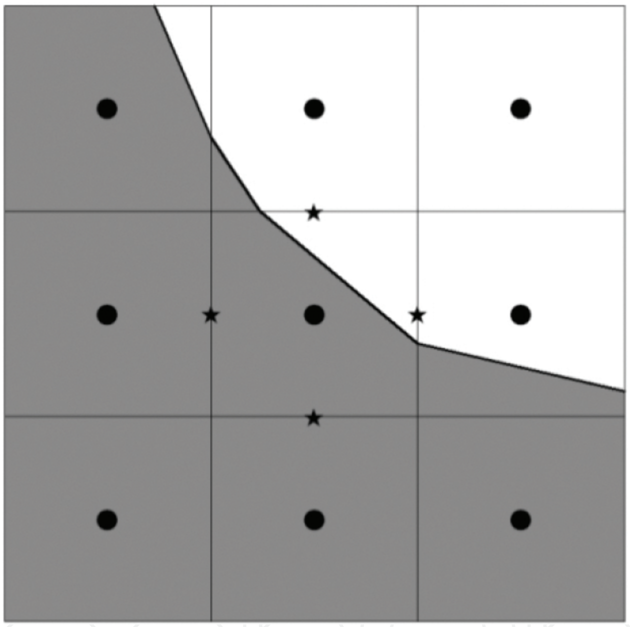

Figure 13. Position of the normal vectors of the surface in the cell faces.

One of the advantages of this approach, the level-set method, is using a distance function which is smooth and uniform, so that it increases the simplicity of the calculation and accuracy of the results.

The value of the body force from surface tension of the cell faces can be calculated in CSF method as:

$$\mathbf{F}_s = \sigma\kappa\mathbf{n}\delta_s = \sigma\kappa\mathbf{n}\frac{\left|\nabla\bar{\alpha}\right|}{[\alpha]} \tag{58}$$

where δ_s is the delta Dirac function which is infinite on the interface and zero otherwise, and its integration is equal to unity. The bar sign in Eq. (58) shows a smoothed (or filtered) value of volume fraction. The bracket sign shows the difference between maximum and minimum of volume of the fluid fraction. In CSF method, this function is estimated using $|\nabla\bar{\alpha}|/[\alpha]$. Some other references use the following equation to improve the results' accuracy [41]:

$$\mathbf{F}_s = \sigma\kappa\mathbf{n}\delta_s = \sigma\kappa\mathbf{n}\frac{|\nabla\bar{\alpha}|}{[\alpha]}\frac{\rho}{[\rho]} \tag{59}$$

in which the ratio of the densities is inserted in order to reduce spurious velocities of the thinner fluid. The discretized version of Eq. (58) can be obtained as follows:

$$F_{x,i+\frac{1}{2},j} = 2\sigma\kappa\frac{\bar{\alpha}_{i+1,j}-\bar{\alpha}_{i,j}}{\Delta x_{i+1}+\Delta x_i}\frac{\rho_{i+\frac{1}{2},j}}{\rho_{\max}-\rho_{\min}} \tag{60}$$

Based on what was discussed for PCIL method, the following relation is adopted to the present method:

$$F_{x,i+\frac{1}{2},j} = 2\sigma\kappa\frac{\bar{\alpha}_{i+1,j}-\bar{\alpha}_{i,j}}{\Delta x_{i+1}+\Delta x_i}H_{i+\frac{1}{2},j} \tag{61}$$

It can be seen that in this equation, the ratio of densities is replaced by the variable H.

5. Parametric method for calculation of curvature of free surfaces

This newly proposed method is based on two sub-models, the Four-Point Method (FPM) and the Three-Line Method (TLM). In the former sub-model, a curve is fitted to the intersection of the points of grid lines for central and two neighboring cells, while the latter fits a curve to the free surface so that the distance between the curve and its linear interface approximation is minimized [42].

