A three-dimensional cell-based Volume-of-Fluid method for conservative simulations of primary atomization

Fabian Fröde[a,*], Temistocle Grenga[a], Vincent Le Chenadec[b], Mathis Bode[a,c],
Heinz Pitsch[a]

[a] *Institute for Combustion Technology, RWTH Aachen University, 52056 Aachen, Germany*
[b] *Multiscale Modeling and Simulation Laboratory, Gustave Eiffel University, Champs-sur-Marne, France*
[c] *Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany*

Abstract

This paper presents an improvement of the three-dimensional unsplit cell-based advection scheme of Le Chenadec and Pitsch [A 3D Unsplit Forward/Backward Volume-of-Fluid Approach and Coupling to the Level Set Method, J. Comput. Phys. 233 (2013) 10-33], which was shown to be prone to conservation errors. The improved method ensures discrete conservation by introducing a correction of the projected cells, which is similar to the three-dimensional cell-based method of Comminal and Spangenberg [Three-dimensional cellwise conservative unsplit geometric VOF schemes, J. Comput. Phys. 442 (2021) 110479], developed in parallel to this work. While the cell projection is performed as in the method of Le Chenadec and Pitsch, additional vertices are introduced to correct the projected cells. The positions of those are obtained from conservative flux volumes, which was inspired from the work of Owkes and Desjardins [A computational framework for conservative, three-dimensional, unsplit, geometric transport with application to the volume-of-fluid (VOF) method, J. Comput. Phys. 270 (2014) 587-612]. The improved method is tested in two- and three-dimensional benchmark cases and compared to literature methods. Moreover, the improved method is applied in the simulation of a realistic coaxial liquid-gas primary atomization case, which suffered from large conservation errors when

---

*Corresponding author
Email address: `f.froede@itv.rwth-aachen.de` (Fabian Fröde)

the method of Le Chenadec and Pitsch was applied.

Keywords: Volume-of-Fluid, Unsplit interface advection, Mass conservation, Primary atomization

1. Introduction

Atomization of liquids is a well-known example of a complex flow with gas-liquid interface with high relevance for engineering applications. In particular, the poorly understood primary atomization is known to play a key role in applications as it often impacts all downstream processes. Due to very small length and time scales, experimental investigations of primary atomization are typically difficult. Thus, accurate and reliable numerical simulations are very helpful to understand the complex processes involved in primary atomization. However, there are several challenges in numerical simulations of primary atomization with the accurate description of the interface evolution being one of them [1]. Numerical methods for describing the interface evolution can be classified into interface tracking and interface capturing methods. Interface tracking methods use an explicit representation of the interface location either by an interface-aligned mesh that moves with the interface or Lagrangian marker particles [2]. Conversely, interface capturing methods represent the interface location implicitly by mapping the interface to one iso-surface of an auxiliary field that is transported by the flow. The most common interface capturing methods are the Level Set (LS) [3, 4] and Volume-of-Fluid (VOF) methods [5].

In LS methods, the interface is represented by an iso-surface of a smooth function, which is typically defined as the signed distance to the interface. LS methods benefit from the smoothness of the function so that well-established high-order advection schemes, e.g. ENO schemes [6], can be used for advancing the interface location. This simplicity has its drawbacks including the fact that LS methods suffer from mass conservation errors. Many improvements have been proposed [7, 8, 9, 10, 11]. However, discrete conservation has not been achieved, and typically the improvements drastically reduce the simplicity.

In contrast, the VOF method represents the interface using an indicator function. While this approach provides the potential of mass conservation, second-order convergence, numerical consistency, numerical stability, and robust treatment of topology changes [12], the discontinuous nature of the indicator function requires special care during interface advection. Numerical advection schemes for advancing the discontinuous interface representation can be categorized into geometrical [13, 5, 14, 15] and algebraic schemes [16, 17]. While algebraic schemes rely on continuum-based discretizations of the indicator function advection equation, geometrical schemes rely on a subcell interface approximation and cell-faces, cells, or phase-specific material volumes that are followed along Lagrangian trajectories.

Geometrical advection schemes can be further categorized in split and unsplit schemes. Following the operator-splitting approach, split schemes separate the multidimensional advection step into a sequence of one-dimensional advection steps. Even though conservative schemes have been formulated for two and three dimensions [18, 19, 20, 21], the advection accuracy suffers from the splitting-approach [2].

While the unsplit approach allows to exploit the full potential of the VOF method, the drawback of those schemes is the need for complex geometrical operations, which results in high computational costs. As a result, Marić et al. [12] pointed out in their comprehensive review on advection schemes that the efficiency of unsplit geometrical schemes needs to be addressed by future work. The need for efficient algorithms is particularly emphasized in the context of primary atomization simulations, which are typically large-scale simulations due to the multiscale nature of primary atomization. Two approaches exist for advancing the interface in the unsplit approach. The cell-based approach traces cells or phase-specific volumes either forward or backward along Lagrangian trajectories and these volumes are subsequently remapped on an Eulerian mesh. In contrast, a flux volume is estimated for each computational cell face in the flux-based approach, and the phase-specific volume inside the flux volume is computed by geometrical operations. Many three-dimensional (3D) unsplit flux-based advection schemes have been proposed, which mainly vary in their construction of the flux volume. Liovic et al. [22] used face velocities to construct the flux volume. While the flux volume magnitude ensures discrete conservation, the flux volumes can intersect each other and form overlaps and gaps with neighboring flux volumes. As a consequence, unphysical over- and undershoots are formed that either need to be clipped (non-conservative) or conservatively redistributed. Hernández et al. [23] proposed the use of unique edge velocities, which significantly reduced over- and undershoots. As the edge velocities are obtained by interpolation, an error is introduced resulting in flux volumes that do not ensure discrete conservation. Marić et al. [24] used unique vertex velocities to construct the flux volume and introduced a triangulation to account for the possible complex shapes of the flux volume. While this avoids over- and undershoots, the flux volume requires correction. For this, an iterative correction was proposed that could again create overlaps. The first conservative and bounded 3D unsplit flux-based advection schemes were simultaneously proposed by Owkes and Desjardins (OD) [25] and Jofre et al. [26]. Both schemes rely on unique vertex velocities, a triangulation of the complex flux volume, and an analytical flux volume correction that avoids overlaps and gaps.

Although the cell-based approach provides a conceptually simple and efficient framework for advancing the interface location, only two 3D schemes have been proposed. In the 3D cell-based scheme of Le Chenadec and Pitsch (LP) [27], computational cells are traced forward and/or backward in time using unique vertex velocities, and the phase-specific volume is remapped from the projected cell on a static Eulerian target mesh. Even though the cell projection and triangulation ensure boundedness, the cell volume is only exactly maintained in linear velocity fields resulting in conservation errors for non-linear velocity fields. Although the conservation error was found to be small in the presented cases with non-linear velocity fields, the method might have strong conservation errors in cases where the velocity field is strongly non-linear, for example, in boundary layers. Recently, in parallel to the present work, Comminal and Spangenberg (CS) [28] proposed a forward and backward 3D conservative cell-based scheme. Similar to the method of LP, the cell is projected using unique vertex velocities and a remapping on a static Eulerian mesh is performed. To enforce conservation, CS introduced a correction of the cell projection based on conservative flux volumes. Furthermore, the use of geometric operations for non-convex polyhedra, as proposed by López et al. [29], removes the need for a triangulation of the cell projection. Similarly to most of the mentioned papers, CS restrict their evaluation and discussion of the scheme properties to benchmark test cases and do not present a realistic application.

In this paper, the method of LP, which has been successfully used for primary atomization simulations [30, 31, 32, 33], is improved by adding discrete conservation as significant conservation errors were found in simulations of a realistic coaxial air-blast atomizer. Similar to the method of CS, this is achieved by correcting the projected cell using an analytical correction that is based on conservative flux volumes. While sharing the same correction idea, the following differences are highlighted, which are mostly devoted to the context of primary atomization simulations. In contrast to the method of CS, the presented improvement relies and discusses cell projection schemes that are realizable with typical time advancement schemes of flow solvers, and a simplified flux volume approximation is used in order to reduce complexity. Furthermore, the improved method is still based on tetrahedral decomposition, so that existing implementations of the method of LP can be easily improved without the need of integrating new polyhedral routines. While the improved method is discretely equivalent to the method of OD, the computational costs are reduced making it favorable for large-scale primary atomization simulations.

This paper is structured as follows: the considered problem is mathematically formulated in Section 2. While the equations presented in Section 2 are exact, Section 3 describes the methodology for a discrete solution of these equations. Section 4 presents the proposed projected cell correction to achieve a conservative framework and its implementation is detailed in Section 5. The performance of the proposed improvement is analyzed and compared to the method of LP and OD in 2D and 3D benchmark test cases in Section 6. The advantages of the improvement are demonstrated in a real application case in Section 7, and the paper finishes with conclusions in Section 8.

2. Mathematical formulation

This section starts with a short introduction to the general ideas of VOF methods and their governing equations in Subsection 2.1. The cell-based and flux-based approaches for unsplit geometric advection are presented in Subsection 2.2 and Subsection 2.3, respectively, as the proposed scheme combines advantages from both approaches.

2.1. Volume-of-Fluid method

In interfacial flows, an interface $\Sigma(t)$ separates the physical domain Ω into two space-filling disjoint phases, $\Omega_1(t)$ and $\Omega_2(t)$. VOF methods aim for capturing the interface evolution by using an indicator function that is defined by

$$f(\mathbf{x},t) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_1(t), \\ 0 & \text{if } \mathbf{x} \notin \Omega_1(t). \end{cases} \tag{1}$$

Here, $\mathbf{x}$ and t indicate the position vector and time, respectively. If phase change is neglected and both phases are considered as incompressible, the evolution of the indicator function is governed by

$$\frac{\partial f}{\partial t} + \nabla \cdot (f\mathbf{u}) = 0, \tag{2}$$

where $\mathbf{u}$ is the velocity field. In addition to the strong form of the indicator transport equation, the evolution of the indicator function can be described by the integral form

$$\frac{\mathrm{D_m}}{\mathrm{D_m}t} \int_{\Omega_\mathrm{m}(t)} f(\mathbf{x},t)\mathrm{d}V(t) = 0, \tag{3}$$

where $\Omega_\mathrm{m}(t)$ denotes a material volume and $\mathrm{D_m}$ the material derivative. In numerical simulations, the physical domain Ω is decomposed into disjoint static subvolumes $\Omega_k^{\mathcal{E}}$ with $k \in C^{\mathcal{E}}$, where $C^{\mathcal{E}} = \{1, \dots, N^{\mathcal{E}}\}$, and $N^{\mathcal{E}}$ is the number of subvolumes. The subvolumes are denoted as Eulerian cells in this work and

Eulerian cell-related quantities are labeled with a superscript $\mathcal{E}$. The surface of each Eulerian cell $\partial\Omega_k^{\mathcal{E}}$ is partitioned into disjoint Eulerian cell faces $\mathcal{A}_{k,i}$ with $i \in C_k^{\mathcal{A}}$. Here, $C_k^{\mathcal{A}} = \{1, \ldots, N_k^{\mathcal{A}}\}$ and $N_k^{\mathcal{A}}$ indicate the number of faces of the k-th Eulerian cell. VOF methods solve for a volume-averaged value, which is generally referred to as the volume fraction. The volume fraction for an Eulerian cell $\Omega_k^{\mathcal{E}}$ is defined by

$$F_k(t) = F_{\Omega_k^{\mathcal{E}}}(t) = \frac{\int_{\Omega_k^{\mathcal{E}}} f(\mathbf{x}, t)\mathrm{d}V}{V_k^{\mathcal{E}}}, \tag{4}$$

where $V_k^{\mathcal{E}} = \int_{\Omega_k^{\mathcal{E}}} \mathrm{d}V$. Even though the equation is solved for a volume-averaged quantity, it is important to note that a sharp representation of the interface is required during advection to obtain meaningful results. Two ways of addressing this challenge are the unsplit cell-based and the unsplit flux-based advection approach, which will be discussed in the following.

2.2. Cell-based volume fraction advection

Unsplit cell-based volume fraction advection relies on solving the integral indicator transport equation (Eq. (3)) over a static Eulerian mesh between t^n and t^{n+1}. In order to update the volume fraction, phase-specific volumes or the entire Eulerian cell are either tracked forward or backward in time and remapped on the Eulerian mesh.

