A Mass-Conserving hybrid interface capturing method for geometrically complicated domains

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties, in het openbaar te verdedigen
op woensdag 15 juni 2016 om 10:00 uur

door

Fahim RAEES
Master of Science in Applied Mathematics,
University of Karachi, Pakistan

geboren te Karachi, Pakistan.

This dissertation has been approved by the promotor:
Prof. dr. ir. C. Vuik

Composition of the doctoral committee:

Rector Magnificus,	Chairman
Prof. dr. ir. C. Vuik,	Promotor, Delft University of Technology
Dr. ir. D. R. Van der Heul,	Copromotor, Delft University of Technology

Independent members:

Prof. dr. A.E.P. Veldman,	University of Groningen
Prof. dr. ir. B. Koren,	Eindhoven University of Technology
Prof. dr. ir. R.A.W.M. Henkes,	Delft University of Technology
Prof. dr. ir. B.J. Boersma,	Delft University of Technology
Prof. dr. ir. B.G.M. van Wachem,	Imperial College London

The work described in this thesis was financially supported by the NED University of Engineering and Technology, Karachi, Pakistan and Delft University of Technology, Delft, The Netherlands.

A Mass-Conserving hybrid interface capturing method for geometrically complicated domains.
Dissertation at Delft University of Technology.
Copyright © 2016 by Fahim Raees
ISBN 978-94-6186-667-7

To my parents:
Raees Hussain & Shahida Parveen

Contents

Summary **ix**

Samenvatting **xii**

1 Introduction **1**

1.0.1 Modeling immiscible, incompressible two-phase flow 2

1.0.2 Interface capturing and interface tracking methods . 4

1.1 Simulation of two-phase flows in geometrically complex domains 6

1.2 Outline of the thesis 7

2 The Level-Set based interface capturing method **9**

2.1 Introduction 9

2.2 The Level-Set method 9

2.3 The Level-Set method for modeling two-phase flow 10

2.4 Discontinuous Galerkin discretisation of the Level-Set equation 12

2.4.1 Spatial discretization of the Level-Set equation . . . 13

2.4.2 Temporal Discretisation 15

2.5 Test Cases 17

2.5.1 Linear translation of a circular interface 18
2.5.2 Rotation of a circular interface 19
2.5.3 The reversed vortex test case 20
2.6 The influence of the degree of the polynomial representation on mass conservation 24
2.7 Conclusions 25

3 An overview of the LS and VoF hybrid methods 27
3.1 Introduction 27
3.2 Hybrid Methods 28
3.2.1 CLSVOF method 28
3.2.2 Hybrid particle LS method 28
3.2.3 MCLS method 29
3.2.4 VOSET method 29
3.3 Conclusions 30

4 MCLS method for unstructured triangular meshes 31
4.1 Introduction 31
4.2 Modeling the evolution of the interface 31
4.2.1 The Level-Set field 31
4.2.2 Volume of Fluid field 32
4.2.3 Hybrid formulation of the MCLS method 32
4.3 Overview of the MCLS algorithm 33
4.4 Efficient computation of the Volume of Fluid field from the LS field and vice versa 34
4.4.1 Converting the LS field to the Volume of Fluid field 35
4.4.2 The Volume of Fluid function 36
4.4.3 Combining the VoF functions for Case-1 and Case-2 43
4.5 Advection of the LS field 46
4.6 Advection of the Volume of Fluid field 47
4.6.1 Mass conserving advection of the Volume of Fluid field for nonlinear velocity field 50

4.7 Inverse function for the Level-Set correction 55
4.8 Computational cost of the mass-conserving correction . . . 57
4.9 Test Cases 58
    4.9.1 Conversion between level-set and volume of fluid representation 58
    4.9.2 Translation of circular region 60
    4.9.3 Rotation of circular region around the center of the domain 61
    4.9.4 The reverse vortex test case 62
4.10 Conclusions 63

5 Extension to tetrahedron control volumes 65

5.1 Introduction 65
5.2 Volume of Fluid function in 3D 65
    5.2.1 Coordinates of the point ξ^E 67
    5.2.2 Coordinates of the points ξ^F and ξ^G 67
    5.2.3 Evaluation of the VoF from the LS function 68
    5.2.4 Case: When two vertices are aside 68
    5.2.5 For point X_I^1 69
    5.2.6 For point X_I^2 70
    5.2.7 Extra volume 71
5.3 Inverse function in 3D 72
5.4 Test Case: The 3D back and forth interface reconstruction . 73
5.5 Conclusions 74

6 The Modified Level-Set method 75

6.1 Introduction 75
6.2 The Modified Level-Set method 77
6.3 A vertex-based limiter for the modified level-set function . . 81
    6.3.1 Limiter for linear polynomial expansion of the modified level-set field 82

6.3.2 Limiter for degree two polynomial expansion of the level-set field 83

6.4 Compressive velocity formulation 85

6.5 Test Cases 86

6.5.1 Translation of the interface in one spatial dimension 87

6.5.2 Rotation of a circular interface in two spatial dimensions 91

6.5.3 The reversed vortex test case 93

6.6 Conclusions 96

7 Comparision of the LS, MCLS and MLS method 97

7.1 Introduction 97

7.1.1 Advection of a bubble with a lens shaped cross section 98

7.1.2 Zalesak's rotating disc 102

7.2 Conclusions 104

8 Conclusions and Future Prospects 107

8.1 Conclusions 107

8.1.1 Mass conservation 108

8.1.2 The Mass-Conserving Level-Set Method in relation to other hybrid interface capturing algorithms 109

8.1.3 The modified Level-Set method in comparison to the Level-Set method 109

8.2 Outlook 109

Curriculum vitae 117

List of publications 118

Acknowledgements 121

Summary

In *multiphase* flow multiple phases, e.g. gasses, liquids and solids, occur simultaneously in the same flow domain, where they influence each other's dynamics. Many industries, e.g. oil and gas recovery, (nuclear) power generation, production of foods and chemicals rely on stable and predictable liquid-gas multiphase flows for safe transport and processing. Nowadays, cost-effective system design and operation rely indispensably on (flow) simulation technology.

When only two different fluids are present in the same domain that do not mix, and the flow speed in either is much smaller than the local speed of sound, such a multiphase flow is classified as *immiscible incompressible two-phase flow*. An important property of this type of flow is that the density is constant along streamlines and only attains two values: the density of either of the two constituent phases.

Modeling the dynamics of two incompressible immiscible fluids is far more challenging than modeling single phase flow because of the conflicting demands imposed by very strict mass conservation and accurately predicting the position of the interface between the phases, for which the smoothness and sharpness have to be preserved. The latter is especially important if the interface curvature and normal vector are required to model surface tension. Furthermore, the solution needs to comply with the jump conditions that hold at the interface.

For Cartesian flow domains it is relatively straightforward to formulate a finite volume discretisation of the flow equations that is discretely conserving mass and momentum. However, when the flow domain has a

more intricate geometry, this becomes very challenging. Development of numerical methods for solving the equations that describe the dynamics of incompressible immiscible two-phase flow on generic domains are the very active field of research. In this thesis, three such methods will be compared for their accuracy, their ability to conserve mass and their efficiency:

- The (classic) Level-Set method,
- A newly developed extension of the Mass Conserving Level Set method, for discretisation on unstructured meshes consisting of triangular control volumes.
- The *modified* Level-Set method.

In this comparison, the evolution of the interface is compared when advected by an imposed two-dimensional velocity field. The interaction between the interface evolution and the flow, as occurs in a complete model for two-phase flow is not taken into account, to eliminate the influence of the specific model chosen for this interaction.

In the Level-Set method, the interface is defined as the zero Level-Set of a signed-distance function and has notoriously poor mass conservation. Therefore, the evolution of the interface is described by a linear advection equation for the Level-Set field. The discontinuous Galerkin method can be used to discretise this type of hyperbolic partial differential equation efficiently and with a high order of accuracy on unstructured simplex meshes. Numerical experiments show that mass-conservation improves upon *hp*-refinement, but is not comparable to what can be achieved using Volume of Fluid method on Cartesian meshes. This 'classic' Level-Set method defines a baseline method that is used as starting point and reference for further development.

In the Mass-Conserving Level-Set method, that has originally been developed for discretisation on structured Cartesian meshes, the interface between both phases is described by a hybrid formulation that involves both the Level-Set field and the Volume of Fluid field that are congruent at all times. During the advection of the Level-Set field the Volume of Fluid field is used to impose a mass-conserving correction on the former field. Simultaneously, the Level-Set field is used to formulate the fluxes for the Volume of Fluid advection equation more efficiently than would be possible in the absence of this information. This synergy and the fact that explicit relations can be derived to convert Volume of Fluid to Level-Set and vice versa make the method both very efficient and strictly mass conserving. In

this thesis, the extension of this method to a discretisation on an unstructured mesh of triangular control volumes is derived. Numerical experiments support the claim of exact mass conservation of the method.

An Arbitrary Eulerian-Lagrangian 'clipping' algorithm is used for the advection of the Volume of Fluid field in the proposed extension of the Mass-Conserving Level-Set method that is exactly mass-conserving for linear velocity fields, but makes the complete algorithm quite computationally intensive. It is investigated how the accuracy and ability to conserve mass of the modified Level-Set method compare to the corresponding properties of the Mass-Conserving Level-Set method. The latter can be used as a stand-alone algorithm or as part of the Mass-Conserving Level-Set method to replace the computationally intensive Volume of Fluid advection.

In the modified Level-Set method, that is often claimed to be more accurately mass-conserving than the classic Level-Set method, the interface is defined as the Level-Set of a smooth indicator function that is an approximation of the indicator function used in the Volume of Fluid method. Just as for the classic Level-Set method the evolution of the interface is described by a linear advection equation. To circumvent oscillations in the indicator function in the vicinity of the interface, a limiter is added to the discontinuous Galerkin discretisation. By augmenting the imposed velocity field with an artificial compressive velocity field the interface remains sharply defined.

A number of test cases show that mass-conservation of the modified Level-Set method is not significantly superior to the classic Level-Set method in strict sense. The application of the limiter does not seem to affect the accuracy of the solution.

Of all three methods, the newly developed Mass-Conserving Level-Set method is only one that is mass conserving up to machine precision, but also the most computationally intensive of the three. For those applications where strict mass conservation is of less importance the modified Level-Set presents an economical alternative to the Mass-Conserving Level-Set method.

Samenvatting

Een *meerfasestroming* onderscheidt zich van een *enkelfasestroming* doordat meerdere fasen, bijvoorbeeld gassen, vloeistoffen en vaste stoffen, tegelijkertijd door hetzelfde systeem stromen en elkaars dynamica beïnvloeden. In heel veel industriële processen spelen zulke stromingen een belangrijke rol. Bij de winning van olie en gas, het opwekken van energie in een kernreactor en in de chemische en voedingsmiddelenindustrie is het van het grootste belang dat de installaties op zo'n manier worden gedimensioneerd dat gegarandeerd kan worden dat de verschillende fasen tegelijkertijd op een efficiënte, milieuvriendelijke, kosteneffectieve en vooral veilige manier door de reactor, de koelinstallatie of het pijpleidingsysteem kunnen stromen.

Als slechts twee fluida in het systeem voorkomen die niet mengbaar zijn en stromen met een snelheid die veel kleiner is dan de lokale geluidssnelheid wordt de meerfasestroming geclassificeerd als *incompressible immiscible two-phase flow*. Een belangrijke eigenschap van dit type stroming is dat de dichtheid constant is langs stroomlijnen en slechts twee waarden aan kan nemen: de dichtheid van de ene of van de andere fase.

Het onderwerp van dit proefschrift is de ontwikkeling van een numerieke methode voor het efficiënt berekenen van nauwkeurige benaderingen van oplossingen van de stromingsvergelijkingen die dit type stroming beschrijven. Terwijl numerieke methoden voor het simuleren van enkelfasestromingen grotendeels uitgekristalliseerd zijn geldt dit niet voor methoden voor de simulatie van meerfasestromingen. Dit is een gevolg van de conflicterende eisen die in dit geval aan de oplossing worden gesteld en het discontinue karakter van de oplossing zelf. Aan de ene kant moeten de stromingsvariabelen voldoen aan de sprongrelaties die gelden op het scheidingsvlak tussen beide fasen, terwijl aan de andere kant de oplossing in de buurt van het scheidingsvlak voldoende glad moet zijn om daaruit de richting van de normaalvector en de kromming van het scheidingsvlak te bepalen in het bijzonder als ook oppervlaktespanning moet worden gemodelleerd. Daarnaast wordt grote waarde gehecht aan (bijna) volmaakt behoud van massa.

In die gevallen waar het stromingsdomein Cartesisch is, is het relatief eenvoudig de stromingsvergelijkingen te discretiseren met een Eindige Volumemethode die exact massa- en impulsbehoudend is. Heeft het stromingsdomein een ingewikkelde vorm dan is dit veel moeilijker te realiseren. De ontwikkeling van numerieke methoden om de vergelijkingen die meerfasestromingen beschrijven te discretiseren op een generiek domein zijn nog volop in ontwikkeling. In dit proefschrift worden drie zulke methoden vergeleken op basis van hun nauwkeurigheid, massabehoud en efficiëntie.

- De (klassieke) Level-Setmethode,
- Een nieuw ontwikkelde uitbreiding van de Mass Conserving Level-Setmethode naar een discretisatie op basis van driehoekige controlevolumes.
- De *modified* Level-Set methode.

In de vergelijking wordt gekeken naar de advectie van het scheidingsvlak tussen beide fasen in een twee-dimensionaal, opgelegd stromingsveld. Hierbij wordt de wederzijdese beïnvloeding van de stroming en het model voor het scheidingsvlak, zoals die in een volledig model voor een tweefase stroming optreedt, buiten beschouwing gelaten om de invloed van de specifieke discretisatie van deze koppeling te elimineren

Omdat bij de Level-Setmethode het scheidingsvlak wordt beschreven door de Level-Set van een signed-distancefunctie wordt de evolutie van het scheidingsvlak beschreven door een lineaire advectievergelijking voor het Level-Setveld. Deze methode wordt gekenmerkt door zijn slechte massabehoud. Door gebruik te maken van een nodal-based discontinuous Galerkinmethode kan deze vergelijking eenvoudig en efficient met een hoge orde van nauwkeurigheid worden gediscretiseerd op ongestructureerde roosters bestaande uit driehoekige controlevolumes. Uit numerieke experimenten komt naar voren dat de mate waarin de methode massabehoudend is weliswaar toeneemt met *hp*-verfijning, maar dat het massabehoud niet vergelijkbaar is met dat van moderne methoden voor Cartesische roosters op basis van de Volume of Fluidmethode. Deze methode is het uitgangspunt voor de ontwikkeling van de Mass Conserving Level-Setmethode.

In de Mass Conserving Level-Setmethode, die oorspronkelijk ontwikkeld is voor een discretisatie op een gestructureerd rooster van Cartesische controlevolumes, wordt het scheidingsvlak tussen de beide fasen op een hybride manier beschreven door zowel een Level-Set- als een Volume of Fluidveld die beide in overeenstemming zijn. Bij de advectie van het Level-Setveld wordt het Volume of Fluidveld gebruikt om op het eerstgenoemde veld een massabehoudende correctie toe te passen. Tegelijkertijd maakt het Level-Setveld het mogelijk om het Volume of Fluidveld op een efficiëntere manier te advecteren dan in een standaard Volume of Fluidmethode. Deze synergie, gecombineerd met expliciete relaties waarmee beide velden naar elkaar kunnen worden geconverteerd, maakt de methode efficiënt en massabehoudend. In dit proefschrift wordt de extensie van deze methode naar een discretisatie op basis van driehoekige controlevolumes beschreven. Numerieke experimenten ondersteunen dat de methode exact massabehoudend is.

Bij de advectiestap van het Volume of Fluidveld in de uitbreiding van de Mass Conserving Level-Setmethode wordt gebruik gemaakt van een Arbitrary Eulerian Lagrangian clipping-algoritme, dat weliswaar strict massabehoudend is voor lineaire snelheidsvelden, maar relatief rekenintensief. Tenslotte is onderzocht hoe de nauwkeurigheid en het massbehoud van de modified Level-Setmethode zich verhouden tot die van de Mass Conserving Level-Setmethode. De modified Level-Set kan als opzichzelfstaande methode of als onderdeel van de Mass Conserving Level-Setmethode kunnen worden gebruikt.

In de modified Level-Setmethode, waarvan wordt beweerd dat deze veel beter massabehoudend is dan de klassieke Level-Setmethode, wordt het scheidingsvlak beschreven door een Level-Set van een gladde indicatorfunctie die een benadering is van de indicatorfunctie die wordt gebruikt in de Volume of Fluidmethode. Ook bij de modified Level-Setmethode wordt de evolutie van het scheidingsvlak beschreven door een lineaire advectievergelijking. Om te voorkomen dat de numerieke oplossing oscillaties vertoont in de directe omgeving van het scheidingsvlak wordt een limiter toegevoegd aan de discontinuous Galerkindiscretisatie. Door ook een artificieel compressief snelheidsveld toe te voegen wordt voorkomen dat door het toepassen van de limiter het scheidingsvlak zijn scherpte verliest.

In verschillende testgevallen komt naar voren dat het massabehoud van de modified Level-Setmethode maar weinig beter is dan van de klassieke Level-Setmethode en dat het toepassen van de limiter geen gevolgen heeft voor de nauwkeurigheid van de oplossing.

Van de drie methoden is de Mass Conserving Level-Setmethode de enige die exact massbehoudend is, maar ook de meest rekenintensieve. Voor die toepassingen waar massbehoud tot machinenauwkeurigheid niet relevant is vormt de modified Level-Setmethode een goedkoper alternatief.

List of Figures

1.1 Examples of two-phase flow 3

2.1 The level-set field and the corresponding zero-level contour of the interface. 10

2.2 Control volume (Ω_k). 13

2.3 The interface $\phi(\boldsymbol{x}, t) = 0$ for the linear translation test case at time levels $t = 0$ (left) and $t = T$ (right). 18

2.4 Evolution of $M(t)/M(0)$ for the translation of a LS representation of a circular interface. 19

2.5 The interface $\phi(\boldsymbol{x}, t) = 0$ for the rotation test case at time levels $t = 0,\ t = \frac{1}{4}T,\ t = \frac{3}{4}T$ and $t = T$. 20

2.6 Evolution of $M(t)/M(0)$ for the rotation of a LS representation of a circular interface. 21

2.7 The interface $\phi(\boldsymbol{x}, t) = 0$ for the reversed vortex test case at time levels $t = 0,\ t = \frac{1}{4}T,\ t = \frac{1}{2}T,\ t = \frac{3}{4}T$ and $t = T$. 22

2.8 Evolution of $M(t)/M(0)$ for the reversed vortex test case using a LS representation of a circular interface. 23

2.9 Evolution of $M(t)/M(0)$ for the reversed vortex test case using polynomial expansion of degree n=2. 25

4.1 The MCLS algorithm is presented in a flow chart to show the transfer of information between the different components involved in advancing a single time step. 34

4.2 Control volume (Ω_k), cell centroid (X_k) and arbitrary point ($X^{'}$). 35

4.3 Case-1: Triangular domain of interest, where the vertex $\boldsymbol{X}_k^{v_3}$ is mapped to the origin. 37

4.4 Interface passing through vertex ($X_k^{v_1}$). 40

4.5 Case-2: Mapping of two nodes a side. 41

4.6 Mapping of the interface parallel to one of its edge. 44

4.7 VoF function for both Case-1 and Case-2. 44

4.8 Combination of two cases. 45

4.9 Advection, remapping and distribution of the content of a control volume that is *not* intersected by the interface. . . . 49

4.10 Advection, remapping and distribution of the content of a control volume that is intersected by the interface. 50

4.11 Straight-sided triangles remain straight-sided after being advected by a linear velocity field (left). When advected by a nonlinear velocity field this property is lost (right). 51

4.12 The position of the interface in a mixed cell in the Lagrangian mesh is based on the advection of the extremities of the interface in the corresponding mixed cell in the Eulerian mesh. 53

4.13 For a nonlinear velocity field, the advected extremities of the interface are first projected on the edges of the Lagrangian control volume, before the appropriate interface position can be determined. 54

4.14 Initialization with the exact volume-of-fluid field leads to a slightly different initial position of the interface but eliminates the mass error incurred by initialization with the exact level-set field. 60

4.15 Time history of mass error for the circular fluid rotation case. 61

4.16 Time history of mass error for the reverse vortex test case. . 63

4.17 The evolution of the interface position for the reverse vortex test case. 64

5.1 Tetrahedron control volume. 66

5.2 Case: where two vertices are aside. 69

5.3 The interface zero-contour after correction for the Zalesak's slotted sphere. 74

6.1 The influence of the parameter δ on the width of the interface for a fixed value of the parameter ϵ/h. 79

6.2 The dimensionless width of the interface w/h for a fixed definition of the interface and different values of the dimensionless parameter ϵ/h. The vertical lines indicate the coordinate value $x_{\min,\max}/h$ where $|\phi_m^\epsilon(x_{\min,\max}) - (\phi_m)_{\min,\max}| < \delta$. 80

6.3 LS, Modified LS for different ϵ and VoF 81

6.4 Control volume Ω_k and it's neighbouring elements (dashed edges) that contribute to the limited solution. 82

6.5 Translation of a circular region in $\Omega_D \subset \mathbb{R}$. The solution is shown at the initial and final time. Clockwise from the top left corner: Exact solution, numerical solution without application of a limiter, numerical solution based on the application of the vertex-based limiter, numerical solution based on the application of the vertex-based limiter and a compressive velocity field. 88

6.6 The interface $\phi_m^\epsilon(\boldsymbol{x}, t) = 0.5$ for the rotation test case at time levels $t = 0,\ t = \frac{1}{4}T,\ t = \frac{3}{4}T$ and $t = T$. 92

6.7 Evolution of $M^\epsilon(t)/M^\epsilon(0)$ for the rotation of the MLS representation of a circular interface. 93

6.8 The interface $\phi_m(\boldsymbol{x}, t) = 0$ for the reversed vortex test case at time levels $t = 0,\ t = \frac{1}{4}T,\ t = \frac{1}{2}T,\ t = \frac{3}{4}T$ and $t = T$. . . 95

7.1 Initial position of the bubble with a lens shaped interface. . 99

7.2 Evolution of the mass ratio for the lens shaped interface of LS and MLS for $n = 1$. 100

7.4 Initial position of the interface in Zalesak's rotating disc. . . 103

7.8 Evolution of the mass ratio $M(t)/M(0)$ for Zalesak's rotating disc test, for the LS (left) and MLS (right) field for $n = 1$. . 104

7.3 The lens shaped interface at the final time $t = T$, for the LS and MLS for $n = 1$ (left) and $n = 2$ (right) and the MCLS for $n = 1$. 105

7.5 Zalesak's slotted disc at $t = T$, for the LS field, using $n = 1$. 106

7.6 Zalesak's slotted disc at $t = T$, for the MCLS field, using $n = 1$. 106

7.7 Zalesak's slotted disc at $t = T$, for the MLS field, using $n = 1$.106

List of Tables

2.1 Coefficients of Low-Storage five-stage fourth-order ERK method. 16

2.2 Average mass error and L_2 error of the LS field for the translation of a circular interface. 19

2.3 Average mass error and L_2 error of the LS field for the rotation of a circular interface. 21

2.4 Average mass error and L_2 error of the LS field for the reversed vortex test case. 23

2.5 Mass error for expansion in degree n=2 polynomial for the reversed vortex test case. 24

4.1 Accuracy analysis of the interface back-and-forth conversion test case. Initialization with the exact level-set field introduces a mass error that converges with second order accuracy. 59

4.2 Average mass error $\overline{E_M^n}$ and error in the level-set field $E_\Phi(T)$ for the translation of a circular interface. 61

4.3 Average mass error $\overline{E_M^n}$ and error in the level-set field $E_\Phi(T)$ for the rotation of a circular interface. 62

4.4 Average mass error $\overline{E_M^n}$ and error in the level-set field $E_\Phi(T)$ for the reverse vortex test case. 63

6.1 Convection of the interface in one spatial dimension, mass error ($|M(T)-M(0)|$), L_2-error of the modified LS field and estimated convergence rate without compressive velocity field ($K_c = 0$) for expansion in linear polynomial basis $N = 1$. . . 89

6.2 Convection of the interface in one spatial dimension, mass error ($|M(T)-M(0)|$), L_2-error of the modified LS field and estimated convergence rate without compressive velocity field ($K_c = 0$) for expansion in quadratic polynomial basis $N = 2$. 90

6.3 Convection of the interface in one spatial dimension, mass error ($|M(T)-M(0)|$), L_2-error and estimated convergence rate of the modified LS field with compressive velocity field ($K_c = 1e-5$) for expansion in linear polynomial basis $N = 1$. 90

6.4 Convection of the interface in one spatial dimension, mass error ($|M(T)-M(0)|$), L_2-error and estimated convergence rate of the modified LS field with compressive velocity field ($K_c = 1e-5$) for expansion in quadratic polynomial basis $N = 2$. 90

6.5 Average mass error and L_2 error of the MLS field for the rotation of circular test case. 93

6.6 Final mass error and L_2 error of the MLS field for the reversed vortex test case. 96

7.1 Averaged mass error and L2-norm error for the MCLS field for $n = 1$. 101

7.2 Averaged mass error and L2-norm error for the LS field for $n = 1$ and $n = 2$. 101

7.3 Averaged mass error and L2-norm error for the MLS field for $n = 1$ and $n = 2$. 101

7.4 Zalesak's slotted disc test, average mass error and L2-norm error for the LS, MCLS and MLS field for $n = 1$. 103

CHAPTER 1

Introduction

In two-phase flow, two media with possibly very different properties and in different phases are present in a single domain. This type of flow plays a very important role in many industrial applications, ranging from petroleum industry, chemical reactors, and the medical sciences.