5.1. The Four-Point Method (FPM)

In the four-point method, free surface (as illustrated in **Figure 14**) is approximated using a continuous function $f(\cdot)\in C^2(x_1, x_4)$ (a set of functions with continuous second derivation) so that the distance between the function and the points is minimized according to Eq. (62), and variations of curvature are bounded based on Eq. (63). In this case, the radius curvature can be calculated as in Eq. (64).

$$\inf_{f(\cdot)\in C^2(x_1,x_4)} I\left(f(\cdot)\right)=\sum_{i=1}^{4}\left|f(x_i)-y_i\right| \tag{62}$$

$$S.t.\begin{cases}\left|\kappa(x)-C_1\right|\le\varepsilon & x\in(x_1,x_2]\\ \left|\kappa(x)-C_2\right|\le\varepsilon & x\in(x_2,x_3]\\ \left|\kappa(x)-C_3\right|\le\varepsilon & x\in(x_3,x_4]\\ C_i\ge 0 & i=1,2,3\end{cases} \tag{63}$$

$$\kappa(x)=\frac{\left|\ddot{f}(x)\right|}{\sqrt{1+\dot{f}^2(x)}} \tag{64}$$

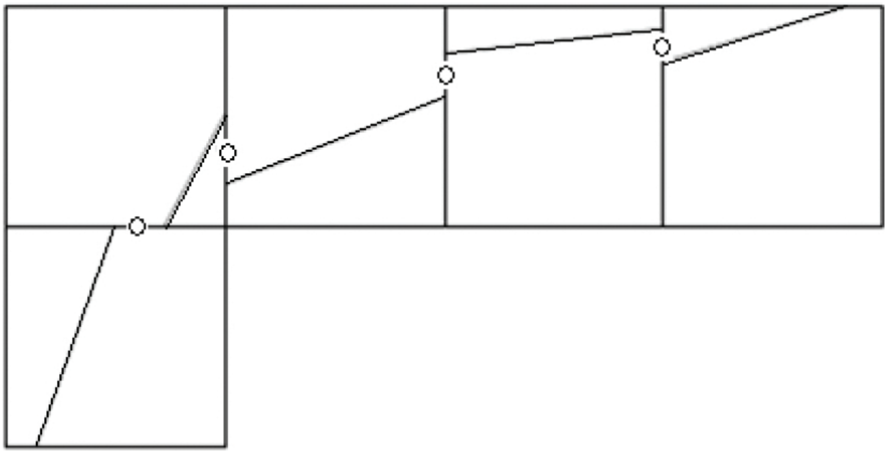

Figure 14. Free surface modeling in FPM.

where ε is a small arbitrary given number.

In this method, the desired function is approximated using an n-degree polynomial function with unknown constant coefficients. Therefore, we have:

$$\inf I\left(P_n(\cdot)\right)=\sum_{i=1}^{4}\left|P_n(x_i)-y_i\right| \tag{65}$$

$$\kappa_n(x)=\frac{\left|\ddot{P}_n(x)\right|}{\sqrt{1+\dot{P}_n^2(x)}} \tag{66}$$

where κ is the curvature of the free surface.

It is supposed that Q is the set of $f(\cdot)\in C^2(x_1,\, x_4)$ such that Eq. (62) is feasible and Q(n) is the set of $P_n(.)$ such that Eq. (64) is feasible. Then, the following theorem proves that the sequence of solutions for Eqs. (65) and (66) converges to the solution of Eqs. (62) and (64) as n goes towards infinity.

Theorem 1: if $\eta = inf_Q I(f(.))$ and $\eta(n) = inf_{Q_n} I(P_n(.))$ then $\eta = lim_{n\to\infty}\eta(n)$.