The improvement is proposed for the backward scheme of LP, which projects the whole computational cell backward in time. The backward projection is schematically shown for a 2D case in Figure 1. The advancement of the volume fraction can be described as

$$F_k^{n+1} = F_{\Omega_k^{\mathcal{E}}}(t^{n+1}) = F_{\Omega_k^{\mathcal{L}}}(t^n) = \frac{\int_{\Omega_k^{\mathcal{L}}} f(\mathbf{x}, t^n)\mathrm{d}V}{V_k^{\mathcal{L}}}, \tag{5}$$

where $\Omega_k^{\mathcal{L}}$ indicates a Lagrangian cell, which is obtained by moving the Eulerian cell $\Omega_k^{\mathcal{E}}$ backward in time from t^{n+1} to t^n. As the Eulerian cell $\Omega_k^{\mathcal{E}}$ is assumed to be a material volume, the motion is described by

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u}(\mathbf{x}, t) \text{ with } \mathbf{x}(t_0) = \mathbf{x}_0, \tag{6}$$

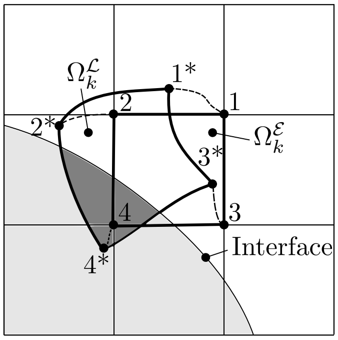


Figure 1: Schematic representation of a Cartesian Eulerian cell $\Omega_k^{\mathcal{E}}$ defined by the vertices 1234 with its corresponding Lagrangian cell $\Omega_k^{\mathcal{L}}$. The dashed lines illustrate the Lagrangian trajectories of the cell vertices. The reference phase at t^n is shown in light grey, while the reference phase volume embedded inside the Lagrangian cell is highlighted in dark grey.

where $\mathbf{x}$ is the position of a point on the surface of the material volume and $\mathbf{x_0}$ its initial position. Mass conservation is ensured if the backward projection of the Eulerian cell conserves the volume.

2.3. Flux-based volume fraction advection

The update equation for unsplit flux-based advection schemes is derived from the strong form of the indicator function transport equation (Eq. (2)). Following the detailed derivation presented by OD, the volume fraction of an Eulerian cell $\Omega_k^{\mathcal{E}}$ can be advanced according to

$$F_k^{n+1} = F_k^n - \frac{1}{V_k^{\mathcal{E}}} \sum_{i \in C_k^{\mathcal{A}}} \left[\int_{\Omega^+_{k,i}} f(\mathbf{x}, t^n) \mathrm{d}V - \int_{\Omega^-_{k,i}} f(\mathbf{x}, t^n) \mathrm{d}V \right]. \tag{7}$$

Here, $\Omega^+_{k,i}$ and $\Omega^-_{k,i}$ are the positive and negative contributions of the total flux volume of the i-th Eulerian cell face. The total flux volume of the i-th Eulerian cell face $\Omega^{\mathrm{F}}_{k,i}$, such that $\Omega^{\mathrm{F}}_{k,i} = \Omega^+_{k,i} \cap \Omega^-_{k,i}$, is formed through a streak-tube that is formed by moving the cell face $\mathcal{A}_{k,i}$ backward in time along its Lagrangian trajectories according to Eq. (6). Positive and negative contributions are defined from the outward-facing normals of the Eulerian cell $\mathbf{n}^{\mathcal{A}}_{k,i}$ and the flux volume $\mathbf{n}^{\mathrm{F}}_{k,i}$. If the normal vectors are pointing in the same direction, the flux contribution is positively defined. It is important to note that the normal vectors are defined on the shared Eulerian cell face $\mathcal{A}_{k,i}$ and that the direction of the flux volume normal vector $\mathbf{n}^{\mathrm{F}}_{k,i}$ changes at the location where the projected cell face crosses the cell face. Figure 2 illustrates the sign convention using the example of a simple negative flux volume and a crossed flux volume for a 2D Cartesian Eulerian cell.

In the flux-based formulation, mass is conserved if the flux velocities, defined by

$$\mathcal{U}_{k,i} = \frac{\int_{\Omega^+_{k,i}} \mathrm{d}V - \int_{\Omega^-_{k,i}} \mathrm{d}V}{\mathcal{A}_{k,i}\Delta t}, \tag{8}$$

are divergence free

$$\sum_{i=1}^{N^{\mathcal{A}}_k} \mathcal{U}_{k,i}\mathcal{A}_{k,i} = 0. \tag{9}$$

It is important to highlight that the cell-based and the flux-based approach are discretely equivalent as rigorously shown by Zhang [34]. Furthermore, the complexity of both approaches is comparable as both require the computation

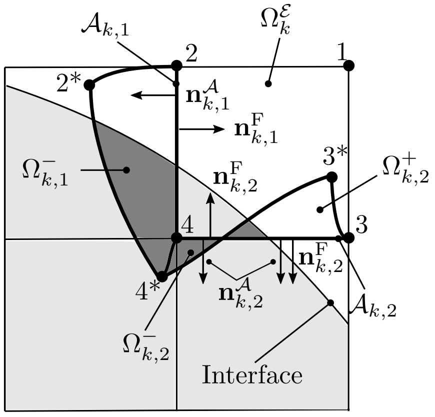


Figure 2: Schematic representation of a negative (index 1) and a crossed (index 2) flux volume (think lines) for an Eulerian cell $\Omega^{\mathcal{E}}_k$ that is defined by the vertices 1234. The reference phase at t^n is shown in light grey, and the reference phase volume embedded in the flux volumes is highlighted in dark grey.

of the reference volume embedded inside an arbitrarily shaped volume. The arbitrary volumes correspond to the Lagrangian cells in the cell-based approach and the flux volumes in the flux-based approach. A notable difference, however, is that the computation of the intersection is only needed once in the cell-based approach, while the flux-based approach requires its computations for each face. Thus, it can already be expected that the cell-based approach is beneficial in terms of computational costs compared to the flux-based approach.

3. Methodology

In this paper, the 3D unsplit cell-based method of LP is improved. The general solution procedure for discretely solving Eq. (5), which will be briefly summarized in this section, remains and comprises the steps *interface reconstruction*, *Lagrangian cell estimation*, and *remapping*. The method of LP is improved by introducing a correction of the Lagrangian cell estimation to ensure discrete mass conservation. The Lagrangian cell correction is presented in depth in Section 4.

3.1. Interface reconstruction

The method of LP is based on the Piecewise Linear Interface Calculation (PLIC) method [15, 14] for approximating the indicator function. The piecewise linear interface approximation inside the k-th Eulerian cell $\tilde{f}_k$ can be described by

$$\tilde{f}_k(\mathbf{x}, t) = \mathcal{H}(d_k^\Gamma - \mathbf{n}_k^\Gamma \cdot \mathbf{x}), \tag{10}$$

with $\mathbf{x} \in \Omega_k^\mathcal{E}$ and $\mathcal{H}$ as the Heaviside function. The normal vector $\mathbf{n}_k^\Gamma$ and position parameter d_k^Γ uniquely parameterize the indicator function approximation in each Eulerian cell $\Omega_k^\mathcal{E}$. Taking into account that the approximated indicator function is discontinuous, Eq. (5) can be reformulated as

$$F_k^{n+1} = \frac{1}{V_k^\mathcal{L}} \sum_{l \in C_k^\mathcal{L}} \int_{\Omega_k^\mathcal{L} \cap \Omega_l^\mathcal{E}} \tilde{f}_l(\mathbf{x}, t^n) \, \mathrm{d}V, \tag{11}$$

where $C_k^\mathcal{L} = \{l \in C^\mathcal{E} \mid \Omega_k^\mathcal{L} \cap \Omega_l^\mathcal{E} \neq \emptyset\}$ and consequently defines a subset of Eulerian cells that intersect with the Lagrangian cell $\Omega_k^\mathcal{L}$.

3.2. Lagrangian cell estimation

The Lagrangian cell $\Omega_k^{\mathcal{L}}$ is defined by the material evolution of the Eulerian cell $\Omega_k^{\mathcal{E}}$. This results in an arbitrarily curved shape as depicted in Figure 1. The method of LP was presented for hexahedral Eulerian cells, and LP proposed to discretize the resulting Lagrangian cell by six tetrahedra, while the vertices connectivity from the Eulerian cell $\Omega_k^{\mathcal{E}}$ is maintained. As a result, the discrete Lagrangian cell, denoted as $\tilde{\Omega}_k^{\mathcal{L}}$, is fully characterized by the vertices whose positions are obtained by integrating Eq. (6) backward in time. In the following, these vertices are called transported vertices and are indicated using a superscript asterisk. As a simplified illustration, a 2D Lagrangian cell discretization based on two triangles is schematically shown in Figure 3. To account for the Lagrangian cell discretization using a set of tetrahedra, Eq. (11) can be rewritten as

$$F_k^{n+1} = \frac{1}{\tilde{V}_k^{\mathcal{L}}} \sum_{j=1}^{N_{\mathrm{tet}}^{\mathcal{L}}} \sum_{l \in C_{k,j}^{\mathcal{T}}} \int_{\Omega_k^{\mathcal{L}} \cap \Omega_l^{\mathcal{E}}} \tilde{f}_l(\mathbf{x}, t^n) \, \mathrm{d}V. \tag{12}$$

Here, $C_{k,j}^{\mathcal{T}} = \{l \in C^{\mathcal{E}} \mid \mathcal{T}_{k,j}^{\mathcal{L}} \cap \Omega_l^{\mathcal{E}} \neq \emptyset\}$ is the subset of the Eulerian cells that intersect with the j-th tetrahedron of the k-th discrete Lagrangian cell $\mathcal{T}_{k,j}^{\mathcal{L}}$. The number of tetrahedra used for the discretization is denoted by $N_{\mathrm{tet}}^{\mathcal{L}}$, and for the method of LP, $N_{\mathrm{tet}}^{\mathcal{L}} = 6$. In the correction presented in Section 4, the total number of tetrahedra used is increased.

3.3. Remapping

The remapping step aims to compute the reference phase volume embedded inside the discrete Lagrangian cell. The task is to compute the reference phase volume inside each of the tetrahedra forming the Lagrangian cell. As indicated by Eq. (12), each tetrahedron first needs to be clipped by the planes describing the Eulerian cell before the resulting convex polyhedron can be further cut by the PLIC representation of the corresponding Eulerian cell. Finally, the volume of the complex polyhedron can be computed, which corresponds to the reference phase volume inside the tetrahedra.

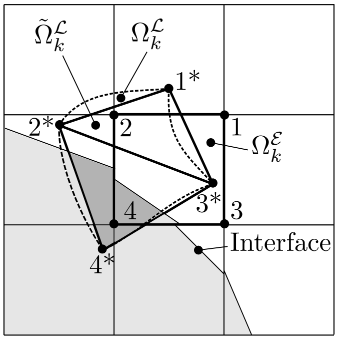


Figure 3: 2D schematic of a discretized Lagrangian cell $\tilde{\Omega}_k^{\mathcal{L}}$. The corresponding Eulerian cell $\Omega_k^{\mathcal{E}}$ and Lagrangian cell $\Omega_k^{\mathcal{L}}$ (dashed line) are also shown. The PLIC reference phase at t^n is shown in light grey, and the reference phase volume embedded in the discrete Lagrangian cell is highlighted in dark grey.

4. Lagrangian cell correction

Due to the numerical integration of Eq. (6) and the triangulation, the discrete Lagrangian cell $\tilde{\Omega}_k^{\mathcal{L}}$ does not maintain the volume of the Eulerian cell $\Omega_k^{\mathcal{E}}$. However, this is required to conserve mass as discussed in Subsection 2.2. Even though the exact volume is known, correcting the vertices to match the exact volume generally results in an under-determined non-linear coupled system of equations, which has to be solved globally.

