



Interface reconstruction and advection schemes for volume of fluid method in axisymmetric coordinates



Ananthan Mohan, Gaurav Tomar*

Department of Mechanical Engineering, Indian Institute of Science, Bangalore, India

ARTICLE INFO

Article history:

Available online 31 August 2021

Keywords:

Volume of fluid
Axisymmetric coordinates
Multiphase simulation

ABSTRACT

Volume of fluid (VOF) method is a sharp interface method employed for simulations of two phase flows. Interface in VOF is usually represented using piecewise linear line segments in each computational grid based on the volume fraction field. While VOF for cartesian coordinates conserve mass exactly, existing algorithms do not show machine-precision mass conservation for axisymmetric coordinate systems. In this work, we propose analytic formulae for interface reconstruction in axisymmetric coordinates, similar to those proposed by Scardovelli and Zaleski (2000) [12] for cartesian coordinates. We also propose modifications to the existing advection schemes in VOF for axisymmetric coordinates to obtain higher accuracy in mass conservation.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Multiphase flows are ubiquitous in several industrial applications. In the last three decades, there has been a surge in the numerical methods and algorithms for simulations of complex multiphase flows. There have been several different types of interface capturing strategies that have been proposed for two-phase flows. The most popular of these are the Level set method, Volume of Fluid (VOF) method, and Front tracking scheme [1,2]. VOF methods with geometric advection strictly conserve the volume of the two phases.

Several improvements have been made since the inception of the method (see Hirt and Nichols [3–8]). The simplest and earliest representation for interface reconstruction is simple line interface calculation (SLIC) in which the interface is approximated by horizontal or vertical lines. Subsequently, piecewise line interface construction (PLIC) was introduced where the interface is approximated as a linear line at an angle in the cell [9]. Higher order interface construction has been proposed (such as Parabolic reconstruction by [4]), but considering the associated computational cost and complexity for geometric advection, PLIC is usually preferred.

In PLIC based method the interface is approximated as a line segment given by $\mathbf{m} \cdot \mathbf{x} = a$, where $\mathbf{m} = (m_x, m_y)$ is the unit normal to the interface and a is the line constant. The reconstruction step in VOF methods consists of two steps, the computation of \mathbf{m} and then the computation of line constant a such that the volume cut by the interface matches the volume fraction of the cell. The interface normal is computed using the volume fraction gradient by using an accurate gradient computing algorithm like Youngs method [9], a least-squares method (e.g.: (E)LVIRA [7]) or a height function method [10]. Another popular method for interface reconstruction is the Moment-of-Fluid (MoF) method [11], which uses the volume fraction value of the cell along with the geometric centroids of the individual fluid volumes within the cell

* Corresponding author.

E-mail address: gtom@iisc.ac.in (G. Tomar).

to reconstruct the interface. The MoF methods require one to track the advection of the geometric centroids of the fluid volumes along with the volume fraction. In the present, work we are primarily interested in the computation of the line constant given an interface normal in the frame work of VOF methods. Nevertheless the reconstruction algorithm presented in the study is quite generic for structured grids.

Scardovilli and Zaleski [12] proposed analytical formulae for the piecewise linear reconstruction of the interface in cartesian coordinates that led to a significant speedup over the earlier iterative schemes. However, for curvilinear coordinates (such as axisymmetric coordinate system), the proposed analytical formulae cannot be employed directly. Diot et al. [13] proposed a semi-analytic formulae for arbitrary convex cells which required the use of Newtons iteration for axisymmetric coordinate system. In the present work, we derive similar analytic formulae for axisymmetric coordinates, which result in a speedup of ~ 28 over the iterative counterparts (Brent's root finding method). Further, we demonstrate that the existing interface advection schemes in VOF for axisymmetric coordinates are not strictly mass conserving. In this study, we propose modifications to the current operator split algorithms that result in machine-precision mass conservation in axisymmetric coordinates. We show the efficacy of the proposed algorithms using several test cases.

The paper is organized as follows. We first present analytical formulae for the interface reconstruction schemes in axisymmetric coordinates in section 2. In section 3, we propose modifications in the existing interface advection algorithm for axisymmetric VOF and present test cases to show the efficacy of the scheme. Finally, in section 4, we discuss the important conclusions.

2. Interface reconstruction scheme

Interface reconstruction in the volume of fluid (VOF) method requires the volume fraction field. Using the volume fraction field, a piece-wise linear or a higher order interface is constructed in a given grid-cell. Interface reconstruction is an integral part of the geometric advection schemes to ensure mass conservation property of the VOF method [2]. Initial condition for a multiphase flow simulation requires the initial distribution of the volume fraction field, usually provided as an implicit function of the spatial coordinates. VOFI library [14] is an open source library to initialize the liquid volume fraction field in cartesian coordinate systems accurately. In VOFI, for cells cut by the interface (see Fig. 1), PLIC reconstruction method [9,6] is employed to approximate the interface as a linear line segment,

$$\mathbf{m} \cdot \mathbf{x} = a, \quad (1)$$

where \mathbf{m} is the local normal at the interface, \mathbf{x} is a point on the plane, and a is the normal distance of the origin from the plane. Analytical relation, given by Scardovilli and Zaleski [12], between the volume fraction and the line constant is employed to determine the line constant a . Thus, for two-dimensional and three-dimensional Cartesian coordinate systems, VOFI library can be directly employed for accurate assignment of the initial volume fraction field on a given discretized domain using an implicit equation of the interface. However, in curvilinear coordinate systems, for a given implicit function, the piece-wise linear interface constructed from VOFI would require computation of the volume fraction field using a formula specific to the curvilinear coordinates. For instance, for axisymmetric coordinate system, the modified Gauss area (shoelace) formula for computation of the area of a convex polygon is given by,

$$V = \frac{\pi}{3} \left| \sum_{i=1}^n (x_i + x_{i+1}) (x_i y_{i+1} - x_{i+1} y_i) \right| \quad (2)$$

where (x_i, y_i) for $i = 1, \dots, n$ (with $x_{n+1} = x_1$ and $y_{n+1} = y_1$) are the coordinates of the vertices of a convex polygon ordered counter clockwise as shown in the Fig. 1.