The dynamics of two-phase flow can be very complex because of the fact that due to the movement of the interface between the two media the material properties strongly change in space and time. Furthermore, in many cases the interface is not just advected by the local flow, but a possible imbalance of inter molecular forces (surface tension) at the interface between the two media leads to local acceleration of the flow.

Based on the phases of the two media, two-phase flow can be categorised as: Liquid-Gas, Liquid-Liquid, Gas-Solid and Liquid-Solid. In all of these, sharp changes in material properties and the dynamics of the interface play an important role. Of these four, the Liquid-Gas and Gas-Solid flows present the larger challenges, because of the difference in (material) properties of both phases, e.g. density and viscosity. To illustrate how diverse the nature of two-phase flow can be, an example of each type of flow will be discussed briefly in the context of petroleum and process engineering: a bubble column reactor, a fluidized bed reactor, a stirred tank reactor and an oil/gas pipeline, as shown in Fig. 1.1, taken from [1–4]

- **Fluidized bed** In the case a chemical reaction between two gasses relies on the presence of a solid catalyst, it it of cardinal importance for the efficiency of the reactor to disperse this catalyst as homogeneously as possible in the reactor vessel. A common approach is to form solid, nearly spherical particles from the catalyst and blow the gas-mixture through the reactor vessel. For a sufficiently large flow rate, the gas-solid mixture behaves like a fluid. This type of reactor is referred to as a *fluidized bed.*

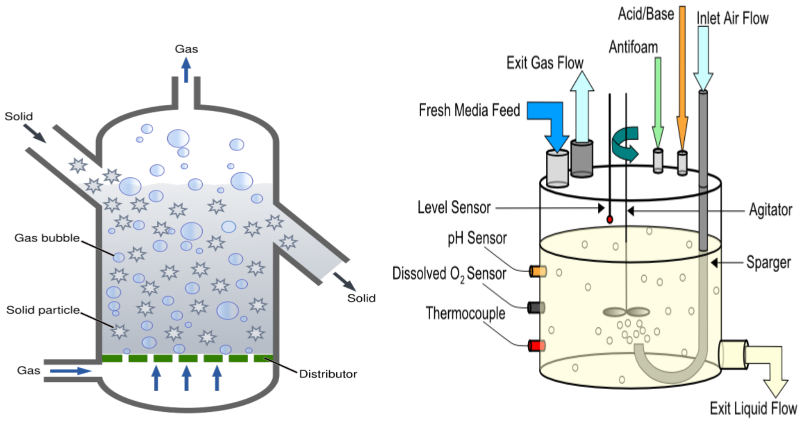
- **Stirred tank reactor** When a chemical reaction involves two liquids, use can be made of a stirred tank reactor. Optimization of this two-phase liquid-liquid flow aims at homogeneous mixing of the two constituents.

- **Bubble column reactor** Reactions between liquids and gasses, e.g. in oxidation processes and many bio-reactions, take place very efficiently in a bubble column reactor. Proper engineering of the gas injection in this gas-liquid flow guarantees finely dispersed gas bubbles with the correct rise speed to complete the reaction.

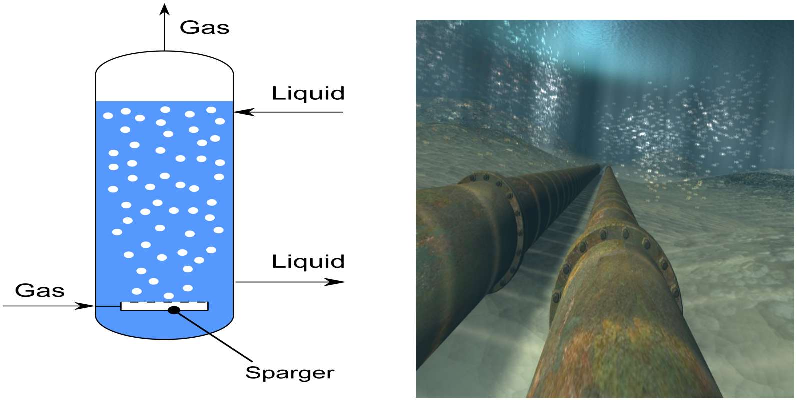
- **Oil/gas pipeline** From the production well to the processing plant, natural gas and crude oil are transported simultaneously through a single pipeline. When the gas and liquid flow rates are relatively small, gravity will separate the two phases into a *stratified* flow. For larger flow rates instabilities can grow on the oil/gas interface that ultimately lead to the formation of *slugs.* These are large chunks of the liquid phase, separated by gas bubbles. The mechanical impact of slugs on delicate monitoring equipment can have catastrophic effects and this type of gas-liquid flow has to be avoided. On the other hand two-phase flow can be utilized to greatly reduce the required pumping power for heavy crude oil, using a special way of water-injection where a thin film of water is formed surrounding a core of crude oil.

1.0.1 Modeling immiscible, incompressible two-phase flow

The research in this thesis is focused on the modeling of immiscible, incompressible gas-liquid and liquid-liquid two-phase flow: the two phases in the domain are separated by a well-defined interface and both phases move at speeds that are far smaller than the local speed of sound. The fact that the location of the interface is part of the solution, and the fact that at this interface a number of jump conditions have to be imposed on the solution makes two-phase flow considerably more hard to simulate than single-phase



(a) Fluidized bed (b) Stirred tank reactor



(c) Bubble column reactor (d) Oil/gas pipeline

Figure 1.1: Examples of two-phase flow

flow. Where for single phase flow the 'standard' order of convergence for industrial flow algorithms gradually moves from second to fourth order, this is not likely to happen for multiphase flow algorithms shortly. Currently experts predict still several decades of development are required to bring multiphase flow models to the technology readiness level of single phase flow solvers. The main difficulty lies in handling the near discontinuous change in the fluid properties across the interface and modeling the effect of surface tension that requires accurate evaluation of the local interface curvature.

1.0.2 Interface capturing and interface tracking methods

Many methods have been designed to simulate two-phase flow. The large majority of these are derived from one of the following archetypes of methods:

- The Marker Particle method [45],
- The Volume of Fluid method (VOF) [24],
- The Level-Set method (LS) [40],
- The *Modified* Level-Set method [41],
- The Front-Tracking method [50],

Each of these methods has pros and cons. Some methods are easy to implement, but do not conserve mass while others are mass conserving but have high computational complexity and this complexity further increases if the mesh is unstructured.

The Marker Particle method is one of the earliest methods for the simulation of the multiphase flow. It is based on a Lagrangian approach. In this method the initial interface is represented by a set of massless particles that are subsequently tracked while they are advected by the local velocity. This method is not computationally efficient, because of the need to track all individual particles and the need for regular particle distribution to avoid particle clusters and voids. This method alone is not a popular choice any more, however, combined with other approaches it may be advantageous [45]. The front-tracking method is another Lagrangian method. Instead of tracking fluid particles, the actual interface is tracked. Naturally the interface is a lower dimensional manifold, requiring an order of magnitude less particles to be represented. However, different from the Marker Particle method, the particles representing the interface are now connected. Generally this connectivity is fixed and only the geometry and not the connectivity of the interface can change.

If a continuous representation of the interface is required with the ability to deform without any geometrical restrictions, two methods qualify for representing the interface location: The Volume of Fluid method (VOF) and the Level-Set method. In the former, a color function is used to identify the presence of either fluid. This color function has a value 1 in one fluid and 0 in the other fluid in order to distinguish between the two. In the VOF method the cell averaged value of the color function is used to represent the quantity of either fluid present in a cell. This cell averaged value of the color function lies by definition in the interval $[0,1]$ and is known as the Volume of Fluid or the void-fraction. A cell which is completely filled with the fluid of interest has volume of fluid fraction value 1 and in the cell which is empty i.e. not containing fluid of interest volume of fluid fraction defined to be 0. The interface location is only approximately defined: it will intersect those cells that have intermediate value of the volume of fluid fraction between 0 and 1, but its exact location is unknown. In each time step the position of the interface is reconstructed within each cell that is intersected by the interface, under the assumption that the interface is either aligned with the Cartesian coordinate directions or locally planar, before it is advanced in time in a Lagrangian manner.

In the Level-Set method the interface position is indicated by the zero Level-Set of a signed distance function. The Level-Set field is advected in time to get the position of the interface. As the Level-Set function is continuous and very smooth in the vicinity of the interface geometric characteristics, like gradient and curvature are easy to obtain. No techniques exist to directly and exactly enforce the signed distance property on the Level-Set field during the advection step. This property has to be reestablished a-posteriori in a process called *reinitialization*.

The VOF method is very accurately mass-conserving, but the interface reconstruction, and the approximation of the local curvature of the interface to define surface tension effects are computationally intensive and involved. On the other hand, the Level-Set approach provides an exact (up to discretization order) representation of the interface and straightforward evaluation of the local properties of the interface, but it is by definition not mass conserving: Exact conservation of the Level-Set function does not imply conservation of mass. The modified Level-Set method [41] is a relatively new approach in which a Level-Set function is chosen that resembles the color function of the Volume-of-Fluid method. By replacing the signed-distance function by a mollified Heaviside function, mass conservation is greatly improved, but the large gradient of the modified Level-Set field at the interface makes it harder to accurately compute the unit normal vector and the curvature and to avoid oscillations in the solution, while keeping the interface sharp.

In recent years an interest has developed in coupled VOF/LS methods that combine the accurate mass conservation properties of the VOF method with the advantage of an exact representation of the interface by the level set method. For Example, the Combined Level Set Volume Of Fluid (CLSVOF) method of Sussman et al [48, 49, 56] and the Mass-Conserving

Level-Set ($MCLS$) method of Van der Pijl et al [53] for Cartesian grids. Although both examples seem to be very similar, the major difference is that the CLSVOF method basically combines the work involved in both the VOF and LS method, while the MCLS method avoids the computationally intensive interface reconstruction step of the VOF method. This is accomplished by the use of a volume of fluid function that directly relates the VOF color function to the LS function and its gradient, without the need for an explicit reconstruction step.

1.1 Simulation of two-phase flows in geometrically complex domains

While simulating two-phase flow in simple Cartesian domains is challenging, applying the same models in geometrically intricate domains presents a real challenge. However, because in many industrial applications such domains are more the rule than the exception, there is a strong drive for the development of such methods. To achieve optimal flexibility in discretizing the flow equations in more intricate domains and resolve boundary layers efficiently and accurately state-of-the-art flow simulation algorithms rely on a discretisation of the domain in an unstructured set of non overlapping control-volumes of mixed type: hexahedrons, tetrahedrons, pyramids and prisms or even more generically *general convex polyhedra.* A general convex polyhedron is any oriented domain with bounded volume that is defined by a set of intersecting planes.

Contrary to the VOF method, the LS and modified-LS methods model equations are relatively straightforward to discretize on tetrahedral cells. Using high order discontinuous Galerkin discretisation of the LS-equation mass loss can be significantly reduced with respect to second order methods. Even so the mass conservation is still not comparable with that of VOF methods that can can conserve mass nearly to machine precision. Extensions of the classic Piecewise Linear Interface Construction (PLIC) VOF method for tetrahedral control volumes currently available are very rare. They are not very attractive because they rely on very complicated geometric reconstructions to position the interface and evaluate the fluxes, that impair their robustness. Furthermore, they are costly to apply.

In this thesis different approaches to simulating immiscible two-phase flow in geometrically complex domain are proposed: The LS-method, the modified LS-method and the extension of the MCLS method of Van der Pijl [53] . The accuracy of the solution, the mass loss and the work involved in obtaining the solution are analyzed and compared.

1.2 Outline of the thesis

In this thesis an extension is presented of the MCLS for a discretisation on a set of unstructured triangular and tetrahedral control volumes as a step-up towards the extension towards discretisation on general polygonal and polyhedral control volumes.

The key components of the MCLS will be shown to be:

- a discretisation of the linear transport equation for the LS field.
- a discretisation of the transport equation for the color function of the VOF method.
- an algorithm for the back-and-forth conversion of the LS field to the VOF field, that does not involve an explicit reconstruction of the interface.

Furthermore, the performance of this extension is compared to two other methods that are applicable for discretisation on triangular control volumes: the standard and the modified Level Set method.

The outline of this thesis is as follows:

In the second chapter of the thesis a higher-order discontinuous Galerkin discretisation of the LS field is proposed. This discretisation is used in the stand-alone Level-Set method and as part of the extension of the Level-Set method. The chapter ends with an evaluation of the performance of the stand-alone LS method for three test cases.

Before presenting its extension, a concise review of the MCLS for structured Cartesian control volumes is given in the third chapter. It is essential to have a clear understanding of the MCLS to appreciate the details of the extension to triangular control volumes.

The extension of the MCLS method towards unstructured triangular grid is presented in the fourth chapter. First the derivation of the conversion function and its inverse are presented for a triangular element that is intersected by the interface. Next the convection algorithm for the color function is presented, formulated as an evolution equation for the VOF field. Finally, the same test cases as presented in the second chapter are repeated with the extension of the MCLS.

In the fifth chapter is discussed how the MCLS algorithm can be extended from two to three dimensions. The conversion function for tetrahedral control volumes is presented together with its inverse. On top of that an extension of the two-dimensional algorithm is presented to handle general polygonal control volumes, that is based on a division of the computational cells in triangular subcells.

Because of its relative simplicity and strict mass conservation properties the modified LS method presents a viable alternative to the MCLS. In the before-last chapter a discretisation based on a limited discontinuous Galerkin scheme is presented and the merits of including a compressive velocity field to enhance interface resolution are investigated.

The thesis concludes with a comparison of the performance of the three proposed methods in the last chapter, followed by conclusions and recommendations how to further improve the MCLS method.

CHAPTER 2

The Level-Set based interface capturing method

2.1 Introduction

In this chapter an interface capturing model based on the solution of the LS equation is presented. The foundations of the LS model are discussed and a linear scalar transport equation for the LS field is derived. The latter equation is formulated in weak form and discretized in space using a discontinuous Galerkin method on rectilinear triangular control volumes and a high order explicit Runge-Kutta method in time. Finally the algorithm is applied for a range of test cases, to set a benchmark for the solutions of the extension of the MCLS method and the modified-LS method that will be presented in subsequent chapters of this thesis.

2.2 The Level-Set method

The LS method is a general approach to model interface evolution problems in a very broad context ranging from solid and fluid mechanics to digital image processing. The method has been introduced by Sethian and Osher [29,40] in 1988, and is still actively developed. Because of its simplicity and robustness, the method is very popular and is included in many commercial simulation suites, e.g. COMSOL and OpenFOAM[1].

2.3 The Level-Set method for modeling two-phase flow

The flow domain Ω is a simply connected subset of $\mathbb{R}^2$ with boundary $\partial\Omega$. The (nonstationary) interface between the two phases is a curve, parametrized as $\boldsymbol{X}(s,t)$, $s \in [0, L(t)]$, where $L(t)$ is its arc length. This curve is defined as an isoline of a C^2 function $l(\boldsymbol{x},t) : \Omega \to \mathbb{R}$. Choosing $l(\boldsymbol{x},t)$ as the signed distance to the interface is one of the common choices, but not a prerequisite for the LS method. The former choice defines the interface as the zero-level contour of the signed distance function $\Phi(\boldsymbol{x},t)$. The region $S = \{\boldsymbol{x} \in \Omega | (\boldsymbol{x},t) > 0\}$ is the *domain of interest*, occupied by the *fluid of interest*. In modeling two-phase flow, strict mass conservation is regarded as very important, as even more important than the accuracy of the LS field and the exact position and shape of the interface. Because the density is constant in either phase, the area enclosed by the interface is proportional to the 'mass' of the domain of interest. The mass $M(t)$ is defined as:

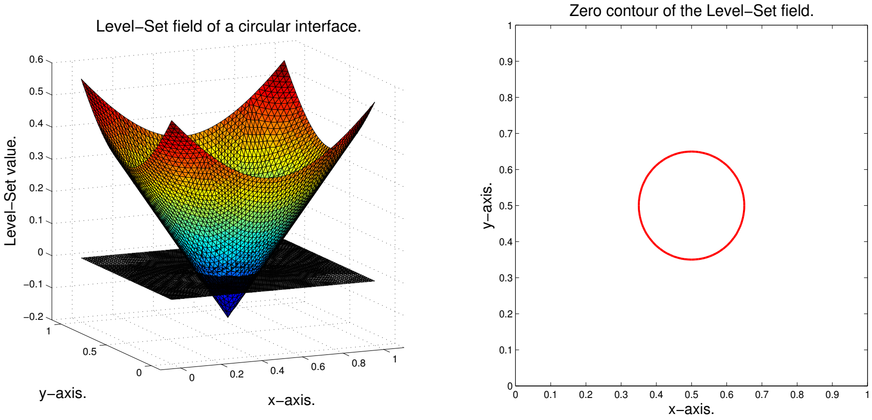


Figure 2.1: The level-set field and the corresponding zero-level contour of the interface.

$$M(t) = \int_{\Omega} H(\Phi(\boldsymbol{x},t))d\boldsymbol{x} = |S|, \tag{2.1}$$

---

[1] Although the OpenFOAM documentation calls the implemented approach to two-phase flow a VoF method, it actually is a (modified) LS method.

where $H(\boldsymbol{x})$ is the Heaviside function.

Due to this implicit definition of the interface, complex topology changes, for example merging of multiple or breaking up of a single interface are accommodated for. If the interface is a smooth curve, the smoothness of the LS function in the vicinity of the zero contour-line allows for a straightforward computation of geometrical quantities of the interface by evaluating derivatives of $\Phi(\boldsymbol{x},t)$ at the interface:

$$\hat{n}_\alpha(\boldsymbol{x},t) = \frac{\Phi_{,\alpha}(\boldsymbol{x},t)}{\sqrt{\Phi_{,\alpha}(\boldsymbol{x},t)\Phi_{,\alpha}(\boldsymbol{x},t)}},\ \boldsymbol{x} \in \boldsymbol{X}(t), \quad \kappa(\boldsymbol{x},t) = \hat{n}_{\alpha,\alpha}(\boldsymbol{x},t),\ \boldsymbol{x} \in \boldsymbol{X}(t). \tag{2.2}$$

where, $\hat{n}_\alpha$ is the unit normal vector (pointing outward from the domain of interest) and κ the curvature of a contour line of $\Phi(\boldsymbol{x},t) = 0$. Because the interface is by definition a contour line of the LS function, the following equation holds *at the interface*:

$$\frac{d}{dt}\Phi(\boldsymbol{x},t) = 0 \Rightarrow \frac{\partial\Phi(\boldsymbol{x})}{\partial t} + u_\alpha\Phi(\boldsymbol{x})_{,\alpha} = 0, \boldsymbol{x} \in \boldsymbol{X}(t), \tag{2.3}$$

where u_α is the velocity of the interface. Continuity conditions for mass and momentum dictate the velocity field is continuous across the interface and therefore uniquely defined *at* the interface. The interface is advected by the local flow.

In the LS method equation (2.3) is postulated to hold for all $\boldsymbol{x} \in \Omega$, but other choices are possible, as long as they are consistent with (2.3) and lead to a function that is at least C^2 continuous in the vicinity of the interface to allow computation of the curvature. Following Osher [22, 40] the velocity $u_\alpha(\boldsymbol{x}), \boldsymbol{x} \in \Omega \setminus \boldsymbol{X}(t)$ can be chosen such that $|\boldsymbol{\nabla}\Phi(\boldsymbol{x})|$ remains as close as possible to unity, i.e. the LS function retains its property of being a (signed) distance function while being advected. Alternatively, the signed-distance property has to be re-established explicitly by a process called *reinitialization*. For an overview of different reinitialization strategies and their merits, the reader is referred to [22, 35, 40, 55].