To overcome this drawback and to provide additional degrees of freedom for correcting the discrete Lagrangian cell, similar to CS, additional vertices on the Lagrangian cell faces are introduced. The additional vertices are denoted as correction vertices and are indicated by a superscript c in the following. In contrast to the transported vertices, the position of the correction vertices is not governed by Lagrangian trajectories. Instead, the positions are obtained imposing the mass conservation condition based on flux-based considerations. The general correction idea is schematically shown for a 2D configuration in Figure 4. The corrected reference phase volumes, either positive or negative, are highlighted in dark grey. The correction requires that the volume of the discrete Lagrangian cell is equal to the Eulerian cell volume subtracted by the

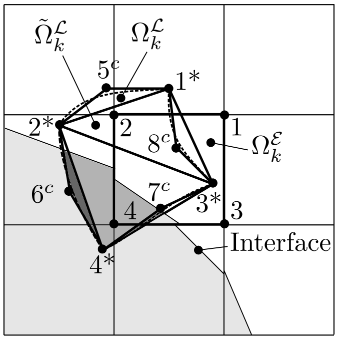


Figure 4: 2D schematic of the Lagrangian cell correction. The reference phase at t^n is shown in light grey, the reference phase volume embedded in the uncorrected discrete Lagrangian cell is highlighted in medium grey, and the reference phase volume considered by the correction is shown in dark grey.

signed volume of each discrete flux volume. In three dimensions, where the Lagrangian cell and flux volumes are discretized using tetrahedra, it follows that

$$\sum_{j=1}^{N_{\text{tet}}^{\mathcal{L}}} \text{Vol}(\mathcal{T}_{k,j}^{\mathcal{L}}) = V_k^{\mathcal{E}} - \sum_{i=1}^{N_k^{\mathcal{A}}} \sum_{j=1}^{N_{\text{tet}}^{\text{F}}} \text{Vol}(\mathcal{T}_{k,i,j}^{\text{F}}) \tag{13}$$

must hold. Here, $N_{\text{tet}}^{\text{F}}$ is the number of tetrahedra used for the discretization of the flux volumes and $\mathcal{T}_{k,i,j}^{\text{F}}$ is the j-th tetrahedron of the discrete flux volume that corresponds to the i-th Eulerian cell face $\mathcal{A}_{k,i}$. The signed volume of a tetrahedron $\text{Vol}(\mathcal{T})$ that is defined by the vertices $\{\mathbf{x}_1, \dots, \mathbf{x}_4\}$ is given by

$$\text{Vol}(\mathcal{T}) = \text{Vol}(\mathbf{x}_1, \dots, \mathbf{x}_4) = \frac{(\mathbf{x}_2 - \mathbf{x}_1) \cdot ((\mathbf{x}_3 - \mathbf{x}_1) \times (\mathbf{x}_4 - \mathbf{x}_1))}{6}. \tag{14}$$

A minimum number of tetrahedra is desired to keep computational costs low. As discussed by LP, a minimum of six tetrahedra is required for the Lagrangian cell estimation to ensure cell-to-cell compatibility, which is needed to avoid gaps and overlaps between adjacent Lagrangian cells that ultimately would result in conservation errors. In addition, a minimum of two tetrahedra is required for the correction pyramid on each face of the discrete Lagrangian cell. As a result, 18 tetrahedra are used to triangulate the Lagrangian cell, which is visualized in Figure 5. Note, that the vertices ordering needs to be chosen such that

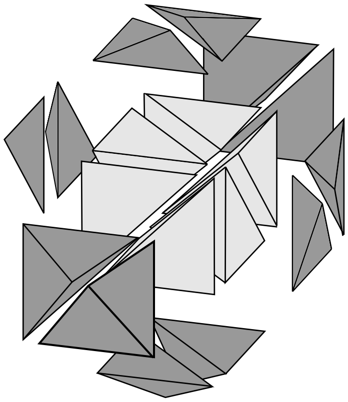


Figure 5: Exploded view of the Lagrangian cell triangulation.

the volumes of unflipped tetrahedra have a consistent sign and that cell-to-cell compatibility is ensured. The discrete flux volumes are formed from a combination of selected Lagrangian cell and Eulerian cell vertices resulting in eight tetrahedra per discrete flux volume. Again, a consistent sign convention and face-to-face compatibility needs to be provided by the vertices ordering. The used triangulation connectivity matrices for the triangulation of the Lagrangian cell and the flux volume are provided in Appendix A.

It is straightforward to show that Eq. (13) holds for a consistent discretization of the Lagrangian cell and the flux volumes. As a result, the position of the correction vertices can be computed using conservative flux volumes, which means that the effective flux velocities are solenoidal. In several flux-based methods [24, 26, 25, 35] the flow solver velocity that is stored on the cell faces has been used to enforce the effective flux velocity to be solenoidal. As detailed in Section 5, the improved method is implemented in a spatially staggered Cartesian code meaning that face velocities are available for the correction. The equality of the solenoidal face velocity $\mathbf{u}^{\mathcal{A}}$ and the discrete effective flux velocity can be written in terms of volume

$$(\mathbf{u}^{\mathcal{A}} \cdot \mathbf{n}^{\mathcal{A}}_{k,i})\mathcal{A}_{k,i}\Delta t = \sum_{j=1}^{8} \mathrm{Vol}(\mathcal{T}^{\mathrm{F}}_{k,i,j}). \tag{15}$$

From this, a volume deficit $V^{\mathrm{c}}_{k,i}$ that needs to be corrected can be defined according to

$$V^{\mathrm{c}}_{k,i} = (\mathbf{u}^{\mathcal{A}} \cdot \mathbf{n}^{\mathcal{A}}_{k,i})\mathcal{A}_{k,i}\Delta t - \sum_{j=3}^{8} \mathrm{Vol}(\mathcal{T}^{\mathrm{F}}_{k,i,j}) = \sum_{j=1}^{2} \mathrm{Vol}(\mathcal{T}^{\mathrm{F}}_{k,i,j}). \tag{16}$$

Note that the choice of the time of the solenoidal face velocity $\mathbf{u}^{\mathcal{A}}$ does not affect the conservation property of the scheme, but it affects the geometrical accuracy. Considering the generic flux volume shown in Figure 6, the only unknown on the right-hand side of Eq. (16) is the position of the correction vertex $\mathbf{o}$. Following OD, the coordinates of vertex $\mathbf{o}$ tangential $\{o_{\mathrm{t}1}, o_{\mathrm{t}2}\}$ to the Eulerian cell face are set to the barycenter of the vertices **bdfh**, and the coordinate of vertex $\mathbf{o}$ normal to the Eulerian cell face is obtained according to

$$o_{\mathrm{n}} = \frac{6V^{\mathrm{c}} - A_{\mathrm{t}_1}o_{\mathrm{t}_1} - A_{\mathrm{t}_2}o_{\mathrm{t}_2} + \mathbf{A}\cdot\mathbf{b} + \mathbf{B}\cdot\mathbf{f} - B_{\mathrm{t}_1}o_{\mathrm{t}_1} - B_{\mathrm{t}_2}o_{\mathrm{t}_2}}{A_{\mathrm{n}} + B_{\mathrm{n}}} \tag{17}$$

with

$$\mathbf{A} = (\mathbf{h} - \mathbf{b}) \times (\mathbf{d} - \mathbf{b}), \tag{18}$$

$$\mathbf{B} = (\mathbf{h} - \mathbf{f}) \times (\mathbf{b} - \mathbf{f}). \tag{19}$$

As the presented conservative flux volume discretization based on a subset of Lagrangian cell vertices is equivalent to the flux volume discretization proposed by OD, the equality of Eq. (13) proves discrete equivalence between the improved

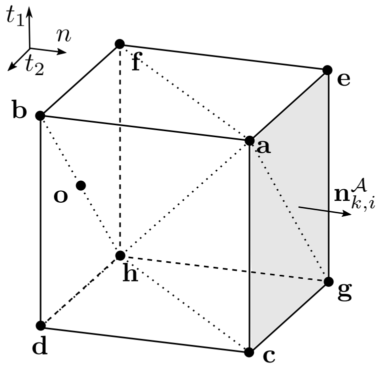


Figure 6: Illustration of a generic discrete flux volume.

method of LP and the method of OD. Thereby, it is assumed that the same indicator function approximation and the same vertex transport is used. A fundamental difference between both methods is, however, that the expensive intersection and volume computations are performed for the tetrahedra of the discrete Lagrangian cell instead for the tetrahedra of the discrete flux volumes.

5. Implementation details

The implementation of the method of LP in the interfacial solver of the inhouse code CIAO, which has been validated in previous studies [31, 32, 33, 36], is extended with the proposed improvement. CIAO uses a 3D, structured Cartesian framework with a staggered variable arrangement in space, where the velocity components are stored on the Eulerian cell faces and the volume fraction is stored at the cell center. The method is implemented in parallel using the domain decomposition strategy.

This section presents implementation details using pseudo-code that focus on efficiency and how existing implementations of the method of LP can be adapted. The general framework for advancing the volume fraction is described in Algorithm 1.

Typically, the volume fraction is constant zero or one in most of the regions in the domain, and only a few cells in the vicinity of the interface need to be updated. To improve the efficiency of the algorithm, a band approach as described by LP is used. In `updateBand`, cells that require an update and their faces are identified and the subset of Eulerian cells and Eulerian cell faces is indicated by C^{Γ} and $C^{\Gamma,\mathcal{A}}$.

The interface reconstruction is performed in `reconstructInterface` and relies on the Mixed-Youngs-Centered (MYC) scheme proposed by Aulisa et al. [20] to estimate the normal of the plane. The position parameter is iteratively adjusted with a relative tolerance of 10^{-16} to match the volume fraction of the cell. Note that analytical relations, as proposed by Scardovelli and Zaleski [37] or Diot and Francois [38], can alternatively be used to obtain the position parameter.

Algorithm 1: Advance volume fraction

```
function advanceVOF($\mathbf{F}^n$, $\mathbf{u}^{\mathcal{A}}$, $\mathbf{x}$, $\Delta t$)
    [$C^{\Gamma}$,$C^{\Gamma,\mathcal{A}}$] $\leftarrow$ updateBand($\mathbf{F}^n$)
    // Detailed in [27]
    [$\mathbf{n}^{\Gamma}, \mathbf{d}^{\Gamma}$] $\leftarrow$ reconstructInterface($C^{\Gamma}$,$\mathbf{F}^n$)
    // Detailed in [20]
    $\mathbf{x}^*$$\leftarrow$ LagrangianCellEstimation($C^{\Gamma}$, $\mathbf{u}^{\mathcal{A}}$, $\mathbf{x}$, $\Delta t$)
    // Solving Eq.(6)
    $\mathbf{x}^{\mathrm{c}}$$\leftarrow$ LagrangianCellCorrection($C^{\Gamma,\mathcal{A}}$, $\mathbf{u}^{\mathcal{A}}$, $\mathbf{x}$, $\mathbf{x}^*$, $\Delta t$)
    // Detailed in Algorithm 2
    $\mathbf{F}^{n+1}$$\leftarrow$ Remapping($C^{\Gamma}$, $C^{\Gamma,\mathcal{A}}$,$\mathbf{x}^*$, $\mathbf{x}^{\mathrm{c}}$, $\mathbf{n}^{\Gamma}$, $\mathbf{d}^{\Gamma}$)
    // Detailed in Algorithm 3
    return $\mathbf{F}^{n+1}$
```

Even though the analytical relations are computationally faster, the iterative approach is adopted from the method of LP as the computational cost share of that part in the volume fraction advection algorithm is negligible. This is discussed in detail in Subsection 6.2. The MYC scheme is formally first-order, but it benefits from its simplicity and low computational costs. Furthermore, numerical tests performed in Subsection 6.1 indicate that the method does not suffer from the first-order reconstruction scheme.

The Lagrangian vertices are estimated by discretely solving Eq. (6) using the Eulerian cell corner vertices as initial conditions. Different integration schemes have been used in the literature ranging from a first-order Euler method [26] to higher-order Runge-Kutta schemes [27, 25, 35]. Following the conclusion of LP stating that the Lagrangian cell estimation improves with a higher-order integration scheme, the presented implementation uses a fourth-order accurate Runge-Kutta scheme for obtaining the Lagrangian vertices. Intermediate velocities are approximated with a tri-linear interpolation in space, and a frozen velocity field is assumed. The frozen field assumption is different from the method of CS, which uses a linear interpolation in time. This, however, requires the knowledge of the velocity at the new time step, which is not known in typical time advancement schemes of flow solver. Typical time advancement schemes either provide the velocity at t^n or $t^{n+1/2}$. Depending on the choice at what time the velocity is frozen, this is a first-order ($\mathbf{u}^{\mathcal{A}} = \mathbf{u}^n$) or second-order ($\mathbf{u}^{\mathcal{A}} = \mathbf{u}^{n+1/2}$) approximation in time. Both vertex velocity approximations are tested in Subsection 6.1 to evaluate the improved method for use along with a flow solvers.

