



Richards C Sunny · Juan Horrillo · Wei Cheng

# VOF with center of mass and Lagrangian particles (VCLP): a surface tracking and advection method for incompressible fluids

Received: 12 March 2022 / Accepted: 23 September 2022 / Published online: 16 October 2022  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

**Abstract** A novel surface tracking, and advection algorithm for incompressible fluid flows in two and three dimensions is presented. This method based on the volume-of-fluid (VOF) method, is named VOF-with-center-of-mass-and-Lagrangian-particles (VCLP), and it uses spatially and temporally localized Lagrangian particles (LPs) inside a finite volume framework. The fluid surface is recaptured and reconstructed piecewise using the mean slope and curvature. The fluid mass inside each cell is discretized spatially by LPs and distributed as blue noise. LPs are then advected cell by cell with a choice of two different advection schemes in time using interpolated velocity and approximated acceleration fields. VCLP continuously tracks the center of mass of the fluid parcels in the Lagrangian way and this helps to reduce the errors due to numerical acceleration that results from lack of information to reconstruct the interface accurately. VCLP's performance is evaluated using standard benchmark tests in 2D and 3D such as translation, single vortex, deformation, and Zalesak's tests from the literature. VCLP is applied to TSUNAMI2D, a 2D Navier–Stokes model to simulate shoaling and breaking of waves.

**Keywords** VCLP · Surface tracking · Fluid advection · VOF · TSUNAMI3D

## 1 Introduction

While modeling two-phase immiscible flows with a significant difference in molecular viscosity and density, the presence of the lighter phase can be neglected, and then the interface is considered as free-surface or surface. This assumption is valid for modeling liquid-gas flows such as ocean waves because of the significant difference in pressure and inertia. Free surface flows are commonly solved using Lagrangian and Eulerian approaches. The early work with the Lagrangian approach for interface tracking is found in the Marker and Cell method (MAC) [1]. In the MAC method, the velocity field models the fluid flow, while the markers represent the fluid in space, and fluid flow is simulated by moving the marker particles through space as per the interpolated velocities from the velocity field. Peskin [2] used the immersed boundary method to analyze the blood flow in the heart where the moving heart walls were treated as inner boundaries. Other Lagrangian methods use a deformable mesh that conforms to the interface at all times as in the front-tracking method

---

Communicated by Shervin Bagheri.

R. C. Sunny · J. Horrillo (✉) · W. Cheng  
Ocean Engineering Department, Texas A&M University, 200 Seawolf Pkwy, Galveston, TX 100190, USA  
E-mail: horrillj@tamug.edu

R. C. Sunny  
E-mail: richardscs@tamu.edu

W. Cheng  
E-mail: chengwei@tamug.edu

[3] and Arbitrary Lagrangian Eulerian (ALE) methods [4–6]. In the front-tracking method, the flow field is discretized on a stationary grid using conservative finite difference approximation, while the interface is identified using an unstructured grid that can move through the stationary grid along with it. The Lagrangian approach is not suitable for fluid flows with large deformations, fragmented interfaces, and fluid-structure interactions as the interfaces are defined as internal boundaries [7]. Even though re-meshing is a solution, it comes with higher computational cost and interpolation errors. Another Lagrangian method called Particle-in-cell (PIC) method, have discrete mass points representing the fluids, and it has been used since the 1960's [8,9]. Differential equations of mass and momentum characterize flow and energy in this method, and mass points move by weighted velocities. Eulerian and Lagrangian frameworks have been coupled previously to avoid shortcomings and to complement each other. Semi-implicit surface tension formulation with a Lagrangian surface mesh on an Eulerian simulation grid by Schroeder [10] and enhancement of pressure calculation by the incorporation of background mesh scheme by Wang [11] are some cases. For interface reconstruction, an extension of the VOF method was introduced by Dyadechko & Shashkov [12] called the moment of fluid method (MoF) that keeps track of the cell-wise material centroids and uses this information to reconstruct interface by minimizing the defect of the first moment along with volume conservation constraint. This method allows the reconstruction of the interface with no information on neighboring cell fluid volume. Very recent work by Wenzel and Garrick [13] introduced the point-mass particle method, which is an Eulerian–Lagrangian approach where Navier–Stokes equations are solved in the Eulerian grid while Lagrangian particles discretize the fluid mass and phase information. Further review on interface tracking methods can be found in [14–17]. Among the interface capturing methods for incompressible fluids, the volume-of-fluid (VOF) method introduced by Hirt and Nichols in 1979 is the most popular [18,19]. VOF's computational efficiency, straightforward treatment of changes in interface topology, and mass conservation makes it a success [20]. VOF is characterized by a static or moving mesh and a discontinuous scalar function that gives the percentage of fluid present inside each mesh cell.

VOF can reconstruct the interface from the  $f$  values. The mean normal direction of the interface could be found out from the spatial gradient of the  $f$  function. The evolution of the  $f$  function is given by  $Df/Dt = 0$ , which states that the  $F$  moves with the fluid. One of the main advantages of the VOF method is that it stores the minimum number of variables and follows the region instead of an interface, and avoids the problems associated with the intersecting and breaking surfaces. In the original VOF advection, the piecewise constant scheme was used, i.e., the interface inside each cell is parallel to one of the axes. An improvement to this method can be seen in Nichols et. al. [21], Hirt and Nichols [19], Torrey et al. [22] and Duff [23]. Improvement is facilitated via an additional degree of freedom to form a stair shaped profile within a cell, according to the local spatial distribution of fluid mass. Piecewise-linear schemes are an improvement in terms of accuracy to piecewise-constant schemes, even though they are more complex and computationally expensive. Since the fluid volume is conserved, the interface is located uniquely from interface normal. Interface normal can be estimated using many algorithms in the context of VOF, but all of them rely on the material volume data from the neighboring cells for interface reconstruction. Some notable piecewise linear schemes include Rider and Kothe [20], Harvie and Fletcher [24,25], Geuyffier et al. [26], and Scardovelli and Zaleski [27,28].

We present a novel surface tracking, and advection algorithm based on the VOF method for incompressible fluid flows in two and three dimensions. This method is named VOF-with-center-of-mass-and-Lagrangian-particles (VCLP), and it uses spatially and temporally localized Lagrangian particles (LPs).

The scope of VCLP is the need for a surface tracking and advection algorithm that combine the strengths of Eulerian methods, such as computational speed and fewer memory requirements and the higher accuracy and surface tracking ability of Lagrangian methods with customizable speed vs. accuracy using a single parameter. The following sections explain its methodology and validation in detail.

## 2 VOF with the center of mass and Lagrangian particles (VCLP) method

The computational domain  $\Omega$  is discretized into  $n_X, n_Y, n_Z$  Cartesian grid cells in positive  $X, Y, Z$  directions. The space occupied by any cell is identified by  $s(i,j,k)$  and its dimensions are  $\delta x_i, \delta y_j$ , and  $\delta z_k$ .  $(i,j,k,m)$  identifies the cell at  $s(i,j,k)$  during the  $m^{th}$  time step of simulation. Now, a binary value function  $F(x, m)$  with outputs zero and one, based on the VOF method, is defined at every point in computational domain. For the cell space  $s(i, j, k, m)$  where  $F$  is defined everywhere, the average of value of  $F$  function gives another function  $f$  known as fractional fluid volume which is a scalar value defined for every cell in the domain with values between 0 and 1.  $f_m$  is a matrix of dimension  $n_X \times n_Y \times n_Z$  filled with  $f(i, j, k, m)$  values, of all cells. Now,

consider three matrices of same dimension containing the velocities  $u(i, j, k, m)$  corresponding to every cell in  $\Omega$ , represented as  $U_{X,m}$ ,  $U_{Y,m}$ , and  $U_{Z,m}$ . Together these three matrices is denoted as  $U_m$ . Similarly the CM matrices is denoted as  $c_m$ . Then the entire purpose of VCLP can be conveyed by the simple relation between input and output parameters given by Eq. 1.

$$\text{VCLP (input parameters)} = (\text{output parameters})$$

$$\boxed{VCLP(f_m, c_m, U_m) = (f_{m+1}, c_{m+1})} \quad (1)$$

Thus VCLP needs velocity field, fluid distribution, and the CM location of the given time step to provide the fluid distribution, and the CM for the next time step. So it takes seven/four scalar input/output in 3D and five/three in 2D.

### VCLP procedure

The VCLP procedure can be summarized as follows.

1. Entering time step  $t_m$
2. Receives velocity field, fluid distribution ( $f$ ), and center of mass locations from the solver.
3. Classification of all cells in to five; Empty, Surface, Inner I, Inner II, Inner III.
4. Interface reconstruction calculation of all Surface cells using PLIC or proposed PCIC method.
  - Finds interface normal(shape parameter) using proposed T- Slope Method.
  - Finds interface curvature using proposed methods.
  - Locates interface using proposed sorted surface constant (SSC) methods.
5. Create an empty copy of the domain for time step  $t_{m+1}$  (empty  $f$  matrix and center of mass locations)
6. Starts working on a single cell
  - Fill the selected cell (Surface, Inner I, Inner II) with LPs using the blue noise template created at the beginning of simulation. Surface cells have PLIC or PCIC interfaces
  - Center of mass of the filled cell is estimated and corrected using the center of mass location that is being tracked and obtained from solver
  - LPs (Fluid) in the selected cell is advected using Taylor's second-order advection scheme or the proposed TJSM scheme.
    - Velocity field of LPs are estimated via trilinear interpolation
    - Acceleration field of LPs are calculated using proposed method
  - The advected LPs are received by one or more cells in the domain created for  $t_{m+1}$
  - The center of mass of all cells that received LPs are updated
  - Moves on to the next surface, inner I, or Inner II cell to advect
7. Once advection of all cells are completed, filters are applied and steps required for skip core optimization is carried out.
8. The domain for  $t_{m+1}$  is complete with new fluid distribution and center of mass location
9.  $m = m + 1$

Notes: The blue noise template of LPs needs to be created only once at the beginning of simulation. The center of mass locations of cells for the first time step is calculated from the geometry and initial conditions using LPs and hence doesn't require correction.

### 2.1 Classification of cells

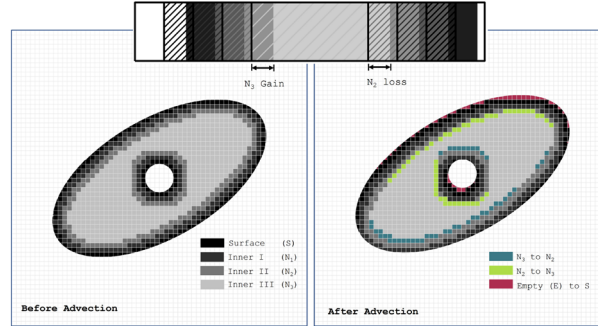
In typical VOF method there are empty ( $E$ ), surface ( $S$ ), and interior cells ( $N$ ) respectively. In VCLP, interior cells are further classified in to three levels, Inner I, Inner II, and Inner III and all classifications are based on cell-type value denoted by  $C_T$  ( Table 1 ) and estimated using the following steps 1 to 6, where  $q$  is the ceiling function value of the Courant-Friedrichs-Lewy (CFL) number ( $q = \lceil CFL \rceil$ ) and the CFL number is given by

$$CFL = \frac{u_{max} \Delta t}{\delta x}. C_T^* \text{ is the temporary } C_T \text{ value which gets updated along the steps.}$$

1.  $C_T^* = \lfloor f(i, j, k, m) \rfloor$
2.  $C_{T1} = \sum_{k=k-q}^{k+q} \sum_{j=j-q}^{j+q} \sum_{i=i-q}^{i+q} C_T^*$

**Table 1** Classification of cells in the computational domain  $\Omega$ 

Class	Notation	$C_T$
Empty	$E$	0
Surface	$S$	1
Inner I	$N_1$	2
Inner II	$N_2$	3
Inner III	$N_3$	4
Interior	$N$	2,3,4



**Fig. 1** Cell Classification: Top: Rectangular block is a simplified 1-D translation diagram with CFL = 0.75 showing if  $N_3$  is not advected the  $N_3$  mass gain equals  $N_2$  mass loss. Hatched is before advection and filled is after. Left: an ellipsoidal fluid body with a circular hole before advection. Right: fluid body after advection with  $q = 1$  and a uniform velocity field of  $\cos(3\pi/4)\hat{i} + \sin(3\pi/4)\hat{j}$ . Left: Classification of cells before advection. Right: cells with changes in classification after advection and the corresponding change

$$\begin{aligned}
 3. \quad C_T &= \begin{cases} 2 & \text{where } C_{T1} = (2q + 1)^3, f > 0.5 \\ C_T^* & \text{otherwise} \end{cases} \\
 4. \quad C_{T2} &= \sum_{k=k-q}^{k+k+q} \sum_{j=j-q}^{j+j+q} \sum_{i=i-q}^{i+i+q} C_T^* \\
 5. \quad C_T^* &= \begin{cases} 3 & \text{where } C_{T2} = 2(2q + 1)^3 \\ C_T^* & \text{otherwise} \end{cases} \\
 6. \quad C_T &= \begin{cases} 4 & \text{where } C_{T3} = 3(2q + 1)^3 \\ C_T^* & \text{otherwise} \end{cases}
 \end{aligned}$$

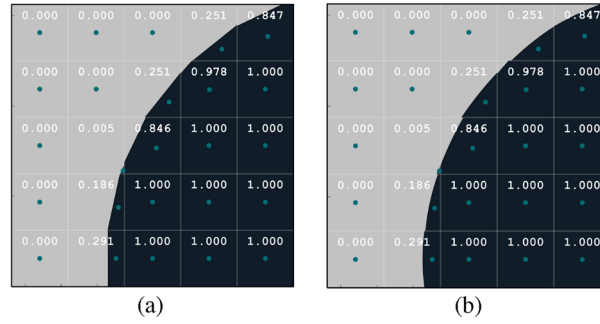
Figure 1 shows an ellipsoidal fluid body with a circular hole in it inside a square computational domain and the classification of cells before and after advection based on steps above.