Reinitialization or retaining the signed-distance property in a more general sense is important when the interface model is coupled to the flow equations. Regularization filters for the viscosity and the commonly used Continuous Surface Model of Brackbill [9] for surface tension both use with the LS field to determine the distance from a grid point to the interface. However, because the coupling to the flow equations will not be considered in this research, reinitialization algorithms will not be considered.

For a solenodial velocity field u_α the extension of (2.3) is given by

$$\frac{\partial\Phi(\boldsymbol{x})}{\partial t} + \left(u_\alpha\Phi(\boldsymbol{x})\right)_{,\alpha} = 0,\ \boldsymbol{x} \in \Omega. \tag{2.4}$$

This equation shows that the LS function is conserved. However, there is no reason why a conservative redistribution of the LS function maintains the area enclosed by the zero contour line, i.e. the interface. This is one of the drawbacks associated with the LS method. Numerical experiments indicate that to some extent this can be remedied by using higher order approximations of the spatial differential operator, e.g. using ENO or WENO schemes in the context of a finite volume discretisation [26, 57], combined with higher order time-integration methods or by applying adaptive grid refinement near the interface. Such a high order solution of the LS field is already required if the interface curvature is to be extracted from this field. In [Ref] it is shown the LS solution has to be discretized with fourth order accuracy, if the interface curvature is to be recovered with second order accuracy. Although the latter is easily achieved on Cartesian meshes, this is not the case for unstructured grids of triangular or tetrahedral control volumes, because it is not straightforward to combine information from neighboring control volumes to obtain an approximation of the fluxes of sufficient accuracy.

An approximation to the solution of (nearly) arbitrary order of accuracy can be obtained using the discontinuous Galerkin finite element method. The key property of this discretisation is the fact it is a completely *local* discretisation method, and the only information that has to be exchanged between neighboring control volumes is the solution at the inter-element interfaces. This makes the discretisation method straightforward to apply on unstructured triangular meshes as opposed to higher order finite volume discretisation methods.

2.4 Discontinuous Galerkin discretisation of the Level-Set equation

The discontinuous Galerkin discretisation combines the advantages of finite volume and finite element discretisation techniques. The solution is expanded in a polynomial basis in each element. On each interface between two elements the flux is uniquely defined (obviously, to have conservation) but the solution is not. This implies the interface is only piecewise (elementwise) continuous together with the curvature and the interface normal vector. Because the coupling of the interface model to the flow equations is outside the scope of this thesis, the challenges presented by the handling of the piecewise continuous curvature will not be considered and are left for future research. The proposed discretisation for the LS equation closely follows the algorithm described in [19, 20, 23, 34].

2.4.1 Spatial discretization of the Level-Set equation

The computational domain Ω will be subdivided into a set of N_T rectilinear triangular control volumes Ω_k. This process is called *tesselation*. Using triangular control volumes is not a prerequisite for the discontinuous Galerkin method. However, if such control volumes are chosen the tesselation can be performed automatically for domains with nearly arbitrary geometry and with minimal user input.

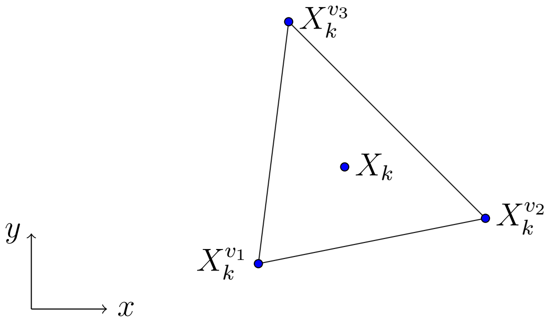


Figure 2.2: Control volume (Ω_k).

Lagrange polynomials $L_i(\boldsymbol{x})$ of degree N are used as basis functions to expand the solution in each element as:

$$\phi_k^h(\boldsymbol{x},t) = \Sigma_{i=1}^{n_p}\phi_i(t)L_i(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega_k, \tag{2.5}$$

where n_p is the total number of nodal points in each individual element. An optimal approximation of the solution to (2.4) can be found by imposing that upon substitution of (2.5) the residual is orthogonal to the polynomial space spanned by this expansion of the solution. This leads to:

$$\int_{\Omega_k}\left(\frac{\partial \phi_k^h(\boldsymbol{x},t)}{\partial t} + \nabla\cdot(\mathbf{u}(\boldsymbol{x},t)\phi_h(\boldsymbol{x},t))\right)L_i(\boldsymbol{x})d\Omega = 0, \quad 1 \le i \le n_p. \tag{2.6}$$

The weak form of (2.4) can be obtained by applying integration by parts using Gauss divergence theorem:

$$\int_{\Omega_k}\frac{\partial \phi_k^h(\boldsymbol{x},t)}{\partial t}L_i(\boldsymbol{x}) - \nabla L_i(\boldsymbol{x})\cdot(\mathbf{u}(\boldsymbol{x},t)\phi_k^h(\boldsymbol{x},t))d\Omega = -\oint_{\partial\Omega_k}\mathbf{n}\cdot(\mathbf{u}\phi_k^h)^* L_i(\boldsymbol{x})d\Omega, \tag{2.7}$$

where $\hat{\mathbf{n}}$ is the outward pointing normal and $(\mathbf{u}\phi_k^h)^*$ is the numerical flux, used to impose boundary conditions on the boundary of each element. Application of integration by parts once again leads to the *strong* [2] formulation

---

[2] Note this is a nonstandard use of the term strong formulation, that is used in the context of discontinuous Galerkin finite element methods.

of the weak form:

$$\int_{\Omega_k}\left(\frac{\partial\phi_k^h(\boldsymbol{x},t)}{\partial t}+\nabla\cdot(\mathbf{u}\phi_k^h(\boldsymbol{x},t))\right)L_i(\boldsymbol{x})d\Omega=\oint_{\partial\Omega_k}\mathbf{n}\cdot(\mathbf{u}\phi_k^h-(\mathbf{u}\phi_k^h)^*)L_i(\boldsymbol{x})d\Omega. \tag{2.8}$$

For the discretisation of (2.4) the latter form is used, but either (2.8) or (2.7) can be used as a starting point. Instead of expanding the numerical solution in a set of polynomial basis functions, the solution can be represented, directly in the value of the solution at the collocation points or nodes. Both formulations are equivalent (related by a bijective mapping by means of a Vandermonde matrix) and choosing either approach is a matter of preference. In this case the latter approach is chosen, i.e. a *nodal* discontinuous formulation is used.

One important parameter in the definition of a discontinuous Galerkin method is the choice of the numerical flux function. For the scalar transport equation (2.4) there are only few options to consider. In [20] a central approximation and a Lax-Friedrichs approximation are assessed. Numerical experiments show that oscillations will occur when a central flux approximation is used, so the use of this approximation is discarded in this research. The Lax-Friedrichs flux formulation for the approximation of the numerical flux leads to the expected monotonic solution. This formulation uses a combination of a central approximation with a correction based on the jump in the convected quantity at the face weighted by a parameter that depends on the magnitude of the velocity in the computational domain, given as:

$$(\mathbf{u}\phi_h)^*=\frac{1}{2}\left((\mathbf{u}\phi_h)^-+(\mathbf{u}\phi_h)^+\right)-\frac{c}{2}\left(\phi_h^+-\phi_h^-\right), \tag{2.9}$$

where the + sign is used to represent the flux and the solution at the edge of the element under consideration and the − sign is used to represents the flux and the solution at the edge of the neighboring element. The parameter c is used to weigh the jump across the edge in the formulation of the flux. This parameter is taken equal to the infinity norm of the velocity in the domain. Naturally, the fluxes at the edges that coincide with $\partial\Omega$ are determined by the boundary conditions.

With the numerical flux approximation defined in Equation (2.8) can now be cast in a linear system of equations. Imposing orthogonality for all individual basis functions leads to a system of n_p linear equations for the solution at the n_p nodal points:

$$M^k\frac{\partial\Phi}{\partial t}+S^k\cdot(\mathbf{u}\Phi)=F^k\left(\mathbf{n}\cdot(\mathbf{u}\Phi-(\mathbf{u}\Phi)^*)\right), \tag{2.10}$$

where $\Phi = \{\phi_1(t), \phi_2(t), ..., \phi_{n_p}(t)\}$ and $\boldsymbol{u} = \{\boldsymbol{u}_1(t), \boldsymbol{u}_2(t), ., \boldsymbol{u}_{n_p}(t)\}$ are the vectors of nodal values of $\phi_k^h(\boldsymbol{x}, t)$ and $\boldsymbol{u}(\boldsymbol{x}, t)$ in element Ω_k at time t respectively, M^k is the *mass* matrix, S^k is the *stiffness* matrix, and F^k is the boundary operator of the element. The latter are defined as:

$$M_{ij}^k = \int_{\Omega_k} L_j(\boldsymbol{x}) L_i(\boldsymbol{x}) d\Omega, \qquad S_{ij}^k = \int_{\Omega_k} \nabla L_j(\boldsymbol{x}) L_i(\boldsymbol{x}) d\Omega,$$

$$F_{ij}^k = \oint_{\partial\Omega_k} L_j(\boldsymbol{x}) L_i(\boldsymbol{x}) \partial\Omega.$$

Equation (2.10) is formulated for each element, but cannot be solved without considering the solution in the neighboring elements, because of the coupling by the fluxes at the common boundary defined through the boundary operator F^k. All systems of dimension $n_p \times n_p$ can be assembled into one linear system of ordinary differential equations that can be symbolically represented as:

$$A \frac{d\overline{\Phi}(t)}{dt} + B\overline{\Phi}(t) = \boldsymbol{g}(t), \tag{2.11}$$

where $\overline{\Phi} = \{\phi_1(t), \phi_2(t), ..., \phi_{n_p * N_T}(t)\}$ and $\boldsymbol{g}(t)$ represents the contributions of inhomogeneous boundary conditions. The approach to reduce a partial differential equation to a system of ordinary differential equations is commonly referred to as the *method-of-lines*, in contrast to methods that simultaneously discretize in space and time. It is important to mentioned that in our research we have focused more towards making the level-set field mass-conservative and we have adopted standard discretization practices for the hyperbolic equation using DG method. Therefore, no new strategies for the solution of the pure LS equation is given. Therefore, for further details regarding the discretization of the pure LS equation based on DG method, see [13, 16, 20, 23, 34].

2.4.2 Temporal Discretisation

To advance (2.11) in time an explicit Runge-Kutta method is used. Contrary to a finite volume discretisation, the fact that the mass matrix is nondiagonal requires that even for an *explicit* time-stepping method a linear system has to be solved in each time step. In the current project a direct solver has been used, because only problems with a very limited number of degrees of freedom are considered. For larger problems the specific properties of the matrix A in (2.11) can be exploited to formulate a very efficient iterative solution method [6, 7].

$$\frac{\overline{\Phi}(t)}{dt} = A^{-1}\left(\boldsymbol{g}(t) - B\overline{\Phi}(t)\right) \equiv L_h(\overline{\Phi}, t). \tag{2.12}$$

The utilized Runge-Kutta method is a five-stage method with fourth order accuracy, proposed in [10, 59]. This type of method is known as a *Low Storage Explicit Runge-Kutta* (LSERK). One important property is the fact that each stage can be computed from the previous stage only, requiring only storage for a single stage vector. The range of the stability region is comparable to other fourth order methods [20]. The algorithm is given by:

$$\begin{aligned}
&\mathbf{p}^0 = \overline{\boldsymbol{\Phi}}(t^n)\\
&i \in [1, ..., 5] : \begin{cases} \mathbf{k}^{(i)} = a_i \mathbf{k}^{(i-1)} + \Delta t L_h(\mathbf{p}^{i-1}, t + c_i \Delta t), \\ \mathbf{p}^i = \mathbf{p}^{i-1} + b_i \mathbf{k}^{(i)}, \end{cases}\\
&\overline{\boldsymbol{\Phi}}(t^{n+1}) = \mathbf{p}^5,
\end{aligned} \tag{2.13}$$

where $\mathbf{p}^i$, $i = 1..5$ are the stage vectors, and the coefficients a_i, b_i and c_i are given in Table 2.1.

i	a_i	b_i	c_i
1	0	$\frac{1432997174477}{9575080441755}$	0
2	$-\frac{567301805773}{1357537059087}$	$\frac{5161836677717}{13612068292357}$	$\frac{1432997174477}{9575080441755}$
3	$-\frac{2404267990393}{2016746695238}$	$\frac{1720146321549}{2090206949498}$	$\frac{2526269341429}{6820363962896}$
4	$-\frac{3550918686646}{2091501179385}$	$\frac{3134564353537}{4481467310338}$	$\frac{2006345519317}{3224310063776}$
5	$-\frac{1275806237668}{842570457699}$	$\frac{2277821191437}{14882151754819}$	$\frac{2802321613138}{2924317926251}$

Table 2.1: Coefficients of Low-Storage five-stage fourth-order ERK method.

2.5 Test Cases

The key characteristics of the method that need to be assessed are the mass conservation property and the accuracy of the LS field. To accomplish this, three test cases are selected based on the properties of the imposed velocity field; constant, linear or nonlinear functions of the spatial coordinates. Including time-dependence of the velocity field is not necessary to assess the accuracy of the discretisation. The test cases chosen are:

- (Solid body) translation of a circular interface with a constant velocity field,
- (solid body) rotation of a circular interface,
- the *reversed vortex test case* [34, 56, 60].

In all cases the problem is nondimensionalized by introducing a reference length and reference velocity $L = 1\ m$ and $T = 1\ s$, respectively.

The expansion of the solution can be done for polynomial basis functions of a user defined degree, without any difficulty. However, the number of nodal points increases with the degree of the basis functions and hence the number of degrees of freedom of the problem

In chapter 4 the extension of the MCLS to a discretisation on a set of unstructured triangular control volumes is presented. The discontinuous Galerkin discretisation of the LS equation is an integral part of this algorithm. However, in that application the basis functions are chosen as polynomials of degree one, in accordance with the assumption of a piecewise linear interface. For this reason, results are shown for an expansion in polynomials of order one for all test cases for three different mesh sizes. These results will be used to assess the improvement in the solution that can be accomplished by imposing the mass conserving correction of the MCLS on a pure LS solution. This is also the reason for choosing separate test cases with a linear and a nonlinear velocity field. For the proposed discretisation of the LS field the behavior of the solution will not be influenced by nonlinearity of the velocity field. However, for the MCLS method the nonlinearity does make a difference and it is worthwhile to consider these cases separately.

To determine the effect of the degree of the polynomial basis on the mass conservation properties of an uncorrected LS solution the mass loss is computed for three mesh sizes and using a degree one, two and three polynomial expansion for the reversed vortex test case. The mass M can be computed up to machine precision when degree one polynomials are used, by a geometrical reconstruction of the domain of interest. However, for the degree two and three polynomials it is approximated by a geometrically computed area (mass) enclosed by the fluid of interest. The details of the geometrical method is given in appendix A.

2.5.1 Linear translation of a circular interface

In this test case a circular fluid region of radius $R = 0.15$ is considered in a domain $\Omega_D = [0,1] \times [0,1]$. Initially, the centre of the circle is located at $\boldsymbol{x}^c(0) = (0.5, 0.2)^T$. The initial condition for the LS field is defined as:

$$\phi(\boldsymbol{x}, 0) = |\boldsymbol{x} - \boldsymbol{x^c(0)}| - R. \tag{2.14}$$

The circular fluid region is advected with a constant velocity field $\boldsymbol{u} = (0, 0.1)^T$ using a time step $\Delta t = 0.01$. The final time is $T = 5$. Because the velocity field is solenoidal, mass should be conserved up to machine precision. During the advection the mass is computed at every time step $(M(t))$ and compared with its initial value $(M(0))$. At the final time the LS field is compared with the exact LS, as shown in Figure 2.4 for three different mesh sizes. The exact LS field at the final time is given by:

$$\phi(\boldsymbol{x}, 5) = |\boldsymbol{x} - \boldsymbol{x^c(T)}| - R, \quad \boldsymbol{x^c(T)} = (0.5, 0.7)^T. \tag{2.15}$$

Figure 2.4 presents the ratio of $M(t)/M(0)$ for three different mesh sizes.

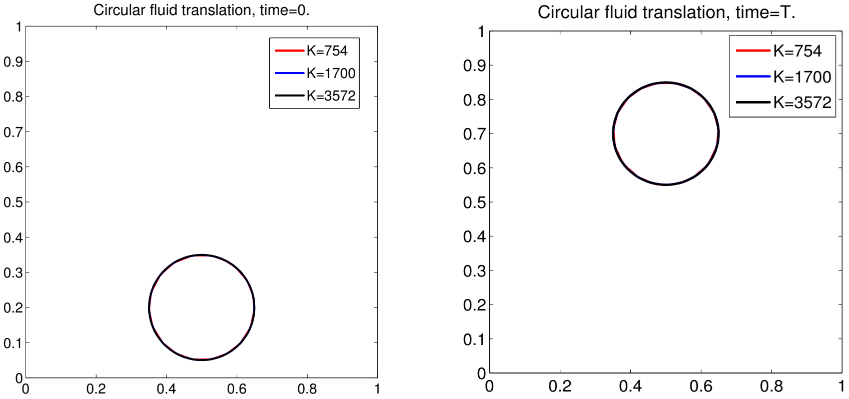


Figure 2.3: The interface $\phi(\boldsymbol{x}, t) = 0$ for the linear translation test case at time levels $t = 0$ (left) and $t = T$ (right).

The graph shows mass is subsequently gained and lost during the advection.

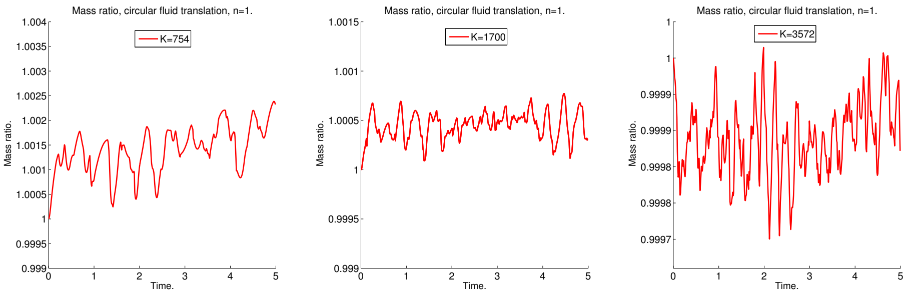


Figure 2.4: Evolution of $M(t)/M(0)$ for the translation of a LS representation of a circular interface.

In Table 2.2 the average mass error $\overline{|M(t)-M(0)|}$ and the L_2 error in the solution at the final time $\|\phi(\boldsymbol{x},T)-\phi_k^h(\boldsymbol{x},T)\|_{L_2}$ are presented for three different mesh sizes, together with the order of convergence estimate of the latter quantity. The results show the expected order of convergence for a discontinuous Galerkin solution using an expansion of the solution in linear polynomials on a general unstructured mesh. However, the loss of mass is quite severe and would be unacceptable for long time simulations.

No. of elements (K)	$\overline{\|M(t)-M(0)\|}$	$\\|\phi(\boldsymbol{x},T)-\phi^h(\boldsymbol{x},T)\\|_{L_2}$	Order
754	9.6836e-05	1.0052e-03	—
1700	3.1375e-05	5.0686e-04	1.981
3572	7.8133e-06	2.4369e-04	2.079

Table 2.2: Average mass error and L_2 error of the LS field for the translation of a circular interface.

2.5.2 Rotation of a circular interface

The second test case is related to the rotation of circular region in a domain $\Omega_D = [0,1]\times[0,1]$. Initially, the centre of the circle is located at $\boldsymbol{x}^c(0) = (0.5, 0.75)^T$. The initial condition for the LS field is defined as:

$$\phi(\boldsymbol{x},0) = |\boldsymbol{x}-\boldsymbol{x^c(0)}| - R. \tag{2.16}$$

The circular fluid region is advected with a linear velocity field $\boldsymbol{u} = (x_1 - 0.5, -x_2+0.5)^T$ using a time step $\Delta t = 0.01$. The final time is $T = 2\pi$. This initial setup is shown in Figure 2.5 . Because the velocity field is solenoidal, mass should be conserved up to machine precision and the velocity field will return the interface to its initial position at $t = T$. Hence, $\phi(\boldsymbol{x}, T) = \phi(\boldsymbol{x}, 0)$ as shown in Figure 2.5. In Figure 2.6 the ratio of $M(t)/M(0)$ is

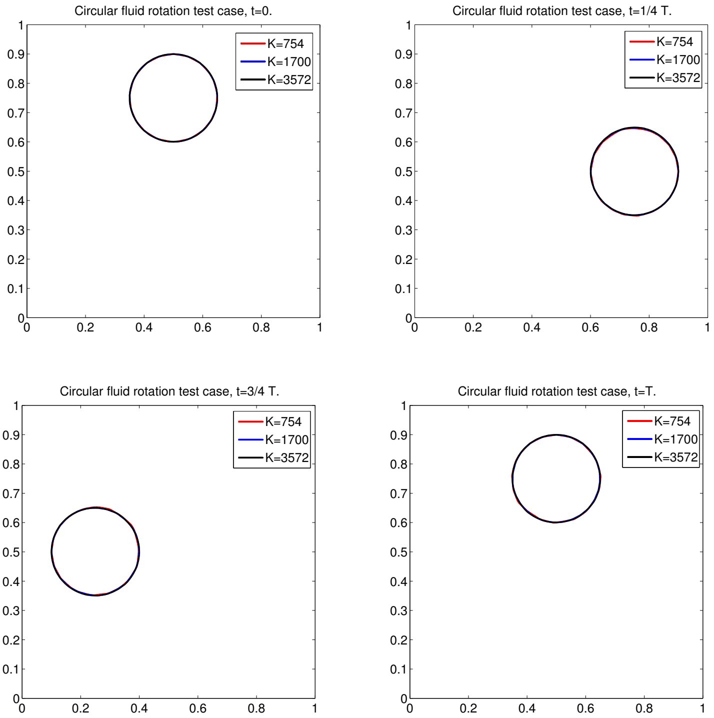


Figure 2.5: The interface $\phi(\boldsymbol{x}, t) = 0$ for the rotation test case at time levels $t = 0$, $t = \frac{1}{4}T$, $t = \frac{3}{4}T$ and $t = T$.

shown for three different mesh sizes. The graph shows mass is subsequently gained and lost during the rotation. In Table 2.3 the average mass error $\overline{|M(t) - M(0)|}$ and the L_2 error in the solution at the final time$\|\phi(\boldsymbol{x}, T) - \phi^h(\boldsymbol{x}, T)\|_{L_2}$ are presented for three different mesh sizes, together with the order of convergence estimate of the latter quantity.