The computation of the correction vertices is detailed in Algorithm 2. As the correction vertices are shared by adjacent cells, it is reasonable to perform the correction vertex computation face-wise and consider all faces either as positive or negative faces. As a result, only the vertex connectivities that belong to positive or negative faces are effectively used. For all Eulerian cell faces of $C^{\Gamma,\mathcal{A}}$, the face direction is determined first in `getFaceDirection`. Afterward, the correction volume and correction vertex position is obtained in `getCorrectionVolume` and `getCorrectionVertex`, respectively.

While the remapping could be done cell-wise for each Eulerian cell summing up the reference volume embedded in each tetrahedron of the discrete Lagrangian cell, it is computationally beneficial to precompute the reference volume em-

Algorithm 2: Compute correction vertices

```
function LagrangianCellCorrection($C^{\Gamma,\mathcal{A}}$, $\mathbf{u}^{\mathcal{A}}$, $\mathbf{x}$, $\mathbf{x}^*$, $\Delta t$)
    forall $i \in C^{\Gamma,\mathcal{A}}$ do
        $d \leftarrow$ getFaceDirection($i$)
        $V_i^{\mathrm{c}} \leftarrow$ getCorrectionVolume($d$, $i$, $\Delta t$, $\mathbf{u}^{\mathcal{A}}$, $\mathbf{x}$, $\mathbf{x}^*$)
        // Solving Eq.(16)
        $x_i^{\mathrm{c}} \leftarrow$ getCorrectionVertex($d$, $i$, $V_i^{\mathrm{c}}$, $\mathbf{x}$, $\mathbf{x}^*$)
        // Solving Eq.(17)
    return $\mathbf{x}^{\mathrm{c}}$
```

bedded in the correction volumes because the correction volumes are shared by adjacent cells. Following this idea, it can be easily shown that the improved method requires $3 - R_{\mathcal{A}}$ times more tetrahedra for intersection computations than the method of LP, where $R_{\mathcal{A}}$ denotes the ratio of shared faces to the total number of faces. In comparison to the method of OD, however, it follows that the improved method uses $(8-4R_{\mathcal{A}})/(3-R_{\mathcal{A}})$ times fewer tetrahedra, meaning that at least two times fewer tetrahedra are needed. This results in a theoretical cost reduction according to [39] of 50%. In Algorithm 3, the remapping step is detailed showing the precomputation structure. In `remapCorrection` and `remapEstimation`, the intersection and volume computations are performed for the correction and estimation tetrahedra using the polyhedra library of LP, already available in CIAO. The remapping steps provide the signed reference phase volume and the signed total volume of the correction tetrahedra and La-

Algorithm 3: Remapping using precomputation of correction tetrahedra

```
function Remapping($C^{\Gamma}$, $C^{\Gamma,\mathcal{A}}$, $\mathbf{x}^*$, $\mathbf{x}^{\mathrm{c}}$, $\mathbf{n}^{\Gamma}$, $\mathbf{d}^{\Gamma}$)
    [$\mathbf{V}^{1,\mathrm{c}}$,$\mathbf{V}^{\mathrm{c}}$] $\leftarrow$ remapCorrection($C^{\Gamma,\mathcal{A}}$, $\mathbf{x}^*$, $\mathbf{x}^{\mathrm{c}}$, $\mathbf{n}^{\Gamma}$, $\mathbf{d}^{\Gamma}$)
    // Detailed in [27]
    [$\mathbf{V}^{1,*}$,$\mathbf{V}^*$] $\leftarrow$ remapEstimation($C^{\Gamma}$, $\mathbf{x}^*$, $\mathbf{n}^{\Gamma}$, $\mathbf{d}^{\Gamma}$)
    // Detailed in [27]
    forall $k \in C^{\Gamma}$ do
        $V^1 \leftarrow V_k^{1,*}$
        $V \leftarrow V_k^*$
        forall $i \in C_k^{\mathcal{A}}$ do
            $V^1 \leftarrow V^1$ - sign($i$) $\cdot$ $V_{k,i}^{1,\mathrm{c}}$
            $V \leftarrow V$ - sign($i$) $\cdot$ $V_{k,i}^{\mathrm{c}}$
        $F_k^{n+1} = V^1/V$
    return $\mathbf{F}^{n+1}$
```

grangian cell estimation. The final step consists of adding or subtracting the precomputed reference phase volumes and total volume to the Lagrangian cell estimates, which depends on the direction of the Eulerian cell face normal. As negative flux volume contributions have a positive contribution to the updated volume fraction, the correction volumes of positive (negative) faces need to be subtracted (added).

6. Performance analysis

In this section, the performance of the improved method is assessed. In Subsection 6.1, various benchmark transport tests are performed to evaluate the conservation and the geometric properties of the scheme. The computational costs of the scheme are investigated in Subsection 6.2.

6.1. Transport tests

Transport tests aim for evaluating the transport properties of advection schemes. In all performed tests, a periodic velocity field is prescribed to transport the interface instead of using a velocity field from a Navier-Stokes solver. Even though the prescribed velocity field could be used during vertex transport, all tests are performed using the velocities at the cell faces either at t^n or $t^{n+1/2}$. This evaluates the transport properties of the method for use along with a temporally first- or second-order Navier-Stokes solver.

Two evaluation metrics are defined. First, the change of reference phase volume is measured after one period T according to

$$E_{\text{mass}} = \frac{\left| \sum_{k \in C^{\mathcal{E}}} V_k^{\mathcal{E}} F_k(T) - \sum_{k \in C^{\mathcal{E}}} V_k^{\mathcal{E}} F_k^0 \right|}{\sum_{k \in C^{\mathcal{E}}} V_k^{\mathcal{E}} F_k^0}, \tag{20}$$

where F_k^0 indicates the initial volume fraction of the k-th Eulerian cell. In periodic flow fields, the interface shape is retained from one period to another. This is, however, not the case for numerically capturing the interface motion. To evaluate this deviation the shape error, which is defined by

$$E_{\text{shape}} = \frac{\sum_{k \in C^{\mathcal{E}}} V_k^{\mathcal{E}} \left| F_k(T) - F_k^0 \right|}{\sum_{k \in C^{\mathcal{E}}} V_k^{\mathcal{E}} F_k^0}, \tag{21}$$

is introduced as a second metric.

The VOFI library [40, 41] is used to initialize the volume fractions. 2D test cases are realized in the 3D code structure by using a one-cell thick 3D domain and periodic boundary conditions in the third direction.

6.1.1. Zalesak's disk

In Zalesak's disk case [16], a notched disk is rotated in a solid body manner to assess the ability of the improved VOF scheme to advect sharp corners. Solid body rotation around the origin of the coordinate system with a period of $T = 1$ is described by a velocity field of

$$\mathbf{u}(\mathbf{x}, t) = \begin{pmatrix} u(\mathbf{x}, t) \\ v(\mathbf{x}, t) \end{pmatrix} = \begin{pmatrix} -2\pi y \\ +2\pi x \end{pmatrix} . \tag{22}$$

The notched disk is initially centered at $\mathbf{x} = (0, 0.25)^T$ within a $[-0.5, 0.5] \times [-0.5, 0.5]$ domain. The disk diameter, notch height, and notch width are 0.3, 0.25, and 0.05, respectively. Simulations are performed for four different mesh resolutions, namely 50^2, 100^2, 200^2, and 400^2. In all cases, a constant time step is used that corresponds to a CFL number of $\pi/4$. As the velocity field is steady, there is no difference in the use of a first- or second-order approximation for the vertex velocities. Figure 7 shows the PLIC representation of the interface initially and after one rotation. Even for the coarsest mesh that resolves the notch width by 2.5 grid points, the general shape of the disk is maintained. In Figure 8, the mass conservation error (left) and shape error (right), evaluated according to Eq. (20) and Eq. (21), respectively, are presented for the improved method. For comparison, the results without the proposed Lagrangian cell correction, which corresponds to the method of LP, and the reported values of OD normalized by the notched disk area are additionally shown. Normalization is required because OD used absolute definitions for the mass and shape error. The mass conservation error is independent of the grid size and always close to machine precision for the improved method and the method of OD. This agrees well with the expectation because both methods are discrete-conservative. The uncorrected method shows a fifth-order scaling as reported by LP. No significant differences in the shape error for the uncorrected and corrected method can be observed. All presented values, including those of OD, show a convergence between first- and second-order.

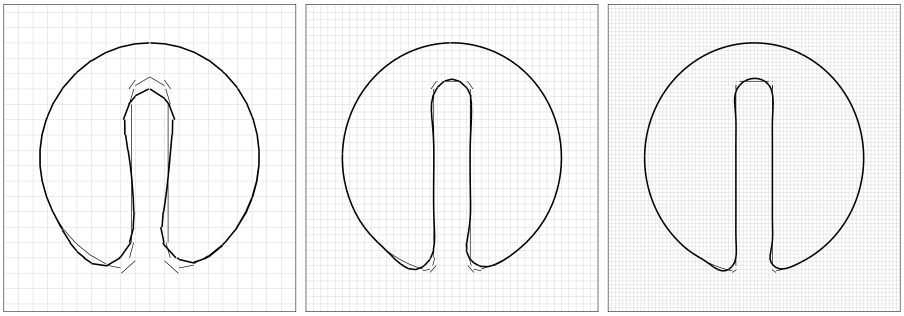
6.1.2. Two-dimensional deformation

The second transport test was initially proposed by Leveque [42] and comprises stretching and un-stretching of a disk in a swirling flow described by

$$\mathbf{u}(\mathbf{x},t) = \begin{pmatrix} u(\mathbf{x},t) \\ v(\mathbf{x},t) \end{pmatrix} = \begin{pmatrix} -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y)\cos(\pi t/T) \\ +2\sin(\pi x)\cos(\pi x)\sin^2(\pi y)\cos(\pi t/T) \end{pmatrix} \tag{23}$$

with $T = 8$. The center of the disk of diameter 0.3 is initialized at $\mathbf{x} = (0.5, 0.75)^T$ within a $[0,1]\times[0,1]$ domain. Simulations are performed for meshes ranging from 32^2 to 512^2 using a constant time step that corresponds to a CFL number of 0.5 based on the maximal velocity of 1.

Figure 9 shows the PLIC interface representation at $t/T = 0.5$ for the meshes 64^2, 128^2, and 256^2 using a second-order velocity field approximation for vertex transport. At that point, the interface is at a maximum stretch and the velocity field reverses. Breakup of the thin tail can be observed for all presented meshes, which is a well-known behavior of interface capturing methods in underresolved regions. The PLIC interface representation after one period is compared to the



(a) 50^2 mesh

(b) 100^2 mesh

(c) 200^2 mesh

Figure 7: PLIC representation of the interface before (thin line) and after (thick line) one rotation of Zalesak's disk for three different meshes.