Proof. It is obvious that $Q(1)\subset Q(2)\subset \ldots \subset Q$, then $\eta(1)\geq\eta(2)\geq\ldots\geq\eta$. Therefore, $\{\eta(n)\}$ is a non-increasing and bounded sequence. Then it converges to a number called ξ. Set $W = \bigcup_{n=1}^{\infty} Q(n)$; therefore, $\inf_{Q_n} I(P_n(\cdot)) = \xi$. Since $W\subset Q$, then $\xi\geq\eta$. By the properties of infimum, for every $\varepsilon>0$, there exists $f(\cdot)\in Q$ such that:

$$\eta < I\left(f\left(\cdot\right)\right) < \eta + \varepsilon \tag{67}$$

As we have $f(\cdot)\in C^2(x_1,\, x_4)$, there is a set of polynomials such that $\{P_n(\cdot)\}$, $\{\dot{P}_n(\cdot)\}$, $\{\ddot{P}_n(\cdot)\}$ are uniformly convergent. Therefore, there exists a natural number such that for every $n\geq N$ we have:

$$\left\|P_n\left(\cdot\right) - f_n\left(\cdot\right)\right\|_\infty < \delta \tag{68}$$

$$\left\|\dot{P}_n\left(\cdot\right) - \dot{f}_n\left(\cdot\right)\right\|_\infty < \delta \tag{69}$$

$$\left\|\ddot{P}_n\left(\cdot\right) - \ddot{f}_n\left(\cdot\right)\right\|_\infty < \delta \tag{70}$$

Now, it is claimed that there is an $N_1\geq N$ such that for $i=1,\,2,\,3,\,4,$ the relation $|\,\kappa_n(x) - C_i\,| \leq \varepsilon;\ \forall\, x\in(x_i,\, x_{i+1}]$ is true. Since otherwise for every $n\geq N$, there is a $|\,\kappa_n(x) - C_i\,| \leq \varepsilon;\ \forall\, x\in(x_i,\, x_{i+1}]$. Therefore, $\lim_{n\to\infty}|\,\kappa_n(x) - C_i\,| > \varepsilon$ which contradicts the assumption of $f(\cdot)\in Q$. Thus, $P_N(\cdot)\in Q(N_1)\subset W\subset Q$. Based on what was mentioned, $|\,I(P_{N_1}(\cdot)) - I(f(\cdot))\,| < \varepsilon$ or $I(P_{N_1}(\cdot)) < I(f(\cdot)) + \varepsilon < \eta + 2\varepsilon$ or $\eta\leq\xi + 2\varepsilon$; therefore, $\xi=\eta$, or $\lim_{n\to\infty}\eta(n) = \eta$.

Thus, our aim is to solve Eqs. (63) and (64), and one can write them in the following forms:

$$\inf I\left(P_n\left(\cdot\right)\right)$$

$$S.t.\begin{cases} \int_{x_1}^{x_2} \left\| \kappa_n(x) - C_1 \right| - \varepsilon + \left\| \kappa_n(x) - C_1 \right| - \varepsilon \right\| dx = 0 \\ \int_{x_2}^{x_3} \left\| \kappa_n(x) - C_2 \right| - \varepsilon + \left\| \kappa_n(x) - C_2 \right| - \varepsilon \right\| dx = 0 \\ \int_{x_3}^{x_4} \left\| \kappa_n(x) - C_3 \right| - \varepsilon + \left\| \kappa_n(x) - C_3 \right| - \varepsilon \right\| dx = 0 \\ C_i \geq 0 \qquad\qquad i = 1,2,3, n = 1,2,\ldots \end{cases} \tag{71}$$

or

$$\inf I\left(P_n(\cdot)\right)$$

$$S.t = \begin{cases} \frac{h_1}{2}\left[E_{1n}(x_1) + E_{1n}(x_1 + h) + \ldots + E_{1n}(x_1 + m_1 h)\right] = 0 \\ \frac{h_2}{2}\left[E_{2n}(x_2) + E_{2n}(x_2 + h) + \ldots + E_{2n}(x_2 + m_2 h)\right] = 0 \\ \frac{h_3}{2}\left[E_{3n}(x_3) + E_{3n}(x_3 + h) + \ldots + E_{3n}(x_3 + m_3 h)\right] = 0 \\ C_i \geq 0 \qquad\qquad i = 1,2,3. \quad n = 1,2,\ldots \end{cases} \tag{72}$$

such that $E_{in}(x) = \| \kappa_n(x) - C_i | - \varepsilon + \| \kappa_n(x) - C_i | - \varepsilon \|$.