2.5.3 The reversed vortex test case

The *reversed vortex* or *single reversed vortex* test case is one of the more challenging tests for models of multiphase flow. The reason for this is the severe stretching and deformation of the convected interface caused by the nonlinear velocity field during the advection which makes accurate mass conservation challenging.

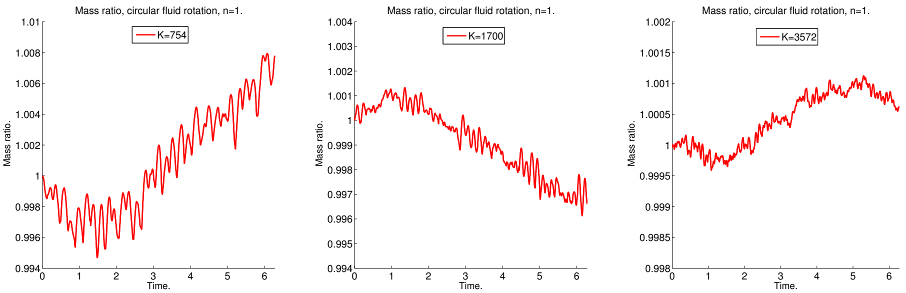


Figure 2.6: Evolution of $M(t)/M(0)$ for the rotation of a LS representation of a circular interface.

No. of elements (K)	$\overline{\|M(t) - M(0)\|}$	$\|\phi(\boldsymbol{x}, T) - \phi^h(\boldsymbol{x}, T)\|_{L_2}$	Order
754	1.0089e-04	6.9105e-04	—
1700	3.6787e-04	4.7271e-04	1.4612
3572	1.4564e-05	1.3806e-04	3.4237

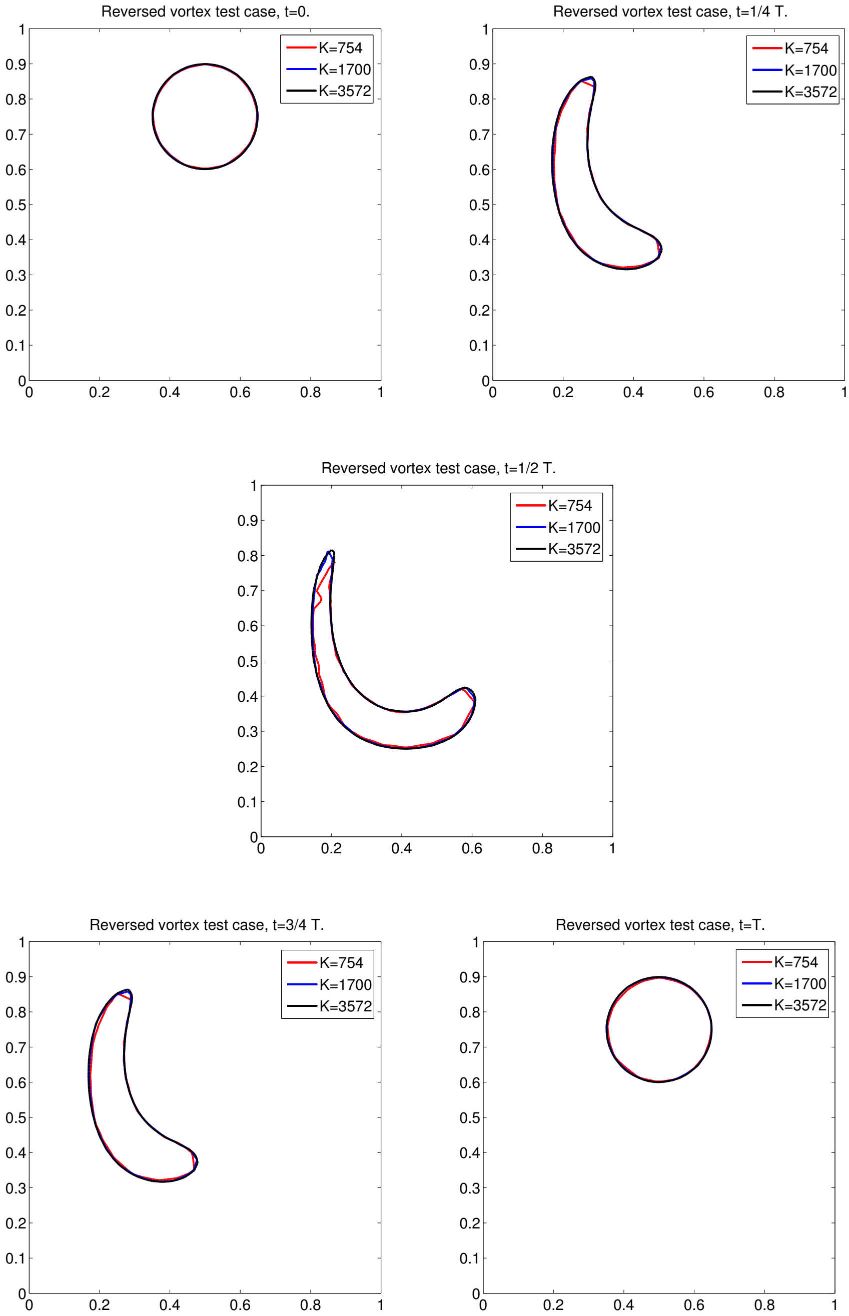
Table 2.3: Average mass error and L_2 error of the LS field for the rotation of a circular interface.

Domain Ω and initial condition are identical as for the linear translation test case. This circular region is advected with a divergence free nonlinear velocity field given by:

$$\boldsymbol{u} = \cos(\pi t/T)(\sin^2(2\pi x_1)\sin(2\pi x_2), -\sin^2(2\pi x_2)\sin(2\pi x_2))^T. \quad (2.17)$$

The Leveque cosine will cause the velocity field to reverse direction at time $t = T/2$. This implies the interface will return to its original position at time T. In this test case $T = 2$ is used. The initial setup is shown in Figure 2.7. Like in the other test cases the mass (area) enclosed by the interface is the quantity of interest. This is computed after every time step and compared with the initial quantity to give the amount of mass loss or gain during advection. Also, at the final time the corrected LS field is compared with the initial condition as the circular interface should return to its initial position and shape at time $t = T$.

In Figure 2.7 the interface is shown at different time levels for three different mesh sizes.



Reversed vortex test case, t=0.

Reversed vortex test case, t=1/4 T.

Reversed vortex test case, t=1/2 T.

Reversed vortex test case, t=3/4 T.

Reversed vortex test case, t=T.

Figure 2.7: The interface $\phi(\boldsymbol{x}, t) = 0$ for the reversed vortex test case at time levels $t = 0,\ t = \frac{1}{4}T,\ t = \frac{1}{2}T,\ t = \frac{3}{4}T$ and $t = T$.

In Figure 2.8 the ratio of $M(t)/M(0)$ is shown for three different mesh sizes. The behavior is very similar to what is observed in the simple linear translation and rotation test cases.

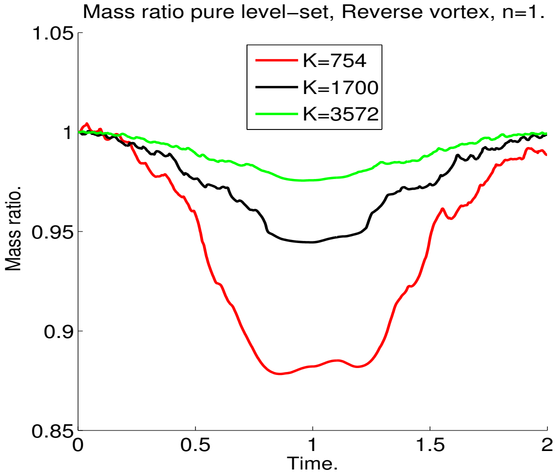


Figure 2.8: Evolution of $M(t)/M(0)$ for the reversed vortex test case using a LS representation of a circular interface.

In Table 2.4 the average mass error $\overline{|M(t) - M(0)|}$ and the L_2 error of the LS field at the final time $\|\phi(\boldsymbol{x}, T) - \phi^h(\boldsymbol{x}, T)\|_{L_2}$ are presented for three different mesh sizes.

No. of elements (K)	$\overline{\lvert M(t) - M(0)\rvert}$	$\|\phi(\boldsymbol{x}, T) - \phi^h(\boldsymbol{x}, T)\|_{L_2}$	Order
754	4.09071e-03	1.26493e-03	—
1700	1.76298e-03	4.88629e-04	2.5887
3572	7.85191e-04	1.98911e-04	2.4565

Table 2.4: Average mass error and L_2 error of the LS field for the reversed vortex test case.

2.6 The influence of the degree of the polynomial representation on mass conservation

Naturally, the accuracy of the solution will improve upon refinement of the mesh and raising the degree of the polynomial expansion. The results of the three test cases show the order of convergence corresponds to the theoretical order of convergence on general unstructured grids. To what extend the degree of the polynomial expansion influences the mass conservation properties is not obvious and will be assessed experimentally. For polynomial expansions of degree larger than one it becomes non-trivial to assess the order of convergence of the LS solution in the vicinity of the interface. The L_2 error in the field will be dominated by the error in the vicinity of the apex of the LS field, where the gradient is not defined. Therefore, only the mass error will be compared. Table 2.5 shows the averaged mass error $\overline{|M(t) - M(0)|}$ for the reversed vortex test case for a polynomial expansion using degree 2, at three different mesh sizes. In Figure 2.9 mass ratio is presented for the same both degree throughout the evolution of the interface. The strong fluctuations in $M(t)$ show there is no (clear) convergence in $M(t)$ upon raising the degree of the polynomial expansion or on reduction of the mesh width.

Characteristic mesh width h	$\overline{\lvert M(t) - M(0)\rvert}$
3.64e-02	9.37467e-05
2.43e-02	4.15348e-05
1.67e-02	1.97897e-05

Table 2.5: Mass error for expansion in degree n=2 polynomial for the reversed vortex test case.

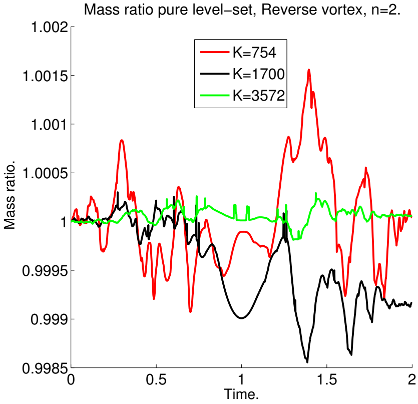


Figure 2.9: Evolution of $M(t)/M(0)$ for the reversed vortex test case using polynomial expansion of degree n=2.

2.7 Conclusions

A baseline method for the interface capturing has been presented based on a LS representation of the interface. The method is applicable on general unstructured triangular meshes. Numerical experiments confirm that the approximation of the LS field converges with the expected order. However, the solution does not conserve mass accurately. The area enclosed by the interface fluctuates strongly and this does not appear upon grid refinement or upon raising the degree of the polynomial expansion. The method discussed in this chapter acts as a baseline for the improvements that are proposed in later chapters: imposing a mass-conserving correction to the interface or switching to a different formulation of the function $\phi(\boldsymbol{x}, t)$.

CHAPTER 3

An overview of the LS and VoF hybrid methods

3.1 Introduction

It is highlighted in the last chapter that apart from the geometrical flexibility provided by the LS for the multiphase flow simulation, mass-conservation can not be achieved from the pure LS field. Which is one of the key problems associated with the LS method. To mitigate this problem as mentioned earlier different solution procedures has been developed and are still developing. One of the popular choices is hybrid method, in which the LS method is combined with another simultaneously advected quantity that used is to make the LS field mass conserving while keeping geometric flexibility provided by the pure LS field. The solution procedure varies from the coupling of LS field with particle approach [17] to LS coupling with VoF field [49]. In general, the coupling is performed to correct the LS field such that it corresponds to the correct quantity of fluid of interest. Usually, this is achieved by using a PLIC method in which the interface is represented by a line segment and then it is adjusted in a control volume such the it corresponds to the exact quantity of the fluid of interest. This correction can be an iterative based or direct i.e. that is without involving any iterative process [43]. It is worthwhile to have an overview of the existing hybrid LS and VoF method for multiphase flow simulation in order to have a better understanding of the proposed method of this research.

3.2 Hybrid Methods

There exists a number of hybrid methods based on the application under consideration. In the following, we have selected those methods that provide the guideline to develop another hybrid method.

3.2.1 CLSVOF method

The Coupled Level-Set Volume of Fluid (CLSVOF) method is proposed by Sussman and Puckett [48,49], in 2000. In this method, the interface is constructed using the LS field while mass is conserved by using simultaneously advected volume of fluid, VoF PLIC method. PLIC is the extended version of SLIC method in which the interface is assumed to be inclined with coordinates axes in a control volume, which is not a right assumption. In PLIC method, the interface is represented by a line segment and can be oriented in any direction. The area enclosed in it represents the quantity of the fluid of interest. In CLSVOF method, there exists no straightforward relation between the LS field and the VoF field [52, 53]. The coupling between the LS and VoF field is done using a PLIC approach which not trivial and leads to an iterative process and also the coupling becomes cumbersome when the grid is an unstructured Cartesian grid. It is also reported that mass conservation of the CLSVOF method is comparable to VoF method [49,52]. In spite, these limitations this method provides a solid guideline towards the coupling of the LS field and the VoF field.

3.2.2 Hybrid particle LS method

A hybrid particle Level-Set method is proposed by Enright et al. [17], in 2002. In this method, the mass conservation property of the LS field can be enhanced by passively advected particles of suitable radii with flow field. These particles are placed near the interface and their region of placement is marked by the sign of the LS field. This idea makes this method different from the traditional marker and particles approach in which particles are placed in a whole domain [21, 45]. The passively advected particles are used to reconstructed the interface in the region of sharp stretching and tearing of the interface. This approach seems suitable but the Lagrangian nature of the approach makes the algorithm computationally involved, as these particles need tracking. These particles are used to track both the characteristics of the flow and also to reconstruct the interface in the regions where the LS method alone failed to accurately conserve mass. In this approach due to the passive nature of the particles they are allowed to overlap with each other, this makes algorithm computationally involve towards interface reconstruction for the mass conservation.

3.2.3 MCLS method

The Mass-Conserving Level-Set (MCLS) method is presented by Pijl et. al [52, 53], in 2005. This method is based on the coupling between LS and VoF field by simultaneously advecting the LS and the VoF field. In this approach, the VoF field is only used for the mass conservation correction to the LS field and it is not involved in the interface reconstruction apart from their counter methods [48, 49]. To achieve this goal i.e. to apply conservation without interface reconstruction of the interface from VoF field an explicit relation between LS and VoF field is obtained in the form of an analytic function based on the assumption that the interface can be represented by a piecewise line segment, which is very common practice in coupled methods. This analytic function is called the volume of fluid function. This idea is the key ingredient of the MCLS method. This method is extendable to 3D. However, there is a limitation to this method as it is designed for Cartesian grids, so complicated geometries can not be handled at this moment. The rectification of this issue in the MCLS method is one of the important research questions that has been answered in this thesis.

3.2.4 VOSET method

A coupled Volume of Fluid and Level-Set method (VOSET) of Sun and Tao [46] is presented in 2010. This method works in different approach as compared to above mentioned hybrid methods. In this method, the LS field is reconstructed from the VoF field. This methodology is contrary to the existing LS and VoF coupled methods. It is important to incorporate this methodology as well for a fair comparison. As the LS field is reconstructed from the VoF field the only quantity that is advected with the flow field is the VoF field. This is an important difference with existing hybrid method, as this certainly computationally favourable. However, like VoF PLIC the interface needs to be reconstructed and then using contour 0.5 value of the VoF fraction as reference interface the LS field is reconstructed by the geometrical iterative process in which distances are computed from the nearest points in order to get continues definition of the LS field. In general, this method seems simple as only one quantity needs to be advected. However, the interface reconstruction needs to be done twice once for the VoF field and then from VoF field to LS field needs to be reconstructed.

3.3 Conclusions

In this chapter, a broader overview of the existing hybrid methods that form the foundation to develop other methods for the simulation of multi-phase flow. These methods have their pros and cons. For example CLSVOF method seems plausible but the interface reconstruction based on the PLIC approach makes that method computationally involved and also the extension to 3D is not so trivial. The flux approximation by using the flux splitting approach makes this method not suitable for non-Cartesian grids. In other words, to make this method robust extra information along with the interface reconstruction needs to be done. In the hybrid particle method tracking mass-less particle and the reconstructing the interface to make sure the LS is mass conserving is also computational involve and it increases more with 3D mesh and unstructured non-Cartesian grids. However, this method is suitable for the flow that involves large sharing and deformations. The VOSET method seems plausible as far as number quantity needs to be advected, However, the interface reconstruction twice made in cumbersome for 3D unstructured meshes as well and also the definition of VoF interface by a 0.5 contours of the VoF field is arbitrary.

If we analysis the MCLS method this method seems plausible in all respect as the LS and the VoF field is needed to be advected independently, this step is similar to CLSVOF method, however, no need to reconstruct the interface, this makes MCLS method suitable for 3D control volumes, but it is designed for Cartesian grid only. Keeping our objective in mind that is we need to formulate numerical methods for the multi-phase flow simulation in geometrically complicated domains, we have decided to choose the MCLS method as a guide to developed method. In order to achieve this goal, we have developed VoF function and inverse function for unstructured 2D triangular and 3D tetrahedron grids. These are the key components of this thesis.

CHAPTER 4

MCLS method for unstructured triangular meshes

4.1 Introduction

In this chapter [1] proposed Mass-Conserving Level-Set method is presented for the 2D unstructured triangular grid.

4.2 Modeling the evolution of the interface

The MCLS algorithm is used to model the dynamics of two-phase immiscible incompressible flow. It uses an 'hybrid' description of the distribution of the two-phases, $Phase_0$ and $Phase_1$, using both the LS and the VoF field.

4.2.1 The Level-Set field

The interface between the two phases $\boldsymbol{X}(t)$ is defined as an isoline of a C^2 function $l(\boldsymbol{x},t) : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^2$, and simply connected. Choosing $l(\boldsymbol{x},t)$ as the signed distance to the interface is one of the common choices.

---

[1] This chapter is based on the article:
F. Raees, D.R. van der Heul and C. Vuik, A mass-conserving level-set method for simulation of multiphase flow in geometrically complicated domains, International Journal for Numerical Methods in Fluids, 2015.

Consequently, this choice defines the interface as the zero-level contour of the signed distance function $\Phi(\boldsymbol{x},t)$. A signed distance function is a natural choice because it allows for accurate computation of the interface curvature and normal vector. However, Olssson et al. propose a LS function that resembles a mollified Heaviside function to improve the mass conservation properties [14, 36, 38]. In the MCLS the LS function is chosen as a signed distance function $\Phi(\boldsymbol{x},t)$. If $\Phi(\boldsymbol{x},t) < 0$, $Phase_0$ is present, while $Phase_1$ is present when $\Phi(\boldsymbol{x},t) > 0$. Due to this implicit definition of the interface, complex topology changes, for example merging of multiple or breaking up of a single interface are accommodated for. The smoothness of the LS function in the vicinity of the zero contour-line allows for a straightforward computation of geometrical quantities of the interface by evaluating derivatives of $\Phi(\boldsymbol{x},t)$ at the interface:

$$\hat{n}_\alpha(\boldsymbol{x},t) = \frac{\Phi_{,\alpha}(\boldsymbol{x},t)}{\sqrt{\Phi_{,\alpha}(\boldsymbol{x},t)\Phi_{,\alpha}(\boldsymbol{x},t)}},\ \boldsymbol{x} \in \boldsymbol{X}(t), \quad \kappa(\boldsymbol{x},t) = \hat{n}_{\alpha,\alpha}(\boldsymbol{x},t),\ \boldsymbol{x} \in \boldsymbol{X}(t), \tag{4.1}$$

where $\hat{n}_\alpha$ is the unit normal vector and κ the curvature of a contour line of $\Phi(\boldsymbol{x},t) = 0$.

4.2.2 Volume of Fluid field

The distribution of the fluids is represented by a C^{-1} color function $c(\boldsymbol{x},t) : \mathbb{R} \to \{0,1\}$: If $c(\boldsymbol{x},t) = 0$, $Phase_0$ is present, while $Phase_1$ is present when $c(\boldsymbol{x},t) = 1$. Based on the color function $c(\boldsymbol{x},t)$ we can define the Volume of Fluid *grid* function $\Psi_k : G \to [0,1]$ as:

$$\Psi_k(t) = \frac{1}{|\Omega_k|}\int_{\Omega_k} c(\boldsymbol{x},t)d\Omega, \tag{4.2}$$

where G is the set of all centers of the control volumes Ω_k and $|\Omega_k|$ their area. The fact that $c(\boldsymbol{x},t)$ is C^{-1} continuous and $\Psi_k(t)$ only discretely defined, makes the computation of the interface curvature challenging and computationally intensive.

4.2.3 Hybrid formulation of the MCLS method

In the hybrid formulation the flow is described by both a LS and a VoF field that are in agreement in each cell in the following way:

$$\Psi_k(t) = \frac{1}{|\Omega_k|}\int_{\Omega_k} H(\Phi(\boldsymbol{x},t))d\Omega, \tag{4.3}$$

where $H(q)$ is the Heaviside function. This means that the LS field has the mass-conserving properties of the VoF field, while at the same time the interface location is exactly defined by the LS field.

4.3 Overview of the MCLS algorithm

In this section an overview of the MCLS method is presented. The proposed algorithm relies on the key idea that a mass-conserving correction can be imposed on the LS field that is based on a simultaneously advected VoF field. The algorithm is visualised in Figure 4.1. Advancing the interface with the MCLS over a single time step consists of the following substeps, assuming the LS Φ^n and VoF field Ψ^n are known:

- The LS field and the VoF field are advected simultaneously. After this advection step, the two fields do not correspond to each other, i.e. the fields do not comply with 4.3.
- The advected LS field Φ^* is used to compute a corresponding VoF field Ψ^* by means of the VoF function.
- The difference between the advected VoF field Ψ^{n+1} and the VoF field Ψ^* is determined.
- In an element, for which the difference is smaller than the prescribed tolerance no correction to the LS field is required. If the difference exceeds the prescribed tolerance, a correction is imposed to the LS field by means of the inverse VoF function to make the LS field mass conserving up to a user specified tolerance.
- At time t^{n+1} a mass-conserving LS field Φ^{n+1} has been obtained along with the VoF field Ψ^{n+1}.