## 2.2 Interface reconstruction

Interface is reconstructed for all cells classified as surface cell according to  $C_T$  value. There are two choices for interface reconstruction calculation; Piecewise Linear Interface Calculation (PLIC) method and the Piecewise-Circular-Interface-Calculation method (PCIC). PLIC is used by many models and it creates a line segment representing the interface in 2D and a plane in 3D, whereas PCIC which we propose creates a circular arc for the interface in 2D and a spherical surface in 3D. Linear interface reconstruction denoted as Eq. 2,

$$L(i, j, k, m) : n_x x + n_y y + n_z z + a = 0 \quad (2)$$

needs interface normal vector  $n = (n_x, n_y, n_z)$  and  $a$ , the planar constant. PCIC is build on top of PLIC meaning PLIC calculation is required for PCIC. The PCIC reconstruction ( $C$ ) is given by Eq. 3,



**Fig. 2** **a** Reconstructed PLIC interfaces and **b** Reconstructed Piecewise Circular Interface Calculation (PCIC), using  $f$  values. Center of mass locations are shown as dots. Fluid body is circular with a radius of 0.15 inside a unit square domain with a resolution of  $32 \times 32$

$$C : (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - R^2 = 0$$

$$\begin{cases} x_{i-1} < x \leq x_i \\ y_{j-1} < y \leq y_j \\ z_{k-1} < z \leq z_k \end{cases} \quad (3)$$

where  $(x_0, y_0, z_0)$  is the sphere's center and  $R$  its radius. The additional information required for PCIC than PLIC is the radius of the sphere  $R$ , or equivalently the interface curvature which is the reciprocal of  $R$  denoted by  $\kappa$ .

Figure 2a depicts the reconstructed fluid domain along the interfaces using PLIC and Fig. 2b shows the fluid domain reconstructed with PCIC.

It can be seen that the PCIC interface reconstructions are more accurate and the radius of curvatures are close to the true value (circle of radius 0.15 m is on cell size of  $1/32 \times 1/32$  m).

#### Interface normal and T-slope method

The interface normal is given by Eq. 4,

$$n = \frac{\nabla F}{|\nabla F|} \approx \frac{\nabla f}{|\nabla f|} \quad (4)$$

but since  $F$  is discretized into  $f$ , numerically its approximate value ( $X$  component) is obtained using Eq. 5.

$$\begin{aligned} \nabla_{n_x} f &= \frac{f_{(x)}(i+1) - f_{(x)}(i)}{\delta x_{i+1} + \delta x_i} \\ &+ \frac{f_{(x)}(i) - f_{(x)}(i-1)}{\delta x_{i-1} + \delta x_i} \end{aligned} \quad (5)$$

where,

$$f_{(x)}(i) = \frac{\sum_{k=k-1}^{k=k+1} \sum_{j=j-1}^{j=j+1} f(i, j, k) \delta y_j \delta z_k}{\sum_{k=k-1}^{k=k+1} \sum_{j=j-1}^{j=j+1} \delta y_j \delta z_k}$$

In literature, the two common ways to estimate the interface normal are by finding the  $\nabla_n f$  using above mentioned equations and by using a method that deals with minimum slope with respect to the axes. We may call these numerical gradient method and the minimum slope error method which can be learned from [19]. VCLP uses a new method that we propose, called T-Slope method to find the interface normal and Eqs. 6–12. shows how the interface normal is estimated in 2D based on the calculated intermediate parameters

$S_x$ ,  $S_y$ ,  $H_x$ ,  $H_y$ ,  $F_x$ ,  $F_y$ ,  $\alpha$  and  $\theta$ . The first two parameters  $S_x$  and  $S_y$  having units of length represent the change of volume with distance and they are given by Eqs. 6 and 7.

$$S_x = \sum_{j=j-1}^{j=j+1} \delta y_j \left( \frac{f_{i+1,j} \delta x_{i+1} - f_{i,j} \delta x_i}{\delta x_i + \delta x_{i+1}} \right) + \delta y_j \left( \frac{f_{i,j} \delta x_i - f_{i-1,j} \delta x_{i-1}}{\delta x_i + \delta x_{i+1}} \right) \quad (6)$$

$$S_y = \sum_{i=i-1}^{i=i+1} \delta x_i \left( \frac{f_{i,j-1} \delta y_{j-1} - f_{i,j} \delta y_j}{\delta y_j + \delta y_{j-1}} \right) + \delta x_i \left( \frac{f_{i,j} \delta y_j - f_{i,j+1} \delta y_{j+1}}{\delta y_j + \delta y_{j-1}} \right) \quad (7)$$

Then, Eq. 8 is used to estimate the two slopes parameters  $H_x$  and  $H_y$  which are non-dimensional.

$$H_x = \sum_{j=j-1}^{j=j+1} \delta y_j \left( \frac{f_{i+1,j} - f_{i,j}}{\delta x_i + \delta x_{i+1}} + \frac{f_{i,j} - f_{i-1,j}}{\delta x_i + \delta x_{i+1}} \right) \quad (8)$$

$$H_y = \sum_{i=i-1}^{i=i+1} \delta x_i \left( \frac{f_{i,j-1} - f_{i,j}}{\delta y_j + \delta y_{j-1}} + \frac{f_{i,j} - f_{i,j+1}}{\delta y_j + \delta x_{j+1}} \right)$$

Equation 9 estimates two non-dimensional parameters which are independent of the cell dimensions.

$$F_x = \frac{\sum_{j=j-1}^{j=j+1} f_{i+1,j} - \sum_{j=j-1}^{j=j+1} f_{i-1,j}}{\sum_{j=j-1}^{j=j+1} f_{i,j}} \quad (9)$$

$$F_y = \frac{\sum_{i=i-1}^{i=i+1} f_{i,j-1} - \sum_{i=i-1}^{i=i+1} f_{i,j+1}}{\sum_{i=i-1}^{i=i+1} f_{i,j}}$$

The magnitudes of these two parameters  $F_x$  and  $F_y$  are used to estimate another parameter  $\alpha$  as in Eq. 10.

$$\alpha = \begin{cases} 1 & \text{if } |F_x| < |F_y| \\ 0 & \text{else} \end{cases} \quad (10)$$

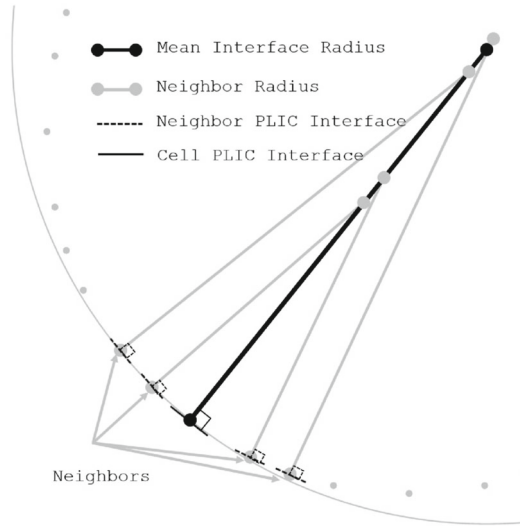
The calculated parameters  $S_x$ ,  $S_y$ ,  $H_x$ ,  $H_y$ ,  $F_x$ ,  $F_y$  and  $\alpha$  are used as in Eq. 11 to estimate  $\theta$  which is the angle of rotation of interface with respect to the X-axis.

$$\theta = \begin{cases} \tan^{-1}(H_y) + \pi & \text{if } \alpha = 0 \text{ \& } S_x \geq 0 \\ -\tan^{-1}(H_y) & \text{if } \alpha = 0 \text{ \& } S_x < 0 \\ \tan^{-1}(H_x) + \frac{\pi}{2} & \text{if } \alpha = 1 \text{ \& } S_y \leq 0 \\ -\tan^{-1}(H_x) - \frac{\pi}{2} & \text{if } \alpha = 1 \text{ \& } S_y > 0 \end{cases} \quad (11)$$

Finally, the components on the interface normal are estimated using Eq. 12.

$$n_x = \cos \theta, \quad n_y = \sin \theta \quad (12)$$

A two-dimensional circular fluid domain is reconstructed from the actual  $f$  values, i.e., from the intersection area of the circle and cell boundaries using PLIC with the three methods, numerical gradient, minimum slope, and T-Slope. Methods are tested for two aspect ratios and the results are compared in appendix A.



**Fig. 3** Interface curvature in 2D for the PCIC reconstruction. Normals are drawn from the mid points of nearest four neighboring PLICs to the normal of the surface cell in consideration, passing through its PLIC reconstruction's midpoint. Neighbor curvatures are averaged to get the PCIC interface curvature

### Interface curvature and PCIC

The Piecewise-Circular-Interface-Calculation (PCIC) is a proposed alternative to PLIC, but with a higher order. It constructs a circular arc (2D) or a spherical surface (3D) for surface cells and a PLIC reconstruction is equivalent to a PCIC reconstruction with infinite radius. As PCIC have higher order than PLIC, improved accuracy is expected in the reconstructed interface, and subsequently in the simulated flow. Since the curvature information can also be used to estimate the surface tension force, it is expected to improve accuracy in related scenarios also.

The procedure developed in 2D to estimate the interface curvature is shown in Fig. 3, where normals drawn at the four neighbor surface cells and their intersection with the normal of the surface cell in consideration provides the neighbor radius'. The interface curvature is estimated by taking average of four neighbor curvatures and its reciprocal gives the radius for the PCIC as given in Eq. 13,

$$R_m = \frac{4}{\sum_{k=1}^{k=4} \frac{1}{r_k}} \quad (13)$$

where  $R_m$  is the mean radius of curvature and  $r_k$  is the neighbor radius.

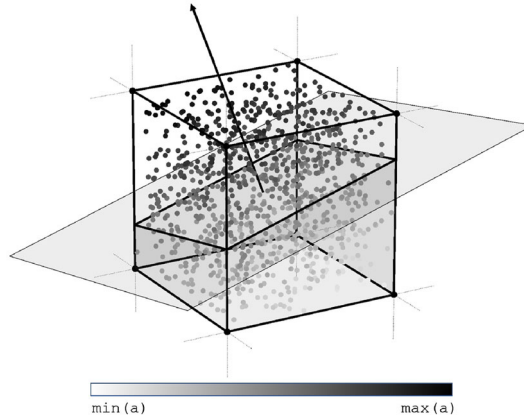
Neighbor radius  $r_k$  is found using Eq. 14,

$$r_k = \frac{|n_{ex}(y_0 - m_y) - n_{ey}(x_0 - m_x)|}{\sqrt{n_x n_{ey} - n_y n_{ex}}} \quad (14)$$

where  $(x_0, y_0)$  is the mid point of the PLIC reconstruction and  $(n_x, n_y)$  is its normal passing through its midpoint,  $(m_x, m_y)$ .  $(n_{ex}, n_{ey})$  is the normal of the neighbor's PLIC reconstruction. For estimating interface curvature in 3D, a new method is developed using the interface normals located at the centroids of PLIC reconstructions inside the cell and inside its three nearest neighbor surface cells. Let's consider the centroid of the cell of consideration as  $e_0$  and its three neighbors as  $e_1, e_2$  and  $e_3$ . Similarly four normals at the centroids are denoted as  $n_0$  to  $n_3$ . Interface curvature ( $\bar{\kappa}$ ) is given by Eq. (15) where  $\nabla \cdot n$  is the divergence of normal at the PLIC centroid.

$$R = \frac{1}{\bar{\kappa}} = -\frac{2}{\nabla \cdot n}, \quad \nabla \cdot n = \frac{\partial n_x}{\partial x} + \frac{\partial n_y}{\partial y} + \frac{\partial n_z}{\partial z} \quad (15)$$





**Fig. 4** Estimation of planar constant  $a$  using the Sorted Surface Constant (SSC) method for the surface cell given the interface normal and the  $f$ -value

The three components of  $\nabla \cdot n$  are found using set of Eq. 16,

$$\begin{aligned}
 D \frac{\partial n_x}{\partial x} &= X_1(y_2z_3 - y_3z_2) + X_2(y_3z_1 - y_1z_3) \\
 &\quad + X_3(y_1z_2 - y_2z_1) \\
 D \frac{\partial n_y}{\partial y} &= Y_1(x_3z_2 - x_2z_3) + Y_2(x_1z_3 - x_3z_1) \\
 &\quad + Y_3(x_2z_1 - x_1z_2) \\
 D \frac{\partial n_z}{\partial z} &= Z_1(x_2y_3 - x_3y_2) + Z_2(x_3y_1 - x_1y_3) \\
 &\quad + Z_3(x_1y_2 - x_2y_1) \\
 D &= x_1(y_2z_3 - y_3z_2) + x_2(y_3z_1 - y_1z_3) \\
 &\quad + x_3(y_1z_2 - y_2z_1)
 \end{aligned} \tag{16}$$

Note that here  $n_0 - n_k = (X_k, Y_k, Z_k)$  and  $e_0 - e_k = (x_k, y_k, z_k)$  where  $k=1, 2,$  and  $3$  representing the three nearest neighboring surface cells.