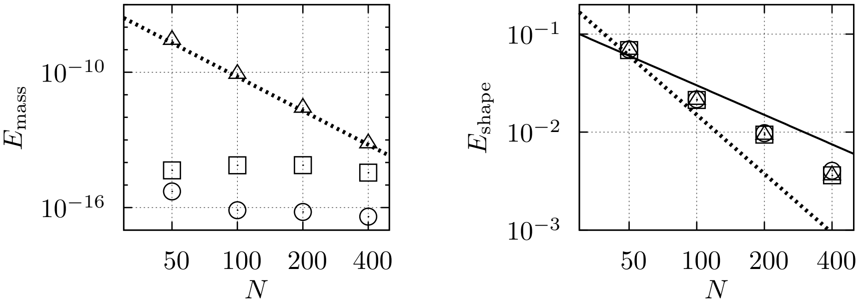
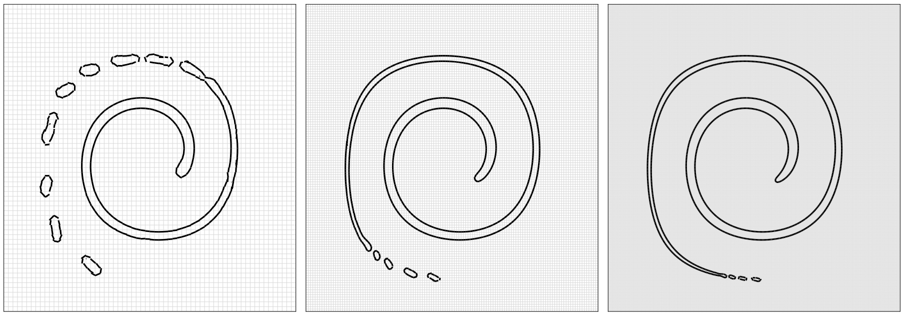


Figure 8: Convergence of the mass conservation error (left) and shape error (right) for Zalesak's disk case for the improved method (squares), the method of LP (triangles), and the method of OD (circles). Dotted lines show fifth-order (left) and second-order (right) scaling. The solid line indicates first-order scaling.



(a) 64^2 mesh

(b) 128^2 mesh

(c) 256^2 mesh

Figure 9: PLIC interface representation at $t/T = 0.5$ for the 2D deformation case computed for three different meshes using the improved method with a second-order approximation for the vertex velocities.

initial PLIC interface representation in Figure 10 for the same three meshes and velocity field approximation. While the interface visibly deviates from the initial interface shape for the 64^2 mesh, those deviations almost disappear on the 256^2 mesh. The mass conservation and shape error convergence are shown in Figure 11 for the improved method using a first- and second-order approximation for the vertex velocities, the method of LP using a second-order approximation for the vertex velocities, and the method of OD (normalized by the disk

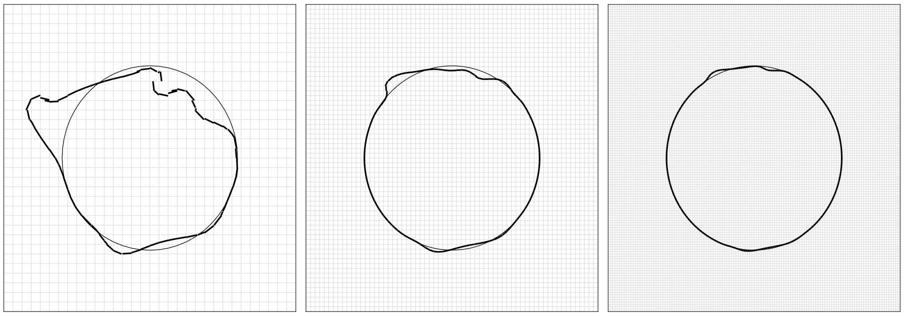

(a) 64^2 mesh (b) 128^2 mesh (c) 256^2 mesh

Figure 10: PLIC interface representation at $t/T = 0$ (thin line) and $t/T = 1$ (thick line) for the 2D deformation case for three different meshes using the improved method with a second-order approximation for the vertex velocities.

area). Note that the values reported by OD are all based on a second-order approximation for the vertex velocities. Discrete conservation is observed for the improved method independent of the temporal velocity field approximation and the method of OD. The method of LP shows a third-order scaling as reported. The shape error of the improved method with a second-order approximation for the vertex velocities shows second-order convergence and demonstrates the robustness of the method in underresolved situations. Also in this test case, no significant influence of the Lagrangian cell correction on the shape error can be identified. Moreover, a similar scaling to the method of OD is visible, although a second-order Runge-Kutta scheme for vertex transport and a second-order ELVIRA scheme [43] for reconstruction were used. A reduced scaling between first- and second-order can be observed for the improved method along with a first-order approximation for the vertex velocities.

6.1.3. Three-dimensional deformation

The last transport test is a 3D extension of the transport test in Subsubsection 6.1.2 and was also proposed by Leveque [42]. This test is used to demonstrate the transport properties and the robustness in underresolved situations in a 3D configuration, and is consequently the most relevant case in the context of

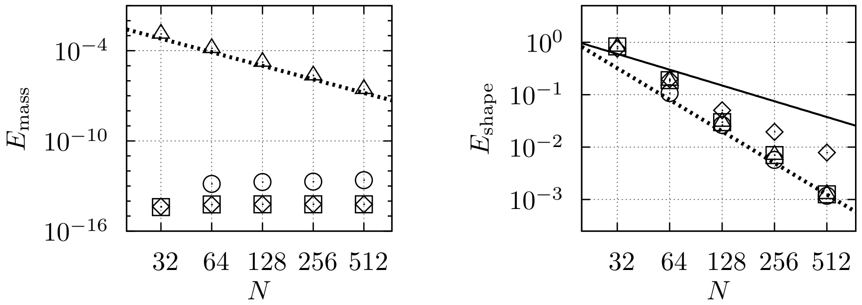


Figure 11: Convergence of the mass error (left) and shape error (right) for the 2D deformation case using the proposed method (first-order: diamonds, second-order: squares), the method of LP (triangles), and the method of OD (circles). The dotted line indicates third-order (left) and second-order (right) convergence. The solid line shows first-order scaling.

primary atomization simulations. The interface is transported with the velocity field

$$\mathbf{u}(\mathbf{x},t)=\begin{pmatrix}u(\mathbf{x},t)\\ v(\mathbf{x},t)\\ w(\mathbf{x},t)\end{pmatrix}=\begin{pmatrix}2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos(\pi t/T)\\ -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\cos(\pi t/T)\\ -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)\cos(\pi t/T)\end{pmatrix} \tag{24}$$

with $T = 3$. The initial interface shape is a sphere of diameter 0.3 and is centered at $\mathbf{x} = (0.35, 0.35, 0.35)^T$ in a $[0,1] \times [0,1] \times [0,1]$ domain. The domain is discretized using a 32^3, 64^3, 128^3, and 256^3 mesh. A constant time step size that corresponds to a CFL number of 0.5 based on the maximal possible velocity is used throughout all simulations. In Figure 12, the PLIC representation of the interface at $t/T = 0.5$ is shown for three meshes using a second-order approximation for the vertex velocities. At $t/T = 0.5$ the sphere is at a maximum stretch and a thin reference phase sheet is formed. For the coarsest mesh, the grid resolution is too low to maintain the thin sheet, which results in artificial breakup. While for the 128^3 mesh still an interface disturbance can be observed, the method well captures the thin sheet on a 256^3 mesh. Figure 13 presents the PLIC representation of the interface after one period and shows discrepancies to the expected interface shape of a sphere. The discrepancies, however, reduce significantly with mesh refinement. In Figure 14, the convergence of the conser-

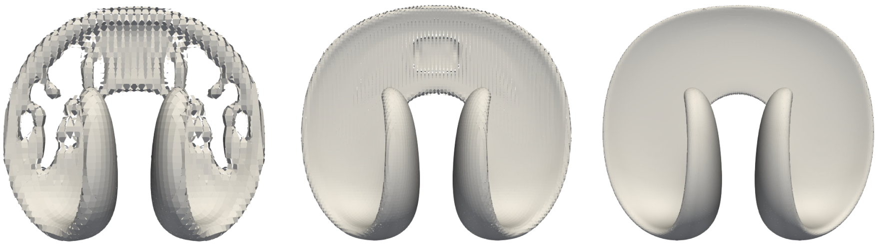

(a) 64^3 mesh (b) 128^3 mesh (c) 256^3 mesh

Figure 12: PLIC interface representation at $t/T = 0.5$ for the 3D deformation case for three different meshes and a second-order approximation for the vertex velocities.

vation (left) and shape error (right) is displayed for the improved method with both vertex velocity approximations and the method of LP using a second-order approximation for the vertex velocities. In addition, the reported error values of OD normalized by the sphere volume are shown for comparison. Similar to the 2D cases, the conservation error for the improved method is always close to machine precision, which is also visible for the values reported by OD. The method of LP shows again a third-order scaling. The shape errors shown on the right of Figure 14 reveal no significant difference and about second-order convergence for the presented methods with a second-order approximation for the vertex velocities as it was also observed in the 2D test cases. The usage of a

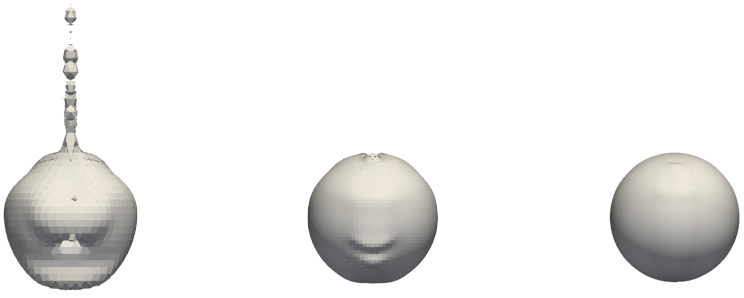

(a) 64^3 mesh (b) 128^3 mesh (c) 256^3 mesh

Figure 13: PLIC interface representation at $t/T = 1$ for the 3D deformation case for three different meshes and a second-order approximation for the vertex velocities.

first-order approximation of the vertex velocities reduces again the scaling, but does not limit the method to first-order. Comparing the improved method and the method of OD, minor differences are observable even though the schemes are discretely equivalent. This can be related to the different vertex integration and interface reconstruction scheme as mentioned in Subsubsection 6.1.2. At this point, it can be concluded that the MYC reconstruction scheme, which is formally first-order, does not significantly impact the geometric properties of the scheme or it can be compensated by a higher-order integration scheme used for vertex transport.

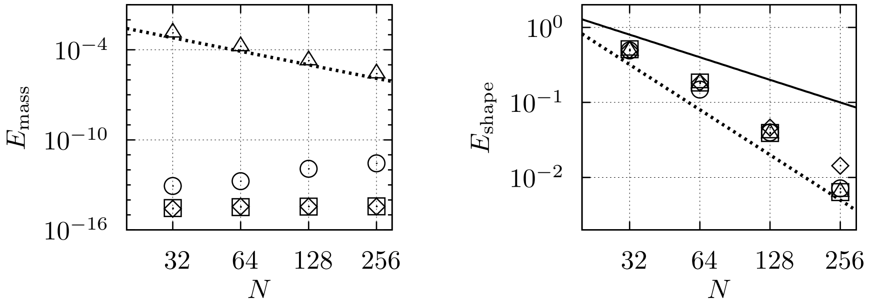


Figure 14: Convergence of the mass conservation error (left) and shape error (right) for the 3D deformation case using the improved method (first-order: diamonds, second-order: squares), the method of LP (triangles), and the method of OD (circles). Dotted line indicate fifth-order (left) and second-order (right) scaling. The solid line shows first-order scaling.

6.2. Computational costs

In this section, the computational costs of the improved method are reported and discussed. The problem of advecting an interface is prone to high load imbalance. As a consequence, the parallel performance strongly depends on the considered case and the corresponding partitioning strategy. To avoid this dependency and to focus on the serial execution benefits of the improved method, this section considers exclusively single-core performance. The computing time is presented for the 3D deformation case (c.f. Subsubsection 6.1.3) using the 64^3 mesh, which was executed on an Intel Xeon E5-2650 V4 processor. During simulation, the number of cells that require an update ranges from 5223 at $t/T = 0$ up to 16 766 at $t/T \approx 0.5$. Each function in Algorithm 1 is measured separately to show the cost distribution of the individual parts of the method. For comparison, the execution time for the method of LP is measured, which essentially just skips the functions `LagrangianCellEstimation` and `remapCorrection` in Algorithm 1.