In the following sections each of the components that are required for the proposed extension of the MCLS algorithm for an unstructured triangular grid are presented. These are: the LS field and VoF field advection algorithms described in Sections 4.5 and refelvof, respectively, and the VoF function (Section 4.4) and its inverse (Section 4.7).

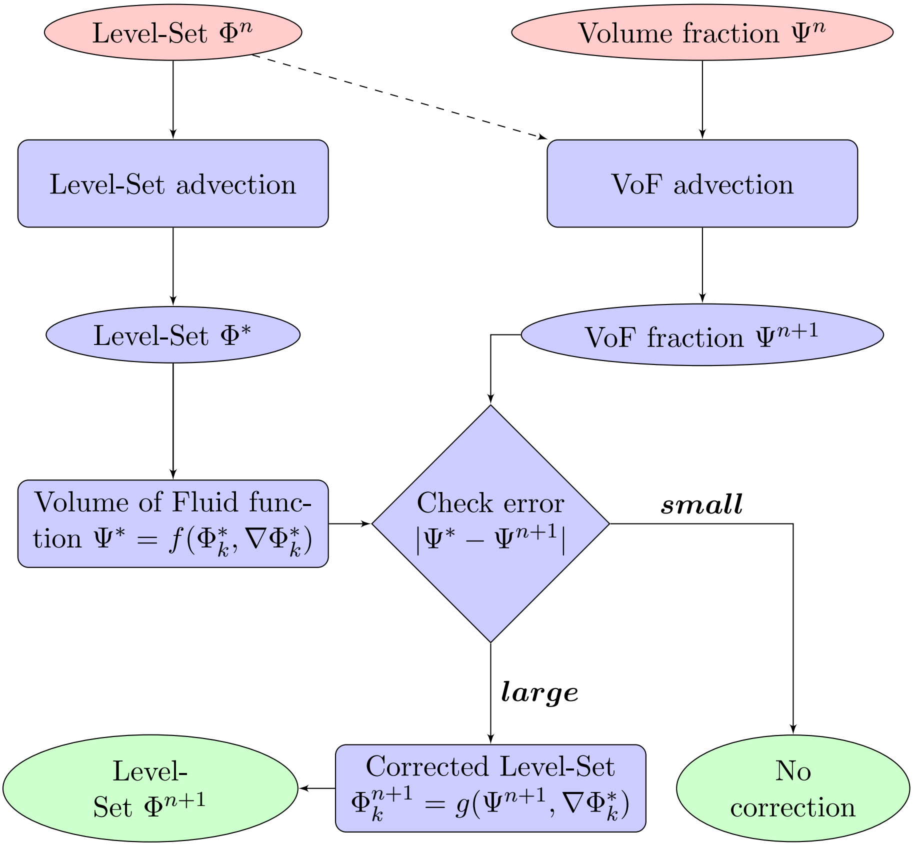
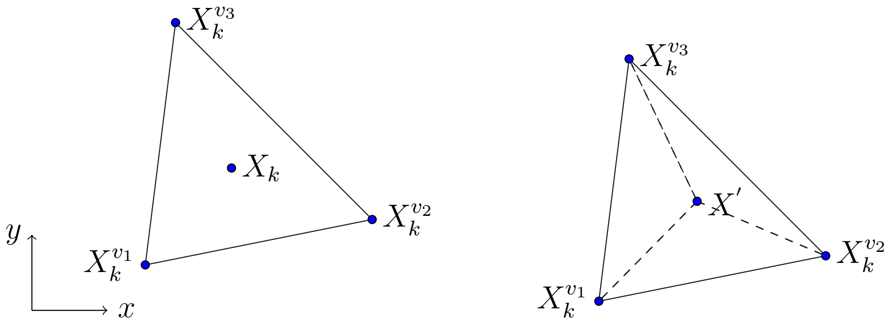

Level-Set Φ^n
Volume fraction Ψ^n
Level-Set advection
VoF advection
Level-Set Φ^*
VoF fraction Ψ^{n+1}
Volume of Fluid function $\Psi^* = f(\Phi_k^*, \nabla\Phi_k^*)$
Check error $|\Psi^* - \Psi^{n+1}|$
small
large
Level-Set Φ^{n+1}
Corrected Level-Set $\Phi_k^{n+1} = g(\Psi^{n+1}, \nabla\Phi_k^*)$
No correction

Figure 4.1: The MCLS algorithm is presented in a flow chart to show the transfer of information between the different components involved in advancing a single time step.

4.4 Efficient computation of the Volume of Fluid field from the LS field and vice versa

Consider the LS field $\Phi(\boldsymbol{x}, t) : \Omega \to R$ and the VoF field $\Psi_k : G \to [0, 1]$. Consider element Ω_k, with vertices $\boldsymbol{X}_k^{v1}$,$\boldsymbol{X}_k^{v2}$ and $\boldsymbol{X}_k^{v3}$, ordered in counter clock wise direction (Figure 4.2(a)). The centroid of the element is $\boldsymbol{X}_k$. If we choose $c(\boldsymbol{x}, t) = H(\Phi(\boldsymbol{x}, t))$ the VoF function in this element is given by 4.3. However, in a mixed cell, i.e. a cell intersected by the interface, the VoF representation based on 4.3 is not computationally efficient. Therefore,



(a) Control volume (Ω_k). (b) Barycentric coordinates.

Figure 4.2: Control volume (Ω_k), cell centroid (X_k) and arbitrary point (X').

4.3 is approximated by a VoF function f:

$$\Psi_k(t) = \frac{1}{|\Omega_k|}\int_{\Omega_k} H(\phi_k(\boldsymbol{x},t) + C_1 h^2)d\Omega = \tag{4.4}$$

$$\frac{1}{|\Omega_k|}\int_{\Omega_k} H(\phi_k(\boldsymbol{x},t))d\Omega + C_2 h^2 = f(\Phi_k, \nabla\Phi_k) + C_3 h^2, \quad C_i \in \mathbb{R},$$

where $\phi_k(\boldsymbol{x},t)$ is the linearization of the LS function ($\Phi(\boldsymbol{x},t)$, $\boldsymbol{x} \in \Omega_k$), around the cell centroid $\boldsymbol{X}_k$, defined as:

$$\phi_k(\boldsymbol{x},t) = \Phi_k(t) + \Phi_{,\alpha}\left(x_\alpha - (\boldsymbol{X}_k)_\alpha\right), \tag{4.5}$$

and where $\boldsymbol{X}_k$ is the coordinate vector of the cell centroid, $h = \sqrt{|\Omega_k|}$, $\Phi_k(t) = \Phi(\boldsymbol{X}_k, t)$ and $\Phi_{,\alpha}$ is the gradient of LS at cell centroid $\Phi_{,\alpha} = \nabla\Phi(\boldsymbol{X}_k)$. Now, the goal is to find the function $\Psi(\boldsymbol{X}_k) = f(\Phi_k, \nabla\Phi_k)$, such that it provides the VoF field in terms of the LS field.

4.4.1 Converting the LS field to the Volume of Fluid field

In order to express the VoF field in the LS field, the triangular control volumes are first mapped from Cartesian space to logical space using a barycentric coordinate transformation.

Barycentric transformation

Consider a triangular element Ω_k, as shown in Figure 4.2(b). This element can be divided into three triangles by assuming an arbitrary point

$\boldsymbol{X}' \in \Omega_k$. This subdivision provides three triangles with areas $\blacktriangle \boldsymbol{X}_k^{v1}\boldsymbol{X}'\boldsymbol{X}_k^{v3}$, $\blacktriangle \boldsymbol{X}_k^{v1}\boldsymbol{X}_k^{v2}\boldsymbol{X}'$ and $\blacktriangle \boldsymbol{X}_k^{v2}\boldsymbol{X}_k^{v3}\boldsymbol{X}'$, respectively.

Barycentric coordinates ξ_β based on these areas, can be defined as;

$$\xi_1 = \frac{\blacktriangle \boldsymbol{X}_k^{v1}\boldsymbol{X}'\boldsymbol{X}_k^{v3}}{|\Omega_k|}, \quad \xi_2 = \frac{\blacktriangle \boldsymbol{X}_k^{v1}\boldsymbol{X}_k^{v2}\boldsymbol{X}'}{|\Omega_k|}, \quad \xi_3 = \frac{\blacktriangle \boldsymbol{X}_k^{v2}\boldsymbol{X}_k^{v3}\boldsymbol{X}'}{|\Omega_k|}. \tag{4.6}$$

These barycentric coordinates define the following coordinate transformation on Ω_k:

$$x_\alpha = (\boldsymbol{X}_k^{v\beta})_\alpha \xi_\beta. \tag{4.7}$$

Naturally, for $\boldsymbol{x} \in \Omega_k$ the barycentric coordinates $\xi_\alpha \in [0, 1]$.

Using the fact that $\xi_1 + \xi_2 + \xi_3 = 1$, the dependency on one of the barycentric coordinates can be eliminated. For the moment we will not decide on which of the coordinates is selected to be eliminated. Above transformation for the possible choices for the elimination of the dependency of either of the barycentric coordinates can be written as $\boldsymbol{x} = TC_i$, Where $T \in \mathbb{R}^{2\times 3}$ defined as:

$$T = \begin{bmatrix} \boldsymbol{X}_k^{v1} & \boldsymbol{X}_k^{v2} & \boldsymbol{X}_k^{v3} \end{bmatrix},$$

and vector C_i represents the i^{th} column of $C \in \mathbb{R}^{3\times 3}$, defined as:

$$C = \begin{bmatrix} 1-\xi_2-\xi_3 & \xi_1 & \xi_1 \\ \xi_2 & 1-\xi_1-\xi_3 & \xi_2 \\ \xi_3 & \xi_3 & 1-\xi_1-\xi_2 \end{bmatrix},$$

where i is the index of the vertex of the triangular element. This information is necessary for the mapping of a specific vertex to the origin in logical space.

4.4.2 The Volume of Fluid function

Our aim is to use a geometric construction to find the function $\Psi_k = f(\Phi_k, \nabla\Phi_k)$. Contour lines of the linearized LS function $\phi_k(\boldsymbol{x})$ are straight line segments. The contour line $\phi_k(\boldsymbol{x}) = 0$ divides Ω_k in two pieces. From 4.3 and 4.4 it follows that $f(\phi_k, \nabla\phi_k)$ is the relative area of the polygon P bounded by the edges of the triangle and the line segment $\phi_k = 0$:

$$P = \{\boldsymbol{x} \in \Omega_k | \phi(\boldsymbol{x}) \leq 0\} \tag{4.8}$$

The polygon can be triangular, quadrilateral or even reduced to a single point, depending on the intersection of the interface with Ω_k. Efficient evaluation of the function $f(\Phi_k, \nabla\Phi_k)$ requires a formulation that considers all possible situations as conveniently as possible.

For nonzero $|P|$ only two configurations are possible for the intersection of the interface with Ω_k. These possibilities are defined as:

- Case-1: $\Phi(\boldsymbol{X}_k^{v_3}) < 0$ for a single vertex. In this case P is a triangular as shown in Figure 4.3.
- Case-2: In this case P is quadrilateral, which means two vertices are in the fluid of interest i.e. $\Phi(\boldsymbol{X}_k^{v_3}) < 0$ and $\Phi(\boldsymbol{X}_k^{v_2}) < 0$. This is shown in Figure 4.5.

However, it is sufficient to only consider Case-1, because in Case-2, we can use that $\Psi_k = f(\Phi_k, \nabla\Phi_k) = 1 - f(-\Phi_k, -\nabla\Phi_k)$. This means the relative area of the quadrilateral can be obtained by subtracting the relative area of the complementary part of the triangle. It is important to point out that this is not the only option to handle Case-2. It is indeed possible to define a single VoF function that can handle both cases.

Case-1: triangular domain of interest

Firstly we consider Case-1. The vertex $\boldsymbol{X}_k^{v_3}$ is selected to be mapped to the origin, as is shown in Figure 4.3. The linear transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is bijective for a non degenerate triangle Ω_k: a triangle for which no vertices coincide and/or edges are aligned. Define the (nonsingular) Jacobian of this

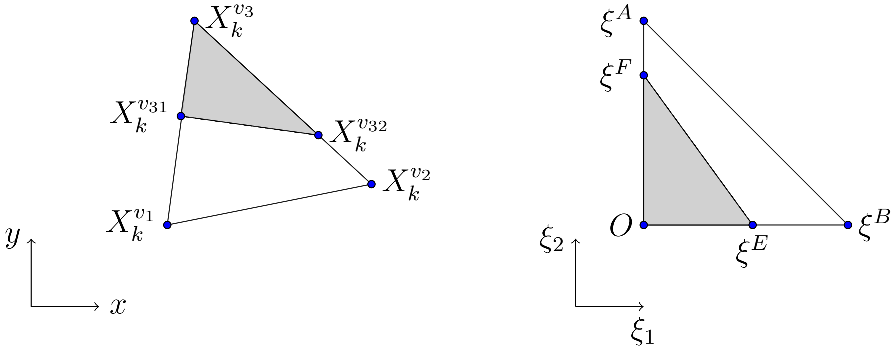


Figure 4.3: Case-1: Triangular domain of interest, where the vertex $\boldsymbol{X}_k^{v_3}$ is mapped to the origin.

mapping as $J = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}$. The linearized LS function $\tilde{\phi}(\boldsymbol{\xi})$ can now be formulated in logical space as:

$$\tilde{\phi}(\boldsymbol{\xi}, t) = \Phi(\boldsymbol{x}(\boldsymbol{\xi}_k), t) + \left.\frac{\partial \Phi}{\partial \boldsymbol{\xi}}\right|_{(\boldsymbol{\xi},t)=(\boldsymbol{\xi}_k,t)} (\boldsymbol{\xi} - \boldsymbol{\xi}_k), \quad \frac{\partial \Phi}{\partial \boldsymbol{\xi}} = J\nabla\Phi, \tag{4.9}$$

where $\boldsymbol{\xi}_k = \boldsymbol{\xi}(\boldsymbol{X}_k)$. Consider Figure 4.3, that shows the interface is intersecting the edges $\overline{\boldsymbol{X}_k^{v3}\boldsymbol{X}_k^{v1}}$ and $\overline{\boldsymbol{X}_k^{v2}\boldsymbol{X}_k^{v3}}$ at the points $\boldsymbol{X}_k^{v31}$ and $\boldsymbol{X}_k^{v32}$, respectively. The vertex $\boldsymbol{X}_k^{v3}$, that is common to the two edges intersected by the interface, is mapped to the origin while the other two vertices are mapped to points on the axes in logical space. It is important to note that vertices $\boldsymbol{X}_k^{v1}$ and $\boldsymbol{X}_k^{v2}$ can be mapped to any of the vertices in logical space, except for the origin. The other two vertices in logical space are located at $(1,0)^T$ and $(0,1)^T$. Furthermore, the interface is represented by a line segment $\overline{\boldsymbol{X}_k^{v31}\boldsymbol{X}_k^{v32}}$. Our target is to compute the area of the sub triangle $\blacktriangle \boldsymbol{X}_k^{v31}\boldsymbol{X}_k^{v3}\boldsymbol{X}_k^{v32}$ with the help of the linearized LS field derived in Equation 4.9.

Let us assume that the vertices $\boldsymbol{X}_k^{v1}$ and $\boldsymbol{X}_k^{v2}$ are mapped to $\boldsymbol{\xi} = (1,0)^T$ and $\boldsymbol{\xi} = (0,1)^T$ respectively, in logical space. Therefore, points $\boldsymbol{X}_k^{v31}$ and $\boldsymbol{X}_k^{v32}$ are mapped to $\boldsymbol{\xi}^E = (\xi_1^E, 0)^T$ and $\boldsymbol{\xi}^F = (0, \xi_2^F)^T$, respectively. This is shown in Figure 4.3. In order to compute the area of the region $\boldsymbol{X}_k^{v31}\boldsymbol{X}_k^{v3}\boldsymbol{X}_k^{v32}$, the images of the points E and F are needed. In the next subsection expressions are derived for $\boldsymbol{\xi}^E$ and $\boldsymbol{\xi}^F$.

Coordinates of the point $\boldsymbol{\xi}^E$

The logical space coordinates of the cell centroid $\boldsymbol{\xi}_k$ are by virtue of the barycentric coordinate transformation always given by $\boldsymbol{\xi}_k = \left(\frac{1}{3}, \frac{1}{3}\right)^T$. This means 4.9 can be written as (dropping the explicit dependence on t):

$$\tilde{\phi}(\boldsymbol{\xi}) = \Phi_k + \frac{\partial \Phi}{\partial \xi_1}\left(\xi_1 - \frac{1}{3}\right) + \frac{\partial \Phi}{\partial \xi_2}\left(\xi_2 - \frac{1}{3}\right). \tag{4.10}$$

Substitution of $\boldsymbol{\xi}^E$ in 4.10 leads to:

$$\tilde{\phi}(\boldsymbol{\xi}^E) = \Phi_k + \frac{\partial \Phi}{\partial \xi_1}\left(\xi_1^E - \frac{1}{3}\right) - \frac{\partial \Phi}{\partial \xi_2}\left(\frac{1}{3}\right) = 0. \tag{4.11}$$

The right-hand side of 4.11 is zero, because the interface passes through the point $\boldsymbol{\xi}^E$. Evaluation of 4.10 in $\boldsymbol{\xi}(\boldsymbol{X}_k^{v3}) = \boldsymbol{0}$ gives:

$$\tilde{\phi}(\boldsymbol{0}) = \Phi_k - \frac{\partial \Phi}{\partial \xi_1}\left(\frac{1}{3}\right) - \frac{\partial \Phi}{\partial \xi_2}\left(\frac{1}{3}\right). \tag{4.12}$$

where $\tilde{\phi}(\mathbf{0})$ is the value of the linearized LS field at the origin in logical space. Solving 4.11 and 4.12 to find ξ_1^E leads to:

$$\xi_1^E = \left| -\frac{\tilde{\phi}(\mathbf{0})}{D_{\xi_1}} \right|, \tag{4.13}$$

where $D_{\xi_1} = \dfrac{\partial \Phi}{\partial \xi_1}$. Therefore, the logical space coordinates of the point E are given by $\boldsymbol{\xi}^E = \left(\dfrac{\tilde{\phi}(\mathbf{0})}{D_{\xi_1}}, 0 \right)$.

Using a similar procedure in the ξ_2 direction the coordinates of the point $\boldsymbol{\xi}^F$ in logical space can be found as $\boldsymbol{\xi}^F = \left(0, \dfrac{\tilde{\phi}(\mathbf{0})}{D_{\xi_2}} \right)$ and $D_{\xi_2} = \dfrac{\partial \Phi}{\partial \xi_2}$.

Evaluation of the VoF from the LS function

Now, we can compute the area enclosed by the points $\mathbf{0}$, $\boldsymbol{\xi}^E$ and $\boldsymbol{\xi}^F$ in logical space. The area of the enclosed region is denoted by $A_{\boldsymbol{\xi}}$ and is given by:

$$A_{\boldsymbol{\xi}} = \frac{(\tilde{\phi}(\mathbf{0}))^2}{2 D_{\xi_1} D_{\xi_2}}. \tag{4.14}$$

Equation 4.14 represents the area of the region enclosed by line segments that connect $\mathbf{0}$, $\boldsymbol{\xi}^E$ and $\boldsymbol{\xi}^F$ in logical space. The product of $A_{\boldsymbol{\xi}}$ and the Jacobian of the transformation J is equal to the area enclosed by the points $\boldsymbol{X}_k^{v31} \boldsymbol{X}_k^{v3} \boldsymbol{X}_k^{v32}$ in physical space.

The area of the image of Ω_k in logical space, $A_{\boldsymbol{\xi}}^{\text{total}}$, is $\frac{1}{2}$. The VoF in both physical and logical space can be defined as,

$$\Psi(\boldsymbol{x}_k) = \frac{A_{\boldsymbol{\xi}}}{A_{\boldsymbol{\xi}}^{\text{total}}} = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}}. \tag{4.15}$$

This is the VoF function that returns the VoF value in each triangular element when the value of the linearized LS field is known at the cell center. However, 4.15 is only valid when the domain of interest is triangular. This means it is not valid when the interface intersects or passes above $\boldsymbol{\xi}^A$. The domain of 4.15 can be derived by considering the limit case of the interface intersecting $\boldsymbol{\xi}^A$. The linearized LS at the origin is given by:

$$\tilde{\phi}(\mathbf{0}) = \Phi_k - \frac{D_{\xi_1}}{3} - \frac{D_{\xi_2}}{3}. \tag{4.16}$$

This can be written as:

$$\tilde{\phi}(\mathbf{0}) = \Phi_k - \frac{D_{\xi_1}}{3} + \frac{2D_{\xi_2}}{3} - D_{\xi_2} \Leftrightarrow$$

$$\tilde{\phi}(\mathbf{0}) = \tilde{\phi}(\boldsymbol{\xi}^{\boldsymbol{F}}) - D_{\xi_2}.$$

Under the assumption that the interface is passing through the vertex $\boldsymbol{\xi}^{\boldsymbol{A}}$, $\tilde{\phi}(\boldsymbol{\xi}^{\boldsymbol{F}}) = 0$ by definition and

$$\tilde{\phi}(\mathbf{0}) = -D_{\xi_2}. \tag{4.17}$$

This means that for all the values of $\tilde{\phi}(\mathbf{0}) \geq -D_{\xi_2}$ 4.15 holds. Furthermore, the assumptions on the mapping to logical space require $\tilde{\phi}(\mathbf{0}) \geq 0$ and

$$\Psi(x_k) = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}}, \qquad -D_{\xi_2} \leq \tilde{\phi}(\mathbf{0}) \leq 0. \tag{4.18}$$

The range of the VoF function can now be defined and the complete definition of the VoF function for Case-1 is given as:

$$\Psi : [-D_{\xi_2}, 0] \times \mathbb{R}^2 \to [0, \frac{D_{\xi_2}}{D_{\xi_1}}], \quad \Psi(\tilde{\phi}(\mathbf{0}), \boldsymbol{D}_{\boldsymbol{\xi}}) = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}}. \tag{4.19}$$

Note that the definition of the mapping implies $D_{\xi_1} D_{\xi_2} \geq 0$.