Thus, for initialization of the volume fraction field, C , once we obtain the linear interface in each grid cell using the VOFI library, we use the above formula to compute the volume, V , and assign volume fraction in each grid cell as,

$$C = \frac{V}{2\pi r_c \Delta x \Delta y}. \quad (3)$$

Here r_c is the distance of the center of the cell from the axis of symmetry, and Δx and Δy are the grid-cell sizes in the radial (r) and axial (y) directions, respectively. We note that the above procedure is followed essentially to minimize the error in the volume-fraction during initialization. To illustrate this, we initialize a torus of minor radius $r_t = 0.25$ and major radius $r = 0.50$ in the center of a computational domain of size 1×1 as shown in the Fig. 2. The volume of the torus can be analytically computed as $V_t = 2\pi^2 r r_t^2$, where the major radius, r , is the distance to the center of the torus from the axis of symmetry. We compare the results for initialized volume for various grid sizes by computing the relative error with respect to volume obtained using analytical formula, given in Table 1.

Thus, we have shown that for curvilinear coordinates volume fraction field can be initialized up to machine accuracy. Now, we derive an analytic relation for PLIC reconstruction for axisymmetric coordinates on the lines of Scardovilli and Zaleski [12]. We use Youngs method [9] to get the interface normal (\mathbf{m} in equation (1)) from fluid-1 ($C = 1$) to fluid-2 $C = 0$ and is given by $\mathbf{m} = -\nabla C / |\nabla C|$. To complete the PLIC interface reconstruction for a given C , in addition to the normal \mathbf{m} ,

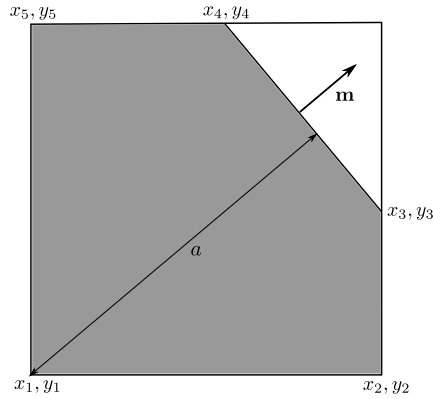


Fig. 1. Coordinates of the vertices of a simple polygon cut by the interface ordered counter clockwise. Here x_3, y_3 and x_4, y_4 represents the end points of the PLIC line segment with interface normal \mathbf{m} and line constant a .

Table 1

Results for relative error in volume during initialization of a torus of radius, $r_t = 0.25$, for different grid sizes.

Relative error in volume: $E = \frac{ V - V_t }{V_t}$	
Grid	Current Solver
16×16	5.4×10^{-16}
32×32	7.1×10^{-16}
64×64	1.8×10^{-16}
128×128	5.4×10^{-16}

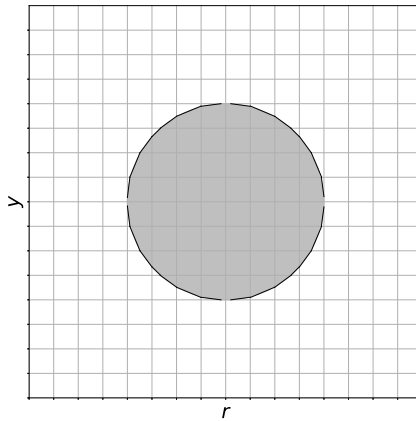


Fig. 2. Reconstructed interface for 16 mesh points of a torus of radius $r_t = 0.25$ and major axis $r = 0.50$ initialized at the center of 1×1 domain where reconstructed PLIC line segments for each mixed cell is depicted.

we also need to obtain the line constant, a , which is the normal distance of the interface from one of the vertices of the computational cell. In what follows, we present a methodology to get the line constant (a) analytically for a given interface normal vector and the volume fraction of a mixed cell.

As discussed in [12], using an analytical relation between the volume fraction (C), interface normal (\mathbf{m}) and the line constant (a), we can implement an *if - else - if - end - if* construct to determine the line constant, a . This approach is computationally much more efficient compared to the alternative iterative approach to get the line constant. Given the equation of the interface, $m_1x + m_2y = a$, all combinations of m_1, m_2 (such that $m_1^2 + m_2^2 = 1$) can be reduced to one of the cases shown in Fig. 3, either by changing the origin or by changing the reference fluid from fluid-1 to fluid-2, such that both m_1 and m_2 are positive and the left bottom corner of the mixed cell under consideration is contained in fluid-1. Fig. 3 shows all the possible configurations for interface arrangement with $m_1 \geq 0$ and $m_2 \geq 0$.

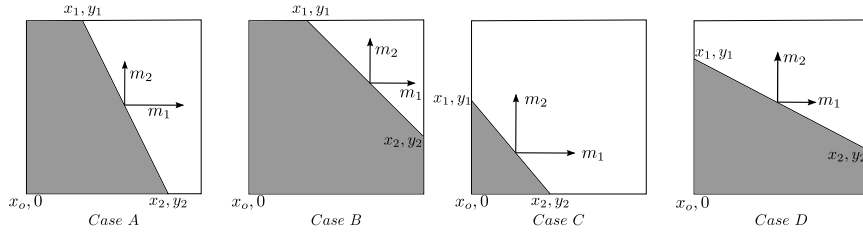


Fig. 3. Various cases which can arise in the standard configuration of the interface with $m_1, m_2 \geq 0$ and fluid 1 is occupying the bottom left corner of the cell.

