

# An unstructured Finite-Volume Level Set / Front Tracking method for capillary flows

Vom Fachbereich Mathematik der Technischen Universität Darmstadt  
Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)  
Genehmigte Dissertation von Tobias Tolle aus Hofgeismar  
Tag der Einreichung: 14. September 2022, Tag der Prüfung: 24. November 2022

Referent: Dr.-Ing. Tomislav Marić  
1. Korreferent: Prof. Dr. rer. nat. Dieter Bothe  
2. Korreferent: Dr.-Ing. Damir Juric  
Darmstadt, Technische Universität Darmstadt, 2022



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Mathematics Department  
Mathematical Modeling and  
Analysis

Lagrangian / Eulerian  
numerical methods for  
multiphase flows

An unstructured Finite-Volume Level Set / Front Tracking method for capillary flows

Accepted doctoral thesis by Tobias Tolle

Date of submission: 14. September 2022

Date of thesis defense: 24. November 2022

Darmstadt, Technische Universität Darmstadt, 2022

Bitte zitieren Sie dieses Dokument als:

URN: urn:nbn:de:tuda-tuprints-233663

URL: <http://tuprints.ulb.tu-darmstadt.de/23366>

Jahr der Veröffentlichung auf TUpriints: 2023

Dieses Dokument wird bereitgestellt von tuprints,

E-Publishing-Service der TU Darmstadt

<http://tuprints.ulb.tu-darmstadt.de>

[tuprints@ulb.tu-darmstadt.de](mailto:tuprints@ulb.tu-darmstadt.de)

Die Veröffentlichung steht unter folgender Creative Commons Lizenz:

Namensnennung – Weitergabe unter gleichen Bedingungen 4.0 International

<https://creativecommons.org/licenses/by-sa/4.0/>

This work is licensed under a Creative Commons License:

Attribution–ShareAlike 4.0 International

<https://creativecommons.org/licenses/by-sa/4.0/>

Meinen Eltern  
*Georg und Beate Tolle*  
und dem  
*Hof Tolle*





---

# Acknowledgements

---

Working on this thesis was in many ways like the ascent of a mountain through an uncharted route. And as such an endeavour, it is only made possible through the aid and support of others. Thus, I want to first express my cordial and earnest gratitude to my thesis supervisors: thank you, Prof. Dr. rer. nat. Dieter Bothe, for giving me the opportunity to work on such a challenging, yet fascinating research topic in your group. Thank you, Dr.-Ing. Tomislav Marić, for your guidance, advise and sheer endless passion for science. I am deeply grateful for our joint efforts to solve the questions and problems encountered over the last years. Every discussion with you was not only a chance to learn something, but also a source of inspiration and motivation.

Further, I would like to thank Dr. Damir Juric for agreeing to be a reviewer of my thesis.

With the Mathematical Modeling and Analysis group, I had a great place to work with great people. Thus, I want to thank my colleagues and former colleagues, especially Anja, Andre, Chiara, Dirk, Johannes and Jun, for many discussions and our joint coffee breaks. I had a great time working with you! Also, thanks to Lilli Jundt-Becker for taking care of all sorts of administrative tasks and reducing exposure to bureaucracy to minimum!

I am also grateful for the scientific exchange with Dr. Jalel Chergui, Prof. Seungwon Shin and Dr. Damir Juric, both remote and in person. From you and your experience, I learned a lot, especially about my research and science, yet also about life.

This research project has been made possible by the funding from the International Research Training Group 1529 and the NFDI4Ing pilot application within the CRC 1194. The provision of computational resources by the Lichtenberg High Performance Cluster at the TU Darmstadt is also gratefully acknowledged.

A wide range of free and open software has been used during the course of this thesis, the Archlinux distribution, the Neovim editor, the GNU Compiler Collection (GCC) and the Git version control system to name a few. Thus, I want to acknowledge these projects in particular, but also the free and open source community in general.

My final thanks go to my friends and my family – my wife Nafthalen, my brother Nils, my grandmother Christa and my parents Beate and Georg: I lack the words to express my esteem and gratitude towards you.



---

# Abstract

---

In this thesis the unstructured Finite-Volume hybrid Level Set / Front Tracking method (LENT) for immiscible two-phase flows is extended to enable the simulation of capillary flows. The major contributions are a more accurate interface curvature approximation, an accuracy driven pressure velocity coupling algorithm, an approximation technique for consistent mass fluxes for momentum convection and two novel approaches for the computation of volume fractions from triangulated surfaces. All proposed techniques and algorithms are devised for unstructured Finite-Volume meshes.

The improved curvature approximation uses a signed distance field as input and utilizes surface-mesh/volume-mesh mappings to reduce curvature variation in interface normal direction. A novel, local correction approach is introduced to further reduce the curvature error in cells intersected by the interface. To ensure a prescribed solution accuracy, an iterative, accuracy driven pressure velocity coupling algorithm is presented that builds on the established segregated solution algorithms. The necessity of consistent mass fluxes for momentum convection in the presence of differing fluid densities is analyzed. For interface advection methods that do not utilize phase-specific volumetric fluxes, a method to obtain approximate, consistent mass fluxes is proposed. The resulting improvements for capillary flows are demonstrated using canonical verification and validation test cases.

Two novel algorithms to compute volume fractions on unstructured volume meshes from oriented triangle surfaces meshes are introduced, one based on geometric intersections and one based on approximation and adaptive refinement. Intended for the phase indicator calculation in the context of Level Set / Front Tracking methods, both algorithms are shown to be sufficiently accurate to initialize volume fractions also for the Volume-of-Fluid method. In fact, test cases demonstrate that both approaches' accuracy is only limited by the resolution of the surface mesh.



---

# Zusammenfassung

---

In dieser Arbeit wird die unstrukturierte Finite-Volumen hybride Level Set / Front Tracking Methode (LENT) für unmischbare Zweiphasen Strömungen erweitert auf die Simulation kapillarer Strömungen. Die wesentlichen Beiträge sind eine genauere Approximation der Krümmung der Grenzfläche, ein genauigkeitsgesteuerter Algorithmus zur Kopplung von Druck und Geschwindigkeit, eine Näherungstechnik für die Berechnung konsistenter Massenflüsse zur Impulskonvektion und zwei neuartige Ansätze zur Berechnung von Volumenfraktionen auf Basis triangulierter Oberflächen. Alle vorgeschlagenen Techniken und Algorithmen sind formuliert für unstrukturierte Finite-Volumen Gitter.

Die verbesserte Krümmungsapproximation nutzt ein vorzeichenbehaftetes Distanzfeld als Eingangparameter und nutzt die Oberflächengitter/Volumengitter Zuordnungen um die Krümmungsänderung entlang der Grenzflächennormale zu reduzieren. Ein neuartiger, lokaler Korrekturansatz wird eingeführt um den Krümmungsfehler weiter zu reduzieren in Zellen, die von der Grenzfläche geschnitten werden. Um eine vorgeschriebene Lösungsgenauigkeit sicherzustellen, wird ein iterativer, genauigkeitsgesteuerter Algorithmus zur Kopplung von Druck und Geschwindigkeit vorgestellt, der auf etablierten segregierten Lösungsalgorithmen aufbaut. Die Notwendigkeit konsistenter Massenflüsse für die Impulskonvektion in der Gegenwart von Fluiden unterschiedlicher Dichte wird analysiert. Für Methoden der Grenzflächenadvektion, die keine phasenspezifischen volumetrischen Flüsse verwenden, wird ein Ansatz vorgeschlagen um genäherte, konsistente Massenflüsse zu erhalten. Die resultierenden Verbesserungen für kapillare Strömungen werden anhand kanonischer Verifikations- und Validierungsfällen gezeigt.

Zwei neuartige Algorithmen zur Berechnung von Volumenfraktionen auf unstrukturierten Volumengittern auf Basis orientierter Dreiecks- und Vierecksoberflächengitter werden vorgestellt, einer basierend auf geometrischen Schneidungen und einer basierend auf Approximation und adaptiver Verfeinerung. Vorgesehen für die Phasenindikatorberechnung im Kontext von hybriden Level Set / Front Tracking Methoden, zeigen beide Algorithmen ausreichend genau zu sein um Volumenfraktionen für die Volume-of-Fluid Methode zu initialisieren. Vielmehr zeigen Testfälle, dass die Genauigkeit beider Ansätze nur durch die Auflösung des Oberflächengitters limitiert wird.



---

# Publications

---

Ideas, figures and text presented in this thesis have appeared previously in the following publications:

## Journal articles

**T. Tolle**, D. Bothe, T. Marić. “SAAMPLE: A segregated accuracy-driven algorithm for multiphase pressure-linked equations.” *Computers & Fluids* 200 (2020): 104450

**T. Tolle**, D. Gründing, D. Bothe, T. Marić. “triSurfaceImmersion: Computing volume fractions and signed distances from triangulated surfaces immersed in unstructured meshes.” *Computer Physics Communications* 273 (2022): 108249

J. Liu, **T. Tolle**, D. Bothe, T. Marić. “A consistent discretization of the single-field two-phase momentum convection term for the unstructured finite volume Level Set / Front Tracking method.” *Journal of Computational Physics – Under Review*

## Conference contributions

“*Extending a hybrid Level Set / Front Tracking method for the simulation of surface tension driven flows*”, The 13th Japanese-German International Workshop on Mathematical Fluid Dynamics, Darmstadt, Germany, 30. November – 2. December, 2016

“*Numerical modelling of jumps in two-phase flows in the context of one-field formulated Navier-Stokes equations*”, The 14th Japanese-German International Workshop on Mathematical Fluid Dynamics, Waseda University, Tokyo, Japan, 8. – 10. March, 2017

---

---

*“Extending a hybrid Level Set / Front Tracking method for the simulation of surface tension driven flows”*, The 3rd International Conference on Numerical Methods in Multiphase Flows, University of Tokyo, Tokyo, Japan, 26. – 29. June, 2017

*“Capillary accuracy of a hybrid Level Set / Front Tracking method on unstructured meshes”*, Workshop on Mathematical Fluid Dynamics, Bad Boll, Germany, 7. – 11. May, 2018

*“Initialization of volume fractions and signed distances on unstructured meshes from triangulated surfaces”*, Jahrestreffen der ProcessNet-Fachgruppen Mehrphasenströmung und Computational Fluid Dynamics, Online, 9. – 10. March, 2021

*“Initialization of volume fractions and signed distances on unstructured meshes from triangulated surfaces”*, 16th OpenFOAM Workshop, Online, 8. – 11. June, 2021

*“A collocated finite volume unstructured Level Set / Front Tracking method for high density ratios”*, 5th International Conference on Droplets, Online, 16. – 18. August, 2021

*“Initialization of volume fractions and signed distances on unstructured meshes from triangulated surfaces”*, 5th International Conference on Droplets, Online, 16. – 18. August, 2021

*“A collocated unstructured finite volume Level Set / Front Tracking method for two-phase flows with large density ratios”*, 18th Multiphase Flow Conference and Short Course, Online, 8. – 12. November, 2021



---

# Contents

---

<b>1. Introduction</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Thesis outline . . . . .	2
<b>2. Mathematical model</b>	<b>3</b>
<b>3. Numerical method</b>	<b>7</b>
3.1. Domain discretization . . . . .	7
3.2. Unstructured Finite-Volume method . . . . .	9
3.2.1. Time derivative . . . . .	9
3.2.2. Convective term . . . . .	10
3.2.3. Diffusion term . . . . .	11
3.2.4. Source terms . . . . .	11
3.2.5. Interpolation . . . . .	12
3.3. Volume fraction and mixture of fluid properties . . . . .	12
<b>4. Computation of signed distances and volume fractions from surface meshes</b>	<b>15</b>
4.1. Literature review . . . . .	15
4.2. Algorithm description . . . . .	21
4.2.1. Octree-based search structure . . . . .	22
4.2.2. Octree decomposition of the surface mesh . . . . .	23
4.2.3. Signed distance computation . . . . .	24
4.2.4. Inside / outside propagation . . . . .	25
4.2.5. Volume fractions by geometrical intersection . . . . .	27
4.2.6. Volume fractions by adaptivity and approximation . . . . .	28
4.3. Numerical results . . . . .	34
4.3.1. Triangulated surfaces . . . . .	35
4.3.2. Implicit surfaces / Level Set surfaces . . . . .	44
4.4. Conclusion . . . . .	47

<b>5. Hybrid Level Set / Front Tracking method</b>	<b>49</b>
5.1. Literature review . . . . .	49
5.2. Method overview . . . . .	51
5.3. Interface evolution . . . . .	54
5.4. Signed distance calculation . . . . .	54
5.5. Phase indicator approximation . . . . .	55
5.5.1. Approximation of area fractions for cell faces . . . . .	55
5.6. Curvature approximation . . . . .	55
5.6.1. Compact curvature calculation . . . . .	57
5.6.2. Spherical curvature correction . . . . .	58
5.7. Surface tension computation . . . . .	59
5.8. Pressure velocity coupling . . . . .	61
5.9. Verification and validation results . . . . .	64
5.9.1. Curvature calculation . . . . .	66
5.9.2. Surface tension force reconstruction . . . . .	71
5.9.3. Stationary droplet . . . . .	72
5.9.4. Translating droplet . . . . .	77
5.9.5. Oscillating droplet . . . . .	79
5.10. Conclusions . . . . .	85
<b>6. Mass flux consistency and high density ratios</b>	<b>87</b>
6.1. Literature overview . . . . .	87
6.2. A solution algorithm for two-phase flows with high density ratios using the collocated unstructured Finite-Volume method . . . . .	94
6.2.1. Numerical consistency of the single-field conservative two-phase momentum convection term . . . . .	94
6.2.2. A semi-implicit solution algorithm for high density ratios . . . . .	97
6.3. Verification and validation . . . . .	104
6.3.1. Time step size . . . . .	104
6.3.2. Translating droplet . . . . .	105
6.4. Conclusions . . . . .	110
<b>7. Summary and outlook</b>	<b>113</b>
<b>Bibliography</b>	<b>115</b>
<b>A. Additional figures</b>	<b>131</b>
A.1. Parameter study figures with legends . . . . .	131

---

## List of Figures

---

2.1. Schematic of a two-phase system domain $\Omega$ subdivided into the phase specific domains $\Omega^-$ and $\Omega^+$ , separated by the interface $\Sigma$ . . . . .	4
3.1. Exemplary cell $\Omega_c$ and quantities required by the finite volume method. . . . .	8
4.1. Calculating volume fractions of a circular interface by numerical integration. . . . .	16
4.2. Polyhedral cell (left) and non-convex cell (right) for which the intersection volume (dark grey) has to be computed. The light grey regions lead to cases that have to be identified and require special treatment increasing the problem complexity far beyond a simple one dimensional integration. . . . .	17
4.3. Calculating volume fractions from a circular interface by volume mesh intersection. . . . .	19
4.4. Steps of the Surface Mesh Intersection / Approximation (SMCI/A) algorithms. . . . .	22
4.5. Centroid decomposition of an interface cell into tetrahedra and calculation of $\alpha_c$ using the Surface-Mesh Cell Intersection/Approximation (SMCI/A) algorithms. . . . .	28
4.6. Steps of the SMCA algorithm following signed distance computation and inside/outside propagation. . . . .	30
4.7. Illustration of the idea of the bounding ball criterion in 2D for clarity. The solid grey line represents $\mathcal{B}(\mathbf{x}_c,  \phi_c )$ , the grey dashed one $\mathcal{B}(\mathbf{x}_c, r_{bb})$ . . . . .	31
4.8. Decomposition of a tetrahedron into eight tetrahedra using edge midpoints. . . . .	32
4.9. $E_v$ errors of the Surface-Mesh / Cell Intersection (SMCI) algorithm for the sphere. The grey dashed line indicates second order convergence. . . . .	36
4.10. CPU times of the SMCI algorithm for the sphere initialized on a cubic unstructured mesh. . . . .	37
4.11. SMCI algorithm used with a sphere and an ellipsoid on an unstructured hexahedral mesh. . . . .	37
4.12. $E_v$ errors of the SMCI algorithm for the ellipsoid. The grey dashed line indicates second order convergence. . . . .	38

---

4.13. $E_v$ errors of the Surface-Mesh / Cell Approximation (SMCA) algorithm using different refinement levels $l_{\max}$ for a sphere. Resolution of volume and surface mesh are fixed to $N_c = 16$ and $\sqrt{N_T} \approx 410$ . The grey dashed line indicates second order convergence. . . . .	39
4.14. Tetrahedral decomposition of a perturbed hex cell used to approximate $\alpha_c$ . Tetrahedra from different refinement levels are shown in different colors (level 1: blue, level 2: grey, level 3: red). Due to adaptivity, the highest refinement level is localized in the vicinity of the surface mesh.. . . . .	40
4.15. $E_v$ errors of the SMCA algorithm for the sphere. The grey dashed line indicates second order convergence. . . . .	41
4.16. $E_v$ errors of the SMCA algorithm for the ellipsoid. The grey dashed line indicates second order convergence. . . . .	41
4.17. CPU times of the SMCA algorithm for the sphere initialized on a cubic unstructured mesh. . . . .	42
4.18. Simulation of the wetting experiment with the fluid interface given as a triangular surface mesh [44]. . . . .	42
4.19. Surface and volume mesh of the Computer-Aided Design (CAD) model test case. . . . .	43
4.20. Inside/outside computation and resulting volume fractions for the CAD geometry. . . . .	44
4.21. $E_v$ errors of the SMCA algorithm using different refinement levels $l_{\max}$ for the CAD model with the reference volume $V_e$ computed by eq. (4.28). The grey dashed line indicates second order convergence. . . . .	45
4.22. Comparing the SMCA algorithm and Jones et al. [55, table 3] on tetrahedral meshes. . . . .	46
5.1. Two-phase flow domain discretization. . . . .	51
5.2. Flowchart of the Level Set / Front Tracking (LENT)-SAAMPLE method. . . . .	53
5.3. Curvature of a sphere with $R = 1$ , evaluated in the normal direction to the sphere, as a function of the signed distance from the sphere $\phi$ using eq. (5.5). Obviously, the exact value $\kappa = 2$ can only be obtained on the sphere, where $\phi = 0$ . . . . .	56
5.4. Compact curvature calculation. . . . .	57
5.5. The cell-face connectivity of a regular hexagonal cell. . . . .	60
5.6. Crosssection of statically refined 3D meshes for hydrodynamic test cases with initial interface. The equivalent resolutions from left to right are $n_e = 64$ , $n_e = 64$ and $n_e = 256$ . . . . .	66

---

5.7. $L_2$ norm of the relative curvature error for a sphere (upper row) and an ellipsoid (lower row) for different curvature models. The exact signed distance (left column) and the signed distance computed from the front (right column) have been used as input. A cubic domain $\Omega : [0, 0, 0] \times [4, 4, 4]$ with $n$ cells in each spatial direction. Each setup has been simulated 20 times by setting the centroid of the interface at a random position around the domain center. The dashed/dotted line indicates first/second order of convergence. . . . .	68
5.8. Convergence of the flux reconstruction. . . . .	71
5.9. Error distribution for the Hadamard-Rybczynsky flow. . . . .	72
5.10. Temporal evolution of the spurious currents for the case of a stationary droplet for different Laplace numbers. The LENT method is configured as in [71] using the Pressure Implicit with Splitting of Operators (PISO) algorithm and exact curvature. On the left, the results for an equivalent resolution of $n_e = 16$ are displayed while the right graph has been computed with an equivalent resolution of $n_e = 64$ . . . . .	74
5.11. Front for the stationary droplet at different times for $La = 120$ . From left to right: initial front (identical for old and new configuration), front of the old configuration at $t = 20$ , front of the current configuration at $t = 20$ and front of the current configuration at $t = 40$ . . . . .	76
5.12. Evolution of spurious currents for the stationary droplet when using the LENT method. The left column shows the results obtained with the configuration from [71], the right column for the current configuration. In the upper row results for a resolution of $n_e = 16$ are displayed, in the lower row for $n_e = 64$ . Each plot shows the results for different Laplace numbers. . . . .	78
5.13. Evolution of spurious currents for the translating droplet when using the LENT method. The left column shows the results obtained with the configuration from [71], the right column for the current configuration. In the upper row results for a resolution of $n_e = 16$ are displayed, in the lower row for $n_e = 64$ . Each plot shows the results for different Laplace numbers. . . . .	80
5.14. Temporal evolution of the $x$ semi-axis $s_x$ for the oscillating droplet. Time is non-dimensionalized with the analytical period $T_2 = 2\pi/\omega_2$ . The dashed lines represent the exact envelope of $s_x$ ( $R_0 + a_2(t)$ and $R_0 - a_2(t)$ , see eq. (5.36)). Each plot shows the results for three mesh resolutions $n_e$ . The case has been simulated using the LENT configuration from [71] (left column) and the current configuration (right column) with different kinematic viscosities of the droplet (upper and lower row). . . . .	82

---

---



---

5.15. Temporal evolution of the $x$ semi-axis $s_x$ for the oscillating droplet replicating the experimental setup described in [125]. The gray area depicts the decay envelope (see eq. (5.36)) according to the experimentally measured $\gamma$ , while the black dots mark measured oscillation peaks. Each plot shows the results for three different mesh resolutions $n_e$ . The left column shows the results using the LENT configuration from [71], the right column the results obtained with the current configuration. Two semi-axes ratios $L/W$ have been simulated. . . . .	84
6.1. A two-phase fixed control volume $\Omega_c$ separated by the interface $\Sigma(t)$ . . . .	97
6.2. Updating the face-centered (mass flux) density in the $\rho$ LENT method. . . .	100
6.3. Computing area fractions from signed distances in the method. . . . .	102
6.4. Translating droplet case setup. . . . .	105
6.5. Half section of mesh $N = 64$ , droplet at initial position. . . . .	106
6.6. Temporal evolution of velocity error norm $L_\infty(\mathbf{v})$ : the left figure depicts the results from old inconsistent Segregated Accuracy-driven Algorithm for Multiphase Pressure-Linked Equations (SAAMPLE) algorithm, the right shows the results from new consistent method. . . . .	107
6.7. Comparison of the strong interface deformation with the inconsistent LENT method (left) and the numerically consistent interface shape of the $\rho$ LENT method. Parameters: $N = 64$ , $\rho^-/\rho^+ = 10^4$ , $t = 0.0008s$ . . . . .	108
6.8. Temporal evolution of velocity error norm $L_\infty(\mathbf{v})$ for the viscous flow with surface tension forces: the left diagram depicts the results from the old inconsistent method, and the right diagram contains the results from the $\rho$ LENT method. The legends of these diagrams are large, and the full information is available in Appendix A: fig. A.1 for fig. 6.8a, fig. A.2 for fig. 6.8b. . . . .	108
6.9. Temporal evolution of velocity error norm $L_\infty(\mathbf{v})$ with pure advection: $\rho$ LENT method used in simulating two-phase flows with different density ratios, mesh resolution: $N = 16, 32, 64$ . . . . .	112
A.1. Full figure of fig. 6.8a . . . . .	132
A.2. Full figure of fig. 6.8b . . . . .	133

---

## List of Tables

---

4.1. Used computing architecture. . . . .	35
5.1. Acronyms of the tested curvature model configurations. Compact calculation refers to the modification described in section 5.6.1 and spherical correction to eq. (5.14). . . . .	67
5.2. Mean error norms ( $\overline{L_\infty}$ and $\overline{L_2}$ ) of the curvature and their standard deviation $\sigma$ for the DG( $\alpha$ ) and sccDG( $\phi$ ) model using an exact signed distance. Between resolutions, the order of convergence is displayed. . . . .	69
5.3. Mean error norms ( $\overline{L_\infty}$ and $\overline{L_2}$ ) of the curvature and their standard deviation $\sigma$ for the DG( $\alpha$ ) and sccDG( $\phi$ ) model using a signed distance computed from the front. Between resolutions, the order of convergence is displayed. . . . .	70
5.4. Number of pressure-correction iterations required to obtain $L_\infty( \mathbf{v} ) < 1e-13$ within the first time step for the stationary droplet with exact curvature. Results are shown for different Laplace numbers and two mesh resolutions. . . . .	74
5.5. Spurious currents of the stationary droplet using the PISO and SAAMPLE algorithm with exact curvature. Results are shown for different Laplace numbers $La$ and mesh resolutions $n_e$ . Magnitude of the spurious currents is given after one time step and at the end of the simulations. . . . .	75
6.1. Realistic fluid properties are combined into four tests: water droplet/air ambient, mercury droplet/air ambient, silicone oil droplet/air ambient, silicone oil droplet/water ambient. . . . .	109
6.2. Execution time for the $\rho$ LENT method. . . . .	110





---

## Nomenclature

---

$\Omega_c$	Mesh cell / a finite volume	
$\delta_0$	One-dimensional Dirac distribution	
$\Omega$	Simulation domain	
$\Omega^+$	Subset of $\Omega$ occupied by phase +	
$\Omega^-$	Subset of $\Omega$ occupied by phase -	
$\mu$	Dynamic viscosity	$\text{kg m}^{-1} \text{s}^{-1}$
$\chi$	Indicator function	
$\Sigma$	Interface separating $\Omega^+$ and $\Omega^-$	
$\delta_\Sigma$	Interface Dirac distribution	
$\nu$	Kinematic viscosity	$\text{m}^2 \text{s}^{-1}$
$\lambda_s$	search radius factor	
$\epsilon_c^R$	Reconstruction error	
$\kappa$	Interface curvature	
$\alpha_f$	Wetted area fraction of a cell face	
$\alpha_c$	Volume fraction	
$\rho$	Fluid density	$\text{kg m}^{-3}$
$\sigma$	Surface tension coefficient	$\text{N m}^{-1}$

---

$\tilde{\Omega}$	Discretized domain	
$\tilde{\Sigma}$	Triangle surface mesh approximating $\Sigma$	
$\phi$	Signed distance to the interface	
$\mathcal{B}$	Closed ball	
$\mathbf{d}_{c,n}$	Vector connecting two cell centres straddling a face	
$\Delta t$	Time step size	
$\mathbf{f}_{\Sigma}$	Surface tension force	
$\mathbf{g}$	Gravitational acceleration	$\text{m s}^{-2}$
$\mathbf{H}$	Operator used in pressure-velocity coupling	
$\dot{m}_f$	Mass flux through a mesh face	
$\mathbf{S}_f$	Face area normal	
$\mathcal{M}_c$	Map associating a triangle index with the closest cell centre.	
$F_h$	Set of mesh faces	
$P_h$	Set of cell corner points	
$\mathbf{n}_{\Sigma}$	Unit normal vector of interface $\Sigma$	
$\mathbf{n}_f$	Face unit normal	
$\mathcal{N}(\tilde{\Sigma})$	Narrow band around $\tilde{\Sigma}$	
$p$	Pressure	$\text{N m}^{-2}$
$\mathbf{T}$	Fluid stress tensor	
$T$	Tetrahedron	
$\text{tol}_{\text{abs}}$	Absolute tolerance	
$\text{tol}_{\text{ls}}$	Linear solver tolerance	

---

---

$\text{tol}_{\text{rel}}$	Relative tolerance	
$\mathcal{T}$	Triangle	
$\mathcal{M}_{\mathcal{T}}$	Map associating a cell index with the closest triangle.	
$\mathbf{v}$	Fluid velocity	$\text{m s}^{-1}$
$\mathbf{f}_v$	Volume forces	$\text{N m}^{-3}$
$\mathbf{x}_c$	Cell centroid	
$\mathbf{x}_f$	Face centroid	
$a$	Speed of sound of a fluid	$\text{m s}^{-1}$
$C_c$	Set of face indices bounding a a cell	
$Ca$	Capillary number	
$E_v$	Relative global volume error	
$L_2$	Two norm	
$L_\infty$	Maximum norm	
$La$	Laplace number	
$r$	Radius	
$S_f$	Cell face	
$V_e$	Reference (exact) volume	
$\partial_t$	Differentiation with respect to time	
$\Delta$	Laplace operator	
$\Delta_\Sigma$	Laplace-Beltrami operator	



---

# 1. Introduction

---

## 1.1. Motivation

A vast array of technical applications involves flows of fluids that do not mix, driven by surface tension forces in geometrically complex domains. Two very timely and relevant examples are the Lab-On-a-Chip (LOC) devices used for fast disease detection [92] and the simulation of the water management problem in Proton-Exchange Membrane (PEM) fuel cells [6]. These applications are geometrically complex, and their development is almost exclusively experimental. Therefore, they would significantly benefit from predictive numerical simulations in two ways. First, predictive simulations would allow to observe physical quantities resolved in time and space, complementing experimental measurements and thereby aiding to better understand the processes of an application. Second, numerical simulations can speed up design cycles through faster evaluation of potential designs and design space exploration using automated parameter studies.

Still, fundamental research of numerical methods is required to make the simulations in such complex technical systems predictive. First, a precise, robust, and computationally efficient tracking of fluid interfaces is necessary. Fluid interfaces deform freely and potentially significantly, merge with, and separate from each other. Second, a numerically stable and accurate solution of two-phase Navier-Stokes equations with high density ratios typically encountered in technical applications is needed. Third, the developed numerical methods must be robust and computationally efficient on high-performance computers.

The amount of scientific literature dedicated to these objectives confirms that they are among the most actively researched Computational Fluid Dynamics topics [136]. Numerical methods that utilize both a Lagrangian (co-moving) reference frame and the Eulerian (fixed) reference frame - so-called Lagrangian / Eulerian methods - show great promise with evolving fluid interfaces.

Efficient workflows are crucial when designing solutions for technical problems: engineers who set up and simulate such problems must be able to do so quickly and automatize them. An unstructured domain discretization (unstructured mesh) significantly simplifies the simulation workflow. Additionally, unstructured meshes support local dynamic Adaptive Mesh Refinement (AMR), enabling a very high degree of accuracy and computational

---

---

efficiency. Only a handful of 3D unstructured Lagrangian / Eulerian methods for simulating surface-tension driven incompressible two-phase flows are available [69]. A stable and accurate simulation of surface-tension driven flow in arbitrary geometrically complex domains is still out of reach [94].

The common methods for the simulation of two-phase flows, e.g. the Volume-of-Fluid (VOF) method [46, 69], the Level Set method [118, 84, 41] and Front Tracking method [129, 126], feature characteristic strengths and weaknesses. Consequently, hybrid methods have been proposed in an attempt to overcome each individual method's shortcomings. Examples for such hybrid methods are the Coupled Level Set / Volume-of-Fluid (CLSVOF) method [117] and the Level Contour Reconstruction Method (LCRM) [105, 107, 109, 106], combining Level Set and Front Tracking techniques. The latter is also an example for an approach that combines Lagrangian (Front Tracking) and Eulerian (Level Set) elements, demonstrating impressive results [106]. Inspired by the capabilities of LCRM, the conceptual feasibility of this approach on unstructured meshes has been demonstrated by Marić et al. [71], showing promising results. Therefore, this thesis aims to extend the Level Set / Front Tracking (LENT) method described in [71] for capillary flows, with the applicability to technical applications in mind.

## 1.2. Thesis outline

This thesis is structured as follows. In chapter 2, the underlying mathematical model based on continuum mechanics along with assumptions about the considered two-phase flow regimes is introduced and summarized in the governing partial differential equations (PDEs). Afterwards, chapter 3 gives a short overview of the employed numerical method, concerning domain discretization and discretization of differential operators. Following the state-of-the-art fundamentals, chapter 4 proposes a novel method for the computation of signed distances and volume fractions from triangulated surfaces. In chapter 5, improvements to the LENT method concerning approximation of interface curvature and solution of the pressure velocity system are presented. This is complemented by a discussion about the role of consistent mass fluxes for momentum convection and a novel method to compute consistent mass fluxes for methods without phase-specific fluxes in chapter 6. The thesis closes with a summary and an outlook in chapter 7.

---

## 2. Mathematical model

---

Starting from the governing conservation principles of continuum mechanics, additional assumptions regarding the flow characteristics for a single fluid and their consequences for the mathematical model are stated. Subsequently, this model is extended to two immiscible fluids.

The dynamics of a fluid obey the conservation laws of continuum mechanics, namely of mass, momentum and energy. In general, a balance equation for each of these three quantities is required. However, for the flows considered in this work, the model does not require an energy equation which is clarified below. In conservative form, mass conservation is expressed by

$$\partial_t(\rho) + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (2.1)$$

and momentum conservation by

$$\partial_t(\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) = \nabla \cdot \mathbf{T} + \mathbf{f}_v \quad (2.2)$$

where  $\rho$  denotes the fluid's density,  $\mathbf{v}$  its velocity,  $\mathbf{T}$  the stress tensor and  $\mathbf{f}_v$  volume forces. The subscript  $t$  denotes differentiation with respect to time. Furthermore, a constitutive relation for the stress tensor  $\mathbf{T}$  is required. This relation is fluid dependent and here only *Newtonian* fluids are considered. Many fluids of technical relevance, e.g. water, air and several oils, belong to the group of Newtonian fluids. For these fluids, the stress tensor is given as

$$\mathbf{T} = (-p + \lambda^* \nabla \cdot \mathbf{v}) \mathbf{I} + \mu (\nabla \mathbf{v} + (\nabla \mathbf{v})^T) \quad (2.3)$$

with the pressure  $p$ , and the fluid specific scalar functions  $\lambda^*$ ,  $\mu$  – dynamic viscosity – depending on the thermodynamic state of the fluid.

While the equations introduced above hold for a wider range of flows, the flows considered within this thesis can be assumed *isothermal*. Furthermore, the fluids are assumed to be *incompressible*. While this assumption is very accurately satisfied for the liquid phase, incompressibility can also be safely assumed for the gas phase in case the Mach number

$$\text{Ma} = \frac{|\mathbf{v}|}{a}, \quad (2.4)$$

with  $a$  denoting the fluid's speed of sound, is significantly smaller than 1. With the speed of sound in air  $a_{\text{air}} = 343 \frac{\text{m}}{\text{s}}$  at standard conditions [45], this assumption will be satisfied for the kind of flows considered in this thesis. Consequently, the material properties  $\rho, \lambda^*, \mu$  reduce from functions to scalar constants. Mass conservation eq. (2.1) reduces to volume conservation

$$\nabla \cdot \mathbf{v} = 0. \quad (2.5)$$

Hence, the term  $\lambda^* \nabla \cdot \mathbf{v}$  vanishes and inserting the remainder of eq. (2.3) into eq. (2.2) yields

$$\partial_t(\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) = -\nabla p + \nabla \cdot \left( \mu \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right) \right) + \mathbf{f}_v. \quad (2.6)$$

Equations 2.5 and 2.6 are known as the *Navier-Stokes* equations for an incompressible, single-phase system. However, they are also valid for each phase of a two-phase system. Such a system is formed if both phases are *immiscible* or only partially miscible on a molecular level. A schematic of a two-phase system is displayed in fig. 2.1. The overall simulation domain  $\Omega \subset \mathbb{R}^3$  is separated into two subdomains  $\Omega = \Omega^+(t) \cup \Omega^-(t)$ , each representing a phase as illustrated for a liquid drop on a surface in fig. 2.1. At the contact line  $\Gamma := \partial\Omega \cap \overline{\Omega^+} \cap \overline{\Omega^-}$ , the liquid-gas interface  $\Sigma$  encloses a contact angle  $\theta$  with the solid surface  $\partial\Omega_{\text{wall}}$ . Furthermore, the normal vector  $\mathbf{n}_\Sigma$  of the interface  $\Sigma$  is oriented such that it points from  $\Omega^-$  into  $\Omega^+$ . Typically, a continuum mechanical model is used

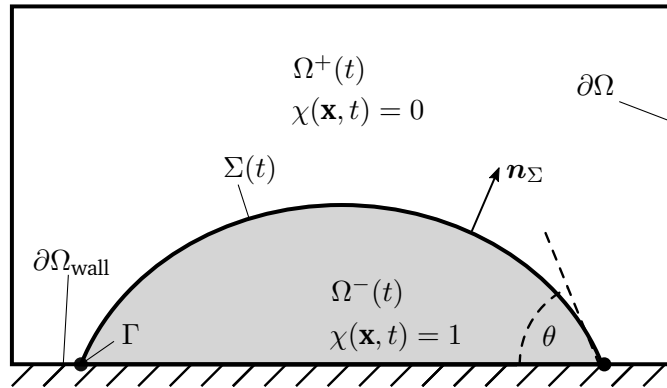


Figure 2.1.: Schematic of a two-phase system domain  $\Omega$  subdivided into the phase specific domains  $\Omega^-$  and  $\Omega^+$ , separated by the interface  $\Sigma$ .

for the description of such a fluid mechanical problem. This description is often based on a sharp interface model, as depicted in fig. 2.1, meaning  $\Sigma$  has zero thickness in the



direction of  $\mathbf{n}_\Sigma$ . With this model, the liquid-gas interface can be described using an indicator function

$$\chi(\mathbf{x}, t) := \begin{cases} 1, & \mathbf{x} \in \Omega^-(t) \subset \mathbb{R}^3, \\ 0, & \text{otherwise.} \end{cases} \quad (2.7)$$

Using  $\chi(\mathbf{x}, t)$ , the phase-wise constant fluid properties can be expressed as a single field for the entire domain by

$$\rho(\mathbf{x}, t) = \chi(\mathbf{x}, t)\rho^- + (1 - \chi(\mathbf{x}, t))\rho^+, \quad (2.8)$$

$$\mu(\mathbf{x}, t) = \chi(\mathbf{x}, t)\mu^- + (1 - \chi(\mathbf{x}, t))\mu^+. \quad (2.9)$$

As in general  $\rho^- \neq \rho^+$  and  $\mu^- \neq \mu^+$ , the single fields  $\rho(\mathbf{x}, t), \mu(\mathbf{x}, t)$  exhibit discontinuities at the interface  $\Sigma$ .

A phenomenon characteristic for two-phase flows is *surface tension*, a force acting on the interface  $\Sigma$ . The surface tension force  $\mathbf{F}_\Sigma$  is given by

$$\mathbf{F}_\Sigma = \sigma\kappa\mathbf{n}_\Sigma + \nabla_\Sigma\sigma \quad (2.10)$$

where  $\kappa = \nabla_\Sigma \cdot (-\mathbf{n}_\Sigma)$  denotes the interface curvature – the sum of the two principle curvatures or twice the mean curvature –,  $\nabla_\Sigma$  the surface gradient and  $\sigma$  the surface tension coefficient [127]. In general,  $\sigma$  depends on the fluid combination and temperature. Furthermore, fluids may contain so-called *surfactants* which accumulate at the interface and thereby influence  $\sigma$  (see e.g. [91]). However, in the scope of this thesis only *pure* fluids, devoid of surfactants or other contaminations, are considered. Taking into account the assumption of isothermal flows as stated above,  $\sigma$  reduces to a scalar constant determined solely by the fluid combination of the two-phase system. Consequently, the term  $\nabla_\Sigma\sigma$  in eq. (2.10) vanishes. The remainder of eq. (2.10) can be reformulated as a volumetric term using an *interface Dirac distribution*

$$\delta_\Sigma(\mathbf{x}, t) = \delta_0((\mathbf{x} - \mathbf{x}_\Sigma) \cdot \mathbf{n}_\Sigma) \quad (2.11)$$

yielding

$$\mathbf{f}_\Sigma = \sigma\kappa\mathbf{n}_\Sigma\delta_\Sigma \quad (2.12)$$

where  $\delta_0$  denotes the one-dimensional Dirac distribution. Adding this term to eq. (2.6) and replacing the generic volume force term with gravity  $\mathbf{f}_v = \rho\mathbf{g}$  gives the mathematical model for flows considered within this thesis:

$$\begin{aligned} \partial_t(\rho\mathbf{v}) + \nabla \cdot (\rho\mathbf{v} \otimes \mathbf{v}) - \nabla \cdot \left( \mu \left( \nabla\mathbf{v} + (\nabla\mathbf{v})^T \right) \right) &= -\nabla p + \rho\mathbf{g} + \mathbf{f}_\Sigma, \\ \nabla \cdot \mathbf{v} &= 0 \end{aligned} \quad (2.13)$$

---

---

with volume conservation eq. (2.5) repeated for completeness. The gravity term  $\rho\mathbf{g}$  is included for completeness only. In the verification and validation cases presented in chapter 5 and chapter 6, gravitational forces are absent. Thus, handling of  $\rho\mathbf{g}$  is not further discussed.

---

## 3. Numerical method

---

As pointed out in chapter 1, the two-phase method presented in this thesis has been developed with technical applications in mind. Consequently, the employed numerical method has to be able to handle the complex domains often encountered in such cases. Therefore, the unstructured Finite-Volume method (FVM) is used for the discretization of the governing partial differential equations 2.13 and 2.5. As this method is already well established, only a brief overview of the main principles and properties is presented here. For further information, the reader is referred to the available literature, e.g. to [34] for a theoretical description of the method or [52, 56] for error analysis. As this thesis builds upon the OpenFOAM framework [134, 82] for finite volume discretization, another noteworthy reference is [77] for a detailed description for FVM as it is implemented in OpenFOAM.

First, the domain discretization and its relevant geometric quantities are introduced. Afterwards, discretization of the different terms appearing in eq. (2.13) is discussed. The chapter closes with some two-phase specific characteristics.

### 3.1. Domain discretization

An approximate solution of the model eq. (2.13) requires a decomposition of the solution domain into volumes that have no volume overlaps, the closed *cells*  $\Omega_c$ , denoted by

$$\Omega \approx \tilde{\Omega} = \{\Omega_c\}_{c \in C} \quad (3.1)$$

where  $C = \{1, 2, 3, \dots, N_c\}$  is a set of indices to mesh cells. The mesh is a set of non-overlapping subsets (*cells*)  $\Omega_c \subset \tilde{\Omega}$ . With non-overlapping, it is meant the volume of an intersection between any two cells is zero. *Index sets* represent the unstructured mesh data [39]. A set of cell corner-points  $P_h$  is considered where each point in  $P_h$  is an element of  $\mathbb{R}^3$ . Geometrically, each cell  $\Omega_c$  is a volume bounded by polygons, so-called *faces*. A global set of faces  $F_h$  is defined, and each face is a sequence of *indices* of points in  $P_h$ . In this context, a cell set  $C_c$  is defined as a set of indices of faces in the set of mesh faces  $F_h$ . Therefore, when referring to a volume defined by the cell,  $\Omega_c$  is used and its magnitude

is then  $|\Omega_c|$ , and when referring to the cell as an unordered index set,  $C_c$  is used and its magnitude  $|C_c|$  is the number of faces that bound the cell.

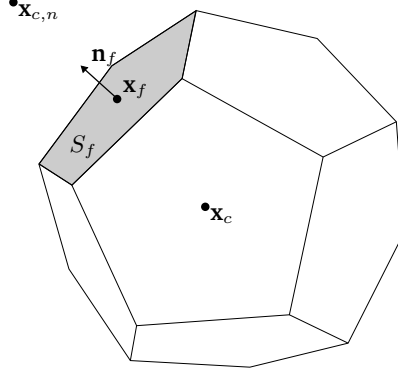


Figure 3.1.: Exemplary cell  $\Omega_c$  and quantities required by the finite volume method.

Figure 3.1 shows an exemplary polyhedral cell  $\Omega_c$  with geometric quantities required for discretization using the Finite-Volume Method. Its centroid  $\mathbf{x}_c$  is defined as

$$\mathbf{x}_c = \frac{1}{|\Omega_c|} \int_{\Omega_c} \mathbf{x} dV. \quad (3.2)$$

Analogue, for each of the cell's bounding faces  $S_f$ , a centroid

$$\mathbf{x}_f = \frac{1}{|S_f|} \int_{S_f} \mathbf{x} dS \quad (3.3)$$

is given with  $|S_f|$  denoting the face area. Assuming an ideal mesh, the polygon  $S_f$  is planar<sup>1</sup> and thus has a constant, unique unit normal  $\mathbf{n}_f$ . It is oriented such that  $(\mathbf{x}_f - \mathbf{x}_c) \cdot \mathbf{n}_f > 0$ . For conciseness, a face area-normal vector  $\mathbf{S}_f = |S_f| \mathbf{n}_f$  is defined. Finally,  $\mathbf{d}_{cn} = \mathbf{x}_{c,n} - \mathbf{x}_c$  denotes the connection between the centroid of  $\Omega_c$  and a neighbor cell  $\Omega_{c,n}$  sharing the face  $S_f$  with  $\Omega_c$ .

Aside from their geometric flexibility, the solution of continuum mechanical problems in geometrically complex domains may significantly benefit from unstructured meshes in additional ways. For example, gradients of solution variables are resolved at geometrically

<sup>1</sup>For complex geometries, only tetrahedral meshes guarantee exact planar faces. Other polyhedrons may exhibit non-planarity.

complex boundaries by employing mesh boundary layers, strongly reducing the number of cells required to achieve specific accuracy. Hence, this reduces the overall required computational resources.

While, in principle, the Finite-Volume method can be applied to a moving mesh where the cell corner points  $P_h$  change over time, within this thesis only static meshes with fixed  $P_h$  are considered.

## 3.2. Unstructured Finite-Volume method

The central approach of the Finite-Volume method is to approximate a spatially varying quantity  $\psi(\mathbf{x})$  with a volume average

$$\psi_c = \frac{1}{|\Omega_c|} \int_{\Omega_c} \psi(\mathbf{x}) dV \quad (3.4)$$

for each cell, resulting in a piecewise constant approximation. Thus, the integral form of eq. (2.13) over a control volume  $\Omega_c$  is considered. Where applicable, the *divergence theorem*

$$\int_{\Omega_c} \nabla \cdot \mathbf{B} dV = \oint_{\partial\Omega_c} \mathbf{B} \cdot \mathbf{n} dS = \sum_{f \in C_c} \int_{S_f} \mathbf{B} \cdot \mathbf{n}_f dS, \quad (3.5)$$

with  $\mathbf{B}$  denoting a vector field in  $\mathbb{R}^3$ , is used to transform volume integrals to surface integrals. The integral over a cell face  $S_f$  defines a so-called *flux*

$$\text{flux}_f(\mathbf{B}) := \int_{S_f} \mathbf{B} \cdot \mathbf{n}_f dS. \quad (3.6)$$

For two cells sharing a face  $S_f$ , the flux only differs in its sign, not its magnitude. This is a crucial observation for ensuring conservativeness at the discrete level.

Below, each term of eq. (2.13) is discussed individually, namely the *time derivative*, the *convective term*, the *diffusion term* and the *source terms*.