#### Locating interfaces and SSC

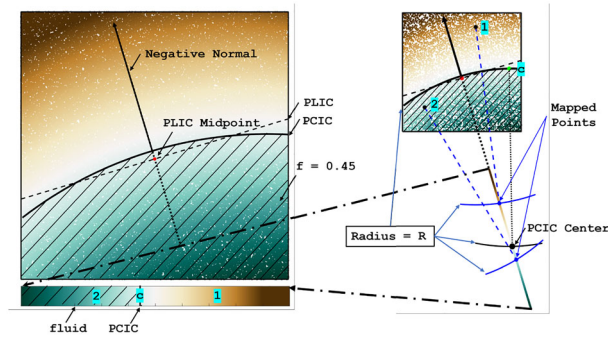
The planar constant ‘ $a$ ’ in PLIC and the center of the sphere for PCIC needs to be calculated as the next step. Two common approaches to locate the interface are the analytical and the numerical iterative methods like the bisection method [29]. Analytical solution is accurate but computationally intensive, especially in 3D. Bisection method is an iterative method stopped when the error is below a predefined limit. VCLP uses a new numerical method named the Sorted-Surface-Constant method (SSC) which works on the principle that planar constants along interface normal vary linearly as illustrated in Fig. 4.

SSC method works for structured, unstructured grids in 2D and 3D with the same efficiency given boundaries are well defined. The application of SSC method for PCIC is depicted in the Fig. 5. Detailed explanation and steps are provided in “Appendix A”.

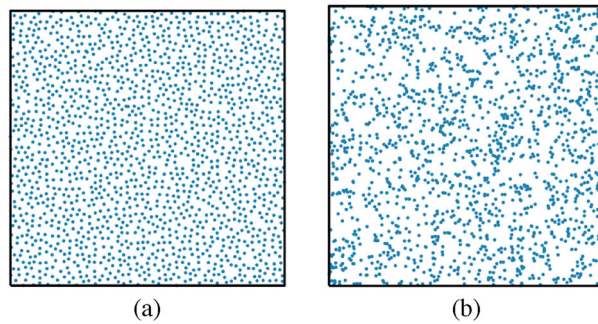
#### 2.3 Lagrangian particles and center of mass tracking

While advecting in VCLP, the fluid inside the cell is represented using Lagrangian particles (LPs) as concentrated mass points. A single template of fluid mass points generated with blue noise distribution is used. Here, LPs representing the fluid are advected cell by cell, and after advection of one cell, the same LPs template is used for the next cell advection. This is an important difference to the common Lagrangian approaches, where the entire fluid domain or interface are represented simultaneously by LPs and all LPs are tracked through the





**Fig. 5** Sorted Surface Constant (SSC) method for PCIC. Each  $N$  points inside cell is mapped to the normal through the PLIC midpoint such that intersecting point is at distance  $R$ . Points are then sorted (color) according to the signed distance (surface constant value) from PLIC midpoint to the intersecting point (example: points 1, 2).  $f N^{\text{th}}$  surface constant provides the center of circle 'c', making fraction of number of points inside PCIC to be  $f$



**Fig. 6** Blue noise and white noise templates

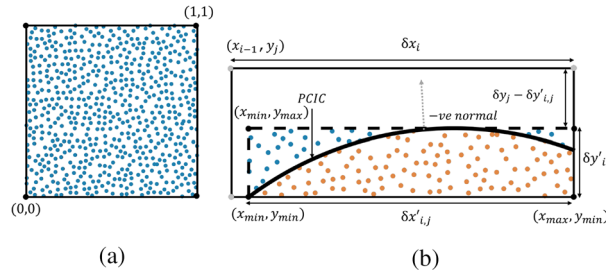
simulation time, requiring very large number of LPs proportional to the domain size for the same resolution. VCLP uses only one blue noise template of LPs with custom number of particles, and hence only a total of 100–1000 LPs are required for the entire simulation irrespective of the domain size in 2D, and 1000 to 8000 LPs in 3D.

### 2.3.1 Lagrangian particle creation

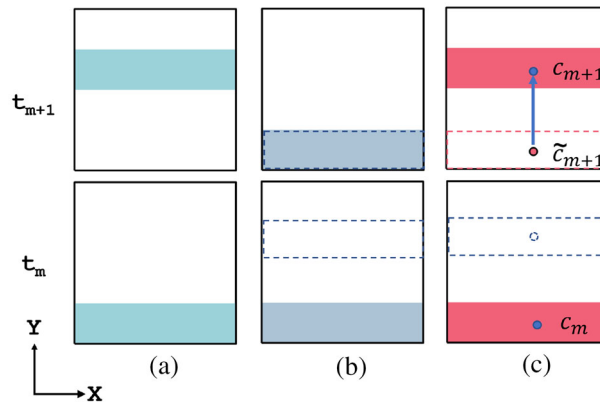
The creation of LPs in 2D and 3D is simple as creating two and three arrays of uniformly distributed random numbers within 0 and 1 (white noise) that represent the coordinates of their location. As LPs represent the fluid with uniform density, the ideal scenario is uniformly distributed LPs without any spatial bias. But most arrangements such as LPs at the vertices of rectangles or hexagons have directional bias but not for a uniformly distributed white noise and blue noise (Poisson disk sampling) [30]. The difference between these sampling techniques can be seen in Fig. 6.

It can be noted that the blue noise has no clusters and every small region looks similar representing uniform density in a better way. Tests were conducted with white and blue noise and proved the same and hence a blue noise template is used. This template is created only once at the start of simulation and can be reused in each cell to advect the fluid for all time steps. If a finer resolution is opted, then a template with larger number of LPs or density is generated. Blue noise template is generated using a new procedure and is provided in appendix. The template is used to populate the fluid region in one cell, and is then advected, then the code moves on to the next cell till entire fluid domain is advected for the given time step. The template is used to fill the cell completely or partially after which LPs outside the PCIC reconstruction  $C$  are removed representing the fluid region. Figure 7 shows the modification of blue noise template on the left side into a representation of fluid in a surface cell with a PCIC interface and elimination function  $E$  as defined in Eq. 17,

$$E = \begin{cases} \text{sgn} \{ n_x p_x + n_y p_y + n_z p_z + a \} : L \\ \text{sgn} \{ \|p - x_0\| - R^2 \} \text{sgn} \{ \nabla \cdot n \} : C \end{cases} \quad (17)$$



**Fig. 7** Modification of the LPs template: **a** Initial Lagrangian particles unit square template with  $N = 25$  particles per cell dimension uniformly distributed. **b** Region of particles from the template is scaled in to rectangle that fills fluid region with no change in the aspect ratio. The orange colored LPs represent the fluid inside PCIC reconstruction. The blue colored LPs outside the interface are removed



**Fig. 8** Center of mass (CM) tracking: A rectangular fluid body (A  $t_m$ ) is inside a square cell of 1 m dimension with a velocity field  $u_y = 0.5$  m/s. The ideal state of fluid body at the start (lower panel) and after one second (upper panel) is shown in **A**. **B** the numerical simulation of fluid flow for a CFL value of 0.5 without CM tracking. **C** the numerical simulation of fluid flow for a CFL value of 0.5 with CM tracking. Blue dot is the true CM ( $c$ ) which is tracked in the Lagrangian way, while red dot, the estimated CM ( $\tilde{c}_{m+1}$ ). The reconstructed fluid region is translated to match the true CM shown by the arrow

is used to determine whether a LPs are retained or not. LPs are eliminated if the elimination function value is negative. A limiting cell subdomain is constructed to improve the accuracy and efficiency in the calculations pertaining to advection. The subdomain is the smallest rectangle that can accommodate the fluid region. The subdomain boundary is shown by the dashed line in the Fig. 7.

### 2.3.2 Center of mass tracking

Accuracy of an interface reconstruction is increased by tracking of the center of mass(CM) of fluid cell following the Lagrangian approach. In theory if the grid cell size is infinitesimal, this converges to pure Lagrangian method. So for control volumes, tracking CM in Lagrangian way is essentially tracking the entire fluid whole time, but with minimal number of points, one per cell. This is the one of the main idea behind the VCLP method. During an interface reconstruction CM mostly will not be match exactly due to different reasons, mainly due to the simplification of the interface into PLIC or PCIC, that too reconstructed from the mean slope and curvature that itself is an approximation. These errors make the fluid move slower or faster in the wrong direction, depending on the errors and velocity-field.

At a given time step, there are two CMs for a cell, the tracked CM denoted as  $c$ , and the estimated CM after the reconstruction given by  $\tilde{c}(i, j, k, m)$ . The LPs representing the fluid inside the cell before the CM correction is given by  $\tilde{\Lambda}(i, j, k, m)$ . The  $X, Y$ , and  $Z$  coordinates of LPs are  $\tilde{\Lambda}_x(i, j, k, m)$ ,  $\tilde{\Lambda}_y(i, j, k, m)$ , and  $\tilde{\Lambda}_z(i, j, k, m)$ . The LPs after CM correction is denoted by  $\Lambda$ . Estimated CM location ( $\tilde{c}$ ) is found using the Eq. 18.  $\bar{\Lambda}$  is the average of the positions all LPs inside the cell space  $s(i, j, k, m)$ .

$$\tilde{c}(i, j, k, m) = \bar{\Lambda}(i, j, k, m) \tag{18}$$

Once  $\tilde{c}$  is known, the positions of the LPs representing the fluid are corrected using the following Eq. 19.

$$\Lambda = \tilde{\Lambda} + c - \tilde{c} \quad (19)$$

When the fluid inside the cell is represented by LPs, each LP would represent a fraction of the total mass of that fluid within the cell. That mass denoted by  $m_p$  is estimated using Eq. 20, where  $N_{\Lambda(i,j,k,m)}$  is the number of LPs in  $\Lambda(i, j, k, m)$ .

$$m_p(i, j, k, m) = \rho \frac{|s(i, j, k, m)| f(i, j, k, m)}{N_{\Lambda(i,j,k,m)}} \quad (20)$$

## 2.4 Fluid advection

VCLP have two choices numerical advection schemes to advect the LPs representing fluid; Taylor's second-order numerical advection scheme and a new proposed scheme named Tracking Journey in Spiral (TJSM) method. Taylor's second order method is given by Eq. 21,

$$p_{m+1} = p_m + u(p_m, t_m) \Delta t_m + \frac{1}{2} a(p_m, t_m) \Delta t_m^2 \quad (21)$$

but note that the acceleration field in this equation is not available from solver. This equation considers a single LP whose position at time step  $t_m$  is denoted by  $p_m = p_x(m)\hat{i} + p_y(m)\hat{j} + p_z(m)\hat{k}$ . The velocity and acceleration is denoted by  $u(p_m, t_m)$  and  $a(p_m, t_m)$ , respectively with time step  $\Delta t_m = t_{m+1} - t_m$ .

### 2.4.1 TJSM advection scheme

TJSM integrates the equation of motion like others, while trying to capture rotational effects inside a velocity field more effectively. Instead of a linear motion, TJSM assumes that the particle undergoes a circular motion with a constant angular velocity during the time step intervals. In the higher-order TJSM, thus particle could move following a spiral path, hence the name. The developed TJSM advection scheme is given by Eq. 22,

$$p_{m+1} = p_m + \frac{\{(\beta \times u_m)(1 - \cos \theta_m) - u_m \|\beta\| \sin \theta_m\} \|u_m\|^2}{\|\beta\|^2} \quad (22)$$

where  $\beta = u_m \times a_m$  and  $\theta_m$  given by Eq. 23.

$$\theta_m = \|\omega_m\| \Delta t_m = \frac{\Delta t_m \|u_m \times a_m\|^5}{\|u_m\|^2 \|\beta \times u_m\|^2} \quad (23)$$

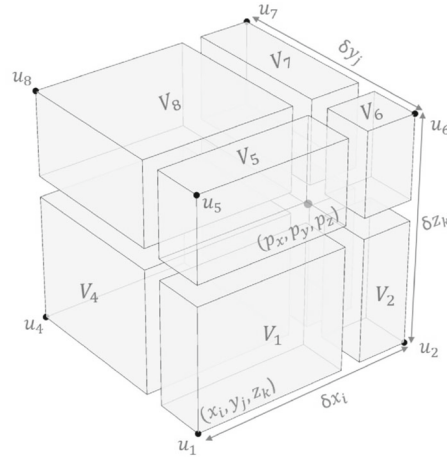
If  $u_m = 0$ , or  $u_m \times a_m = 0$ , then Taylor's second order advection method is to be used to advect LPs. The performance is tested using the deformation benchmark test for the interface tracking in 2D and 3D. As the advection scheme alone is tested, LPs are moved in purely Lagrangian way. The derivation of TJSM scheme in 3D and test results are provided in the appendix.