We first discuss Case A shown in the Fig. 3 and similar procedure can be followed to obtain relations for other cases. For the axisymmetric coordinate system, the shaded area shown in the Fig. 3 for Case A is given by,

$$V = \frac{\pi}{3} (x_0 + x_2)^3 \frac{m_1}{m_2} - \frac{\pi}{3} (x_0 + x_1)^3 \frac{m_1}{m_2} - \pi x_0^2 y_1. \tag{4}$$

Using the equation of line $m_1x + m_2y = a$ we have $x_1 = a/m_1 - (m_2\Delta y)/m_1$ and $x_2 = a/m_1$. In the present study, we assume $\Delta x = \Delta y$, but the same analysis can be easily extended for $\Delta x \neq \Delta y$. Substituting x_1 and x_2 in the equation (4) and collecting the terms in powers of a , we obtain,

$$\left(\frac{\pi \Delta y}{m_1^2}\right) a^2 + \left(-\frac{\pi m_2 \Delta y^2}{m_1^2} + \frac{2\pi \Delta y x_0}{m_1}\right) a + \left(\frac{\pi m_2^2 \Delta y^3}{3m_1^2} - \frac{\pi \Delta y^2 m_2 x_0}{m_1}\right) = V. \tag{5}$$

Thus, we have an analytical relation between the volume and the line constant a . We note that the above relation holds true only when the interface cuts through the top and the bottom edges of the cell shown for Case A in Fig. 3: $(x_2 - x_0) \leq \Delta x$, $y_1 = \Delta y$ and $y_2 = 0$. These conditions yield the bounds on the values of a : $m_2\Delta y \leq a \leq m_1\Delta x$. Substituting the above bounds for a in the equation (5), yield the bounds on the limiting volumes:

$$V_1 = \frac{\pi \Delta y (3\Delta x^2 m_1^2 + 3\Delta x m_1 (2m_1 x_0 - \Delta y m_2) + \Delta y m_2 (\Delta y m_2 - 3m_1 x_0))}{3m_1^2}, \tag{6}$$

and

$$V_2 = \frac{\pi \Delta y^2 m_2 (\Delta y m_2 + 3m_1 x_0)}{3m_1^2} \tag{7}$$

For a given volume fraction C and interface normal (m_1, m_2) , the volume occupied by fluid-1 in the configurations shown in the Fig. 3 is given by: $V = 2\pi r \Delta x \Delta y C$ where $r = x_0 + \Delta x/2$ is the distance from the axis of symmetry to the cell center. If $V_1 \leq V \leq V_2$ then the analytical relation given by equation (5) can be used to determine the line constant a . For the quadratic equation in a given by equation (5), we note that only one of the roots will satisfy the required bounds on a for case A. For cases B and C, we obtain cubic equations that can be solved for a using Cardano's formula or using Brent's method to find the appropriate root with necessary bounds for the line constant. Following the same approach we can get bounds on volume for case D as:

$$V_3 = \frac{\pi \Delta y^2 (-2\Delta y m_1 + 3\Delta y m_2 - 3m_1 x_0 + 6m_2 x_0)}{3m_2} \tag{8}$$

and

$$V_4 = \frac{\pi \Delta y^2 m_1 (\Delta y + 3x_0)}{3m_2}. \tag{9}$$

Fig. 4 shows all the possible configurations of the interface and the volume bounds which separate each case as the interface normal in radial direction varies from minimum to a maximum. Fig. 4 clearly shows that the various bounds for the cases shown in Fig. 3 do not overlap and provide a unique criterion for computing the line constant a . Thus, we can use the following algorithm to classify each case.

We list below the analytical relation between the line constant (a) and the volume (V) for each case:

Case A

$$\left(\frac{\pi \Delta y}{m_1^2}\right) a^2 + \left(-\frac{\pi m_2 \Delta y^2}{m_1^2} + \frac{2\pi \Delta y x_0}{m_1}\right) a + \left(\frac{\pi m_2^2 \Delta y^3}{3m_1^2} - \frac{\pi \Delta y^2 m_2 x_0}{m_1}\right) = V. \tag{10}$$

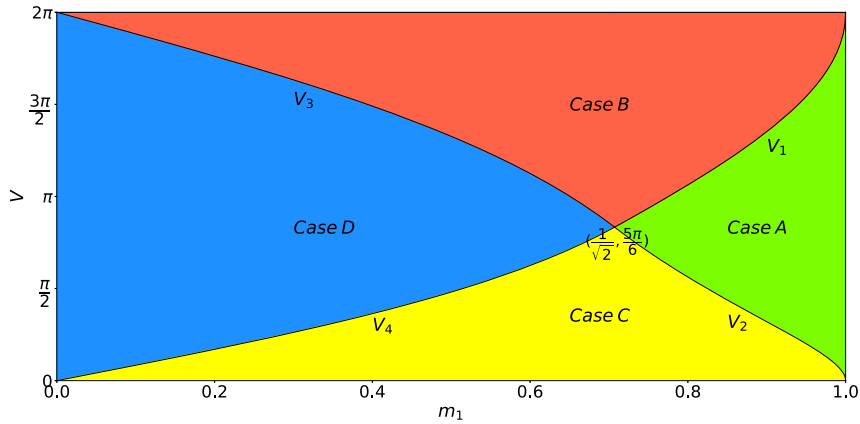


Fig. 4. Bounds for different cases shown in Fig. 3 as a function of the x-component of the interface normal, m_1 . The interfacial cell is placed at a distance of 1 from the axis of symmetry with $\Delta x = \Delta y = 1$. The total volume of the cell is $V_{cell} = 2\pi$ given by the top boundary in the plot.

```

Data: V, m1
Result: Identification of the case in which the standard interface belongs to.
if m1 ≥ 1/√2 then
    if V ≥ V1 then
        | Case B
    else if V ≤ V2 then
        | Case C
    else
        | Case A
    end
else
    if V ≥ V3 then
        | Case B
    else if V ≤ V4 then
        | Case C
    else
        | Case D
    end
end
// V1 > V > V2
// V3 > V > V4
    
```

Algorithm 1: Classification of the standard case of the reconstructed interface where $m_1, m_2 \geq 0$.

Case B

$$\begin{aligned}
 & -\left(\frac{\pi}{3m_1^2m_2}\right)a^3 + \left(\frac{\pi\Delta y}{m_1^2} - \frac{\pi x_0}{m_1m_2}\right)a^2 + \left(\frac{\pi(\Delta y + x_0)^2}{m_2} - \frac{\pi m_2\Delta y^2}{m_1^2} + \frac{2\pi\Delta yx_0}{m_1} - \frac{\pi x_0^2}{m_2}\right)a \\
 & + \left(-\frac{\pi\Delta y(\Delta y + x_0)^2m_1}{m_2} + \frac{\pi m_2^2\Delta y^3}{3m_1^2} - \frac{\pi\Delta y^2m_2x_0}{m_1} - \frac{\pi m_1x_0^3}{3m_2} + \frac{\pi m_1(\Delta y + x_0)^3}{3m_2}\right) = V
 \end{aligned} \tag{11}$$

Case C

$$\left(\frac{\pi}{3m_1^2m_2}\right)a^3 + \left(\frac{\pi x_0}{m_1m_2}\right)a^2 = V \tag{12}$$

Case D

$$a = \frac{2\pi\Delta y^3m_1 + 3\pi\Delta y^2m_1x_0 + 3m_2V}{3\pi\Delta y(\Delta y + 2x_0)} \tag{13}$$

We note here that the other cases can be readily transformed into one of the cases listed in the Fig. 3 by either changing the fluid (by using $(1 - C)$ instead of C to compute the volume and inverting the interface normal \mathbf{m}) or by changing the origin (keeping the location of the axis-of-symmetry same but inverting its direction).