Now, the intervals $[x_1, x_2]$, $[x_2, x_3]$ and $[x_3, x_4]$ are divided into three equal sections, m_1, m_2, m_3, respectively. This means that $h_1 = \frac{x_2 - x_1}{m_1}$, $h_2 = \frac{x_3 - x_2}{m_2}$ and $h_3 = \frac{x_4 - x_3}{m_3}$. Thus, using a numerical integration method such as trapezoidal rule, the problem above can be reformulated as:

$$\inf \sum_{i=1}^{4} \left| P_n(x_i) - y_i \right|$$

$$S.t = \begin{cases} \frac{h_1}{2}\left[E_{1n}(x_1) + E_{1n}(x_1 + h) + \ldots + E_{1n}(x_1 + m_1 h)\right] = 0 \\ \frac{h_2}{2}\left[E_{2n}(x_2) + E_{2n}(x_2 + h) + \ldots + E_{2n}(x_2 + m_2 h)\right] = 0 \\ \frac{h_3}{2}\left[E_{3n}(x_3) + E_{3n}(x_3 + h) + \ldots + E_{3n}(x_3 + m_3 h)\right] = 0 \\ C_i \geq 0 \qquad\qquad i = 1,2,3. \quad n = 1,2,\ldots \end{cases} \tag{73}$$

This is a nonlinear set of equations and can be easily solved using Matlab or Lingo software.

5.2. The Three-Line Method (TLM)

In this method as illustrated in **Figure 15**, the main goal is to find a function as $f(\cdot)\in C^2(x_1, x_4)$ within $x\in[x_j, x_{j+1}]$ such that the distance between the function $f(x)$ and the line $L_i(x)$ which connects the given points (x_i, y_i) and (x_{i+1}, y_{i+1}) is minimized for $i=1, 2, 3$. Thus, variation of curvature is bounded according to Eq. (74):

$$\inf I\left(f\left(\cdot\right)\right)=\sum_{i=1}^{3}\int_{x_i}^{x_{i+1}}\left|f\left(x\right)-L_i\left(x\right)\right|dx$$

$$S.t.\begin{cases}\left|\kappa\left(x\right)-C_1\right|\le\varepsilon & x\in\left(x_1,x_2\right]\\ \left|\kappa\left(x\right)-C_2\right|\le\varepsilon & x\in\left(x_2,x_3\right]\\ \left|\kappa\left(x\right)-C_3\right|\le\varepsilon & x\in\left(x_3,x_4\right]\\ C_i\ge 0 & i=1,2,3\end{cases}\tag{74}$$

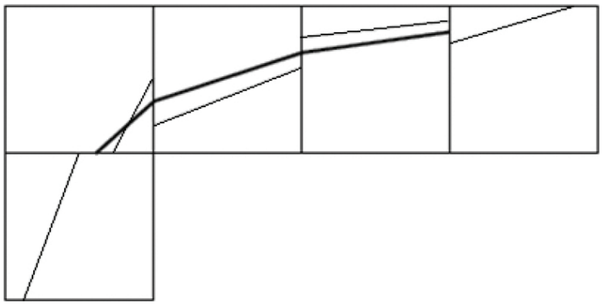

Figure 15. Free surface modeling by TLM.