### 2.4.2 Acceleration field for advection

For second-order advection methods described in Eqs. 21 and 22, velocity and acceleration terms are required to advect the particle. Figure 9 shows a LP at position  $(p_x, p_y, p_z)$  within the cell.

Velocities at the eight vertices are  $u_1$  to  $u_8$ .  $V_1$  to  $V_8$  are the eight weight factors (proportional to the volumes of the blocks) used in the trilinear interpolation. The velocity at position  $p$  is thus determined using Eq. 24.

$$u(p) = \frac{1}{\sum_{k=1}^8 V_k} (V_1 u_7 + V_4 u_6 + V_5 u_3 + V_8 u_2 + V_2 u_8 + V_3 u_5 + V_6 u_4 + V_7 u_1) \quad (24)$$



**Fig. 9** Trilinear interpolation with partial volumes as weights to interpolate advection velocity for a LP at  $(p_x, p_y, p_z)$  using known velocities at the vertices

The acceleration of a particle in the compact Eulerian form is written in Eq. 25.

$$a = \frac{\partial u}{\partial t} + (u \cdot \nabla)u \quad (25)$$

The partial derivative terms of  $u_a$ ,  $\frac{\partial u_a}{\partial x}$ ,  $\frac{\partial u_a}{\partial y}$ , and  $\frac{\partial u_a}{\partial z}$  in Eq. 25 are estimated using Eq. 26 to Eq. 28 where  $a$  takes  $x$ ,  $y$  and  $z$  depending on the component axis.

$$-\frac{\partial u_a}{\partial x} = \frac{u_{a1,2}}{\delta x_i} + \frac{p'_y E_{1a}}{\delta x_i \delta y_j} + \frac{p'_z E_{2a}}{\delta x_i \delta z_k} + \frac{p'_y p'_z E_{0a}}{\delta x_i \delta y_j \delta z_k} \quad (26)$$

$$-\frac{\partial u_a}{\partial y} = \frac{u_{a1,4}}{\delta y_j} + \frac{p'_x E_{1a}}{\delta x_i \delta y_j} + \frac{p'_y E_{3a}}{\delta y_j \delta z_k} + \frac{p'_z p'_x E_{0a}}{\delta x_i \delta y_j \delta z_k} \quad (27)$$

$$-\frac{\partial u_a}{\partial z} = \frac{u_{a1,5}}{\delta z_k} + \frac{p'_x E_{2a}}{\delta x_i \delta z_k} + \frac{p'_y E_{3a}}{\delta y_j \delta z_k} + \frac{p'_y p'_x E_{0a}}{\delta x_i \delta y_j \delta z_k} \quad (28)$$

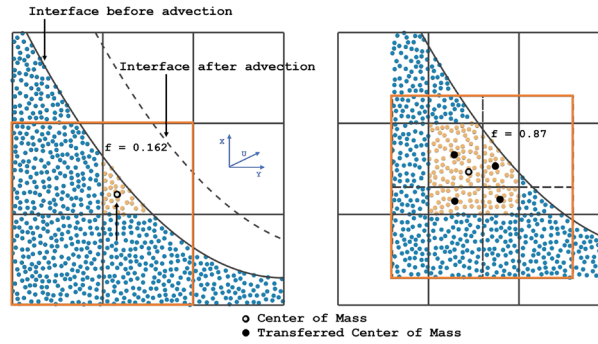
The terms  $E_{0a}$  to  $E_{3a}$  seen in the above equations are calculated using the set of Eqs. 29 given below.

$$\begin{aligned} E_{0a} &= (u_{a1} - u_{a2} + u_{a3} - u_{a4} \\ &\quad - u_{a5} + u_{a6} - u_{a7} + u_{a8}) \\ E_{1a} &= (-u_{a1} + u_{a2} - u_{a3} + u_{a4}) \\ E_{2a} &= (-u_{a1} + u_{a2} + u_{a5} - u_{a6}) \\ E_{3a} &= (-u_{a1} + u_{a4} + u_{a5} - u_{a8}) \\ u_{ab,c} &= u_{ab} - u_{ac} \\ p'_x &= p_x - x_i \\ p'_y &= p_y - y_j \\ p'_z &= p_z - z_k \end{aligned} \quad (29)$$

Finally the partial derivative of acceleration with respect to time (Eq. 25), is determined by Eq. 30,

$$\frac{\partial a}{\partial t_m} = \frac{u_m - u_{m-1}}{t_m - t_{m-1}} \quad (30)$$

Term  $\left(\frac{\partial a}{\partial t}\right)$  requires the velocities from the previous time step. Having obtained all terms described in the Eqs. 25, the approximate acceleration for all LPs are obtained. Thus, LPs are advected by either the Taylor's second order (Eq. 21) or the TJSM (Eq. 22) numerical advection schemes.



**Fig. 10** Mass transfer and CM Update: Left: LPs representing the fluid before advection.  $f$ - value and center of mass is shown for the surface cell(middle cell). Right: advected LPs from the middle and neighbor cells into the middle cell (orange color) and their transferred CMs. The updated  $f$ - value and CM is also shown. Note that cells are advected one by one and this figure is a combined representation of the process

## 2.5 Updating mass and center of mass

After advection the next step is The updating of  $f$  value and the CM. An new copy of the computational domain  $\Omega$  is created to receive the fluid mass advected, denoted by  $\Omega_m^*$ . This new domain receives the advected LPs from cell-by cell and new values for the fluid mass and CM keeps updating. ‘\*’ denotes that the functions and parameters belongs to  $\Omega^*$ . After advection of all the fluid in the cells, this updated domain copy,  $\Omega_m^*$  becomes the next time step value,  $\Omega_{m+1}$ . A typical interface before and after advection along with the LPs are shown on the left and right panels, respectively. Transferred mass and transferred CM from three neighbor cells into the surface cell in the middle are shown on right panel of Fig. 10, along with the final CM location and the  $f$  value (right panel).

If a cell space receives  $n$  LPs from a cell with mass per point  $m_p$  as indicated in Eq. (20), then the updated  $f$  value is estimated by Eq. (31) as

$$f_{up}^*(i, j) = f^*(i, j) + \frac{nm_p}{\rho \delta x_i \delta y_j} \quad (31)$$

Finally, Eq. 32 provides the updated CM, where ( $c_{tr}$ ) is the transferred CM.

$$c_{up}^* = \frac{c_e^* f^* \delta x_i \delta y_j \rho + c_{tr} n m_p}{n m_p + \delta x_i \delta y_j \rho} \quad (32)$$

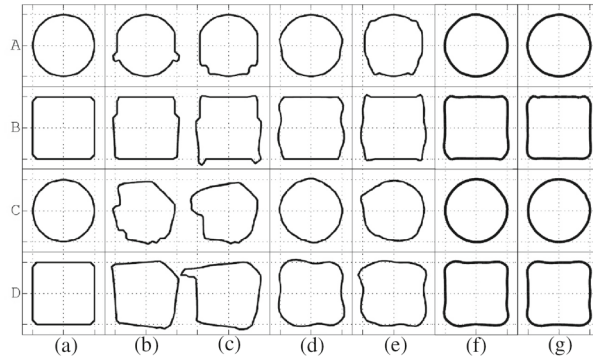
### Filters and optimization

When the velocity field is interpolated to advect LPs the divergence of the interpolated velocity field at certain location become non-zero. Hence, it can be said the  $f$  value larger than one at times is the result of the non-zero divergence resulting from the necessity to interpolate the discretized velocity field. As this is unavoidable, a filter is used for cells having  $f > 1$  and takes the excess and redistributes mass to non-empty neighboring cells uniformly. VCLP method proposes an optimization technique named skip core optimization (SCO) to increase computational efficiency. This technique allows us to skip the computation for a portion of cells which are far (three times CFL distance) from the interface. SCO can be understood as follows. Cells are classified before advection in to empty ( $E$ ), surface ( $S$ ), and interior cells ( $N_1, N_2, N_3$ ) based on  $q$  as explained in Sect. 2.1. This classification ensures cells always appear in the order  $E, S, N_1, N_2, N_3$  and during advection cell of one class can interact only with the neighboring ones as restricted by CFL number. if we denote  $A_{i,j}$  as mass of fluid advected from cells of class  $i$  to class  $j$ , then the Table 2 is obtained.

So, if we use SCO, some of  $N_3$  cells will have  $f > 1$  and the additional mass will be exactly  $A_{N_3 N_2}$ . Also, some of  $N_2$  cells will have  $f < 1$  and the deficiency would be exactly  $A_{N_3 N_2}$ . Now, because of CFL number based classification, an  $N_3$  cell can only become  $N_2$  and  $N_2$  can become either  $N_1$  or  $N_3$ . In all of these scenarios  $f$  will be equal to one and this shows that we can skip advection in  $N_3$  cells and by setting  $f = 1$  for  $N_3$  cells with  $f > 1$  and  $N_2$  with  $f < 1$  after advection and by skipping CM update in  $N_3$  and  $N_2$  cells. The application of filters is the final step of VCLP producing the final outputs the new  $f$  value matrix ( $f_{m+1}$ ) and the new CM matrices ( $c_{m+1}$ ).

**Table 2** Fluid mass before and after advection

Before	After advection
$E$	$E + A_{SE}$
$S$	$S + A_{N_1S} - A_{SE} - A_{SN_1}$
$N_1$	$N_1 + A_{SN_1} + A_{N_2N_1} - A_{N_1S} - A_{N_1N_2}$
$N_2$	$N_2 + A_{N_1N_2} + A_{N_3N_2} - A_{N_2N_1} - A_{N_2N_3}$
$N_3$	$N_3 + A_{N_2N_3} - A_{N_3N_2}$
Before	After Advection with SCO
$E$	$E + A_{SE}$
$S$	$S + A_{N_1S} - A_{SE} - A_{SN_1}$
$N_1$	$N_1 + A_{SN_1} + A_{N_2N_1} - A_{N_1S} - A_{N_1N_2}$
$N_2$	$N_2 + A_{N_1N_2} - A_{N_2N_1} - A_{N_2N_3}$
$N_3$	$N_3 + A_{N_2N_3}$



**Fig. 11** Translation of a circle (rows A and C) and a square (rows B and D) in two different velocity fields  $u = \hat{j}$  ( rows A and B) and  $u = 2\hat{i} + \hat{j}$  ( rows C and D). **a** Initial condition, **b** Hirt and Nichols without local height function, **c** Hirt and Nichols with local height function, **d** Young's without local height function, **e** Young's with local height function, **f** VCLP- with PLIC and **g** VCLP+ with PCIC Note: Results from column (b) to (e) are obtained from [29]

### 3 Benchmark tests

To evaluate the accuracy of VCLP method for interface reconstruction and advection, three benchmark tests in 2D and one in 3D from literature are conducted. The tests in 2D are translation, single vortex, and Zalesak's test while in 3D deformation test is done. In each of these tests, a well defined 2D fluid body inside a computational domain with specified grid resolution and dimensions is allowed to flow according to a specified solenoidal (non-divergent) velocity field. The flow is then simulated till the given time  $T$ , using the proposed method with the specified time intervals. The ideal final state of the fluid body at time  $T$  is known and the performance of the proposed method is evaluated on the basis of its qualitative and quantitative similarity with the simulation result. In the following sections, VCLP with PLIC is written as 'VCLP-' and with PCIC as 'VCLP+'. 'VCLP+' with Skip Core Optimization is given as 'VCLP+ SC' and 'VCLP+' while using Eq. 21 for advection it is shown as 'VCLP+ T2'. Advection using TJSM is denoted as 'TJ'.

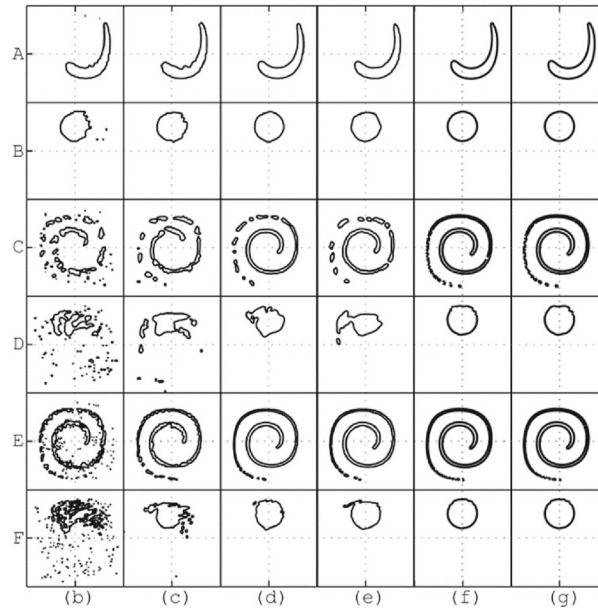
#### 3.1 Translation test

For translation tests, a circular and a square fluid body are placed inside two different velocity fields. The fluid bodies are transported in 1000 time steps over a distance approximately five times the diameter of the fluid. The number of grid cells along the diameter or the size of the fluid body is equal to 15. The test results are shown in Fig. 11.