We now compare the above described analytical method with the iterative method for finding the line constant for the case A given in Fig. 3. The relative error in the line constant is given in Table 2 for the analytical and iterative methods with different tolerances. We note that the iterative method to reach an error with a tolerance of 10^{-8} is about 28 slower compared to the analytical method.

Table 2
Relative error in line constant a and the ratio of CPU time required by iterative method (Brent's algorithm) to that required by analytical method for 10000 repetitions for case A shown in Fig. 3 for different tolerances used in the iterative method.

Comparison between analytical and iterative reconstruction methods			
Method	Tolerance	Relative Error	$t_{iterative}/t_{analytical}$
Analytical	–	0	1
Iterative	1.0×10^{-4}	1.8×10^{-4}	14.2
Iterative	1.0×10^{-6}	1.1×10^{-6}	21.2
Iterative	1.0×10^{-8}	6.7×10^{-11}	28.3

Once the line constant, a , is obtained, the position of the endpoints of the linear approximation of the interface can be computed thus completing the construction of a planar interface in a given computational cell. As discussed earlier, this more precise description of the interface within the grid cell allows geometric advection which gives the VOF method its strict mass conservation property while maintaining a sharp interface.

In what follows, we discuss an operator split algorithm for the geometric advection of the interface in axisymmetric coordinates. We note that the straightforward extension of the 2D cartesian algorithm does not yield accurate results, as also indicated by the results obtained from the existing open source codes.

3. Advection of the interface

We present here a scheme for accurate geometric advection of the volume fraction in the axisymmetric coordinates. We have used a uniform grid to describe the variables with volume fraction being stored at the cell centers ($C_{i,j}$). The incompressible fluid flow is determined by the velocity field which is defined at the cell faces ($u_{i+1/2,j}, v_{i,j+1/2}$). Here, u denotes the radial direction velocity and v is the axial velocity. The velocity field satisfies the discrete divergence free condition given by:

$$\frac{(ru)_{i+\frac{1}{2},j} - (ru)_{i-\frac{1}{2},j}}{r_i \Delta x} + \frac{v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}}{\Delta y} = 0. \tag{14}$$

Motion of the interface is governed by the advection equation for the volume fraction field,

$$\frac{\partial C}{\partial t} + \mathbf{u} \cdot \nabla C = 0 \tag{15}$$

For incompressible fluids, conservation of the individual volumes of the two fluids results in the conservation of mass. In the volume of fluid method, geometric advection of the volume fraction field is employed instead of using a higher order scalar advection scheme to solve equation (15) since it limits the diffusion of volume fraction field thus maintaining a sharp interface and at the same time ensures the conservation of mass to high degree of accuracy [2]. Given a volume fraction field, reconstructed interface and solenoidal velocity field, we can solve the equation (15) using an operator splitting algorithm consisting of an r -sweep and a y -sweep following [15]. In order to employ an operator splitting algorithm, the advection of the interface (equation (15)), using $\nabla \cdot \mathbf{u} = 0$, can be written as:

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u}C) = C(\nabla \cdot \mathbf{u}). \tag{16}$$

The RHS in equation (16) is the dilatation term or the divergence correction term and is needed in an operator split approach since in each step of an operator split algorithm the volume fraction C is advected first in one direction and then in the other direction(s). The discrete divergence term or the dilatation term is included in the formulation since in each step of an operator split algorithm the volume fraction C is advected in a one dimensional flow which is not divergence free. Given a volume fraction ($C_{i,j}^n$) and velocity field ($u_{i+1/2,j}^n, v_{i,j+1/2}^n$) at the n th time step, the discretized equation (16) is given by,

$$C_{i,j}^{n+1} = C_{i,j}^n + \frac{\Delta t}{r_{i,j} \Delta x} (\delta V_{i-1/2,j} - \delta V_{i+1/2,j}) + \frac{\Delta t}{\Delta y} (\delta V_{i,j-1/2} - \delta V_{i,j+1/2}) + C_{i,j}^n \left(\frac{\Delta t}{r_{i,j} \Delta x} (r_{i+1/2,j} u_{i+1/2,j}^n - r_{i-1/2,j} u_{i-1/2,j}^n) + \frac{\Delta t}{\Delta y} (v_{i,j+1/2}^n - v_{i,j-1/2}^n) \right) \tag{17}$$

where $\delta V_{i+1/2,j} = (ruC)_{i+1/2,j}^n$ is the amount of volume fraction fluxed through the right cell face. Similarly, fluxes $\delta V_{i-1/2,j}, \delta V_{i,j+1/2}$ and $\delta V_{i,j-1/2}$ can be computed for other cell faces.

Using operator splitting, we can split the above equation as follows:

$$C_{i,j}^* = C_{i,j}^n + \frac{\Delta t}{r_{i,j} \Delta x} (\delta V_{i-1/2,j} - \delta V_{i+1/2,j}) + C_{i,j}^c \left(\frac{\Delta t}{r_{i,j} \Delta x} (r_{i+1/2,j} u_{i+1/2,j}^n - r_{i-1/2,j} u_{i-1/2,j}^n) \right) \tag{18}$$

$$C_{i,j}^{n+1} = C_{i,j}^* + \frac{\Delta t}{\Delta y} (\delta V_{i,j-1/2} - \delta V_{i,j+1/2}) + C_{i,j}^c \left(\frac{\Delta t}{\Delta y} (v_{i,j+1/2}^n - v_{i,j-1/2}^n) \right) \tag{19}$$

where $C_{i,j}^*$ is the intermediate value of the volume fraction. Here, $C_{i,j}^c$ is a function based on which we can get different operator splitting approaches. If $C_{i,j}^c = C_{i,j}^*$, then we get Sussman and Puckett’s operator split approach [15]. An implicit scheme is used in the first direction and an explicit scheme in the second direction to maintain the conservation of volume fraction [16]. The order of sweep of direction is alternated every timestep [17] (“Strang splitting”) to achieve second order accuracy in time.