Similar to what was discussed in the four-point method, in this method the function $f(\cdot)$ can be replaced with an n-degree polynomial, $P_n(\cdot)$ as below:

$$\inf I\left(P_n\left(\cdot\right)\right)=\sum_{i=1}^{3}\int_{x_i}^{x_{i+1}}\left|P_n\left(x\right)-L_i\left(x\right)\right|dx$$

$$S.t.\begin{cases}\left|\kappa_n\left(x\right)-C_1\right|\le\varepsilon & x\in\left[x_1,x_2\right]\\ \left|\kappa_n\left(x\right)-C_2\right|\le\varepsilon & x\in\left(x_2,x_3\right]\\ \left|\kappa_n\left(x\right)-C_3\right|\le\varepsilon & x\in\left(x_3,x_4\right]\\ C_i\ge 0 & i=1,2,3,\quad n=1,2,\ldots\end{cases}\tag{75}$$

Theorem 2: Sequence of the solution of Eq. (75) converges to the solution of Eq. (74).

Proof: The method of proof of this theorem is similar to the previous theorem. In the same approach of FPM, the following problem is achieved:

$$\inf I\left(P_n(\cdot)\right)=\sum_{i=1}^{3}\frac{h_i}{2}\begin{pmatrix}\left|P_n(x_1)-L_i(x_1)\right|+2\left|P_n(x_1+h)-L_i(x_1+h)\right|+\ldots\\ +2\left|P_n(x_1+(m_i-1)h)-L_i(x_1+(m_i-1)h)\right|+\\ \left|P_n(x_1+m_ih)-L_i(x_1+m_ih)\right|\end{pmatrix}$$

$$S.t=\begin{cases}\frac{h_1}{2}\left[E_{1n}(x_1)+E_{1n}(x_1+h)+\ldots+E_{1n}(x_1+m_1h)\right]=0\\ \frac{h_2}{2}\left[E_{2n}(x_2)+E_{2n}(x_2+h)+\ldots+E_{2n}(x_2+m_2h)\right]=0\\ \frac{h_3}{2}\left[E_{3n}(x_3)+E_{3n}(x_3+h)+\ldots+E_{3n}(x_3+m_3h)\right]=0\\ C_i\geq 0 \qquad\qquad i=1,2,3. \quad n=1,2,\ldots\end{cases} \tag{76}$$

The steps of using the above equations are as follows:

Step 1: Read ε, and set n=1.

Step 2: Solve Eq. (73) or (76) in the FPM or the TLM, respectively.

Step 3: If the previous step is infeasible, set n=n+1, and go to step 2, else set the value of target function in I_n.

Step 4: Set n=n+1, and solve Eq. (73) or (76) in the FPM or the TLM, respectively.

Then set the value of target function in I_n.

Step 5: If $|I_n-I_{n-1}|>\varepsilon_1$ then go to step 4, else I_n is the final answer.

6. Conclusions

In this chapter volume of fluid (VOF) scheme was introduced. This is one of the most effective methods employed in the simulation of two fluid flows interfaces with dramatic changes in density and viscosity. . These interfaces are represented implicitly by the values of a color function which is the fluid volume fraction. The advantage of the method is its ability to deal with arbitrarily shaped interfaces and to cope with large deformations, as well as interface rupture and coalescence in a natural way. In VOF the mass is rigorously conserved, provided the discretization is conservative. However, advecting the interface without diffusing, dispersing, or wrinkling is a big issue. This can either be performed algebraically, in schemes such as CICSAM or geometrically, in schemes such as PLIC. Herein, the viscous fluid governing equations which are Navier-Stokes coupled with VOF equation were presented. Then the most popular VOF schemes such as donor-acceptor, Hirt-Nichols, FCT, Youngs, and PLIC were explained. CICSAM, HiRAC, HRIC, STACS, and some other up-to-date proposed methods were introduced and the accuracy and time calculation of each method were evaluated. Moreover, surface tension modeling and parametric study of interfaces were discussed. The author hopes this brief presentation of the VOF method will be beneficial for scientists and students in their further researches and will help them to massively and continuously expand this very challenging field of fluid mechanics.