### 3.2.1. Time derivative

Integration of the time derivative in eq. (2.13) over a fixed control volume  $\Omega_c$  and applying the *Reynolds transport theorem* to interchange the order of integration and time derivative yields

$$\int_{\Omega_c} \partial_t(\rho \mathbf{v}) dV = \partial_t \int_{\Omega_c} \rho \mathbf{v} dV. \quad (3.7)$$

Using quadrature to approximate the integral by midpoint rule and, for example, the Euler scheme to approximate the time derivative results in

$$\partial_t \int_{\Omega_c} \rho \mathbf{v} dV \approx \frac{|\Omega_c|}{\Delta t} (\rho \mathbf{v}^{n+1} - \rho \mathbf{v}^n) \quad (3.8)$$

with discrete time levels  $t^{n+1} = t^n + \Delta t$ ,  $n \in \mathbb{N}$ , and  $\mathbf{v}^{n+1} = \mathbf{v}(\mathbf{x}_c, t^{n+1})$ ,  $\mathbf{v}^n = \mathbf{v}(\mathbf{x}_c, t^n)$  associated with the cell centroid  $\mathbf{x}_c$ . This approximation is first order accurate in time and second order accurate in space. Whether a term is discretized explicitly or implicitly with respect to time is explained below. The density  $\rho$  is left without a time index on purpose. As  $\rho$  is assumed constant for each phase,  $\rho^{n+1} = \rho^n$  holds for bulk cells. For interface cells,  $\rho$  is a function of  $\Sigma$  (see eq. (2.8)) and its calculation within the solution algorithm is described in chapter 5.

### 3.2.2. Convective term

Applying the divergence theorem eq. (3.5) to the convective term integrated over  $\Omega_c$  results in

$$\int_{\Omega_v} \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) dV = \sum_{f \in C_c} \int_{S_f} \rho \mathbf{v} \otimes \mathbf{v} \cdot \mathbf{n}_f dS = \sum_{f \in C_c} \int_{S_f} (\rho \mathbf{v} \cdot \mathbf{n}_f) \mathbf{v} dS. \quad (3.9)$$

With the midpoint rule for the surface integrals the second order accurate approximation

$$\sum_{f \in C_c} \int_{S_f} (\rho \mathbf{v} \cdot \mathbf{n}_f) \mathbf{v} dS \approx \sum_{f \in C_c} (|S_f| \rho \mathbf{v}_f \cdot \mathbf{n}_f \mathbf{v}_f) \quad (3.10)$$

is obtained with  $\mathbf{v}_f$  denoting the velocity at a face centre  $\mathbf{x}_f$ . It is linearized by employing two different time levels for  $\mathbf{v}_f$ ,  $|S_f| \rho \mathbf{v}_f^n \cdot \mathbf{n}_f \mathbf{v}_f^{n+1} = \dot{m}_f^n \mathbf{v}_f^{n+1}$ , using the known velocity  $\mathbf{v}_f^n$  to compute the *mass flux*  $\dot{m}_f^n$ . Whether the non-linearity is further accounted for depends on the solution algorithm. The Pressure Implicit with Splitting of Operators (PISO) algorithm [49], a common solution method for transient flows, does not further consider the non-linearity of the convective term. In contrast, within the scope of this thesis an algorithm has been developed, section 5.8, which allows to update  $\dot{m}_f$  in an iterative manner, thereby considering the non-linearity.

Within the cell-centred Finite-Volume method, face-centred quantities like  $\mathbf{v}_f$  are not readily available. Consequently, interpolation from cell-centres is required. This is explained in section 3.2.5 further down.

### 3.2.3. Diffusion term

As for the convective term, the divergence theorem eq. (3.5) is used to transform the volume integral of the diffusion term into a surface integral

$$\int_{\Omega_c} \nabla \cdot \left( \mu \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right) \right) dV = \sum_{f \in C_c} \int_{S_f} \mu \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right) \cdot \mathbf{n}_f dS. \quad (3.11)$$

Replacing the surface integrals with midpoint rule quadrature yields the approximation

$$\sum_{f \in C_c} \int_{S_f} \mu \left( \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right) \cdot \mathbf{n}_f dS \approx \sum_{f \in C_c} \mu_f \left( \nabla \mathbf{v}_f \cdot \mathbf{n}_f + (\nabla \mathbf{v})_f^T \cdot \mathbf{n}_f \right) |S_f|. \quad (3.12)$$

Discretization with respect to time is split into an implicit and an explicit contribution. The scalar product  $\nabla \mathbf{v}_f \cdot \mathbf{n}_f$  is discretized implicitly by

$$\nabla \mathbf{v}_f^{n+1} \cdot \mathbf{n}_f \approx \frac{\mathbf{v}_{c,n}^{n+1} - \mathbf{v}_c^{n+1}}{\|\mathbf{d}_{cn}\|_2} \quad (3.13)$$

using the cell-centred  $\mathbf{v}_c^{n+1}$  of  $\Omega_c$  and cell-centred  $\mathbf{v}_{c,n}^{n+1}$  from the neighboring cell straddling face  $S_f$ . The term  $\left( \nabla \mathbf{v}_f^n \right)^T \cdot \mathbf{n}_f$  is discretized explicitly in the following way. First, the discrete cell-centred gradient  $\nabla \mathbf{v}_c^n$  is computed with the Green-Gauss scheme as

$$\nabla \mathbf{v}_c^n = \frac{1}{|\Omega_c|} \sum_{f \in C_c} \mathbf{v}_f^n \otimes \mathbf{n}_f |S_f| \approx \frac{1}{|\Omega_c|} \int_{\Omega_c} \nabla \mathbf{v}^n dV \quad (3.14)$$

with  $\mathbf{v}_f^n$  interpolated from cell centres. Afterwards,  $\nabla \mathbf{v}_c^n$  is interpolated to face centres to compute  $\left( \nabla \mathbf{v}_f^n \right)^T \cdot \mathbf{n}_f$ .

Overall, given an interpolation that is at least second order accurate and in the absence of additional mesh-related errors, the discretization eq. (3.12) is second-order accurate in space. More information about the mesh-induced discretization errors for the unstructured FVM can be found in [52, 56].

### 3.2.4. Source terms

In principle, the pressure gradient  $\nabla p$  in eq. (2.13) can be discretized using the Green-Gauss scheme, analogue to eq. (3.14). However, this would lead to an inconsistency with respect to the discretization employed in the pressure-correction equation (see section 5.8).

Thus, the reconstruction operator  $R$ , defined by eq. (5.17), is used to obtain a cell-centred  $\nabla p_c$  from the pressure fluxes at face centres

$$\nabla p_f \cdot \mathbf{S}_f \approx \frac{p_n - p_c}{\|\mathbf{d}_{cn}\|_2} |S_f|. \quad (3.15)$$

This approach is part of a *well-balanced* discretization [35, 94] ensuring that equilibria of the model eq. (2.13) also exist in its discrete form.

Handling of surface tension is a central part of this thesis and as such it is described in section 5.7.

### 3.2.5. Interpolation

The discretization schemes described above require interpolation from cell centres to face centres. With the exception of the convective term, quantities are interpolated linearly

$$\psi_f = (1 - w)\psi_c + w\psi_{c,n} \quad (3.16)$$

with  $\psi_c$  denoting a generic cell-centred quantity. The interpolation weight  $w$  is computed by

$$w = \frac{\|\mathbf{x}_f - \mathbf{x}_c\|_2}{\|\mathbf{x}_{c,n} - \mathbf{x}_c\|_2}. \quad (3.17)$$

For the convective term, linear interpolation may negatively impact the stability of the resulting linear system of equations. Instead, either *upwind* interpolation

$$\psi_f = \begin{cases} \psi_c, & \dot{m}_f \geq 0, \\ \psi_{c,n}, & \dot{m}_f < 0 \end{cases} \quad (3.18)$$

or a *high resolution scheme* [77, chapter 12] is used.

## 3.3. Volume fraction and mixture of fluid properties

For the Finite-Volume method, instead of evaluating eq. (2.8) and eq. (2.9) at  $\mathbf{x}_c$  to compute density and viscosity for a cell, the cell-averaged fluid properties are calculated with a volume-averaged phase indicator  $\alpha_c$ . This so-called *volume fraction* is given, according to eq. (3.4), by

$$\alpha_c = \frac{1}{|\Omega_c|} \int_{\Omega_c} \chi(\mathbf{x}, t) dV. \quad (3.19)$$



---

With the volume fraction, the single-field cell-centred fluid properties are computed by

$$\rho_c = \alpha_c \rho^- + (1 - \alpha_c) \rho^+, \quad (3.20)$$

$$\mu_c = \alpha_c \mu^- + (1 - \alpha_c) \mu^+, \quad (3.21)$$

giving the corresponding bulk values for cells away from the interface and intermediate values for cells intersected by  $\Sigma$ . Equation (3.20) follows directly from mass conservation since

$$\begin{aligned} |\Omega_c| \rho_c &= |\Omega_c| (\alpha_c \rho^- + (1 - \alpha_c) \rho^+) \\ &= |\Omega_c| \frac{1}{|\Omega_c|} \left[ \int_{\Omega_c} \chi dV \rho^- + \int_{\Omega_c} (1 - \chi) dV \rho^+ \right] \\ &= \int_{\Omega_c} \rho^- \chi + \rho^+ (1 - \chi) dV \\ &= \int_{\Omega_c} \rho dV. \end{aligned}$$

For viscosity, in contrast, there is no obvious functional relation between  $\mu_c$  and  $\mu^-, \mu^+$ . Additionally, according to eq. (3.12), viscosity values are required at the face centres. Here, the same approach as for  $\rho_c$  is used to calculate  $\mu_c$  which is interpolated to obtain  $\mu_f$ . However, there are also alternatives, e.g. using the harmonic mean, reported in the literature [58, 119, 62].

Evaluation of eq. (3.19) is trivial for bulk cells as  $\chi$  is constant within those cells. For interface cells  $\Omega_c \cap \Sigma \neq \emptyset$ , however, accurate approximation is more involved and depends on the representation of  $\Sigma$ . Computation of  $\alpha_c$  is discussed for the case when  $\Sigma$  is represented by a triangle surface mesh in the following chapter.



---

## 4. Computation of signed distances and volume fractions from surface meshes

---

In the context of hybrid Level Set / Front Tracking methods, the volume fractions  $\alpha_c$  are required for the calculation of the one-field fluid properties, eqs. (3.20) and (3.21). However,  $\alpha_c$  is not directly available for this method type and thus has to be approximated from the available interface representations, meaning either the discrete level set values or the triangular surface. This calculation has an interesting secondary application, namely calculation of initial conditions for another two-phase method, the VOF method.

In this chapter, a new numerical algorithm that calculates initial conditions for simulations of two-phase flow problems for fluid interfaces of complex shapes is presented. The initial conditions are calculated in the form of signed distances and volume fractions from fluid interfaces approximated as arbitrarily shaped triangular surfaces immersed in unstructured meshes. The signed distances are relevant as initial conditions for the Level Set method [115, 114] for multiphase flow simulation. Volume fractions on unstructured meshes are required for the unstructured Volume-of-Fluid method (cf. [69] for a recent review). In fact, the proposed algorithms have been applied to model experimental fluid interfaces from wetting experiments [44], which was not possible using available contemporary approaches that model fluid interfaces using (compositions of) implicit functions or parameterized surfaces. The proposed algorithm approximates the surfaces using triangle meshes that are omnipresent in Computer-Aided Design (CAD) because of their versatility: they can approximate basic surfaces such as spheres and ellipsoids, but also surfaces of mechanical parts, disjoint surfaces in mechanical assemblies, or surfaces resulting from imaging scans.

### 4.1. Literature review

The unstructured VOF methods [69] rely on the volume fraction field  $\alpha_c$  to track interface with the advecting velocity obtained from the solution of two-phase Navier-Stokes equations in a single-field formulation. All multiphase flow simulation methods that utilize the

single-field formulation of Navier-Stokes equations approximate the phase indicator function similarly to eq. (3.19). The phase indicator approximation utilizes signed distances in the Level Set [115, 113, 114] method, the volume fractions approximate the phase indicator for the Volume-of-Fluid [26, 81, 46, 98] method.

Various methods exist that compute the volume fraction  $\alpha_c$  based on the exact phase indicator  $\chi(\mathbf{x}, t)$ . The majority of methods calculate the integral in eq. (3.19) numerically, as schematically shown in fig. 4.1, using numerical quadrature.

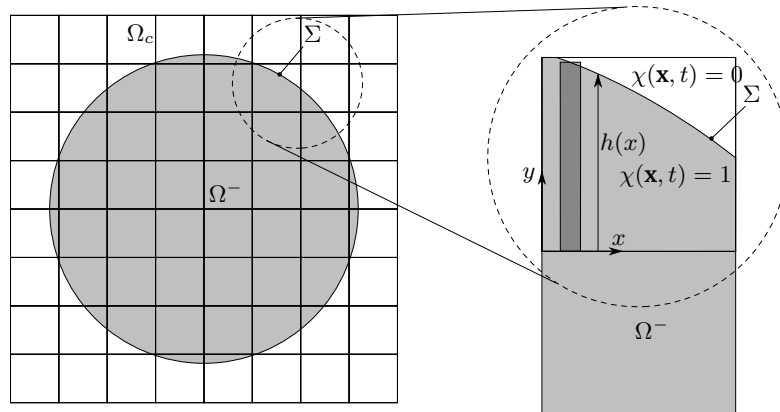


Figure 4.1.: Calculating volume fractions of a circular interface by numerical integration.

Different approaches are outlined below with increasing complexity in terms of admissible shapes of the fluid interface. The admissible shapes range from analytic descriptions of basic geometric shapes such as spheres and ellipsoids to implicit functions (or their combinations) and more general shapes approximated with volume meshes.

Strobl et al. [112] propose an exact intersection between a sphere and a tetrahedron, a wedge, or a hexahedron. The proposed algorithm is exact and fast, though it is limited to a spherical interface shape.

Fries and Omerović [36] represent the fluid interface as a level set and propose a higher-order quadrature for the integral on the right-hand side of eq. (3.19). The parametrization of the surface uses roots of the implicit function found by the closest-point algorithm. Results are presented for hexahedral and tetrahedral unstructured meshes that may also be strongly deformed. Fries and Omerović [36, fig. 52, fig. 53] also show results with higher-order ( $> 2$ ) convergence for the volume integration of an arbitrary non-linear function on hexahedral and tetrahedral meshes. However, the volume and area integration error is reported for a single function. While a relative global volume error between  $1e-08$  and  $1e-06$  is reported, no information about the required CPU times is provided. In the

approach proposed by Fries and Omerović [36], fluid interfaces with complex shapes are modeled as a composition of implicit functions.

Kromer and Bothe [60] propose an efficient third-order accurate quadrature to approximate eq. (3.19). Contrary to Jones et al. [55], who decompose cells into tetrahedrons, Kromer and Bothe [60] locally approximate the hypersurface by a paraboloid based on the principal curvatures. Applying the Gaussian divergence theorem to eq. (3.19) then yields contributions from the cell boundary and the approximated hypersurface patch. Using the surface divergence theorem, Kromer and Bothe [60] reformulate the contribution from the hypersurface patch into a set of line integrals, where the associated integrand emerges from the solution of a Laplace-Beltrami-type problem. The method of Kromer and Bothe [60] is directly applicable to unstructured meshes. However, locally, i.e., within a cell, the fluid interface has to be  $C^2$  and simply connected.

Aulisa et al. [10] and Bnà et al. [16, 17] calculate the volume fraction by representing the indicator function as a height function inside cubic cells, using the structure of the underlying Cartesian mesh. Numerical integration of the height function is illustrated by fig. 4.1. However, extending this approach to unstructured meshes raises many questions. First, constructing a height function in a specific direction is complex and computationally expensive [88]. Second, the orientation of the interface in the chosen coordinate system may easily make the problem ill-conditioned. Finally, required mesh-search operations are complicated as the face normals of polyhedral cells are typically not aligned with the coordinate axes.

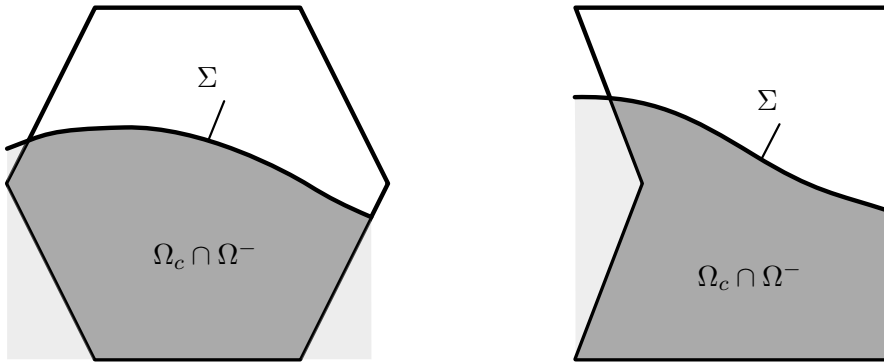


Figure 4.2.: Polyhedral cell (left) and non-convex cell (right) for which the intersection volume (dark grey) has to be computed. The light grey regions lead to cases that have to be identified and require special treatment increasing the problem complexity far beyond a simple one dimensional integration.

---

The calculation of the volume fraction given by  $\alpha_c = \frac{|\Omega^- \cap \Omega_c|}{|\Omega_c|}$  can be reformulated into the integration of a function  $f = 1$  within  $\Omega^- \cap \Omega_c$ . Since  $\partial\Omega_c$  consists of piecewise-planar surfaces (faces), the complexity lies in the non-planar part of the surface  $\partial\Omega^- \cap \Omega_c = \Sigma(t) \cap \Omega_c$ . Trimmed isogeometric analysis can be used to integrate  $f = 1$  within  $\Omega^- \cap \Omega_c$  by representing  $\partial\Omega^- \cap \Omega_c$  using a trimmed NURBS surface, effectively resulting in  $\alpha_c = \frac{|\Omega^- \cap \Omega_c|}{|\Omega_c|}$  for complex non-linear CAD surfaces. Although not yet applied to volume fraction calculation ( $f = 1$  integration), trimmed isogeometric analysis has been applied to solving PDEs in solution domains bounded by NURBS surfaces [59, 102, 80]. Similarly, the immersed isogeometric analysis (e.g. [32]) requires function integration in cut cells, where the integration of  $f = 1$  in the cut cell is equivalent to computing  $|\Omega^- \cap \Omega_c|$  used in volume fraction calculation. Although it is a potentially interesting alternative approach for computing volume fractions from CAD surfaces, the isogeometric analysis requires NURBS trimming, octree refinement, and higher-order quadratures. These efforts are worthwhile for the goal of achieving higher-order solutions for PDEs in complex solution domains. However, as demonstrated in the results section 4.3, the proposed algorithms achieve sufficient accuracy for signed distances and volume fractions on unstructured meshes while relying on straightforward second-order accurate discretization.

The signed distances in the Level Set Method require re-distancing (correction). The re-distancing methods are usually based on approximate solutions of PDEs that ensure the signed-distance property [100]. Contrary to this approach, the unstructured Level Set / Front Tracking method [71] *geometrically* computes minimal signed distances from  $\tilde{\Sigma}$ . This calculation is relatively straightforward on structured meshes [105, 109], but significantly more complex on unstructured meshes [71]. Here, the calculation of signed distances from [71] is significantly extended by introducing an efficient approximate propagation of the inside/outside information from  $\tilde{\Sigma}$ .

Volume fraction calculation methods outlined so far model the fluid interface using exact functions and handle more complex interface shapes via combinations of these functions. A combination of exact functions cannot accurately capture the shape of the fluid interface in many cases. For example, when the interface shape is prescribed experimentally [44].

One approach exists that can handle arbitrarily complex interface shapes. In this approach, the fluid interface encloses a volumetric mesh as its boundary surface mesh. This mesh given by the fluid interface is intersected with a "background" mesh that stores volume fractions. This approach is called *volume mesh intersection*. An example for such an intersection between  $\tilde{\Omega}$  and cells from  $\tilde{\Omega}^-$  is shown in fig. 4.3. In principle, this approach is relatively straightforward, provided an accurate geometrical intersection of tetrahedrons is available. However, geometrical operations based on floating-point numbers are not stable and can lead to severe errors [123, chap. 45].

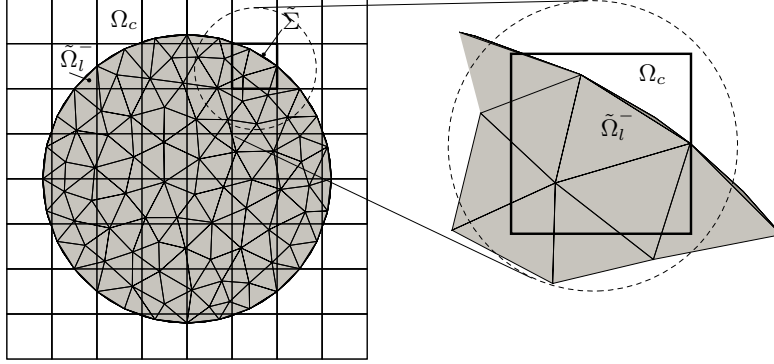


Figure 4.3.: Calculating volume fractions from a circular interface by volume mesh intersection.

Ahn and Shashkov [4] have initialized volume fractions by volume mesh intersection as shown in fig. 4.3. In this approach, the approximated phase  $\tilde{\Omega}^-(t)$  is decomposed into volumes (an unstructured mesh), equivalently to the decomposition  $\tilde{\Omega}$  given by eq. (3.1). The boundary  $\partial\Omega^-$  is the fluid interface  $\Sigma(t)$ , and it is approximated as a polygonal surface mesh, leading to

$$\Omega^- \approx \tilde{\Omega}^- := \{\tilde{\Omega}_l^-\}_{l \in L}, \quad (4.1)$$

i.e. an approximation of  $\Omega^-$ . Generally, as shown in the detail in fig. 4.3, a cell  $\Omega_c$  of the background mesh  $\tilde{\Omega}$  may overlap with multiple cells  $\Omega_l$  from the  $\tilde{\Omega}^-$  mesh, and vice versa. A set of indices  $l$  of cells  $\tilde{\Omega}_l^-$  in  $\tilde{\Omega}^-$  is defined that overlap with the cell  $\Omega_c$ : the so-called *cell stencil* of  $\Omega_c$  in  $\tilde{\Omega}_l^-$ , namely

$$\mathcal{S}(\Omega_c, \tilde{\Omega}^-) = \{l \in L : \Omega_c \cap \tilde{\Omega}_l^- \neq \emptyset, \text{ where } \Omega_c \in \tilde{\Omega}, \tilde{\Omega}_l^- \in \tilde{\Omega}^-\}, \quad (4.2)$$

where  $L$  is an index set, containing indices of cells from  $\tilde{\Omega}^-$ . Volume fractions  $\{\alpha_c\}_{c \in C}$  can then be calculated by performing the intersection

$$\alpha_c = \frac{|\cup_{l \in \mathcal{S}(\Omega_c, \tilde{\Omega}^-)} \Omega_c \cap \tilde{\Omega}_l^-|}{|\Omega_c|}. \quad (4.3)$$

Since each  $\tilde{\Omega}_l^-$  overlaps with at least a one cell from  $\tilde{\Omega}$ , and the number of cells from  $\tilde{\Omega}$  that intersect each cell from  $\tilde{\Omega}^-$  can be approximated as

$$N(\tilde{\Omega}^-, \tilde{\Omega}) \approx |\tilde{\Omega}^-| \text{mean}_{l \in L}(|\mathcal{S}(\tilde{\Omega}_l^-, \tilde{\Omega})|), \quad (4.4)$$

---

where  $|\tilde{\Omega}^-|$  denotes the number of cells in the mesh  $\tilde{\Omega}^-$ . The average number of cells  $\Omega_c$  overlapping  $\tilde{\Omega}_l^-$ ,  $\text{mean}_{l \in L} |C(\tilde{\Omega}_l^-, \tilde{\Omega})|$ , depends on the mesh densities of both meshes,  $\tilde{\Omega}$  and  $\tilde{\Omega}^-$ . However, it is safe to assume that  $\text{mean}_{l \in L} |C(\tilde{\Omega}_l^-, \tilde{\Omega})| > 1$ . Next, it is known that  $|\tilde{\Omega}^-|$  grows quadratically in  $2D$  and cubically in  $3D$  with a uniform increase in mesh resolution, taken as the worst case scenario. It grows linearly in  $2D$  and quadratically in  $3D$  if  $\tilde{\Omega}^-$  is refined only near the interface  $\tilde{\Sigma} := \partial\tilde{\Omega}^-$ . Consequently, the computational complexity of the volume mesh intersection algorithm in terms of cell/cell intersections is quadratic in  $2D$  and cubic in  $3D$  in the worst case, and linear in  $2D$  and quadratic in  $3D$  if local refinement is used to increase the resolution of  $\tilde{\Sigma}$ . The quadratic complexity in  $3D$  is a serious drawback of this algorithm, especially for large simulations where  $|\tilde{\Omega}^-|$  easily reaches hundred thousand cells per CPU core. Menon and Schmidt [75] have extended the volume mesh intersection algorithm from Ahn and Shashkov [4] to perform a volume conservative remapping of variables in the collocated FVM with second-order accuracy on unstructured meshes. Their results confirm the polynomial computational complexity in terms of absolute CPU times for this volume mesh intersection algorithm [75, table 3].

López et al. [66] propose a volume truncation algorithm for non-convex cells and apply it to the initialization of volume fractions from exact functions on unstructured meshes. Cell-subdivision is introduced to handle cases for which the interface crosses an edge of a cell twice. Non-planar truncated volumes are triangulated [66, fig 18], and second-order accuracy is demonstrated in terms of the relative global volume error for a uniform resolution and a higher-order accuracy when locally refined sub-grid meshes are used.

Ivey and Moin [50] initialize volume fractions on unstructured meshes using tetrahedral decomposition of non-convex cells and perform geometrical intersections with a similar approach as the approach from Ahn and Shashkov [4]. Unlike Ahn and Shashkov [4], Ivey and Moin [50] compute volume fractions of intersected tetrahedrons by intersecting them with exact signed distance functions that are used to model the fluid interface. Therefore, this algorithm cannot directly utilize arbitrarily shaped interfaces. However, their approach utilizes a linear interpolation of intersection points between the tetrahedron and the signed-distance function and yields second-order accuracy. Accuracy is further increased using adaptive mesh refinement.

The approaches reviewed so far require an exact representation of the interface using explicit analytic expressions, which hinders the direct application of such algorithms to initial conditions resulting from experiments as these are typically not available as function compositions. The volume mesh intersection algorithm [4] is flexible but computationally expensive, and it requires highly accurate and robust geometrical intersections.

The following sections outline the proposed algorithm that uses an unstructured surface



---

mesh  $\tilde{\Sigma}$  to compute signed distances and volume fractions on unstructured meshes. Relying on unstructured surface meshes retains the ability to handle arbitrary-shaped surfaces while avoiding computationally expensive cell/cell intersections. Of course, using surface meshes to approximate the fluid interface renders the proposed algorithm second-order accurate; however, sufficient absolute accuracy is achievable with second-order accurate methods using local mesh refinement on the background mesh [24, 35]. Applying local mesh refinement on the background mesh in the close vicinity of the triangulated surface increases the accuracy and limits it to the resolution of the surface mesh, not the background mesh that stores volume fractions and signed distances. The proposed algorithm geometrically computes signed distances near the fluid interface. These signed distances (so-called *narrow-band* signed-distances) are then propagated throughout  $\tilde{\Omega}$  by an approximate solution of a diffusion equation. The propagated signed distances determine the value of the phase indicator  $\chi(\mathbf{x}, t)$  in those cells that are either completely empty ( $\alpha_c = 0$ ), or completely full ( $\alpha_c = 1$ ). Finally, second-order accurate volume fraction values are calculated in intersected cells ( $0 < \alpha_c < 1$ ). This work enables the calculation of complex initial conditions for different multiphase simulation methods. These include in particular geometric [54, 50, 86, 70], geometric/algebraic [101] and algebraic VOF methods [128, 29]. The calculation of volume fractions from a surface mesh (marker points in 2D) was done in the mixed markers / VOF method by Aulisa et al. [9]: the proposed algorithm significantly extends this idea towards an accurate and fast volume fraction model for Front Tracking methods [126], as well as the hybrid Level Set / Front Tracking methods on structured [105, 109] or unstructured [71] meshes. Signed distances and the respective inside-outside information from triangulated surfaces are available for unstructured Level Set and Immersed Boundary methods.

## 4.2. Algorithm description

The calculation of volume fractions by the proposed Surface-Mesh Cell Intersection/Approximation (SMCI/A) algorithms, outlined in fig. 4.4, requires signed distances to the interface at cell centres and cell corner points. As a naive computation is computationally expensive (section 4.2.2), we employ an octree based approach to the calculation of signed distances. Starting point of the octree based search is the calculation of search radii at the relevant points.

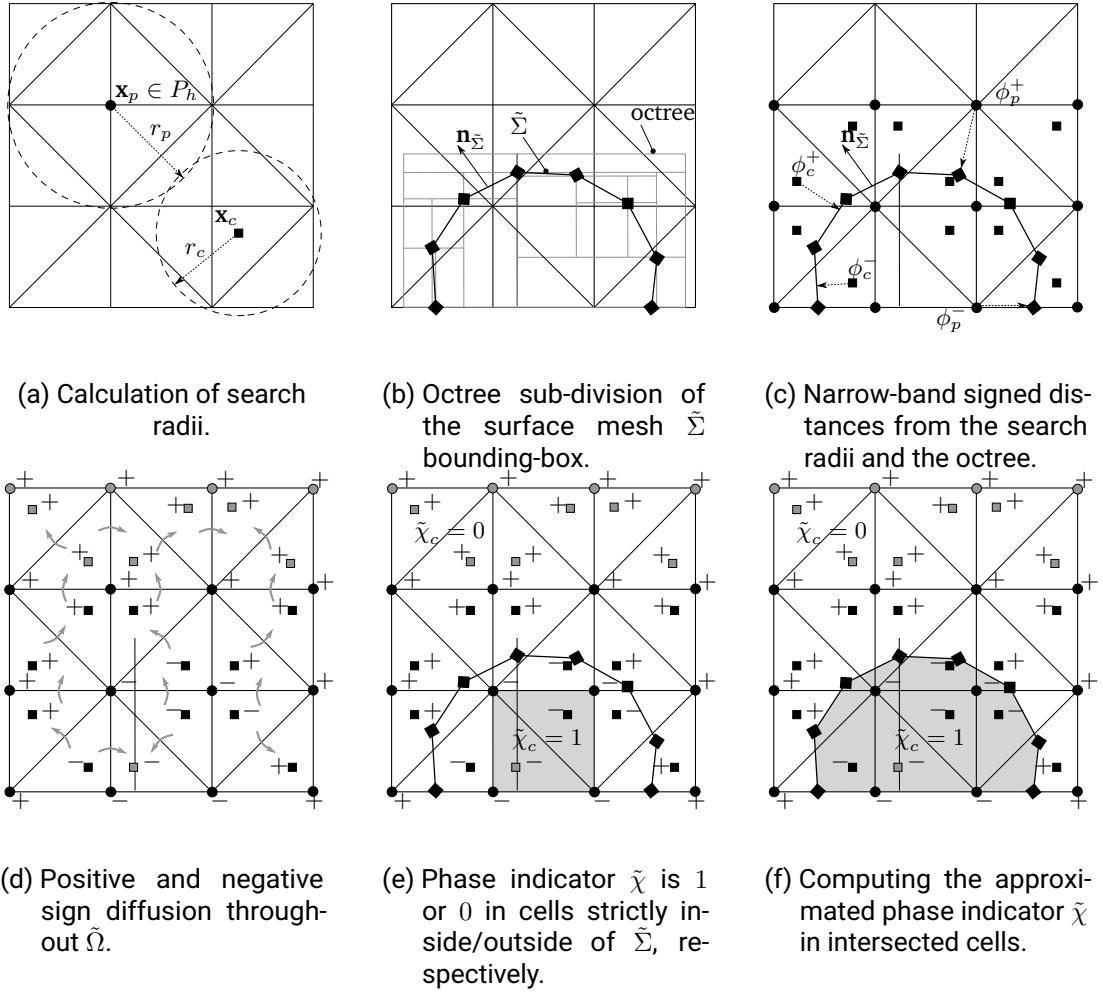


Figure 4.4.: Steps of the Surface Mesh Intersection / Approximation (SMCI/A) algorithms.

#### 4.2.1. Octree-based search structure

In the first step, a search radius  $r_c$  and  $r_p$  is calculated at each cell center and cell-corner point, respectively. This is illustrated in fig. 4.4a. Here, the cell search radius  $r_c$  is defined by

$$r_c = \lambda_s \min_{f \in C_c} \|\mathbf{x}_{f,O} - \mathbf{x}_{f,N}\|_2, \quad (4.5)$$

where  $\mathbf{x}_c$  is the cell center,  $\lambda_s > 0$  is the *search radius factor* detailed below and  $\mathbf{x}_{f,O}$ ,  $\mathbf{x}_{f,N}$  are the cell centers of two cells that share the face with index  $f$  of the cell  $\Omega_c$  ( $O$  for owner cell with a smaller cell index than the neighbor cell  $N$ ). Here, the index set  $C_c$  contains the indices of those faces that form the boundary of  $\Omega_c$ . Based on (4.5), the corner-point search radius  $r_p$  is defined by

$$r_p = \lambda_s \min_{c \in C_p(\mathbf{x}_p)} r_c, \quad (4.6)$$

where  $\mathbf{x}_p$  is the cell-corner point, while the *point-cell stencil* is the index set  $\mathcal{S}(\mathbf{x}_p, \tilde{\Omega})$ , that contains indices of all cells from  $\tilde{\Omega}$  whose corner-point is  $\mathbf{x}_p$ .

The search radii introduced above are used to define search balls in 3D (circles in 2D), which are used to reduce the number of calculations to determine signed distances between the cell corner points  $\mathbf{x}_p$  and the cell centers  $\mathbf{x}_c$  with respect to the provided surface mesh  $\tilde{\Sigma}$ .

#### 4.2.2. Octree decomposition of the surface mesh

In contrast to various other approaches for volume fraction initialization, the here interface is not represented by some kind of function, but as a set of triangles. First, to define the interface  $\tilde{\Sigma}$ , the convex hull of a set of  $n$  points  $P^n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ ,  $\mathbf{x}_i \in \mathbb{R}^3$  is denoted by

$$\text{conv}(P^n) := \left\{ \mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \sum_{\mathbf{x}_i \in P^n} \gamma_i \mathbf{x}_i, \sum_{i=1}^n \gamma_i = 1 \right\}. \quad (4.7)$$

Using this, a triangle is defined as the convex hull of a point triple:  $\mathcal{T} := \text{conv}(P^3)$ . Consequently, the surface mesh is defined as

$$\tilde{\Sigma} := \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_n\}. \quad (4.8)$$

With the structure of  $\tilde{\Sigma}$  in mind, it should be emphasized why an octree based approach is the key to obtaining reasonable computation times. Consider the case where a minimal distance between a point  $\mathbf{x}$  and  $\tilde{\Sigma}$  would be calculated for each cell center  $\mathbf{x}_c$  and cell-corner point  $\mathbf{x}_p$ . The need for the spatial subdivision and search operations becomes obvious, as this would require a distance computation between each point of the interface mesh and each cell centers and cell corner points of the background mesh. Consequently, this would require  $|C| |\tilde{\Sigma}|$  operations to compute the geometric signed distances at cell centers and additional computations for evaluating signed distances at cell-corner points. For the computations below, the number  $|C|$  often reaches the order of 1e05 per CPU core, while

$|\tilde{\Sigma}|$  is typically on the order of  $1e04$  per CPU core. Aiming at redistancing computations for a dynamic setting in multiphase flows where  $\tilde{\Sigma} = \tilde{\Sigma}(t)$ , such a large number of distance computations makes such a brute force redistancing approach prohibitively expensive.

The first step of the signed distance calculation is the computation of an Axis-Aligned Bounding Box (AABB) from the surface mesh  $\tilde{\Sigma}$ . The AABB is used to build an octree data structure, illustrated as a  $2D$  quadtree subdivision in fig. 4.4b, which is used to access  $\tilde{\Sigma}$ . The octree data structure enables fast search queries involving cell centers and cell corner-points that are close to the surface mesh  $\tilde{\Sigma}$ , with a logarithmic computational complexity with respect to the number of vertices in  $\tilde{\Sigma}$  [73, 74]. The structure of the octree depends on the ordering of vertices in  $\tilde{\Sigma}$ : since  $\tilde{\Sigma}$  is an unstructured surface mesh, its vertices are generally sufficiently unordered, which makes the octree well-balanced. Once the octree has been constructed, it can be used to find the closest points  $\mathbf{x} \in \tilde{\Sigma}$  to cell centres  $\mathbf{x}_c$  and cell corner points  $\mathbf{x}_p$ . Note that this is only true for those  $\mathbf{x}_c, \mathbf{x}_p$  which are sufficiently close to  $\tilde{\Sigma}$  in terms of their search radius  $r_c, r_p$ . Thus, the search radii define a so-called *narrow band* around  $\tilde{\Sigma}$ , where the nearest distances are calculated geometrically. The narrow band of  $\tilde{\Sigma}$  is denoted with  $\mathcal{N}(\tilde{\Sigma})$ , and the closed ball  $\mathcal{B}(\mathbf{x}^*, r) := \{\mathbf{x} \in \mathbb{R}^3 \mid \|\mathbf{x} - \mathbf{x}^*\|_2 \leq r\}$  with a radius  $r$  around a point  $\mathbf{x}$ . Then

$$\mathcal{N}(\tilde{\Sigma}) := \left\{ \mathbf{x} \in \mathbb{R}^3 \mid \exists \mathcal{T} \in \tilde{\Sigma} \text{ such that } \mathcal{T} \cap \mathcal{B}(\mathbf{x}, r) \neq \emptyset \right\}, \quad (4.9)$$

where  $r$  is either  $r_p$  or  $r_c$ .

### 4.2.3. Signed distance computation

For a point  $\mathbf{x} \in \mathcal{N}(\tilde{\Sigma})$ , the octree provides the closest point  $\mathbf{x}_{\min} \in \mathcal{T}_{\min}$  for some  $\mathcal{T} \in \tilde{\Sigma}$  and the corresponding triangle  $\mathcal{T}_{\min}$  itself. While the absolute distance can be directly computed as  $\|\mathbf{x} - \mathbf{x}_{\min}\|_2$ , care must be taken when computing the sign with respect to the orientation of  $\tilde{\Sigma}$ . Directly using the triangle normals  $\mathbf{n}_{\mathcal{T}}$  may lead to false signs and consequently, to erroneous volume fractions. Thus, the work of [120, 11] is followed and *angle weighted normals*

$$\mathbf{n}_{\mathbf{x}_v} = \frac{\sum_{\mathcal{T} \in \text{ngh}(\mathbf{x}_v)} \beta_{\mathcal{T}} \mathbf{n}_{\mathcal{T}}}{\sum_{\mathcal{T} \in \text{ngh}(\mathbf{x}_v)} \beta_{\mathcal{T}}} \quad (4.10)$$

are computed at the vertices  $\mathbf{x}_v$  of  $\tilde{\Sigma}$ . Here,  $\text{ngh}(\mathbf{x}_v)$  denotes the set of all triangles containing  $\mathbf{x}_v$ ,  $\mathbf{n}_{\mathcal{T}}$  a triangle normal and  $\beta_{\mathcal{T}}$  the inner angle of  $\mathcal{T}$  at  $\mathbf{x}_v$ . Baerentzen and Aanaes [11] propose a classification of the point  $\mathbf{x}_{\min}$  whether it is located within a triangle, on an edge, or a vertex and base the choice of the normal on this classification. While such a classification is simple in theory, a robust implementation is difficult due to the

limited precision of floating point arithmetic. Thus, instead a linear interpolation of  $\mathbf{n}_{\mathbf{x}_v}$  within  $\mathcal{T}_{\min}$  to  $\mathbf{x}_{\min}$  is used, denoted  $\mathbf{n}_I(\mathbf{x}_{\min}, \mathcal{T}_{\min})$ . With this normal computation, the signed distance between  $\mathbf{x}$  and  $\mathbf{x}_{\min}$  is calculated by

$$\phi^g(\mathbf{x}, \tilde{\Sigma}) = \text{sign}((\mathbf{x} - \mathbf{x}_{\min}) \cdot \mathbf{n}_I(\mathbf{x}_{\min}, \mathcal{T}_{\min})) \|\mathbf{x} - \mathbf{x}_{\min}\|_2. \quad (4.11)$$

where the supindex  $g$  indicates a geometrical construction. This procedure is illustrated in fig. 4.4c. The robustness of this approach with regard to inside/outside classification is demonstrated in section 4.3.1.

Using the spatial subdivision provided by the octree, the computational complexity for finding the minimal distances between mesh points and  $\tilde{\Sigma}$  is reduced severely, as the vast majority of cell centers  $\mathbf{x}_c$  are not even considered for calculation as no triangle  $\mathcal{T} \in \tilde{\Sigma}$  exists within the corresponding search ball. The closest triangles of those points  $\mathbf{x}_c$ , whose ball  $\mathcal{B}(\mathbf{x}_c, r_c)$  intersects  $\tilde{\Sigma}$  are found with logarithmic search complexity with respect to  $|\tilde{\Sigma}|$ . This significant reduction of complexity can potentially enable a future application of the proposed algorithm on moving interfaces  $\tilde{\Sigma}(t)$  as a geometrically exact marker field model for unstructured Front Tracking methods. Therefore, it is crucial to understand that the  $\min_{\mathcal{T} \in \tilde{\Sigma}}$  operation in eq. (4.11) throughout this text relies on the octree spatial subdivision and search queries.

#### 4.2.4. Inside / outside propagation

After the calculation of geometric signed distances in the narrow band around  $\tilde{\Sigma}$ , the signed distances are propagated to the bulk of different phases, as shown in fig. 4.4d. In [71], the geometric signed distances are set to large positive numbers throughout the domain, and a graph-traversal algorithm is used to iteratively correct the signs of signed distances using face-cell and point-point graph connectivity provided by the unstructured mesh. Graph-traversal is computationally expensive and complicated to implement in parallel. Here, a straightforward alternative that instantaneously propagates signs of signed distances through the solution domain is proposed which is parallelized easily. This approach relies on the diffusion equation for the signed distances, namely

$$\begin{aligned} -\Delta\phi &= 0, \\ \nabla\phi &= 0, \quad \text{for } \mathbf{x} \in \partial\Omega \end{aligned} \quad (4.12)$$

and its discretization using the unstructured FVM in OpenFOAM [52, 56, 77], giving a linear system of equations. The key idea to sign propagation is to apply a few iterations ( $< 5$ ) of an iterative linear solver to this system. Here, a Conjugate Gradient approach

with an incomplete lower upper preconditioner has been used. With the initial field set to

$$\phi(\mathbf{x}) = \begin{cases} \phi_g(\mathbf{x}, \tilde{\Sigma}), & \text{if } \mathbf{x} \in \mathcal{N}(\tilde{\Sigma}) \\ 0, & \text{otherwise,} \end{cases} \quad (4.13)$$

this small number of iterations suffices to properly propagate  $\text{sign}(\phi)$  with respect to the orientation of  $\tilde{\Sigma}$  throughout  $\Omega$ . Prerequisite for this approach to work is that the narrow band has a certain minimum width in interface normal direction. At least four cells on each side of the interface are required to ensure a robust propagation. This is achieved by setting a global search radius factor  $\lambda_s := 4$  in eq. (4.5) used to calculate  $r_c$  at cell centers. Note that increasing  $\lambda_s$  beyond this value only increases computational costs, and does not impact the accuracy of the proposed algorithm, as with a larger value of  $\lambda_s$  the narrow band  $\mathcal{N}(\Sigma)$  becomes wider and consequently the geometrical signed distances are calculated at more points  $\mathbf{x}_c, \mathbf{x}_p$ , using eqs. (4.14) and (4.17), respectively.

Two aspects have to be considered when solving the linear system of equations resulting from the discretization of eq. (4.12). First, cells for which  $\mathbf{x}_c \in \mathcal{N}(\tilde{\Sigma})$  have to be excluded from the vector of unknowns as  $\phi^g(\mathbf{x}_c)$  is already known for those. Second, for cells away from  $\mathcal{N}(\tilde{\Sigma})$  the only relevant information is  $\text{sign}(\phi_c)$  indicating  $\Omega_c \in \Omega^-$  or  $\Omega_c \in \Omega^+$ , respectively. A few iterations of a linear solver suffice to reliably propagate  $\text{sign}(\phi_c)$  to the entire domain. The resulting field is

$$\phi_c = \begin{cases} \phi_c^g, & \text{if } \mathbf{x}_c \in \mathcal{N}(\tilde{\Sigma}), \\ \phi_c^a, & \text{otherwise,} \end{cases} \quad (4.14)$$

with  $\phi_c^g$  denoting geometric signed distances and  $\phi_c^a$  approximate values from the solution of eq. (4.12) carrying inside/outside information but without geometric meaning.

Once the cell-centered signed distances  $\phi_c$  are computed, they are used to calculate the signed distances at cell corner-points via

$$\phi_p^I = \sum_{c \in C_p} w_{p,c} \phi_c, \quad (4.15)$$

where  $C_p$  is the index set of cells that contain the cell corner point  $\mathbf{x}_p$  and the supindex  $I$  indicating interpolation. Furthermore,  $w_{p,c}$  is the inverse-distance weighted (IDW) interpolation weight

$$w_{p,c} = \frac{\|\mathbf{x}_c - \mathbf{x}_p\|_2^{-1}}{\sum_{\tilde{c} \in C_p} \|\mathbf{x}_{\tilde{c}} - \mathbf{x}_p\|_2^{-1}}. \quad (4.16)$$

As with  $\phi_c$ , the accuracy of  $\phi_p$  is irrelevant outside of the narrow band of  $\tilde{\Sigma}$ , only the sign of the signed distance is important in the bulk. To correct for the error introduced by the

IDW interpolation in eq. (4.15), signed distances at cell-corner points of intersected cells are calculated geometrically

$$\phi_p = \begin{cases} \phi_p^g, & \text{if } \mathbf{x}_p \in \mathcal{N}(\tilde{\Sigma}), \\ \phi_p^I, & \text{otherwise.} \end{cases} \quad (4.17)$$

Equations (4.14) and (4.17) define the final signed distances at cell centers and cell-corner points, respectively. These quantities will have the value of a geometrical distance to  $\tilde{\Sigma}$  in the narrow band, while outside of the narrow band only the correct sign resulting from the approximative solution of eq. (4.12) is relevant.