If we use,

$$C_{i,j}^c = \begin{cases} 1 & C_{i,j}^n \geq 0.5 \\ 0 & C_{i,j}^n < 0.5 \end{cases} \tag{20}$$

where $C_{i,j}^c$ depends only on the previous time step value of volume fraction we get the axisymmetric form of Weymouth and Yue’s operator split approach [18]. This method is explicit in both the sweep directions. For an operator split method to be strictly conservative it has to fulfill, following Weymouth and Yue [18], a set of concurrent requirements,

1. The volume flux terms are conservative, and
2. the dilatation term sums to zero, and
3. no clipping or filling of a cell is needed to impose $0 \leq C \leq 1$.

Using geometric advection (described below) for volume fluxing and a divergence free velocity field, both Sussman and Puckett’s and Weymouth and Yue’s methods satisfy the first two requirements. Sussman and Puckett’s method does require clipping or filling of a cell to impose $0 \leq C \leq 1$ and therefore is not strictly conservative. However the mass conservation errors introduced in this method are small and thus it has been widely adopted previously. Weymouth and Yue [18] proved that, by imposing a grid Courant number restriction of,

$$\Delta t \sum_{d=1}^N \left| \frac{u_d}{\Delta x_d} \right| < \frac{1}{2} \tag{21}$$

in their method one can satisfy the third requirement for the operator split approach to be conservative. Here N is the number of sweep directions and u_d and Δx_d are the maximum velocity and grid size in each of those directions. Thus by using Weymouth and Yue’s operator splitting approach we can preserve the mass exactly when advecting the volume fraction field.

The volume flux through cell faces, $\delta V_{cell-face}$, is computed geometrically. Consider the schematic in Fig. 5, where the shaded region shows the volume of fluid-1 in the cell to be fluxed through the right face ($\delta V_{i+1/2,j}$). Considering the face velocity ($u_{i+1/2,j}$) to be positive, the flux can be computed as,

$$\delta V_{i+\frac{1}{2},j} = \frac{(ru)_{i+1/2,j} V}{2\pi r \Delta r \Delta y} \tag{22}$$

where V is the volume of fluid 1 fluxed through the right face (shown as the shaded region in Fig. 5), Δr is the distance in the radial direction which contains the volume advected in this timestep and r is the distance to the center of this volume from the axis of symmetry. Here the volume V depends on the orientation of the interface. We can calculate Δr by considering the conservation of volume fluxed through the right face and solving the resulting quadratic equation, which yields, $\Delta r = r_{i+\frac{1}{2},j} - \sqrt{r_{i+\frac{1}{2},j}^2 - 2r_{i+\frac{1}{2},j} u_e \Delta t}$. Using the section of the piece-wise reconstructed interface lying in the volume to be fluxed through the cell-face over Δt time-step and employing the Gauss area formula, given by equation (2), we can calculate the volume cut by this region.

The above small correction in computing Δr along with the accurate Gauss formula for axisymmetric simulations allows us to improve upon the existing volume fraction advection schemes. Existing schemes [15] modify the velocity in 2D algorithms by using ru for velocity field and use 2D geometric advection scheme which results in a third order error ($O(\Delta t^2 h)$), where h is the grid size and Δt is the timestep) in mass conservation. To illustrate this geometric fluxing error, consider the radial advection of the volume fraction field for a full cell ($C = 1$) as shown in Fig. 6. The shaded region represents the volume advected through the right face of the cell with a radial cell face velocity u_e .

The existing schemes compute the volume as $V_c = 2\pi(r_e - \frac{u_e \Delta t}{2})(u_e \Delta t) \Delta y$, where r_e is the distance of the east cell face from the axis of symmetry, Δt is the timestep, and Δy is the height of the cell. Whereas, the proposed scheme yields

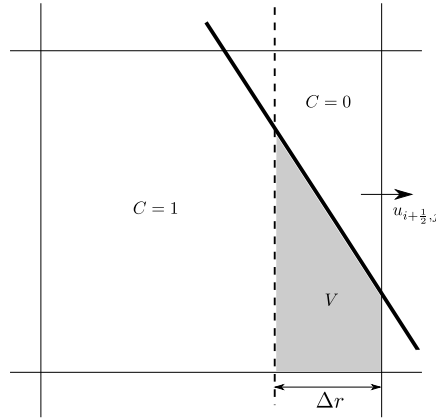


Fig. 5. The fluxed volume through the right face of the cell when $u_{i+1/2,j}$ is positive.

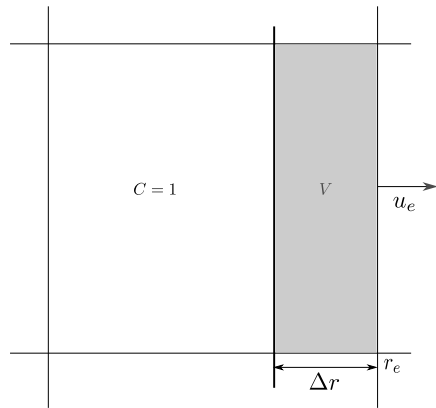


Fig. 6. The shaded region is the volume advected through the east cell face in a single timestep.

the exact volume, $V = 2\pi(r_e - \frac{\Delta r}{2})\Delta r\Delta y$ with $\Delta r = r_e - \sqrt{r_e^2 - 2r_e u_e \Delta t}$. Thus, the error in volume calculation is given by, $E = \pi(u_e \Delta t)^2 \Delta y$.

We validate the proposed modifications with the following test cases and compare with the results obtained using the open source multiphase flow solver Gerris [19] which uses a simpler extension of the cartesian operator split approach for geometric fluxing.