#### 3.2 Single vortex

The single vortex test for evaluating the accuracy of a surface tracking and advection method was proposed by Rider and Kothe [31] and the solenoidal flow velocity field was suggested by Bell et al. [32]. For this test, a





**Fig. 12** Single vortex test result: Advection of a circular body of fluid placed in a single-vortex field. The maximum simulation time is equal to  $T = 8$  except for rows A and B, where  $T = 2$ . Snapshots are shown at time  $T/2$  (rows A, C, and E) and at time  $T$  (rows B, D, and F). The computational grid consists of  $60 \times 60$  cells (rows A, B, C and D) or  $100 \times 100$  cells (rows E and F). **a** Fluid body at time  $t = 0$  or initial state, **b** Hirt and Nichols results without local height function, **c** Hirt and Nichols with local height function, **d** Young's without local height function, **e** Young's with local height function, **f** VCLP with PLIC and **g** VCLP with PCIC. Note: Results from column (a) to (e) are obtained from [29]

circular fluid body of radius 0.15 m, centered at (0.5,0.75) is placed inside a time reversing solenoidal velocity field given by 33

$$\begin{aligned} u_x &= -\sin^2(\pi x) \sin(2\pi y) \cos\left(\frac{\pi t}{T}\right) \\ u_y &= \sin^2(\pi y) \sin(2\pi x) \cos\left(\frac{\pi t}{T}\right) \end{aligned} \quad (33)$$

where the  $T$  is the time period of the single vortex and  $t$  is the time. In an ideal case, the initial circular fluid body in a solenoidal velocity field gets stretched into a filament spiraling around the vortex center till time  $t = \frac{T}{2}$ , and gets back into the initial circular shape at  $t = T$ . Similarity of the final state to the initial state is the measure of accuracy for the surface tracking and advection method. Figure 12 depicts the single vortex test results of the VCLP methods along with that of H&N and Young's methods.

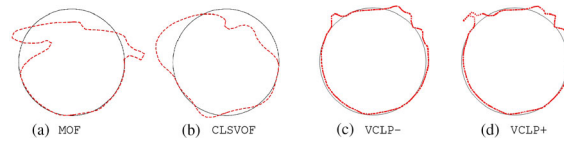
Single vortex tests are also conducted for CFL = 0.5 and  $T = 8$  s with two grid resolutions of  $64 \times 64$  and  $128 \times 128$  to compare VCLP results with two newer methods, moment of fluid (MOF) and coupled level set volume of fluid (CLSVOF). Figure 13 shows the test results at time  $t = T$  for the grid resolution of  $64 \times 64$  where the circle drawn in black color is the ideal case, whereas the dashed red lines are the test results. Single vortex test results a and b are obtained from [33].

#### Quantitative analysis of single vortex tests

Geometric error is used to quantify advection tests' accuracy by comparing the initial and final fluid distribution or the  $f$ -matrix. The  $f$ -matrix error is integrated over the computational domain, to account for the error in geometry of the fluid body. Geometric error in 2D is estimated using the Eq. 34,

$$E_{geo} = \sum_{i=1}^{i=n_x} \sum_{j=1}^{j=n_y} |f(i, j, T) - f(i, j, 0)| dx_i dy_j \quad (34)$$





**Fig. 13** Time reversed single vortex test results for grid resolution  $64 \times 64$  and  $CFL = 0.5$  for  $T = 8$  s. Results at  $t = T = 8$  s for **a** Moment of Fluid (MOF) and **b** Coupled Level Set Volume of Fluid (CLSVOF) **c** VCLP- and **d** VCLP+. Results **a** and **b** are obtained from [33]

**Table 3** Geometric errors  $E_{geo}$  for single vortex time reversible tests with time period  $T = 8$  s (lower panel)

$T = 8$ s	$32 \times 32$	$64 \times 64$	$128 \times 128$
H&N-	1.01E-1	7.96E-2	6.76E-2
H&N+	8.39E-2	6.05E-2	3.17E-2
Young's-	5.95E-2	2.00E-2	8.25E-3
Young's+	1.10E-1	4.49E-2	8.02E-3
Rider & Kothe	4.78E-2	6.96E-3	1.44E-3
PLIC [34]	6.5E-2	1.6E-2	2.5E-3
UB [34]	5.5E-2	2.5E-2	1.6E-2
EB [34]	5.5E-2	2.8E-2	1.2E-2
THINC [34]	1.5E-1	6.2E-2	2.5E-2
SW [34]	8.8E-2	5.7E-2	2.0E-2
SB [34]	1.0E-1	7.6E-2	4.1E-2
AR [34]	1.1E-1	8.5E-2	6.0E-2
VCLP - TJ	9.03E-3	2.10E-3	7.25E-4
VCLP - T2	8.14E-3	1.97E-3	6.90E-4
VCLP + TJ	9.49E-3	2.03E-3	7.93E-4
VCLP + T2	1.02E-2	2.24E-3	7.10E-4
VCLP + T2 SC	8.32E-3	2.43E-3	6.53E-4

H&N and Young's method with '+' sign denotes the application of local height function and '-' denotes without, while for VCLP, '+' means PCIC is used for interface reconstruction and '-' means PLIC is used. The Taylor second order advection scheme is denoted as 'T2' and TJSM as 'TJ'. 'SC' denotes the use of skip core optimization. Results for all tests except VCLP are obtained from literature [20,29]

**Table 4** Geometric errors  $E_{geo}$  for single vortex time reversible tests with a grid resolution of  $64 \times 64$  and  $CFL = 1$

Method	$T = 2$ s	$T = 4$ s	$T = 6$ s
1 <sup>st</sup> OU	5.96E-2	9.66E-2	1.14E-1
PPM	3.48E-3	1.26E-2	3.42E-2
Level Sets	4.42E-3	1.35E-2	2.45E-2
SLIC	1.70E-2	3.52E-2	7.05E-2
PLIC	7.00E-4	1.55E-3	4.03E-3
VCLP - TJ	1.39E-3	8.59E-4	1.40E-3
VCLP - T2	8.74E-4	6.47E-4	1.29E-3
VCLP + TJ	1.35E-3	8.63E-4	1.53E-3
VCLP + T2	9.11E-4	6.76E-4	1.42E-3
VCLP + T2 SC	9.16E-4	6.79E-4	1.36E-3

VCLP+ denotes VCLP with PCIC and VCLP- denotes VCLP with PLIC. 'T2' is Taylor second order advection and 'TJ' is the TJSM method while 'SC' is the Skip Core Optimization. Results for all tests except VCLP methods are taken from literature obtained from [31]. PPM is Piecewise Parabolic Method and 1<sup>st</sup> OU is 1<sup>st</sup> Order Upwind

where  $t = T$  is final time. Table 3 shows the single vortex test results of VCLP and five other surface tracking and advection methods. The tests are conducted for three grid resolutions:  $32 \times 32$ ,  $64 \times 64$  and  $128 \times 128$ .

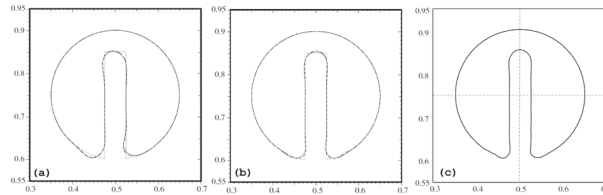
Single vortex tests for time periods  $T = 2$ , 4 and 6 are conducted for a grid resolution of  $64 \times 64$  and  $CFL = 1$  to compare with results from five other methods, obtained from [31]. The test results are presented in Table 4.

VCLP results are compared with newer methods such as MOF and CLSVOF in Table 5 which gives the geometric errors for the single vortex tests for the three grid resolutions,  $32 \times 32$ ,  $64 \times 64$  and  $128 \times 128$  with  $CFL = 0.5$ .

**Table 5** Geometric errors  $E_{geo}$  for single vortex time reversible tests with CFL = 0.5 and  $T = 8$  s

Method	$32 \times 32$	$64 \times 64$	$128 \times 128$
MOF	4.06E-2	1.61E-2	5.62 E-3
CLSVOF	6.46E-2	1.73E-2	3.45E-3
VCLP - T2	7.58E-3	2.93E-3	7.44E-4
VCLP + T2 SC	7.63E-3	2.79E-3	8.69E-4

VCLP+ denotes VCLP with PCIC and VCLP- denotes VCLP with PLIC. 'T2' is Taylor second order advection and 'TJ' is the TJSM method while 'SC' is the Skip Core Optimization. VCLP test results are compared against MOF & CLSVOF results obtained from [33]



**Fig. 14** Comparison of Zalesak's rigid body rotation test results for  $100 \times 100$  resolution obtained from [28] with VCLP result. **a** ELVIRA reconstruction (solid line) and the linear best fit (dashed line) **b** with the quadratic fit alone (solid line) and with the continuity algorithm (dashed line) **c** VCLP. Dotted line is the ideal result

The relative mass losses (RML) found out using Eq. 35,

$$RML = \frac{\sum A(i, j, 0) - \sum A(i, j, T)}{\sum A(i, j, 0)} \quad (35)$$

where  $A(i, j, T) = f(i, j, T)\delta_{x_i}\delta_{y_j}$  and  $\sum$  denotes summation of all cells during the single vortex tests using VCLP are of the order  $10^{-15}$ , with nearly perfect mass conservation. In contrast, the relative mass loss using CLSVOF method are  $1.42 \times 10^{-5}$  by Zhao et al. [35],  $4.0 \times 10^{-4}$  by Wang et al. [36], and  $1 \times 10^{-4}$  Menard et al. [37].

In conclusion, according to the single vortex test results, VCLP method have lesser geometric and relative position error along with negligible mass loss.

### 3.3 Zalesak's disk test

Zalesak's test problem (Zalesak et al. [38]) is a modified version of the rigid body rotation test where a circular fluid body a slotted disk (Zalesak's disk) is used. The slotted disk centered at (0.5, 0.75) have a radius  $r = 0.15$  m. The slot in the disk is defined such that it satisfies the relations,  $|x - 0.5| \leq 0.025$  and  $y \leq 0.85$ . The slotted fluid is placed inside the velocity field for rigid body rotation given by Eqs. 36.

$$\begin{aligned} u_x &= -2\pi(y - 0.5) \\ u_y &= 2\pi(x - 0.5) \end{aligned} \quad (36)$$

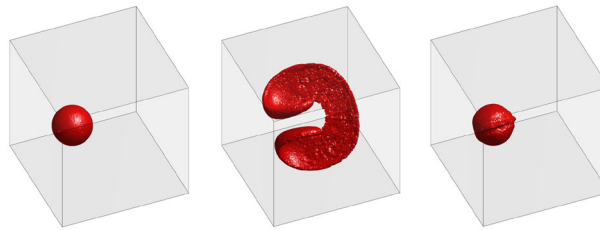
The slotted disk undergoes a rigid body rotation of  $2\pi$  radian (one complete circle) during 628 time-steps. Figure 14 compare the results for  $100 \times 100$  resolution with the work of Scardovelli et al. [28] using ELVIRA reconstruction method and its variations.

#### Quantitative analysis of Zalesak's disk tests

Table 6 gives the relative error of Zalesak's disk test results for the  $200 \times 200$  grid resolution. The relative errors of 10 other methods are presented from literature. VCLP methods have the least error, even lesser than purely Lagrangian methods. TJSM advection scheme have slightly lesser error than Taylor's second-order scheme. PCIC reconstruction performs better than PLIC with lesser error. So, overall VCLP methods performs very well in the Zalesak's disk test.