#### 4.2.5. Volume fractions by geometrical intersection

Once signed distances at cell centers  $\{\phi_c\}_{c=1,2,\dots,|\tilde{\Omega}|}$  and cell corner points  $\{\phi_p\}_{p=1,2,\dots,|P_h|}$  are calculated as outlined in the previous section, the Surface-Mesh / Cell Intersection (SMCI) algorithm calculates the volume fractions in a straightforward way. The volume fraction calculation is shown schematically for the SMCI algorithm in fig. 4.5b. Each cell is decomposed into tetrahedrons, using the cell centroid  $\mathbf{x}_c$  as the base point of the tetrahedron, the centroid of the face  $\mathbf{x}_{c,f}$ , and two successive points from the cell-face,  $\mathbf{x}_{c,f,i}$ ,  $\mathbf{x}_{c,f,i+1}$ . The resulting tetrahedron has the distance  $\phi_c$  associated to the cell centroid, the distance  $\phi_{c,f}$  associated to the face centroid, and and  $(\phi_{c,f,i}, \phi_{c,f,i+1})$  pair of distances associated with a pair of points that belong to the cell-face  $(c, f)$ , as shown in fig. 4.5b. If all the distances of the tetrahedron are negative, the tetrahedron lies in the negative halfspace with respect to  $\tilde{\Sigma}$ , and its total volume contributes to the sum of the volume of phase 1 inside the volume  $\Omega_c$ . If a pair of distances in a tetrahedron has different signs, the tetrahedron is intersected by the interface approximated by the surface mesh  $\tilde{\Sigma}$ . The volume of this intersection is calculated by geometrically intersecting the tetrahedron with those triangles from  $\tilde{\Sigma}$ , that have a non-zero intersection with a ball  $\mathcal{B}$  enclosing the tetrahedron. The center of the ball  $\mathcal{B}_{c,f,i} := \mathcal{B}(\mathbf{x}_{c,f,i}, R_{c,f,i})$  is the centroid of the tetrahedron  $\mathbf{x}_{c,f,i} = 0.25(\mathbf{x}_c + \mathbf{x}_{c,f} + \mathbf{x}_{c,f,i} + \mathbf{x}_{c,f,\text{mod}(i+1,|F_{c,f}|)})$ , where  $i = 0, \dots, |F_{c,f}| - 1$ , and  $F_f$  is the oriented set of indices of the points  $\mathbf{x}$  (cf. fig. 4.5b) that belong to the face  $f$  of the cell  $\Omega_c$ . The radius of the tetrahedron-ball  $\mathcal{B}_{c,f,i}$  is then

$$R_{c,f,i} = \max(\|\mathbf{x}_c - \mathbf{x}_{c,f,i}\|, \|\mathbf{x}_{c,f} - \mathbf{x}_{c,f,i}\|, \|\mathbf{x}_{c,f,j} - \mathbf{x}_{c,f,i}\|, \|\mathbf{x}_{c,f,\text{mod}(j+1,|F_{c,f}|)} - \mathbf{x}_{c,f,i}\|), \quad (4.18)$$

$j = 0, \dots, |F_{c,f}| - 1$ . This sub-set of  $\tilde{\Sigma}$  is found using the octree data structure with logarithmic complexity with respect to  $\tilde{\Sigma}$ , as outlined in the previous section. For the example tetrahedron in the cell shown in fig. 4.5b, the resulting intersection between

the approximated interface  $\tilde{\Sigma}$  and a tetrahedron from the cell  $\Omega_c$  is shown as the shaded volume. The magnitude of this volume is computed by applying the Gauss divergence theorem using eq. (4.28). The phase-specific volumes from cell-tetrahedrons are summed up for the cell  $\Omega_c$ , into the total phase-specific volume of the phase 1 within the cell  $\Omega_c$ , and the volume fraction is therefore computed as

$$\alpha_c = \frac{\sum_{f=0,\dots,|C_c|-1} \sum_{i=0,\dots,|F_{c,f}|-1} |T(\mathbf{x}_c, \mathbf{x}_{c,f}, \mathbf{x}_{c,f,i}, \mathbf{x}_{c,f,\text{mod}(i+1,|F_{c,f}|)}) \cap (\mathcal{B}_{c,f,i} \cap \tilde{\Sigma})|}{|\Omega_c|} \quad (4.19)$$

with  $T := \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$  denoting a tetrahedron. The SMCI algorithm is summarized by

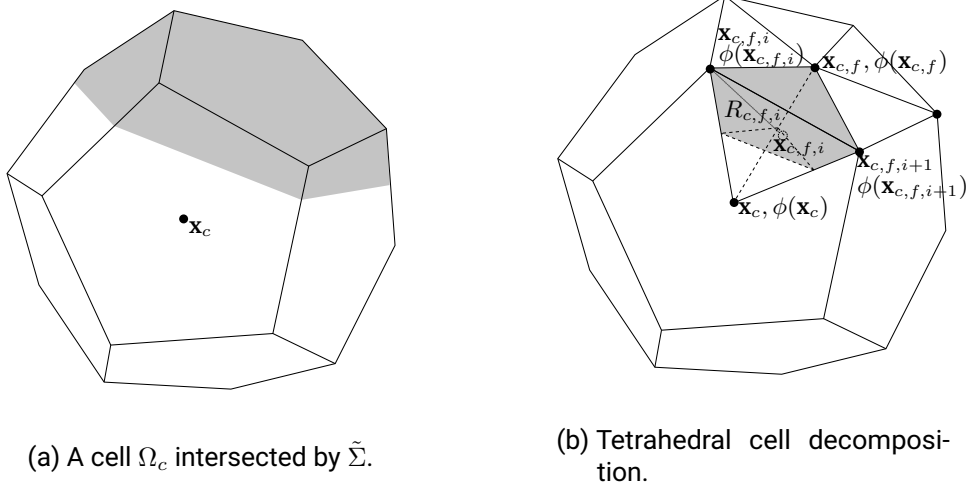


Figure 4.5.: Centroid decomposition of an interface cell into tetrahedra and calculation of  $\alpha_c$  using the SMCI/A algorithms.

algorithm 1.

#### 4.2.6. Volume fractions by adaptivity and approximation

This section presents an alternative approach to the computation of volume fractions presented in section 4.2.5. While section 4.2.5 details a method based on geometric intersections, this section introduces an algorithm based on volumetric reconstruction by adaptive mesh refinement. Detrixhe and Aslam [30] introduce a second-order accurate approximation for the volume fraction of a triangle (2D) or a tetrahedron (3D). Their model is an algebraic expression taking the signed distances  $\phi$  of the vertices as arguments.



---



---

**Algorithm 1** The Surface-Mesh / Cell Intersection Algorithm (SMCI)

---

```

1:  $\alpha_c = 0, \phi_{c,p} = 0$ 
2: Compute search radius for cell centers  $r_{c \in C}$  using eq. (4.5).
3: for cell centroids  $\{\mathbf{x}_c\}_{c \in C}$  do
4:   Place the vertices of  $\tilde{\Sigma}$  into an octree (section 4.2.2).
5:   Find the triangle  $\mathcal{T}_n \in \tilde{\Sigma}$  nearest to  $\mathbf{x}_c$  within a ball  $\mathcal{B}(\mathbf{x}_c, r_c)$ .
6:   Set  $\phi_c^g := \phi^g(\mathbf{x}_c, \mathcal{T}_n)$  using eq. (4.11).
7: end for
8: Approximately solve eq. (4.12) to propagate  $\text{sign}(\phi_c)$ .
9: Compute search radius for cell corner points  $r_{p \in P}$  using eq. (4.6).
10: Find all intersected cells  $I = \{c, \phi_c \phi_p < 0 \text{ for at least one } p\}$ .
11: Use eq. (4.14) to correct  $\phi_c$  within the narrow band.
12: Compute  $\phi_p$  in the bulk using eq. (4.15).
13: Use eq. (4.17) to correct  $\phi_p$  within the narrow band.
14: for cells  $\{\Omega_c\}_{c \in C}$  do
15:   if  $\phi_c \leq 0$  and all corner-point distances  $\phi_p \leq 0$  then            $\triangleright$  Cell is inside the negative
      $\tilde{\Sigma}$ -halfspace.
16:      $\alpha_c = 1$ 
17:   end if
18:   if cell  $\Omega_c$  is intersected,  $c \in I$  then                                $\triangleright$  Cell is intersected by  $\tilde{\Sigma}$ .
19:      $\alpha_c$  given by eq. (4.19).
20:   end if
21: end for

```

---

In contrast, here a volume fraction initialization algorithm is proposed that employs this model in combination with an adaptive tetrahedral cell decomposition and the octree-based signed distance calculation described in section 4.2. This algorithm is termed Surface-Mesh / Cell Approximation (SMCA) and it is outlined below.

The SMCA algorithm is based on the signed distance results of the SMCI algorithm introduced in section 4.2. The steps depicted in fig. 4.4a - 4.4d of the SMCI/A are used to compute  $\phi_c, \phi_p$  in the narrow band and propagate inside/outside information in the rest of the mesh points. Subsequent steps for the computation of volume fractions are displayed in fig. 4.6. First, all cells intersected by  $\tilde{\Sigma}$  are identified to reduce computational costs, as only these cells have intermediate values  $0 < \alpha_c < 1$ . This step is depicted in fig. 4.6a. Each cell for which  $\mathbf{x}_c \in \mathcal{N}(\tilde{\Sigma})$  is checked with the *bounding ball criterion*. A bounding ball (bb) is defined for a point  $\mathbf{x}_{bb} \in \Omega_c$  using  $r_{bb} = \max_{\mathbf{x} \in \Omega_c} \|\mathbf{x} - \mathbf{x}_{bb}\|_2$ . This ball is the smallest ball that contains all points of  $\Omega_c$ . This bounding ball is compared to  $\mathcal{B}(\mathbf{x}_{bb}, |\phi(\mathbf{x}_{bb})|)$ . These balls are shown in fig. 4.7, where the bounding ball is illustrated by a dashed and the other ball by a continuous line. As a general observation, if the bounding ball is contained

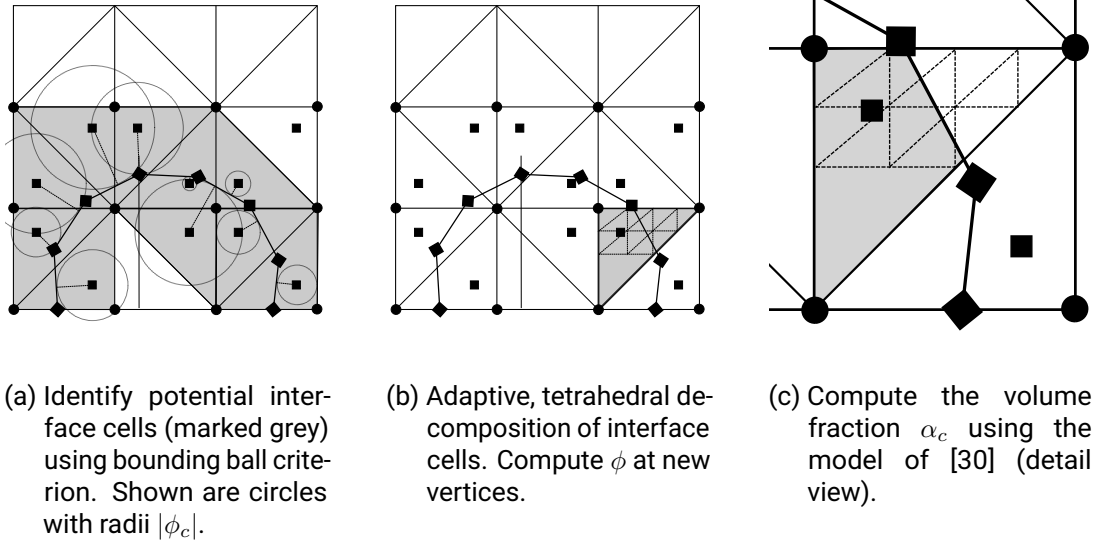


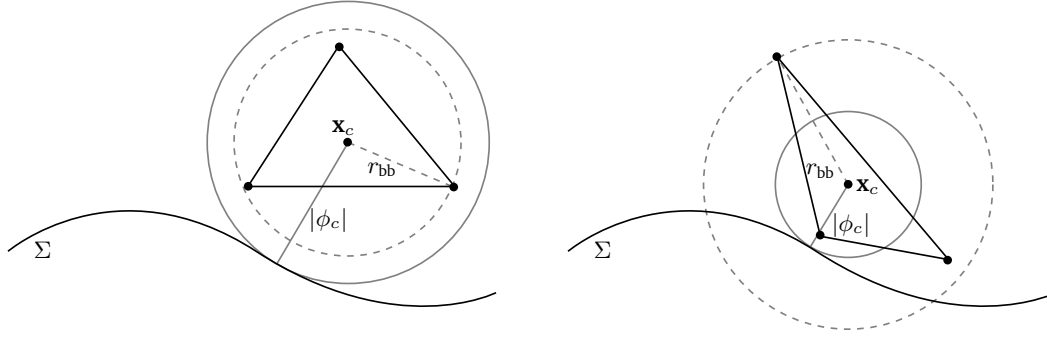
Figure 4.6.: Steps of the SMCA algorithm following signed distance computation and inside/outside propagation.

in the ball with the radius  $|\phi(\mathbf{x}_{bb})|$ , i.e.  $\mathcal{B}(\mathbf{x}_{bb}, r_{bb}) \subseteq \mathcal{B}(\mathbf{x}_{bb}, |\phi(\mathbf{x}_{bb})|)$ , then such a cell is guaranteed to be a bulk cell. This cell can then be removed from the set of cells in the narrow band to reduce the number of cells which are considered for decomposition in the next step. If the criterion is not satisfied, the cell is considered an interface cell. Two remarks on this criterion: first, the existence of such a  $\mathbf{x}_{bb}$  is not a necessary but a sufficient condition. Second, in a practical implementation evaluation of this criterion is only feasible for a small number of points when aiming to keep computational costs reasonable. Thus, the actual check is performed by evaluating

$$f_{bb}(\mathbf{x}, \phi_{\mathbf{x}}, \Omega_c) = \begin{cases} 1, & \max_{\mathbf{x}_i \in \Omega_c} \|\mathbf{x}_i - \mathbf{x}\|_2 \leq |\phi_{\mathbf{x}}|, \\ 0, & \text{otherwise} \end{cases} \quad (4.20)$$

with  $\mathbf{x} \in \Omega_c$ . The evaluation of the max-operator is based on a comparison to the corner points  $\mathbf{x}_i$  of the cell  $\Omega_c$ . For example, in the software implementation this function is only evaluated at cell centres  $\mathbf{x}_c$  (original mesh cells, see below) or cell corner points (tetrahedra resulting from decomposition). As a consequence, a few bulk cells are considered as interface cells (fig. 4.7b). This is deemed acceptable as this only has a minor impact on the computational time, but not on the computed volume fractions.

After identification of interface cells, the cell volume fractions are initialized according



(a) Bulk cell: the ball  $\mathcal{B}(\mathbf{x}_c, |\phi_c|)$  contains the cell bounding ball  $\mathcal{B}(\mathbf{x}_c, r_{bb})$ .

(b) False positive: a bulk cell which is not detected by the bounding ball criterion as  $\mathcal{B}(\mathbf{x}_c, r_{bb}) \not\subseteq \mathcal{B}(\mathbf{x}_c, |\phi_c|)$ .

Figure 4.7.: Illustration of the idea of the bounding ball criterion in 2D for clarity. The solid grey line represents  $\mathcal{B}(\mathbf{x}_c, |\phi_c|)$ , the grey dashed one  $\mathcal{B}(\mathbf{x}_c, r_{bb})$ .

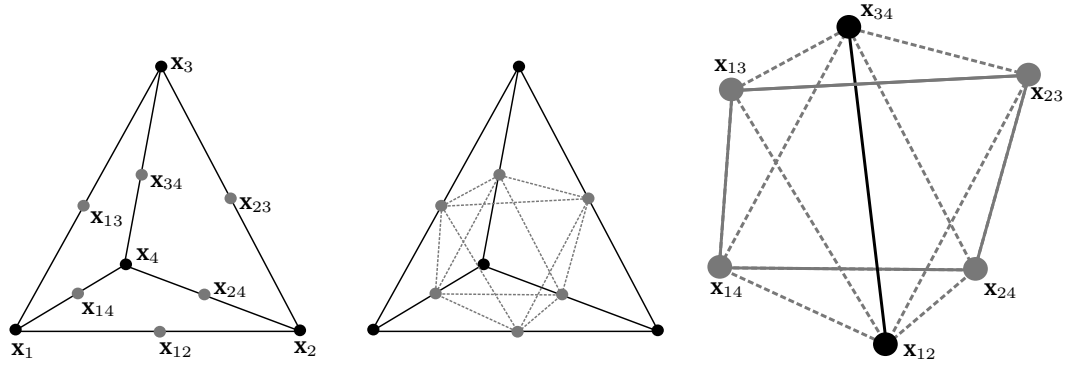
to the sign of  $\phi_c$ ,

$$\alpha_c = \begin{cases} 1, & \phi_c \leq 0, \\ 0, & \text{otherwise.} \end{cases} \quad (4.21)$$

This gives correct volume fractions for bulk cells, while the values of interface cells are updated as described below. Each cell flagged as an interface cell by the method described above is decomposed into tetrahedra using its centroid and cell face centroids as shown in fig. 4.5. Each resulting tetrahedron is further refined in an adaptive manner such that resolution is only subsequently increased where a new tetrahedron is again intersected by the interface. To achieve this, a tetrahedron  $T$  is checked with the bounding ball criterion eq. (4.20). The criterion is only evaluated at the vertex  $\mathbf{x}_{\max} \in T$  for which  $|\phi(\mathbf{x}_{\max})| = \max_{\mathbf{x} \in T} |\phi(\mathbf{x})|$ . Only if  $f_{bb}(\mathbf{x}_{\max}, \phi, T) = 0$  (eq. (4.20)),  $T$  is considered for further decomposition. An obvious choice would be decomposition at the centroid of  $T$ . However, repeated application of this approach results in increasingly flattened tetrahedra. To avoid this problem, the decomposition shown in fig. 4.8 is applied. First, from the vertices edge centres of the tetrahedron

$$\mathbf{x}_{ij} = \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j), \quad i, j \in \{1, 2, 3, 4\}, i \neq j \quad (4.22)$$

are computed (fig. 4.8a). By combining each vertex  $\mathbf{x}_i$  with the three edge centres of



- (a) Original tetrahedron with vertices ( $x_i$ , black) and edge midpoints ( $x_{ij}$ , grey).
- (b) Four tetrahedra are created by combining each vertex with its connected edge midpoints (indicated by dashed lines).
- (c) Decompose octahedron into four tetrahedra by combining each grey edge with the black line formed by two opposite points (here  $x_{12}$ ,  $x_{34}$ ).

Figure 4.8.: Decomposition of a tetrahedron into eight tetrahedra using edge midpoints.

the adjacent edges, four new tetrahedra are created (fig. 4.8b). The remainder of the original tetrahedron is an octahedron (fig. 4.8b grey dashed lines) constituted by the edge centres  $x_{ij}$ . This octahedron is decomposed into four additional tetrahedra by choosing two opposite edge centres as shown by the black line in fig. 4.8c. The indices of vertices of such a line are the numbers one to four. From the remaining four edge centres, point pairs are created such that  $\{x_{mn}, x_{mo}\}$  or  $\{x_{mn}, x_{on}\}$ , yielding four pairs. Combining each pair with  $\{x_{ij}, x_{kl}\}$  (e.g. black edge in fig. 4.8c) gives the aforementioned four tetrahedra. Subsequently,  $\phi$  is computed for the added vertices  $x_{ij}$ . The decomposition is based on the pair of edge centres that have the smallest distance between each other. Refinement is completed when a maximum refinement level  $l_{\max}$  is reached. This can either be an arbitrary prescribed value or can be computed such that the edge length of the refined tetrahedra is comparable to the edge length of surface triangles. In the latter case,

$$l_{\max} = \min_{l \in \mathbb{N}} \left( \frac{L_{\text{tet}}}{L_{\text{tri}}} < 2^l \right) \quad (4.23)$$

with  $L_{\text{tet}}$  and  $L_{\text{tri}}$  being cell specific reference lengths for tetrahedra and surface triangles,

respectively. Different choices for  $L_{\text{tet}}$  and  $L_{\text{tri}}$  are possible. Here,

$$L_{\text{tet}} = \frac{1}{n_t} \sum_{\mathbf{e} \in E_{\text{cdc}}} |\mathbf{e}|,$$

$$L_{\text{tri}} = \min_{\mathbf{e} \in E_{\tilde{\Sigma},c}} |\mathbf{e}|$$

is chosen with  $E_{\text{cdc}}$  denoting the set of edges resulting from tetrahedral decomposition of a cell  $\Omega_c$  at its centroid,  $n_t$  the number of edges in  $E_{\text{cdc}}$  and  $E_{\tilde{\Sigma},c}$  a subset of edges of  $\tilde{\Sigma}$ . The set  $E_{\tilde{\Sigma},c}$  consists of all edges of  $\mathcal{T} \in \tilde{\Sigma}$  for which  $\mathcal{T} \cap \mathcal{B}(\mathbf{x}_{\text{cp}}, r_{\text{cp}}) \neq \emptyset$ . Here,

$$\mathbf{x}_{\text{cp}} = \frac{1}{|P_{\text{cp}}|} \sum_{\mathbf{x}_i \in P_{\text{cp}}} \mathbf{x}_i,$$

$$P_{\text{cp}} := \{\mathbf{x} \in \tilde{\Sigma} : \min_{\mathbf{x}_i \in \Omega_c} \|\mathbf{x} - \mathbf{x}_i\|_2\}$$

and the radius  $r_{\text{cp}} = \max_{\mathbf{x} \in P_{\text{cp}}} \|\mathbf{x} - \mathbf{x}_{\text{cp}}\|_2$ .

Finally, after computing a tetrahedral decomposition of each interface cell, the volume fraction of a cell  $\Omega_c$  is calculated as

$$\alpha_c = \frac{1}{|\Omega_c|} \sum_{T \in T_c} \alpha(T) |\text{conv}(T)| \quad (4.24)$$

where  $T_c$  denotes the set of tetrahedra resulting from the decomposition of  $\Omega_c$  and  $|\text{conv}(T)|$  the volume of  $T$ . The volume fraction  $\alpha(T)$  is computed with the approach of [30] (eq. 7), repeated here

$$\alpha(T) = \begin{cases} 1, & \phi_4 \leq 0, \\ 1 - \frac{\phi_4^3}{(\phi_4 - \phi_1)(\phi_4 - \phi_2)(\phi_4 - \phi_3)}, & \phi_3 \leq 0 < \phi_4, \\ 1 - \frac{\phi_1 \phi_2 (\phi_3^2 + \phi_3 \phi_4 + \phi_4^2) + \phi_3 \phi_4 (\phi_3 \phi_4 - (\phi_1 + \phi_2)(\phi_3 + \phi_4))}{(\phi_1 - \phi_3)(\phi_2 - \phi_3)(\phi_1 - \phi_4)(\phi_2 - \phi_4)}, & \phi_2 \leq 0 < \phi_3, \\ -\frac{\phi_1^3}{(\phi_2 - \phi_1)(\phi_3 - \phi_1)(\phi_4 - \phi_1)}, & \phi_1 \leq 0 < \phi_2, \\ 0 & \phi_1 > 0, \end{cases} \quad (4.25)$$

where  $\phi_4 \geq \phi_3 \geq \phi_2 \geq \phi_1$  are the signed distances at the vertices  $\mathbf{x}_i$  of  $T$ . The overall approach is summarized in algorithm 2.

---

---

**Algorithm 2** The Surface-Mesh / Cell Approximation Algorithm (SMCA)

---

- 1: Follow algorithm 1 up to step 13.
  - 2: Identify interface cells (eq. (4.20))
  - 3: Set bulk  $\alpha_c$  (eq. (4.21))
  - 4: Centroid decomposition of cells into tetrahedra (fig. 4.5)
  - 5: **for**  $l \in \{1, \dots, l_{\max}\}$  **do**
  - 6:     Flag tetrahedra for further refinement (eq. (4.20))
  - 7:     Decompose flagged tetrahedra (fig. 4.8)
  - 8:     Compute  $\phi$  for new points (eq. (4.11))
  - 9: **end for**
  - 10: Compute  $\alpha_c$  for interface cells (eq. (4.24))
- 

### 4.3. Numerical results

The software implementation is available on GitLab [68]: the specific version (git tag) used to generate results described below is referred to. Detailed information on how to build and use the software is provided in the `README.md` file in the root folder of the software repository.

The difference between the total volume given by the volume fraction calculated from the surface on the unstructured mesh, and the exact volume bounded by the surface, namely

$$E_v = \frac{1}{V_e} \left| V_e - \sum_{c \in C} \alpha_c |\Omega_c| \right|, \quad (4.26)$$

is used as the measure of accuracy of the proposed algorithms. Here,  $V_e$  is the volume given by the exact surface function, or the volume that is bounded by a given surface mesh if an exact surface function is not available, e.g. in section 4.3.1. In these cases, we calculate  $V_e$  using

$$V_e = \frac{1}{3} \left| \int_{V_e} \nabla \cdot \mathbf{x} dV \right| = \frac{1}{3} \left| \int_{\partial V_e} \mathbf{x} \cdot \mathbf{n} dS \right| \quad (4.27)$$

where  $\partial V_e$  is the surface that bounds  $V_e$ . As this surface is triangulated, eq. (4.27) can be expanded further

$$V_e = \frac{1}{3} \left| \sum_{t \in 1..N_{\Sigma}} \int_{\mathcal{T}_t} \mathbf{x} \cdot \mathbf{n} dS \right| = \frac{1}{3} \left| \sum_{t \in 1..N_{\Sigma}} \int_{\mathcal{T}_t} (\mathbf{x} - \mathbf{x}_t + \mathbf{x}_t) \cdot \mathbf{n} dS \right| = \frac{1}{3} \left| \sum_{t \in 1..N_{\Sigma}} \mathbf{x}_t \cdot \mathbf{S}_t \right| \quad (4.28)$$

where  $N_{\tilde{\Sigma}}$  is the number of triangles in  $\tilde{\Sigma}$ ,  $\mathcal{T}_t \in \tilde{\Sigma}$  are triangles that form the interface mesh, and  $\mathbf{x}_t, \mathbf{S}_t$  are their respective centroids and area normal vectors.

Computing architecture	
CPU	vendor_id : AuthenticAMD cpu family : 23 model : 49 model name : AMD Ryzen Threadripper 3990X 64-Core Processor frequency : 2.90 GHz
Compiler	version : g++ (Ubuntu 10.2.0-5ubuntu1 20.04) 10.2.0 optimization flags : -std=c++2a -O3

Table 4.1.: Used computing architecture.

Table 4.1 contains the details on the computing architectures used to report the absolute CPU times in the result section. The CPU frequency has been fixed to 2.9 GHz to stabilize the CPU time measurements.

### 4.3.1. Triangulated surfaces

#### Sphere and ellipsoid

Exact initialization algorithms for spheres are available on unstructured meshes [112, 60]. The sphere and ellipsoid test cases are used to confirm the second-order convergence of SMCI/A algorithms and their applicability as a volume fraction model for the unstructured Level Set / Front Tracking method [71]. The sphere case consists of a sphere with a radius  $R = 0.15$ , and the ellipsoid half-axes are  $(0.4, 0.3, 0.2)$ . Both the sphere and ellipsoid center are at  $(0.5, 0.5, 0.5)$ , in a unit box domain. Error convergence, CPU time and additional data are publicly available [122].

#### SMCI Algorithm

Figure 4.9 shows the expected second-order convergence of the global error  $E_v$  given by eq. (4.26) on cubic fig. 4.9a and irregular hexahedral fig. 4.9b unstructured meshes. In fig. 4.9,  $N_c$  is the number of cells used along each spatial dimension of  $\tilde{\Omega}$  and  $N_T$  is the number of triangles used to resolve the sphere.

The CPU times reported in fig. 4.10 for the architecture A1 in table 4.1 show that the SMCI algorithm is a promising candidate for a volume fraction model for the unstructured

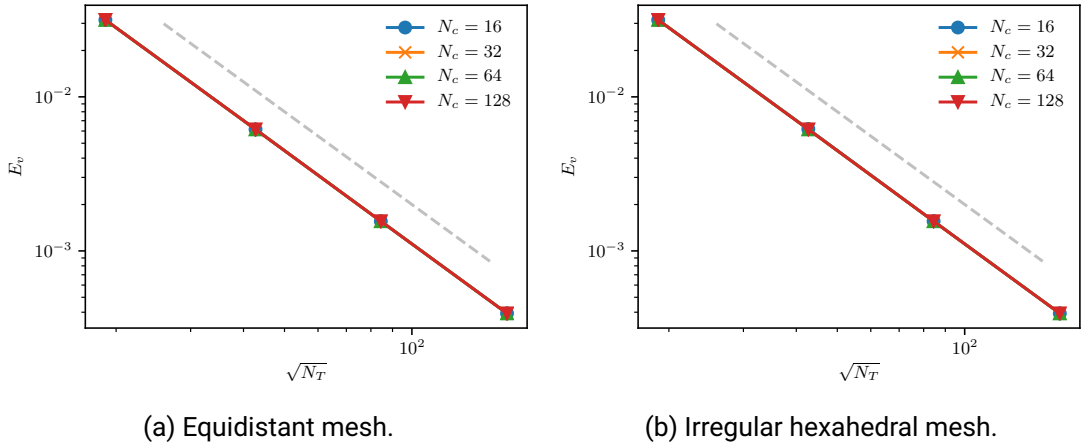


Figure 4.9.:  $E_v$  errors of the SMCI algorithm for the sphere. The grey dashed line indicates second order convergence.

Level Set / Front Tracking method. The complexity of the algorithm expressed in terms of the measured CPU time remains, linear for a constant ratio  $\sqrt{N_T}/N_c$ . The computational complexity increases to quadratic with an increasing number of triangles per cell  $\sqrt{N_T}/N_c$ : this happens when a very fine surface mesh is used to compute volume fractions on a very coarse volume mesh. An intersection between a highly resolved surface mesh and single cell of a relatively coarse mesh is shown in fig. 4.11a.

This configuration is relevant for accurate initialization of volume fractions on coarse meshes, but irrelevant for calculating the phase indicator for Front Tracking, where only a small number of triangles per multimaterial cell ( $\leq 10$ ) is present. Therefore, linear complexity of the SMCI algorithm for small ratios  $\sqrt{N_T}/N_c$  makes SMCI a potential candidate for a highly accurate geometrical volume fraction model for the unstructured Level Set / Front Tracking method. When considering the absolute CPU times, it is important to note that the SMCI algorithm has not yet been optimized for performance.

The volume error  $E_v$  for a sphere is shown in fig. 4.9b for a perturbed hexahedral mesh. An example perturbed mesh from this parameter study is shown in fig. 4.11b. The mesh is distorted by randomly perturbing cell corner points, using a length scale factor  $\alpha_e \in [0, 1]$  for the edges  $e$  that surround the mesh point.  $\alpha_e = 0.25$  is used, resulting in perturbations that are of the size of  $0.25 \times$  the edge length. This results in a severe perturbation of the mesh shown in fig. 4.11b, as well as non-planarity of the faces of hexahedral cells. Still, as shown in fig. 4.9b, SMCI retains second-order convergence, which is also the case for the initialization of the ellipsoid on the equidistant fig. 4.12 and perturbed hexahedral



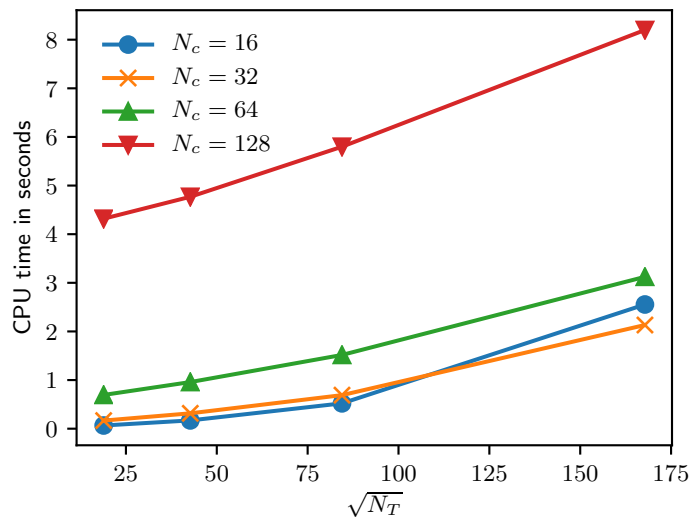


Figure 4.10.: CPU times of the SMCI algorithm for the sphere initialized on a cubic unstructured mesh.

mesh fig. 4.12b.

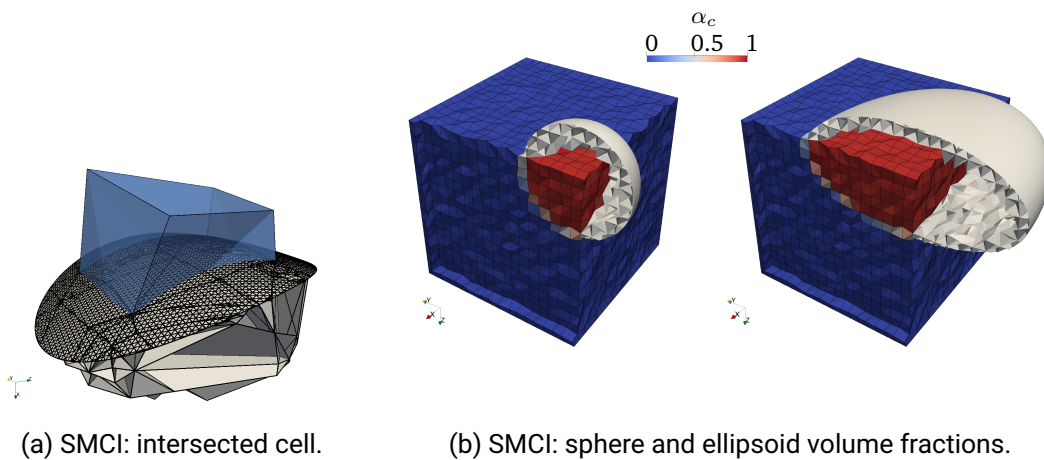


Figure 4.11.: SMCI algorithm used with a sphere and an ellipsoid on an unstructured hexahedral mesh.

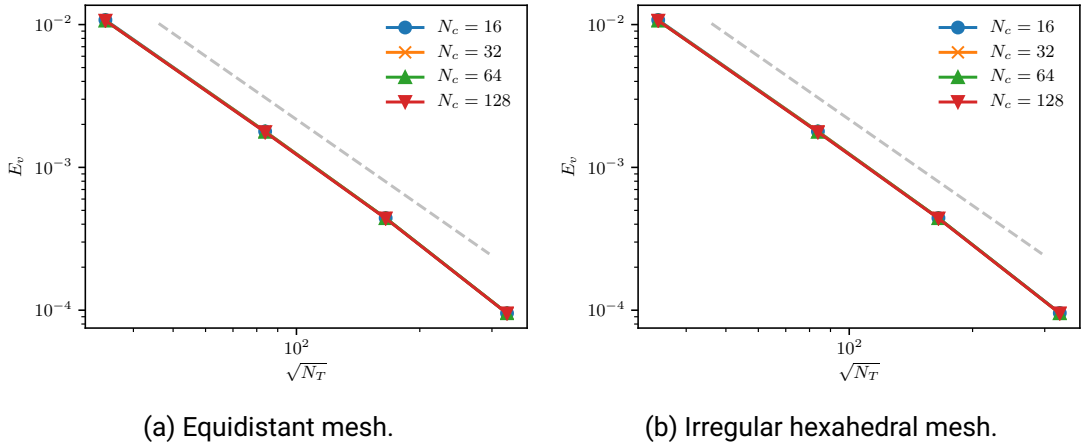


Figure 4.12.:  $E_v$  errors of the SMCI algorithm for the ellipsoid. The grey dashed line indicates second order convergence.

### SMCA algorithm

First, the effectiveness of the local adaptivity employed in the SMCA algorithm is examined with a spherical interface as described in section 4.3.1. Resolution of the volume mesh is fixed to  $N_c = 16$  cells in each direction while the sphere is resolved with  $\sqrt{N_T} \approx 410$  triangles. Maximum refinement levels  $l_{\max}$  from 0 to 3 are manually prescribed. In fig. 4.13, the resulting global volume errors  $E_v$  are displayed. This test case confirms the expected second-order convergence of  $E_v$  with adaptive refinement. An exemplary tetrahedral decomposition of a perturbed hex cell with a part of the the surface mesh is displayed in fig. 4.14. It demonstrates that the adaptive refinement based on the bounding ball criterion eq. (4.20) works as intended. Refinement is localized to the vicinity around the interface. Yet, the approach ensures all tetrahedra intersected by the interface are actually refined. The effectiveness of the local adaptive refinement compared to a uniform one becomes apparent when comparing the resulting number of tetrahedra. The adaptive approach yields around 2247 tetrahedra per interface cell on average for the spherical interface with  $\sqrt{N_T} \approx 410$ ,  $N_c = 16$  and  $l_{\max} = 3$ . A uniform decomposition, on the contrary, would result in  $M_i \times M_r^{l_{\max}} = 24 \times 8^3 \approx 47.9 \times 10^3$  tetrahedra, where  $M_i$  denotes the number of tetrahedra from initial cell decomposition and  $M_r$  the number of tetrahedra from refining a tetrahedron. Thus, the local adaptive refinement reduces the required overall number of tetrahedra by a factor of 5.5 in comparison to a uniform refinement, without affecting the accuracy.

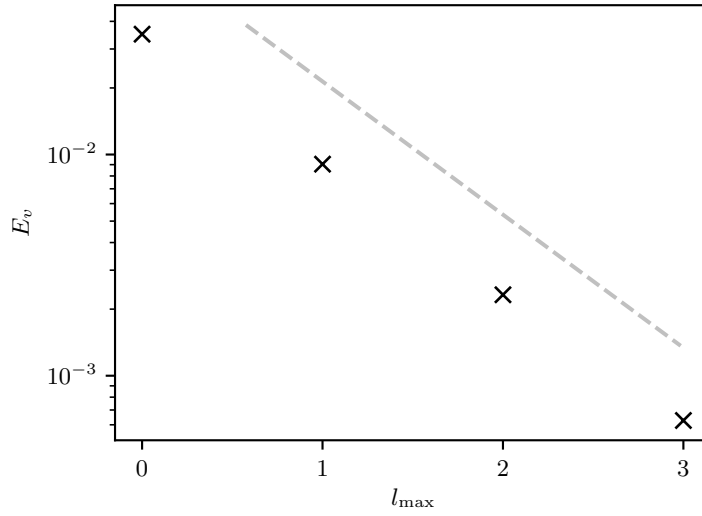


Figure 4.13.:  $E_v$  errors of the SMCA algorithm using different refinement levels  $l_{\max}$  for a sphere. Resolution of volume and surface mesh are fixed to  $N_c = 16$  and  $\sqrt{N_T} \approx 410$ . The grey dashed line indicates second order convergence.

Having verified the refinement procedure, accuracy of the SMCA algorithm and its convergence with respect to surface mesh resolution is assessed in the following. As for the SMCI algorithm, a sphere and an ellipsoid are used for this purpose. Results for the sphere in terms of the global volume error  $E_v$  (eq. (4.26)) are shown in fig. 4.15 for cubic cells (fig. 4.15a) and perturbed hexahedral cells (fig. 4.15b). Domain size, sphere centre and radius are identical to the SMCI setup as well as the perturbation factor  $\alpha_e = 0.25$ . The maximum refinement level is computed according to eq. (4.23). Both mesh types yield nearly identical results and show second-order convergence. Resolution of the volume mesh  $N_c$  has a minor influence for coarser surface meshes which vanishes for  $\sqrt{N_T} > 100$ . For the ellipsoidal interface, the errors  $E_v$  are shown in fig. 4.16. The results are qualitatively and quantitatively similar to those of the spherical interface. Absolute computational times required for the initialization of a sphere with the SMCA algorithm are displayed in fig. 4.17. Run times have been measured on the architecture listed in table 4.1. As the implementation SMCI algorithm, the implementation of the SMCA algorithm has not yet been optimized for performance.

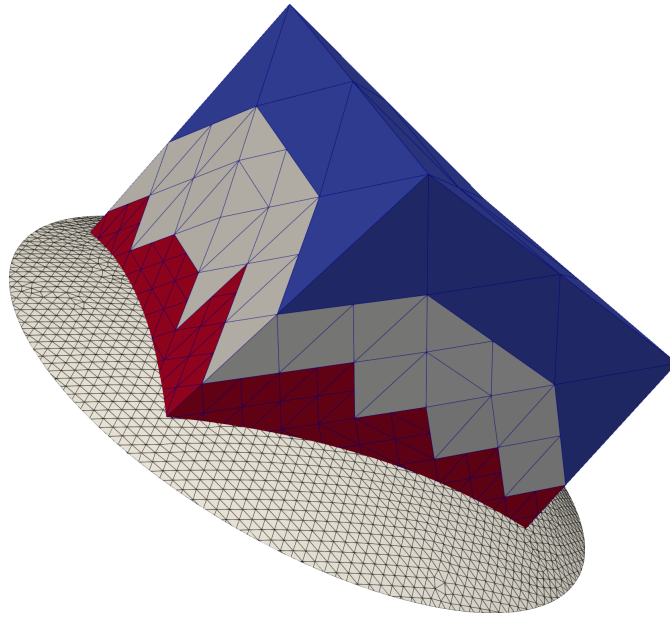
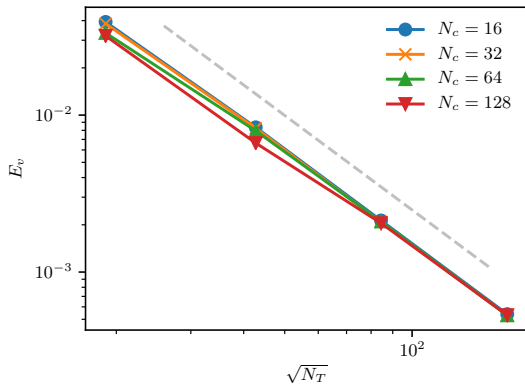


Figure 4.14.: Tetrahedral decomposition of a perturbed hex cell used to approximate  $\alpha_c$ . Tetrahedra from different refinement levels are shown in different colors (level 1: blue, level 2: grey, level 3: red). Due to adaptivity, the highest refinement level is localized in the vicinity of the surface mesh..

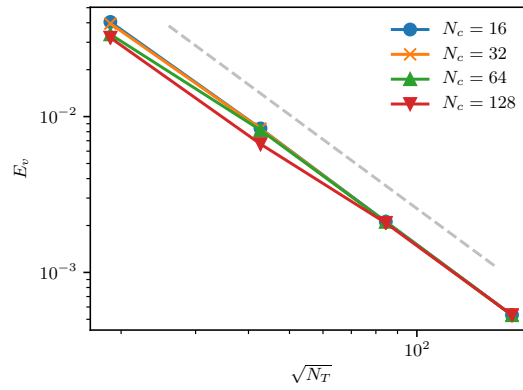
## Complex surfaces

### Surface of a fluid from an experiment

Some methods that are surveyed in section 4.1 can initialize volume fractions from exact implicit surfaces, such as a sphere or an ellipsoid, analyzed in section 4.3.1. One novelty of SMCI/A algorithms is their ability to compute volume fractions from arbitrary surfaces on arbitrary unstructured meshes. For example, volume fractions given by an experimental surface were calculated by the SMCI algorithm in [44] for studying breakup dynamics of a capillary bridge on a hydrophobic stripe between two hydrophilic stripes. In [44], the experimental setup involves a liquid bridge that is formed between two larger droplets across a hydrophobic stripe. The hydrophobic stripe drives the collapse of this liquid bridge, that is observed experimentally and in a simulation in [44]. The quantitative comparison of the simulation and the experiment from [44] is shown in fig. 4.18a. The experimental surface from [44], used to initialize volume fractions, is shown in fig. 4.18b.

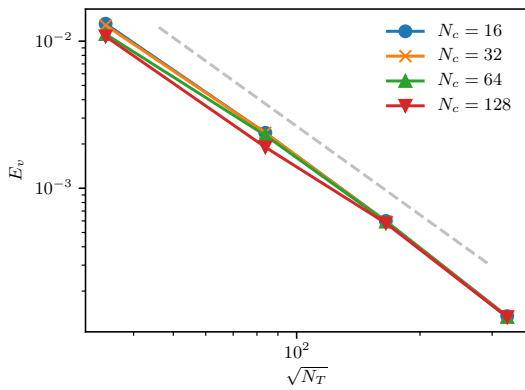


(a) Equidistant mesh.

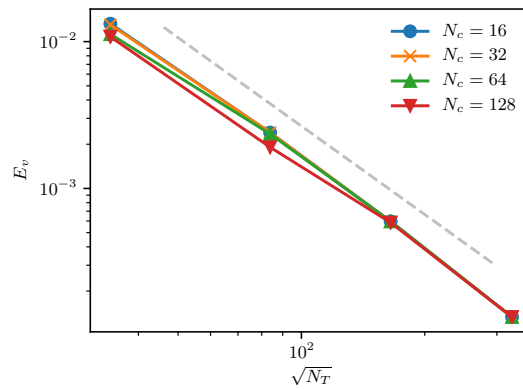


(b) Irregular hexahedral mesh.

Figure 4.15.:  $E_v$  errors of the SMCA algorithm for the sphere. The grey dashed line indicates second order convergence.



(a) Equidistant mesh.



(b) Irregular hexahedral mesh.

Figure 4.16.:  $E_v$  errors of the SMCA algorithm for the ellipsoid. The grey dashed line indicates second order convergence.

The SMCI algorithm computes the volume fractions of the experimental fluid interface from [44] with the volume error  $E_v = 7.789e-06$ . As shown in section 4.3.1, the accuracy of the initialization depends on the quality of the surface mesh, not on the resolution of the volume mesh, that is chosen in this case to appropriately resolve the hydrodynamics in [44].