3.1. Advection of a torus

In this test case, a torus of radius 0.25 is initialized at (0.35, 0.5) in a computational domain of size (2.0, 1.0). The torus is advected under the steady state velocity of $u = 0.1/r$ for $r > 0.05$ and $v = 0$, where r is the distance from the axis of symmetry. The fluid is advected forward in time $t = 7.8125$ and then the velocity is reversed so as to advect the fluid backwards in time till time $t = 15.625$. In order to illustrate the effect of accurate geometric fluxing, we advect the torus using Sussman and Puckett’s method for a grid Courant number of 1 and compare the results with that obtained using Gerris flow solver. For a grid size of $\Delta x = \Delta y = 1/128$, this results in a time step of $\Delta t = 0.0078125$. Thus the fluid is advected a 1000 timesteps forward in time and then backwards. As seen from the Fig. 7, after 1000 timesteps the torus is highly compressed during the advection as less area (due to axisymmetry) occupies the same volume as we move away from the axis of symmetry. We note that the final interface shape matches very well with the initial position of the torus, thus validating our algorithm.

The relative error in the volume between the initial and final distribution of fluid-1 for various number of forward and backward advection time steps is given in Table 3. The corresponding relative change in the volume obtained for the same test case simulated using Gerris flow solver is also given for comparison. We note that the error obtained from the present schemes is highly accurate in comparison to those obtained from Gerris (Table 3).

We repeat this test for different Courant numbers using both Sussman and Puckett’s and Weymouth and Yue’s advection schemes and compare the results with those obtained using Gerris flow solver. Table 4 gives the relative error in volume

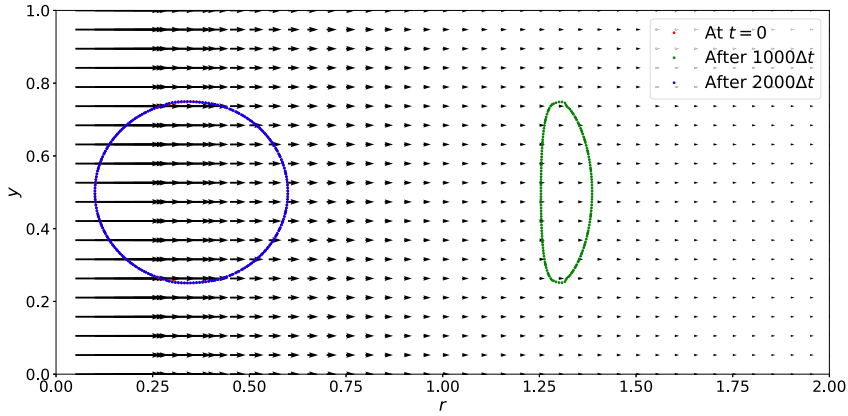


Fig. 7. Interface shape after 1000 time steps forward and backward after advecting the torus of radius 0.25 with a grid Courant (CFL) number of 1 advected using Sussman and Puckett's method for grid Courant number of 1.

Table 3

Results for relative error in the volume for a torus of radius 0.25 advected in radial direction forward and backward in time for different number of timesteps.

Relative error in volume		
Number of timesteps	Current Solver (Sussman and Puckett)	Gerris Solver
1	5.2×10^{-15}	1.1×10^{-5}
10	2.9×10^{-14}	7.2×10^{-5}
100	4.8×10^{-13}	2.1×10^{-4}
1000	3.4×10^{-10}	3.8×10^{-4}

Table 4

Results for relative error in the volume for a torus of radius 0.25 advected in radial direction forward and backward in time for different grid Courant numbers.

Effect of Courant number on relative error in volume			
Grid Courant number	Gerris Solver	Current Solver (Sussman and Puckett)	Current Solver (Weymouth and Yue)
0.1	9.5×10^{-5}	1.4×10^{-9}	1.9×10^{-12}
0.2	3.9×10^{-6}	1.9×10^{-10}	2.4×10^{-12}
0.3	1.3×10^{-4}	7.1×10^{-11}	2.6×10^{-13}
0.4	2.4×10^{-4}	1.0×10^{-10}	2.5×10^{-15}

for various grid Courant numbers. We observe that using Weymouth and Yue's operator splitting approach along with the interface reconstruction algorithm presented we can obtain machine accurate volume conservation. We observe a small increase in error as Courant number decreases which can be attributed to the large number of time steps which result in a larger number of interface reconstruction events thus leading to larger error accumulation.

As suggested by Kothe et al. [20], simple linear advection test cases do not reveal the efficacy of advection algorithms appropriately. Thus, we further test the efficacy of the algorithm by subjecting it to a more severe test case of advection of a torus in a Hill's vortex [21] with a superimposed radial velocity field which results in a divergence free velocity field upto machine-precision. This is axisymmetric equivalent of the circle in a vortex test case for 2D cases [20]. For this velocity field, the interface undergoes strong topological changes including fragmentation and merging due to strong shear effects. A torus of radius 0.1 is initialized at (0.2, 0.8) in a computational domain of size (1.0, 1.0) with $L = 0.5$. The fluid is advected under highly strained steady state velocity field given by (for $r \geq 0.05$)

$$u = 0.1 \left(\frac{r}{L} \frac{(y-L)}{L} \right) + \frac{0.05}{r} \tag{23}$$

$$v = 0.1 \left[1 - \left(\frac{y-L}{L} \right)^2 - 2 \left(\frac{r}{L} \right)^2 \right]. \tag{24}$$

The fluid is advected forward in time $t = 4.0$ and then the velocity is reversed so as to advect the fluid backwards in time till time $t = 8.0$. For a grid size of $\Delta x = \Delta y = 1/128$, and grid Courant number of 0.1 this results in a time step of $\Delta t = 1.0 \times 10^{-3}$. The fluid is advected 4000 timesteps forward in time and then the velocity is reversed to advect 4000 timesteps backwards in time.

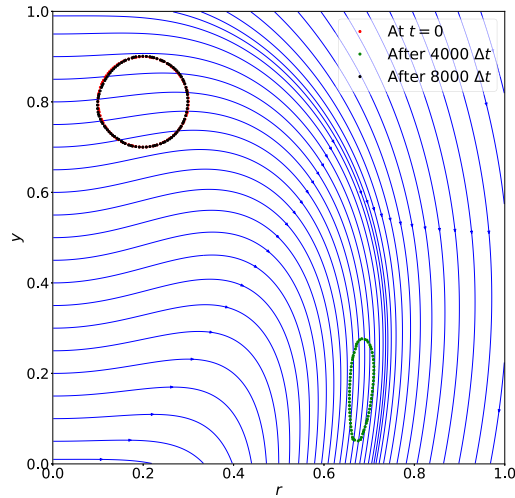


Fig. 8. Interface shape after 4000 time steps forward and backward after advecting the torus placed in a vortex using Weymouth and Yue's operator split approach for a grid Courant number of 0.1.