The measured computing time for each function is visualized in Figure 15 and additionally listed in Table 1 for better readability. As expected, the remapping has the highest share with over 90% independent of whether the improved method or the method of LP is used. The proposed correction increases the total execution time by approximately 57%, where the functions `updateBand`, `LagrangianCellEstimation`, and `remapCorrection` are responsible for the increase. `updateBand` requires more time as the identification of Eulerian cell faces is only needed for the improved method. It is interesting to note, that the computation of the correction vertex position (`LagrangianCellCorrection`) has only a marginal share of the total computation costs. The remapping of the correction tetrahedra increases the total time required for remapping by 58%. During the simulation it was measured that approximately 80% of the Eulerian

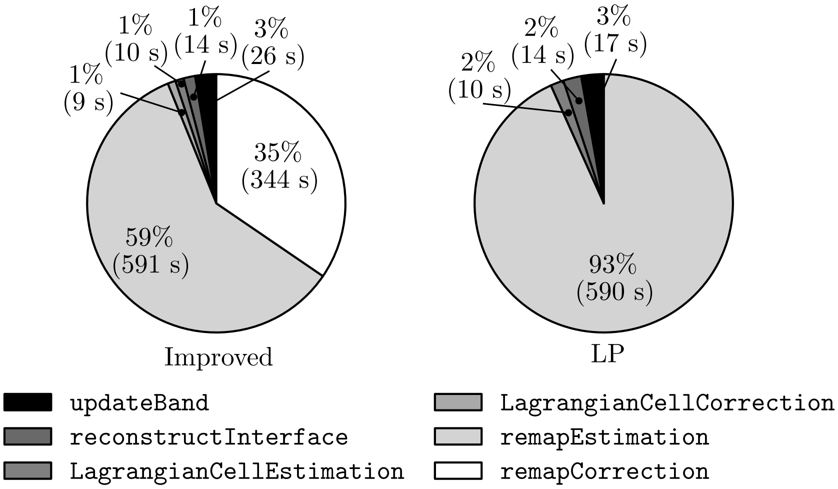


Figure 15: Share of computational costs for the improved method and the method of LP.

	Abs. cost / s		Rel. cost / %	
Function	Improved	LP	Improved	LP
`updateBand`	26	17	3	3
`reconstructInterface`	14	14	1	2
`LagrangianCellEstimation`	10	10	1	2
`LagrangianCellCorrection`	9	0	1	0
`remapEstimation`	591	590	59	93
`remapCorrection`	344	0	35	0

Table 1: Absolute and relativ computational costs of each function for the improved method and the method of LP.

cell faces are shared, which gives a theoretical increase in computing time based on the number of tetrahedra of 120%. Hence, the theoretical estimation overpredicts the increase in computational costs, and it cannot be expected that the improved method reduces the costs to the method of OD by 54%, which is the theoretical estimation for this case. The observed deviation from the theoretical estimation can be explained by the fact that the correction tetrahedra are small compared to the estimation tetrahedra and consequently span fewer Eulerian cells. Hence, fewer intersection and volume computations are required, which reduces computational costs.

To study this further, the required intersection and volume computations to obtain the reference-phase volume embedded in each tetrahedron of the improved method are counted during run-time. In addition to the estimation and correction tetrahedra, the analysis is performed for the tetrahedra forming the discrete flux volumes to provide a comparison to the method of OD. Figure 16 shows the average number of intersection and volume computations required per time step for the improved method and the method of OD. As both methods use equivalent correction tetrahedra, differences are limited to the estimation part of the methods. Here, the discrete flux volumes without the correction tetrahedra are considered as the estimation part of the method of OD. As expected

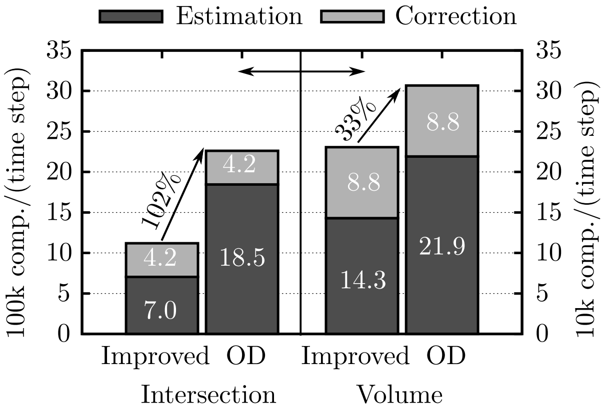


Figure 16: Average number of intersection and volume computations required for the improved method and the method of Owkes and Desjardins (OD). [25]

from the above discussion, the presented values show that fewer computations are required for the correction tetrahedra even though more correction than estimation tetrahedra are used in the improved method. In comparison to the improved method, it follows that the method of OD requires 102% more intersection computations and 33% more volume computations. Note that the values for the intersection computations are an order of magnitude higher, and it was found that the intersection computations require more computing time than the volume computations. It is difficult to obtain a fair and reliable estimation of the effective cost reduction from the presented numbers since the required computational costs for an intersection and volume computation depend strongly on the particular situation. Nevertheless, the presented results show that a noticeable cost reduction can be expected when using the improved method compared to the method of OD.

7. Application

In this section, the improved advection scheme is applied in a simulation of a coaxial air-blast atomizer. The purpose is to demonstrate the conservation property of the improved scheme along with a Navier-Stokes solver in a strongly non-linear realistic flow problem and not to study primary breakup.

7.1. Configuration and flow solver

The computed atomizer configuration is used by the SpraySyn burner [44], which is a standardized burner for researching flame spray synthesis and is in the interest of current research. The non-dimensional numbers of the problem considered are: $\mathrm{Re}_\mathrm{l} = U_\mathrm{l}\rho_\mathrm{l}D_\mathrm{l}/\eta_\mathrm{l} = 80$, $\mathrm{Re}_\mathrm{g} = U_\mathrm{g}\rho_\mathrm{g}D_\mathrm{g}/\eta_\mathrm{g} = 3412$, $\mathrm{We} = U_\mathrm{g}^2\rho_\mathrm{g}D_\mathrm{l}/\sigma = 104$, $M = U_\mathrm{g}^2\rho_\mathrm{g}/(U_\mathrm{l}^2\rho_\mathrm{l}) = 102$, $m = U_\mathrm{l}\rho_\mathrm{l}D_\mathrm{l}^2/(U_\mathrm{g}\rho_\mathrm{g}D_\mathrm{m}h) = 0.22$, $\rho_\mathrm{l}/\rho_\mathrm{g} = 597$, and $\eta_\mathrm{l}/\eta_\mathrm{g} = 52$. Here, U is the bulk velocity, ρ is the density, η is the dynamic viscosity, σ the surface tension, D_l the liquid jet diameter, D_m the average diameter of the annular gas stream, and h its gap thickness. The indices l and g indicate the parameters of the liquid and gas phases, respectively.

The case is particularly challenging for methods as the interface moves along the nozzle tip to the high-speed gas inlet. In that region, the velocity field is strongly non-linear leading to high interpolation errors for the vertex velocities. The error is additionally amplified in cells that are adjacent to the wall on one side and the high-speed inlet on the other. These cells suffer from an imbalance of the error meaning that the wall vertices have no transport error, and the first vertices inside the fluid have a high transport error. One cell layer further downstream, the vertices suffer similar transport errors, but these balance each other. Simply spoken, if on one side too much is fluxed into the cell, on the other side too much is fluxed out of the cell. The error imbalance leads to a large Lagrangian cell compression, which causes high mass conservation errors. The proposed improvement corrects the Lagrangian cell compression and avoids the conservation errors.

In contrast to the transport tests performed in Subsection 6.1, the velocity field, which transports the interface, is not prescribed but is obtained from a two-phase flow Navier-Stokes solver. For this section, the improved advection scheme is exemplarily coupled to a first-order in time implementation (available in CIAO) of the sharp interfacial flow solver of LP [45]. Note that the improved method is not limited to the usage along with flow solvers providing a temporally first-order vertex velocity approximation and can be coupled to flow solvers providing a temporally second-order vertex velocity approximation. In that case, the improved advection scheme is second-order as shown in Subsection 6.1.

The sharp interfacial flow solver of LP [45] solves the two-phase Navier-Stokes equations for Newtonian incompressible fluids with constant surface tension. In the presence of discontinuities in pressure, viscosity, density, and tangential velocity at the interface, special care is required for the individual terms. The flow solver employs a projection method. After advancing the volume fraction, described in detail in Section 5, the velocity is first advanced to a temporary velocity field using the convective and viscous terms and then projected onto a space of divergence-free velocity fields using the pressure gradient. For convective momentum transport, the interfacial flow solver of LP applies similarly to the original volume fraction advection scheme of LP a geometrical semi-Lagrangian approach for each momentum control volume. Consistently to the volume fraction advancement, the convective momentum transport uses a temporally first-order vertex velocity approximation. Note that away from the interface, a central difference scheme is used for the convective term as no special treatment is required in the single-phase regions of the flow. The viscous term is also discretized using central differences, where a harmonic average for the viscosity is employed. Subsequently, the pressure field follows from solving the Poisson equation. In both the pressure gradient as well as the Laplacian of the pressure, the Ghost-Fluid Methodology (GFM) [4] is employed to account for the pressure jump due to surface tension forces. In the original formulation, the GFM requires the knowledge of whether the interface passes in between two cells, the pressure jump at the interface, and the line averaged density. In the volume fraction representation of the interface used in this work, an interface passes in between two cells, indicated with k and $k+1$, if $F_k > 0.5 \wedge F_{k+1} \leq 0.5 \vee F_{k+1} > 0.5 \wedge F_k \leq 0.5$. This condition follows from a Heaviside function consideration as described in detail by Popinet [46]. The curvature is computed using the height function methodology presented by Popinet [47], which provides the curvature at the interface meaning that no further interpolation is required. In contrast to the original GFM [4], the interfacial flow solver of LP uses the volume-averaged density of the corresponding momentum control volume instead of the line averaged density, which leads to a momentum conservative projection step and improved stability at high density ratios. Note that except for convective transport close to the interface, the projection method employs an explicit second-order Runge-Kutta scheme for time integration.

An $11.25D_\mathrm{l} \times 7.5D_\mathrm{l} \times 7.5D_\mathrm{l}$ domain is discretized using a $576 \times 384 \times 384$ (Mesh A) and a $384 \times 256 \times 256$ (Mesh B) mesh resolving the liquid diameter by about 51 and 34 mesh cells, respectively. The time step is limited by a convective CFL number of 0.65. The simulation is advanced for 40 time units, defined by $t^* = tU_\mathrm{g}/D_\mathrm{l}$, after the spray reached steady-state. For comparison, the same time is computed with the advection scheme of LP also.

7.2. Results

A time series of the PLIC representation of the interface obtained using the improved method (top row) and the method of LP (bottom row) on Mesh A along with the nozzle geometry is shown in Figure 17. In general, the disintegration of the liquid core into small droplets and large ligaments can be observed, highlighting the usability as a complex case for stressing the improved advection scheme. After only 23 time unites, differences between the solutions of both methods can be observed in all parts of the breakup process. The close-up in Figure 17 highlights the thickening of the liquid core close to the nozzle for the method of LP. The thickening results from the substantial mass conservation errors in that region, which has been discussed in Subsection 7.1. As visible in Figure 17, the thickened liquid core tends to create unphysical droplets due to an excess of liquid volume. Similar observations can be made for the PLIC representations of the interface using both methods on Mesh B, which are shown in Figure 18. As indicated in the most left image showing the initial interface used for the comparison, a small liquid structure has been recirculated and deposited

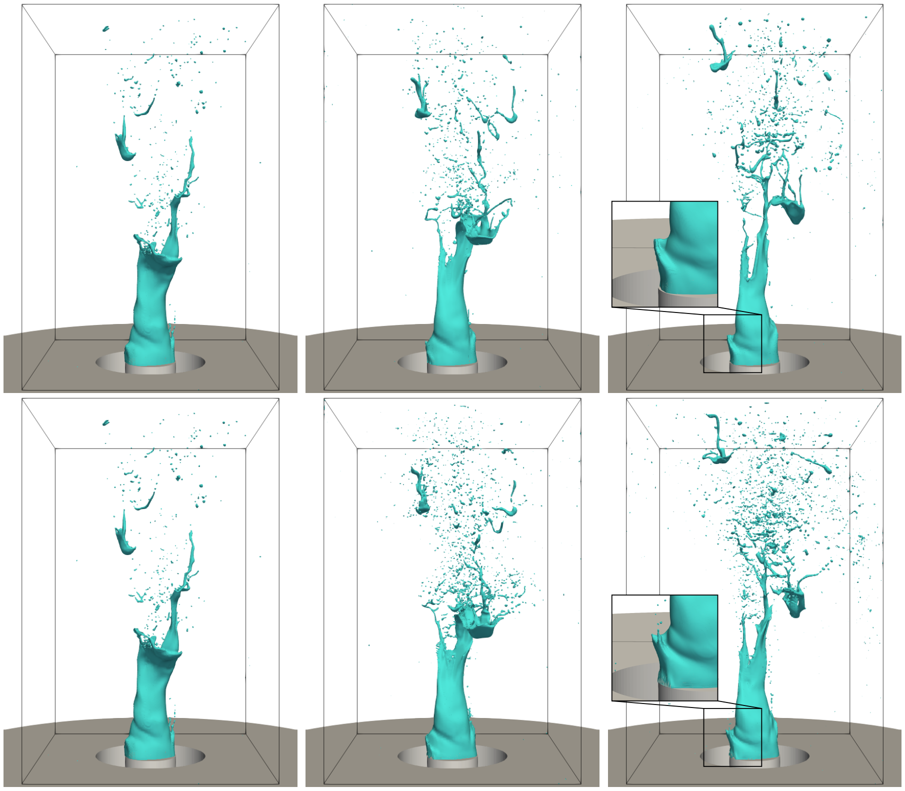


Figure 17: PLIC representation of the interface obtained using the improved method (top row) and the method of LP (bottom row) on Mesh A shown at $t^* = 0, t^* = 12$, and $t^* = 23$ (left to right).

at the nozzle tip close to the outer diameter of the high-speed gas inlet. This initiates, for the method of LP, the growth of a liquid structure at the outer diameter of the high-speed gas inlet. Again, the growth is the result of the substantial mass conservation errors in that region of the method of LP.