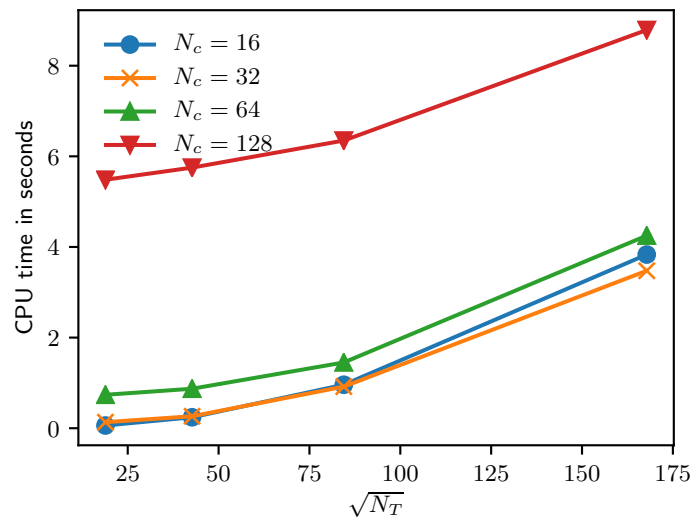
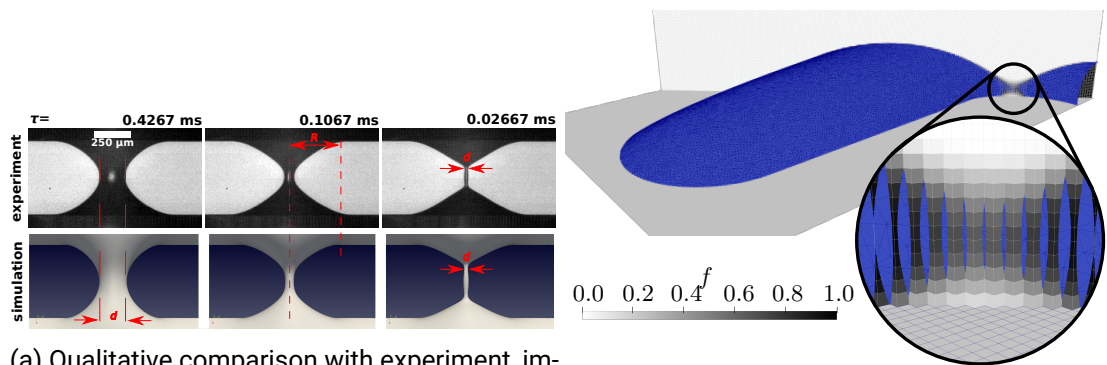


Figure 4.17.: CPU times of the SMCA algorithm for the sphere initialized on a cubic unstructured mesh.



(a) Qualitative comparison with experiment, image from [44].

(b) Initialization of volume fractions  $f$  for the wetting experiment, image adapted from [44].

Figure 4.18.: Simulation of the wetting experiment with the fluid interface given as a triangular surface mesh [44].

---

## CAD model

To demonstrate that the SMCI/A algorithms are able to handle interfaces more complex than shown above, the surface mesh from a CAD model displayed in fig. 4.19a is used. In

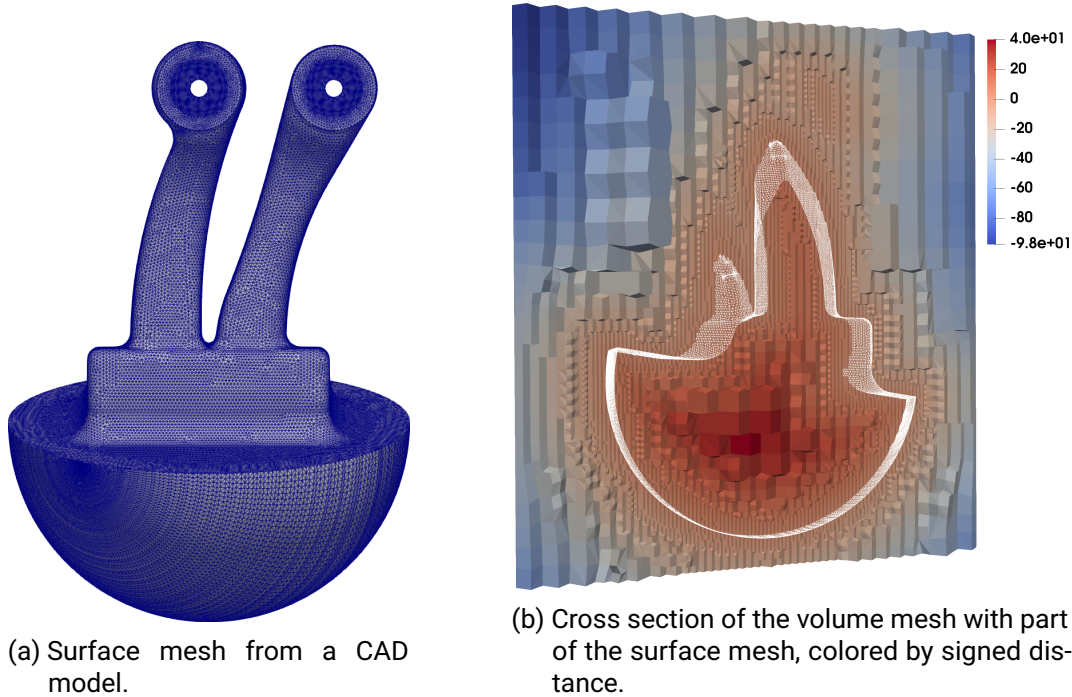


Figure 4.19.: Surface and volume mesh of the CAD model test case.

contrast to the previous interfaces, this one features sharp edges and geometric features of distinctly different sizes. The mesh for this test case has been generated with the *cartesianMesh* tool of cfMesh [57]. Refinement is used in the vicinity of the interface. This meshing procedure is chosen to obtain a mesh that closer resembles that of an industrial application than a uniform cubic mesh. A cross section of the mesh is depicted in fig. 4.19b. Before examining the computed volume fractions for this case, the signed distance calculation (section 4.2.2) and sign propagation (section 4.2.4) are verified. The presence of sharp edges (see fig. 4.19a) makes this test case more prone to false inside/outside classifications than the others shown so far. Yet the proposed procedure yields the correct sign for the distance in all cells as shown in fig. 4.20a. The enclosed volume of the surface mesh is considered as  $\Omega^+$ , thus  $\phi > 0$  for all points  $\mathbf{x} \in \Omega^+$ . As

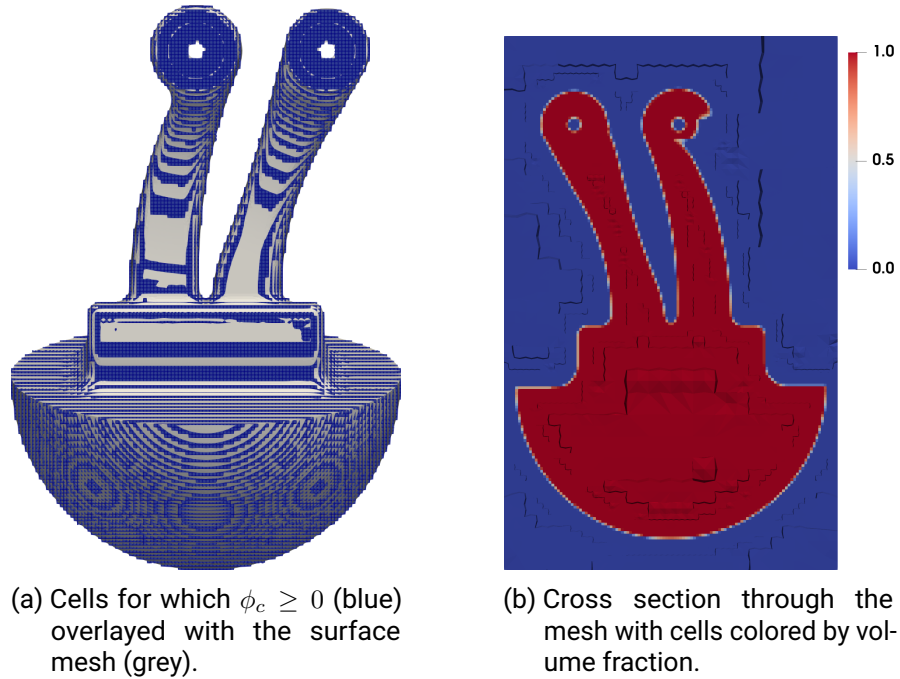


Figure 4.20.: Inside/outside computation and resulting volume fractions for the CAD geometry.

displayed in fig. 4.20a and confirmed by further manual inspection of the results, the proposed signed distance calculation correctly classifies all cells within the narrow band and robustly propagates this information to the entire domain. This is reflected in the volume fractions as computed, shown in fig. 4.20b. Bulk cells are assigned values of either 1 or 0, depending on whether they are located in  $\Omega^+$  or  $\Omega^-$  and mixed cells with  $0 < \alpha_c < 1$  are only found where the surface mesh is located. Accuracy-wise, the global errors  $E_v$  depicted in fig. 4.21 have been obtained with the SMCA algorithm using different refinement levels. As for the spherical interface (see fig. 4.13), second-order convergence is achieved, even though the surface mesh approximates a non-smooth interface here.

### 4.3.2. Implicit surfaces / Level Set surfaces

Because of the algebraic calculation of volume fractions from signed distances, the SMCA algorithm allows a direct comparison with volume fraction initialization methods on unstructured meshes that represent the fluid interface using function composition. Consid-



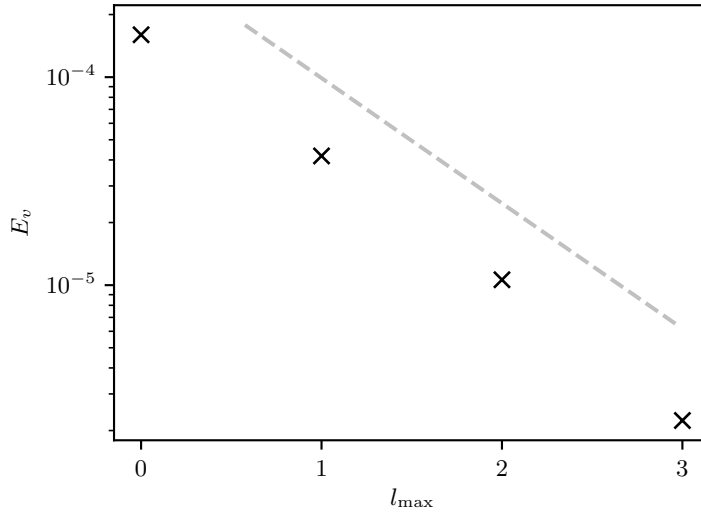


Figure 4.21.:  $E_v$  errors of the SMCA algorithm using different refinement levels  $l_{\max}$  for the CAD model with the reference volume  $V_e$  computed by eq. (4.28). The grey dashed line indicates second order convergence.

ering section 4.1, logical choices for the comparison are the methods of Ahn and Shashkov [4], Fries and Omerović [36], and Jones et al. [55]. However, Ahn and Shashkov [4] do not provide convergence results for the 3D initialization and Fries and Omerović [36] integrate a function that is  $\neq 1$  within their 3D surface, so the result of the quadrature does not correspond to the volume enclosed by the surface. Therefore, a direct comparison with Jones et al. [55] is provided, specifically Jones et al. [55, table 3].

Absolute volume errors are computed for an octant of a sphere with radius  $R = 0.5$ , placed at  $(0, 0, 0)$  within a unit-length cubical domain, and are shown in fig. 4.22. Tetrahedral unstructured meshes are generated using the Delaunay algorithm in gmsht [38], by providing a discretization length that results in a number of mesh points comparable to Jones et al. [55, table 3, No Nodes]. As shown in fig. 4.22, the accuracy of the SMCA algorithm depends on the volume mesh resolution and the number of refinement levels when an implicit (exact) sphere is used as interface description. This is expected since both parameters influence the size of the refined tetrahedra which are used to approximate the volume fraction. Consequently, the achievable accuracy is not limited by the volume mesh resolution and can be controlled through the number of refinement levels. The lowest absolute errors are in the order of magnitude of  $10^{-9}$ , achieved by SMCA using 10

refinement levels, and correspond to relative errors in the order of magnitude of  $10^{-8}$ , which is around 4 orders of magnitude lower than minimal VoF advection errors reported so far in the literature [69], and are therefore admissible as initial volume fraction values. Even higher levels of absolute accuracy, comparable to Jones et al. [55, table 3,  $\bar{\epsilon}_6, \bar{\epsilon}_9$ ], can be achieved with further refinement, with substantially increased computational expense. However, such further increase in accuracy is without significance to the volume fraction advection [69]. Contrary to the implicit (exact) sphere, resolving a sphere using a triangular mesh is more challenging, as the absolute accuracy depends on the resolution of the surface mesh. Results for spheres triangulated using the Frontal Algorithm in gmsH [38] are shown in fig. 4.22. Doubling the resolution of the surface mesh, as expected, doubles the accuracy of SMCA with triangulated surfaces as input. This approach of course does not make sense for a sphere, whose implicit (exact) function is easily defined. For geometrically complex surfaces shown below, it is important to have in mind that the resolution of the surface mesh together with the refinement level determine the absolute accuracy and computational costs.

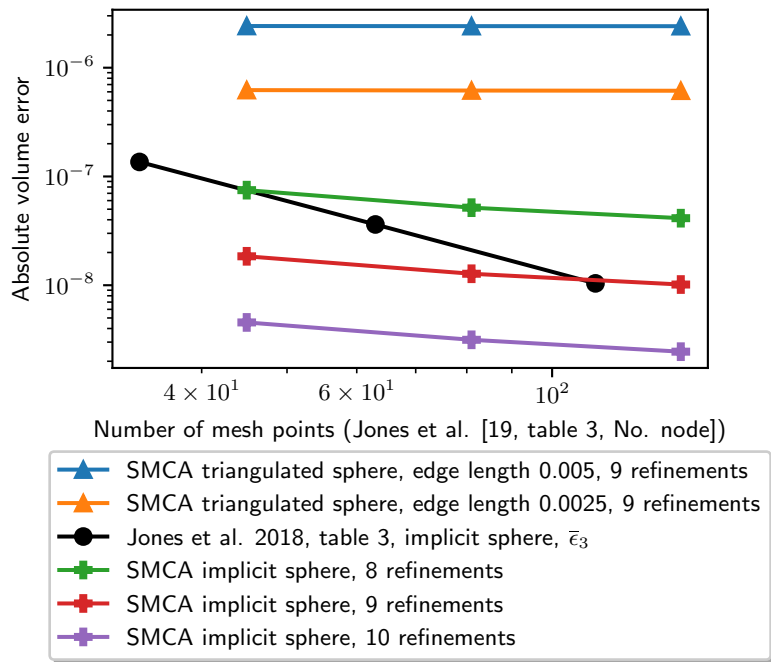


Figure 4.22.: Comparing the SMCA algorithm and Jones et al. [55, table 3] on tetrahedral meshes.

---

## 4.4. Conclusion

The proposed Surface-Mesh Cell Intersection / Approximation algorithms accurately compute signed distances from arbitrary surfaces intersecting arbitrary unstructured meshes. Geometrical calculations ensure the accuracy of signed distances near the discrete surface. The signed distances (actually their inside / outside information) are propagated into the bulk using an approximate solution of a Laplace equation. Once the signed distances are available in the full simulation domain, the SMCI algorithm computes volume fractions by intersecting arbitrarily-shaped mesh cells with the given surface-mesh, while the SMCA algorithm approximates volume fractions using signed distances stored at cell corner points. Both algorithms are robust and show second-order convergence for exact surfaces and arbitrarily shaped surface meshes. The SMCI algorithm scales linearly with a small number of surface triangles per cut-cell. Since a small number of triangles per cell is a requirement for Front Tracking, this linear-complexity makes SMCI an interesting candidate for computing volume fractions in the 3D unstructured Level Set / Front Tracking method [71], which will be the subject of future investigations.



---

## 5. Hybrid Level Set / Front Tracking method

---

At their core, numerical methods for multiphase flow simulations attempt to accurately and efficiently approximate the evolution of interfaces that form between immiscible fluid phases. An accurate, stable and efficient motion of the fluid interface in the context of multiphase flows consists of two components: the kinematics of the interface and the solution of a multiphase Navier-Stokes system.

In a previous publication [71], a new LENT hybrid Level Set / Front Tracking method was developed on unstructured meshes. In this chapter extensions to the LENT method towards two-phase flows driven by the surface tension forces are presented. For this purpose, the Segregated Accuracy-driven Algorithm for Multiphase Pressure-Linked Equations (SAAMPLE) is developed to stabilize the single-field formulation of Navier-Stokes equations on unstructured meshes.

### 5.1. Literature review

Before the new solution algorithm of the LENT method is described, it should be placed in the context of other contemporary contributions. Research of multiphase simulation methods has produced a substantial amount of scientific contributions over the years. Here, the focus is placed only on the methods that are directly or indirectly related to the hybrid Level Set / Front Tracking method.

Widely used multiphase flow simulation methods can be categorized into: Front Tracking [129, 42, 126], Level Set [103, 115, 41] and Volume-of-Fluid (VOF) [46, 98] methods. Each method has specific advantages and disadvantages with respect to the other methods. All methods are still very actively researched and a relatively recent research avenue is focused on hybrid methods. Hybrid methods are set to outperform original methods by combining their sub-algorithms, with the goal of combining strengths and avoiding weaknesses of individual methods.

A notable example is the widely used coupled Level Set and Volume-of-Fluid method (CLSVOF) [117]. CLSVOF was developed to address the disadvantage of the Volume-of-Fluid method in terms of accurate surface tension calculation and the disadvantage of the

---

---

Level Set method in terms of volume conservation. A similar hybrid method between the Moment of Fluid (MoF) method [33] and the Level Set method has been developed using a collocated solution approach and block-structured adaptive mesh refinement (AMR) [53].

A very promising hybrid method is the hybrid Level Set / Front Tracking method. Here, the Level Set method is used to simplify the handling of topological changes of the interface and improve the accuracy of the curvature approximation, while the Front Tracking method is employed for its widely known accuracy in tracking the interface.

The Front Tracking method approximates the fluid interface using a set of mutually connected lines in 2D and triangles in 3D. Coalescence and breakup change the connectivity of the Front and these operations are possibly global, because coalescence or breakup may involve interaction between arbitrary parts of the fluid interface. Global topological operations are therefore required to handle topological changes in the connectivity of the Front, and the corresponding changes in connectivity then complicate an efficient implementation. This especially concerns the efficiency of the parallel implementation of the Front Tracking method in non-periodic solution domains. More information about the Front Tracking method is available in [127].

The hybrid Level Contour Reconstruction Method (LCRM) [105, 110, 107, 109, 106] simplifies the topological changes of the interface while ensuring stability, accuracy and computational efficiency of the fluid interface motion. The connection between LCRM and the original Level Set method is the use of a signed distance field. The signed distance field is computed in the near vicinity of the Front and it is updated as the Front moves in space. A zero level set (i.e. an iso-surface) reconstruction from this distance field automatically handles topological changes of the interface. Iso-surface algorithms do not require large cell stencils, so an efficient parallel implementation can be achieved using a straightforward domain decomposition approach. Other researchers have extended the hybrid Level Set / Front Tracking method with block-adaptive structured mesh refinement (block AMR). Block AMR is applied near the interface in order to increase accuracy and reduce errors in mass conservation [21]. Hybrid Level Set / Front Tracking has also been successfully developed using the Finite-Element discretization, for fluid-solid interaction [13] and two-phase flows [14]. In this approach, the immersed Front is used as a surface onto which vertices of a 2D unstructured mesh are projected, to ensure the necessary alignment of face and interface normal vectors.

All the aforementioned Front Tracking and hybrid Level Set / Front Tracking methods are developed on structured meshes. Structured methods can employ very accurate interpolations and still maintain high computational efficiency [106]. On structured meshes, geometrically complex solution domains are often handled using the Immersed Boundary Method (IBM) [76].

Unstructured meshes greatly simplify simulations of multiphase flows in geometrically complex domains, in terms of a relatively straightforward domain discretization. However, unstructured meshes also introduce additional challenges when used with hybrid Level Set / Front Tracking methods in the context of the Finite-Volume method (FVM). To address the specific challenge of an accurate and stable solution of the two-phase Navier-Stokes system for the LENT method, the new SAAMPLE segregated solution algorithm is proposed which is outlined in the following sections.

## 5.2. Method overview

The LENT hybrid Level Set / Front Tracking method [71] is used for the evolution of the interface, and the unstructured Finite-Volume method in the OpenFOAM computational fluid dynamics platform [52, 56, 77] is used for the discretization of two-phase Navier-Stokes eqs. (2.5) and (2.13). This contribution improves the LENT method [71], in terms of the phase indicator and curvature approximation as well as the pressure-velocity coupling algorithm.

Algorithms of the LENT method and their respective improvements with respect to capillary flows are outlined in the following sections.

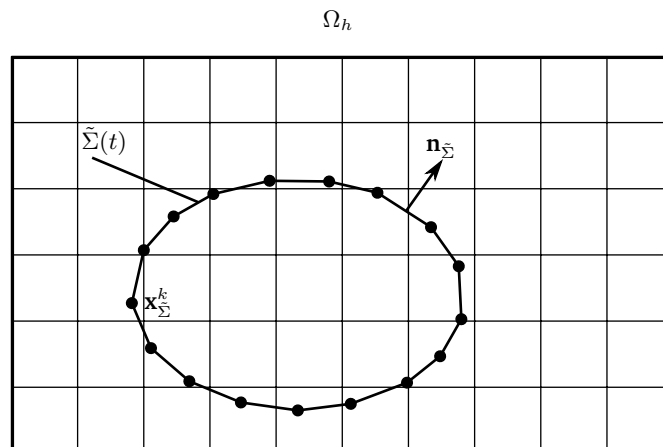


Figure 5.1.: Two-phase flow domain discretization.

The LENT method is outlined together with the SAAMPLE algorithm in fig. 5.2. Algorithms of the LENT-SAAMPLE method that are not modified with respect to the previous publication [71] are accordingly referenced. Figure 5.2 shows the difference in controlling

---

---

the convergence between the SAAMPLE and the PISO internal loop, in terms of disregarding a fixed number of iterations and relying on the pressure residual error norm. To prevent the decoupling of the acceleration from the forces acting at the interface, the cell-centered velocity is reconstructed using the operator defined by eq. (5.17).





### 5.3. Interface evolution

The solution domain  $\Omega$  is discretized into the discrete domain  $\tilde{\Omega}$  that consists of non-overlapping polyhedral finite volumes  $\Omega_c$  (cf. fig. 5.1) such that  $\tilde{\Omega} = \cup_c \Omega_c$ ,  $c \in \mathcal{C}$ , as described in section 3.1. As the initialization procedure in chapter 4, the LENT method approximates the fluid interface  $\Sigma(t)$  with a set of triangles  $\tilde{\Sigma}(t)$  (cf. fig. 5.1): the so-called *Front*. The motion of the Front is then given by a kinematic equation for each Front vertex  $\mathbf{x}_{\tilde{\Sigma}}^k$ , i.e.

$$\partial_t \mathbf{x}_{\tilde{\Sigma}}^k = \mathbf{v}(\mathbf{x}_{\tilde{\Sigma}}^k, t), \quad k \in K. \quad (5.1)$$

The velocity  $\mathbf{v}(\cdot, t)$  is obtained from the numerical solution of eqs. (2.5) and (2.13). The Front vertex  $\mathbf{x}_{\tilde{\Sigma}}^k$  does not in general coincide with mesh points used in the domain discretization. Therefore, the velocity  $\mathbf{v}(\mathbf{x}_{\tilde{\Sigma}}^k, t)$  must be interpolated from the cell centres  $\mathbf{x}_c$  to  $\mathbf{x}_{\tilde{\Sigma}}^k$ . To interpolate the velocity, the front vertices  $\mathbf{x}_{\tilde{\Sigma}}^k$  must first be located with respect to mesh cells. This is achieved by a combination of octree space subdivision, described in section 4.2.1, and known-neighborhood search algorithms [71]. For handling the topological changes of the interface, the iso-surface reconstruction by marching tetrahedra [124] is used.

The time step restriction imposed by the resolution of capillary waves renders higher-order methods unnecessary, which are usually used for the interpolation of  $\mathbf{v}(\mathbf{x}_{\tilde{\Sigma}}^k, t)$ , as well as the temporal integration of eq. (5.1). Therefore, the Inverse Distance Weighted (IDW) approximation is used for interpolation of the velocity  $\mathbf{v}(\mathbf{x}_{\tilde{\Sigma}}^k, t)$  and the explicit Euler method for the integration of eq. (5.1), same as in [71].

### 5.4. Signed distance calculation

In contrast to pure level set methods, the signed distance field  $\phi$  is not transported by solving an advection equation. Instead,  $\phi$  is computed geometrically from the front  $\tilde{\Sigma}$  each time step. For a point  $\mathbf{x}$ , it is calculated as

$$\phi(\mathbf{x}) = \text{sign}((\mathbf{x} - \mathbf{x}_{\tilde{\Sigma}, \min}) \cdot \mathbf{n}_{\mathcal{T}}) \|\mathbf{x} - \mathbf{x}_{\tilde{\Sigma}, \min}\|_2 \quad (5.2)$$

where  $\mathbf{x}_{\tilde{\Sigma}, \min} \in \tilde{\Sigma}$  is the point closest to  $\mathbf{x}$ , similar to the procedure described in section 4.2.3. The sign is determined with the normal vector  $\mathbf{n}_{\mathcal{T}}$  of the triangle which contains  $\mathbf{x}_{\tilde{\Sigma}, \min}$ . The signed distance is evaluated at cell centres and cell vertices. Since  $\phi$  is only required in the vicinity of  $\tilde{\Sigma}$ , the octree space subdivision is used to limit the computation to a narrow band  $\mathcal{N}(\tilde{\Sigma})$ . A detailed description of the procedure is given in [71].

---

## 5.5. Phase indicator approximation

As explained in section 3.3, an approximation of  $\alpha_c$  is required, amongst others to compute the material properties of cells  $\Omega : \Omega \cap \tilde{\Sigma} \neq \emptyset$  according to eqs. (3.20) and (3.21). In the previous publication [71]  $\alpha_c$  is approximated with a harmonic function of  $\phi$  which was also used in the LEFT hybrid level set / front tracking method [21]. The width of the marker field computed by this approach can be as large as the narrow band (4-5 cells) or limited to the single layer of cells that are intersected by the front.

Here, the SMCA procedure, section 4.2.6, is used for the approximation of  $\alpha_c$ . However, no additional tetrahedral refinement is employed, only a tetrahedral decomposition of  $\Omega_c$ . An approach different from section 4.2.6 to decomposition is chosen, namely the decomposition method proposed by Bloomenthal [15]. This approach only uses the cell centre  $\mathbf{x}_c$  and cell corner points  $\mathbf{x}_p$ . At these points, the signed distances are already available and computation at additional locations is avoided.

### 5.5.1. Approximation of area fractions for cell faces

The discretization of the convective term in eq. (2.13) requires the mass flux  $\dot{m}_f$  at cell faces  $f$  (see section 3.2.2). In the LENT method  $\dot{m}_f$  needs to be computed from the volumetric flux  $F_f$  and the density at the face  $\rho_f$ . While  $\rho_f$  is simply the corresponding fluid's density for faces of bulk cells, attention must be paid for faces of interface cells. Thus,  $\rho_f$  is calculated analogously to the density at cell centers eq. (2.8) by taking an area weighted average of the bulk densities

$$\rho_f = \alpha_f \rho^- + (1 - \alpha_f) \rho^+ \quad (5.3)$$

where  $\alpha_f$  denotes the fraction of face  $S_f$  wetted by the corresponding phase. The area fractions are computed in a similar fashion as the volume fractions by using the two-dimensional variant of [30]. For the sake of efficiency,  $\alpha_f$  is only computed in this way for faces intersected by the Front, i.e.  $S_f \in \Omega_c : \Omega_c \cap \tilde{\Sigma} \neq \emptyset$ .

## 5.6. Curvature approximation

The curvature  $\kappa$  of the interface  $\Sigma$  is given by

$$\kappa = -\nabla_{\Sigma} \cdot \mathbf{n}_{\Sigma}. \quad (5.4)$$

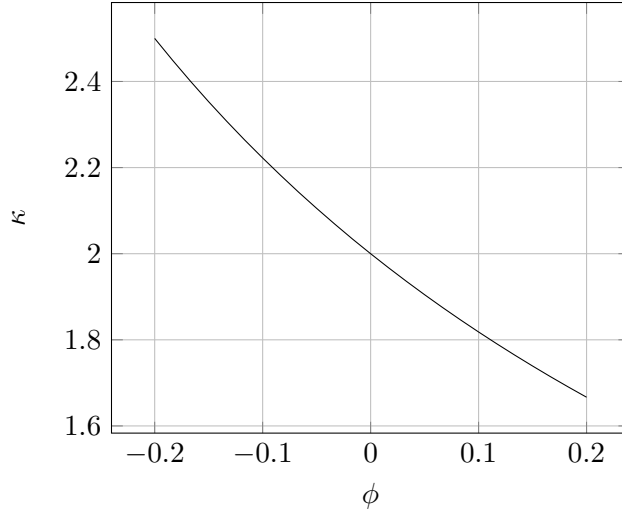


Figure 5.3.: Curvature of a sphere with  $R = 1$ , evaluated in the normal direction to the sphere, as a function of the signed distance from the sphere  $\phi$  using eq. (5.5). Obviously, the exact value  $\kappa = 2$  can only be obtained on the sphere, where  $\phi = 0$ .

With a level set field  $\psi(\mathbf{x})$  representing  $\Sigma$  at the iso-contour  $\psi(\mathbf{x}) = \psi_\Sigma$ , eq. (5.4) can be replaced by

$$\kappa = -\nabla \cdot \frac{\nabla\psi(\mathbf{x})}{|\nabla\psi(\mathbf{x})|} \quad \text{for } \mathbf{x} \in \Sigma. \quad (5.5)$$

In the context of the LENT method  $\psi$  is either the phase indicator  $\alpha$  or the signed distance  $\phi$ . For the sake of simplicity  $\psi = \alpha$  has been chosen in [71] for a preliminary coupling of LENT with the Navier-Stokes equations. The present work, however, uses  $\phi$  for the calculation of  $\kappa$ .

As pointed out in [116] and [94], eq. (5.5) does not yield the curvature of the interface if  $\mathbf{x} \notin \Sigma$ . Instead, eq. (5.5) gives the curvature of the contour that passes through the point where eq. (5.5) is evaluated. For example, at a cell center, the curvature of a contour  $\phi = \phi_c$  is computed. Thus,  $\kappa$  changes in normal direction of the interface as illustrated for a sphere in fig. 5.3. This error can be mitigated to some degree by the so-called *compact curvature calculation*.

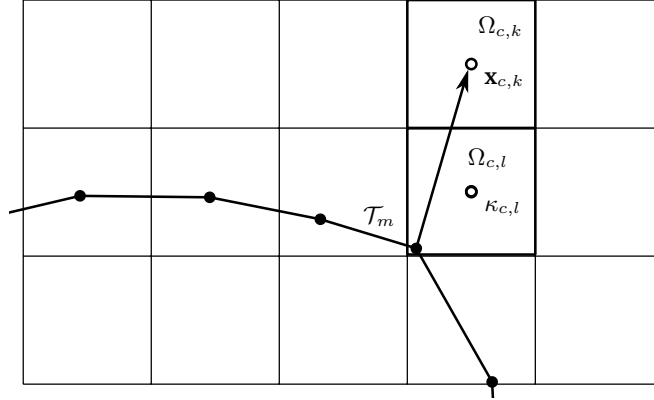


Figure 5.4.: Compact curvature calculation.

### 5.6.1. Compact curvature calculation

In [104] the authors introduce the concept of *compact curvature calculation* for their LCRM method to propagate the curvature in the narrow band as follows. A different approach to the actual curvature approximation is used in [104], so here the compact curvature calculation only is only adopted for the correction of the curvature in the narrow band of cells surrounding the Front.

Consider the example Front and narrow band configuration in fig. 5.4. Each cell center  $\mathbf{x}_c$  in the narrow band is associated with a point  $\mathbf{x}_{\min} \in \tilde{\Sigma}$  such that the distance between  $\mathbf{x}_c$  and  $\tilde{\Sigma}$  is minimal. Using this connection  $\kappa$  is first interpolated to  $\mathbf{x}_{\min}$ , and then  $\kappa(\mathbf{x}_c) = \kappa(\mathbf{x}_{\min})$  is set.

In the LENT method,  $\kappa$  is first computed using eq. (5.5). This gives a curvature field that varies along interface normal direction in general (cf. fig. 5.3). To reduce the curvature variation in the interface normal direction, only the curvature values computed at cells  $\Omega_c$  for which  $\tilde{\Sigma} \cap \Omega_c \neq \emptyset$  are kept. This information is propagated in approximate interface normal direction by combining two maps. First

$$\mathcal{M}_{\mathcal{T}}(k) : k \rightarrow \mathcal{T}_m \quad (5.6)$$

gives the closest triangle  $\mathcal{T}_m$  for each cell  $\Omega_{c,k}$  in the narrow band *not intersected by the Front*, i.e.  $\tilde{\Sigma} \cap \Omega_{c,k} = \emptyset$ . The second map

$$\mathcal{M}_c(m) : \mathcal{T}_m \rightarrow l \quad (5.7)$$

associates the triangle  $\mathcal{T}_m$  with a narrow-band cell  $\Omega_{c,l}$ , *that is intersected by the Front* and is nearest to  $\mathcal{T}_m$ . Taken together, the maps relate a non-intersected narrow band cell

$\Omega_{c,k}$  to an interface cell  $\Omega_{c,l}$ , the nearest Front-intersected narrow band cell. Now the curvature of the nearest non-intersected cell  $\Omega_{c,k}$  is set as

$$\kappa_{c,k} = \kappa_{c,l}. \quad (5.8)$$

Results presented in section 5.9.1 confirm that this compact calculation improves the curvature approximation considerably in terms of accuracy. Yet, the source of error does not vanish since  $\phi(\mathbf{x}_{c,l}) \neq 0$  in general. The maximum error can be estimated for a spherical interface of radius  $R_\Sigma$  and a cubic cell with an edge length  $h$ . The radius of the bounding ball of this cell is  $R_{\text{bb}} = \sqrt{3}h/2$ . This is also the maximum distance between the interface and an interface cell since

$$|\phi(\mathbf{x}_{c,l})| \leq R_{\text{bb}} \quad \forall \quad \Omega_c : \tilde{\Sigma} \cap \Omega_c \neq \emptyset. \quad (5.9)$$

The exact curvature of a sphere is given by

$$\kappa = \frac{2}{R}, \quad (5.10)$$

while the approximate curvature is

$$\tilde{\kappa} = \frac{2}{R_\Sigma + \phi}. \quad (5.11)$$

Thus, the relative curvature error is given by

$$e_{\kappa,\text{rel}}(\phi) = \frac{|\tilde{\kappa} - \kappa|}{\kappa} = \left| \frac{-\phi}{R_\Sigma + \phi} \right|. \quad (5.12)$$

Setting  $\phi = -R_{\text{bb}} = -\sqrt{3}h/2$  and expressing  $h = R_\Sigma/n$ , where  $n$  is the number of cells per radius, yields

$$e_{\kappa,\text{rel}}(n) = \frac{\sqrt{3}}{2n - \sqrt{3}}. \quad (5.13)$$

This indicates first order convergence of the maximum relative curvature error with respect to mesh resolution. To further reduce the curvature error an additional correction for the curvature computed at interface cells is employed.

### 5.6.2. Spherical curvature correction

Though the interface is not spherical in the general case, a correction assuming the interface to be locally spherical is proposed due to the following observations:

- (i) The proposed correction is consistent and vanishes in the limiting case  $\phi_{c,l} \rightarrow 0$ .
- (ii) Accurate curvature approximation becomes more important with more dominant surface tension often involving close to spherical interface configurations.

If the interface is assumed to be locally spherical, the curvature error introduced by the cell signed distance  $\phi_{c,l}$  can be remedied in a rather simple way. If the initial curvature  $\tilde{\kappa}_{c,l}$  is given by the compact curvature correction, eq. (5.11) can be used to compute an equivalent interface radius  $R_\Sigma$ . Inserting  $R_\Sigma$  in eq. (5.10) yields a distance-corrected curvature

$$\kappa_{c,l} = 2 \left( \frac{2}{\tilde{\kappa}_{c,l}} + \phi_{c,l} \right)^{-1}. \quad (5.14)$$

## 5.7. Surface tension computation

A semi-implicit surface tension model, proposed by Raessi et al. [96], is used which is an extension of the original Continuum Surface Force (CSF) model of Brackbill et al. [18]. In [96], the surface tension is modeled as

$$\mathbf{f}_\Sigma^{n+1} = \sigma(\kappa \mathbf{n}_\Sigma)^n \delta_\Sigma + \sigma \Delta t (\Delta_\Sigma \mathbf{v}^{n+1}) \delta_\Sigma \quad (5.15)$$

when a backward Euler scheme is used for temporal discretization.  $\Delta_\Sigma$  denotes the Laplace-Beltrami operator. As in the original CSF the approximations  $\delta_\Sigma \approx |\nabla \alpha|$  and  $\mathbf{n}_\Sigma \delta_\Sigma \approx \nabla \alpha$  are employed. To achieve an implicit discretization of the Laplace-Beltrami operator the representation

$$\Delta_\Sigma \mathbf{v} = \underline{\nabla \cdot (\nabla \mathbf{v})} - \nabla \cdot [(\mathbf{n}_\Sigma \cdot \nabla \mathbf{v}) \otimes \mathbf{n}_\Sigma] - \kappa [(\nabla \mathbf{v} - (\mathbf{n}_\Sigma \cdot \nabla \mathbf{v}) \otimes \mathbf{n}_\Sigma)] \cdot \mathbf{n}_\Sigma \quad (5.16)$$

is used. The underlined term  $\underline{\nabla \cdot (\nabla \mathbf{v})}$  is discretized implicitly as described in section 3.2.3 while the remaining terms are treated explicitly. Since an iterative approach is used to solve the pressure velocity system (see section 5.8), the converged solution is not affected.

However, the explicit contribution to the surface tension force requires additional attention. The pseudo-staggered unstructured FVM stores scalar numerical flux values at face centers. However, the solution algorithm requires cell-centered vector values for the surface tension force in the momentum equation. Therefore, the explicit surface tension force term is reconstructed from face-centered scalar flux values. The reconstruction operator that approximates  $\psi_c$  in OpenFOAM is given as

$$\psi_c^R \approx R(\psi_f) = \left[ \sum_{\tilde{f}} \mathbf{S}_{\tilde{f}} \mathbf{S}_{\tilde{f}} \right]^{-1} \cdot \sum_f \mathbf{n}_f \cdot \mathbf{S}_f \cdot \psi_f = S_c^{-1} \cdot \sum_f \mathbf{n}_f \cdot \mathbf{S}_f \cdot \psi_f, \quad (5.17)$$

where  $\psi_c^R$  is the cell-centered reconstructed vector value,  $f$  is the index of a polygonal face that belongs to the polyhedral cell  $\Omega_c$ , and  $\psi_c^R$  is the reconstructed vector value, associated with the centroid of the polyhedral cell  $\Omega_c$  and  $\mathbf{n}_f, \mathbf{S}_f$  are the respective outward oriented unit normal and area normal vector of face  $f$ . Usually, the vector quantity  $\psi_f$  is not available, otherwise it would be possible to compute  $\psi_c$  using interpolation. Instead, the product  $\psi_f \cdot \mathbf{S}_f$  is given, namely the *scalar flux* of the vector quantity  $\psi$  through the face  $S_f$ . The reconstruction operator introduces an error  $\epsilon_R$  in eq. (5.17) as

$$\psi_c = S_c^{-1} \cdot \sum_f \mathbf{n}_f \cdot \mathbf{S}_f \cdot \psi_f + \epsilon_R. \quad (5.18)$$

The reconstruction error is thus expressed as

$$\epsilon_c^R = S_c^{-1} \cdot \sum_f \mathbf{n}_f \cdot \mathbf{S}_f \cdot (\psi_c - \psi_f). \quad (5.19)$$

Equation (5.19) is the exact equation for the error introduced by the reconstruction operator given by eq. (5.17). If  $\psi_f$  is sufficiently smooth, then

$$\psi_f = \psi_c + \nabla\psi|_c \cdot \mathbf{c}\mathbf{f} + \nabla\nabla\psi|_c : (\mathbf{c}\mathbf{f}\mathbf{c}\mathbf{f}) + \dots, \quad (5.20)$$

and eq. (5.20) can be used to replace  $\psi_c - \psi_f$  in eq. (5.19).

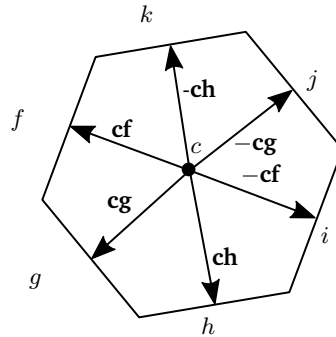


Figure 5.5.: The cell-face connectivity of a regular hexagonal cell.

The vector  $\mathbf{c}\mathbf{f} = \mathbf{x}_f - \mathbf{x}_c$  shown in fig. 5.5 connects the centroid of the cell  $c$  and the centroid of the face  $f$ . Inserting eq. (5.20) into eq. (5.19), while disregarding higher-order



terms, leads to

$$\tilde{\epsilon}_c^R = -S_c^{-1} \cdot \left( \sum_f \mathbf{n}_f \cdot \mathbf{S}_f \cdot \nabla \psi|_c \cdot \mathbf{c}\mathbf{f} + \sum_f \mathbf{n}_f \cdot \mathbf{S}_f \cdot \nabla \nabla \psi|_c : (\mathbf{c}\mathbf{f}\mathbf{c}\mathbf{f}) \right). \quad (5.21)$$

The first sum cancels out in convex orthogonal cells with an even number of faces. This can be easily shown if one considers fig. 5.5. Regular polygons and polyhedra exist, that have an even number of faces and form so-called *orthogonal unstructured meshes*<sup>1</sup>. For each face  $i$  of such cells, there exists another face  $j$ , such that  $\mathbf{c}\mathbf{i} = -\mathbf{c}\mathbf{j}$  and  $\mathbf{S}_i = -\mathbf{S}_j$ . Because these cells have an even number of faces, the first sum in eq. (5.21) can be split into two sums of equal length. This, together with the opposite sign of the  $\mathbf{S}_j$  and  $\mathbf{c}\mathbf{j}$  vectors leads to

$$\sum_f \mathbf{n}_f \cdot \mathbf{S}_f \cdot \nabla \psi|_c \cdot \mathbf{c}\mathbf{f} = \sum_i \mathbf{n}_{f,i} \cdot \mathbf{S}_i \cdot \nabla \psi|_c \cdot \mathbf{c}\mathbf{i} + \sum_j \mathbf{n}_{n,j} \cdot \mathbf{S}_j \cdot \nabla \psi|_c \cdot \mathbf{c}\mathbf{j} \quad (5.22)$$

$$= \sum_i (\mathbf{n}_{f,i} \cdot \mathbf{S}_i \cdot \nabla \psi|_c \cdot \mathbf{c}\mathbf{i} - \mathbf{n}_{f,i} \cdot -\mathbf{S}_i \cdot \nabla \psi|_c \cdot -\mathbf{c}\mathbf{i}) = \mathbf{0}. \quad (5.23)$$

The cancellation in eq. (5.23) happens for quadratic and hexagonal cells in  $2D$  and hexahedral cells with orthogonal faces, as well as regular dodecahedron cells. For slightly irregular cells (e.g. hexahedrons with slightly non-orthogonal faces), the error cancellation is partial, but it may still be strong, due to the fact that  $\mathbf{c}\mathbf{t}$  is almost, but not quite, equal to  $-\mathbf{c}\mathbf{f}$ . The canceling rate thus directly depends on the mesh non-orthogonality.

Therefore, for orthogonal meshes, the reconstruction error is given as

$$\tilde{\epsilon}_c^{R,orth} = -S_c^{-1} \sum_f \mathbf{n}_f \cdot \mathbf{S}_f \cdot \nabla \nabla \psi|_c : (\mathbf{c}\mathbf{f}\mathbf{c}\mathbf{f}). \quad (5.24)$$

Equation (5.24) shows that the reconstruction of the surface tension force is second-order accurate on orthogonal unstructured meshes: the reconstruction error  $\tilde{\epsilon}_c^{R,orth}$  is zero if the vector field is linear. On non-orthogonal meshes, the  $\mathbf{c}\mathbf{f}$  vectors do not cancel out and the reconstruction may deteriorate to first order of accuracy, given by eq. (5.19), depending on mesh non-orthogonality.

## 5.8. Pressure velocity coupling

In order to solve the discretized pressure-velocity system a new segregated solution algorithm based on the PISO approach [49] is developed. An overview of the different

<sup>1</sup>A mesh is orthogonal if the face normal vectors are collinear with the line segments between face-neighboring cell centroids.

pressure-correction methods and how they are related can be found in [25]. Barton [12] compares several PISO and solution procedures based on the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) in terms of accuracy, robustness and computational efficiency. He concludes that PISO is the preferred algorithm for transient flows considering all metrics. This is the reasoning behind using the PISO algorithm in [71], however with  $n = 4$  pressure correction iterations and followed by an additional solution of the momentum equation with the updated pressure.

Given that PISO has originally been proposed as a solution procedure for single-phase flows, some drawbacks become apparent when it is used for two-phase flows. There is no control over the solution accuracy because the PISO algorithm is controlled by a fixed iteration count. How this may manifest, is demonstrated in section 5.9.3. In the OpenFOAM framework, used to develop the LENT method, the explicit velocity update involves the use of the reconstruction operator given by eq. (5.17), that introduces additional errors given by eq. (5.24). Another problem that is emphasized by multiphase flows is the inability of the PISO algorithm to account for the non-linearity of the convective term. The main contribution of this work is the alleviation of these issues in the context of surface-tension driven multiphase flows.

Here, a new PISO-based solution algorithm termed SAAMPLE (Segregated Accuracy-driven Approach for Multiphase Pressure Linked Equations) is proposed that overcomes these disadvantages. Additionally, SAAMPLE avoids the use of case-dependent parameters like under-relaxation factors of the original SIMPLE method [89]. SAAMPLE is outlined in algorithm 3. It employs the same equations as the original PISO algorithm, namely a momentum predictor (eq. (5.25)), a pressure correction equation (eq. (5.26)) and an explicit velocity update (eq. (5.27)), given here in a semi-discrete form as

$$a_c^{\mathbf{v}} \mathbf{v}_c + \sum_{n(c)} a_n^{\mathbf{v}} \mathbf{v}_n = \mathbf{b}_c^{\mathbf{v}} - V_c (\nabla p)_c, \quad (5.25)$$

$$\mathbf{v}_c^* + \mathbf{H}_c[\mathbf{v}^*] = -\mathbf{D}_c^{\mathbf{v}} (\nabla p^{\text{prev}})_c + \mathbf{B}_c^{\mathbf{v}},$$

$$\nabla \cdot (\mathbf{D}_c^{\mathbf{v}} (\nabla p^*)_c) = \nabla \cdot (\mathbf{H}_c[\mathbf{v}^*] + \underline{\mathbf{H}_c[\mathbf{v}']} - \mathbf{B}_c^{\mathbf{v}}), \quad (5.26)$$

$$\mathbf{v}^{**} = -\mathbf{H}_c[\mathbf{v}^*] - \mathbf{D}_c^{\mathbf{v}} (\nabla p^*)_c + \mathbf{B}_c^{\mathbf{v}}, \quad (5.27)$$

in which

$$(\nabla p)_c = \frac{1}{|\Omega_c|} \int_{\Omega_c} \nabla p dV, \quad \mathbf{H}_c[\mathbf{v}] = \frac{1}{a_c^{\mathbf{v}}} \sum_{n(c)} a_n^{\mathbf{v}} \mathbf{v}_n, \quad \mathbf{B}_c^{\mathbf{v}} = \frac{\mathbf{b}_c^{\mathbf{v}}}{a_c^{\mathbf{v}}}, \quad \mathbf{D}_c^{\mathbf{v}} = \frac{V_c}{a_c^{\mathbf{v}}}.$$

As in PISO, the underlined term  $\underline{\mathbf{H}_\Omega[\mathbf{v}]}$  in eq. (5.26) is neglected as the velocity corrections  $\mathbf{v}'$  are unknown.