Table 5

Results for relative error in volume for a torus of radius 0.1 placed in a complex velocity field advected forward and backward in time for different grid Courant numbers.

Effect of Courant number on relative error in volume			
Grid Courant number	Gerris Solver	Current Solver (Sussman and Puckett)	Current Solver (Weymouth and Yue)
0.1	2.7×10^{-4}	3.5×10^{-8}	2.7×10^{-12}
0.2	1.6×10^{-4}	3.4×10^{-7}	2.2×10^{-15}
0.3	8.1×10^{-5}	7.6×10^{-7}	4.1×10^{-15}
0.4	5.4×10^{-4}	4.2×10^{-6}	1.9×10^{-15}

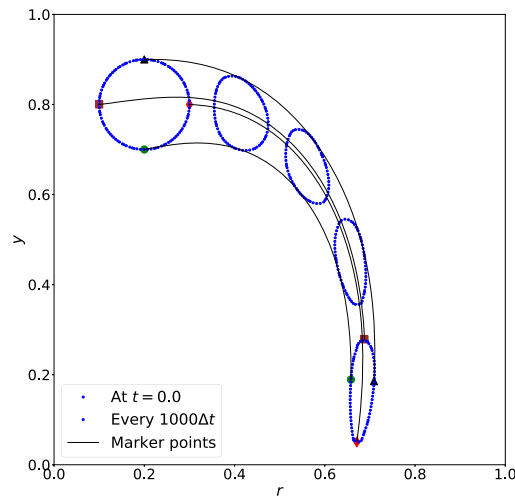


Fig. 9. Interface shape after every 1000 time steps compared against the trajectories of Lagrangian marker particles for simulation using a grid Courant number of 0.1 using Weymouth and Yue's operator splitting approach.

As seen from Fig. 8 after 4000 timesteps the shape of the interface is highly distorted due to highly strained velocity field. The final interface shape matches very well with the initial position of the toroid thus validating our algorithm.

We repeat this test for different Courant numbers using both Sussman and Puckett's and Weymouth and Yue's advection schemes and compare the results with those obtained using Gerris flow solver. Table 5 gives the relative error in volume for various grid Courant numbers. We observe that relative error using Sussman and Puckett's operator splitting approach is an order smaller compared to the results from Gerris flow solver even for larger number of time steps. Using Weymouth and Yue's scheme we observe machine accurate volume conservation.

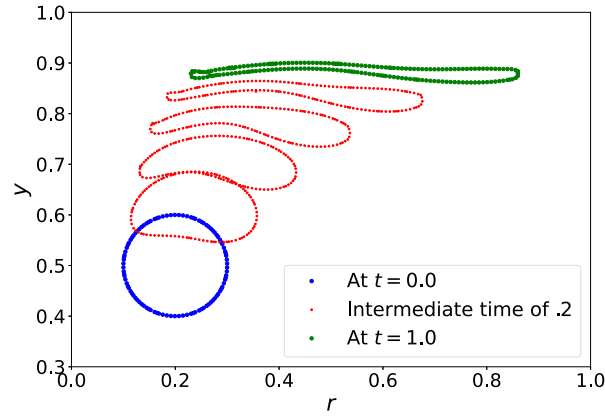


Fig. 10. Interface shape after time $t = 1$. The interface is flattened against the top wall and stretched due to the underlying velocity. Even when the cross-sectional area changes the volume of the toroidal bubble is maintained very accurately.

Table 6

Results for relative error in the volume for a bubble in stagnation point flow for $t_{final} = 1.0$ for different tolerance of Pressure-Poisson solver for both Sussman and Puckett's and Weymouth and Yue's advection methods.

Tolerance of Poisson Solver	Divergence of velocity field	Relative error (Sussman and Puckett)	Relative error (Weymouth and Yue)
1.0×10^{-4}	1.1×10^{-4}	1.1×10^{-4}	2.3×10^{-5}
1.0×10^{-6}	5.1×10^{-5}	1.1×10^{-6}	1.5×10^{-6}
1.0×10^{-8}	2.6×10^{-7}	2.5×10^{-6}	2.3×10^{-9}
1.0×10^{-10}	2.67×10^{-9}	2.6×10^{-6}	4.8×10^{-10}
1.0×10^{-12}	1.5×10^{-11}	2.6×10^{-6}	1.7×10^{-11}
1.0×10^{-14}	7.5×10^{-13}	2.6×10^{-6}	1.4×10^{-11}

In order to illustrate the accuracy of the reconstructed interface we populate the initial interface position with Lagrangian marker particles and advect them with the underlying velocity with a fifth order accurate time integrator [22] for time $t = 4.0$. We compare the final positions of the marker particles with that of the advected interface using Weymouth and Yue's advection scheme and for a grid Courant number of 0.1. The Lagrangian marker particles match the position of the reconstructed interface very well as shown in Fig. 9 thus demonstrating the efficacy of the algorithm.

3.2. Bubble in a stagnation point flow

In this test case we implement the VOF algorithm presented in this paper for a more complex flow. We solve Navier-Stokes equations in one fluid form given by:

$$\rho(C) \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \nabla \cdot [\mu(C) (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] + \rho(C) \mathbf{g} + \mathbf{f}_V' \quad (25)$$

where \mathbf{u} and p are the velocity vector and pressure, respectively, $\rho(C)$ and $\mu(C)$ are the fluid density and viscosity which are a function of volume fraction field, C . We use Chorin's projection method [23] to solve the above equation (25) where we discretized the advection term using a second order ENO scheme [24] and the diffusion terms using central differencing. Surface tension forces, \mathbf{f}_V' are acting only at the interface and have been modeled as volumetric body force using the continuum surface force model of Brackbill, Kothe, and Zemach [25]. The interface is captured using CLSVOF algorithm given by Sussman and Puckett [15]. This algorithm is mass conserving and calculates the curvature and surface normal with high accuracy which is used for surface tension force calculation. The interface is advected by solving the advection equations for level-set function, ϕ , and volume fraction, C . We initialize a toroidal bubble of radius 0.1 at (0.2, 0.5) in a computational domain of unit size, 1×1 . The bottom boundary has an inlet velocity of unity in the upward axial direction and the right boundary has outflow boundary conditions. The top boundary acts as a rigid wall with no-slip and impermeable surface. The density ratio and viscosity ratio is 10 with the Laplace number of the bubble, $La = \frac{\rho D \sigma}{\mu^2} = 0.048$. The incoming axial velocity drags the bubble and flattens it against the top wall, stretching it in the axial direction considerably (see Fig. 10). Even though the fluid interface undergoes drastic change in its shape, the volume is conserved with a high degree of accuracy.