The temporal evolution of the mass conservation error, which is defined similarly to Eq. (20), however with a changing exact liquid volume due to in- and outflows, is presented in Figure 19 for the improved method and the method of LP for Mesh A. Additionally, the infinity norm of the relative Lagrangian cell compression or expansion, defined by $E_{\text{div}} = \max_k(|V_k^{\mathcal{L}} - V_k^{\mathcal{E}}|/V_k^{\mathcal{E}})$ is provided as a local conservation metric. In contrast to the transport tests with a prescribed

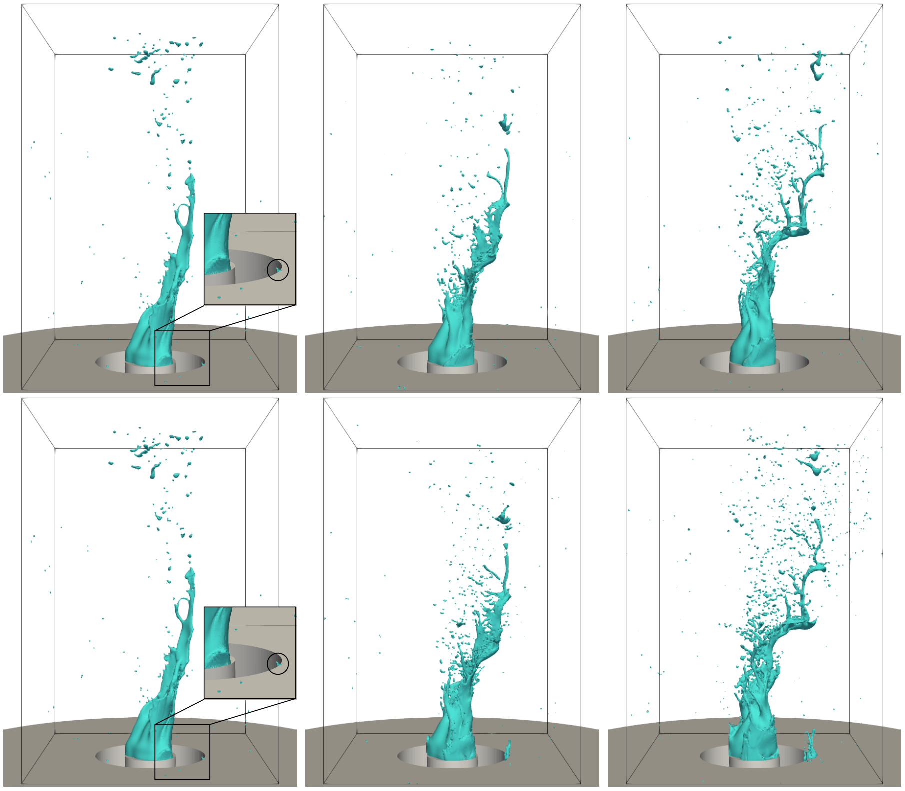


Figure 18: PLIC representation of the interface obtained using the improved method (top row) and the method of LP (bottom row) on Mesh B shown at $t^* = 0, t^* = 12$, and $t^* = 23$ (left to right).

solenoidal velocity field, the velocity field obtained from the Navier-Stokes solver suffers from a residual divergence, which originates from the approximate solution of the Poisson equation in the pressure projection step. In this context, E_{div} provides an upper bound for the local contribution to the mass conservation error and is consequently a useful metric to evaluate the mass conservation error as it can not be expected to be close to machine precision. Figure 19 shows that the mass conservation error for the method of LP converges to a value in the order of 10^{-1}, while the improved method limits errors to 10^{-11}. Note that the observable spike in the evolution of the method of LP is caused by the logarithmic plot at the zero crossing of the signed error. E_{div} shows a similar

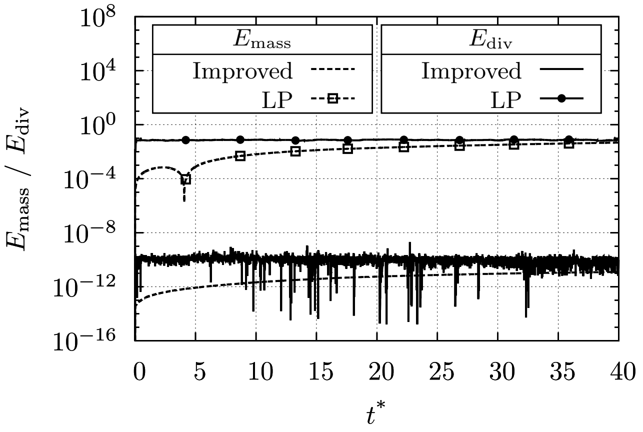


Figure 19: Temporal evolution of the mass conservation error E_{mass} and the infinity norm of the relative Lagrangian cell compression/expansion E_{div} for the improved method and the method of LP for Mesh A in the considered application case.

behavior. While for the improved method E_{div} is in the order of 10^{-10}, which is consistent with the pressure solver tolerance used (10^{-10}), the method of LP shows E_{div} values in the order of 10^{-2}. Note that the value of E_{div} can always be assigned to a cell that is adjacent to the wall and close to the high-speed gas inlet meaning that E_{div} is driven by the error imbalance discussed above.

Before discussing the impact of the grid resolution in the application case, it should be mentioned that the simulations on Mesh A and Mesh B are different realizations. As the details of the temporal evolution of E_{mass} depend on the specific realization, the comparison is focused on E_{div} for discussing the impact of the grid resolution. Nevertheless, similar trends for E_{mass} were found for Mesh A and Mesh B meaning that the error converges to the corresponding level of E_{div}. Figure 20 shows the temporal evolution of E_{div} for the improved method and the method of LP for both meshes. Note that for E_{div} of the improved method only every 20^{th} data point is shown to maintain readability and that different ordinates are used for the improved method and the method of LP. In Figure 20, no systematic difference between the temporal evolution of E_{div} for Mesh A and B can be observed for the improved method. On the other

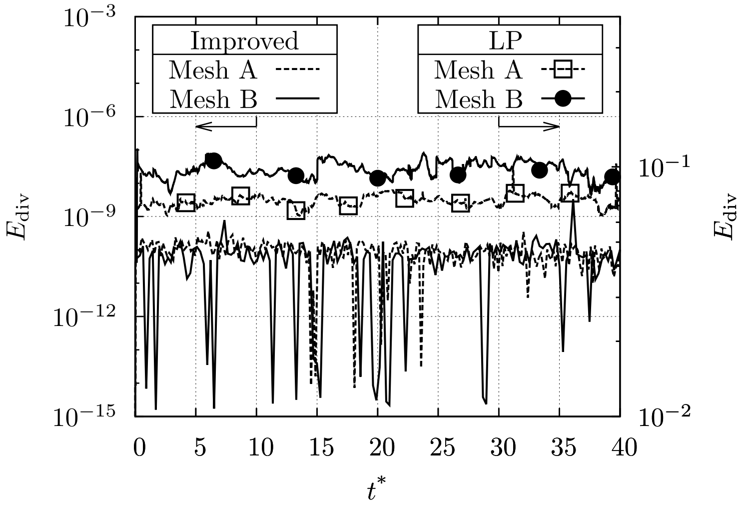


Figure 20: Temporal evolution of the infinity norm of the relative Lagrangian cell compression/expansion E_{div} for the improved method and the method of LP for Mesh A and Mesh B in the considered application case. Note the different ordinates.

hand, a grid dependence is visible for the method of LP. E_{div} for Mesh B is approximately 1.5 times larger than for Mesh A, which indicates a first-order convergence. Further investigations about the grid convergence for the method of LP are not intended and have been performed by LP [30] as the focus of this work is the improvement, which has been demonstrated to be conservative independent of the grid resolution in transport tests (Subsection 6.1) and a complex realistic primary atomization case.

The high mass conservation errors (almost 10% for Mesh A and more than 10% for Mesh B) obtained by the method of LP manifest themselves visibly in the solution and are consequently not suitable for accurate predictions of atomization meaning that the method of LP fails for the considered case. Conversely, this means that the improved method enables the computation of the considered real engineering application case. Of course, this is to be expected for the method of OD as well as both methods are discretely equivalent.

8. Conclusions

In this paper, the 3D unsplit cell-based method of LP is improved by adding discrete conservation. This provides a substantial improvement over earlier methods that led to significant conservation errors, for instance, in simulations of primary atomization in a realistic application. To ensure discrete conservation, a Lagrangian cell correction was introduced. Similar to the method of CS, the correction relies on additional cell vertices to describe the Lagrangian cells. The position of those is obtained from conservative flux volumes. In contrast to the method of CS, a less complex flux volume discretization is used, which is equivalent to the method of OD, making the improved method discretely equivalent to the flux-based method of OD. The performance of the improved scheme has been analyzed using benchmark transport test cases using two vertex velocity approximations that are realizable with typical time advancement schemes of flow solvers. The test cases demonstrate the discrete conservation property and the improvement over the method of LP. Second-order geometric convergence is achieved even on coarse meshes. A computational cost analysis shows that in comparison to the method of OD significantly fewer intersection and volume computations are required meaning that a noticeable cost reduction can be expected. Finally, it has been shown that the improved method enables the conservative computation of a complex realistic primary atomization case in which the method of LP suffers large conservation errors.

Acknowledgments

The authors gratefully acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through SPP1980 (Project number 375857587) and by the European Union's Horizon 2020 research and innovation program under the Center of Excellence in Combustion (CoEC) project (Grant agreement number 952181). Moreover, the authors gratefully acknowledge the computing time granted by the NHR4CES Resource Allocation Board and provided on the supercomputer CLAIX at RWTH Aachen University as part of the NHR4CES infrastructure. The calculations for this research were conducted with computing resources under the project p0020069.

References

[1] M. Gorokhovski, M. Herrmann, Modeling Primary Atomization, Annu. Rev. Fluid Mech. 40 (1) (2008) 343–366. `doi:10.1146/annurev.fluid.40.111406.102200`.

[2] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603. `doi:10.1146/annurev.fluid.31.1.567`.

[3] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. 79 (1) (1988) 12–49. `doi:10.1016/0021-9991(88)90002-2`.

[4] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method), J. Comput. Phys. 152 (2) (1999) 457–492. `doi:10.1006/jcph.1999.6236`.

[5] 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. `doi:10.1016/0021-9991(81)90145-5`.

[6] C. W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (2) (1988) 439–471. `doi:10.1016/0021-9991(88)90177-5`.

[7] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (1) (2005) 225–246. `doi:10.1016/j.jcp.2005.04.007`.

[8] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow II, J. Comput. Phys. 225 (1) (2007) 785–807. `doi:10.1016/j.jcp.2006.12.027`.

[9] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, J. Comput. Phys. 227 (18) (2008) 8395–8416. `doi:10.1016/j.jcp.2008.05.027`.