---

**Algorithm 3** Pseudo code of the SAAMPLE algorithm for the segregated solution of the pressure-velocity system. The operator ‘:=’ denotes assignment.

---

```

1: conv-vol-fluxes := False
2: pU-converged := False
3:  $K := 0$ 
4:  $\mathbf{v} := \mathbf{v}^n$ 
5:  $p := p^n$ 
6:
7: while not pU-converged and  $K < K_{\max}$  do
8:   if not conv-vol-fluxes then
9:     Update mass flux:  $\dot{m}_f := \rho_f F_f$  (eq. (5.3))
10:  end if
11:  conv-vol-fluxes := eq. (5.28)
12:  Solve momentum predictor eq. (5.25):  $\mathbf{v} := \mathbf{v}^*$ 
13:
14:   $I := 0$ 
15:  correct-pressure := True
16:  while correct-pressure and  $I < I_{\max}$  do
17:    Setup pressure-correction eq. (5.26)
18:    Compute  $r$  as norm of initial residual  $\mathbf{r}$ 
19:    if  $r > \text{tol}_s$  then
20:      Solve for  $p$ :  $p := p^*$ 
21:      Update volumetric fluxes  $F_f$ 
22:      Explicit velocity update eq. (5.27):  $\mathbf{v} := \mathbf{v}^{**}$ 
23:    else
24:      correct-pressure := False
25:      if  $I = 0$  and conv-vol-fluxes then
26:        pU-converged := True
27:      end if
28:    end if
29:     $I := I + 1$ 
30:  end while
31:   $K := K + 1$ 
32: end while
33:
34:  $\mathbf{v}^{n+1} := \mathbf{v}$ 
35:  $p^{n+1} := p$ 

```

---

Contrary to PISO, SAAMPLE is an iterative algorithm that is driven by the solution accuracy. It consists of two nested loops with specific purposes. The outer loop updates the mass fluxes  $\dot{m}_f$  as long as the function

$$\text{conv}(F_f) = \begin{cases} 1, & \text{if } L_\infty \left( \left| F_f - F_f^{\text{prev}} \right| / L_\infty(F_f) \right) < \text{tol}_{\text{rel}} \\ 1, & \text{if } L_\infty \left( \left| F_f - F_f^{\text{prev}} \right| \right) < \text{tol}_{\text{abs}} \\ 0, & \text{otherwise} \end{cases} \quad (5.28)$$

evaluates to 0. The parameters  $\text{tol}_{\text{rel}}$  and  $\text{tol}_{\text{abs}}$  are prescribed tolerances for the relative and absolute change of  $F_f$  between two consecutive outer iterations. Subsequently, the momentum predictor eq. (5.25) is solved with a known pressure field, either from a previous time step or previous outer iteration.

The inner loop performs the pressure correction to enforce discrete volume conservation. This is achieved with a series of corrector steps as in the original PISO algorithm [49]. First, eq. (5.26) is solved implicitly for  $p$ . The Laplacian operator on the left hand side is discretized using surface normal gradients at each cell face as implemented in OpenFOAM (see sections 3.2.3 and 3.2.4). Subsequently,  $\mathbf{v}$  is updated explicitly according to eq. (5.27). As reported in [25], this removes the need for underrelaxation as each corrector iteration partly recovers the neglected term of eq. (5.26). This agrees with the findings of Venier et al. [132]. They investigate the stability of the PISO algorithm using Fourier analysis and conclude that more corrector iterations provide a stronger coupling of pressure and velocity. Iteration of the inner loop is stopped if either the initial residuals of the pressure equation are below a prescribed threshold  $\text{tol}_{\text{ls}}$  or the maximum number of inner iterations  $I_{\text{max}}$  is exceeded.

The overall algorithm is considered converged when condition eq. (5.28) has been fulfilled and for the initial residual  $r$  of the pressure equation in the first iteration of the inner loop  $r < \text{tol}_{\text{ls}}$  is fulfilled. It means that  $\mathbf{v}^*$  obtained from momentum predictor eq. (5.25) satisfies  $\nabla \cdot \mathbf{v}^* < \text{tol}_{\text{ls}}$  in a discrete sense. If convergence is not reached within  $K_{\text{max}}$  outer iterations, the current fields for  $\mathbf{v}$  and  $p$  are considered as solutions for the time step.

## 5.9. Verification and validation results

The following sections show the improvements achieved in terms of curvature approximation, surface tension calculation and pressure velocity coupling on standard validation cases found in the literature. Because the quality of the Front is improved significantly

---

---

by smoothing [61], the reconstruction of the interface can be avoided for small interface deformations.

Research data containing the numerical results are publicly available for: vector field reconstruction in OpenFOAM <sup>2</sup>, SAAMPLE algorithm data for the stationary droplet and low amplitude oscillation, <sup>3</sup>, and the validation of the SAAMPLE algorithm with large amplitude droplet oscillations <sup>4</sup>.

While uniformly resolved meshes are used for curvature and surface tension force reconstruction test cases, statically refined meshes are used for the hydrodynamic test cases to reduce the total number of cells. The adaptive meshing procedure is described in algorithm 4. It starts from a coarse Cartesian mesh, e.g.  $n = 4$  cells in each spatial direction and a box shaped region  $bb$ . This region is chosen such that it contains the interface  $\delta_\Sigma$  and cells with  $|\mathbf{f}_\Sigma| \neq 0$  over the simulated time. All cells which are at least partly located in  $bb$  are refined by bisecting their edges. Thus, each refined cell is replaced by eight cells. This procedure is repeated until the cell size in  $bb$  is equivalent to  $n_e$  cells along a spatial direction of  $\Omega$ . In fig. 5.6, an exemplary mesh with the initial interface is shown for stationary, translating and oscillating droplet. Note that the refinement process also ensures that the size of neighbouring cells only differs by one refinement level.

---

**Algorithm 4** Generation procedure for statically refined hexahedral meshes. The operator  $':='$  denotes assignment.

---

```
1:  $m :=$  coarse initial mesh
2:  $bb :=$  boxed shaped region,  $bb \subset \Omega$ ,  $\tilde{\Sigma}(t) \cap bb = \tilde{\Sigma}(t) \forall t \in [t_0, t_{\text{end}}]$ 
3:  $n :=$  coarse initial resolution
4:  $n_e :=$  fine resolution
5: while  $n_c \leq n_f$  do
6:   for cell  $\Omega_c$  in  $m$  do
7:     if  $\Omega_c \cap bb \neq \emptyset$  then
8:       Refine  $\Omega_c$ : split into eight new cells
9:     end if
10:  end for
11:   $n := 2n$ 
12: end while
```

---

<sup>2</sup><http://dx.doi.org/10.25534/tudatalib-61>

<sup>3</sup><http://dx.doi.org/10.25534/tudatalib-62>

<sup>4</sup><http://dx.doi.org/10.25534/tudatalib-136>

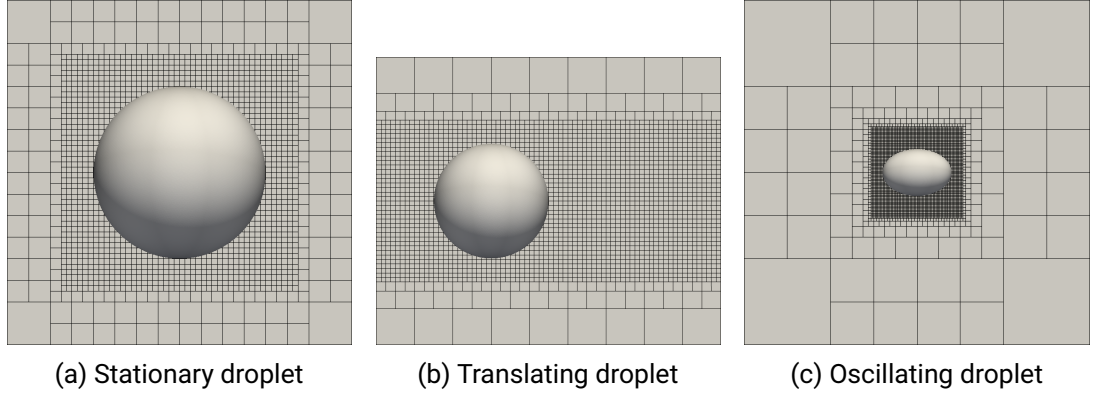


Figure 5.6.: Crosssection of statically refined 3D meshes for hydrodynamic test cases with initial interface. The equivalent resolutions from left to right are  $n_e = 64$ ,  $n_e = 64$  and  $n_e = 256$ .

### 5.9.1. Curvature calculation

Due to the important role of curvature approximation for the computation of surface tension the accuracy of the techniques described in section 5.6 is investigated here. The following test setup has been used to obtain the results presented in this section: A cubic domain  $\Omega : [0, 0, 0] \times [4, 4, 4]$  is used, discretized with  $n \in [16, 32, 64, 128]$  cells in each spatial direction. Two interface geometries, a sphere with radius  $R = 1$  and an ellipsoid with semi-axes  $\mathbf{s} = [3/2, 1, 1/2]$ , are employed. While the radius / semi-axes are kept constant, the interface centroids are generated randomly in a box-shaped region. The center of this region coincides with the center of  $\Omega$  and its size is chosen such that the narrow band does not touch or intersect the domain boundary. To examine the influence of the signed distance calculation, both the exact signed distance and the signed distance computed from the front are used in different setups. Accuracy of the different curvature models is evaluated with two norms of the relative curvature error

$$L_\infty(e_{\kappa,rel}) = \max_i \left( \frac{|\kappa_i - \kappa_{exact}|}{|\kappa_{exact}|} \right) \quad (5.29)$$

and

$$L_2(e_{\kappa,rel}) = \frac{1}{|\kappa_{exact}|} \sqrt{\frac{1}{m} \sum_i (\kappa_i - \kappa_{exact})^2} \quad (5.30)$$

where index  $i$  denotes all cell faces at which the surface tension is evaluated and  $m$  is the number of such faces. The face centers are chosen as evaluation locations rather than the

acronym	input field	compact calculation	spherical correction
DG( $\alpha$ )	$\alpha$	no	no
DG( $\phi$ )	$\phi$	no	no
cDG( $\phi$ )	$\phi$	yes	no
sccDG( $\phi$ )	$\phi$	yes	yes

Table 5.1.: Acronyms of the tested curvature model configurations. Compact calculation refers to the modification described in section 5.6.1 and spherical correction to eq. (5.14).

cell centers since the surface tension is discretized at the cell faces. Each setup (resolution, interface shape, signed distance calculation procedure, curvature model) is repeated 20 times with random placement as described above.

The results for the  $L_2$ -norm are depicted as scatter plots in fig. 5.7 and the four different configurations are summarized in table 5.1.

Several conclusions can be drawn from these plots. First of all, scattering range is small compared to the error differences between different resolutions and different models. The only exception are the cDG( $\phi$ ) and the sccDG( $\phi$ ) model for the ellipsoidal interface where there is an overlap of data points. Furthermore, the accuracy of the models increases in the order as they are listed in table 5.1 for all setups. While the signed distance as input field significantly improves the accuracy compared to the phase indicator field, the qualitative behavior is left unchanged: both models show convergence of decreasing order up to  $n = 64$ . Further increase of the resolution does not reduce the  $L_2$ -norm (DG( $\alpha$ )) or reveals the onset of divergence (DG( $\phi$ )).

Applying the compact curvature calculation idea of [104] described in section 5.6.1 yields higher absolute accuracy and consistent convergence behavior. Order of convergence lies in-between one and two for a sphere which agrees with the estimation of the maximum error eq. (5.13).

For the spherical correction approach eq. (5.14), two observations can be made. First, there is no negative impact on the accuracy when applied to the non-spherical, ellipsoidal interface. Second, as can be expected for a spherical interface, the errors are reduced by an order of magnitude or more compared to the compact calculation without correction.

Finally, comparison of fig. 5.7a and fig. 5.7b shows the impact of the signed distance calculation on the curvature calculation when the sccDG( $\phi$ ) model is used. For the curvature models DG( $\alpha$ ) and sccDG( $\phi$ ) which are used for the hydrodynamic test cases the results are summarized in table 5.2 and table 5.3. They illustrate the distinct improvement of both the  $L_2$ - and  $L_\infty$ -norm compared to the previous publication [71].

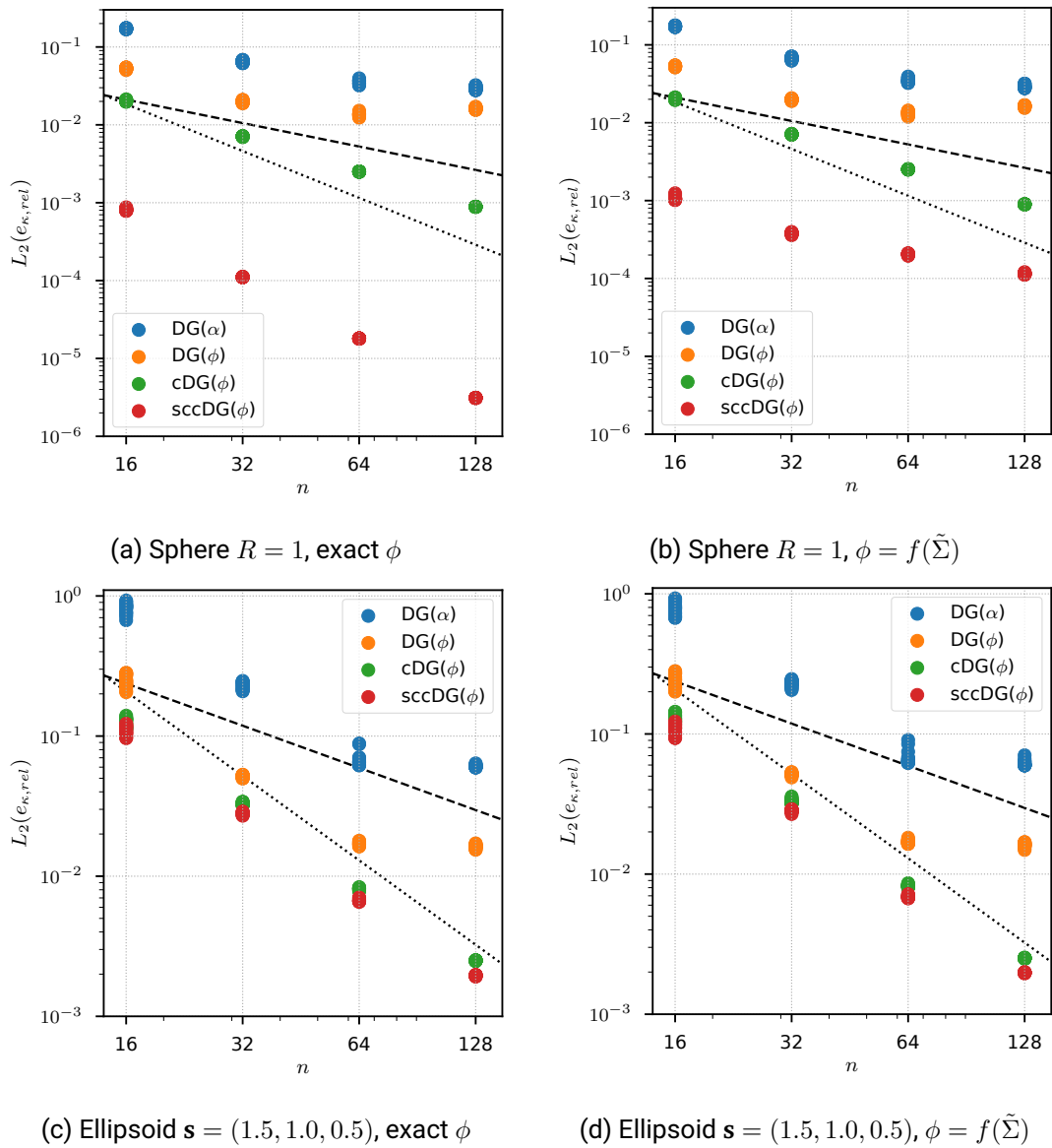


Figure 5.7.:  $L_2$  norm of the relative curvature error for a sphere (upper row) and an ellipsoid (lower row) for different curvature models. The exact signed distance (left column) and the signed distance computed from the front (right column) have been used as input. A cubic domain  $\Omega : [0, 0, 0] \times [4, 4, 4]$  with  $n$  cells in each spatial direction. Each setup has been simulated 20 times by setting the centroid of the interface at a random position around the domain center. The dashed/dotted line indicates first/second order of convergence.



model	interface	$n$	$\overline{L_\infty}(e_{\kappa,\text{rel}})$	$\overline{L_2}(e_{\kappa,\text{rel}})$	$\sigma(L_\infty(e_{\kappa,\text{rel}}))$	$\sigma(L_2(e_{\kappa,\text{rel}}))$
sccDG( $\phi$ )	ellipsoid	16	1.17e+00	1.10e-01	1.20e-01	7.99e-03
		-	0.83	1.97	-	-
		32	6.58e-01	2.80e-02	9.97e-02	4.99e-04
		-	1.19	2.05	-	-
		64	2.88e-01	6.74e-03	2.42e-02	1.41e-04
		-	1.16	1.79	-	-
	sphere	128	1.29e-01	1.95e-03	4.34e-03	1.46e-05
		-	9.37e-03	8.14e-04	5.16e-04	2.27e-05
		-	2.69	2.87	-	-
		32	1.45e-03	1.11e-04	2.21e-05	5.06e-07
		-	2.32	2.62	-	-
		64	2.91e-04	1.80e-05	1.67e-06	5.84e-08
-	2.10	2.53	-	-		
-	128	6.79e-05	3.12e-06	2.04e-07	5.10e-09	
DG( $\alpha$ )	ellipsoid	16	1.42e+01	8.19e-01	3.52e+00	6.78e-02
		-	0.86	1.84	-	-
		32	7.85e+00	2.29e-01	8.49e-01	1.30e-02
		-	0.10	1.77	-	-
		64	7.32e+00	6.72e-02	4.32e+00	7.56e-03
		-	-0.59	0.14	-	-
	sphere	128	1.10e+01	6.09e-02	2.09e+00	1.25e-03
		-	2.14e+00	1.72e-01	3.00e-01	2.19e-03
		-	-0.29	1.40	-	-
		32	2.62e+00	6.50e-02	8.08e-01	2.32e-03
		-	-0.93	0.87	-	-
		64	5.00e+00	3.56e-02	1.81e+00	2.02e-03
-	-0.63	0.27	-	-		
-	128	7.74e+00	2.96e-02	4.48e+00	1.44e-03	

Table 5.2.: Mean error norms ( $\overline{L_\infty}$  and  $\overline{L_2}$ ) of the curvature and their standard deviation  $\sigma$  for the DG( $\alpha$ ) and sccDG( $\phi$ ) model using an exact signed distance. Between resolutions, the order of convergence is displayed.

model	interface	$n$	$\overline{L_\infty}(e_{\kappa,\text{rel}})$	$\overline{L_2}(e_{\kappa,\text{rel}})$	$\sigma(L_\infty(e_{\kappa,\text{rel}}))$	$\sigma(L_2(e_{\kappa,\text{rel}}))$
sccDG( $\phi$ )	ellipsoid	16	1.12e+00	1.09e-01	1.59e-01	9.13e-03
		-	0.76	1.95	-	-
		32	6.60e-01	2.83e-02	8.38e-02	5.74e-04
		-	1.13	2.03	-	-
		64	3.02e-01	6.91e-03	2.68e-02	1.34e-04
		-	1.22	1.80	-	-
	sphere	128	1.30e-01	1.98e-03	2.86e-03	9.53e-06
		-	8.45e-03	1.07e-03	5.67e-04	5.39e-05
		-	0.46	1.50	-	-
		32	6.16e-03	3.79e-04	4.96e-04	8.87e-06
		-	-0.01	0.87	-	-
		64	6.20e-03	2.05e-04	1.12e-03	3.73e-06
DG( $\alpha$ )	ellipsoid	-	-0.29	0.83	-	-
		128	7.59e-03	1.15e-04	1.48e-03	2.44e-06
		16	1.31e+01	7.78e-01	4.23e+00	7.97e-02
		-	0.78	1.77	-	-
		32	7.61e+00	2.28e-01	9.29e-01	1.22e-02
		-	0.1	1.77	-	-
	sphere	64	7.08e+00	6.69e-02	4.21e+00	7.75e-03
		-	-0.69	0.12	-	-
		128	1.14e+01	6.14e-02	4.39e+00	2.78e-03
		16	2.23e+00	1.73e-01	3.02e-01	2.54e-03
		-	-0.53	1.37	-	-
		32	3.22e+00	6.68e-02	8.69e-01	2.49e-03
sphere	-	-0.44	0.93	-	-	
	64	4.36e+00	3.51e-02	1.62e+00	1.94e-03	
	-	-0.73	0.26	-	-	
	128	7.23e+00	2.93e-02	4.22e+00	1.35e-03	

Table 5.3.: Mean error norms ( $\overline{L_\infty}$  and  $\overline{L_2}$ ) of the curvature and their standard deviation  $\sigma$  for the DG( $\alpha$ ) and sccDG( $\phi$ ) model using a signed distance computed from the front. Between resolutions, the order of convergence is displayed.

## 5.9.2. Surface tension force reconstruction

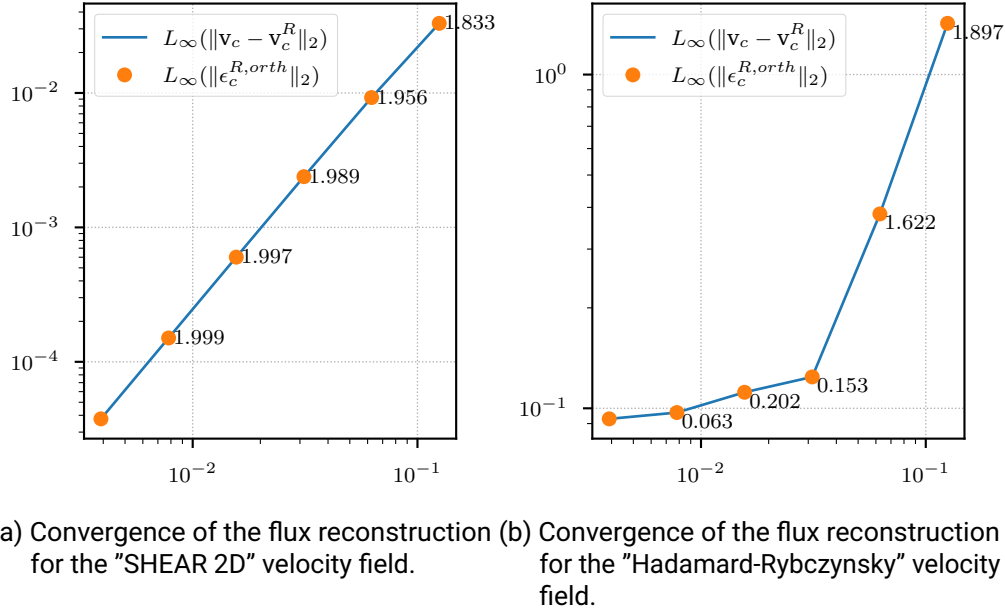


Figure 5.8.: Convergence of the flux reconstruction.

Figure 5.8 contains results from two tests used to verify the second-order accuracy of the reconstruction operator described in section 5.7. In both cases, velocity is reconstructed in cell centers, from the numerical scalar flux exactly defined at face centers. The first case, shown in fig. 5.8a, is a single-phase solenoidal velocity field function known as the "single vortex test", often used to validate the advection of the fluid interface [98].

For the other verification case, the creeping flow around a spherical interface is used, as given by the Hádarnard-Rybczynsky model. The velocity expressions available in [19] are used. The interface is a circle of radius  $R = 0.15$ , centered at  $(0.5011, 0.507, 0.05)$ . The outside viscosity is  $\mu_o = 5e-04$  and the inside viscosity  $\mu_i = 5e-03$  and the free stream velocity  $\mathbf{v}_\infty = 1.0$ . Figures 5.8a and 5.8b show that the reconstruction error  $\epsilon_c^{R,orth}$  given by eq. (5.24) exactly corresponds with the computed  $L_\infty$  error norm  $L_\infty(\|\mathbf{v}_c - \mathbf{v}_c^R\|_2)$ . Therefore, eq. (5.24) can be used to verify the reconstruction operator convergence behavior both for single and two-phase flows. Second order accuracy is obtained only in the single-phase scenario, shown in fig. 5.8a. However, as the discontinuity of the velocity gradient in the Hádarnard-Rybczynsky model increases with increasing mesh resolution,

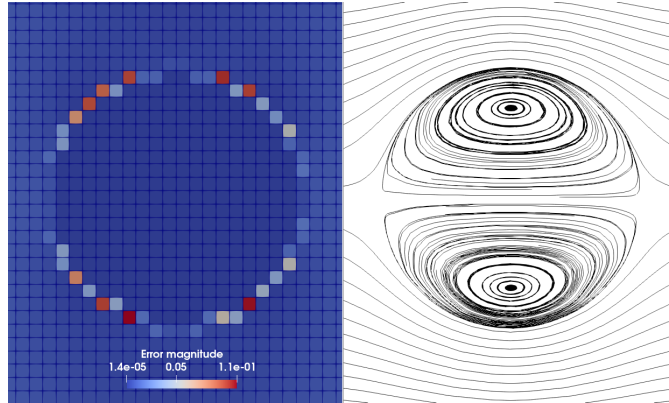


Figure 5.9.: Error distribution for the Hadamard-Rybczynsky flow.

the convergence of the reconstruction operator deteriorates, as shown in cf. fig. 5.8b. Figure 5.9 confirms this, by showing the  $L_\infty$  error norm distribution for the velocity field, where the error is concentrated at the fluid interface.

These results have an important consequence. Field reconstruction is performed by the SAAMPLE algorithm at the r.h.s. of the momentum equation for the surface tension force (together with other forces) as well as for the velocity field, at the end of the internal loop of the SAAMPLE algorithm. As clearly visible in fig. 5.9, both the velocity field, and the surface tension force reconstruction introduce errors at the interface. Further improvements of the reconstruction operator are expected to improve the convergence and stability of the SAAMPLE algorithm and are left as future work.

### 5.9.3. Stationary droplet

According to the Young-Laplace law, the velocity for a spherical droplet in equilibrium in the absence of gravity is  $\mathbf{v} = 0$  because the surface tension is balanced by the pressure jump across the interface. With a prescribed, constant curvature this case allows to test if a numerical method is *well-balanced* [35]. With a numerically approximated curvature, limitations of the numerical method with respect to the capillary number

$$Ca = \frac{|\mathbf{v}_{\text{ref}}| \mu}{\sigma} \quad (5.31)$$

due to so-called *spurious currents* can be investigated. A setup given in [93, 1] is adapted here. The setup introduced below differs from those publications in two regards. First, it is three-dimensional instead of two-dimensional. Second, no symmetry is used: the complete

droplet is simulated. The domain is  $\Omega : [0, 0, 0] \times [1.6, 1.6, 1.6]$  with a spherical interface of  $R = 0.4$  centered at  $[0.800000012, 0.799999932, 0.800000054]$ , to avoid exact overlap with mesh points. The material properties are identical for the droplet and the ambient fluid with the density  $\rho = 1$ , the kinematic viscosity  $\nu = [8.165e-2, 2.582e-2, 8.165e-3, 0]$  and a surface tension coefficient of  $\sigma = 1$ . The values of  $\nu$  are chosen such that the Laplace number

$$La = \frac{2R\sigma}{\rho\nu^2} \quad (5.32)$$

assumes  $La = [120, 1200, 12000, \infty]$ . Dirichlet boundary conditions are prescribed for the pressure  $p = 0$  on  $\partial\Omega$  and for the velocity  $\nabla\mathbf{v} \cdot \mathbf{n} = 0$  on  $\partial\Omega$ . The initial conditions are  $p(t_0) = 0$  and  $\mathbf{v}(t_0) = 0$ . A time step of  $\Delta t = 0.5\Delta t_{cw}$  is chosen where

$$\Delta t_{cw} = \sqrt{\frac{\rho h^3}{\pi\sigma}} \quad (5.33)$$

is the time step limit due to capillary waves according to [27].

### Prescribed exact, constant curvature

To test if our numerical method is well-balanced, a stationary droplet is simulated with a prescribed, constant curvature. Figure 5.10 shows the temporal evolution of the spurious currents when the PISO algorithm is used to solve the pressure velocity system. Except for the inviscid case, the PISO algorithm does not obtain the equilibrium state within the first time step. Instead, there is a transient phase before the velocity magnitude falls below the linear solver tolerance. This behavior can be understood by reformulating eq. (5.26) as

$$\nabla \cdot [\mathbf{D}_c^v(\nabla p')_c] = \nabla \cdot \mathbf{v}^* + \underline{\nabla \cdot \mathbf{H}_c[\mathbf{v}']} \quad (5.34)$$

where  $p' = p^* - p^{\text{prev}}$  is a correction to the old pressure field  $p^{\text{prev}}$ . Again, the underlined term  $\underline{\nabla \cdot \mathbf{H}_c[\mathbf{v}']}$  is neglected. Thus, from eq. (5.34) it is clear that the pressure correction is driven by the divergence of the preliminary velocity field  $\mathbf{v}^*$ . So, starting with a constant pressure field the only acting forces in the solution of eq. (5.25) are surface tension and viscous forces. The latter counteracts surface tension, thus the resulting force is lower than in the inviscid case. One can then expect the volume defect also to be smaller than in the inviscid case. However, since the volume defect is the only source term for eq. (5.34), the gradient of the updated pressure field  $p^* = p^{\text{prev}} + p'$  does not balance surface tension. Consequently,  $L_\infty(|\mathbf{v}^*|) > \text{tol}_s$  can be expected after the explicit velocity update eq. (5.27). Increasing the number of pressure correction iterations reduces  $L_\infty(|\mathbf{v}(t = \Delta t)|)$ , but it may require a considerable number of iterations to reach a given

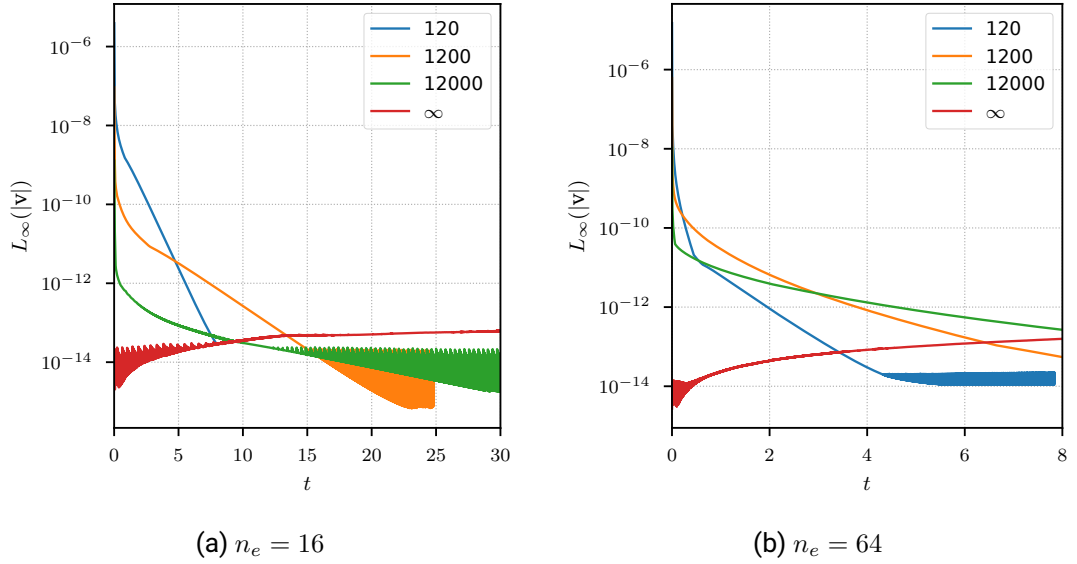


Figure 5.10.: Temporal evolution of the spurious currents for the case of a stationary droplet for different Laplace numbers. The LENT method is configured as in [71] using the PISO algorithm and exact curvature. On the left, the results for an equivalent resolution of  $n_e = 16$  are displayed while the right graph has been computed with an equivalent resolution of  $n_e = 64$ .

$n_e$	Laplace number			
	120	1200	12000	$\infty$
16	15	9	7	1
64	19	11	8	1

Table 5.4.: Number of pressure-correction iterations required to obtain  $L_\infty(|\mathbf{v}|) < 1e-13$  within the first time step for the stationary droplet with exact curvature. Results are shown for different Laplace numbers and two mesh resolutions.

$La$	$n_e$	$L_\infty( \mathbf{v}(t = \Delta t) )$		$L_\infty( \mathbf{v}(t = t_{\text{end}}) )$		$t_{\text{end}}$
		PISO	SAAMPLE	PISO	SAAMPLE	
120	16	3.90e-06	1.47e-14	3.32e-14	1.14e-14	7.8
	32	7.40e-06	7.74e-15	1.25e-14	3.74e-15	7.8
	64	1.49e-05	1.77e-14	2.17e-14	3.40e-14	7.8
	128	2.37e-05	1.29e-14	8.55e-11	3.61e-14	0.3
1200	16	9.29e-08	1.06e-14	7.75e-16	1.33e-15	24.8
	32	2.25e-07	2.70e-14	6.32e-15	6.62e-16	24.8
	64	6.10e-07	1.56e-14	5.47e-14	9.84e-16	8.0
	128	1.39e-06	1.61e-14	2.53e-11	1.72e-14	0.3
12000	16	1.27e-09	4.26e-14	1.39e-14	1.72e-14	78.4
	32	3.38e-09	4.85e-14	1.59e-15	1.37e-15	50.0
	64	1.07e-08	1.74e-14	2.67e-13	2.66e-14	8.0
	128	2.99e-08	1.94e-14	5.81e-12	2.77e-14	0.3
$\infty$	16	2.60e-15	7.29e-14	2.03e-13	1.75e-14	100
	32	2.97e-15	2.34e-14	1.56e-13	1.25e-13	35.0
	64	2.73e-15	2.18e-14	1.58e-13	7.54e-14	8.0
	128	2.70e-15	1.58e-14	2.19e-14	1.20e-14	0.3

Table 5.5.: Spurious currents of the stationary droplet using the PISO and SAAMPLE algorithm with exact curvature. Results are shown for different Laplace numbers  $La$  and mesh resolutions  $n_e$ . Magnitude of the spurious currents is given after one time step and at the end of the simulations.

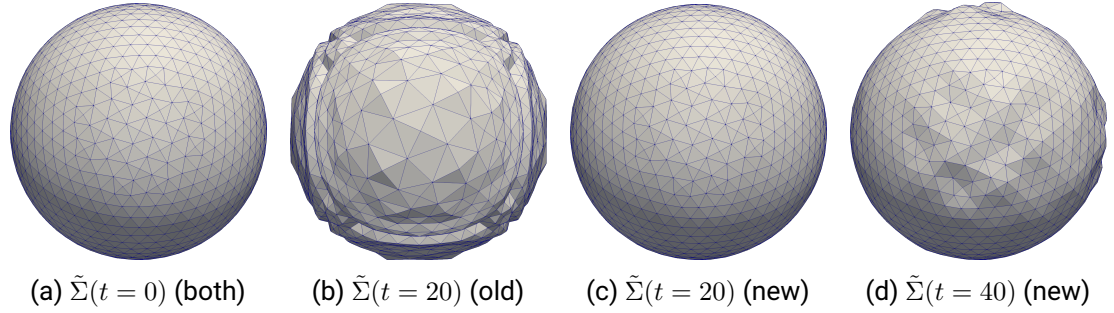


Figure 5.11.: Front for the stationary droplet at different times for  $La = 120$ . From left to right: initial front (identical for old and new configuration), front of the old configuration at  $t = 20$ , front of the current configuration at  $t = 20$  and front of the current configuration at  $t = 40$ .

threshold as displayed in table 5.4. This behavior motivated the development of the accuracy controlled SAAMPLE algorithm 3. Table 5.5 compares  $L_\infty(|\mathbf{v}|)$  of PISO and SAAMPLE after the first time step and at the end of simulation. For all configurations, the SAAMPLE algorithm maintains  $L_\infty(|\mathbf{v}|) < \text{tol}_{ls}$  over the simulated time. This indicates that our method is balanced in the sense of [35] and that SAAMPLE is a suitable segregated solution algorithm for two-phase flows.

### Numerically approximated curvature

Two curvature models are used. For the LENT configuration from [71] the  $DG(\alpha)$  model is used while the current configuration employs the  $\text{sccDG}(\phi)$  model (see table 5.1). The results are compared in fig. 5.12 for two resolutions and four Laplace numbers. Overall, the new configuration of LENT reduces the spurious currents between one and two orders magnitude for the simulated time and Laplace numbers. With the old configuration [71] simulations over the depicted time is only possible for  $La = 120$  ( $n_e = 16$ ) and  $La = [120, 1200]$  ( $n_e = 64$ ). Applying the modifications described in chapter 5 allows to simulate more physical time for all Laplace numbers. To put these results into perspective, a study of Abadie et al. which investigates the influence of surface tension and advection for various VOF and Level Set methods [1] is taken as a reference. They show results of a stationary droplet with  $La = 12000$  and a resolution of  $n = 64$  which corresponds to the green graph in fig. 5.12d. One should keep in mind though that a two-dimensional setup employing a structured mesh with staggered variable arrangement is used in [1]. All reported combinations in figure 3 of [1] show either almost constant  $Ca_{\max}$  over a



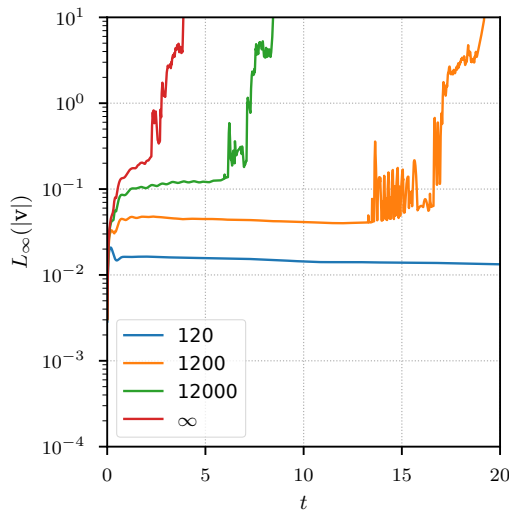
---

simulated time of around 40 s or decreasing  $Ca_{\max}$  before reaching a quasi stationary state. For the first group,  $Ca_{\max}$  lies between  $O(10^{-6})$  and  $O(10^{-4})$ , while for the second group  $O(10^{-16}) < Ca_{\max} < O(10^{-8})$  is reported. With  $La = 12000$ , the method proposed within this thesis maintains an almost constant  $Ca$  of  $O(10^{-6})$  up to  $t \approx 2$ . Afterwards,  $Ca_{\max}$  increases until an oscillatory state is reached around  $t \approx 7$  with  $O(10^{-4}) < Ca_{\max} < O(10^{-3})$ . A similar behaviour is observed for other Laplace numbers and the coarser resolution  $n_e = 16$ , with  $La = 120$  and  $n_e = 16$  being the only exception. A possible cause for this behaviour is the average number of front triangles per interface cell. For both resolutions, each interface cell contains 8-9 triangles on average, meaning that the front's resolution is notably finer than the resolution of the volume mesh. So, the relatively coarse resolution of the velocity field, which drives the motion of the front, may prevent that an equilibrium or quasi stationary state is reached. Instead, small scale perturbations accumulate in the vertex positions. These perturbations feed back through different parts of the algorithm (signed distance calculation, curvature approximation, surface tension) into the velocity. Over time, the perturbations become visible as shown in fig. 5.11d. Currently, it is not possible to change the average number of triangles per cell as this number is inherently linked to the interface reconstruction algorithm, whose improvement is left as future work.

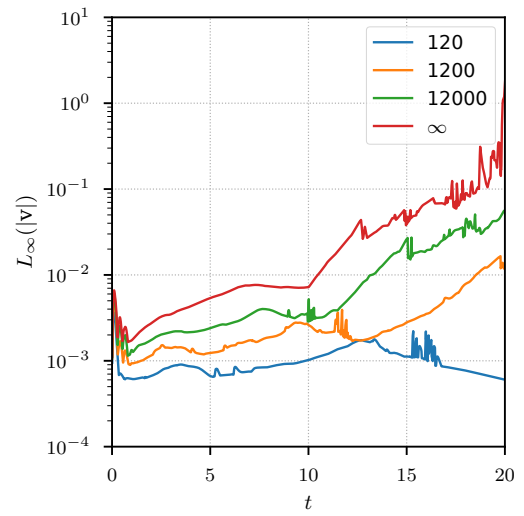
#### 5.9.4. Translating droplet

As pointed out by Popinet [93], the solution to a stationary droplet also holds in a moving reference frame. Yet this variant is better suited to study the influence of interface advection as the droplet moves through the fixed cells of the mesh. Again, the two-dimensional setup from [93] is adapted for three spatial dimensions. Material properties are same as for the stationary droplet (section 5.9.3). The radius of the droplet is  $R = 0.2$  and its center initially placed at  $\mathbf{c} = [0.5, 0, 5, 0.4]$  in  $\Omega : [0, 0, 0] \times [5R, 5R, 6R]$ . The constant, uniform background velocity is  $\mathbf{v}_{\text{bg}} = [0, 0, 1]$ . As boundary conditions  $\nabla \mathbf{v} \cdot \mathbf{n} = 0$  and  $p = 0$  is prescribed for the boundary part where  $z = 6R$ , for the rest of the boundary  $\mathbf{v} = \mathbf{v}_{\text{bg}}$  and  $\partial p / \partial \mathbf{n} = 0$  is set. The initial conditions are  $p(t_0) = 0$  and  $\mathbf{v}(t_0) = \mathbf{v}_{\text{bg}}$ . Simulation duration is chosen as  $t = 0.4$ , so the that the droplet is advected by one diameter.

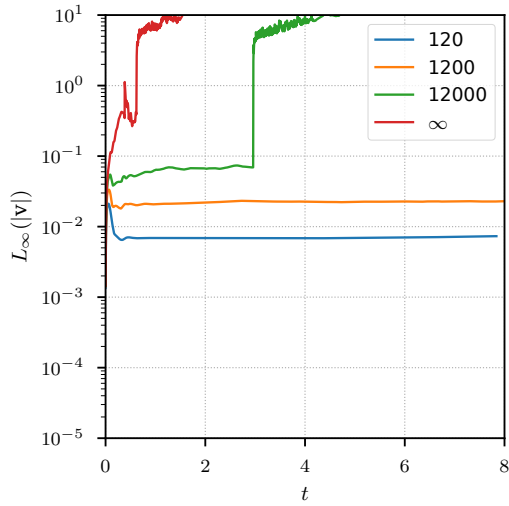
In fig. 5.13 the temporal evolution of the velocity deviations from the background velocity field  $\mathbf{v}_{\text{bg}}$  is displayed. As for the stationary droplet (section 5.9.3), the figure compares two configurations of LENT for two resolutions. The improvements are similar to the stationary droplet with spurious currents reduced between one and two orders of magnitude. For  $n_e = 64$  and  $La = [12000, \infty]$ , the qualitative behavior changed also. The magnitude of spurious currents oscillates around its initial level while it increases for the previous configuration. Popinet [93] reports the period of the oscillations to scale with



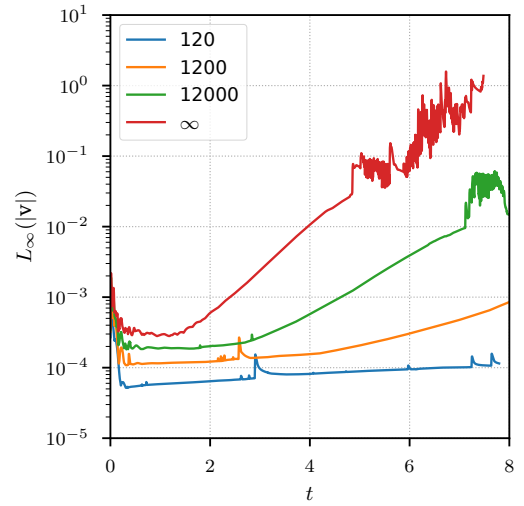
(a) Resolution  $n_e = 16$ , old state



(b) Resolution  $n_e = 16$ , new state



(c) Resolution  $n_e = 64$ , old state



(d) Resolution  $n_e = 64$ , new state

Figure 5.12.: Evolution of spurious currents for the stationary droplet when using the LENT method. The left column shows the results obtained with the configuration from [71], the right column for the current configuration. In the upper row results for a resolution of  $n_e = 16$  are displayed, in the lower row for  $n_e = 64$ . Each plot shows the results for different Laplace numbers.

$|\mathbf{v}|/h$  as the droplet moves through the cell layers of the mesh. Compared to the results of the stationary droplet shown in fig. 5.12, a notable difference is that the spurious currents remain in the same order of magnitude. This can be attributed to the different physical times simulated. For the translating droplet, this is a time span of 0.4 s. During this time in fig. 5.12b and fig. 5.12d,  $L_\infty(|\mathbf{v}|)$  decreases for the stationary droplet, while the onset of increasing  $L_\infty(|\mathbf{v}|)$  lies beyond  $t = 1$  s.

To put these results into perspective, figure 7 of Abadie et al. [1] is taken as a reference. The parameters are the same as for fig. 5.13d ( $n_e = 64$ ,  $La = 12000$ ), albeit in a two-dimensional setting with staggered variable arrangement. In contrast to the stationary droplet, the qualitative behaviour of spurious currents is quite similar for LENT and most of the VOF / Level Set configurations shown in [1] which also exhibit oscillations. Quantitatively, Abadie et al. [1] report results in the range  $O(10^{-4}) < Ca_{\max} < O(10^{-3})$  for the VoF methods and in the range  $O(10^{-6}) < Ca_{\max} < O(10^{-5})$  for the level set methods. With  $|\mathbf{v}|_{\max} \approx 3e-3$ , LENT maintains  $Ca_{\max} \approx 2.4e-5$ , achieving more accurate results than the tested VOF methods and comparable accuracy with regard to level set methods. In this case, the Lagrangian advection is advantageous as the movement of the front vertices due to  $\mathbf{v}_{bg}$  is captured exactly by first order spatial interpolation and first order temporal integration (eq. (5.1)). The errors arise from the signed distance calculation, influencing the calculation of  $\alpha$  and the approximation of  $\kappa$ .