Author details

Mohammad Javad Ketabdari

Address all correspondence to: ketabdar@aut.ac.ir

Faculty of Marine Technology, Amirkabir University of Technology, Tehran, Iran

References

[1] Noh WF. CEL: A Time-Dependent, Two Dimensional, Coupled Eulerian-Lagrangian Code. in Alder, B., Fernbach, S. and Rotenberg, M. (Eds.), Methods in Computational Physics, Volume 3, Fundamental Methods in Hydrodynamics, Academic Press, New York, NY 1964; 117–179.

[2] Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. Journal of Computational Physics 1992; 100: 335–354.

[3] Ketabdari MJ, Saghi H. A novel algorithm of advection procedure in volume of fluid method to model free surface flows. International Scholarly Research Network 2012; 10: 1–16.

[4] Ubbink O. Numerical prediction of two fluid systems with sharp interfaces. PhD Thesis, University of London 1997.

[5] Ramshaw JD, Trapp JA. A numerical technique for low-speed homogeneous two-phase flow with sharp interfaces. Journal of Computational Physics 1976; 21: 438–453.

[6] Hirt CW, Nichols, BD. Volume of Fluid (VOF) method for the dynamics of free boundaries. Journal of Computational Physics 1981; 39: 201–225.

[7] Torrey MD, Mjolsness RC, Stein LR. NASA-VOF3D: a three dimensional computer program for incompressible flows with free surfaces. Technical Report LA-11009-MS, Los Alamos National Laboratory 1987.

[8] Lafaurie B, Nardone C, Scardovelli R, Zaleski S. Zanetti G. Modelling merging and fragmentation in multiphase flows with SURFER. Journal of Computational Physics 1994; 113: 134–147.

[9] Heinrich P. Nonlinear numerical model of landslide-generated water waves. International Journal of Engineering Fluid Mechanics 1991; 4(4): 403–416.

[10] Tomiyama A, Zun I, Sou A, Sakaguchi T. Numerical analysis of bubble motion with the VOF method. Nuclear Engineering and Design 1993; 141: 69–82.

[11] Tomiyama A, Sou A, Hisato M, Sakaguchi T. Numerical analysis of a single bubble by VOF method. JSME International Journal Series B Fluids and Thermal Engineering 1993; 36(1): 51–56.

[12] Tomiyama A, Sou A, Sakaguchi T. Numerical simulation of a Taylor bubble in a stagnant liquid inside a vertical pipe. JSME International Journal Series B Fluids and Thermal Engineering 1996; 39(3): 517–524.

[13] Ketabdari MJ, Nobari MRH, Moradi Larmaei M. Simulation of waves group propagation and breaking in coastal zone using a Navier-Stokes solver with an improved VOF free surface treatment. Applied Ocean Research 2008; 30: 130–143.

[14] Nobari MRH, Ketabdari MJ, Moradi Larmaei M. A modified volume of fluid advection method for uniform Cartesian grids. Applied Mathematical Modelling 2009; 33(5): 2298–2310.

[15] Rezaei H, Ketabdari MJ. A modified flux corrected transport method for incompressible free surface flow. Canadian Journal on Computing in Mathematics, Natural Sciences, Engineering and Medicine 2012; 3(6): 213–219.

[16] Boris JP, Book DL. Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. Journal of Computational Physics 1973; 11: 38–69.

[17] Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. Journal of Computational Physics 1979; 31: 335–362.

[18] Youngs DL. Time-Dependent Multi-Material Flow with Large Fluid Distortion. in Morton, K.W. and Baines, M.J. (Eds.), Numerical Methods in Fluid Dynamics 1982; Academic Press, Reading, UK.

[19] Rudman M. Volume-tracking methods for interfacial flow calculations. International Journal for Numerical Methods in Fluids 1997; 24: 671–691.

[20] Rider WJ, Kothe DB. Reconstructing volume tracking. Journal of Computational Physics 1998; 141: 112–152.

[21] Shirani E, Ashgriz N, Mostaghimi J. Interface pressure calculation based on conservation of momentum for front capturing methods. Journal of Computational Physics 2005; 203: 154–175.