If the interface passes through one of the vertices, 4.15 is valid. This is illustrated in Figure 4.4 where the interface is passing through vertex $\boldsymbol{X}_k^{v1}$. In this case vertex $\boldsymbol{X}_k^{v3}$ is selected to be mapped to the origin and vertex $\boldsymbol{X}_k^{v1}$ is mapped to the ξ_2 axis, as it has the lower magnitude of the LS field of the two other vertices i.e. $\Phi(\boldsymbol{X}_k^{v1}) = 0$. If the interface is exactly

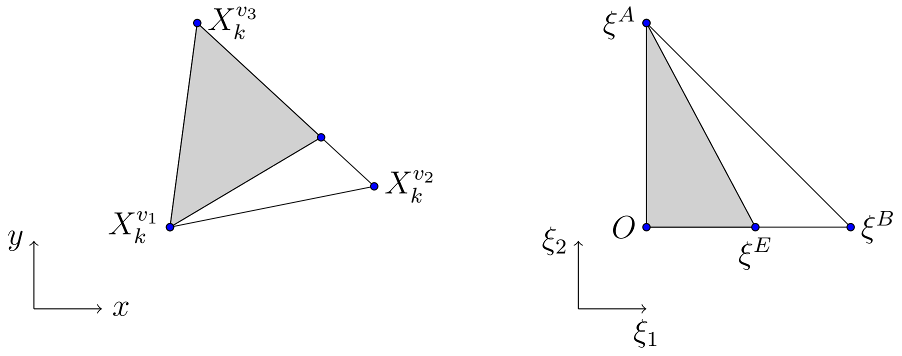


Figure 4.4: Interface passing through vertex $(X_k^{v_1})$.

aligned with one of the edges of the triangular element Ψ_k is either 0 or 1, depending on the fluid of interest. In this case there is no need to use 4.19. The definition of the mapping excludes the possibility that the interface is aligned with one of the coordinate axes in logical space for Case-1. However, for Case-2 it is possible that the interface ends up aligned with one of the axes in the logical space.

Case-2: quadrilateral domain of interest

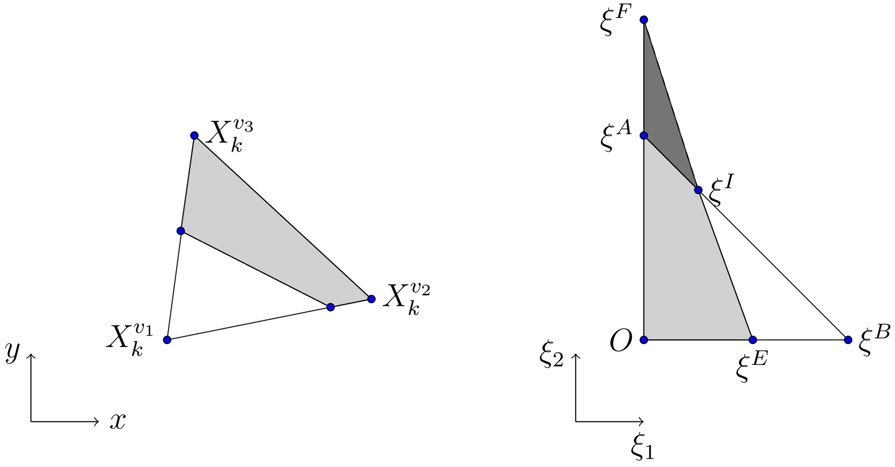


Figure 4.5: Case-2: Mapping of two nodes a side.

An example of Case-2 is presented in Figure 4.5, where vertices $\Phi(\boldsymbol{X}_k^{v3}) < 0$ and $\Phi(\boldsymbol{X}_k^{v2}) < 0$ are now both located in the domain of interest. We assume vertex $\boldsymbol{X}_k^{v2}$ has a lower magnitude of the LS as compared to vertex $\boldsymbol{X}_k^{v3}$. Therefore, vertex $\boldsymbol{X}_k^{v2}$ is mapped to the ξ_2 axis and $\boldsymbol{X}_k^{v3}$ to the origin in logical space. The fluid of interest is enclosed by the quadrilateral $\mathbf{0}, \boldsymbol{\xi}^E, \boldsymbol{\xi}^I, \boldsymbol{\xi}^A$ (Figure 4.5). The vertex $\boldsymbol{\xi}^F$ is located outside the mapped region and intersects the ξ_2 axis, provided that the interface is not parallel to one of the edges of the triangle. This will only occur, when the interface is exactly equidistant from the vertices $\boldsymbol{X}_k^{v3}$ and $\boldsymbol{X}_k^{v2}$. This case will be discussed separately in Section 4.4.2.

We can now use 4.14 to compute the area of the triangular region enclosed by vertices $\mathbf{0}, \boldsymbol{\xi}^E, \boldsymbol{\xi}^I, \boldsymbol{\xi}^F$ and $\boldsymbol{\xi}^A$. Let us call this area $\boldsymbol{A_{tri.}}$. To find the area of the quadrilateral region $\boldsymbol{A_{\xi}}$ the area of the triangular region defined by the vertices $\boldsymbol{\xi}^F, \boldsymbol{\xi}^I, \boldsymbol{\xi}^A$, which we will refer to as $\boldsymbol{A_{ex.}}$, needs to be subtracted from the area $\boldsymbol{A_{tri.}}$:

$$\boldsymbol{A_{\xi}} = \frac{(\tilde{\phi}(\mathbf{0}))^2}{2D_{\xi_1}D_{\xi_2}} - \boldsymbol{A_{ex.}}. \tag{4.20}$$

In order to compute $\boldsymbol{A_{ex.}}$, the coordinates of the point $\boldsymbol{\xi}^I$ are required. This point is the intersection of the interface and the edge of the triangle opposing the origin of the reference triangle. At $\boldsymbol{\xi}^I$, we have $\Phi(\boldsymbol{\xi}^I) = 0$. The linearized LS at this point is defined as:

$$\phi(\boldsymbol{\xi}^I) = \Phi_k + D_{\xi_1}\left(\xi_1^I - \frac{1}{3}\right) + D_{\xi_2}\left(\xi_2^I - \frac{1}{3}\right) = 0. \tag{4.21}$$

At the edge $\overline{\boldsymbol{\xi}^A\boldsymbol{\xi}^B}$, we have $\xi_2^I = 1 - \xi_1^I$, using this relation in 4.21 leads to:

$$\xi_1^I = -\frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})}{D_{\xi_1} - D_{\xi_2}}, \quad \xi_2^I = \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_1})}{D_{\xi_1} - D_{\xi_2}}, \tag{4.22}$$

where $\tilde{\phi}(\mathbf{0}) = \Phi_k - \frac{D_{\xi_1}}{3} - \frac{D_{\xi_2}}{3}$. The coordinates of the other two points that define the quadrilateral area are $\boldsymbol{\xi}^F = \boldsymbol{\xi}^F(0, \frac{-\tilde{\phi}(\mathbf{0})}{D_{\xi_2}})$ and $\boldsymbol{\xi}^A = \boldsymbol{\xi}^A(0,1)$. Define the following vectors: $\overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^A}$ and $\overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^I}$:

$$\overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^A} = \left(0, 1 + \frac{\tilde{\phi}(\mathbf{0})}{D_{\xi_2}}\right)^T, \tag{4.23}$$

$$\overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^I} = \left(-\frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})}{D_{\xi_1} - D_{\xi_2}}, \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_1})}{D_{\xi_1} - D_{\xi_2}} + \frac{\tilde{\phi}(\mathbf{0})}{D_{\xi_2}}\right)^T.$$

Then the excess area $\boldsymbol{A_{ex.}}$ is defined as;

$$\boldsymbol{A_{ex.}} = \frac{|[\overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^A} \times \overrightarrow{\boldsymbol{\xi}^F\boldsymbol{\xi}^I}]|}{2} = \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{2D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}. \tag{4.24}$$

Substitution of 4.24 in 4.20 leads to:

$$\boldsymbol{A_{\xi}} = \frac{(\tilde{\phi}(\mathbf{0}))^2}{2D_{\xi_1}D_{\xi_2}} - \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{2D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}. \tag{4.25}$$

The VoF function for Case-2 is now given by:

$$\Psi_k = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1}D_{\xi_2}} - \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}. \tag{4.26}$$

Note that $(D_{\xi_1} - D_{\xi_2})$ cannot be zero, because this will only occur when the interface is parallel to the edge $\overline{\boldsymbol{\xi}^A\boldsymbol{\xi}^B}$. The domain of the VoF function 4.26 needs to be defined such that the range of the function is a subset of the unit interval. In order to derive an upper bound, consider Figure 4.5

. Because of the definition of the mapping $|\tilde{\phi}(\boldsymbol{\xi^A})| < |\tilde{\phi}(\boldsymbol{\xi^B})|$. Therefore, the interface will pass outside the element when $\tilde{\phi}(\mathbf{0}) > -D_{\xi_1}$. The value of $\tilde{\phi}(\mathbf{0}) = -D_{\xi_1}$ corresponds to $\Psi_k = 1$. Therefore, the domain of the VoF function 4.26 is equal to the interval $[-D_{\xi_1}, -D_{\xi_2}]$ and

$$\Psi_k = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}} - \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{D_{\xi_2}(D_{\xi_1} - D_{\xi_2})} \quad \forall \quad -D_{\xi_1} \leq \tilde{\phi}(\mathbf{0}) \leq -D_{\xi_2}. \tag{4.27}$$

Given the domain of the VoF function, its range can be derived. The complete definition of the VoF function for a quadrilateral domain of interest is given as:

$$\begin{aligned} \Psi : [-D_{\xi_1}, -D_{\xi_2}] \times \mathbb{R}^2 &\to \left[\frac{D_{\xi_2}}{D_{\xi_1}}, 1\right], \\ \Psi(\tilde{\phi}(\mathbf{0}), \boldsymbol{D_\xi}) &= \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}} - \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}. \end{aligned} \tag{4.28}$$

When the interface is parallel to one of the edges

Equation 4.28 is derived for the case when the interface is not parallel to the edge $\overline{O\boldsymbol{\xi^A}}$. The latter occurs when the interface is equidistant from both vertices $\boldsymbol{O}$ and $\boldsymbol{\xi^A}$. In this case the extra area would become unbounded, so equation 4.26 can not be used. In this situation the domain of interest is a trapezoid as shown in Figure 4.6 and its area can be computed easily as:

$$\boldsymbol{A_\xi} = \frac{-\tilde{\phi}(\mathbf{0})(\tilde{\phi}(\mathbf{0}) + 2D_{\xi_1})}{2}, \tag{4.29}$$

and the VoF for this case is given as:

$$\Psi_k = -\tilde{\phi}(\mathbf{0})(\tilde{\phi}(\mathbf{0}) + 2D_{\xi_1}) \tag{4.30}$$

4.4.3 Combining the VoF functions for Case-1 and Case-2

The union of the domains of the two VoF functions for Case-1 and Case-2 together covers the set $[-D_{\xi_1}, 0]$ of possible values of $\tilde{\phi}(\mathbf{0})$, under the assumption that the control volume of interest is intersected by the interface. Both VoF functions have a quadratic dependence on $\tilde{\phi}(\mathbf{0})$ and are C^1 continuous at the common point $\tilde{\phi}(\mathbf{0}) = -D_{\xi_2}$. This is shown in Figure 4.7. The range of the VoF function should be $[0, 1]$.

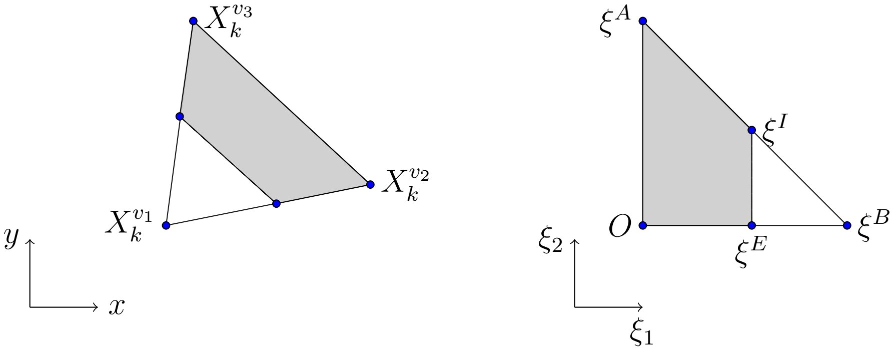


Figure 4.6: Mapping of the interface parallel to one of its edge.

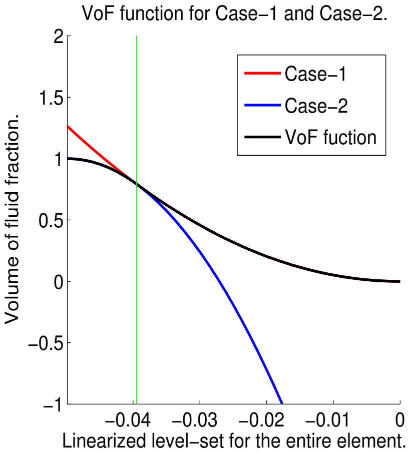


Figure 4.7: VoF function for both Case-1 and Case-2.

In order to select correct branch of the two VoF functions, we define a switching parameter $S_p(\tilde{\phi}(\mathbf{0}))$ as:

$$S_p(\tilde{\phi}(\mathbf{0})) = \begin{cases} 0 & : \tilde{\phi}(\mathbf{0}) \geq -D_{\xi_2} \\ 1 & : \tilde{\phi}(\mathbf{0}) < -D_{\xi_2} \end{cases} . \tag{4.31}$$

Using 4.31, 4.19 and 4.28 can be combined to a single VoF function as:

$$\Psi : [-D_{\xi_1}, 0] \times \mathbb{R}^2 \rightarrow [0,1] \,, \tag{4.32}$$

$$\Psi(\tilde{\phi}(\mathbf{0}), \boldsymbol{D_\xi}) = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}} - S_p(\tilde{\phi}(\mathbf{0})) \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}.$$

Let us further define two coefficients:

$$c_1 = \frac{1}{D_{\xi_1} D_{\xi_2}}, \qquad c_2 = \frac{-S_p(\tilde{\phi}(\mathbf{0}))}{D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}. \tag{4.33}$$

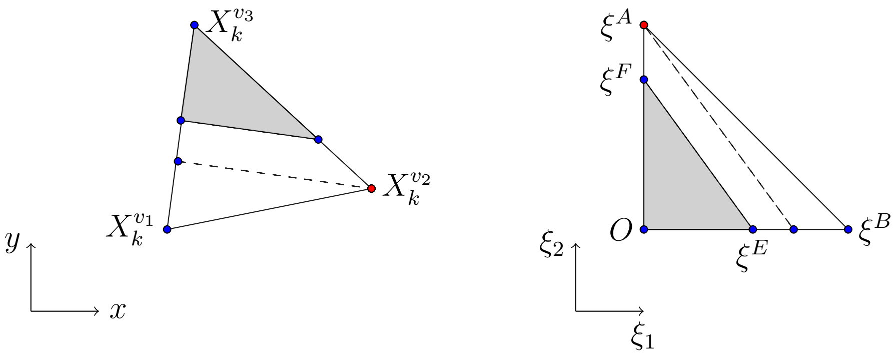


Figure 4.8: Combination of two cases.

Using 4.33 and 4.32 a uniformly valid expression for Ψ_k can be formulated as:

$$\Psi_k = (c_1 + c_2)(\tilde{\phi}(\mathbf{0}))^2 + 2c_2 D_{\xi_2}(\tilde{\phi}(\mathbf{0})) + c_2 D_{\xi_2}^2 \;\forall \quad -D_{\xi_1} \leq \tilde{\phi}(\mathbf{0}) \leq 0. \tag{4.34}$$

It is important to note that 4.34 also holds for the case when the interface is aligned with or parallel to the edge $\overline{O\boldsymbol{\xi}^{\boldsymbol{A}}}$ i.e. $D_{\xi_2} = 0$. In Case-1, i.e. $S_p(\tilde{\phi}(\mathbf{0})) = 0$, the interface can be aligned with edge $\overline{O\boldsymbol{\xi}^{\boldsymbol{A}}}$ and in this situation the VoF is either 0 or 1. In Case-2, i.e. $S_p(\tilde{\phi}(\mathbf{0})) = 1$, the interface can be parallel to the edge $\overline{O\boldsymbol{\xi}^{\boldsymbol{A}}}$. but the behavior of the coefficients $c_1 + c_2$ and $D_{\xi_2} c_2$ in the polynomial is not singular when $D_{\xi_2} = 0$.

The VoF function 4.34 is expressed in the value of the linearized LS field at the origin in logical space and the partial derivatives of the LS function with respect to the logical space coordinates. However, we have defined the VoF function 4.4 in terms of the cell centroid value and gradient of the LS. To formulate the VoF function explicitly in the latter form 4.16 is used in 4.34 to find the following expression for $f(\Phi_k, \nabla\Phi_k)$:

$$\Psi_k = f(\Phi_k, \nabla\Phi_k) = [\frac{-2D_{\xi_1} + D_{\xi_2}}{3}, \frac{D_{\xi_1} + D_{\xi_2}}{3}] \rightarrow [0, 1], \tag{4.35}$$

$$f(\Phi_k, \nabla\Phi_k) = (c_1 + c_2)\Phi_k^2 + \left(\frac{(c_1 + c_2)D_{\xi_1}^2 + (2c_1 - 4c_2)D_{\xi_1}D_{\xi_2} + (c_1 + 4c_2)D_{\xi_2}^2}{9}\right)\Phi_k$$

$$-\frac{2\left((c_1 + c_2)D_{\xi_1} + (c_1 - 2c_2)D_{\xi_2}\right)}{3}, \quad [D_{\xi_1} \;\; D_{\xi_2}]^T = J\nabla\Phi_k.$$

4.5 Advection of the LS field

Because the interface is by definition a contour line of the LS function, the following equation holds *at the interface*:

$$\frac{d}{dt}\Phi(\boldsymbol{x},t)=0 \Rightarrow \frac{\partial \Phi(\boldsymbol{x},t)}{\partial t}+u_\alpha(\boldsymbol{x},t)\Phi(\boldsymbol{x},t)_{,\alpha}=0,\ \boldsymbol{x}\in \boldsymbol{X}(t),\ t>0. \tag{4.36}$$

In the MCLS method equation 4.36 is postulated to hold for all $\boldsymbol{x}\in\Omega$, but other choices are possible, as long as they are consistent with 4.36 and lead to a function that is at least C^2 continuous in the vicinity of the interface to allow computation of the curvature. For a solenodial velocity field $u_\alpha(\boldsymbol{x},t)$ the extension of (4.36) is given by (omitting the explicit dependence of $\boldsymbol{x}$ ant t of $u_\alpha(\boldsymbol{x},t)$):

$$\frac{\partial \Phi(\boldsymbol{x},t)}{\partial t}+\left(u_\alpha\Phi(\boldsymbol{x},t)\right)_{,\alpha}=0,\ \boldsymbol{x}\in\Omega,\ t>0. \tag{4.37}$$

This equation shows that the LS function is conserved. However, a conservative redistribution of the LS function can lead to a change in the area enclosed by the interface. This is one of the drawbacks associated with the LS method. To some extent this can be remedied by using higher order approximations of the convection operator, e.g. using ENO or WENO schemes in the context of a finite volume discretisation [26, 57], combined with higher order time-integration methods or by applying adaptive grid refinement near the interface. However, all the remedies are not computationally efficient when 4.37 is discretised on an unstructured set of triangular control volumes. In the current paper a higher-order discontinuous Galerkin (DG) finite element method is used for the discretization of 4.37. This approach leads to a high order of accuracy boundary conforming discretisation for domains with arbitrary geometrical complexity. For integration in time a low storage Runge-Kutta method is used. A brief description of the spatial and temporal discretization of 4.37 is presented, based on [19,20,23,34]. The computational domain Ω is discretized into K non-overlapping, straight sided triangular elements Ω_k. Lagrange polynomials $L_i(\boldsymbol{x})$ of degree N are used as basis functions to expand the solution in element Ω_k as:

$$\Phi_k^h(\boldsymbol{x},t)=\Sigma_{i=1}^m\phi_i^k(t)L_i(\boldsymbol{x}),\ \boldsymbol{x}\in\Omega_k,\ t>0, \tag{4.38}$$

where m is the total number of nodal points in an element. In the DG frame work, the residual is made orthogonal to the polynomial space in Ω_k, by requiring (omitting the explicit dependence of $\boldsymbol{x}$ ant t of $\Phi_k^h(\boldsymbol{x},t)$)

$$\int_{\Omega_k}\left(\frac{\partial\Phi_k^h}{\partial t}+(u_\alpha\Phi_k^h)_{,\alpha}\right)L_i(\boldsymbol{x})d\Omega=0,\ i=1\ldots m. \tag{4.39}$$

The weak form of Equation 4.37 can be obtained by applying integration by parts:

$$\int_{\Omega_k} \frac{\partial \Phi_k^h}{\partial t} L_i(\boldsymbol{x}) - L_{i,\alpha}(\boldsymbol{x}) u_\alpha \Phi_k^h d\Omega = -\oint_{\partial\Omega_k} (\hat{n}_\alpha u_\alpha \Phi^h)^* L_i(\boldsymbol{x}) d\Omega,\ i = 1 \dots m, \tag{4.40}$$

where $\hat{n}_\alpha$ is the outward pointing unit normal and $(u_\alpha \Phi^h)^*$ is the numerical flux, used to impose boundary conditions on each element. Integrating by parts once again leads to the strong formulation of the DG method:

$$\int_{\Omega_k} \left(\frac{\partial \Phi^h}{\partial t} + (u_\alpha \Phi^h)_{,\alpha} \right) L_i(\boldsymbol{x}) d\Omega = \oint_{\partial\Omega_k} n_\alpha (u_\alpha \Phi^h - (u_\alpha \Phi^h)^*) L_i(\boldsymbol{x}) d\Omega,\ i = 1 \dots m. \tag{4.41}$$

In the MCLS 4.41 is combined with the Lax-Friedrichs approximation of the numerical flux [23]. The system (4.41) is formulated for all control volumes $\Omega_k,\ k = 1 \dots K$. All equations are coupled through the numerical flux function and advanced simultaneously.