### 5.9.5. Oscillating droplet

#### Comparison to analytic solution

To analyze the accuracy of LENT with interface deformation a setup of an oscillating droplet given in [105, 109] is adopted. For this case, Lamb derived an analytical solution. The oscillation frequency of an inviscid droplet is given by

$$\omega_n^2 = \frac{n(n+1)(n-1)(n+2)\sigma}{[(n+1)\rho_d + n\rho_a]R_0^3} \quad (5.35)$$

with the mode number  $n$ , the droplet density  $\rho_d$ , the density of the ambient fluid  $\rho_a$  and the radius of the unperturbed droplet  $R_0$ . In case of a viscous fluid, the amplitude  $a_n(t)$  decreases over time

$$a_n(t) = a_0 e^{-\gamma t}, \quad \gamma = \frac{(n-1)(2n+1)\nu}{R_0^2}. \quad (5.36)$$

The initial interface shape is

$$R(\theta, t) = R_0 + \epsilon P_n(\cos \theta) \sin(\omega_n t), \quad \theta \in [0, 2\pi], \quad (5.37)$$

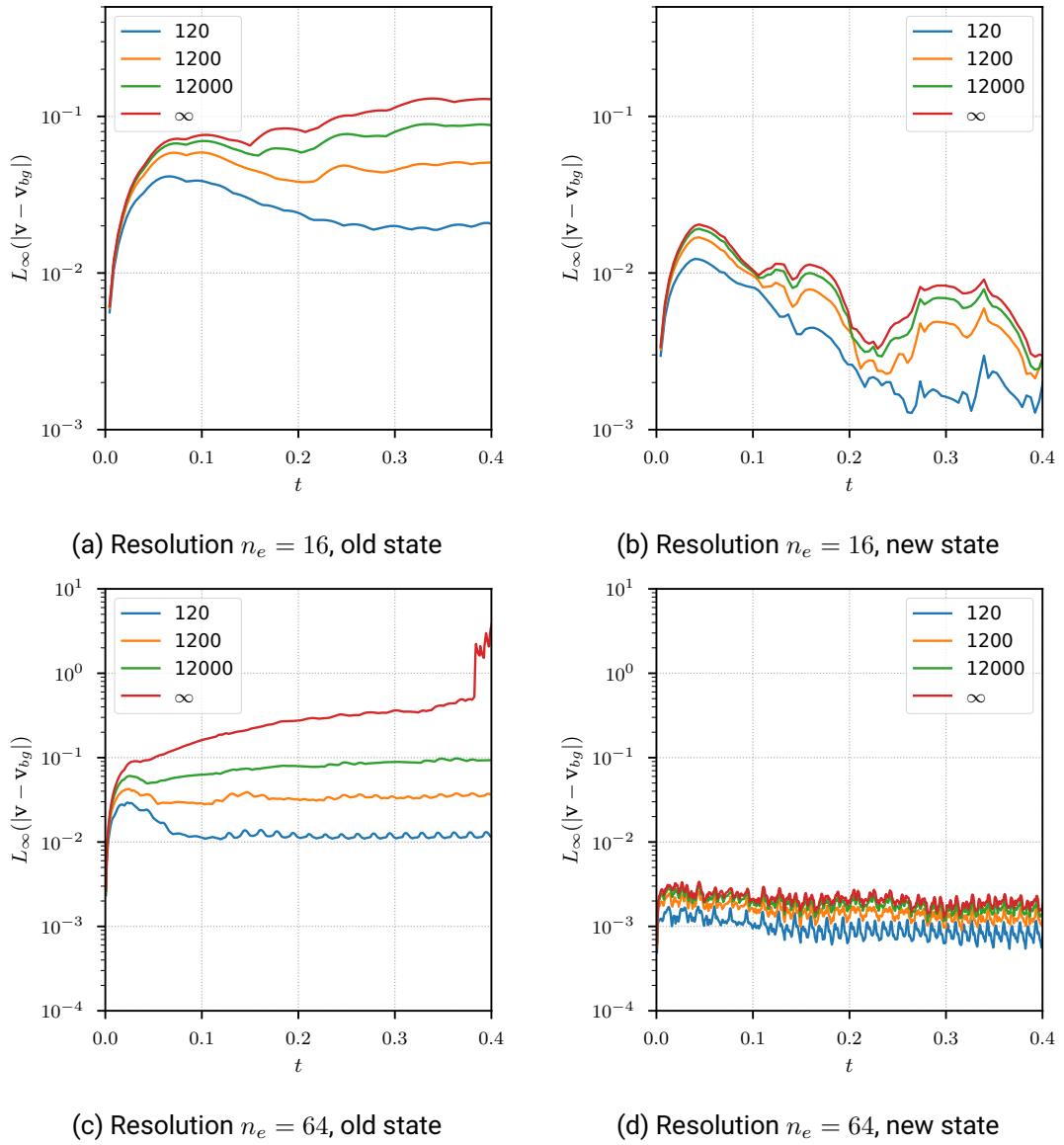


Figure 5.13.: Evolution of spurious currents for the translating droplet when using the LENT method. The left column shows the results obtained with the configuration from [71], the right column for the current configuration. In the upper row results for a resolution of  $n_e = 16$  are displayed, in the lower row for  $n_e = 64$ . Each plot shows the results for different Laplace numbers.

where  $P_n$  denotes the  $n$ -th order Legendre polynomial.

The domain is  $\Omega : [0, 0, 0] \times [4, 4, 4]$ , the interface is initialized with  $R_0 = 1$ ,  $n = 2$ ,  $\epsilon = 0.025$  and  $t = \pi/(2\omega_n)$  with its center at  $[2.00001, 1.99999, 2.0000341]$ . Material parameters are  $\rho_d = 10$ ,  $\rho_a = 0.1$ ,  $\nu_d = [0.05, 0.005]$ ,  $\nu_a = 5e-4$  and  $\sigma = 10$ . Initial fields at  $t = 0$  are  $\mathbf{v}_0 = 0$  and  $p_0 = 0$ . Dirichlet boundary conditions are used for the pressure ( $p = 0$ ) and  $\nabla \mathbf{v} \cdot \mathbf{n} = 0$  for  $\mathbf{v}$ . The semi-axis length is computed as

$$s_x = \frac{\max_k \left( \mathbf{x}_{\Sigma}^k \cdot \mathbf{e}_x \right) - \min_k \left( \mathbf{x}_{\Sigma}^k \cdot \mathbf{e}_x \right)}{2} \quad (5.38)$$

in each time step.

In fig. 5.14 the evolution of the semi-axis  $s_x$  is depicted for the previous configuration of LENT [71] and the current one with two different kinematic viscosities. For  $\nu_d = 0.05$ , both configurations capture the qualitative behavior. However, the previous configuration shows considerable deviations with regard to the temporal evolution of  $s_x$ . This can be attributed to development of spurious currents. First, the droplet deforms towards a cubic shape similar to fig. 5.11b, resulting in a smaller  $s_x$  around  $t^* = 1/2$  than analytically predicted. A second effect is that small wave like perturbations with wavelength comparable to the cell size  $h$  grow over time. Since the displacement of a single vertex can already change the result of eq. (5.38),  $s_x$  is considerably larger than the analytical prediction at later times, depending on the resolution. The setup reported in this thesis, however, shows much smaller deviations from the analytical solution. While  $s_x$  decays a bit slower than predicted by eq. (5.36), the numerical period converges with mesh refinement.

Setting  $\nu = 0.005$ , the previous configuration is not able to simulate one oscillation. Due to decreased dissipation, perturbations of the front amplify themselves faster than for the more viscous setup and eventually lead to the crash of the simulation. With the new configuration, however, this setup becomes viable. Both amplitude and period converge with mesh resolution and at  $n_e = 100$  the numerical results agree very well with eq. (5.35) and eq. (5.36).

### Comparison to experiment

The numerical results presented so far rely on analytical solutions for verification. In this section, the proposed method is validated against experiments conducted by Trinh and Wang [125]. They investigate oscillations of droplets for which, in contrast to section 5.9.5, the amplitude cannot be considered small compared to the equivalent radius of the droplet. For the experiments, single silicone oil drops are suspended in water. Each drop is kept at a stable position using acoustic radiation pressure generated by an ultrasonic transducer. A

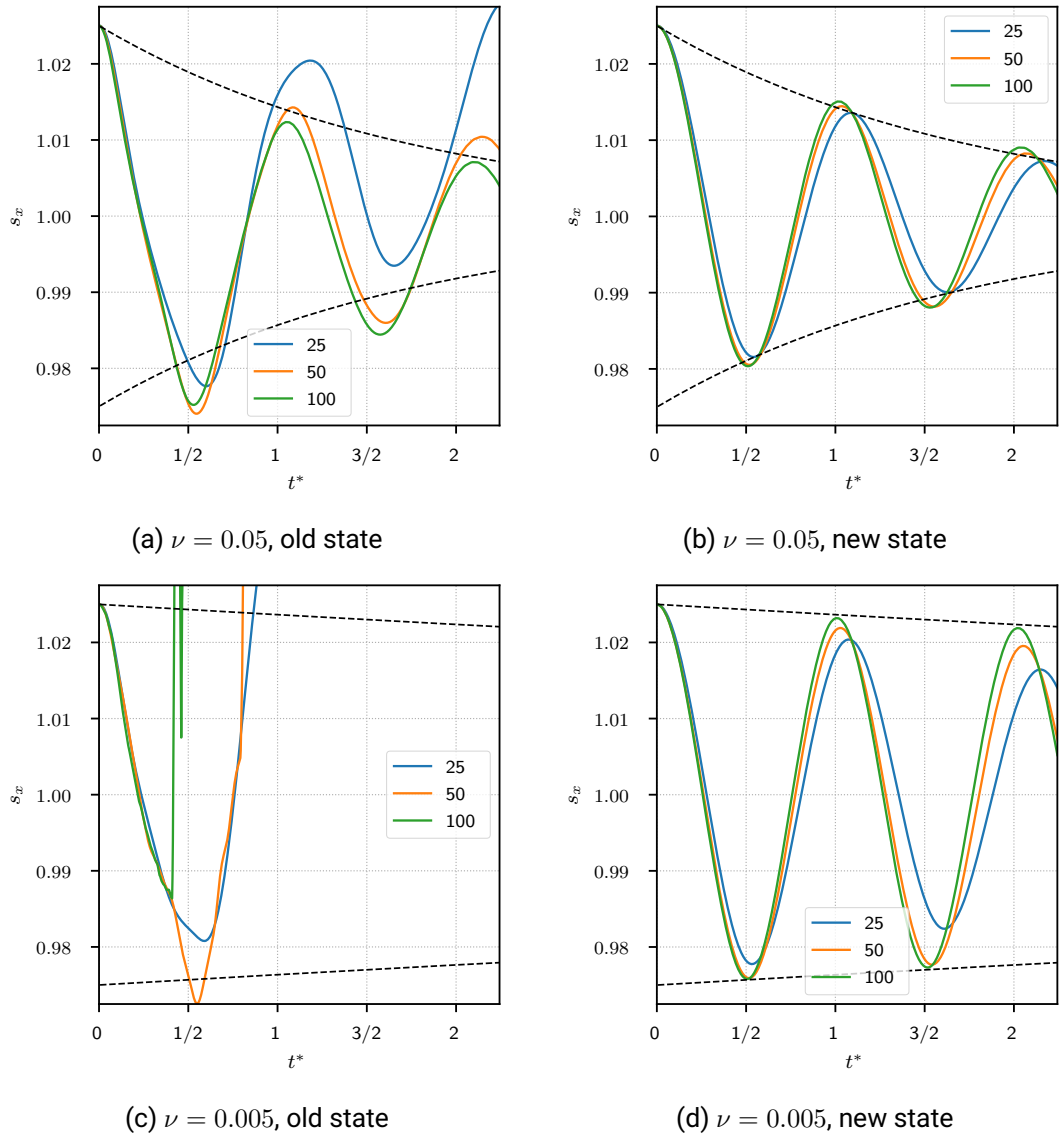


Figure 5.14.: Temporal evolution of the  $x$  semi-axis  $s_x$  for the oscillating droplet. Time is non-dimensionalized with the analytical period  $T_2 = 2\pi/\omega_2$ . The dashed lines represent the exact envelope of  $s_x$  ( $R_0 + a_2(t)$  and  $R_0 - a_2(t)$ , see eq. (5.36)). Each plot shows the results for three mesh resolutions  $n_e$ . The case has been simulated using the LENT configuration from [71] (left column) and the current configuration (right column) with different kinematic viscosities of the droplet (upper and lower row).

---

second transducer is used to drive the droplet oscillations. Besides forced oscillations, the authors investigate the damping of free large amplitude oscillations (section 5 in [125]). In the following, it is examined to what degree the LENT method is able to reproduce Trinh and Wang's experimental results.

The numerical setup is as follows. The fluid properties are as given in [125] with  $\rho_a = 998 \text{ kg/m}^3$ ,  $\nu_a = 0.95\text{e-}6 \text{ m}^2/\text{s}$  for the ambient phase (water),  $\rho_d = 1001 \text{ kg/m}^3$ ,  $\nu_d = 3.2\text{e-}6 \text{ m}^2/\text{s}$  for the droplet phase (silicone oil) and  $\sigma = 0.037 \text{ N/m}$ . A domain  $\Omega : [0 \text{ cm}, 0 \text{ cm}, 0 \text{ cm}] \times [8 \text{ cm}, 8 \text{ cm}, 8 \text{ cm}]$  is used with equivalent resolutions of  $n_e \in [64, 128, 256]$ . The initial fields are  $p_0 = 0$  and  $\mathbf{v}_0 = 0$ . Homogeneous Dirichlet boundary conditions are used for the pressure and homogeneous Neumann boundary conditions for the velocity field. The interface is initialized as a prolate spheroid, centered at  $[4 \text{ cm}, 4 \text{ cm}, 4 \text{ cm}]$  with two semi-axes configurations:  $\mathbf{s}_A = [8.02 \text{ mm}, 5.46 \text{ mm}, 5.46 \text{ mm}]$  and  $\mathbf{s}_B = [9.18 \text{ mm}, 5.1 \text{ mm}, 5.1 \text{ mm}]$ . Configuration *A* corresponds to a droplet volume of  $V_d = 1 \text{ cm}^3$  and a semi-axes ratio of  $\frac{L}{W} = 1.47$ , while configuration *B* corresponds to  $V_d = 1 \text{ cm}^3$  and  $\frac{L}{W} = 1.80$  in [125]. The time step is set to  $\Delta t = 0.5\Delta t_{cw}$  (eq. (5.33)), giving  $\Delta t_{64} = 2 \text{ ms}$ ,  $\Delta t_{128} = 0.72 \text{ ms}$  and  $\Delta t_{256} = 0.26 \text{ ms}$ . In fig. 5.15, the previous configuration of LENT [71] and the one described in this publication are compared to Trinh and Wang's experimental results given in section five of [125]. The semi-axis  $s_x$  is evaluated according to eq. (5.38). Except for the lowest resolution  $n_e = 64$ , the old configuration shows good agreement of amplitude and period for the first peak at  $t \approx 0.28 \text{ s}$ . However, afterwards, the behavior is qualitatively similar to fig. 5.14. Due to parasitic currents, perturbations accumulate in the front and feed back into the velocity field through surface tension. Subsequently, the semi-axis evolution starts to severely deviate from the expected behavior during the second oscillation period. Between  $t \approx 0.35 \text{ s}$  and  $t \approx 0.5 \text{ s}$ , depending on resolution and semi-axes ratio, the graphs no longer resemble a harmonic oscillation. With the modifications proposed here, however, the simulations yield the qualitatively expected behavior. Agreement between the experimental oscillation period and the simulated one is quite good with a relative difference of  $e_{\text{rel}}(L/W = 1.47) \approx 0.07$  and  $e_{\text{rel}}(L/W = 1.80) \approx 0.05$  for  $n_e = 256$ . The amplitude decays noticeably faster in the simulation compared to the experiment. This is related to the reconstruction operator eq. (5.17) and its diminishing convergence, illustrated in section 5.9.2. Another cause lies in the semi-implicit surface tension model eq. (5.15) as the second term is effectively a diffusion term. Improvement of the balanced discretization between the surface tension force and the pressure gradient on unstructured meshes by introducing an alternative field reconstruction operator in OpenFOAM, as well as the introduction of the new algorithm for the reconstruction of the Front are ongoing work. It is important to note, though, that the simulation results computed with the existing numerical method converge toward the experimental data with increasing mesh resolution.

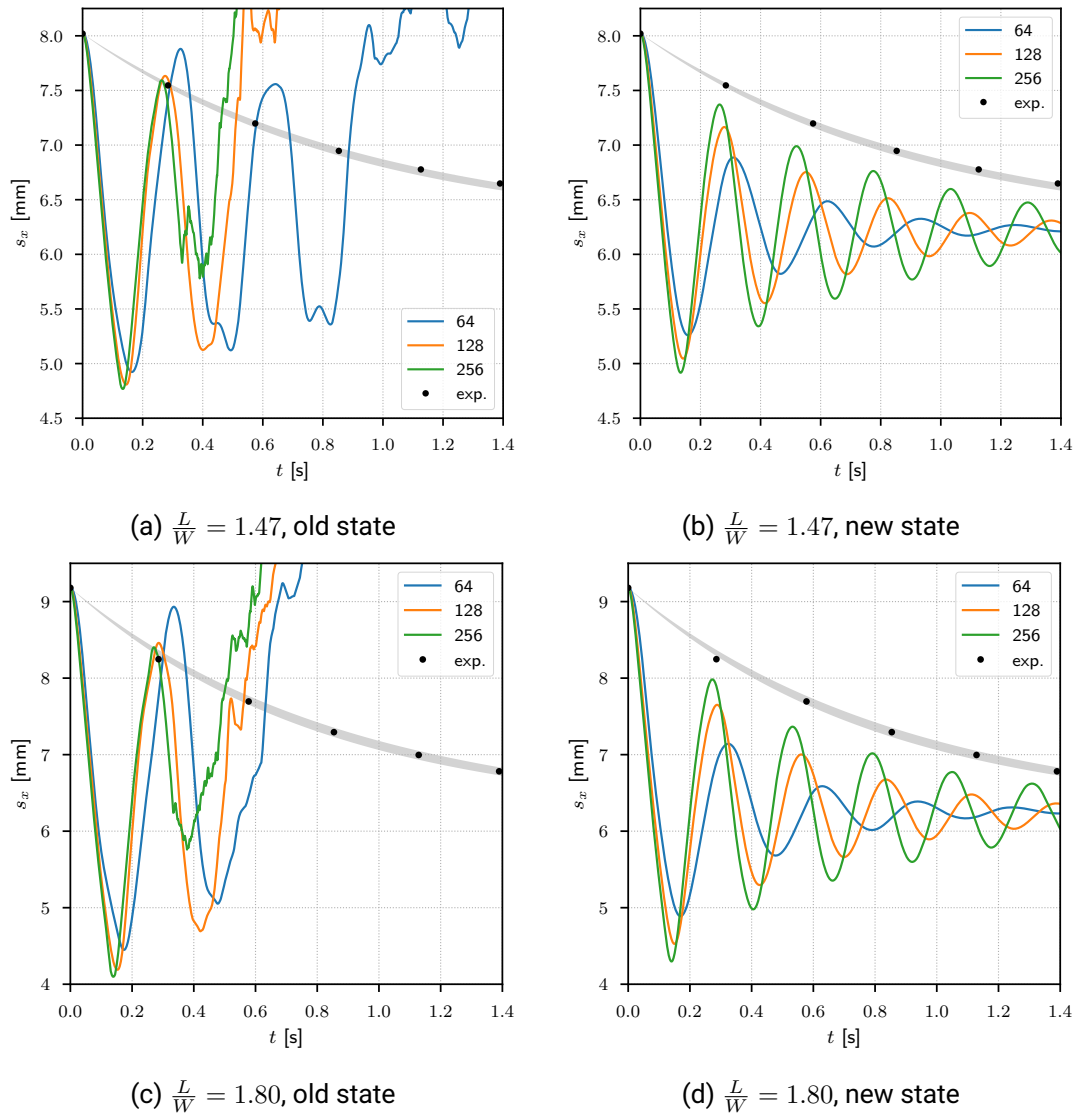


Figure 5.15.: Temporal evolution of the  $x$  semi-axis  $s_x$  for the oscillating droplet replicating the experimental setup described in [125]. The gray area depicts the decay envelope (see eq. (5.36)) according to the experimentally measured  $\gamma$ , while the black dots mark measured oscillation peaks. Each plot shows the results for three different mesh resolutions  $n_e$ . The left column shows the results using the LENT configuration from [71], the right column the results obtained with the current configuration. Two semi-axes ratios  $L/W$  have been simulated.



---

## 5.10. Conclusions

The proposed SAAMPLE algorithm together with the improvements in the curvature approximation, phase indicator approximation and implicit surface tension modeling significantly increases the numerical robustness of the unstructured LENT hybrid Level Set / Front Tracking method when simulating surface-tension driven flows, when compared to both the previous publication [71] and contemporary Level Set and VOF methods on structured meshes. For the experimental case reported in section 5.9.5, overdamping of the solution is still present. However, the solution converges to the experimental observation with increased mesh resolution.

It is found that the field reconstruction from scalar values on unstructured meshes in OpenFOAM diverges for fields that are not at least  $C^1$ . This behavior of the reconstruction operator has not been reported so far in the literature, and it is crucial for the segregated equation coupling in OpenFOAM for multiphase flows. The field reconstruction is also used for combustion, spray simulations, electromagnetic simulations and heat transfer (weak compressibility), so the findings reported in section 5.7 might be of significant importance for those applications as well.

Additionally, the length scale of the reconstructed Front should be connected with the length scale of the Eulerian mesh by developing a new Front reconstruction algorithm on unstructured meshes which does not construct the connectivity of Front elements. The absence of connectivity between the Front elements will make the parallelization of the method, using the message passing parallel programming model, more straightforward and will enable to more accurately tackle physical problems such as the one in section 5.9.5, by allowing much higher mesh resolutions.

Improvements of the field and Front reconstruction algorithms are left as future work.



---

## 6. Mass flux consistency and high density ratios

---

In the previous chapter, both fluids featured either the same density or only a small difference to isolate the errors arising from the numerical treatment of surface tension. While such settings are useful for testing, they do not reflect the conditions usually encountered in applications. A variety of natural and industrial two-phase flow processes involve gas/liquid flows, characterized by density ratios  $\rho^-/\rho^+ \geq 1000^1$ , such as the atomization of fuel jets [65], sloshing tank [43], mold filling [111], water flooding [37]. Large density ratios at the fluid interface cause severe challenges for numerical simulations [28]: the pressure Poisson equation becomes ill-conditioned due to the discontinuity, and spurious numerical errors in the solution of the momentum equation accumulate because of inconsistencies between mass and momentum advection.

Within this chapter, the conditions for a consistent transport of mass and momentum in two-phase setting with different density values are derived and an approach for calculating consistent fluxes for methods that do not provide volumetric phase fluxes, as e.g. the LENT method, is introduced.

### 6.1. Literature overview

Ghods and Herrmann [40] point out that for Level Set methods mass and momentum are typically transported in different, inconsistent ways. While mass is transported by a solution of the level set equation, momentum is obtained from solving a non-conservative form of a momentum balance equation. Hence, a large non-physical change in the momentum can be generated by a small error in the interface position when the density ratio is high. Nangia et al. [79] state that the abrupt change in density often introduces notable shear at the interface and adds difficulties in the discretization of governing momentum equations at the interface, which further leads to higher stiffness of the linear equation system.

---

<sup>1</sup>Within this chapter  $\rho^-$  denotes the density of the denser fluid, so that  $\rho^- \geq \rho^+$  and  $\rho^-/\rho^+ \geq 1$  holds.

---

Many researchers have addressed these problems, and some indicated further that because of the sizeable numerical error resulting from high-density ratios, some flow algorithms or solvers can only be used to solve low density-ratio cases with  $\rho^-/\rho^+ \in [1, 10]$  [97]. However, in engineering applications, density ratios usually range from 100 to 1000, and even 10000 for molten metals or water-water vapor systems. Hence, a solution algorithm with the ability to handle a broader range of density ratio problems is required to simulate real-world engineering problems.

A pioneering attempt to alleviate numerical instability of the VOF method caused by high-density ratios was made by Rudman [99]. Rudman [99] has used a sub-mesh with a doubled mesh resolution for advecting volume fractions, compared to the mesh used for the momentum and pressure equations. The goal of this two mesh approach was the reduction of small errors in the discrete momentum that cause large errors in the velocity. However, an additional higher mesh resolution for the volume fractions requires a discrete divergence free velocity on the finer mesh. Furthermore, using an additional mesh for the volume fractions increases the computational costs significantly, and it is not applicable to general unstructured meshes. Rudman [99] demonstrates qualitatively a reduction of parasitic currents for the stationary droplet case with  $\rho^-/\rho^+ = 100$ , and improved results for more complex cases. Another important finding of Rudman [99] is the role of the densities used in the mass flux and the momentum flux in ensuring numerical consistency of the two-phase momentum advection.

Bussmann et al. [20] extended the work of Rudman [99] for the unstructured collocated finite volume method. Bussmann et al. [20] employ the conservative form for the momentum convection. At first, the momentum convection is solved separately, using an explicit Euler time integration scheme. Bussmann et al. [20] use the unstructured unsplit Volume-of-Fluid method of Rider and Kothe [98], which enables the simplification of the numerical consistency requirement for the density and momentum equations. Specifically, the solution of the volume fraction equation results in phase-specific volumes at face centers. Those phase-specific volumes are then used to compute the volume fractions at face centers. These volume fractions are used by Bussmann et al. [20], together with a simple average of cell densities, and velocities calculated by the least squares reconstruction technique, to compute the momentum fluxes at face centers. Since the velocity is continuous at the interface, the least squares approximation is acceptable. However, calculating face-centered densities by an average does not yield numerical stability in all cases. Contrary to Rudman [99], Bussmann et al. [20] do not require an additional finer mesh. They do, however, limit the solution to first-order accuracy in time and introduce the Courant–Friedrichs–Lewy (CFL) condition by solving the momentum advection equation explicitly. Bussmann et al. [20] introduce the important case of a translating droplet in a quiescent ambient fluid. This test case can be used to demonstrate numerical

---

---

consistency in the momentum transport. Their solutions show accurate results for high density ratios, especially considering the fact that even the unsplit VOF method distorts the interface during the translation [22]. However, for  $\rho^-/\rho^+ \in [1, 100]$ , the constant translation velocity is modified by the solution of the pressure and momentum equations, which indicates a remaining numerical inconsistency in this approach.

Sussman et al. [119] employ the CLSVOF method [117] for obtaining a robust and stable solution for the density ratio of 1000 by extrapolating the liquid velocities into the gas domain. The interface is advected using the extrapolated liquid velocity field only.

Raessi and Pitsch [97] propose a 2D staggered discretization of conservative single-field form of two-phase Navier-Stokes equations for handling high density ratios. Like Bussmann et al. [20] did, Raessi and Pitsch [97] first solve the momentum advection equation, using second-order (or higher) explicit integration schemes, and upwinding for the velocity near the interface. The density used in the momentum convective term is computed as a weighted combination of signed distances from the old and the new time step. For the partially submerged line segments bounding 2D rectangular cells, intersection between the mesh and the zero level set (iso-surface) is performed using the marching cubes algorithm. Raessi and Pitsch [97] point out that there is still an inconsistency between the face-centered density and the momentum transport, as the Level Set equation remains decoupled / inconsistent with the momentum transport. The verification of numerical stability was done using the translating droplet case from Bussmann et al. [20], and results demonstrate qualitative improvement for the density ratio  $\rho^-/\rho^+ = 10^6$ . Other density ratios have not been verified. A viscous oscillating droplet case demonstrates quantitative improvement in terms of the improved amplitude decay rate, compared to non-conservative form of the momentum equation.

Le Chenadec and Pitsch [63] extend their forward/backward Lagrangian tracking and Eulerian remapping VOF method [63] for handling high density ratios. Equivalent to volume fractions in [63], the density and the momentum are advected in the Lagrangian forward/backward tracking step by observing the control volume as a material volume and moving the mesh forward / backward with the flow velocity. While the content of material volumes does not change on the continuum level, this condition cannot be discretely ensured and is a source of conservation errors. In the Eulerian re-mapping step, physical properties are transferred from the Lagrangian to the Eulerian mesh, and the geometrical intersections between the PLIC interface on the forward/backward image of the mesh, and the background mesh, are another source of volume conservation errors. Ensuring numerical consistency further requires the transfer of velocities located at the center of mass. Since the velocities associated with the cell centroids are used, an inconsistency is introduced. Qualitative results show significant improvements for the stationary droplet with  $\rho^-/\rho^+ = 10^9$ , and quantitative improvement is shown for the standing wave by

---

---

Prosperetti [95] with  $\rho^-/\rho^+ = 850$ .

Ghods and Herrmann [40] have developed a consistent rescaled momentum transport (CRMT) method. The CRMT method discretizes the conservative form of the single-field Navier-Stokes equations using a collocated unstructured Finite-Volume method. To increase the numerical stability for high density ratio, CRMT solves a mass conservation equation using a mass flux either by upwinding the face-centered density in the interface cells and their face-neighbors (defined by a volume fraction tolerance), or by averaging the densities elsewhere. The same discretization scheme used for the face-centered density is also applied to the mass flux in the convective term of the momentum equation. A difference is therefore introduced in the mass flux of the continuity equation and the mass flux in the convective term of the momentum equation when upwinding is used, because the upwinded face-centered density in the continuity equation uses the face-centered velocity, while the upwinded mass flux in the momentum equation includes both the upwind velocity and density. Within this thesis it is shown that any difference in the discretization of the mass flux is a source of numerical inconsistency for the two-phase momentum advection. Like Bussmann et al. [20], the explicit discretization of the momentum convective term introduces the CFL condition, limiting the time step for convection-dominated multiphase flows, where high density ratios play a major role. Using upwind schemes makes the discretization first-order accurate. The droplet translation case [20], with  $\rho^-/\rho^+ = 10^6$ , is compared in terms of the droplet shape, that remains stable. Other density ratios are not reported for this verification case. It is the author's opinion, that the droplet shape errors may result from the interface advection scheme<sup>2</sup>, and should be generally substituted by the  $L_\infty$  norm of the velocity error to demonstrate numerical consistency.

Vaudor et al. [131] base their approach on a CLSVOF code from Aniszewski et al. [7], which can switch between Level Set based and VOF based mode to calculate momentum fluxes. They chose the VOF-based momentum fluxes calculation mode and implemented the framework of Rudman's method [99] but with more accurate interpolation schemes for velocities and velocity gradients on faces of staggered meshes to ensure consistency. This method is developed in two-dimensions and exploits two sets of meshes. To provide a more widely applicable method, Vaudor et al. [130] advanced the method in their more recent study. In contrast to the previous work [131], the Level Set method tracks the interface, while the VOF method is utilized to update density. They exploited the identical scheme to discretize conservative convective term in mass and momentum equation. In addition, the mass flux is also identical in both discretized equations. A new strategy that leverages half cell-faces' and half cells' quantities of volume fraction and density to couple

---

<sup>2</sup>The Level Set and VOF methods do not exactly preserve the shape of a translating droplet.

---

---

staggered mass cells and momentum cells is introduced to avoid the need for a refined mesh in the original method by Rudman [99]. A prominent feature of this new method is that it can be used to simulate three-dimensional applications. Besides, comparing with the method from Rudman [99], the new method shows relatively low computational cost when simulating the same 2D application.

Owkes and Desjardins [87] presented a three-dimensional, un-split, second-order semi-Lagrangian VOF scheme that conserves mass and momentum and ensures consistency between the mass (volume fraction) and momentum fluxes. The volume fractions are geometrically transported near the fluid interface using the method from [85]. As in [99], Owkes and Desjardins [87] introduce an additional refined mesh for the calculation of semi-Lagrangian fluxes. The motivation for the refined mesh is to enforce the consistency between semi-Lagrangian mass and momentum fluxes, similar to Rudman [99]. Results confirm mass and momentum conservation, and stability of the momentum convection. The method proposed by Owkes and Desjardins [87] relies on the staggered variable arrangement and this, together with the use of the additional finer mesh, makes this approach inapplicable to unstructured finite volume meshes.

Orazzo et al. [83], similarly to Rudman [99], resolve the volume fraction function on twice finer sub-cells and update density from the volume fraction. After that, they update face-centered density on mass cells by averaging density on sub-cells, and then evaluate the mass flux on the faces of standard staggered momentum cells. These density and mass flux values are used to initialize and calculate interim momentum and velocity during the prediction step. Zuzio et al. [139] made no changes and applied Orazzo's method [83]. Besides, they further verified and validated this method with new, different cases and analyzed the results of simulations using more novel ways. Yang et al. [135] notice that the high-density ratio has a profound effect on robustly simulating two-phase flows at high Reynolds numbers. To mitigate the problem, they adopt the consistent framework from Nangia et al. [79] and replace the interface-capturing method in [79], which is standard Level Set, with CLSVOF method [117] to ensure mass conservation.

Patel and Natarajan [90] employ the method of Ghods and Herrmann [40], a high-resolution scheme called Cubic Upwind Interpolation (CUI) for the convective terms of momentum and volume fraction transport equations, and the solution of a momentum equation in the face-normal direction. The face-normal momentum equation leads to a combined collocated/staggered variable arrangement, that requires the use of nonlinear solvers, as this equation is a nonlinear algebraic equation. Patel and Natarajan [90] demonstrate the balanced nature of their discretization for the stationary droplet using exact curvature and density ratios  $\rho^-/\rho^+ \in [10, 1000]$ . Numerical stability is demonstrated with reduced parasitic currents when the curvature is approximated numerically for  $\rho^-/\rho^+ = 10$  and a Weber number of  $We = 1$ . For the verification test case of the two-

---

---

phase momentum convection problem,  $\rho^-/\rho^+ = 10^6$  is used without surface tension and viscous forces and qualitative results show slight deformations of the interface shape, the  $L_\infty$  norm of the velocity error is not reported. With enabled surface tension and viscous forces and exact curvature prescribed, and density ratios  $\rho^-/\rho^+ = 1, 1000$ , the velocity error in the  $L_\infty$  norm lies within  $[10^{-3}, 10^{-2}]$ .

Manik et al. [67], similarly to [90], attempt to enforce numerical consistency by applying the similar discretization scheme on the conservative form of the volume fraction advection equation and the momentum conservation equation. Manik et al. [67] are using a collocated unstructured Finite-Volume method for the equation discretization and the CUBISTA scheme (Alves et al. [5]) to discretize convective terms. The verification of the numerical consistency for the two-phase momentum advection is done using the droplet translation case of Bussmann et al. [20] and density ratios  $\rho^-/\rho^+ = 10^3, 10^6$ , that demonstrates qualitative improvement compared to a naive discretization of the momentum convective term with the upwind method. The qualitative evaluation is based on the shape of the droplet, given by the 0.5 iso-surface of the volume fraction. Although the proposed method demonstrates improvement w.r.t. an obviously inconsistent approach, some shape deformation is still visible, so one can conclude that  $L_\infty(\mathbf{v}) \neq 0$  and some non-zero velocities are still generated.

A recent second-order accurate Level Set method is proposed by Nangia et al. [79], extending the work from Ghods and Herrmann [40] that is first-order accurate. Similar to the method proposed by Ghods and Herrmann [40], an additional mass conservation equation is solved, and the identical mass flux is used for both mass and momentum transport. Two techniques are employed: one is the third-order accurate Koren's limited CUI, which is modified to consistently discretize the convective term of both mass and momentum equation. This scheme satisfies the convection-boundedness criterion (CBC) and is total variation diminishing (TVD). The second technique is the solution of an update equation for the face-centred densities. In this step, a third-order accurate strong stability preserving Runge-Kutta (SSP-RK3) scheme is used for time integration. The update is performed in every fix-point iteration, and the updated face-centered density is then employed to solve the discretized momentum equation.

Zuzio et al. [139] also follow Ghods and Herrmann [40] by solving an auxiliary continuity equation for increasing the numerical consistency in discretizing the two-phase momentum convection term. Their Consistent Mass-Momentum (CMOM) transport method utilizes a staggered Cartesian variable arrangement and utilizes the two-phase incompressible Navier-Stokes equations in the conservative form, solved using Chorin's projection method together with the CLSVOF method for tracking the fluid interface. The solution of the auxiliary density equation requires the evaluation of staggered (face-centered) densities, by constructing staggered control volumes, and evaluating the densities using



---

sub-grid quadtree (octree in 3D) refinement and intersection with the PLIC interfaces. This aspect of CMOM shows the importance of evaluating the densities at face-centers that are required for the solution of the auxiliary continuity equation. Momentum flux reconstruction scales the fluxed phase-specific volume from the VOF method. Finally, the two-phase momentum is advected in the staggered cells, and scaled with the corresponding density to obtain velocity components in all spatial directions. Zuzio et al. [139] demonstrate significant improvements in numerical stability in a very detailed way, reporting shape, position and kinetic energy errors for canonical verification and validation cases. The kinetic energy for the dense translating droplet [20] with a density ratio of  $10^6$  is reported, and CMOM recovers a numerically stable solution.

Arrufat et al. [8] consider the conservative form of the advection equation of a discontinuous property to enforce numerical consistency of the advected two-phase momentum, using face averages that are derived by integrating the advection equation in space and time. Since the discontinuity of the property introduced by the interface complicates the evaluation of the face averages, two additional equations are introduced, one for each phase. The method is derived for the MAC staggered variable arrangement. Results demonstrate a numerically stable droplet shape when it is advected with a constant velocity, however, the authors consider this case to only test the consistency of the implementation and not the numerical consistency of the method so the results are not quantified in terms of kinetic energy or  $L_\infty$  velocity errors. Still, the method shows significant improvements for realistic multiphase flows with high density ratios.

The high-density ratio is also challenging for other numerical methods for two-phase flows, like the phase-field and lattice Boltzmann. The corresponding surveys are beyond the scope of this work, more details can be found in [31, 133, 47, 48, 64, 137]. Contrary to the numerical two-phase methods mentioned so far, the difficulties with high density ratios are far less pronounced for Front Tracking methods [109] because the marker field (phase-indicator) is not as sharp as in the unstructured Volume-of-Fluid method [69] and the unstructured Level Set / Front Tracking method [71].

The methods of Bussmann et al. [20], Ghods and Herrmann [40], Patel and Natarajan [90], and Manik et al. [67] utilize the unstructured Finite-Volume equation discretization, other above-mentioned methods utilize a staggered variable arrangement that is not applicable to unstructured meshes. Compared to contemporary collocated Finite-Volume methods, the proposed  $\rho$ LENT method achieves the numerical consistency in the two-phase momentum advection exactly. The requirement for the auxiliary mass conservation equation, introduced by Ghods and Herrmann [40], and the requirement for the face-centered (flux) density from the mass conservation principle are derived. Compared to a similar observation by [8], the proposed approach avoids the integration in time that complicates the evaluation of face-centered quantities, as demonstrated in detail

below. Although hybrid Level Set / Front Tracking LENT method [71] is used for interface capturing, the  $\rho$ LENT solution algorithm can be used with other interface capturing methods, where there is a discrepancy in the evaluation of the collocated density. Compared to all contemporary collocated methods,  $\rho$ LENT allows an implicit discretization of the two-phase momentum convective term which removes the CFL stability condition.

## 6.2. A solution algorithm for two-phase flows with high density ratios using the collocated unstructured Finite-Volume method

### 6.2.1. Numerical consistency of the single-field conservative two-phase momentum convection term

Bussmann et al. [20] were the first to consider the problem of numerical consistency of the two-phase momentum convective term in the setting of the collocated unstructured Finite-Volume method. Within this chapter, their work is extended by improving the accuracy of the face-centered density evaluation and employing a solution algorithm that allows for an implicit discretization of the convective term, thus removing the CFL condition.

Consider first the Euler explicit collocated unstructured Finite-Volume discretization of the mass conservation equation,

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (6.1)$$

namely,

$$\frac{\rho_c^{n+1} - \rho_c^n}{\Delta t} + \frac{1}{|\Omega_c|} \sum_{f \in C_c} \rho_f^n F_f^n = 0, \quad (6.2)$$

$$\rho_c^{n+1} = \rho_c^n - \frac{\Delta t}{|\Omega_c|} \sum_{f \in C_c} \rho_f^n F_f^n, \quad (6.3)$$

where  $F_f$  is the volumetric flux at the face defined by the index  $f$  from the set of all indexes of cell-faces  $C_c$  that belong to  $\Omega_c$ , defined as

$$F_f := \mathbf{v}_f \cdot \mathbf{S}_f, \quad (6.4)$$

with  $\mathbf{v}_f^n$  as the face-centered velocity and  $\mathbf{S}_f := |S_f| \mathbf{n}_f$  as the face area-normal vector, with  $|S_f|$  denoting the area of the face  $S_f$ .

Second, the focus is on the two-phase momentum advection with a prescribed initial velocity, spatially constant throughout the solution domain and its inlet/outlet boundaries. Without forces on the r.h.s of eq. (2.13), the momentum conservation equation becomes

$$\partial_t(\rho\mathbf{v}) + \nabla \cdot (\rho\mathbf{v} \otimes \mathbf{v}) = 0. \quad (6.5)$$

Without forces on the r.h.s. of eq. (6.5), eq. (6.5) keeps the initial velocity spatially constant and without acceleration/deceleration. Therefore, a numerically consistent unstructured collocated FVM discretization of the two-phase momentum convection equation (6.5) must ensure that no artificial acceleration or deceleration occurs. For example, just like eq. (6.3), the Euler explicit discretization of eq. (6.5) is

$$\rho_c^{n+1}\mathbf{v}_c^{n+1} = \rho_c^n\mathbf{v}_c^n - \frac{\Delta t}{|\Omega_c|} \sum_{f \in C_c} \rho_f^n F_f^n \mathbf{v}_f^n. \quad (6.6)$$

If this discretization is numerically consistent, and the velocity field remains spatially constant, then

$$\mathbf{v}_f^n = \mathbf{v}_c^n, \quad (6.7)$$

which is, of course, ensured for the initial spatially constant velocity ( $\mathbf{v}_f^0 = \mathbf{v}_c^0$ ). Equation (6.7), applied to eq. (6.6), results in

$$\rho_c^{n+1}\mathbf{v}_c^{n+1} = \mathbf{v}_c^n \left( \rho_c^n - \frac{\Delta t}{|\Omega_c|} \sum_{f \in C_c} \rho_f^n F_f^n \right), \quad (6.8)$$

and dividing by  $\rho_c^{n+1}$  finally gives

$$\mathbf{v}_c^{n+1} = \frac{\mathbf{v}_c^n \left( \rho_c^n - \frac{\Delta t}{|\Omega_c|} \sum_{f \in C_c} \rho_f^n F_f^n \right)}{\rho_c^{n+1}}. \quad (6.9)$$

As there are no forces on the r.h.s. of eq. (6.5), the velocity should not be changed simply by advecting the two-phase momentum, i.e.

$$\mathbf{v}_c^{n+1} = \mathbf{v}_c^n, \quad (6.10)$$

and this condition is ensured in eq. (6.9) if

$$\frac{\rho_c^n - \frac{\Delta t}{|\Omega_c|} \sum_{f \in C_c} \rho_f^n F_f^n}{\rho_c^{n+1}} = 1, \quad (6.11)$$

---

which is equivalent to eq. (6.3): the Euler explicit discretized form of the mass conservation equation. Consequently, a numerically consistent discretization of the momentum convection equation requires the new cell-centered density  $\rho_c^{n+1}$  to be computed by solving a mass conservation equation. This observation justifies theoretically the use of the auxiliary mass conservation equation, introduced by Ghods and Herrmann [40], and used by [79, 97, 83, 139, 23].

Modern unstructured geometric flux-based Volume-of-Fluid methods ([51, 87, 70, 101], see [69] for a recent review), potentially ensure this property, since they solve the conservative formulation of the volume fraction advection equation for  $\alpha_c^{n+1}$  by computing phase-specific fluxed volumes, and then use the cell-centered volume fraction  $\alpha_c^{n+1}$  to compute  $\rho_c^{n+1}$  with eq. (3.20). However, the temporal discretization scheme used in the momentum equation for the convective term, must be consistent with the integration of the fluxed phase-specific volumes, used to obtain  $\alpha_c^{n+1}$ . Additionally, the  $\alpha_c^{n+1} \in [0, 1]$  must hold near machine epsilon. Any correction to  $\alpha_c^{n+1}$  performed after the numerical solution of the volume fraction advection equation, that bounds  $\alpha_c^{n+1}$  within  $[0, 1]$ , results in a discrepancy between  $\rho_c^{n+1}$  computed using the mass flux that gives unbounded  $\alpha_c^{n+1}$ , and the  $\rho_c^{n+1}$  computed from the a-posteriori bounded  $\alpha_c^{n+1}$  using eq. (3.20).

It is important to note that if the pressure gradient is included on the r.h.s of eq. (6.5), any error in  $\mathbf{v}_c^{n+1}$  will result in non-zero source terms on the r.h.s. of the resulting pressure equation, in the  $p - \mathbf{v}$  coupling algorithm. Since the pressure gradient enforces  $\nabla \cdot \mathbf{v} = 0$  ( $\sum_{f \in C_c} F_f = 0$  on the discrete level), this results in artificial velocities similar to parasitic currents caused by the surface tension force.

However, numerical methods such as Front Tracking, Level Set, and their hybrids, update the cell-centered density  $\rho_c^{n+1}$  from the approximated fluid interface that is not advected by solving the volume fraction advection equation. Bussmann et al. [20] rely on this consistency of the Volume-of-Fluid method and use the VOF fluxes to first solve eq. (6.5) explicitly in the first step, followed by the second step that includes volume and surface forces. The approach from Bussmann et al. [20] works very well for the VOF method, however it cannot be directly applied to the Level Set method, the Front Tracking method or their hybrids, when the unstructured collocated Finite-Volume method is used to discretize the single-field Navier-Stokes equations. The solution algorithm for high density ratios that is proposed avoids the CFL condition imposed by Bussmann et al. [20] and increases the accuracy of the face-centered density  $\rho_f$  required by the mass flux, and it is applicable to any multiphase flow simulation method that utilizes the single-field formulation of the Navier-Stokes equations.