[22] Davis SF. Flux difference splittings and limiters for the resolution of contact discontinuities. Applied Mathematics and Computation 1994; 65: 3–18.

[23] Ghobadian A. Development of a method for numerical simulation of flows with moving interfaces. National Power Report 1991; TEC/L/0077/M91.

[24] Van Leer B. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. Journal of Computational Physics 1977; 23: 276–299.

[25] Pericleous KA, Chan KS. The SEA Method for Free-Surface Problems with Heat Transfer and Change of Phase. in Crowe, C.T. (Ed.), Numerical Methods in Multiphase Flows 1994, presented at the ASME fluids engineering division summer meeting 1994; American Society of Mechanical Engineers, 227–236.

[26] Ubbink O, Issa RI. A method for capturing sharp fluid interfaces on arbitrary meshes. Journal of Computational Physics 1999; 153: 26–50.

[27] Hogg PW, Gu XJ, Emerson DR. An implicit algorithm for capturing sharp fluid interfaces in the volume of fluid advection method. European Conference on Computational Fluid Dynamics 2006; ECCOMAS CFD, TU Delft, The Netherlands.

[28] Heyns JA, Malan AG, Harms TM, Oxtoby OF. Development of a compressive surface capturing formulation for modelling free-surface flow by using the volume-of-fluid approach. International Journal for Numerical Methods in Fluids 2013; 71: 788–804.

[29] Muzaferija S, Perić M. Computation of Free Surface Flows Using Interface-Tracking and Interface-Capturing Methods. in Mahrenholtz, O. and Markiewicz, M. (Eds.), Nonlinear Water Wave Interaction, Computational Mechanics Publications, Southampton 1998; p. Chap. 3.

[30] Tsui YY, Lin SW, Cheng TT, Wu TC. Flux-blending schemes for interface capture in two-fluid flows. International Journal of Heat and Mass Transfer 2009; 52: 5547–5556.

[31] Darwish M., Moukalled F. Convective schemes for capturing interfaces of free-surface flows on unstructured grids. Numerical Heat Transfer, Part B: Fundamentals 2006; 49: 19–42

[32] Jasak H, Weller HG. Interface Tracking Capabilities of the Inter-Gamma Differencing Scheme. Imperial College, University of London, 1995; p. 9.

[33] Peng D, Merriman B, Osher S, Zhao H, Kang M. A PDE-based fast local level set method. Journal of Computational Physics 1999; 155: 410–438.

[34] Sussman M, Puckett EG. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. Journal of Computational Physics 2000; 162: 301–337.

[35] Renardy Y, Renardy M. PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method. Journal of Computational Physics 2002; 183: 400–421.

[36] Jamet D, Torres D, Brackbill JU. On the theory and computation of surface tension: the elimination of parasitic currents through energy conservation in the second-gradient method. Journal of Computational Physics 2002; 182: 262–276.

[37] Popinet S, Zaleski S. A front-tracking algorithm for accurate representation of surface tension. International Journal for Numerical Methods in Fluids 1999; 30: 775–793.

[38] Bohluly A, Borghei SM, Saidi, MH. A new method in two phase flow modeling of a non-uniform grid. Scientia Iranica, Transaction B: Mechanical Engineering 2009; 16(5): 425–439.

[39] Chorin AJ. Curvature and solidification. Journal of Computational Physics 1985; 58:472–490.

[40] Ashgriz N, Poo JY. A computational method for determining curvatures. Journal of Computational Physics 1989; 84: 483–491.

[41] Kothe DB, Rider WJ, Mosso SJ, Brock JS, Hochstein JI. Volume Tracking of Interfaces Having Surface Tension in Two and Three Dimensions. AIAA 34th Aerospace Sciences Meeting and Exhibit 1996; AIAA 96-0859, Reno, NV, USA.

[42] Saghi H, Ketabdari MJ, Zamirian M. A novel algorithm based on parameterization method for calculation of curvature of the free surface flows. Applied Mathematical Modelling 2013; 37: 570–585.