4.6 Advection of the Volume of Fluid field

The MCLS algorithm uses a simultaneously advanced VoF field to impose a correction on the LS field, to make the latter mass conserving. An evolution equation for the VoF field is derived by taken the derivative of 4.3, with respect to time:

$$\frac{d\Psi_k(t)}{dt} = \frac{d}{dt} \frac{1}{|\Omega_k|} \int_{\Omega_k} H(\Phi(\boldsymbol{x}, t)) d\Omega, \tag{4.42}$$

Because $H(\Phi(\boldsymbol{x}, t))$ is a material property and assuming a solenoidal velocity field $u_\alpha(\boldsymbol{x}, t)$, the right-hand side of 4.42 can be formulated as (dropping the explicit dependences on $\boldsymbol{x}$ and t);

$$\frac{1}{|\Omega_k|} \int_{\Omega_k} \frac{dH(\Phi)}{dt} d\Omega = -\frac{1}{|\Omega_k|} \int_{\Omega_k} H_{,\alpha}(\Phi) u_\alpha d\Omega = -\frac{1}{|\Omega_k|} \int_{\partial\Omega_k} H(\Phi) u_\alpha \hat{n}_\alpha dS, \tag{4.43}$$

where, $\hat{n}_\alpha$ is the normal vector on the boundary of the element. Therefore, the evolution of the VoF field is governed by:

$$\frac{d\Psi_k}{dt} = -\frac{1}{|\Omega_k|} \int_{\partial\Omega_k} H(\Phi) u_\alpha \hat{n}_\alpha dS \tag{4.44}$$

The fact that $H(\Phi)$ is C^{-1} continuous on $\partial\Omega$ makes accurate discretisation of 4.44 very challenging. The application of 'standard' schemes developed for scalar hyperbolic equations leads either to strong oscillations in the solution or unacceptable smearing of the interface, jeopardizing mass conservation.

For Cartesian control volumes, accurate and efficient discretisation schemes have been developed that utilize directional splitting, e.g. the scheme presented in [49] and its strictly mass conserving adaptation [58]. Apart from the latter scheme, strict mass conservation can only be achieved with very intricate *unsplit* schemes. Also the method proposed by Jofre, Lluıs & et. al. [8, 12, 27, 28] can be suitable choice for the VoF advection but the interface reconstruction based on the VoF field is computationally involved.

In this paper we propose to use a different approach for the advection of the VoF field. This approach is also used in [60] and is known as Eulerian-Lagrangian VoF evolution. The algorithm consists of three steps: Lagrangian advection of the color function, interface reconstruction, and remapping.

In the first step each mesh element is considered as a material element and advected in a Lagrangian frame of reference. During this process the color function (a material property) is passively advected with the flow in each element. Because the velocity field is solenodial each element retains its VoF value while being deformed and rotated. Furthermore, if a divergence free *linear* velocity field is assumed this ensures a straight sided triangle remains a straight sided triangle, irrespective of how the relative position of the vertices changes (excluding folding). The assumption of a piecewise linear velocity distribution is consistent with the linear approximation of the interface.

The approach is based on using two grids. The first one is a (fixed) Eulerian grid and the second one a Lagrangian grid that consists of the same elements whose vertices have been translated. As the mesh elements are advected such that their area remains the same, this advection is consistent with Equation 4.43, i.e. the advection is mass conserving.

The coordinates of the vertices of the Lagrangian grid are obtained by using a second order Runge-Kutta (RK) scheme, defined as,

$$\boldsymbol{X}_{k_L}^{v\ n+\frac{1}{2}} = \boldsymbol{X}_{k_E}^{v} + \frac{\Delta t}{2}\mathbf{u}(\boldsymbol{X}_{k_E}^{v}, t^n), \tag{4.45}$$

$$\boldsymbol{X}_{k_L}^{n+1} = \boldsymbol{X}_{k_E}^{v} + \Delta t\mathbf{u}(\boldsymbol{X}_{k_L}^{n+\frac{1}{2}}, t^{n+\frac{1}{2}}), \tag{4.46}$$

where, $k = 1....., N_v$ and N_v is the total number of vertices. The $\boldsymbol{X}_{k_L}^{v\ n+1}$ represent the vertices of the Lagrangian mesh, $\boldsymbol{X}_{k_E}^{v}$ the vertices of the Eulerian mesh. Because of the assumptions on the velocity field, the advection

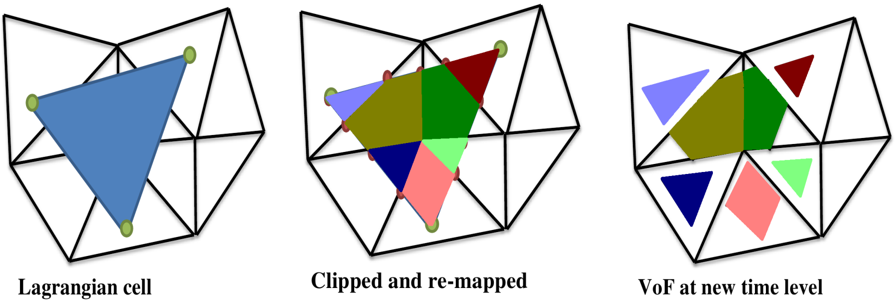


Figure 4.9: Advection, remapping and distribution of the content of a control volume that is *not* intersected by the interface.

defines an element-wise linear map from the Eulerian to the Lagrangian grid. The interface position in each element is defined by this mapping and does not need to be reconstructed. Once the interface position at the Lagrangian mesh is known, the color function can be mapped from the Lagrangian mesh back to the Eulerian mesh by means of a geometric process known as polygon-polygon *clipping* as described in [5, 60, 63].

In order to explain the clipping procedure, the elements of the Eulerian mesh are divided into two groups. We will refer to the cells that are completely filled with $Phase_0$ as *fully filled* cells. Those elements that contain both $Phase_0$ and $Phase_1$ will be referred to as *mixed cells.* In Figure 4.9 and 4.10 both cases are presented. A fully filled element is advected, as shown in Figure 4.9, using Equations 4.45 and 4.46. This determines the position of the Lagrangian element corresponding to this Eulerian element. Now the fluid contained in this element needs to be distributed among the elements of the Eulerian grid. We refer to this process by *remapping*. Remapping of a fully filled cell on the Lagrangian grid is carried out by first locating the set of elements on the Eulerian grid that have a non-empty intersection with the Lagrangian element. Each of these intersections are polygonal regions defined by the coordinates of the intersections of the edges of the Lagrangian element with the Eulerian mesh.

A similar procedure is adopted for a mixed cell, as shown in Figure 4.10. In this case the intersection of that part of the Lagrangian cell that is occupied by $Phase_0$ with the elements of the Eulerian mesh has to be determined. Each of these intersections are polygonal regions defined by the coordinates of the intersections of the edges of the Lagrangian element and the interface with the Eulerian mesh.

The area of the clipped region is computed by means of the following

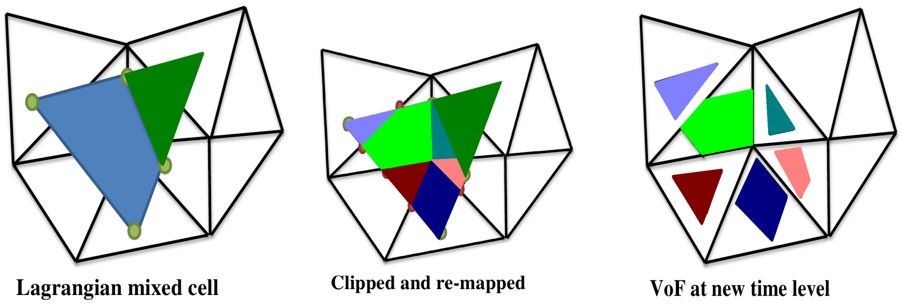


Figure 4.10: Advection, remapping and distribution of the content of a control volume that is intersected by the interface.

relation for the area of a closed polygon in 2D [18, 47]:

$$Area = \frac{1}{2} \mid \sum_{i=0}^{n-1} (x_1^i x_2^{i+1} - x_1^{i+1} x_2^i) \mid, \qquad (4.47)$$

where i is the index for all the nodes and the first and last node coincide. In the current research, use is made of the MATLAB$^{\text{TM}}$ Mapping Toolbox to perform the polygon-polygon clipping. However, many different polygon clipping algorithms are available from image processing, for example the Sutherland-Hodgeman algorithm and the Weiler-Atherton algorithm [11, 44].

4.6.1 Mass conserving advection of the Volume of Fluid field for nonlinear velocity field

The Eulerian-Lagrangian VoF advection method is accurately mass conserving when the velocity field is linear. In the case of a nonlinear velocity field maintaining this accuracy presents a challenge. This is the result of the assumption that the triangular elements remain straight sided triangles after advection. In reality the nonlinear velocity field will deform the edges of the triangular element and representing it as a straight sided triangle causes a loss or gain of mass (area) at the Lagrangian mesh. This deformation is shown in Figure 4.11. In the proposed algorithm exact mass conservation is reestablished using a method similar to the one proposed in [4.37]. To prevent mass loss or gain due to the nonlinear velocity field an intermediate step is introduced that is applied to the solution on the Lagrangian mesh before applying the clipping and redistribution algorithm. In this step the total mass (area) at the Lagrangian mesh will be enforced to be equal to the total mass at the Eulerian mesh at previous time level

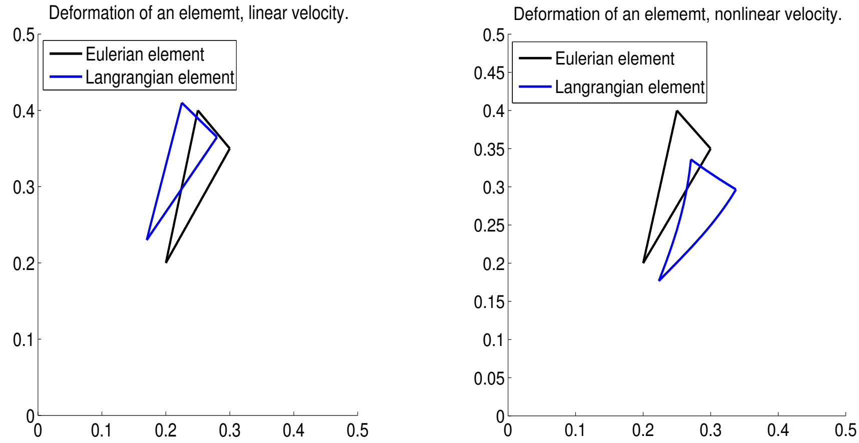


Figure 4.11: Straight-sided triangles remain straight-sided after being advected by a linear velocity field (left). When advected by a nonlinear velocity field this property is lost (right).

t^n. This is done by adjusting the volume fraction in all mixed cells such that the total mass error is compensated for. The complete procedure is explained in the following sections.

Mass error adjustment: Fully filled cells

Consider the fully filled k^{th} element of the Eulerian grid at time t^n. It has a volume fraction $\Psi_k^E = 1$ and area $\mid \Omega_k^E \mid$. Where superscript E refers to the Eulerian grid. The mass (M_k^E) enclosed by this element is equal to:

$$M_k^E = \Psi_k^E \mid \Omega_k^E \mid . \tag{4.48}$$

Now, this element is advected with a nonlinear velocity field which results in a deformed triangle so the area enclosed by the element is not the same as on the Eulerian grid i.e. a mass error has incurred. However, during advection the fully filled cell should remain fully filled so it has volume fraction $\Psi_k^L = 1$ and the area $\mid \Omega_k^L \mid$. Where superscript L represents the Lagrangian grid. The mass (M_k^L) enclosed by the Lagrangian element is;

$$M_k^L = \Psi_k^L \mid \Omega_k^L \mid . \tag{4.49}$$

The mass lost or gained by the element due to the deformation caused by the nonlinear velocity field is

$$M_k^{err.} = M_k^E - M_k^L. \tag{4.50}$$

Where, $M_k^{err.}$ is the mass error of the k^{th} fully filled cell. If there are N_f fully filled elements that are advected, using a nonlinear velocity field, then the total mass error $(TM^{err.})$ caused by the fully filled cells can be defined as

$$TM^{err.} = \sum_{i=1}^{N_f} \left(M_i^E - M_i^L\right). \tag{4.51}$$

The total mass error computed in Equation 4.51 is distributed evenly among the mixed cells. If there are N_m mixed cells then the average mass error (M_{avg}) can be defined as:

$$M_{avg.} = \sum_{i=1}^{N_f} \left(\frac{M_i^E - M_i^L}{N_m}\right). \tag{4.52}$$

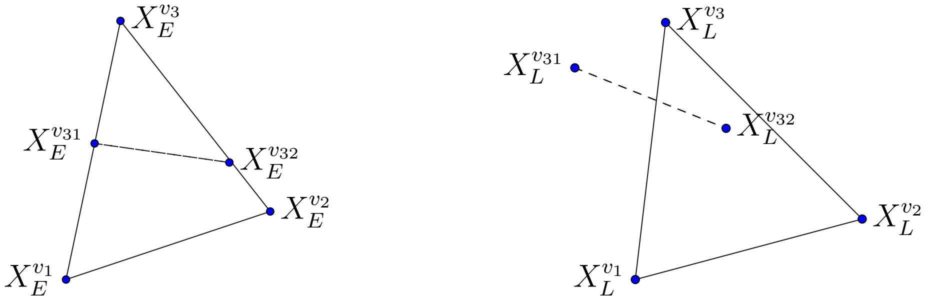
This average mass error is added to each of the mixed cells to ensure global mass conservation.

Mass error adjustment: Partially filled cells

In case of a partially filled or mixed cell a different procedure is adopted. The interface position is adjusted in each mixed control volume in the Lagrangian grid such that the mass contained within is equal to the mass contained within the corresponding control volume on the Eulerian grid plus the average mass error that follows from 4.52. Both the position and the orientation of the interface have to be determined. The orientation of the interface can be determined from the LS field, but this approach would require the interpolation of the LS field from the Eulerian to the Lagrangian grid at time t^{n+1}. This idea is not computationally efficient, for details see [60]. To mitigate this problem a different procedure is used, which is based on the assumption that the interface can be reconstructed on the Lagrangian element by advecting the two points that represent the interface in the corresponding Eulerian element. Naturally, this puts a threshold on the size of the time step that can be taken. However, the idea is computationally efficient and also in line with the idea of distributing the average mass error to the mixed cells. The details of the interface adjustment for the control volumes in the Lagrangian mesh are presented in the following subsection.

Interface adjustment for partially filled cells

Consider a partially filled cell at the Eulerian grid at time t^n, as shown in Figure 4.12(a). The vertices of the element are $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_1}}$, $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_2}}$ and $\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v_3}}$. The



(a) Eulerian element. (b) Lagrangian element.

Figure 4.12: The position of the interface in a mixed cell in the Lagrangian mesh is based on the advection of the extremities of the interface in the corresponding mixed cell in the Eulerian mesh.

interface is represented by a linear polynomial i.e. a line segment defined by its end points $\boldsymbol{X_E^{v31}}$ and $\boldsymbol{X_E^{v32}}$. Also, the area of the triangle formed by $\boldsymbol{X_E^{v3}}$, $\boldsymbol{X_E^{v31}}$ and $\boldsymbol{X_E^{v32}}$ represents the exact amount of fluid of interest in this element. The location of the points $\boldsymbol{X_E^{v31}}$ and $\boldsymbol{X_E^{v32}}$ is obtained by using the LS information of the element. Assume vertex $\boldsymbol{X_E^{v3}}$ is in the negative LS region, then $\boldsymbol{X_E^{v1}}$ and $\boldsymbol{X_E^{v2}}$ are in the positive region and the interface normal points towards the positive LS region. Using the linear interpolation parameters β_i for $i = 1, 2$ the intersection of the interface with the edges $\overline{\boldsymbol{X_E^{v3} X_E^{v1}}}$ and $\overline{\boldsymbol{X_E^{v3} X_E^{v2}}}$ can be found in the following way:

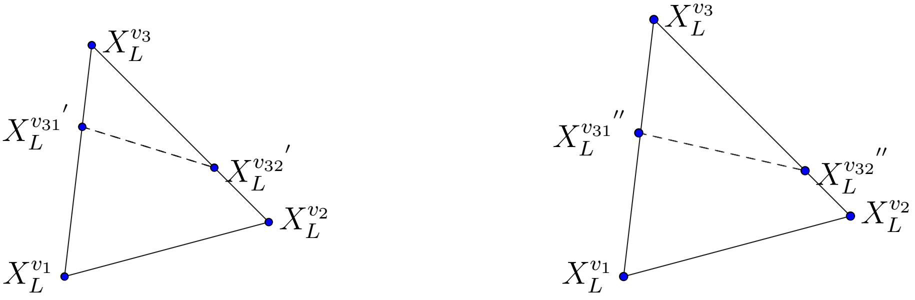
$$\beta_i = \frac{-\Phi(\boldsymbol{X_E^{v3}})}{\Phi(\boldsymbol{X_E^{v3}}) - \Phi(\boldsymbol{X_E^{v_i}})}, \tag{4.53}$$

where $\Phi(\boldsymbol{X_E^{v1}})$, $\Phi(\boldsymbol{X_E^{v2}})$ and $\Phi(\boldsymbol{X_E^{v3}})$ are the LS values at the vertices $\boldsymbol{X_E^{v1}}$, $\boldsymbol{X_E^{v2}}$ and $\boldsymbol{X_E^{v3}}$, respectively. Using the parameters β_i the interface end points $\boldsymbol{X_E^{v31}}$ and $\boldsymbol{X_E^{v32}}$ can be computed as:

$$\boldsymbol{X_E^{v3(i)}} = \beta_i \boldsymbol{X_E^{v(i)}} + (1 - \beta_i)\boldsymbol{X_E^{v3}} \qquad i = 1, 2. \tag{4.54}$$

Once the extremities of the interface are known they are advected along with the vertices of the element, using Equation 4.45 and 4.46. This defines the corresponding Lagrangian element as shown in Figure 4.12(b). The corresponding vertices on the Lagrangian elements are $\boldsymbol{X_L^{v1}}$, $\boldsymbol{X_L^{v2}}$ and $\boldsymbol{X_L^{v3}}$ and the interface extremities are $\boldsymbol{X_L^{v31}}$ and $\boldsymbol{X_L^{v32}}$, respectively.

The end points of the interface are not necessarily located on the edges. This is because the triangle has deformed. In order to avoid overlapping with neighboring elements. The interface end points are projected on the



(a) Interface projection. (b) Mass adjustment.

Figure 4.13: For a nonlinear velocity field, the advected extremities of the interface are first projected on the edges of the Lagrangian control volume, before the appropriate interface position can be determined.

edges of the element. This is done by computing the intersection of the edges $\overline{\boldsymbol{X_L^{v3}X_L^{v1}}}$ and $\overline{\boldsymbol{X_L^{v3}X_L^{v2}}}$ with the interface. This defines points $\boldsymbol{X_L^{v31}}'$ and $\boldsymbol{X_L^{v32}}'$, as shown in Figure 4.13(a). The interface orientation is still the same, the only difference is that it is mapped to the edges of the Lagrangian element. Clearly, the interface can be reconstructed in a straightforward way on the Lagrangian mesh without the need for information of the LS field.

After the interface reconstruction step, the goal is to achieve exact mass conservation i.e. the interface has to be shifted from its current position to make sure the mass within the Lagrangian control volume corresponds to the mass in the corresponding Eulerian control volume plus the average mass error defined by 4.52.

Assume the area of the Eulerian element is $A_E = \blacktriangle \boldsymbol{X_E^{v3}X_E^{v31}X_E^{v32}}$ and the area of the Lagrangian element is $A_L = \blacktriangle \boldsymbol{X_L^{v3}X_L^{v31'}X_L^{v32'}}$. Now the goal is to find the points on the edges $\overline{\boldsymbol{X_L^{v3}X_L^{v1}}}$ and $\overline{\boldsymbol{X_L^{v3}X_L^{v2}}}$, such that the area enclosed by these points and the vertex $\boldsymbol{X_L^{v3}}$ is equal to A_E plus $M_{avg.}$. This is achieved by computing the following ratio,

$$\theta = \sqrt{\frac{A_E + M_{avg.}}{A_L}}. \tag{4.55}$$

Using, the value θ points on the edges can be computed as follows,

$$\boldsymbol{X_L^{v3(i)}}'' = \theta \boldsymbol{X_L^{v3(i)}}' + (1-\theta)\boldsymbol{X_L^{v3}}, \tag{4.56}$$

where, the points $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v3(i)}''}$, for $i = 1, 2$, correspond to the edges $\overline{\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v3}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v1}}}$ and $\overline{\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v3}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v2}}}$, respectively, as shown in Figure 4.13(b). This makes sure the area is identical to the area of the Eulerian cell. This holds for the case where the fluid of interest is within a triangular region $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v3}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v31}'}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v32}'}$. For the region enclosed by the points $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v1}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v2}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v32}'}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v31}'}$, a slightly different procedure is adopted. Here, it is important to note that as the triangles are allowed to deform during advection, the total area of the Lagrangian element is not equal to the total area of the corresponding Eulerian element. Therefore, using equation 4.55, to conserve the area of the region by subtracting the area enclosed in $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v1}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v2}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v32}'}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v31}'}$ from A_E will not conserve the region $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v1}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v2}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v32}'}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v31}'}$ on the Lagrangian mesh. In order to circumvent this problem equation 4.55 is modified. Figure 4.13 is used again to explain the procedure, as all other steps except the use of equation 4.55 are the same. Consider a fluid of interest enclosed in the region $A_E = \blacksquare\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v1}}\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v2}}\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v32}}\boldsymbol{X}_{\boldsymbol{E}}^{\boldsymbol{v31}}$ at the Eulerian mesh. After advection and projecting the interface points at the edges of the Lagrangian cell the corresponding Lagrangian element is $A_L = \blacksquare\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v1}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v2}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v32}'}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v31}'}$ and the total area of the Lagrangian element is $A_{LT} = \blacktriangle\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v1}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v2}}\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v3}}$. Using this information the modified form of equation 4.55 is obtained as;

$$\theta = \sqrt{\frac{A_{LT} - (A_E + M_{avg.})}{A_{LT} - A_L}}. \tag{4.57}$$

Similarly, points $\boldsymbol{X}_{\boldsymbol{L}}^{\boldsymbol{v3(i)}''}$ for $i = 1, 2$, on the edges are computed, using 4.56 . This will ensure the mass (area) of the cell corresponds to the mass of the Eulerian cell. Once mass conservation is achieved at the Lagrangian mesh, the clipping procedure is applied to map the VoF to the Eulerian mesh for time t^{n+1}.

4.7 Inverse function for the Level-Set correction

To make the LS mass-conserving a correction is applied based on the advected VoF field. In this approach the advected LS field Φ_k^* is used to compute a tentative VoF field $\Psi^* = f(\Phi_k^*, \nabla\Phi_k^*)$ in every mixed cell, using the VoF function 4.32. The tentative VoF field is compared locally with the Eulerian-Lagrangian advected VoF field Ψ_k^{n+1}. If the difference is within a given tolerance then there is no need to correct the LS in that particular cell, as it is already mass-conserving, i.e. the LS field complies with 4.3, and $\Phi_k^{n+1} = \Phi_k^*$. However, if this difference exceeds the prescribed tolerance, a correction is required. In order to perform the correction the inverse of $f(\Phi_k, \nabla\Phi_k)$ is required. This function is referred to as the *inverse VoF function* $g(\Psi_k, \nabla\Phi_k)$. This function is derived from 4.32.