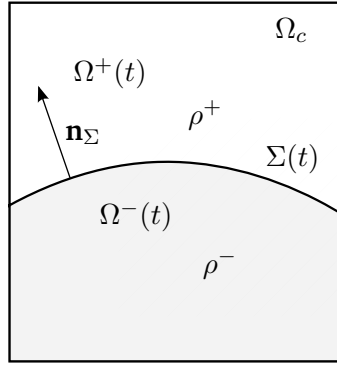


Figure 6.1.: A two-phase fixed control volume  $\Omega_c$  separated by the interface  $\Sigma(t)$ .

### 6.2.2. A semi-implicit solution algorithm for high density ratios

Section 6.2.1 provides the formal reasoning behind solving the mass conservation equation (or its equivalent) for  $\rho_c^{n+1}$ . Since Ghods and Herrmann [40] introduced the use of the "auxiliary" mass conservation equation, other researchers have adopted this approach, with the main difference in the way the face-centered (mass flux) density  $\rho_f$  is evaluated both in the discretized mass conservation equation (6.1) and the discretized momentum equation (2.13).

The condition given by eq. (6.11), derived from eqs. (6.9) and (6.10) can be fulfilled only if the same face-centered (mass flux) density is used when discretizing the auxiliary mass conservation and momentum equations. Going one step further, the volumetric flux  $F_f$  must also be the same in the discretized auxiliary mass conservation and momentum equations. Put together, the mass flux in the auxiliary discretized mass conservation equation must be equal to the mass flux in the discretized momentum conservation equation: this is the requirement for the mass flux consistency, mentioned throughout the literature.

It is relevant to point out that the same model for the single-field density given by eq. (3.20) is used throughout the literature. The basis of this model is mass conservation, and this fundamental principle further leads to an interesting conclusion regarding the evaluation of the face-centered (mass flux) density  $\rho_f$  in the discretized mass and momentum conservation equations. The face centered density is evaluated differently throughout scientific publications reviewed in section 6.1, and here it is shown that there is a strict relationship between the phase indicator and the face centered density  $\rho_f$ .

Consider the fixed control volume  $\Omega_c$  in fig. 6.1, that is separated by the fluid interface  $\Sigma(t)$  into two parts, occupied by fluids  $\Omega^\mp(t)$ . The single-field density model given by

eq. (2.8) is adopted in every publication reviewed in section 6.1, and in the rest of the scientific literature on two-phase flow simulations. The mass conservation principle together with the single-field density model (2.8) give

$$\frac{d}{dt} \int_{\Omega_c} \rho dV = - \int_{\partial\Omega_c} \rho \mathbf{v} \cdot \mathbf{n} dS = - \int_{\partial\Omega_c} [\rho^- \chi + \rho^+ (1 - \chi)] \mathbf{v} \cdot \mathbf{n} dS. \quad (6.12)$$

The equality of surface integrals in eq. (6.12),

$$\int_{\partial\Omega_c} \rho \mathbf{v} \cdot \mathbf{n} dS = \int_{\partial\Omega_c} [\rho^- \chi + \rho^+ (1 - \chi)] \mathbf{v} \cdot \mathbf{n} dS,$$

demonstrates that the mass flux of the single-field density over  $\partial\Omega_c$  is determined by the constant densities  $\rho^\mp$  and the phase indicator given by eq. (2.7), if eq. (2.8) is used to model the single-field density. In other words, the single-field density at  $\partial\Omega_c$  should be computed using the phase indicator as done on the r.h.s. of eq. (6.12), otherwise the mass conservation of the single-field density model given by eq. (2.8) will not be upheld. This relevant condition transfers to the discrete level, leading to an interesting consequence for the computation of the face-centered (mass flux) density, that has so far been computed in many ways throughout the literature.

Specifically, when the surface integrals in eq. (6.12) are discretized using the unstructured collocated finite volume method,

$$\begin{aligned} \sum_{f \in C_c} \rho_f F_f &= \sum_{f \in C_c} \left[ \rho^- \left( \int_{S_f} \chi dS \right) \mathbf{v}_f \cdot \mathbf{n}_f + \rho^+ \left( \int_{S_f} dS \right) \mathbf{v}_f \cdot \mathbf{n}_f - \rho^+ \left( \int_{S_f} \chi dS \right) \mathbf{v}_f \cdot \mathbf{n}_f \right] \\ &= \sum_{f \in C_c} \left[ \rho^- \frac{\|\mathbf{S}_f\|}{\|\mathbf{S}_f\|} \left( \int_{S_f} \chi dS \right) \mathbf{v}_f \cdot \mathbf{n}_f + \rho^+ \left( \int_{S_f} dS \right) \mathbf{v}_f \cdot \mathbf{n}_f \right. \\ &\quad \left. - \rho^+ \frac{\|\mathbf{S}_f\|}{\|\mathbf{S}_f\|} \left( \int_{S_f} \chi dS \right) \mathbf{v}_f \cdot \mathbf{n}_f \right] \\ &= \sum_{f \in C_c} [\rho^- \alpha_f + \rho^+ (1 - \alpha_f)] F_f, \end{aligned} \quad (6.13)$$

where

$$\alpha_f := \frac{1}{|S_f|} \int_{S_f} \chi dS \equiv \frac{|\Omega^-(t) \cap S_f|}{|S_f|} \quad (6.14)$$

is the *area fraction* of the face  $S_f \subset \partial\Omega_c$ , i.e. the ratio of the area of  $S_f$  submerged in  $\Omega^-(t)$ , and the total face-area  $|S_f|$ . Further,  $\|\mathbf{S}_f\| \equiv |S_f|$ , and  $F_f$  is the volumetric flux  $\mathbf{v}_f \cdot \mathbf{S}_f$  (eq. 6.4) in eq. (6.13).

An important consequence of eq. (6.13) is the requirement for the evaluation of the face-centered (mass flux) density, necessary for ensuring the numerical consistency of the single-field two-phase momentum convection. Equation (6.13) requires all methods<sup>3</sup> that define  $\rho$  using eq. (3.20) to either compute  $\rho_f$  using the area fractions or  $\int_{S_f} \chi dS$  from eq. (6.13), or to achieve this equivalently when computing  $\rho_c^{n+1}$  from the advected volume fractions  $\alpha_c^{n+1}$ , which is possible for the flux-based VOF methods [69].

Another important realization is that eq. (6.13) is valid at any time  $t$  - which is very relevant for the semi-implicit discretization developed within the  $\rho$ LENT method, that applies eq. (6.13) at  $t^{n+1}$ .

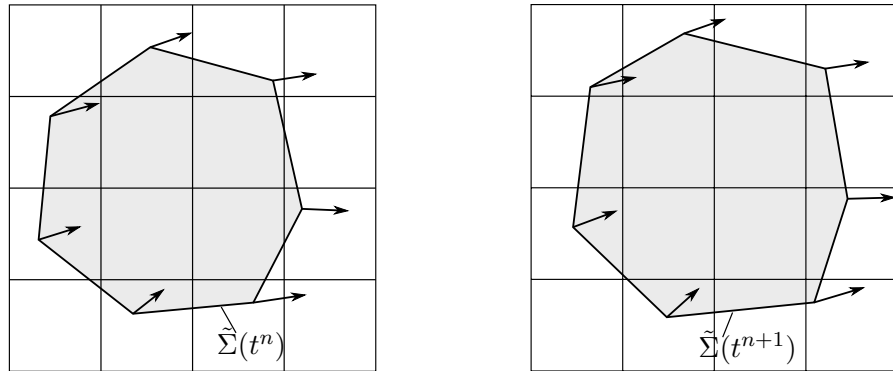
It is true that many two-phase simulation methods do not advect the phase indicator when advecting the fluid interface  $\Sigma(t)$ , but this does not infer that eq. (6.13) cannot be applied. The idea of using an auxiliary mass conservation equation introduced by [40], made into a formal requirement by eqs. (6.9) and (6.10), allows the use of eq. (6.13):  $\alpha_f$  can be computed regardless of the approximation of the fluid interface  $\Sigma(t)$  and the method used to advect it.

Similar to other contemporary methods, the  $\rho$ LENT method also first advects the interface using the velocity from the previous time step as shown in the left image of fig. 6.2a, resulting in the new position of the interface shown in the right image in fig. 6.2a, that is then used to *geometrically* calculate the face-centered density  $\rho_f^{n+1}$ , by calculating *area fractions*  $\alpha_f^{n+1}$  from the interface approximation, as shown in fig. 6.2b. The face-centered density  $\rho_f^{n+1}$  and the volumetric flux  $F_f^m$  are then used to update the cell-centered density  $\rho_c^{n+1}$  by solving a mass conservation equation. The index  $m$  in the volumetric flux refers to the linearization of the convective term in the momentum equation. The same mass flux  $\rho_f^{n+1} F_f^m$  is used in the *implicitly discretized* momentum conservation equation. The pressure-velocity coupling algorithm iterates the linearized volumetric flux  $F_f^m$  to  $F_f^{n+1}$ . Finally, the cell-centered velocity  $\mathbf{v}_c^{n+1}$  is obtained, which is used to evolve the fluid interface in the next time step, from  $t^{n+1}$  to  $t^{n+2}$ . At this point, the numerically consistent cell-centered density  $\rho_c^{n+1}$  has served its purpose and is reset using eq. (3.20), to make it consistent again with the fluid interface approximation.

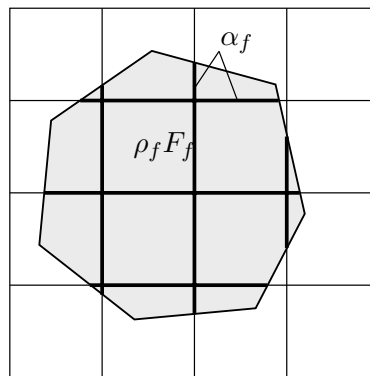
Any two-phase flow simulation method has the possibility to compute the face-centered density  $\rho_f(t)$  from the interface approximation in some way. The  $\rho$ LENT method computes the face-centered density  $\rho_f(t)$  by computing the face area fraction  $\alpha_f(t)$  (short: area fraction) of the face  $S_f$ , submerged in the phase  $\Omega^-(t)$ . The calculation of  $\alpha_f$  uses signed distances available in the unstructured LENT method [71]. Other two-phase flow simulation methods rely on a different approximation of the fluid interface  $\Sigma(t)$ , that

---

<sup>3</sup>All two-phase flow simulation methods encountered by the authors use eq. (3.20).



(a) Interface  $\tilde{\Sigma}$  at  $t^n$  and  $t^{n+1}$  and the respective  $\Omega^-(t^n)$  and  $\Omega^-(t^{n+1})$  in gray color, used to compute  $\alpha_c^n$  and  $\alpha_c^{n+1}$ , that are further used to compute  $\rho_c^n$  and  $\rho_c^{n+1}$  in an inconsistent way.



(b) Interface at  $\tilde{\Sigma}(t^{n+1})$  used to compute  $\alpha_f^{n+1}$ , then  $\rho_f^{n+1}$  and finally  $\rho_c^{n+1}$  in a consistent way, by solving a mass conservation equation.

Figure 6.2.: Updating the face-centered (mass flux) density in the  $\rho$ LENT method.

can be used to geometrically approximate the area fraction  $\alpha_f$  without resorting to an interpolation of the field that abruptly changes in the interface-normal direction.

In the original Front Tracking method, the density is updated utilizing the new position of marker points (the approximated interface) [78]. After the velocity field in the current step is computed, the position of marker points in the new time step can be updated



immediately by

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{v}_p^n, \quad (6.15)$$

where  $\mathbf{x}_p, \mathbf{v}_p$  indicate the position and interpolated velocity of marker points respectively, and  $\Delta t$  is the time step length. The advection of marker points along Lagrangian trajectories eventually corrupts the triangular mesh, leading to discrepancies in the ratios of triangular angles and areas and self-intersections of the triangular mesh. The original Front Tracking method [126] deals with this by redistributing marker points based on quality criteria imposed on the triangular mesh, which involves manipulating the connectivity of the triangular mesh.

Contrary to original Front Tracking [126], the LENT method reuses the principles from LCRM / Local Front Reconstruction Method (LFRM) methods [108, 110, 109, 106] and reconstructs the interface using an iso-surface reconstruction algorithm. The iso-surface reconstruction does not add/delete marker points locally by changing the connectivity of the triangular surface mesh; it reconstructs the entire interface in the solution domain as an iso-surface. Following the strategy from LCRM / LFRM, the physics of the problem determines the iso-surface reconstruction frequency. The LENT method uses the marching tetrahedra [124] algorithm to enable the iso-surface reconstruction on unstructured meshes. However, the marching tetrahedra algorithm introduces many triangles per cell (even with regularization), causing instabilities in front tracking. At the moment of writing, an alternative iso-surface reconstruction that relies on a higher-order signed-distance interpolation and results in a favorable ratio of triangle-to-cell length scales is developed.

Once the marker points are advected and redistributed, the cell density is updated depending on  $\mathbf{x}_p^{n+1}$ , namely

$$\rho^{n+1} = \rho(\mathbf{x}_p^{n+1}). \quad (6.16)$$

The face-centered density used for the mass flux is then interpolated from densities of two adjacent cells. Contrary to LENT, the face-centered density is updated by  $\rho$ LENT using the phase indicator approximated at each cell-face. A 2D interface is depicted in fig. 6.2b, where  $\alpha_f^{n+1}$  is the area fraction at  $t^{n+1}$ : the ratio of the cell-face area submerged in the phase  $\tilde{\Omega}^-(t^{n+1}) \approx \Omega^-(t^{n+1})$ , and the total face area  $|S_f|$ . More precisely, the area fraction  $\alpha_f^{n+1}$  is computed by the  $\rho$ LENT method using a second-order accurate approximation from signed distances [30], used in section 4.2.6 to equivalently approximate the volume fraction  $\alpha_c$  (see eq. (3.19)). The Level Set component of the LENT method [71] calculates signed distances from the triangular surface mesh that approximates the interface  $\Sigma(t^{n+1}) \approx \tilde{\Sigma}(t^{n+1}) := \partial\tilde{\Omega}^-(t^{n+1})$ . With the narrow band approach from [71], described in section 4.2, the signed distances can be computed efficiently at any point

in a close vicinity of  $\tilde{\Sigma}(t)$ . The original LENT method [71] computes signed distances at cell-centers and cell corner-points, and the proposed  $\rho$ LENT additionally computes signed distances at face centers. Each face  $S_f$  is triangulated using its centroid  $\mathbf{x}_f$ , as shown in fig. 6.3. The face centroid  $\mathbf{x}_f$ , together with the two successive cell-corner points that belong to the face  $S_f$ ,  $\mathbf{x}_{f,i}$ ,  $\mathbf{x}_{f,i+1}$ , forms a triangle  $(\mathbf{x}_f, \mathbf{x}_{f,i}, \mathbf{x}_{f,i+1})$ . Face-triangles may be partially submerged in the phase  $\tilde{\Omega}^-(t^{n+1})$ , in which case the submerged area of the triangle is computed using the nearest signed distances to  $\tilde{\Sigma}(t^{n+1})$  from the triangle points  $(\mathbf{x}_f, \mathbf{x}_{f,i}, \mathbf{x}_{f,i+1})$ , namely  $(\phi_f, \phi_{f,i}, \phi_{f,i+1})$ , as shown in fig. 6.3. The second-order approximation developed in [30] is used here for computing the area fraction of a triangle submerged in  $\tilde{\Omega}^-(t^{n+1})$ . Any other second-order method can be applied. For example, a linear interpolation of signed distances along the edges of the triangle may be used equivalently, or a geometrical intersection between  $\tilde{\Omega}^-(t^{n+1})$  and the triangle. The total submerged area of the face  $S_f$  is then the sum of the submerged areas of face-triangles

$$A_f^{n+1} := |\Omega^-(t^{n+1}) \cap S_f| = \sum_{t \in \mathcal{T}_f} |\Omega^-(t^{n+1}) \cap \mathcal{T}_t|, \quad (6.17)$$

where  $\mathcal{T}_f$  is the set of indexes of the triangles in the triangulation of the face  $S_f$ . As mentioned above, other two-phase flow simulation methods may compute  $|\Omega^-(t^{n+1}) \cap \mathcal{T}_t|$  differently.

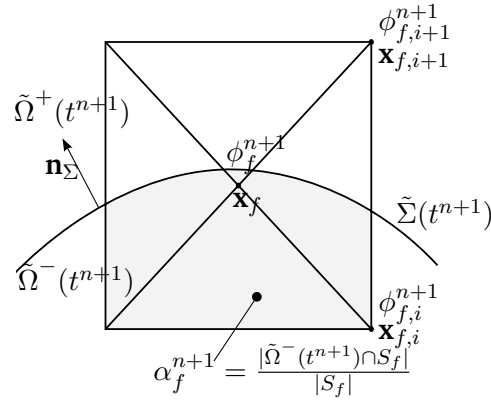


Figure 6.3.: Computing area fractions from signed distances in the method.

The area fraction  $\alpha_f^{n+1}$  is then computed as

$$\alpha_f^{n+1} := \frac{|\tilde{\Omega}^-(t^{n+1}) \cap S_f|}{|S_f|} = \frac{A_f}{|S_f|}, \quad (6.18)$$

as shown in fig. 6.3. Once the area fraction  $\alpha_f^{n+1}$  is approximated, it is used to compute the face-centered densities required by eq. (6.13), namely

$$\rho_f^{n+1} = \alpha_f^{n+1} \rho^- + (1 - \alpha_f^{n+1}) \rho^+, \quad (6.19)$$

at the new time step, because the interface has been advected forward in time to  $t^{n+1}$  with the available velocity  $\mathbf{v}^n$ . The discretized continuity equation (6.3) then obtains the form

$$\rho_c^{n+1} = \rho_c^n + \frac{\Delta t}{|V_{\Omega_c}|} \sum_f \rho_f^{n+1} F_f^m, \quad F_f^m = \mathbf{v}_f^m \cdot \mathbf{S}_f. \quad (6.20)$$

It is important to note that, although  $\rho_f^{n+1}$  appears in eq. (6.20),  $\rho$ LENT does not use an implicit discretization for eq. (6.20):  $\rho_f^{n+1}$  is geometrically computed from the fluid interface approximation  $\tilde{\Sigma}(t^{n+1})$ , so eq. (6.20) is solved explicitly (exactly). The exact (non-iterative) evaluation of  $\rho_c^{n+1}$  from eq. (6.20), alongside eq. (6.9), further infers the possibility of *exact numerical consistency* for the discretized convective term in the single-field momentum equation, which is in fact achieved and supported by the results.

In addition to density, the viscosity is updated utilizing the area fraction  $\alpha_f$ . Note that there is no need to calculate the cell-centered viscosity for the unstructured FVM discretization, only the face-centered viscosity is updated as follows

$$\mu_f^{n+1} = \alpha_f^{n+1} \rho^- \nu^- + (1 - \alpha_f^{n+1}) \rho^+ \nu^+. \quad (6.21)$$

The non-linearity of the convective term in the momentum equation eq. (2.13), namely  $\rho \mathbf{v} \mathbf{v}$ , is usually linearized when solving the single-field Navier-Stokes equations using the unstructured Finite-Volume method. The convective term (see section 3.2.2) is discretized as

$$\int_{\Omega_c} \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) dV \approx \sum_{f \in C_c} \rho_f^{n+1} F_f^m \mathbf{v}_f^{n+1}. \quad (6.22)$$

Numerical consistency also does not depend on the implicit / explicit discretization. As an analogue, observe the Euler implicit and Euler explicit discretization of the momentum convection equation: in the  $\lim_{\Delta x \rightarrow 0, \Delta t \rightarrow 0}$ , both implicit and explicit discretization must converge to eq. (2.13). Therefore, the requirement given by eqs. (6.9) and (6.10), is valid for an implicit discretization as well.

The volumetric flux  $F_f^m$  is initialized to  $F_f^n$  and iterated within the SAAMPLE pressure-velocity coupling algorithm loop, introduced in section 5.8, until  $m = n + 1$  is reached. The  $\rho$ LENT algorithm is outlined in algorithm 5 and it extends the SAAMPLE algorithm. It is relevant to note that  $F_f^m$  is iterated from  $F_f^n$  to  $F_f^{n+1}$  and  $p^m$  is solved for from  $p^n$  to  $p^{n+1}$  such that the discrete incompressibility condition  $\sum_{f \in C_c} F_f^{n+1} = 0$  is ensured.

---

**Algorithm 5** The  $\rho$ LENT solution algorithm.

---

- 1: **while** simulation time  $\leq$  end time **do**
  - 2:   Advect the interface to  $\tilde{\Sigma}(t^{n+1})$ . ▷ [71].
  - 3:   Compute the signed distance  $\phi^{n+1}$  from  $\tilde{\Sigma}(t^{n+1})$  at  $\mathbf{x}_c, \mathbf{x}_f, \mathbf{x}_p$  in the narrow-band. ▷ [71]
  - 4:   Compute  $\alpha_c^{n+1}$  from  $\phi_c^{n+1}, \phi_p^{n+1}$ . ▷ [121]
  - 5:   Compute the area fraction  $\alpha_f^{n+1}$  from the signed distance fields  $\phi_f^{n+1}, \phi_p^{n+1}$ . ▷ Figure 6.3
  - 6:   Compute the face-centered densities  $\rho_f^{n+1}$  using  $\alpha_f^{n+1}$ . ▷ Equation (6.19)
  - 7:   Solve the continuity equation using  $\rho_f^{n+1} F_f^m$  for cell-centered densities  $\rho_c^{n+1}$ . ▷ eq. (6.20).
  - 8:   Use  $\rho_c^{n+1}$  and  $\rho_f^{n+1} F_f^m$  in  $p - \mathbf{v}$  coupling to compute  $\mathbf{v}_c^{n+1}, F_f^{n+1}$ . ▷ [121] and eq. (6.22).
  - 9:   Make  $\rho_c^{n+1}$  consistent with  $\tilde{\Sigma}^{n+1}$ , i.e.  $\rho_c^{n+1} = \alpha_c^{n+1} \rho^- + (1 - \alpha_c^{n+1}) \rho^+$ .
  - 10:   Make  $\mu_c^{n+1}$  consistent with  $\tilde{\Sigma}^{n+1}$ , i.e.  $\mu_c^{n+1} := \alpha_c^{n+1} \rho^- \nu^- + (1 - \alpha_c^{n+1}) \rho^+ \nu^+$ .
  - 11: **end while**
- 

## 6.3. Verification and validation

Results presented in this section are publicly available [72].

### 6.3.1. Time step size

The time step size limit due to the CFL condition is given by

$$\Delta t \leq \Delta t_{CFL} = \frac{h}{U}, \quad (6.23)$$

where  $h$  is cell length and  $U$  is a characteristic velocity. In the cases,  $h$  is the ratio of domain side length to resolution  $N$ , while  $U$  is equal to magnitude of the ambient flow velocity vector, i.e.  $U = |\mathbf{v}_a| = 1$ . Another restriction for the time step size arises from the propagation of capillary waves on interfaces between two fluids. This time step constraint was introduced first by Brackbill et al. [18], and later revised by Denner and Wachem [27]. It has the form

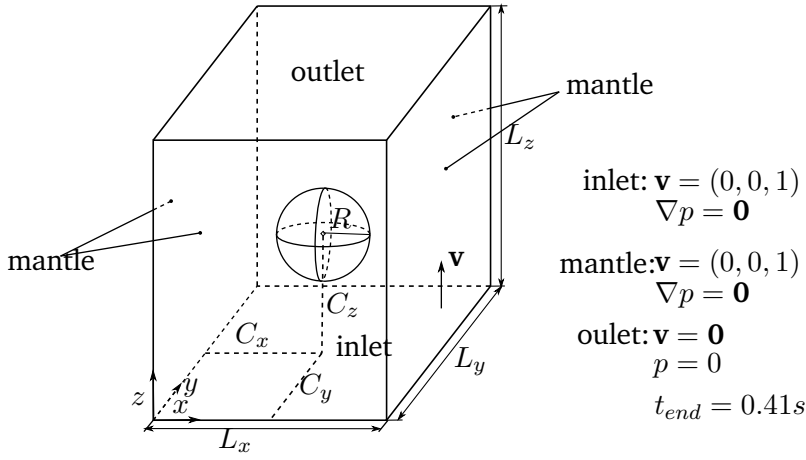
$$\Delta t \leq \Delta t_{cw} = \sqrt{\frac{(\rho_d + \rho_a) h^3}{2\pi\sigma}}, \quad (6.24)$$

in which  $\rho_d$  and  $\rho_a$  are density of droplet and ambient fluid, respectively,  $\sigma$  is the surface tension coefficient. In the case setup procedure, the smallest time step step

$$\Delta t = \min(k_{cw}\Delta t_{cw}, k_{CFL}\Delta t_{CFL}) \quad (6.25)$$

is chosen where  $k_{cw}$  and  $k_{CFL}$  are arbitrary scale factors between 0 and 1. In the following,  $k_{cw} = 0.5$  and  $k_{CFL} = 0.2$  are used.

### 6.3.2. Translating droplet



$$R = 0.2, C_x = C_y = 0.5, C_z = 0.4, L_x = L_y = 5R, L_z = 6R$$

Figure 6.4.: Translating droplet case setup.

Following the setup of Popinet [93], a sphere of radius  $R = 0.2$  translates in a rectangular domain having side lengths  $L_x = L_y = 5R, L_z = 6R$ . The initial position of the sphere's centroid is  $C_x = C_y = 0.5, C_z = 0.4$ . One corner of the rectangular domain is located in the origin as shown in fig. 6.4. The boundary conditions of the rectangular domain are set as follows:  $\nabla \mathbf{v} \cdot \mathbf{n} = 0$  and  $p = 0$  for the outlet,  $\mathbf{v} = \mathbf{v}_a$  and constant normal pressure gradient  $\partial p / \partial \mathbf{n}$  at the mantle and the inlet. The initial conditions for internal field is set to  $p(t_0) = 0$  and  $\mathbf{v}(t_0) = \mathbf{v}_a$ . The end time of simulation is set to  $t_{end} = 0.41$  s, which corresponds to a droplet displacement of one diameter.

Two groups of cases are tested to verify the  $\rho$ LENT method, respectively. For the first group, only the convection of momentum is considered, and the ambient flow has a constant density  $\rho_a = 1$ , while the density of the droplet assumes values of  $\rho_d \in$

[1, 10<sup>2</sup>, 10<sup>3</sup>, 10<sup>4</sup>], resulting in four density ratios. Three mesh resolutions  $N \in [16, 32, 64]$  are tested. For each mesh resolution  $N$ , the domain is discretized equidistantly into  $1.2N^3$  hexahedral cells, as shown in fig. 6.5. The exact solution is given by  $\mathbf{v}_c^{n+1} = \mathbf{v}_c^n = \mathbf{v}_c(t_0) = \mathbf{v}_a$  and can be used to verify the numerically consistent discretization of the single-field conservative two-phase momentum convection.

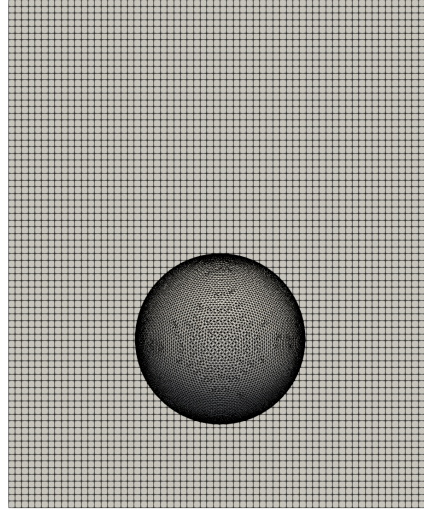


Figure 6.5.: Half section of mesh  $N = 64$ , droplet at initial position.

Viscosity and surface tension forces are included in the second test case group. A smaller range of density ratios is simulated,  $\rho^- / \rho^+ \in [1, 10, 10^2, 10^3]$ . The same kinematic viscosity is used for the ambient and the droplet phase, such that the Laplace number (eq. (5.32)) assumes values of  $La \in [120, 1200, 12000, \infty]$  for the ambient phase. Consequently, the ratio of dynamic viscosities is the same as the ratio of densities. The surface tension coefficient is constant  $\sigma = 1$ .

### Droplet translation without viscosity and surface tension forces

When the momentum is transported only by advection, no forces are exerted on the droplet body and surface. As a result, the velocity field in the overall domain should remain spatially constant and equal to  $\mathbf{v}_a = (0, 0, 1)$ . The maximum norm  $L_\infty$  is employed to measure how much the numerical velocity deviates from the analytical one, i.e.,

$$L_\infty(\mathbf{v}) = \max_i \left( \frac{\|\mathbf{v}_i - \mathbf{v}_\infty\|}{\|\mathbf{v}_\infty\|} \right), \quad (6.26)$$

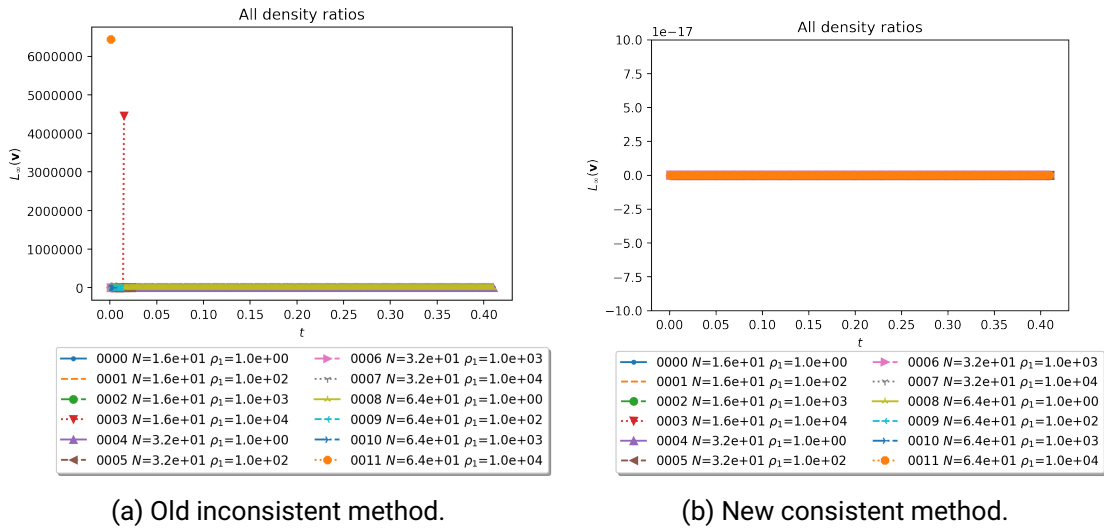


Figure 6.6.: Temporal evolution of velocity error norm  $L_\infty(\mathbf{v})$ : the left figure depicts the results from old inconsistent SAAMPLE algorithm, the right shows the results from new consistent method.

where  $\mathbf{v}_i$  denotes velocity of all interface-intersected cells, and  $\mathbf{v}_\infty = \mathbf{v}_a = (0, 0, 1)$ . The numerically inconsistent solution can cause large nonphysical interface deformations leading to a complete deterioration of the solution, visible for a verification configuration in the left image in fig. 6.7. The deterioration is amplified by the  $p - \mathbf{v}$  coupling algorithm that will calculate a pressure field  $p$  that enforces  $\nabla \cdot \mathbf{v} = 0$ . This, in turn, causes artificial acceleration in all cells where  $\mathbf{v}_c^{n+1} \neq \mathbf{v}_a$ . The consistent  $\rho$ LENT method ensures the shape of the droplet is preserved, as shown on the right image in fig. 6.7.

The fig. 6.6a contains the velocity error calculated with the old, inconsistent method. Every line in the diagram is labeled by the number of the case, mesh resolution  $N$ , and droplet density  $\rho^-$ . The default ambient density is 1. Thus, the  $\rho^-$  also represents the density ratio. As shown in fig. 6.6a, the cases with the density ratio  $\rho^- = 1$  run successfully until end time  $t_{end} = 0.41s$ . All cases with a density ratio higher than 1, namely  $\rho^- > 1$ , diverge and stop early. Cases with a very high density ratio of  $10^4$  (e.g., case 0011 and 0003) fail catastrophically.

When  $\rho$ LENT is used, as shown in fig. 6.6b, the velocity error remains exactly 0 in all cases. This means that the interface velocity remains consistent with the ambient flow and is unaffected by the mesh resolution and density ratio. The results demonstrate the exact recovery of numerical consistency for the convection of the two-phase momentum,

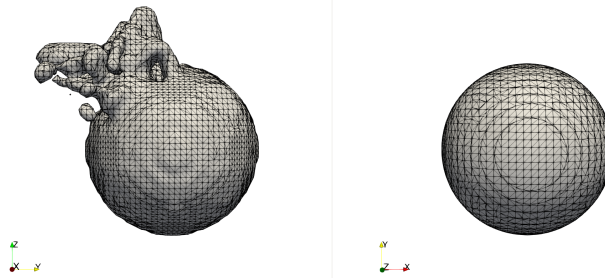
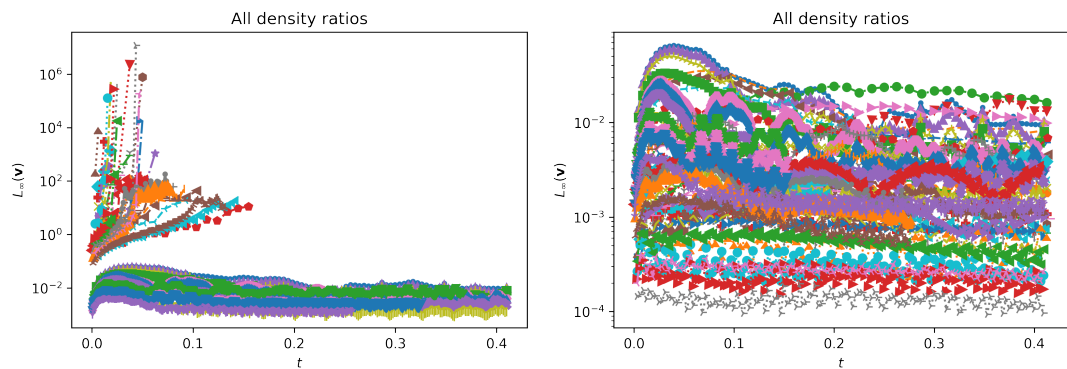


Figure 6.7.: Comparison of the strong interface deformation with the inconsistent LENT method (left) and the numerically consistent interface shape of the  $\rho$ LENT method. Parameters:  $N = 64$ ,  $\rho^-/\rho^+ = 10^4$ ,  $t = 0.0008s$ .

using the conservative formulation of single-field two-phase Navier-Stokes equations.

### Droplet translation with viscosity and surface tension forces



(a) Old inconsistent method: interface stable only for cases with density ratio  $\rho^-/\rho^+ = 1$  (b) New consistent method: interface stable for density ratios  $\rho^-/\rho^+ \in [1, 10, 100, 1000]$

Figure 6.8.: Temporal evolution of velocity error norm  $L_\infty(\mathbf{v})$  for the viscous flow with surface tension forces: the left diagram depicts the results from the old inconsistent method, and the right diagram contains the results from the  $\rho$ LENT method. The legends of these diagrams are large, and the full information is available in Appendix A: fig. A.1 for fig. 6.8a, fig. A.2 for fig. 6.8b.



Here, viscous and capillary forces are taken into account when solving the momentum equation. Since SAAMPLE is a well-balanced algorithm, as demonstrated in chapter 5, surface tension force is balanced by the pressure gradient, if the curvature is exactly calculated and propagated as a constant in the normal direction with respect to the interface. In the absence of gravity, such a droplet does not accelerate or decelerate. In other words, the velocity error  $L_\infty$  stays the same, namely 0, in theory. The temporal evolution of  $L_\infty$  is shown in fig. 6.8. The inconsistent method remains stable only for  $\rho^-/\rho^+ = 1$ . For the results of all other cases, i.e., with  $\rho^-/\rho^+ > 1$ , the velocity error increases exponentially, and the simulations crash. In contrast, as depicted in fig. 6.8b, the  $\rho$ LENT demonstrates numerically stable results for all tested density ratios. Additional numerical errors are introduced compared with two-phase momentum advection, specifically when approximating the curvature (see section 5.6). Therefore,  $L_\infty$  cannot exactly be equal to zero, as shown in fig. 6.6b. However, as seen in fig. 6.8b, the final  $L_\infty$  error given by eq. (6.26)  $10^{-4}$  and  $10^{-2}$ , which is acceptable.

### Translating sub-millimeter droplet with realistic physical properties

materials/properties (25 °C)	density ( $\text{kg m}^{-3}$ )	kinematic viscosity ( $\text{m}^2 \text{s}^{-1}$ )	surface tension ( $\text{N m}^{-1}$ )	density ratio
air	1.1839	$1.562 \times 10^{-5}$	---	--- [3]
water	997.05	$8.926 \times 10^{-7}$	0.07213 (in air)	842.17 (in air) [3]
mercury	$13.5336 \times 10^3$	$1.133 \times 10^{-7}$	0.4855 (in air)	11431.37(in air) [3]
silicone oil (cSt 10)	$0.934 \times 10^3$	$1.088 \times 10^{-5}$	0.0201 (in air)	788.92(in air) [138]
silicone oil (cSt 50)	$0.96 \times 10^3$	$5 \times 10^{-5}$	0.032 (in water)	0.96 (in water) [125]

Table 6.1.: Realistic fluid properties are combined into four tests: water droplet/air ambient, mercury droplet/air ambient, silicone oil droplet/air ambient, silicone oil droplet/water ambient.

Table 6.1 contains the physical properties used for the test-case configuration of the translating sub-millimeter droplet with realistic physical properties. In terms of size, a spherical droplet of radius  $R = 0.25$  mm is translating a distance of three diameters with velocity 0.01 m/s in  $z$ -direction of the rectangular solution domain ( $L_x = L_y = 5R$ ,  $L_z = 10R$ ). The initial centroid position of the droplet is  $(2.5R, 2.5R, 2R)$ . Surface tension and viscous forces are not considered for this setup. As depicted in fig. 6.9, it is obvious that  $L_\infty(\mathbf{v})$  remains stable over time when the droplet translates. Even in the cases with a density ratio of over 10000, as shown in fig. 6.9d, no matter how high the resolution is, the results from  $\rho$ LENT the method can reach machine precision.

Apart from the observation mentioned above, table 6.2 reveals another advantage of  $\rho$ LENT method, that is, the conspicuous computational efficiency. As shown in table 6.2,

the execution time of the solver using  $\rho$ LENT method to simulate a case is always short.

case	resolution	execution time (s)
silicone oil droplet / water	16	6.62
	32	74.34
	64	929.79
water droplet / air	16	7.77
	32	91.64
	64	1324.38
mercury droplet / air	16	7.71
	32	94.57
	64	1334.36
silicone oil droplet / air	16	7.31
	32	83.43
	64	1318.43

Table 6.2.: Execution time for the  $\rho$ LENT method.

## 6.4. Conclusions

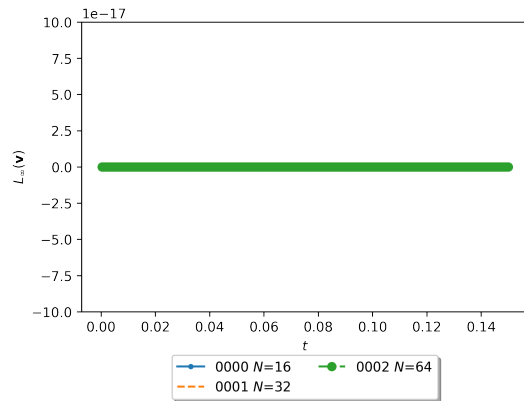
The proposed  $\rho$ LENT method exactly ensures numerical consistency of the single-field incompressible two-phase momentum convection, discretized by the unstructured collocated Finite-Volume Method. The  $\rho$ LENT method is straightforward and can be applied directly to any two-phase flow simulation method that relies on the collocated Finite-Volume Method for equation discretization: the only difference is the computation of area fractions  $\alpha_f^{n+1}$  from the approximated fluid interface  $\tilde{\Sigma}^{n+1}$ . It is shown, by analyzing the two-phase momentum advection equation, that the numerical consistency requires the computation of the cell-centered density  $\rho_c^{n+1}$  using a mass flux identical to the one used in the two-phase momentum convective term. This provides the theoretical reasoning behind the auxiliary mass conservation equation, originally introduced by Ghods and Herrmann [40]. Following the importance of the face-centered (mass flux) density pointed out by [139], the expression for the mass flux density using the principle of mass conservation is derived and the mass flux density is connected with the phase indicator. Here, this is achieved by avoiding the temporal integration of the conserved property as done very recently by Arrufat et al. [8], which allows to express the mass fluxes using the phase indicator in a discrete setting. The consistent cell-centered density  $\rho_c^{n+1}$  is used in the  $p - \mathbf{v}$  coupling algorithm section 5.8 to obtain the velocity  $\mathbf{v}_c^{n+1}$ , necessary to evolve the fluid

---

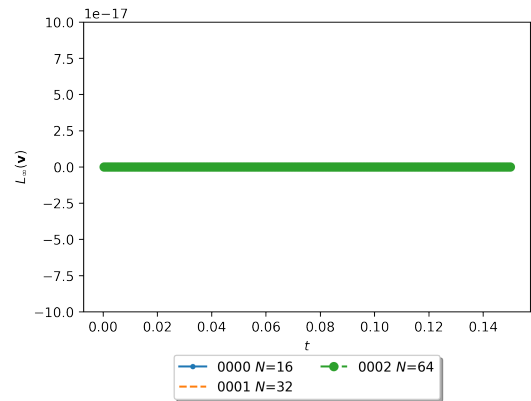
---

interface in the next step from  $t^{n+1}$  to  $t^{n+2}$ . Once the velocity is obtained by  $p - \mathbf{v}$  coupling, the cell-centered density  $\rho_c^{n+1}$  is again made consistent with the fluid interface. Using the face-centered (mass-flux) density in the  $p - \mathbf{v}$  coupling and advecting the interface first, enables  $\rho$ LENT to discretize the momentum convection term implicitly, compared to the explicit convective term discretization that is used by Bussmann et al. [20] and Ghods and Herrmann [40] in the collocated Finite-Volume setting. The consistency of the mass flux in the auxiliary density equation with the mass flux computed using the phase indicator, justifies theoretically the use of the same schemes for these two fluxes by Ghods and Herrmann [40], Patel and Natarajan [90], and Manik et al. [67].

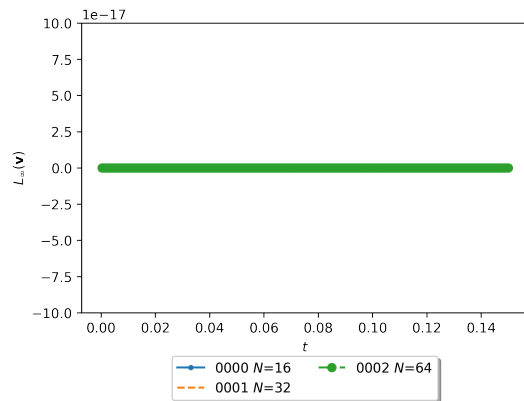
Results demonstrate the recovery of an exact solution, with the error in the  $L_\infty$  norm exactly equaling 0, for the canonical droplet translation verification case studies [93]. Droplets with sub-millimeter diameters and with realistic fluid properties are also advected exactly. Validation cases with realistic surface tension forces and viscosity demonstrate numerical stability of  $\rho$ LENT, resulting in the *relative*  $L_\infty$  norm for the parasitic currents between  $10^{-4}$  and  $10^{-2}$  for realistic density ratios.



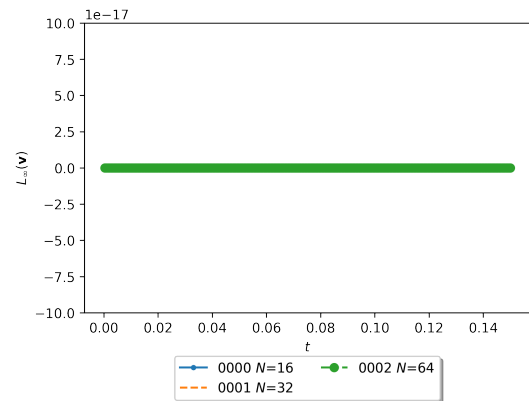
(a) Silicone oil droplet in water, density ratio 0.96



(b) Silicone oil droplet in air, density ratio 788.92



(c) Water droplet in air, density ratio 842.17



(d) Mercury droplet in air, density ratio 11431.37

Figure 6.9.: Temporal evolution of velocity error norm  $L_\infty(\mathbf{v})$  with pure advection:  $\rho$ LENT method used in simulating two-phase flows with different density ratios, mesh resolution:  $N = 16, 32, 64$ .

---

## 7. Summary and outlook

---

Within this thesis an existing Level Set / Front Tracking method using an unstructured Finite-Volume method [71] for equation discretization has been extended to make simulations of capillary flows feasible. Several aspects of the original method have been improved for this purpose. First, a more accurate approximation of the interface curvature has been devised. The new approach still relies on the Eulerian Level Set field, but ensures curvature is only approximated for cells intersected by the interface. For these cells, the absolute accuracy is further increased by a so-called spherical correction which accounts for the distance between cell centre and fluid interface. Propagation of curvature within a narrow band around the interface is achieved by adopting the idea of compact curvature calculation [104], modified for unstructured meshes. It is demonstrated that the absolute accuracy is improved considerably with these techniques while also showing convergence across the entire range of mesh resolutions investigated. In addition, the convergence behavior of the reconstruction operator used in the OpenFOAM framework [82] is examined. It is derived that this operator yields second-order accuracy for special meshes and first-order accuracy for general unstructured meshes given that the original field is at least  $C^1$ . A novel pressure velocity coupling algorithm termed SAAMPLE is proposed for the segregated solution of pressure and velocity for two-phase flows. In contrast to the PISO algorithm [49] it builds upon, it is driven by solution accuracy and accounts for the non-linearity of the convective term. The overall improvements for capillary flows are demonstrated for canonical two-phase verification and validation test cases.

The necessity for consistency between mass fluxes and the cell-centred density when dealing with high density differences is clarified. For a momentum balance in conservative form in the absence of additional forces, it is shown that the updated cell-centred density has to match the mass fluxes across cell faces. Otherwise, the mass inconsistency leads to an artificial acceleration or deceleration in order to preserve momentum. As these consistent fluxes are not inherently available for Level Set or Front Tracking methods, a new approach termed  $\rho$ LENT is proposed that approximates mass fluxes by computing area fractions for faces intersected by the fluid interface. These fluxes are used to compute an intermediate cell density for the solution of the momentum balance equation. It is demonstrated that this approach recovers exactly the momentum convection for a


---

translating droplet in the absence of other forces and ensures numerical stability for the translating droplet under influence of viscous and surface-tension forces.