**Table 6** Relative error for Zalesak's disk rigid body rotation test for resolution of  $200 \times 200$ 

Reconstruction/advection methods	Rel. error
Higher-order bounded convection/Eulerian [39]	5.72E-3
Youngs/Stream (unsplit) [28]	1.07 E-2
Puckett/Stream /Eulerian [28]	1.00E-2
ELVIRA/Lagrangian (split) [28]	1.00E-2
Linear fit/Lagrangian [28]	9.42E-3
gVoFoam/Eulerian [40]	1.36E-2
interFoam/Eulerian [40]	6.61E-2
OpenFoam®/Eulerian [41]	7.03E-2
Quadratic fit/Lagrangian [28]	5.47E-3
Quadratic fit+continuity/Lagrangian [28]	4.16E-3
PLIC [34]	1.20E-2
THINC [34]	2.92E-2
EB [34]	3.44E-2
UB [34]	3.78E-2
SW [34]	5.15E-2
SB [34]	6.87E-2
AR [34]	1.89E-1
VCLP – T1/ Eulerian–Lagrangian	4.21E-3
VCLP – T2/ Eulerian–Lagrangian	4.36E-3
VCLP + T1/ Eulerian–Lagrangian	3.79E-3
VCLP + T2/ Eulerian–Lagrangian	3.96E-3
VCLP + T2 SC/ Eulerian–Lagrangian	4.05E-3

**Fig. 15** 3D Deformation test results of VCLP with  $T = 3$  s and  $CFL = 0.5$  for a grid resolution of  $64 \times 64 \times 64$ . Left panel: initial state at  $t = 0$ , middle panel: maximum deformed state at  $t = \frac{T}{2} = 1.5$  s, Right panel: final results for  $t = T = 3$  s

### 3.4 3D deformation test

In the 3D deformation test a spherical fluid body centered at  $(0.35, 0.35, 0.35)$  with a radius of 0.15 m is placed inside a time reversing solenoidal velocity field with vortices. Equations 37 gives the velocity field for the deformation test proposed by LeVeque [42].

$$\begin{aligned}
 u_x &= 2 \sin^2(\pi x) \sin(2\pi y) \sin(2\pi z) \cos(\pi t/T) \\
 u_y &= -\sin(2\pi x) \sin^2(\pi y) \sin(2\pi z) \cos(\pi t/T) \\
 u_z &= -\sin(2\pi x) \sin(2\pi y) \sin^2(\pi z) \cos(\pi t/T)
 \end{aligned}
 \tag{37}$$

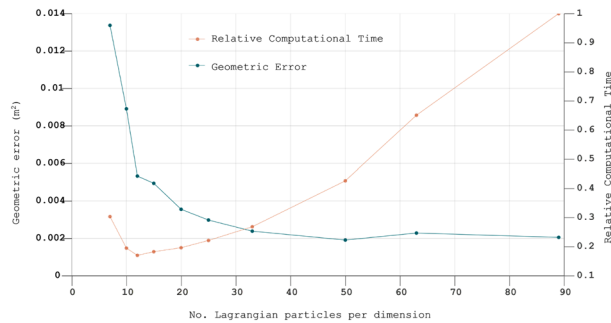
This test use 20 LPs per dimension for two grid resolutions,  $32^3$  and  $64^3$  cells. Figure 15 displays the result for resolution  $64^3$ . The spherical fluid body on the left panel is the initial state, the middle panel is the maximum stretched shape at  $t = T = 1.5$  s. The final result at  $t = T = 3$  s, is the right panel. Table 7 provides the geometric error comparison with Moment of fluid method (MOF) [43] and CLSVOF [44] obtained from [33].

### 3.5 Computational performance of VCLP

A parametric study was conducted for LPs per dimension distributed as blue noise to evaluate the optimum number of LPs. The single vortex test with resolution of  $64 \times 64$  and  $T = 8$  s was chosen for the study. Absolute errors and relative computational times were obtained as indicated in Fig. 16. The total computational time for time-steps is normalized with the case using 89 LPs.

**Table 7** Geometric errors  $E_{geo}$  and order of convergence for 3D deformation tests with CFL = 0.5 and  $T = 3$  s. MOF and CLSVOF test results are obtained from [33]

Method	$32 \times 32 \times 32$	Order	$64 \times 64 \times 64$
MOF	5.62E-3	1.26	2.35E-3
CLSVOF	7.77E-3	1.15	3.50E-3
PLIC [34]	7.8E-3	0.89	4.20E-3
EB [34]	9.0E-3	1.68	2.8E-3
VCLP-	4.40E-3	1.34	1.73E-3
VCLP+	4.35E-3	1.33	1.73E-3

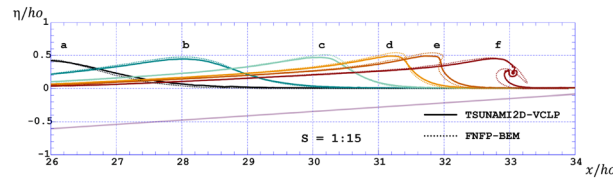
**Fig. 16** Geometric error and relative computational time for VCLP-PCIC with skip core optimization with respect to number of Lagrangian particles for the single Vortex test with  $T = 8$  s in a  $64 \times 64$  grid**Table 8** Computational time and error compared of VCLP method compared with original VOF method for Single Vortex  $T=8$ ,  $\delta x = \delta y = 1/60$  with the circle

Method	Comp. time (s)	Geo. error $m^2$
No advection	07.359	–
VOF	14.078	0.0851
VCLP T2 10	16.594	0.0396
VCLP T2 33	38.484	0.0169
VCLP TJ 10	19.875	0.0393
VCLP TJ 33	79.250	0.0168
VCLP T2 CM 10	16.765	0.0066
VCLP T2 CM 33	40.141	0.0042
VCLP TJ CM 10	20.438	0.0063
VCLP TJ CM 33	83.219	0.0045
VOF ( $120 \times 120$ )	105.50	0.07924

Last row have resolution have  $\delta x = \delta y = 1/120$

The following Table 8 compares the computational time and accuracy for the single vortex test. The resolution of the VCLP tests is  $60 \times 60$  and are conducted for 10 and 33 LPs per dimension. In the Table 8, non-advective means the advection is entirely skipped from calculation to account for the computational time all methods have in common.

The computational performance study of the VCLP method for the single 2D vortex test for resolution of  $60 \times 60$  shows that the VCLP method with 10 LPs, Taylor  $2^{nd}$  order advection, and CM tracking takes 19 % more time than SOLA-VOF, but the error reduced by 1289 %. The CM tracking seems to reduce the error from 2.68 to 6 times with less than five percent more of computational time. For a resolution of  $120 \times 120$ , SOLA-VOF takes 105.5 s with an error of 0.07924, which is 629 % more computational time and 1200 % times more error than VCLP with 10 LPs per dimension. This supports the claim that using VCLP at lower resolution obtains better results than SOLA-VOF with lesser computational time. Overall, from the computational performance analysis in 2D, the VCLP method shows considerable improvement with a slightly more computational time.



**Fig. 17** Comparison of the evolution of a solitary wave model using TSUNAMI2D-VCLP and FNFP-BEM model by Grilli et al. [46] with  $H_0/h_0 = 0.45$ , over a slope of 1:15. The solid lines are FNFP-BEM model results and dotted lines are TSUNAMI2D-VCLP

## 4 VCLP application

A validated CFD code, TSUNAMI2D by Horrillo [45] with the VCLP advection module is used to simulate the breaking wave problem from literature. Additional information on the TSUNAMI2D model can be found in [45].

### 4.1 Breaking wave problem

The breaking wave is a challenging transient multiphase problem to numerically simulate and solve for converging breaker jet free surfaces. Grilli et al. [46] have numerically modeled shoaling and breaking of 2D solitary waves generated by a moving lateral boundary and traveling over a plane slope. This experiment is replicated using TSUNAMI2D-VCLP, where a solitary wave of height  $H_0 = 0.45$  m in a 1 m deep wave tank shoals and breaks over a slope of 1:15.

The movement of the piston wavemaker to generate the solitary waves is defined by Eq. 38,

$$x_p(t) = \frac{H_0}{\kappa} \left[ \tanh \frac{\kappa}{h_0} (ct - x_p(t) - \lambda + \tanh \frac{\kappa \lambda}{h_0}) \right] \quad (38)$$

where  $\kappa = \frac{1}{2} \sqrt{3H_0'}$ . Here  $H_0$  is the solitary wave height in constant depth  $h_0$ , and  $c$  is the celerity of the solitary wave. The motion of the piston wavemaker corresponds to a wave profile that is cut at  $\lambda$  distance on both sides of the wave crest. More detailed information about the piston wavemaker can be found in [46]. The TSUNAMI2D-VCLP results till the breaking point is compared with the FNFP-BEM results by Grilli et al. [46] obtained from [47] and are given in Fig. 17. The dotted line is FNFP-BEM, while the solid line represents the TSUNAMI2D-VCLP results. The solitary wave at locations 'a' to 'f' matches well in terms of height position and shape.

## 5 Conclusion

A surface tracking and advection method for incompressible and immiscible fluid flows, named VCLP is proposed. Mathematical description of VCLP method is provided in detail and the accuracy of the VCLP method for interface reconstruction and advection is evaluated based on benchmark tests such as translation, single vortex, and Zalesak's disk test in 2D and deformation test in 3D. The benchmark test results show that VCLP performs qualitatively well compared to conventional methods such as Hirt and Nichols (H&N), Young's, and newer methods such as the moment of fluid (MoF) and Coupled Level Set Volume of Fluid methods (CLSVOF). Later, the quantitative error comparisons support this observation. From the 2D single vortex field test results, VCLP has lesser geometric error than H&N, Young's, 1<sup>st</sup> order upwind, PPM, Level sets, SLIC, and PLIC. Quantitative results of the Zalesak's disk test and deformation tests using VCLP also show less or similar error than the other presented methods. From the Zalesak's test result for grid resolution of  $200 \times 200$ , VCLP errors are smaller than successful methods from literature such as Youngs, Puckett, ELVIRA, gVoFoam, interFoam, OpenFoam, Quadratic fit, and Quadratic fit with continuity. For deformation and Zalesak's disk tests, newly proposed TJSM advection scheme performs better than Taylor's second order advection scheme. Also, VCLP-PCIC have lesser error than VCLP-PLIC for Zalesak's disk tests. This suggests that the proposed advection scheme TJSM and the new reconstruction method, PCIC are successful.

The challenging breaking wave problem was simulated using TSUNAMI2D-VCLP and results were compared with the FNFP-BEM and the results are very comparable. Thus, the applications of the VCLP method in the TSUNAMI2D solver shows overall satisfactory results.

**Data availability statement** The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

## Appendix A Comparison of T-slope with other methods to find the interface normal

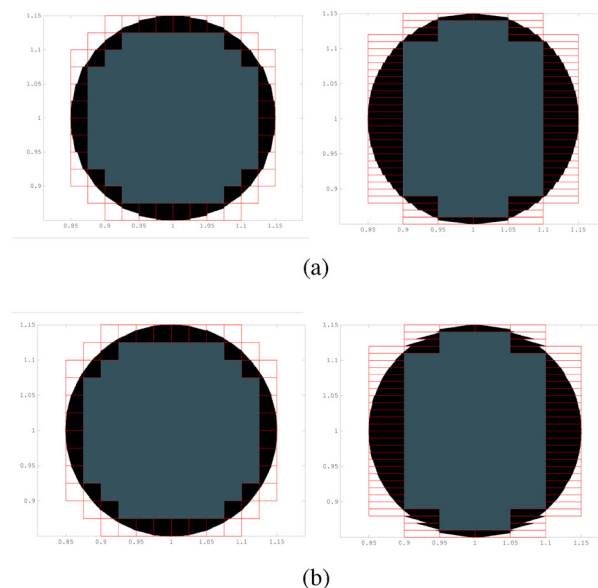
A two-dimensional circular fluid domain is reconstructed from the actual  $f$  values, i.e., from the intersection area of the circle and cell boundaries using PLIC with the three methods, numerical gradient, minimum slope, and T-slope. The circular domain is centered at (1,1) with a radius of 0.15 units. The grid resolution for two cases are (1)  $\delta x = 0.025$ ,  $\delta y = 0.025$ , with aspect ratio 1:1, and (2)  $\delta x = 0.05$  and  $\delta y = 0.01$  with aspect ratio 5:1. The qualitative results are shown in Figs. 18 and 19. The numerical gradient method interfaces are not smooth and have significant discontinuities in both cases with different aspect ratios. The minimum slope method works well with 1:1 aspect ratio, however for the larger or smaller aspect ratios it is biased and have larger errors. It can be seen that the estimated slope near the top and bottom parts of the circular fluid are more horizontal than the true slope for the aspect ratio of 5:1. The introduced T-slope method seems to resolve this issue, producing good result for both scenarios as can be seen from Fig. 19.

## Appendix B Sorted surface constant method

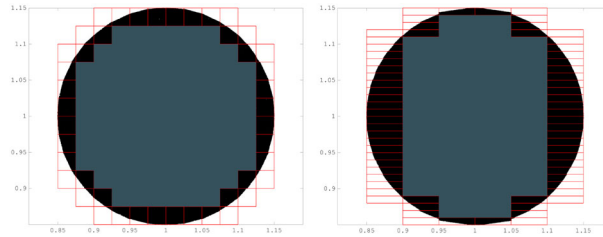
The algorithm to apply SSC method for the PLIC in 2D and 3D follows.

1. Fill the surface cell uniformly with  $N$  points with a blue noise distribution.
2. Calculate planar constants for all  $N$  points by simply substituting coordinates of each points in Eq. (B1).
3. Sort the planar constants in ascending order and find  $fN^{th}$  planar constant.