[10] M. Owkes, O. Desjardins, A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows, J. Comput. Phys. 249 (2013) 275–302. `doi:10.1016/j.jcp.2013.04.036`.

[11] J. O. McCaslin, O. Desjardins, A localized re-initialization equation for the conservative level set method, J. Comput. Phys. 262 (2014) 408–426. `doi:10.1016/j.jcp.2014.01.017`.

[12] T. Marić, D. B. Kothe, D. Bothe, Unstructured un-split geometrical Volume-of-Fluid methods – A review, J. Comput. Phys. 420. `doi:10.1016/j.jcp.2020.109695`.

[13] W. F. Noh, P. Woodward, SLIC (Simple Line Interface Calculation), in: A. I. van de Vooren, P. J. Zandbergen (Eds.), Proc. Fifth Int. Conf. Numer. Methods Fluid Dyn. June 28 – July 2, 1976 Twente Univ. Enschede, Springer Berlin Heidelberg, Berlin, Heidelberg, 1976, pp. 330–340.

[14] R. B. DeBar, Fundamentals of the KRAKEN code, Tech. Rep. (March) (1974) UCID–17366.

[15] D. L. Youngs, Time-dependent multi-material flow with large fluid distortion, in: K. W. Morton, M. J. Baines (Eds.), Numer. Methods Fluid Dyn., Academic Press, 1982, pp. 273–285.

[16] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (3) (1979) 335–362. `doi:10.1016/0021-9991(79)90051-2`.

[17] O. Ubbink, R. I. Issa, A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes, J. Comput. Phys. 153 (1) (1999) 26–50. `doi:10.1006/jcph.1999.6276`.

[18] 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. `doi:10.1002/fld.431`.

[19] E. Aulisa, S. Manservisi, R. Scardovelli, S. Zaleski, A geometrical area-preserving Volume-of-Fluid advection method, J. Comput. Phys. 192 (1) (2003) 355–364. `doi:10.1016/j.jcp.2003.07.003`.

[20] E. Aulisa, S. Manservisi, R. Scardovelli, S. Zaleski, Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry, J. Comput. Phys. 225 (2) (2007) 2301–2319. `doi:10.1016/j.jcp.2007.03.015`.

[21] G. D. Weymouth, D. K. Yue, Conservative Volume-of-Fluid method for free-surface simulations on Cartesian-grids, J. Comput. Phys. 229 (8) (2010) 2853–2865. `doi:10.1016/j.jcp.2009.12.018`.

[22] P. Liovic, M. Rudman, J. L. Liow, D. Lakehal, D. Kothe, A 3D unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction, Comput. Fluids 35 (10) (2006) 1011–1032. `doi:10.1016/j.compfluid.2005.09.003`.

[23] J. Hernández, J. López, P. Gómez, C. Zanzi, F. Faura, A new volume of fluid method in three dimensions - Part I: Multidimensional advection method with face-matched flux polyhedra, Int. J. Numer. Methods Fluids 58 (8) (2008) 897–921. `doi:10.1002/fld.1776`.

[24] T. Marić, H. Marschall, D. Bothe, voFoam - A geometrical Volume of Fluid algorithm on arbitrary unstructured meshes with local dynamic adaptive mesh refinement using OpenFOAM (2013). `arXiv:1305.3417`.

[25] M. Owkes, O. Desjardins, A computational framework for conservative, three-dimensional, unsplit, geometric transport with application to the volume-of-fluid (VOF) method, J. Comput. Phys. 270 (2014) 587–612. `doi:10.1016/j.jcp.2014.04.022`.

[26] L. Jofre, O. Lehmkuhl, J. Castro, A. Oliva, A 3-D Volume-of-Fluid advection method based on cell-vertex velocities for unstructured meshes, Comput. Fluids 94 (2014) 14–29. `doi:10.1016/j.compfluid.2014.02.001`.

[27] V. Le Chenadec, H. Pitsch, A 3D unsplit Forward/Backward Volume-of-Fluid approach and coupling to the level set method, J. Comput. Phys. 233 (1) (2013) 10–33. `doi:10.1016/j.jcp.2012.07.019`.

[28] R. Comminal, J. Spangenberg, Three-dimensional cellwise conservative unsplit geometric VOF schemes, J. Comput. Phys. 442. `doi:10.1016/j.jcp.2021.110479`.

[29] J. López, J. Hernández, P. Gómez, F. Faura, Non-convex analytical and geometrical tools for volume truncation, initialization and conservation enforcement in VOF methods, J. Comput. Phys. 392 (2019) 666–693. `doi:10.1016/j.jcp.2019.04.055`.

[30] V. Le Chenadec, H. Pitsch, A conservative framework for primary atomization computation and application to the study of nozzle and density ratio effects, At. Sprays 23 (12) (2013) 1139–1165. `doi:10.1615/AtomizSpr.2013007243`.

[31] M. Bode, F. Diewald, D. O. Broll, J. F. Heyse, V. Le Chenadec, H. Pitsch, Influence of the injector geometry on primary breakup in diesel injector systems, in: SAE Tech. Pap., Vol. 1, SAE International, 2014. `doi:10.4271/2014-01-1427`.

[32] M. Bode, T. Falkenstein, V. Le Chenadec, S. Kang, H. Pitsch, T. Arima, H. Taniguchi, A New Euler/Lagrange Approach for Multiphase Simulations of a Multi-Hole GDI Injector, in: SAE Tech. Pap., Vol. 2015-April, SAE International, 2015. `doi:10.4271/2015-01-0949`.

[33] M. Bode, T. Falkenstein, M. Davidovic, H. Pitsch, H. Taniguchi, K. Murayama, T. Arima, S. Moon, J. Wang, A. Arioka, Effects of Cavitation and

Hydraulic Flip in 3-Hole GDI Injectors, Int. J. Fuels Lubr. 10 (2) (2017) 380–393. `doi:10.2307/26274120`.

[34] Q. Zhang, On a Family of Unsplit Advection Algorithms for Volume-of-Fluid Methods, SIAM J. Numer. Anal. 51 (5) (2013) 2822–2850. `doi:10.1137/120897882`.

[35] T. Marić, H. Marschall, D. Bothe, An enhanced un-split face-vertex flux-based VoF method, J. Comput. Phys. 371 (2018) 967–993. `doi:10.1016/j.jcp.2018.03.048`.

[36] M. Bode, A. Y. Deshmukh, T. Falkenstein, S. Kang, H. Pitsch, Hybrid scheme for complex flows on staggered grids and application to multiphase flows, J. Comput. Phys. (in press) 0`doi:10.1016/j.jcp.2018.12.041`.

[37] R. Scardovelli, S. Zaleski, Analytical Relations Connecting Linear Interfaces and Volume Fractions in Rectangular Grids, J. Comput. Phys. 164 (1) (2000) 228–237 . `doi:10.1006/jcph.2000.6567`.

[38] S. Diot, M. M. François, An interface reconstruction method based on an analytical formula for 3D arbitrary convex cells, J. Comput. Phys. 305 (2016) 63–74. `doi:10.1016/j.jcp.2015.10.011`.

[39] M. Owkes, O. Desjardins, A mass and momentum conserving unsplit semi-Lagrangian framework for simulating multiphase flows, J. Comput. Phys. 332 (2017) 21–46. `doi:10.1016/j.jcp.2016.11.046`.

[40] S. Bnà, S. Manservisi, R. Scardovelli, P. Yecko, S. Zaleski, Numerical integration of implicit functions for the initialization of the VOF function, Comput. Fluids 113 (2015) 42–52. `doi:10.1016/j.compfluid.2014.04.010`.

[41] S. Bnà, S. Manservisi, R. Scardovelli, P. Yecko, S. Zaleski, Vofi - A library to initialize the volume fraction scalar field, Comput. Phys. Commun. 200 (2016) 291–299. `doi:10.1016/j.cpc.2015.10.026`.

[42] R. J. Leveque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (2) (1996) 627–665. `doi:10.1137/0733033`.

[43] J. E. Pilliod, E. G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2) (2004) 465–502. `doi:10.1016/j.jcp.2003.12.023`.

[44] F. Schneider, S. Suleiman, J. Menser, E. Borukhovich, I. Wlokas, A. Kempf, H. Wiggers, C. Schulz, SpraySyn-A standardized burner configuration for nanoparticle synthesis in spray flames, Rev. Sci. Instrum. 90 (8) (2019) 85108. `doi:10.1063/1.5090232`.

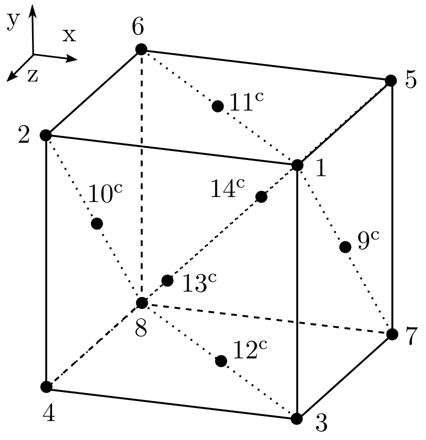
[45] V. Le Chenadec, H. Pitsch, A monotonicity preserving conservative sharp interface flow solver for high density ratio two-phase flows, J. Comput. Phys. 249 (2013) 185–203. `doi:10.1016/j.jcp.2013.04.027`.

[46] S. Popinet, Numerical Models of Surface Tension, Annu. Rev. Fluid Mech. 50 (1) (2018) 49–75. `doi:10.1146/annurev-fluid-122316-045034`.

[47] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, J. Comput. Phys. 228 (16) (2009) 5838–5866. `doi:10.1016/j.jcp.2009.04.042`.

Appendix A. Triangulation connectivities

The used vertices naming convention and triangulation connectivity matrix is provided for the discrete Lagrangian cell in Figure A.21. The presented vertices ordering is chosen such that unflipped tetrahedra have a positive volume. The used tetrahedra vertex connectivity matrix is presented for the generic discrete flux volume (s. Figure 6) in Table A.2 (left) along with the selection of Lagrangian cell and Eulerian cell vertices used for each Eulerian cell face (right). The presented vertex ordering provides positive volumes for tetrahedra that have a positive (negative) flux volume contribution when the tetrahedron corresponds to an Eulerian cell faces with a positive (negative) normal vector.



Nr.	v1	v2	v3	v4
1	7	1	3	9^c
2	7	5	1	9^c
3	8	4	2	10^c
4	8	2	6	10^c
5	6	2	1	11^c
6	6	1	5	11^c
7	8	3	4	12^c
8	8	7	3	12^c
9	4	1	2	13^c
10	4	3	1	13^c
11	8	5	7	14^c
12	8	6	5	14^c
13	8	7	1	3
14	8	7	5	1
15	8	6	1	5
16	8	6	2	1
17	8	4	1	2
18	8	4	3	1

Figure A.21: Vertex naming convention (left) and triangulation connectivity matrix (right) used in this work. The dotted lines in the top left schematic indicate the face triangulation. The horizontal line in the connectivity matrix (right) separates the correction tetrahedra (top) from the estimation tetrahedra (bottom).

Nr.	v1	v2	v3	v4
1	**h**	**d**	**b**	**o**
2	**h**	**b**	**f**	**o**
3	**h**	**g**	**a**	**c**
4	**h**	**g**	**e**	**a**
5	**h**	**f**	**a**	**e**
6	**h**	**f**	**b**	**a**
7	**h**	**d**	**a**	**b**
8	**h**	**d**	**c**	**a**

Face	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**o**
+x	1	1^*	3	3^*	5	5^*	7	7^*	9^c
-x	2	2^*	4	4^*	6	6^*	8	8^*	10^c
+y	1	1^*	5	5^*	2	2^*	6	6^*	11^c
-y	3	3^*	7	7^*	4	4^*	8	8^*	12^c
+z	5	5^*	6	6^*	7	7^*	8	8^*	14^c
-z	1	1^*	2	2^*	3	3^*	4	4^*	13^c

Table A.2: Left: Vertex connectivity matrix for the discrete flux volume. The horizontal line separates the correction tetrahedra from the estimation tetrahedra. Right: Connectivity matrix between the generic flux volume vertices and the selection of Lagrangian cell and Eulerian cell vertices used for the corresponding Eulerian cell face.