An efficient and robust method, the SMCI/A algorithm, is proposed that enables the calculation of signed distances and volume fractions from oriented triangulated surfaces. Building upon an octree search structure, a novel normal computation is introduced which enables a robust inside/outside computation with respect to the surface mesh triangle normals. An approximate solution of a diffusion equation is used to propagate inside/outside information from a small narrow band around the surface mesh to the entire domain efficiently. Volume fractions are computed either by means of geometrical intersection or approximation from signed distances in combination with adaptive tetrahedral refinement. For spheres and ellipsoids it is verified that the accuracy of both approaches is ultimately limited by the surface mesh resolution. Furthermore, the global volume error is second-order convergent with respect to surface mesh resolution for both approaches. The SMCI/A algorithm is applicable to complex triangulated surfaces, making it a promising algorithm for volume fraction and signed distance computation in other simulation contexts. Implementation of the SMCI/A algorithm publicly available and full reproducibility of results is given by a fixed virtualized computing environment and by fully automated test case execution and evaluation [68].

While substantial accuracy improvements have been achieved for the LENT method when applied to capillary flows further work is required to make predictive simulations for technical applications feasible. The current iso-surface reconstruction approach limits the method to cases with small to moderate interface deformation. Moreover, it also limits the maximum of simulated physical time due to the accumulation of numerical interface perturbations. Thus, an alternative approach to the iso-surface reconstruction is required. Higher-order interpolation of the signed distances is desirable to reduce volume and position errors introduced by reconstruction. This will also allow to reconstruct the interface frequently, thereby ensuring a proper surface mesh quality. Given a surface mesh with less perturbations and larger triangles may also make curvature approximation directly on the discrete interface feasible.

Another point of future work is the replacement of the reconstruction operator which is not suited for discontinuous or strongly varying fields. It makes sense to consider this question together with the mathematical model for surface tension and its numerical discretization to ensure the resulting method is *well-balanced* [35, 94]. Following the ideas proposed in [94, 2], an interesting avenue for surface tension is to base a model on the integral formulation. As shown in [2], such an approach allows to have a conservative discretization of surface forces. In addition, this may remove the necessity to approximate interface curvature and rely on interface normals only, meaning only first derivatives of the interface position would be required.



---

For any potential solution to the future research tasks listed above, their computational costs and their suitability for parallelization should be considered. These are important aspects to make predictive simulations of technical applications feasible in the long run.





---

## Bibliography

---

- [1] T. Abadie, J. Aubin, and D. Legendre. “On the combined effects of surface tension force calculation and interface advection on spurious currents within Volume of Fluid and Level Set frameworks”. In: *Journal of Computational Physics* 297 (2015), pp. 611–636. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2015.04.054.
- [2] Moataz O. Abu-Al-Saud, Stéphane Popinet, and Hamdi A. Tchelepi. “A conservative and well-balanced surface tension model”. In: *Journal of Computational Physics* 371 (2018), pp. 896–913. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.02.022.
- [3] Arthur W Adamson, Alice Petry Gast, et al. *Physical chemistry of surfaces, 6th Edition*. Vol. 150. New York: Wiley, 1997.
- [4] Hyung Taek Ahn and Mikhail Shashkov. “Multi-material interface reconstruction on generalized polyhedral meshes”. In: *J. Comput. Phys.* 226.2 (2007), pp. 2096–2132. ISSN: 00219991. DOI: 10.1016/j.jcp.2007.06.033.
- [5] M.A. Alves, P.J. Oliveira, and F.T. Pinho. “A convergent and universally bounded interpolation scheme for the treatment of advection”. In: *International journal for numerical methods in fluids* 41.1 (2003), pp. 47–75. DOI: 10.1002/flid.428.
- [6] M. Andersson et al. “A review of cell-scale multiphase flow modeling, including water management, in polymer electrolyte fuel cells”. In: *Appl. Energy* 180 (2016), pp. 757–778. ISSN: 03062619. DOI: 10.1016/j.apenergy.2016.08.010.
- [7] Wojciech Aniszewski, Thibaut Ménard, and Maciej Marek. “Volume of Fluid (VOF) type advection methods in two-phase flow: A comparative study”. In: *Computers & Fluids* 97 (2014), pp. 52–73. DOI: 10.1016/j.compfluid.2014.03.027.
- [8] T. Arrufat et al. “A mass-momentum consistent, Volume-of-Fluid method for incompressible flow on staggered grids”. In: *Comput. Fluids* 215 (2021), p. 104785. ISSN: 00457930. DOI: 10.1016/j.compfluid.2020.104785.
- [9] E. Aulisa, S. Manservigi, and R. Scardovelli. “A mixed markers and volume-of-fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows”. In: *J. Comput. Phys.* 188.2 (2003), pp. 611–639. ISSN: 00219991. DOI: 10.1016/S0021-9991(03)00196-7.

- 
- [10] E. Aulisa et al. “Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry”. In: *J. Comput. Phys.* 225.2 (2007), pp. 2301–2319. ISSN: 00219991. DOI: 10.1016/j.jcp.2007.03.015.
- [11] J. A. Baerentzen and H. Aanaes. “Signed distance computation using the angle weighted pseudonormal”. In: *IEEE Transactions on Visualization and Computer Graphics* 11.3 (2005), pp. 243–253. DOI: 10.1109/TVCG.2005.49.
- [12] I. E. Barton. “Comparison of SIMPLE- and PISO-type algorithms for transient flows”. In: *Int. J Numer. Methods. Fluids.* 26.4 (Feb. 1998), pp. 459–483. ISSN: 0271-2091. DOI: 10.1002/(SICI)1097-0363(19980228)26:4<459::AID-FLD645>3.0.CO;2-U.
- [13] Steffen Basting and Martin Weismann. “A hybrid level set-front tracking finite element approach for fluid-structure interaction and two-phase flow applications”. In: *J. Comput. Phys.* 255 (2013), pp. 228–244. ISSN: 10902716. DOI: 10.1016/j.jcp.2013.08.018.
- [14] Steffen Basting and Martin Weismann. “A hybrid level set/front tracking approach for finite element simulations of two-phase flows”. In: *J. Comput. Appl. Math.* 270 (2014), pp. 471–483. ISSN: 03770427. DOI: 10.1016/j.cam.2013.12.014.
- [15] J. Bloomenthal. “An Implicit Surface Polygonizer”. In: *Graphics Gems IV*. Academic Press, 1994, pp. 324–349.
- [16] S. Bnà et al. “Numerical integration of implicit functions for the initialization of the VOF function”. In: *Comput. Fluids* 113 (2015), pp. 42–52. DOI: 10.1016/j.compfluid.2014.04.010.
- [17] S. Bnà et al. “Vofi - A library to initialize the volume fraction scalar field”. In: *Comput. Phys. Commun.* 200 (2016), pp. 291–299. ISSN: 00104655. DOI: 10.1016/j.cpc.2015.10.026.
- [18] J.U. Brackbill, D.B. Kothe, and C. Zemach. “A continuum method for modeling surface tension”. In: *Journal of Computational Physics* 100.2 (1992), pp. 335–354. ISSN: 0021-9991. DOI: 10.1016/0021-9991(92)90240-Y.
- [19] Günter Brenn. *Analytical solutions for transport processes*. Springer, 2016.
- [20] Markus Bussmann, Douglas B. Kothe, and James M. Sicilian. “Modeling high density ratio incompressible interfacial flows”. In: *American Society of Mechanical Engineers, Fluids Engineering Division (Publication) FED* 257.1 B (2002), pp. 707–713. DOI: 10.1115/FEDSM2002-31125.

- 
- 
- [21] Hector D. Cenicerros et al. “A Robust, Fully Adaptive Hybrid Level-Set/Front-Tracking Method for Two-Phase Flows with an Accurate Surface Tension Computation”. In: *Commun. Comput. Phys.* (2010), pp. 1–47. DOI: 10.4208/cicp.050509.141009a.
- [22] Gregor Cerne, Stojan Petelin, and Iztok Tiselj. “Numerical errors of the volume-of-fluid interface tracking algorithm”. In: *International journal for numerical methods in fluids* 38.4 (2002), pp. 329–350. DOI: 10.1002/flid.228.
- [23] Vincent Le Chenadec and Heinz Pitsch. “A 3D unsplit Forward/Backward Volume-of-Fluid approach and coupling to the level set method”. In: *J. Comput. Phys.* 233.1 (2013), pp. 10–33. ISSN: 00219991. DOI: 10.1016/j.jcp.2012.07.019.
- [24] S. J. Cummins, M. M. Francois, and Douglas B. Kothe. “Estimating curvature from volume fractions”. In: *Comput. Struct.* 83.6-7 (2005), pp. 425–434. ISSN: 00457949. DOI: 10.1016/j.compstruc.2004.08.017.
- [25] M. Darwish and F. Moukalled. “A Unified Formulation of the Segregated Class of Algorithms for Fluid Flow at all Speeds”. In: *Numerical Heat Transfer, Part B* 7790.March (2001). ISSN: 1040-7790. DOI: 10.1080/104077901750475887.
- [26] R.B. DeBar. *Fundamentals of the KRAKEN code. [Eulerian hydrodynamics code for compressible nonviscous flow of several fluids in two-dimensional (axially symmetric) region]*. Tech. rep. California Univ., Livermore (USA). Lawrence Livermore Lab., 1974.
- [27] Fabian Denner and Berend G.M. van Wachem. “Numerical time-step restrictions as a result of capillary waves”. In: *Journal of Computational Physics* 285 (2015), pp. 24–40. ISSN: 10902716. DOI: 10.1016/j.jcp.2015.01.021.
- [28] O. Desjardins and V. Moureau. “Methods for multiphase flows with high density ratio”. In: *Center for Turbulence Research Proceedings of the Summer Program* (2010), pp. 313–322. URL: [https://web.stanford.edu/group/ctr/Summer/SP10/6\\_02\\_desjardins.pdf](https://web.stanford.edu/group/ctr/Summer/SP10/6_02_desjardins.pdf).
- [29] S. S. Desphande, L. Anumolu, and M. F. Trujillo. “Evaluating the performance of the two-phase flow solver interFoam”. In: *Comput. Sci. Discov.* 5.014016 (2012), pp. 1–36. DOI: 10.1088/1749-4699/5/1/014016.
- [30] Miles Detrixhe and Tariq D. Aslam. “From level set to volume of fluid and back again at second order accuracy”. en. In: *Int. J. Numer. Methods Fluids* 80.4 (Feb. 2015), pp. 231–255. ISSN: 02712091. DOI: 10.1002/flid.4076.

- 
- 
- [31] Hang Ding, Peter DM Spelt, and Chang Shu. “Diffuse interface model for incompressible two-phase flows with large density ratios”. In: *Journal of Computational Physics* 226.2 (2007), pp. 2078–2095. DOI: 10.1016/j.jcp.2007.06.028.
- [32] Sai C Divi et al. “Error-estimate-based adaptive integration for immersed isogeometric analysis”. In: *Comput. Math. with Appl.* 80.11 (2020), pp. 2481–2516. ISSN: 0898-1221. DOI: 10.1016/j.camwa.2020.03.026.
- [33] Vadim Dyadechko and Mikhail Shashkov. “Reconstruction of multi-material interfaces from moment data”. In: *Journal of Computational Physics* 227.11 (2008), pp. 5361–5384. ISSN: 00219991. DOI: 10.1016/j.jcp.2007.12.029.
- [34] Robert Eymard, Thierry Gallouët, and Raphaële Herbin. “Finite Volume Methods”. In: *Solution of Equation in  $R^n$  (Part 3), Techniques of Scientific Computing (Part 3)*. Ed. by J. L. Lions and Philippe Ciarlet. Vol. 7. Handbook of Numerical Analysis. Elsevier, 2000, pp. 713–1020. DOI: 10.1016/S1570-8659(00)07005-8. URL: <https://hal.archives-ouvertes.fr/hal-02100732>.
- [35] M. M. Francois et al. “A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework”. In: *J. Comput. Phys.* 213.1 (2006), pp. 141–173. ISSN: 00219991. DOI: 10.1016/j.jcp.2005.08.004.
- [36] T. P. Fries and S. Omerović. “Higher-order accurate integration of implicit geometries”. In: *Int. J. Numer. Methods Eng.* 106.5 (2016), pp. 323–371. ISSN: 10970207. DOI: 10.1002/nme.5121.
- [37] Zhiliang Gao, Qiuxin Gao, and Dracos Vassalos. “Numerical simulation of flooding of a damaged ship”. In: *Ocean Engineering* 38.14-15 (2011), pp. 1649–1662. DOI: 10.1016/j.oceaneng.2011.07.020.
- [38] Christophe Geuzaine and Jean-François Remacle. “Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities”. In: *International journal for numerical methods in engineering* 79.11 (2009), pp. 1309–1331.
- [39] S. Ghali. *Introduction to geometric computing*. Springer Science & Business Media, 2008. URL: <https://www.springer.com/gp/book/9781848001145>.
- [40] S. Ghods and M. Herrmann. “A consistent rescaled momentum transport method for simulating large density ratio incompressible multiphase flows using level set methods”. In: *Physica Scripta* 88.T155 (2013). ISSN: 00318949. DOI: 10.1088/0031-8949/2013/T155/014050.

- 
- 
- [41] Frederic Gibou, Ronald Fedkiw, and Stanley Osher. “A review of level-set methods and some recent applications”. In: *Journal of Computational Physics* 353 (2018), pp. 82–109. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2017.10.006.
- [42] James Glimm et al. “Three-Dimensional Front Tracking”. In: *SIAM J. Sci. Comput.* 19.3 (1998), pp. 703–727. ISSN: 1064-8275. DOI: 10.1137/S1064827595293600.
- [43] Bernhard Godderidge et al. “An investigation of multiphase CFD modelling of a lateral sloshing tank”. In: *Computers & Fluids* 38.2 (2009), pp. 183–193. DOI: 10.1016/j.compfluid.2007.11.007.
- [44] Maximilian Hartmann et al. “Breakup Dynamics of Capillary Bridges on Hydrophobic Stripes”. In: *International Journal of Multiphase Flow* (2021), p. 103582.
- [45] William M. Haynes. *CRC handbook of chemistry and physics*. 94. ed. Boca Raton, Fla. [u.a.], 2013. ISBN: 9781466571143.
- [46] C. W. Hirt and B. D. Nichols. “Volume of fluid (VOF) method for the dynamics of free boundaries”. In: *Journal of Computational Physics* 39.1 (1981), pp. 201–225. ISSN: 10902716. DOI: 10.1016/0021-9991(81)90145-5.
- [47] Ziyang Huang, Guang Lin, and Arezoo M. Ardekani. “Consistent, essentially conservative and balanced-force Phase-Field method to model incompressible two-phase flows”. In: *Journal of Computational Physics* 406 (2020), p. 109192. ISSN: 10902716. DOI: 10.1016/j.jcp.2019.109192.
- [48] T. Inamuro et al. “A lattice Boltzmann method for incompressible two-phase flows with large density differences”. In: *Journal of Computational physics* 198.2 (2004), pp. 628–644. DOI: 10.1016/j.jcp.2004.01.019.
- [49] R.I Issa. “Solution of the implicitly discretised fluid flow equations by operator-splitting”. In: *Journal of Computational Physics* 62.1 (1986), pp. 40–65. ISSN: 0021-9991. DOI: 10.1016/0021-9991(86)90099-9.
- [50] C. B. Ivey and P. Moin. “Accurate interface normal and curvature estimates on three-dimensional unstructured non-convex polyhedral meshes”. In: *J. Comput. Phys.* 300 (2015), pp. 365–386. ISSN: 10902716. DOI: 10.1016/j.jcp.2015.07.055.
- [51] Christopher B. Ivey and Parviz Moin. “Conservative and bounded volume-of-fluid advection on unstructured grids”. In: *J. Comput. Phys.* 350 (2017), pp. 387–419. ISSN: 10902716. DOI: 10.1016/j.jcp.2017.08.054.
- [52] H. Jasak. “Error analysis and estimation for the finite volume method with applications to fluid flows.” PhD thesis. Imperial College London (University of London), 1996.

- 
- 
- [53] Matthew Jemison et al. “A coupled level set-moment of fluid method for incompressible two-phase flows”. In: *J. Sci. Comput.* 54.2-3 (2013), pp. 454–491. ISSN: 08857474. DOI: 10.1007/s10915-012-9614-7.
- [54] Lluís Jofre et al. “A 3-D Volume-of-Fluid advection method based on cell-vertex velocities for unstructured meshes”. In: *Comput. Fluids* 94 (2014), pp. 14–29. ISSN: 00457930. DOI: 10.1016/j.compfluid.2014.02.001.
- [55] Bevan W.S. Jones, Arnaud G. Malan, and Niran A. Ilangakoon. “The initialisation of volume fractions for unstructured grids using implicit surface definitions.” In: *Comput. Fluids* 179 (2019), pp. 194–205. ISSN: 00457930. DOI: 10.1016/j.compfluid.2018.10.021.
- [56] F. Juretić. “Error analysis in finite volume CFD”. PhD thesis. Imperial College London (University of London), 2005.
- [57] F. Juretić. *The cfMesh library for polyhedral mesh generation*. Accessed: 2022-09-12. URL: <https://sourceforge.net/projects/cfmesh/>.
- [58] Myungjoo Kang, Ronald P. Fedkiw, and Xu-Dong Liu. “A boundary condition capturing method for multiphase incompressible flow”. In: *Journal of Scientific Computing* 15.3 (2000), pp. 323–360. DOI: 10.1023/A:1011178417620.
- [59] Hyun-Jung Kim, Yu-Deok Seo, and Sung-Kie Youn. “Isogeometric analysis for trimmed CAD surfaces”. In: *Comput. Methods Appl. Mech. Eng.* 198.37-40 (2009), pp. 2982–2995. DOI: 10.1016/j.cma.2009.05.004.
- [60] Johannes Kromer and Dieter Bothe. “Highly accurate computation of volume fractions using differential geometry”. In: *J. Comput. Phys.* 396.July (2019), pp. 761–784. ISSN: 00219991. DOI: 10.1016/j.jcp.2019.07.005.
- [61] Andrew Kuprat et al. “Volume Conserving Smoothing for Piecewise Linear Curves, Surfaces, and Triple Lines”. In: *Journal of Computational Physics* 172.1 (2001), pp. 99–118. ISSN: 0021-9991. DOI: 10.1006/jcph.2001.6816.
- [62] Benjamin Lalanne et al. “On the computation of viscous terms for incompressible two-phase flows with Level Set/Ghost Fluid Method”. In: *Journal of Computational Physics* 301 (2015), pp. 289–307. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2015.08.036.
- [63] Vincent Le Chenadec and Heinz Pitsch. “A monotonicity preserving conservative sharp interface flow solver for high density ratio two-phase flows”. In: *Journal of Computational Physics* 249 (2013), pp. 185–203. ISSN: 10902716. DOI: 10.1016/j.jcp.2013.04.027.

- 
- 
- [64] Taehun Lee and Ching-Long Lin. “A stable discretization of the lattice Boltzmann equation for simulation of incompressible two-phase flows at high density ratio”. In: *Journal of Computational Physics* 206.1 (2005), pp. 16–47. DOI: 10.1016/j.jcp.2004.12.001.
- [65] Xiaoyi Li and Marios C. Soteriou. “High fidelity simulation and analysis of liquid jet atomization in a gaseous crossflow at intermediate Weber numbers”. In: *Physics of Fluids* 28.8 (Aug. 2016), p. 082101. ISSN: 1070-6631. DOI: 10.1063/1.4959290.
- [66] J. López et al. “Non-convex analytical and geometrical tools for volume truncation, initialization and conservation enforcement in VOF methods”. In: *J. Comput. Phys.* 392 (2019), pp. 666–693. ISSN: 10902716. DOI: 10.1016/j.jcp.2019.04.055.
- [67] Jai Manik, Amaresh Dalal, and Ganesh Natarajan. “A generic algorithm for three-dimensional multiphase flows on unstructured meshes”. In: *International Journal of Multiphase Flow* 106 (2018), pp. 228–242. ISSN: 03019322. DOI: 10.1016/j.ijmultiphaseflow.2018.04.010.
- [68] T. Marić, T. Tolle, and D. Gründing. *The argo OpenFOAM module: the implementation of Surface Mesh Cell Intersection / Approximation algorithms*. <https://gitlab.com/leia-methods/argo/-/tree/2021-02-17-SMCIA-SUBMISSION>. Accessed: 2022-09-12.
- [69] Tomislav Marić, Douglas B. Kothe, and Dieter Bothe. “Unstructured un-split geometrical Volume-of-Fluid methods – A review”. In: *Journal of Computational Physics* 420 (2020), p. 109695. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109695.
- [70] Tomislav Marić, Holger Marschall, and Dieter Bothe. “An enhanced un-split face-vertex flux-based VoF method”. In: *Journal of Computational Physics* 371 (2018), pp. 967–993. ISSN: 10902716. DOI: 10.1016/j.jcp.2018.03.048.
- [71] Tomislav Marić, Holger Marschall, and Dieter Bothe. “lentFoam – A hybrid Level Set/Front Tracking method on unstructured meshes”. In: *Computers & Fluids* 113 (2015). Small scale simulation of multiphase flows, pp. 20–31. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2014.12.019.
- [72] Tomislav Marić, Tobias Tolle, and Jun Liu. *A collocated unstructured finite volume Level Set / Front Tracking method for two-phase flows with large density-ratios - data*. <https://doi.org/10.48328/tudatalib-627>. [Online; accessed 03-September-2021]. 2021.



- 
- 
- [73] D. Meagher. “Geometric modeling using octree encoding”. In: *Computer Graphics and Image Processing* 19.2 (1982), pp. 129–147.
- [74] D. P. Mehta and S. Sahni. *Handbook of data structures and applications*. CRC Press, 2004.
- [75] S. Menon and D. P. Schmidt. “Conservative interpolation on unstructured polyhedral meshes: An extension of the supermesh approach to cell-centered finite-volume variables”. In: *Comput. Methods Appl. Mech. Eng.* 200.41-44 (2011), pp. 2797–2804. ISSN: 00457825. DOI: 10.1016/j.cma.2011.04.025.
- [76] R Mittal and G Iaccarino. “Immersed boundary methods”. In: *Annu. Rev. Fluid Mech.* 37 (2005), pp. 239–261.
- [77] F. Moukalled, L. Mangani, M. Darwish, et al. *The finite volume method in computational fluid dynamics*. Springer, 2016.
- [78] Metin Muradoglu and Gretar Tryggvason. “A front-tracking method for computation of interfacial flows with soluble surfactants”. In: *Journal of computational physics* 227.4 (2008), pp. 2238–2262. DOI: 10.1016/j.jcp.2007.10.003.
- [79] Nishant Nangia et al. “A robust incompressible Navier-Stokes solver for high density ratio multiphase flows”. In: *Journal of Computational Physics* 390 (2019), pp. 548–594. ISSN: 10902716. DOI: 10.1016/j.jcp.2019.03.042. arXiv: 1809.01008.
- [80] Alessandro Nitti et al. “An immersed-boundary/isogeometric method for fluid–structure interaction involving thin shells”. In: *Computer Methods in Applied Mechanics and Engineering* 364 (2020), p. 112977. ISSN: 0045-7825. DOI: 10.1016/j.cma.2020.112977.
- [81] W. F. Noh and P. R. Woodward. “SLIC (Simple Line Interface Calculation) method”. In: *Proc. Fifth Int. Conf. Numer. Methods Fluid Dyn. June 28–July 2, 1976 Twente Univ. Enschede* (1976), pp. 330–340. DOI: 10.1007/3-540-08004-X\_336.
- [82] OpenCFD Ltd. *OpenFOAM: user guide*. Last accessed on 2022-08-01. 2022. URL: <https://www.openfoam.com/documentation/user-guide>.
- [83] Annagrazia Orazzo et al. “A vof-based consistent mass-momentum transport for two-phase flow simulations”. In: *American Society of Mechanical Engineers, Fluids Engineering Division (Publication) FEDSM*. Vol. 1C-2017. American Society of Mechanical Engineers, July 2017, pp. 1–11. ISBN: 9780791858066. DOI: 10.1115/FEDSM2017-69190.



- 
- 
- [84] Stanley Osher and Ronald P. Fedkiw. “Level Set Methods: An Overview and Some Recent Results”. In: *Journal of Computational Physics* 169.2 (2001), pp. 463–502. ISSN: 0021-9991. DOI: 10.1006/jcph.2000.6636.
- [85] Mark Owkes and Olivier Desjardins. “A computational framework for conservative, three-dimensional, unsplit, geometric transport with application to the volume-of-fluid (VOF) method”. In: *J. Comput. Phys.* 270 (2014), pp. 587–612. ISSN: 10902716. DOI: 10.1016/j.jcp.2014.04.022.
- [86] Mark Owkes and Olivier Desjardins. “A mass and momentum conserving unsplit semi-Lagrangian framework for simulating multiphase flows”. In: *J. Comput. Phys.* 332 (2017), pp. 21–46. ISSN: 10902716. DOI: 10.1016/j.jcp.2016.11.046.
- [87] Mark Owkes and Olivier Desjardins. “A mass and momentum conserving unsplit semi-Lagrangian framework for simulating multiphase flows”. In: *Journal of Computational Physics* 332 (2017), pp. 21–46. ISSN: 10902716. DOI: 10.1016/j.jcp.2016.11.046.
- [88] Mark Owkes and Olivier Desjardins. “A mesh-decoupled height function method for computing interface curvature”. In: *J. Comput. Phys.* 281 (2015), pp. 285–300. ISSN: 10902716. DOI: 10.1016/j.jcp.2014.10.036.
- [89] S. V. Patankar and D. B. Spalding. “A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows”. In: *International Journal of Heat and Mass Transfer* 15.10 (Oct. 1972), pp. 1787–1806. ISSN: 00179310. DOI: 10.1016/0017-9310(72)90054-3.
- [90] Jitendra Kumar Patel and Ganesh Natarajan. “A novel consistent and well-balanced algorithm for simulations of multiphase flows on unstructured grids”. In: *Journal of Computational Physics* 350 (2017), pp. 207–236. ISSN: 10902716. DOI: 10.1016/j.jcp.2017.08.047.
- [91] Chiara Pesci. “Computational Analysis of Fluid Interfaces Influenced by Soluble Surfactant”. en. PhD thesis. Darmstadt: Technische Universität, July 2019. URL: <http://tuprints.ulb.tu-darmstadt.de/9303/>.
- [92] Daniel Podbiel et al. “Fusing MEMS technology with lab-on-chip: nanoliter-scale silicon microcavity arrays for digital DNA quantification and multiplex testing”. In: *Microsystems & Nanoengineering* 6 (2020), pp. 2055–7434. DOI: 10.1038/s41378-020-00187-1.
- [93] Stéphane Popinet. “An accurate adaptive solver for surface-tension-driven interfacial flows”. In: *J. Comput. Phys.* 228.16 (2009), pp. 5838–5866. ISSN: 00219991. DOI: 10.1016/j.jcp.2009.04.042.

- 
- [94] Stéphane Popinet. “Numerical Models of Surface Tension”. In: *Annual Review of Fluid Mechanics* 50 (Jan. 2018), pp. 49–75. DOI: 10.1146/annurev-fluid-122316-045034. URL: <https://hal.archives-ouvertes.fr/hal-01528255>.
- [95] Andrea Prosperetti. “Motion of two superposed viscous fluids”. In: *The Physics of Fluids* 24.7 (1981), pp. 1217–1223. DOI: 10.1063/1.863522.
- [96] M. Raessi, M. Bussmann, and J. Mostaghimi. “A semi-implicit finite volume implementation of the CSF method for treating surface tension in interfacial flows”. In: *Int. J. Numer. Methods Fluids* 59.10 (Apr. 2009), pp. 1093–1110. ISSN: 02712091. DOI: 10.1002/flid.1857.
- [97] Mehdi Raessi and Heinz Pitsch. “Consistent mass and momentum transport for simulating incompressible interfacial flows with large density ratios using the level set method”. In: *Computers and Fluids* 63 (2012), pp. 70–81. ISSN: 00457930. DOI: 10.1016/j.compfluid.2012.04.002.
- [98] William J. Rider and Douglas B. Kothe. “Reconstructing Volume Tracking”. In: *Journal of Computational Physics* 141.2 (1998), pp. 112–152. ISSN: 00219991. DOI: 10.1006/jcph.1998.5906.
- [99] Murray Rudman. “A volume-tracking method for incompressible multifluid flows with large density variations”. In: *International Journal for Numerical Methods in Fluids* 28.2 (1998), pp. 357–378. ISSN: 02712091. DOI: 10.1002/(SICI)1097-0363(19980815)28:2<357::AID-FLD750>3.0.CO;2-D.
- [100] G. Russo and P. Smereka. “A Remark on Computing Distance Functions”. In: *J. Comput. Phys.* 163.1 (2000), pp. 51–67. ISSN: 00219991. DOI: 10.1006/jcph.2000.6553.
- [101] Henning Scheufler and Johan Roenby. “Accurate and efficient surface reconstruction from volume fraction data on general meshes”. In: *J. Comput. Phys.* 383 (2019), pp. 1–23. ISSN: 10902716. DOI: 10.1016/j.jcp.2019.01.009. arXiv: 1801.05382.
- [102] Robert Schmidt, Roland Wüchner, and Kai-Uwe Bletzinger. “Isogeometric analysis of trimmed NURBS geometries”. In: *Computer Methods in Applied Mechanics and Engineering* 241-244 (2012), pp. 93–111. ISSN: 0045-7825. DOI: 10.1016/j.cma.2012.05.021.
- [103] J. A. Sethian. “A fast marching level set method for monotonically advancing fronts.” In: *Proceedings of the National Academy of Sciences* 93.4 (1996), pp. 1591–1595. DOI: 10.1073/pnas.93.4.1591.

- 
- 
- [104] S. Shin and D. Juric. “A hybrid interface method for three-dimensional multiphase flows based on front tracking and level set techniques”. In: *Int. J. Numer. Methods Fluids* 60.7 (2009), pp. 753–778. DOI: 10.1002/flid.1912.
- [105] S. Shin and D. Juric. “Modeling Three-Dimensional Multiphase Flow Using a Level Contour Reconstruction Method for Front Tracking without Connectivity”. In: *Journal of Computational Physics* 180.2 (2002), pp. 427–470. ISSN: 00219991. DOI: 10.1006/jcph.2002.7086.
- [106] Seungwon Shin, Jalel Chergui, and Damir Juric. “A solver for massively parallel direct numerical simulation of three-dimensional multiphase flows”. In: *J. Mech. Sci. Technol.* 31.4 (2017), pp. 1739–1751. ISSN: 1738494X. DOI: 10.1007/s12206-017-0322-y. arXiv: 1410.8568.
- [107] Seungwon Shin and Damir Juric. “High order level contour reconstruction method”. In: *J. Mech. Sci. Technol.* 21.2 (2007), pp. 311–326. ISSN: 1738494X. DOI: 10.1007/BF02916292.
- [108] Seungwon Shin and Damir Juric. “Modeling three-dimensional multiphase flow using a level contour reconstruction method for front tracking without connectivity”. In: *Journal of Computational Physics* 180.2 (2002), pp. 427–470. DOI: 10.1006/jcph.2002.7086.
- [109] Seungwon Shin, Ikroh Yoon, and Damir Juric. “The Local Front Reconstruction Method for direct simulation of two- and three-dimensional multiphase flows”. In: *Journal of Computational Physics* 230.17 (2011), pp. 6605–6646. ISSN: 00219991. DOI: 10.1016/j.jcp.2011.04.040.
- [110] Seungwon Shin et al. “Accurate representation of surface tension using the level contour reconstruction method”. In: *Journal of Computational Physics* 203.2 (2005), pp. 493–516. ISSN: 00219991. DOI: 10.1016/j.jcp.2004.09.003.
- [111] S. Soukane and F. Trochu. “Application of the level set method to the simulation of resin transfer molding”. In: *Composites Science and Technology* 66.7-8 (2006), pp. 1067–1080. DOI: 10.1016/j.compscitech.2005.03.001.
- [112] Severin Strobl, Arno Formella, and Thorsten Pöschel. “Exact calculation of the overlap volume of spheres and mesh elements”. In: *Journal of Computational Physics* 311 (2016), pp. 158–172. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2016.02.003.

- 
- 
- [113] M. Sussman and E. Fatemi. “An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow”. In: *SIAM Journal on scientific computing* 20.4 (1999), pp. 1165–1191. DOI: 10.1137/S1064827596298245.
- [114] M. Sussman et al. “An adaptive level set approach for incompressible two-phase flows”. In: *Journal of Computational Physics* 148.1 (1999), pp. 81–124. DOI: 10.1006/jcph.1998.6106.
- [115] M. Sussman et al. “An improved level set method for incompressible two-phase flows”. In: *Computers & Fluids* 27.5-6 (1998), pp. 663–680. DOI: 10.1016/S0045-7930(97)00053-4.
- [116] Mark Sussman and Mitsuhiro Ohta. “A Stable and Efficient Method for Treating Surface Tension in Incompressible Two-Phase Flow”. In: *SIAM Journal on Scientific Computing* 31.4 (Jan. 2009), pp. 2447–2471. ISSN: 1064-8275. DOI: 10.1137/080732122.
- [117] Mark Sussman and Elbridge Gerry Puckett. “A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows”. In: 337 (2000), pp. 301–337. DOI: 10.1006/jcph.2000.6537.
- [118] Mark Sussman, Peter Smereka, and Stanley Osher. “A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow”. In: *J. Comput. Phys.* 114.1 (1994), pp. 146–159. DOI: 10.1006/jcph.1994.1155.
- [119] Mark Sussman et al. “A sharp interface method for incompressible two-phase flows”. In: *Journal of computational physics* 221.2 (2007), pp. 469–505. DOI: 10.1016/j.jcp.2006.06.020.
- [120] G. Thürrner and Wüthrich C. A. “Computing Vertex Normals from Polygonal Facets”. In: *Journal of Graphics Tools* 3.1 (1998), pp. 43–46. DOI: 10.1080/10867651.1998.10487487.
- [121] Tobias Tolle, Dieter Bothe, and Tomislav Marić. “SAAMPLE: A Segregated Accuracy-driven Algorithm for Multiphase Pressure-Linked Equations”. In: *Computers & Fluids* 200 (2020), p. 104450. DOI: 10.1016/j.compfluid.2020.104450.
- [122] Tobias Tolle et al. *Computing volume fractions and signed distances from arbitrary surfaces on unstructured meshes: diagram data*. 2021-02-18. DOI: 10.48328/tudatalib-410. URL: <https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2581.3>.
- [123] C. D. Toth, J. O’Rourke, and Jacob E. Goodman. *Handbook of discrete and computational geometry*. Chapman and Hall/CRC, 2017.

- 
- 
- [124] G.M. Treece, R.W. Prager, and A.H. Gee. “Regularised marching tetrahedra: improved iso-surface extraction”. In: *Computers & Graphics* 23.4 (1999), pp. 583–598. ISSN: 0097-8493. DOI: 10.1016/S0097-8493(99)00076-X.
- [125] E. Trinh and T. G. Wang. “Large-amplitude free and driven drop-shape oscillations: Experimental observations”. In: *Journal of Fluid Mechanics* 122.-1 (Sept. 1982), pp. 315–338. ISSN: 14697645. DOI: 10.1017/S0022112082002237.
- [126] G. Tryggvason et al. “A Front-Tracking Method for the Computations of Multiphase Flow”. In: *J. Comput. Phys.* 169.2 (2001), pp. 708–759.
- [127] Grétar Tryggvason, Ruben Scardovelli, and Stéphane Zaleski. *Direct Numerical Simulations of Gas–Liquid Multiphase Flows*. Cambridge University Press, 2011. ISBN: 978-0-521-78240-1.
- [128] O. Ubbink. “Numerical prediction of two fluid systems with sharp interfaces”. PhD thesis. Imperial College of Science, Technology and Medicine, 1997.
- [129] Salih Ozen Unverdi and Grétar Tryggvason. “A front-tracking method for viscous, incompressible, multi-fluid flows”. In: *Journal of Computational Physics* 100.1 (1992), pp. 25–37. ISSN: 10902716. DOI: 10.1016/0021-9991(92)90307-K.
- [130] G. Vaudor et al. “A consistent mass and momentum flux computation method for two phase flows. Application to atomization process”. In: *Computers and Fluids* 152 (2017), pp. 204–216. ISSN: 00457930. DOI: 10.1016/j.compfluid.2017.04.023.
- [131] G. Vaudor et al. “A consistent mass and momentum flux computation method using rudman-type technique with a clsvof solver”. In: *Fluids Engineering Division Summer Meeting*. Vol. 46230. American Society of Mechanical Engineers, 2014, V01CT23A012. DOI: 10.1115/FEDSM2014-21802.
- [132] Cesar M. Venier et al. “On the stability analysis of the PISO algorithm on collocated grids”. In: *Computers & Fluids* 147 (2017), pp. 25–40. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2017.01.017.
- [133] Y. Wang et al. “A mass-conserved diffuse interface method and its application for incompressible multiphase flows with large density ratio”. In: *Journal of Computational Physics* 290 (2015), pp. 336–351. ISSN: 10902716. DOI: 10.1016/j.jcp.2015.03.005.
- [134] H. G. Weller et al. “A tensorial approach to computational continuum mechanics using object-oriented techniques”. In: *Comput. Phys.* 12.6 (1998), pp. 620–631. DOI: 10.1063/1.168744.

- 
- 
- [135] Zixuan Yang, Min Lu, and Shizhao Wang. “A robust solver for incompressible high-Reynolds-number two-fluid flows with high density contrast”. In: *Journal of Computational Physics* (2021), p. 110474. DOI: 10.1016/j.jcp.2021.110474.
- [136] Stéphane Zaleski and Feng Xiao. “Special issue: Numerical methods and modeling of multiphase flows”. In: *Journal of Computational Physics* 402, 108902 (Feb. 2020), p. 108902. DOI: 10.1016/j.jcp.2019.108902.
- [137] H.W. Zheng, Chang Shu, and Yong-Tian Chew. “A lattice Boltzmann model for multiphase flows with large density ratio”. In: *Journal of computational physics* 218.1 (2006), pp. 353–371. DOI: 10.1016/j.jcp.2006.02.015.
- [138] R Zivojnovic. *Silicone oil in vitreoretinal surgery*. Vol. 12. Springer Science & Business Media, 2012.
- [139] Davide Zuzio et al. “A new efficient momentum preserving Level-Set/VOF method for high density and momentum ratio incompressible two-phase flows”. In: *Journal of Computational Physics* 410 (2020), p. 109342. ISSN: 10902716. DOI: 10.1016/j.jcp.2020.109342.



---

## **A. Additional figures**

---

### **A.1. Parameter study figures with legends**

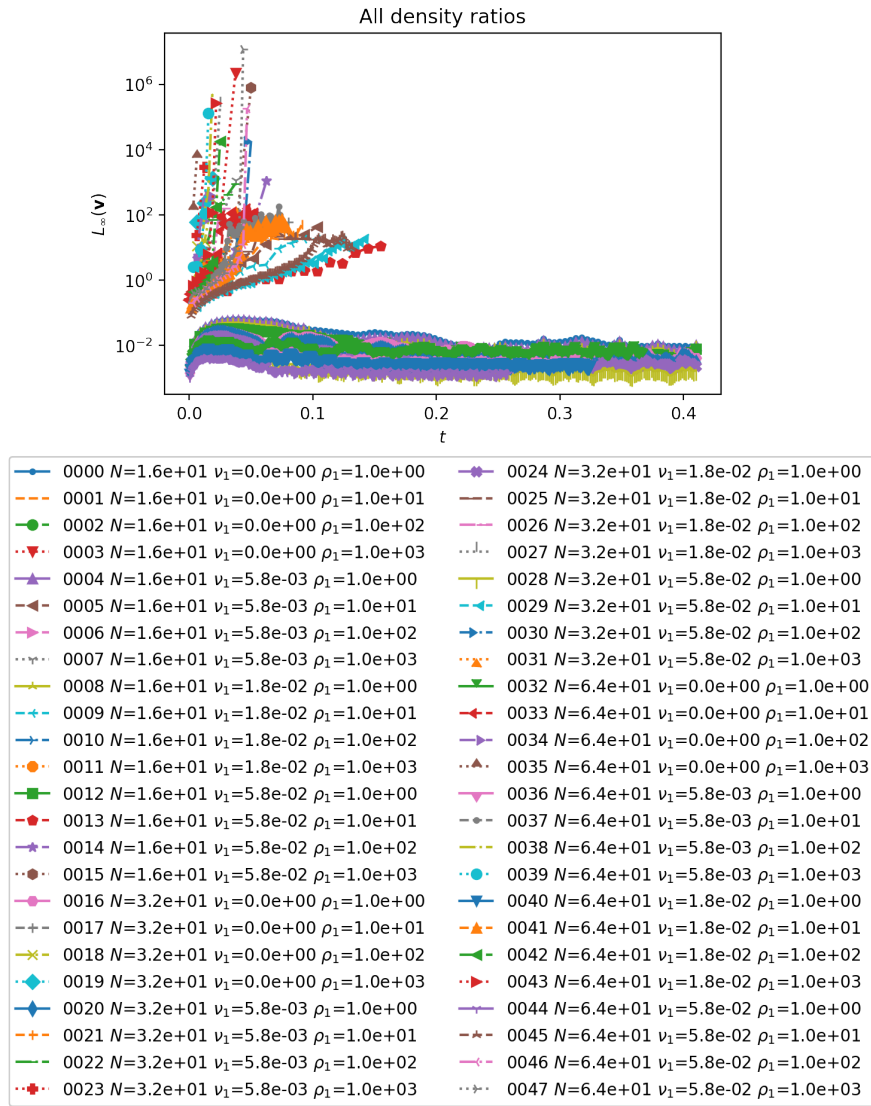
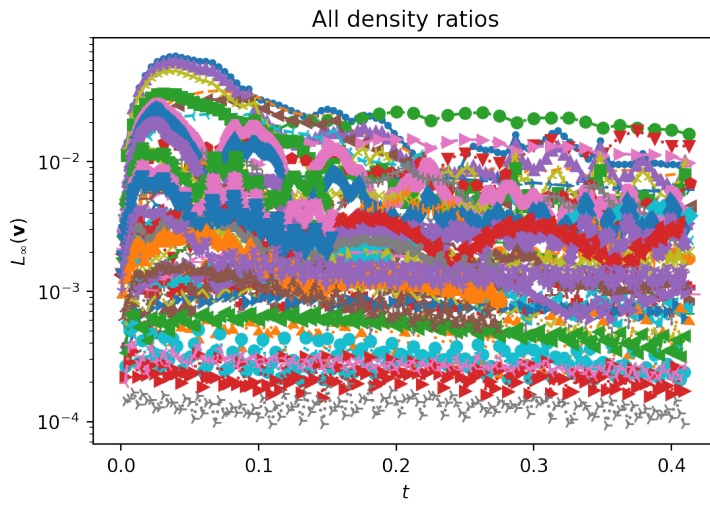


Figure A.1.: Full figure of fig. 6.8a





0000 $N=1.6e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+00$	0024 $N=3.2e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+00$
0001 $N=1.6e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+01$	0025 $N=3.2e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+01$
0002 $N=1.6e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+02$	0026 $N=3.2e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+02$
0003 $N=1.6e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+03$	0027 $N=3.2e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+03$
0004 $N=1.6e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+00$	0028 $N=3.2e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+00$
0005 $N=1.6e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+01$	0029 $N=3.2e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+01$
0006 $N=1.6e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+02$	0030 $N=3.2e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+02$
0007 $N=1.6e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+03$	0031 $N=3.2e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+03$
0008 $N=1.6e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+00$	0032 $N=6.4e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+00$
0009 $N=1.6e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+01$	0033 $N=6.4e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+01$
0010 $N=1.6e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+02$	0034 $N=6.4e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+02$
0011 $N=1.6e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+03$	0035 $N=6.4e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+03$
0012 $N=1.6e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+00$	0036 $N=6.4e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+00$
0013 $N=1.6e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+01$	0037 $N=6.4e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+01$
0014 $N=1.6e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+02$	0038 $N=6.4e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+02$
0015 $N=1.6e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+03$	0039 $N=6.4e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+03$
0016 $N=3.2e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+00$	0040 $N=6.4e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+00$
0017 $N=3.2e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+01$	0041 $N=6.4e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+01$
0018 $N=3.2e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+02$	0042 $N=6.4e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+02$
0019 $N=3.2e+01$ $v_1=0.0e+00$ $\rho_1=1.0e+03$	0043 $N=6.4e+01$ $v_1=1.8e-02$ $\rho_1=1.0e+03$
0020 $N=3.2e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+00$	0044 $N=6.4e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+00$
0021 $N=3.2e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+01$	0045 $N=6.4e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+01$
0022 $N=3.2e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+02$	0046 $N=6.4e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+02$
0023 $N=3.2e+01$ $v_1=5.8e-03$ $\rho_1=1.0e+03$	0047 $N=6.4e+01$ $v_1=5.8e-02$ $\rho_1=1.0e+03$

Figure A.2.: Full figure of fig. 6.8b



---

## Curriculum Vitae

---

- 09/04 - 06/07 **Secondary school II**, *Jacob-Grimm-Schule*, Kassel, Germany,  
Abitur
- 10/08 - 10/11 **Bachelor's studies**, *Technische Universität Darmstadt*, Darmstadt, Germany,  
Bachelor of Science Mechanical and Process Engineering  
Bachelor's thesis: *Implementierung und Validierung einer hybriden Parallelisierungsmethode in FASTEST*
- 10/11 - 12/15 **Master's studies**, *Technische Universität Darmstadt*, Darmstadt, Germany,  
Master of Science Mechanical and Process Engineering  
Master's thesis: *Direct Numerical Simulations (DNS) of Two-Phase Flows Using the LENT Hybrid Level Set / Front Tracking Method in OpenFOAM*
- 02/16 - 11/22 **Doctoral studies**, *Technische Universität Darmstadt*, Darmstadt, Germany,  
scholarship holder of the IRTG 1529; subsequently research assistant in the working group *Mathematical Modeling and Analysis*
- 04/17 - 06/17 **Research stay**, *Waseda University*, Tokyo, Japan
- 09/14/2022 **Submission of the doctoral thesis (Dissertation)**, *An unstructured Finite-Volume Level Set / Front Tracking method for capillary flows at Technische Universität Darmstadt*, Darmstadt, Germany
- 11/24/22 **Defense of the doctoral thesis**, overall assessment *magna cum laude / sehr gut bestanden*