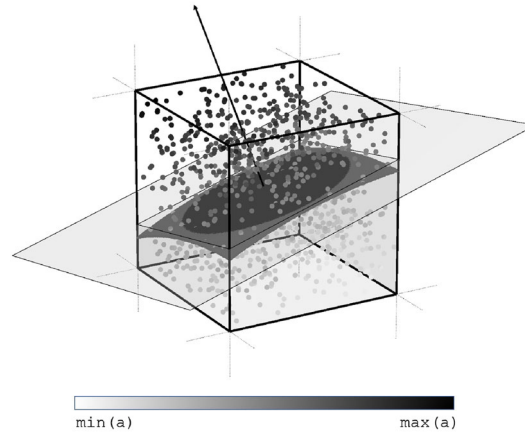
If  $fN$  is not an integer, find weighted average of  $\lfloor fN \rfloor^{th}$  and  $\{\lfloor fN \rfloor + 1\}^{th}$  planar constants. Here ‘ $\lfloor \cdot \rfloor$ ’ is the floor function which gives the largest integer smaller than the next integer.



**Fig. 18** Circular fluid domain reconstructed from the  $f$  values using PLIC with interface normal calculated using. **a** Normal Gradient Method. **b** Minimum Slope Method



**Fig. 19** Circular fluid domain reconstructed from the  $f$  values using PLIC with interface normal calculated using the T-Slope Method



**Fig. 20** Piecewise Circular Interface Calculation (PCIC) in 3D. The rectangular plane is the PLIC interface, and the spherical surface has some portion above and below the plane to have the same  $f$ -value. For each point in the cell, ‘a’ is the surface-constant, signed distance from the centroid of  $L$  to the intersection of normal through centroid and of a sphere of signed radius  $R$  (mean radius of curvature) centered at that point

$$a = -(n_x x_i + n_y y_j + n_z z_k) \tag{B1}$$

Sorted surface constant method for PCIC

The first step is to fill the cell with random points of blue noise distribution. Then all points are mapped to the PLIC normal line passing through the PLIC mid point. For a given point, (example: 1 or 2 in Fig. 5) the mapped point on the normal line is located at distance  $R$  from it. The mapped points are then sorted based on the surface-constant ‘a’ which is the signed distance from the midpoint to the mapped point to locate the interface.

Now while mapping, there are two possible location along the normal. If  $R$  is negative, the point with minimum surface constant value is chosen else the maximum. After sorting, we find the  $f N^{th}$  (surface constant), which gives the sphere’s center. Thus, the final equation of  $C$  (Eq. 3) is achieved. There is an alternative approach for the SSC, which is SSC\*. The steps for SSC\* is the following.

1.  $N$  uniformly distributed random points are created inside a unit square with blue noise distribution but sorted in positive y direction. Random points need to fill a larger volume or area than the cell. For example: a square or cube with a side length equal to the length of the cell’s largest diagonal.
2. Rotate and orient points such that sorted direction align with the normal direction.
3. Remove points outside the cell ending up with  $M$  points.
4. Find  $(fM)^{th}$  point’s planar constant.

If  $fM$  is not an integer, find weighted average of  $\lfloor fM \rfloor^{th}$  and  $\{\lfloor fM \rfloor + 1\}^{th}$  planar constants.

Figure 20 shows PCIC reconstruction with spherical surface. Parts of PCIC surface would lie on both sides of equivalent  $L$  as both have same  $f$  value in comparison with Fig. 4.



The computational speed of SSC is depended on the speed of sorting the array (surface constant ‘a’). This is a one step process. It does not require to solve any equations as in the analytical case, which usually gets complicated in 3D and for unstructured and curvilinear systems. SSC is flexible because speed vs. accuracy is dependent only on number of points  $N$ , which can be customized. While using large number of points, SSC\* may be more efficient than SSC as sorting is required in the latter.

### Appendix C Blue noise template

Blue noise

Creation of blue noise template with  $N$  LPs along each dimension using the proposed method is as follows.

- Estimate the radius  $r$ , where  $r = \frac{1}{\sqrt{2}N}$  in 2D and  $r = \frac{\sqrt{3}}{2N}$  in 3D.
- Divide the cell equally with  $\delta x = \delta y = \frac{1}{N}$  creating  $N^2$  square cell division in 2D and into  $N^3$  cell divisions in 3D with  $\delta x = \delta y = \delta z = \frac{1}{N}$
- Generate 2-tuple of  $N^2$  uniformly distributed random numbers multiplied by  $\delta x$  in 2D and 3-tuple of  $N^3$  uniformly distributed random numbers multiplied by  $\delta x$  and place one LP inside each cell division using the random numbers.
- Check through all cell divisions if shortest distance between the LPs inside the neighbor cell is less than  $r$ , in which case delete the LP in the cell division (not the LP in neighbor cell).
- Consider the cell division from which LP was deleted in the previous step. Fill those cell divisions with 30 to 100 random particle in 2D and 200 to 1000 in 3D and remove newly generated LPs that are within distance  $r$  from the neighbors. Select just a single LP from the remaining set of newly generated LPs that makes sure every cell division have one LP and shortest distance between any two LPs are larger than  $r$ .

### Appendix D Derivation of TJSM advection scheme and its comparison with other popular schemes in 2D and 3D

Derivation of TJSM

In a 3D domain, the position, velocity and acceleration of the particle at time step  $t_m$  within  $\Delta t_m$  are  $p_m$ ,  $u_m$ , and  $a_m$ . This is written as Eq. D2.

$$\begin{aligned} p_m &= x(t_m)\hat{i} + y(t_m)\hat{j} + z(t_m)\hat{k} \\ u_m &= u_x(p_m, t_m)\hat{i} + u_y(p_m, t_m)\hat{j} + u_z(p_m, t_m)\hat{k} \\ a_m &= a_x(p_m, t_m)\hat{i} + a_y(p_m, t_m)\hat{j} + a_z(p_m, t_m)\hat{k} \end{aligned} \quad (D2)$$

Now impose a small time duration much smaller than the time step  $\Delta t_m$ ,  $\delta_t$  for which the particle is advected in a straight line. The position and velocity of the advected particle after  $\delta_t$  are given by  $p_{m'}$  and  $u_{m'}$  (Eq. D3). From the Taylor series expansion, their approximate values can be evaluated as shown in .

$$\begin{aligned} p_{m'} &= p_m + u_m \delta_t \\ u_{m'} &= u_m + a_m \delta_t \end{aligned} \quad (D3)$$

Now we assume that during  $\delta_t$  particle is undergoing a uniform circular planar motion which means then there would be a center of rotation ( $c_r$ ), a plane of rotation, a radius ( $r$ ), and an angular velocity ( $\omega$ ). The center of rotation should pass through the plane which is parallel to the planes that can carry both  $u_{m'}$  and  $u_m$  and passes through the point  $p_m$ . We name this as plane  $P1$ . The  $c_r$  would also pass through a plane with normal as  $u_m$  and passing through point  $p_m$ , plane  $P2$ . Finally  $c_r$  passes also through a plane with normal as  $u_{m'}$  and

through point  $p_{m'}$ , plane  $P3$ . The intersection of  $P1$ ,  $P2$ , and  $P3$  uniquely identify the center of rotation,  $c_r$ . The equations for the three planes are given by Eqs. D4–D6.

$$P1 : (u_m \times u_{m'}) \cdot (x - p_m) = 0 \quad (D4)$$

$$P2 : u_m \cdot (x - p_m) = 0 \quad (D5)$$

$$P3 : u_{m'} \cdot (x - p_{m'}) = 0 \quad (D6)$$

Substituting Eqs. D3 and D2 in Eq. D4, we get Eq. D7

$$\begin{aligned} P1 : (u_m \times (u_m + a_m \delta_t)) \cdot (x - p_m) &= 0 \\ (u_m \times a_m) \cdot (x - p_m) &= 0 \end{aligned} \quad (D7)$$

Substituting Eqs. D3 D5 and in Eqs. D4, D8 is obtained.

$$\begin{aligned} P3 : (u_m + a_m \delta_t) \cdot (x - (p_m + u_m \delta_t)) &= 0 \\ (a_m \cdot x) - (a_m \cdot p_m) - \|u_m\|^2 &= 0 \end{aligned} \quad (D8)$$

The term multiplied by  $\delta_t$  is neglected as it tends to zero. The three equations Eqs. D5, D7 and D8 where ‘ $\|\cdot\|$ ’ is the Euclidean norm is solved for  $c_r$ . After simplification using identities we get Eq. D9 for  $c_r$ .

$$c_{r,m} = p_m + \{(u_m \times a_m) \times u_m\} \frac{\|u_m\|^2}{\|(u_m \times a_m)\|^2} \quad (D9)$$

From Eqs. D9, D10 for radius vector is obtained as indicated below.

$$r_m = - \{(u_m \times a_m) \times u_m\} \frac{\|u_m\|^2}{\|(u_m \times a_m)\|^2} \quad (D10)$$

Once radius vector and velocity vector is known, angular velocity vector  $\omega$  can be estimated from D13, by using Eqs. D11 and D12 indicated below.

$$\omega_m = \frac{(r_m \times u_m)}{\|r_m\|^2} \quad (D11)$$

$$\omega_m = \frac{- [\{\beta \times u_m\} \times u_m] \frac{\|u_m\|^2}{\|\beta\|^2}}{\|\beta \times u_m\|^2 \frac{\|u_m\|^4}{\|\beta\|^4}} \quad (D12)$$

where  $\beta = u_m \times a_m$ . After simplification using identities, the angular velocity vector is obtained as

$$\omega_m = - \frac{\beta \|\beta\|^4}{\|u_m\|^2 \|\beta \times u_m\|^2} \quad (D13)$$

Now, the angular rotation  $\theta$  during the time step and normalized vector  $\omega$  is given by Eq. D14 and Eq. D15

$$\theta_m = \|\omega_m\| \Delta t_m = \frac{\Delta t_m \|\beta\|^5}{\|u_m\|^2 \|\beta \times u_m\|^2} \quad (D14)$$

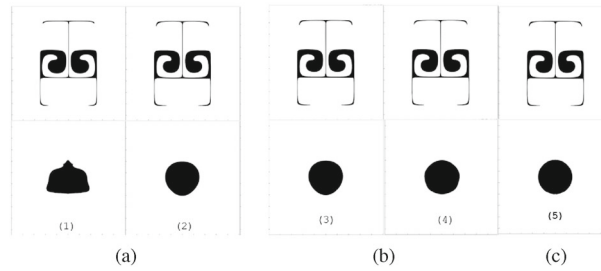
$$\hat{\omega}_m = - \frac{\beta}{\|\beta\|} \quad (D15)$$

Using Rodrigues’ formula given by Eq. D16 [48], after simplification using vector identities, we get the final advected location of the particle as given by Eq. D17.

$$\begin{aligned} p_{m+1} &= c_{r,m} + r_m \cos \theta_m + (\hat{\omega}_m \times r_m) \sin \theta_m \\ &\quad + \hat{\omega}_m (\hat{\omega}_m \cdot r_m) (1 - \cos \theta_m) \end{aligned} \quad (D16)$$

$$p_{m+1} = p_m + \frac{\{(\beta \times u_m) (1 - \cos \theta_m) - u_m \|\beta\| \sin \theta_m\} \|u_m\|^2}{\|\beta\|^2} \quad (D17)$$

If  $u_m = 0$ , or  $u_m \times a_m = 0$ , then Taylor’s second order advection method is to be used to advect LPs.



**Fig. 21** Deformation test results  $t = 1 = T/2$  (Top),  $t = T = 2$  s (Bottom). **a** Taylor's first order (left) and Taylor's Second order (right). **b** Taylor's third order (left) and Runge-Kutta fourth order (right) order **c** TJSM second-order (error  $\Delta t^3$ )

### Comparison of TJSM advection scheme with other methods

The performance of the proposed numerical advection scheme, the TJSM method is tested using one of the deformation benchmark test for the interface tracking. In this test, a circular fluid body of radius 0.15 centered at (0.5,0.75) inside a square domain of unit size with its left bottom corner at the origin (0,0). The fluid body is inside a solenoidal velocity field that deforms the circular fluid body. Deformation test induces radical deformation and topology change of fluid bodies, which is more challenging than single vortex test [20]. There are sixteen vortices in the domain, and flow fields are time-reversed. Time reversed flow field means that for half of the simulation time, the fluid body undergoes deformation and then for the next half the velocity field is time-reversed causing the circular fluid to go back to its original shape and location. For the numerical case where time is discretized, final state of the fluid body is not exactly same as the starting state. Thus the similarity of the final state to the initial state is the measure of goodness for the advection method. The stream function for this velocity field is given by Rider et al. [31] as shown in Eq. (D18).

$$\Psi = \frac{1}{4\pi} \sin \left\{ 4\pi \left( x + \frac{1}{2} \right) \right\} \cos \left\{ 4\pi \left( y + \frac{1}{2} \right) \right\} \quad (\text{D18})$$

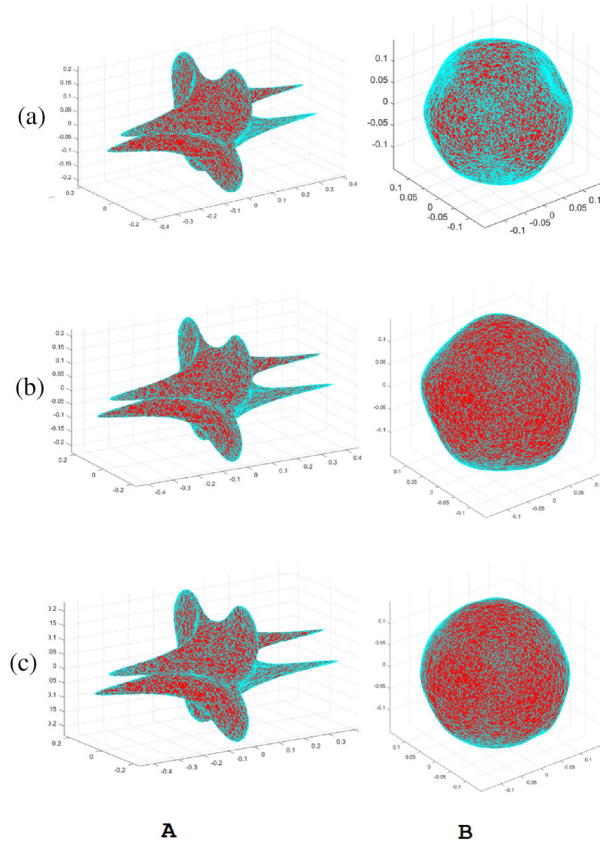
The velocity field is given by Eq. D19.