An accurately computed divergence free velocity field is essential for accurate volume conservation in VOF advection. In numerical simulations of incompressible multiphase flows using projection method, this is achieved by solving a pressure Poisson equation to a high degree of accuracy by setting a small tolerance value for solving the matrix system. The order

of error in the divergence free condition for velocity is of the order of the set tolerance value. Solving Poisson equation is the most expensive step in the entire simulation [2], thus studying the effect of limiting tolerance for Poisson solver on the conservation of volume in VOF methods is relevant to determine the accuracy of the simulation. Using the above test case for a grid size of $\frac{1}{128}$ and time step of 5.0×10^{-4} , we study the effect of accuracy of discretized velocity field in mass conservation of the two VOF advection schemes presented in this paper. The divergence of the velocity field is the sum of each discretized value of divergence of velocity at the cell center given by (14) for each computational cell.

Table 6 gives the effect of divergence of discretized velocity field in volume conservation for VOF advection schemes. Sussman and Puckett's operator split approach results in fairly uniform error in volume conservation for lower tolerances since the error due to clipping and filling of the cells to maintain the boundedness of volume fraction field contributes the most to the error. As the divergence free nature of the velocity field deteriorates the error in volume conservation scales down with the tolerance. Due to the very small grid Courant number used in the simulation even for a highly divergence free velocity field there is small error in the volume conservation when Weymouth and Yue's operator splitting method is employed. The relative error scales uniformly with the tolerance and is better than Sussman and Puckett's method.

4. Conclusions

In the present work, we have presented several improvements for the implementation of volume of fluid method in axisymmetric coordinates. We have presented analytical relations for the reconstruction of piecewise linear interface in axisymmetric coordinates similar to those given by Scardovelli and Zaleski [12] for cartesian coordinates. The proposed scheme substantially reduces the computational cost in comparison to the iterative schemes usually employed for the reconstruction. Further, we showed that for axisymmetric coordinate system, machine-precision advection of volume fraction field can be achieved. We illustrated the improvements by comparing the results with the popular open source multiphase flow solver Gerris. Finally, we would like to note that similar modifications in the advection scheme for volume of fluid method in other curvilinear coordinate systems (such as elliptic coordinates) can be derived using the approach presented in this work.

CRedit authorship contribution statement

Ananthan Mohan: Methodology, Software, Writing – original draft. **Gaurav Tomar:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] A. Prosperetti, G. Tryggvason, *Computational Methods for Multiphase Flow*, Cambridge University Press, 2009.
- [2] G. Tryggvason, R. Scardovelli, S. Zaleski, *Direct Numerical Simulations of Gas–Liquid Multiphase Flows*, Cambridge University Press, 2011.
- [3] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* 39 (1) (1981) 201–225.
- [4] Y. Renardy, M. Renardy, Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method, *J. Comput. Phys.* 183 (2) (2002) 400–421.
- [5] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, *Num. Methods Fluid Dyn.* (1982).
- [6] J.E. Pilliod, An Analysis of Piecewise Linear Interface Reconstruction Algorithms for Volume-of-Fluid Methods, Univ. of Calif., Davis, 1992.
- [7] J.E. Pilliod Jr, E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, *J. Comput. Phys.* 199 (2) (2004) 465–502.
- [8] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection, *Int. J. Numer. Methods Fluids* 41 (3) (2003) 251–274.
- [9] D.L. Youngs, An interface tracking method for a 3d Eulerian hydrodynamics code, *AWRE Tech. Rep.* 44 (92) (1984) 35.
- [10] M.M. Francois, B.K. Swartz, Interface curvature via volume fractions, heights, and mean values on nonuniform rectangular grids, *J. Comput. Phys.* 229 (3) (2010) 527–540.
- [11] V. Dyadechko, M. Shashkov, Moment-of-fluid interface reconstruction, Los Alamos Report LA-UR-05-7571, 2005.
- [12] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, *J. Comput. Phys.* 164 (1) (2000) 228–237.
- [13] S. Diot, M.M. François, E.D. Dendy, An interface reconstruction method based on analytical formulae for 2d planar and axisymmetric arbitrary convex cells, *J. Comput. Phys.* 275 (2014) 53–64.
- [14] S. Bnà, S. Manservigi, R. Scardovelli, P. Yecko, S. Zaleski, Vofi—a library to initialize the volume fraction scalar field, *Comput. Phys. Commun.* 200 (2016) 291–299.
- [15] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows, *J. Comput. Phys.* 162 (2) (2000) 301–337.
- [16] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, W.J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, *J. Comput. Phys.* 130 (2) (1997) 269–282.
- [17] G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* 5 (3) (1968) 506–517.
- [18] G.D. Weymouth, D.K.-P. Yue, Conservative volume-of-fluid method for free-surface simulations on Cartesian-grids, *J. Comput. Phys.* 229 (8) (2010) 2853–2865.
- [19] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries, *J. Comput. Phys.* 190 (2) (2003) 572–600.
- [20] W. Rider, D. Kothe, Stretching and tearing interface tracking methods, in: 12th Computational Fluid Dynamics Conference, 1995, p. 1717.

- [21] R.L. Panton, *Incompressible Flow*, John Wiley & Sons, 2013.
- [22] S.D. Cohen, A.C. Hindmarsh, P.F. Dubois, Cvode, a stiff/nonstiff ode solver in c, *Comput. Phys.* 10 (2) (1996) 138–143.
- [23] A.J. Chorin, Numerical solution of the Navier-Stokes equations, *Math. Comput.* 22 (104) (1968) 745–762.
- [24] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* 77 (2) (1988) 439–471.
- [25] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* 100 (2) (1992) 335–354.