The time-varying field have its maximum magnitude of velocity at  $t = 0$  and  $t = T$  and zero at  $t = T/2$  where it reverses.

$$\begin{aligned} u_x &= \sin(4\pi y) \sin(4\pi x) \cos \left( \frac{\pi t}{T} \right) \\ u_y &= \cos(4\pi x) \cos(4\pi y) \cos \left( \frac{\pi t}{T} \right) \end{aligned} \quad (\text{D19})$$

The 2D test is conducted by putting marker particles on the circle perimeter, and each particle is advected in time using several numerical advection schemes. The numerical advection schemes tested are (1) Euler's Method, (2) Taylor  $2^{nd}$  order, (3) Taylor  $3^{rd}$  order, and (4) Runge–Kutta  $4^{th}$  order, and (5) TJSM  $2^{nd}$  order. The number of time steps is 256 for a simulation time of  $T = 2$  s. The deformation test results are depicted in the Fig. 21 where marker particles bound the fluid area to improve the contrast.

The 2D deformation test results for the numerical advection schemes show that at  $t = T/2$  difference between methods are not easily noticeable except for the T-shaped tip of the top part in the Taylor's first order method, which is shorter than other methods (see top panels of Fig. 21). At  $t = T$ , the final shape is ideally the initial circle itself, however all methods show some deviations from it. Taylor's first order have large deviation from circular shape, while other cases are closer to the initial circle. Taylor's second and third order have an elliptical shape at the bottom and the third order have slightly closer shape to the initial circle than the second order. The Runge-Kutta fourth order have an overall circular and symmetric shape, however edges are flatter over the bottom half of fluid body and is similar to a polygon like shape. The TJSM result looks very close to the initial circle and very slight deformation can be noticed at the top and bottom part. While comparing all methods qualitatively, the closeness to the circular shape is in the order (1) TJSM second order, (2) Runge-Kutta fourth order, (3) Taylor's third order, (4) Taylor's second order, and (5) Taylor's first order. So, as per the test results, the proposed advection method, TJSM performs well. The deformation test in 3D is carried out in the similar



**Fig. 22** Deformation test results for three numerical advection schemes a) Euler's method b) Taylor's second order c) TJSM second order. Column A:  $t = \frac{T}{2} = 6$  s, B:  $t = T = 12$  s

way as in 2D, where the spherical fluid body of radius 0.15 centered at (0.5,0.5,0.5) is placed inside a solenoidal time-reversing velocity field. The velocity field for this test is given by given by Eqs. D20–D22.

$$u_x = 2 \sin^2(\pi x) \sin(2\pi y) \sin(2\pi z) \cos\left(\frac{\pi t}{T}\right) \quad (\text{D20})$$

$$u_y = -\sin^2(\pi y) \sin(2\pi x) \sin(2\pi z) \left(\frac{\pi t}{T}\right) \quad (\text{D21})$$

$$u_z = -\sin^2(\pi z) \sin(2\pi x) \sin(2\pi y) \left(\frac{\pi t}{T}\right) \quad (\text{D22})$$

The simulation time of  $T = 12$  s is completed in 32 time steps. Figure 22 shows the deformation test results of the fluid body. Qualitatively, the resemblance of the final results to the spherical shape are in the order (1) TJSM second order, (2) Taylor's second order, and (3) Taylor's first order. So, as per the test results in 3D, the proposed numerical advection scheme, TJSM seems to perform well in comparison with the two standard methods.

## References

1. Harlow, F.H.: Mac numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluid* **8**, 12 (1965)
2. Peskin, C.S.: Numerical analysis of blood flow in the heart. *J. Comput. Phys.* **25**(3), 220–252 (1977)
3. Tryggvason, G., Bunner, B., Esmaeeli, A., Juric, D., Al-Rawahi, N., Tauber, W., Han, J., Nas, S., Jan, Y.-J.: A front-tracking method for the computations of multiphase flow. *J. Comput. Phys.* **169**(2), 708–759 (2001)

4. Hirt, C., Amsden, A., Cook, J.: An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *J. Comput. Phys.* **135**(2), 203–216 (1997)
5. Baiges, J., Codina, R., Pont, A., Castillo, E.: An adaptive fixed-mesh ale method for free surface flows. *Comput. Methods Appl. Mech. Eng.* **313**, 159–188 (2017)
6. Nithiarasu, P.: An arbitrary Lagrangian Eulerian (ALE) formulation for free surface flows using the characteristic-based split (CBS) scheme. *Int. J. Numer. Meth. Fluids* **48**(12), 1415–1428 (2005)
7. Marchandise, E., Remacle, J.-F., Chevaugeon, N.: A quadrature-free discontinuous Galerkin method for the level set equation. *J. Comput. Phys.* **212**(1), 338–357 (2006)
8. Harlow, F.H.: The particle-in-cell computing method for fluid dynamics. *Methods Comput. Phys.* **3**, 319–343 (1964)
9. Snider, D.: An incompressible three-dimensional multiphase particle-in-cell model for dense particle flows. *J. Comput. Phys.* **170**(2), 523–549 (2001)
10. Schroeder, C., Zheng, W., Fedkiw, R.: Semi-implicit surface tension formulation with a Lagrangian surface mesh on an Eulerian simulation grid. *J. Comput. Phys.* **231**(4), 2092–2115 (2012)
11. Wang, L., Khayyer, A., Gotoh, H., Jiang, Q., Zhang, C.: Enhancement of pressure calculation in projection-based particle methods by incorporation of background mesh scheme. *Appl. Ocean Res.* **86**, 320–339 (2019)
12. Dyadechko, V., Shashkov, M.: Reconstruction of multi-material interfaces from moment data. *J. Comput. Phys.* **227**(11), 5361–5384 (2008)
13. Wenzel, E., Garrick, S.: A point-mass particle method for the simulation of immiscible multiphase flows on an Eulerian grid. *J. Comput. Phys.* **397**, 108835 (2019)
14. Kothe, D.B.: Perspective on eulerian finite volume methods for incompressible interfacial flows. In: *Free Surface Flows*, pp. 267–331. Springer (1998)
15. Kothe, D., Rider, W., Mosso, S., Brock, J., Hochstein, J.: Volume tracking of interfaces having surface tension in two and three dimensions. In: *34th Aerospace Sciences Meeting and Exhibit*, p. 859 (1996)
16. Rudman, M.: A volume-tracking method for incompressible multifluid flows with large density variations. *Int. J. Numer. Meth. Fluids* **28**(2), 357–378 (1998)
17. Scardovelli, R., Zaleski, S.: Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech.* **31**(1), 567–603 (1999)
18. Aulisa, E., Manservigi, S., Scardovelli, R., Zaleski, S.: A geometrical area-preserving volume-of-fluid advection method. *J. Comput. Phys.* **192**(1), 355–364 (2003)
19. Hirt, C.W., Nichols, B.D.: Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* **39**(1), 201–225 (1981)
20. Rider, W.J., Kothe, D.B.: Reconstructing volume tracking. *J. Comput. Phys.* **141**(2), 112–152 (1998)
21. Nichols, B., Hirt, C., Hotchkiss, R.: Sola-vof: A solution algorithm for transient fluid flow with multiple free boundaries. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (1980)
22. Torrey, M.D., Cloutman, L.D., Mjolsness, R.C., Hirt, C.: Nasa-vof2d: a computer program for incompressible flows with free surfaces. NASA STI/Recon Technical Report N **86** (1985)
23. Elliot, S.: Fluid flow aspects of solidification modeling: Simulation of low pressure die casting. PhD thesis, Ph. D. Thesis, Department of Mining & Metallurgical Engineering, University (1999)
24. Harvie, D.J., Fletcher, D.F.: A new volume of fluid advection algorithm: the defined donating region scheme. *Int. J. Numer. Meth. Fluids* **35**(2), 151–172 (2001)
25. Harvie, D.J., Fletcher, D.F.: A new volume of fluid advection algorithm: the stream scheme. *J. Comput. Phys.* **162**(1), 1–32 (2000)
26. Gueyffier, D., Li, J., Nadim, A., Scardovelli, R., Zaleski, S.: Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *J. Comput. Phys.* **152**(2), 423–456 (1999)
27. Scardovelli, R., Zaleski, S.: Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *J. Comput. Phys.* **164**(1), 228–237 (2000)
28. Scardovelli, R., Zaleski, S.: Interface reconstruction with least-square fit and split Eulerian–Lagrangian advection. *Int. J. Numer. Meth. Fluids* **41**(3), 251–274 (2003)
29. Gerrits, J., Veldman, A.: Dynamics of liquid-filled spacecraft. *J. Eng. Math.* **45**(1), 21–38 (2003)
30. Cook, R.L.: Stochastic sampling in computer graphics. *ACM Trans. Gr. (TOG)* **5**(1), 51–72 (1986)
31. Rider, W., Kothe, D.: Stretching and tearing interface tracking methods. In: *12th Computational Fluid Dynamics Conference*, p. 1717 (1995)
32. Bell, J.B., Colella, P., Glaz, H.M.: A second-order projection method for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **85**(2), 257–283 (1989)
33. Mukundan, A.A., Ménard, T., de Motta, J.C.B., Berlemont, A.: A 3d moment of fluid method for simulating complex turbulent multiphase flows. *Comput. Fluids* **198**, 104364 (2020)
34. Pirozzoli, S., Di Giorgio, S., Iafrati, A.: On algebraic TVD–VOF methods for tracking material interfaces. *Comput. Fluids* **189**, 73–81 (2019)
35. Zhao, Y., Chen, H.-C.: Violent free surface flow simulations by a coupled level-set and volume-of-fluid method in overset grid systems. *Int. J. Offshore Polar Eng.* **24**(02), 114–121 (2014)
36. Wang, Z., Yang, J., Stern, F.: A new volume-of-fluid method with a constructed distance function on general structured grids. *J. Comput. Phys.* **231**(9), 3703–3722 (2012)
37. Ménard, T., Tanguy, S., Berlemont, A.: Coupling level set/vof/ghost fluid methods: Validation and application to 3d simulation of the primary break-up of a liquid jet. *Int. J. Multiph. Flow* **33**(5), 510–524 (2007)
38. Zaleski, S.T.: Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.* **31**(3), 335–362 (1979)
39. Garoosi, F., Mahdi, T.-F.: Presenting a novel higher-order bounded convection scheme for simulation of multiphase flows and convection heat transfer. *Int. J. Heat Mass Transf.* **172**, 121163 (2021)
40. Cifani, P., Michalek, W., Priems, G., Kuerten, J.G., van der Geld, C., Geurts, B.J.: A comparison between the surface compression method and an interface reconstruction method for the vof approach. *Comput. Fluids* **136**, 421–435 (2016)

41. Li, M.-J.: Interaction between free surface flow and moving bodies with a dynamic mesh and interface geometric reconstruction approach. *Comput. Math. Appl.* **81**, 649–663 (2021)
42. LeVeque, R.J.: High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.* **33**(2), 627–665 (1996)
43. Dyadechko, V., Shashkov, M.: Moment-of-fluid interface reconstruction. Los Alamos Report LA-UR-05-7571 (2005)
44. Sussman, M., Puckett, E.G.: A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *J. Comput. Phys.* **162**(2), 301–337 (2000)
45. Horrillo, J.J.: Numerical method for tsunami calculation using full Navier–Stokes equations and the volume of fluid method (2006)
46. Grilli, S.T., Subramanya, R.: Numerical modeling of wave breaking induced by fixed or moving boundaries. *Comput. Mech.* **17**(6), 374–391 (1996)
47. Lachaume, C., Biaisser, B., Fraunié, P., Grilli, S.T., Guignard, S.: Modeling of breaking and post-breaking waves on slopes by coupling of bem and vof methods. In: *The Thirteenth International Offshore and Polar Engineering Conference*. OnePetro (2003)
48. Mebius, J.E.: Derivation of the euler-rodrigues formula for three-dimensional rotations from the general formula for four-dimensional rotations. [arXiv:math/0701759](https://arxiv.org/abs/math/0701759) (2007)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.