Assume the value of the VoF field for Ω_k is given by:

$$\Psi_k^{n+1} = f(\Phi_k^{n+1}, \nabla\Phi_k^{n+1}), \tag{4.58}$$

for yet unknown Φ_k^{n+1}. Because the mass conserving corrections that need to be imposed in the MCLS are very small, the orientation of the interface is assumed to remain unchanged. Therefore, we can replace 4.58 by,

$$\Psi_k^{n+1} = f(\Phi_k^{n+1}, \nabla\Phi_k^*). \tag{4.59}$$

This means that in the conversion the height of the interface is adjusted without changing its orientation. For a triangular (or more generally a convex) element there is a unique value of Φ_k^{n+1} that fulfills 4.59 for $\Psi_k \in [0, 1]$. Therefore, the inverse of $f(\Phi_k, \nabla\Phi_k)$ with respect to its first argument can be defined as

$$\Phi_k = g(\Psi_k, \nabla\Phi_k). \tag{4.60}$$

In order to obtain a uniformly valid inverse function, a switching parameter similar to 4.31 can be defined as:

$$S_p^{'}(\Psi)_k = \begin{cases} 1 & : \Psi_k \geq \dfrac{D_{\xi_2}}{D_{\xi_1}} \\ 0 & : \Psi_k < \dfrac{D_{\xi_2}}{D_{\xi_1}} \end{cases}. \tag{4.61}$$

A critical value of the VoF is $\Psi_k = \dfrac{D_{\xi_2}}{D_{\xi_1}}$, because for this value the interface is passing through the vertex $\boldsymbol{\xi}^A$, see Figure 4.8. Using 4.61, 4.27 can be written as:

$$\Psi_k = \frac{(\tilde{\phi}(\mathbf{0}))^2}{D_{\xi_1} D_{\xi_2}} - S_p^{'}(\Psi_k)\frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^2}{D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}. \tag{4.62}$$

Define the coefficients $c_{1,2}^{'}$ as:

$$c_1^{'} = \frac{1}{D_{\xi_1} D_{\xi_2}}, \qquad c_2^{'} = \frac{-S_p^{'}(\Psi_k)}{D_{\xi_2}(D_{\xi_1} - D_{\xi_2})}. \tag{4.63}$$

Now $\tilde{\phi}(\mathbf{0})$ is the unique root of the following quadratic equation:

$$(c_1^{'} + c_2^{'})(\tilde{\phi}^{n+1}(\mathbf{0}))^2 + 2c_2^{'} D_{\xi_2}(\tilde{\phi}^{n+1}(\mathbf{0})) + c_2^{'} D_{\xi_2}^2 - \Psi_k = 0, \tag{4.64}$$
$$\forall\ \Psi_k \in [0, 1].$$

within the interval $[-D_{\xi_1}, 0]$. This unique root is given by:

$$\tilde{\phi}(\mathbf{0}) = -(2c_2^{'}D_{\xi_2}) - \frac{\sqrt{(2c_2^{'}D_{\xi_2})^2 - 4(c_1^{'}+c_2^{'})(c_2^{'}D_{\xi_2}^2 - \Psi_k)}}{2(c_1^{'}+c_2^{'})}. \tag{4.65}$$

The inverse function $g(\Phi_k, \nabla\Phi_k)$ can now be defined as:

$$\Phi_k = g(\Psi_k, \nabla\Phi_k) : [0,1] \rightarrow \left[\frac{-2D_{\xi_1}+D_{\xi_2}}{3}, \frac{D_{\xi_1}+D_{\xi_2}}{3}\right], \tag{4.66}$$

$$g(\Psi_k, \nabla\Phi_k) = -(2c_2^{'}D_{\xi_2}) - \frac{\sqrt{(2c_2^{'}D_{\xi_2})^2 - 4(c_1^{'}+c_2^{'})(c_2^{'}D_{\xi_2}^2 - \Psi_k)}}{2(c_1^{'}+c_2^{'})}) + \frac{D_{\xi_1}+D_{\xi_2}}{3}.$$

Where, $[D_{\xi_1} \quad D_{\xi_2}]^T = J\nabla\Phi_k$. The conversion of LS to VoF depends on the value and gradient of the LS at the center of the control volume. Key difference between the original MCLS, and the proposed algorithm is that in the latter, the discretisation of the LS field with a discontinuous Galerkin method leads to a *local*, element-wise definition of the LS field. This means that the LS field can be corrected locally, contrary to the original MCLS algorithm where an iterative procedure is used to have both Φ_k^{n+1} and $\nabla\Phi_k^{n+1}$ comply with 4.58. This is a major improvement with respect to efficiency and robustness in comparison with the original MCLS algorithm.

4.8 Computational cost of the mass-conserving correction

Accurately mass-conserving advection of the VoF field on a discretisation of general unstructured control volumes is very challenging but necessary in many but certainly not all applications. Incorporating this correction makes the algorithm significantly more costly than a 'pure' LS approach, but comparable in cost to algorithms that are based on an explicit reconstruction of the interface. The Eulerian-Lagrangian approach of advection and the required 'clipping' algorithm are computationally intensive. However, these 'clipping' type of algorithms are used extensively in computer graphics and under continuous development. Furthermore, they are highly parallelizable and readily portable to GPU architectures. Alternative algorithms, for example based on the use of compressive schemes used for modified LS formulations are currently investigated. The use of a discontinuous Galerkin discretisation and the resulting locality of the mass conserving correction lead to a significant gain in efficiency with respect to the original formulation of the MCLS method.

4.9 Test Cases

The key characteristics of the method that need to be validated are the mass conservation property and the accuracy of the LS field. Therefore, three test cases are selected based on different velocity fields; constant, linear and nonlinear. The test cases chosen are the translation of a circular interface with a constant velocity, the rotation of a circular interface around the center of the domain with linear velocity and the very popular single reverse vortex test case [49, 60]. These test cases are sufficient to demonstrate both the mass conservation properties and accuracy of the method. Apart from these test cases, the accuracy of the conversion between the LS field and the VoF field is considered separately using the VoF function and its inverse. Also the initial mass error that is incurred due to the piecewise linear representation of the interface is analysed. This test case is referred to as *interface back and forth reconstruction.* We define the mass error E_M^n and the discrete L_2 error in the LS field $E_\Psi(t^n)$ as

$$E_M^n = M^n - M_{\text{exact}}(t^n), M^n = \sum_{k=1}^{K} \Psi_k^n |\Omega_k|, \tag{4.67}$$

$$E_\Phi(t^n) = \sqrt{\frac{\sum_{k=1}^{K} \left(\Phi_K^n - \Phi(\boldsymbol{X}_k, t^n)_{\text{exact}}\right)^2}{K}},$$

where $M_{\text{exact}}(t^n)$ and $\Phi(\boldsymbol{X}_k, t^n)_{\text{exact}}$ are the exact area contained within the interface and the exact LS field, respectively. The order of $E_\Phi(t^n)$ is estimated through Richardson extrapolation, on a sequence of unstructured meshes. We expect to find second order convergence of both $E_\Phi(t^n)$ and E_M^n, because of the use of linear basis functions for the LS field and the use of the linearized LS field in the conversion to VoF. The VoF advection scheme is mass conserving to machine precision, so apart from an initialization error, the mass loss should be virtually negligible over the duration of the simulations. We will only focus on the spatial accuracy of the discretisation. In each test case all quantities are made dimensionless by introducing reference length and time scales equal to 1 m and 1 s, respectively.

4.9.1 Conversion between level-set and volume of fluid representation

This test case is considered to demonstrate the accuracy of the conversion of the LS field to the VoF field and vice versa using 4.35 and its inverse 4.66: the VoF field is converted to the LS field using Equation 4.64, without

Characteristic meshwidth h	M^n	$\lvert E_M \rvert$	Order
3.64e-02	1.2456e-01	1.10008e-03	—
2.43e-02	1.2515e-01	5.0951e-04	2.16
1.67e-02	1.2540e-01	2.5679e-04	1.98
1.02e-02	1.2556e-01	9.4813e-05	2.71

Table 4.1: Accuracy analysis of the interface back-and-forth conversion test case. Initialization with the exact level-set field introduces a mass error that converges with second order accuracy.

involving any advection of the interface. Naturally, inter-conversion of the LS field and the VoF field without loss of mass is a prerequisite for the algorithm to achieve (nearly) exact mass conservation. This test shows that the error induced by the inter-conversion is negligible. A circular fluid region is considered with radius 0.2 in a domain of size $\Omega_D = [0,1] \times [0,1]$. The initial condition for the LS field is given as:

$$\Phi(\boldsymbol{x},0) = \tfrac{1}{25} - (x_1 - \tfrac{1}{2})^2 - (x_2 - \tfrac{1}{2})^2, \quad \boldsymbol{x} \in \Omega_D. \tag{4.68}$$

The corresponding VoF field is given by:

$$\Psi(\boldsymbol{x},0) = \left\{ \begin{array}{ll} 1, & (x_1 - \frac{1}{2})^2 + (x_2 - \frac{1}{2})^2 - \frac{1}{25} < 0, \\ 0, & (x_1 - \frac{1}{2})^2 + (x_2 - \frac{1}{2})^2 - \frac{1}{25} > 0. \end{array} \right. , \quad \boldsymbol{x} \in \Omega_D. \tag{4.69}$$

Furthermore, if we consider $Phase_1$ to be the fluid of interest, the total area of the domain of interest is $M_{\text{exact}} = \frac{\pi}{25}$. The interface is represented by the zero-contour of the LS field and the interface normal points outwards from the negative LS region. Firstly, the LS field is converted into the VoF field using 4.32. Secondly, the obtained VoF field is converted back to the LS field using 4.66. After that the difference between the LS field before and after the conversion is evaluated for all mixed cells. We found that the LS field remains unchanged upto machine precision.

In the MCLS the interface is approximated by a set of line segments, because of the fact that the conversion between LS and VoF is based on the linearized LS field. Depending on the procedure for the initialization, this approximation will incur an error in the mass(area) when the LS field is initialized exactly, or an error in the LS field when the VoF field is initialized exactly. We consider 4.68 as initial condition for the LS field. In Table 4.1 E_M is presented for four different characteristic mesh widths $h = \sqrt{\Sigma |\Omega_k| / N}$. Clearly, $E_M = \mathcal{O}(h^2)$, and the initial error strongly dominates the error incurred by the back-and-forth conversion. Alternatively, the LS field can be initialized by conversion of the *exact VoF field* 4.69. This approach will incur an error in the LS field as shown in Figure 4.14, where the zero level countour of the LS field initialized with 4.68 is compared with the LS field converted from 4.69. Because the conversion between the two fields is based on the linearized LS field, the conversion induces a small movement of the interface, which can be shown to decrease with second order upon grid refinement. Clearly, to achieve nearly exact mass conservation, it is essential to initialize the LS field by converting the initial condition for the VoF field.

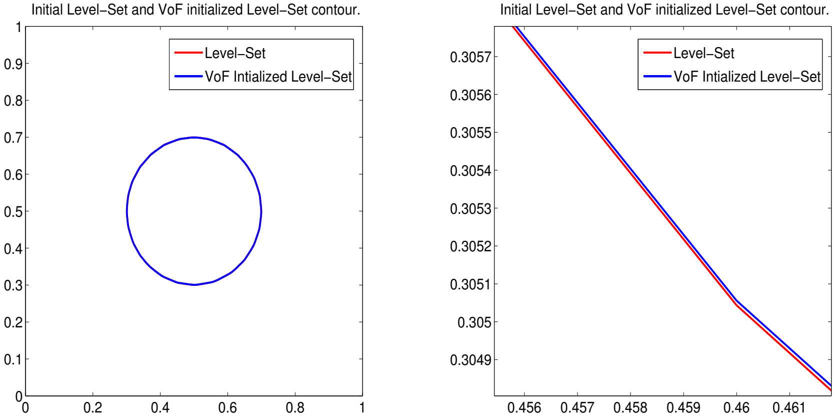


Figure 4.14: Initialization with the exact volume-of-fluid field leads to a slightly different initial position of the interface but eliminates the mass error incurred by initialization with the exact level-set field.

4.9.2 Translation of circular region

In this test case the translation of a circular interface of radius 0.15 is considered in a domain of size $\Omega_D = [0,1] \times [0,1]$. Initially, the interface $\phi(\boldsymbol{x},0) = 0$ is centered at $(0.5, 0.2)^T$. The LS and VoF fields are advected with a constant velocity field $\boldsymbol{u}(\boldsymbol{x}) = (0, 0.1)^T$ for $0 < t \leq T = 5$. The exact LS field at the final time is a circular region with unchanged radius but centered at $(0.5, 0.7)^T$. In Table 4.2 the average mass error $\overline{E_M^n}$ and the error of the LS field $E_\Phi(T)$ are presented for three different mesh sizes. The results indicate that the solution of the LS field converges with second order accuracy.

Characteristic meshwidth h	$\overline{E_M^n}$	$E_\Phi(T)$	Order of $E_\Psi(T)$
3.64e-02	2.6835E-15	8.0531E-04	—
2.43e-02	2.2830E-15	5.1876E-04	1.55
1.67e-02	8.2976E-16	2.3053E-04	2.25

Table 4.2: Average mass error $\overline{E_M^n}$ and error in the level-set field $E_\Phi(T)$ for the translation of a circular interface.

4.9.3 Rotation of circular region around the center of the domain

The third test case concerns the rotation of an initially circular interface. Consider a circular region of radius 0.15 in a domain of size $\Omega_D = [0,1] \times [0,1]$. Initially, the interface $\phi(\boldsymbol{x}, 0) = 0$ is centered around $(0.5, 0.75)^T$. The LS and VoF fields are advected with a divergence free linear velocity field $\boldsymbol{u}(\boldsymbol{x})$ defined as:

$$\boldsymbol{u}(\boldsymbol{x}) = (x_1 - 0.5, -x_2 + 0.5)^T, \ \boldsymbol{x} \in \Omega_D. \tag{4.70}$$

for $0 < t \leq T = 2\pi$. At $t = T$ the LS field is compared with the initial LS field as the interface should return to its initial position after one complete revolution.

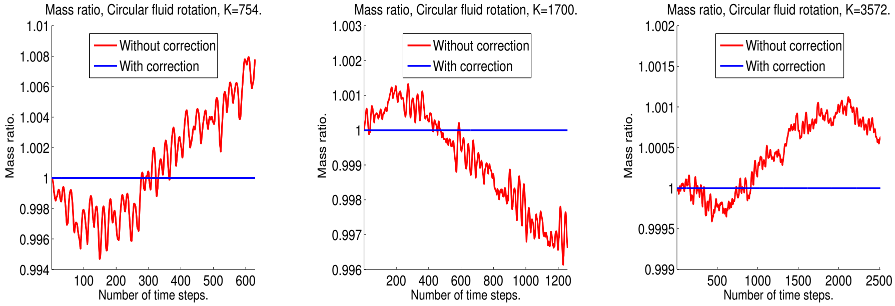


Figure 4.15: Time history of mass error for the circular fluid rotation case.

Figure 4.15 shows the time history of the mass error during one full rotation for three different mesh sizes. In Table 4.3 the average mass error $\overline{E_M^n}$ and the error of the LS field $E_\Phi(T)$ are presented. The solution of the LS field converges with second order accuracy.

Characteristic mesh width h	$\overline{E_M^n}$	$E_\Psi(T)$	Order of $E_\Psi(T)$
3.64e-02	1.6199E-14	6.6142E-03	—
2.43e-02	4.2944E-16	2.2104E-03	2.99
1.67e-02	1.7417E-15	1.2526E-03	1.77

Table 4.3: Average mass error $\overline{E_M^n}$ and error in the level-set field $E_\Phi(T)$ for the rotation of a circular interface.

4.9.4 The reverse vortex test case

The reverse vortex or single reverse vortex test case is one of the more challenging tests for models for immiscible two-phase flow. The reason for this is the severe stretching and deformation of the convected interface caused by the nonlinear velocity field during the advection which makes accurate mass conservation challenging.

Consider an initially circular fluid region of radius 0.15 in a domain of size $\Omega_D = [0,1] \times [0,1]$. Initially, the center of the fluid is at $(0.5, 0.75)$. This circular region is advected with a divergence free nonlinear velocity field $\boldsymbol{u}(\boldsymbol{x},t)$ defined as:

$$\boldsymbol{u}(\boldsymbol{x},t) = (\sin^2(\pi x_1)\sin(2\pi x_2), -\sin^2(\pi x_2)\sin(2\pi x_1))^T \cos(\tfrac{\pi t}{T}), \qquad (4.71)$$
$$\boldsymbol{x} \in \Omega_D,\ 0 \le t \le T.$$

This definition of the velocity field implies that the interface should return to its original position at time $t = T$, which is also the final time. The value of T is set to 2.

The evolution of the interface is shown in Figure 4.17 for three different mesh widths, together with the exact initial and final position of the interface.

Figure 4.16 shows the time history of the mass error during one full rotation for three different mesh sizes. In Table 4.4 the average mass error $\overline{E_M^n}$ and the error of the LS field $E_\Phi(T)$ are presented. The solution of the LS field converges with second order accuracy, while even for this nonlinear, velocity field mass is conserved (nearly) exactly.

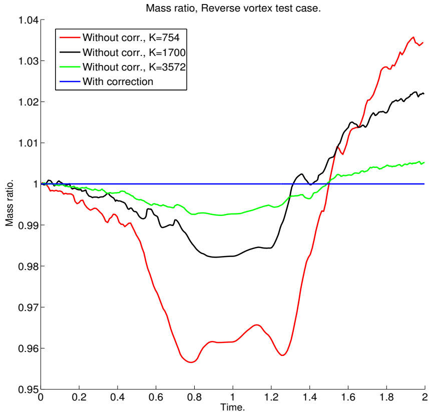


Figure 4.16: Time history of mass error for the reverse vortex test case.

Characteristic mesh width h	$\overline{E_M^n}$	$E_\Psi(T)$	Order of $E_\Psi(T)$
3.64e-02	1.48167E-14	5.6716E-3	—
2.43e-02	3.3508E-16	1.8284E-3	3.101
1.67e-02	4.7809E-16	1.0951E-3	1.669

Table 4.4: Average mass error $\overline{E_M^n}$ and error in the level-set field $E_\Phi(T)$ for the reverse vortex test case.

4.10 Conclusions

We have developed a (nearly) exactly mass-conserving method for the simulation of immiscible incompressible two phase flow in geometrically intricate two-dimensional domains. The method developed is the extension of the MCLS method towards unstructured triangular grids. The VoF function and the inverse function derived for a triangular mesh are very simple, robust and efficient to evaluate. Due to the use of a discontinuous Galerkin discretisation method for the LS field, the mass-conserving correction can be applied locally in each element. This approach is significantly more efficient and robust than the original MCLS formulation. Numerical experiments indicate the LS field converges with second order accuracy in space and mass is conserved up to machine precision.

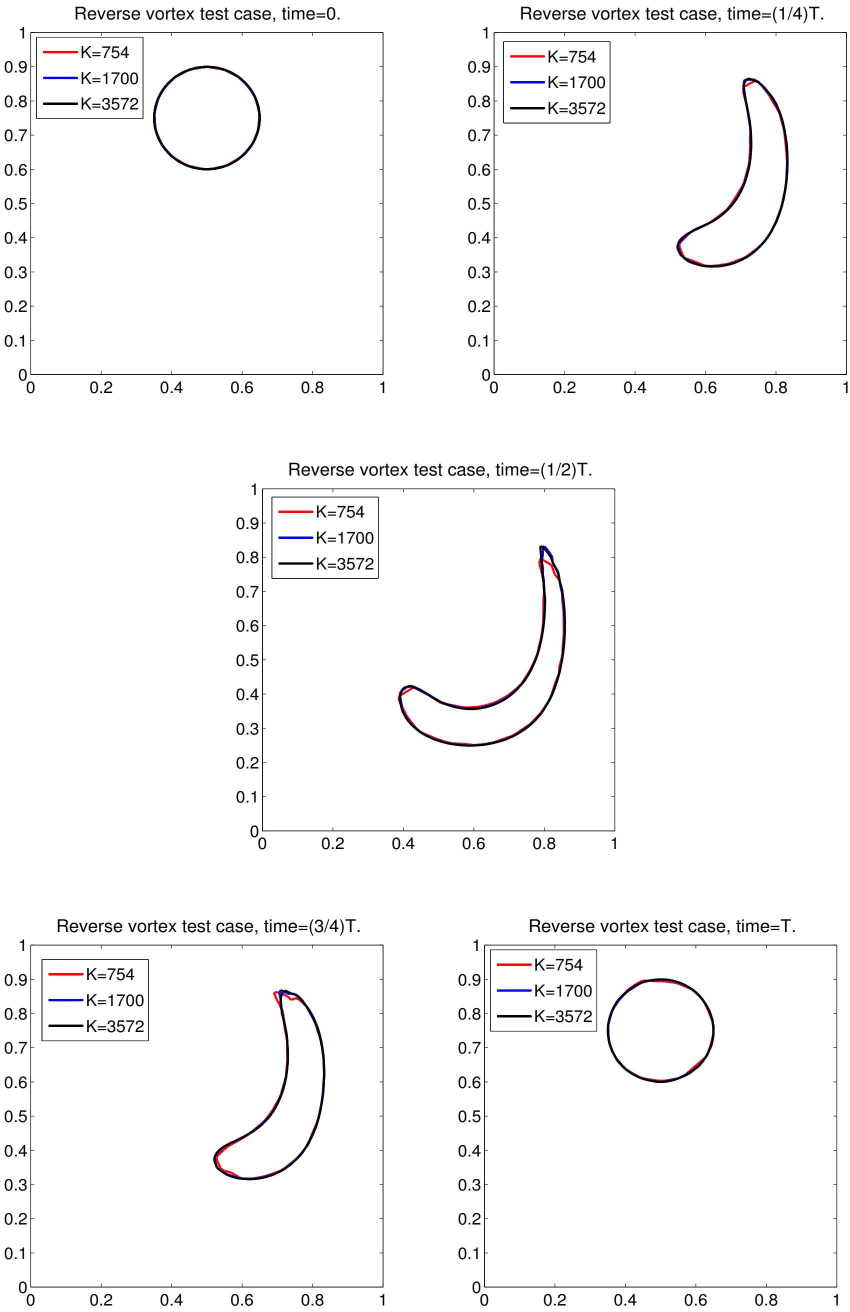


Figure 4.17: The evolution of the interface position for the reverse vortex test case.

CHAPTER 5

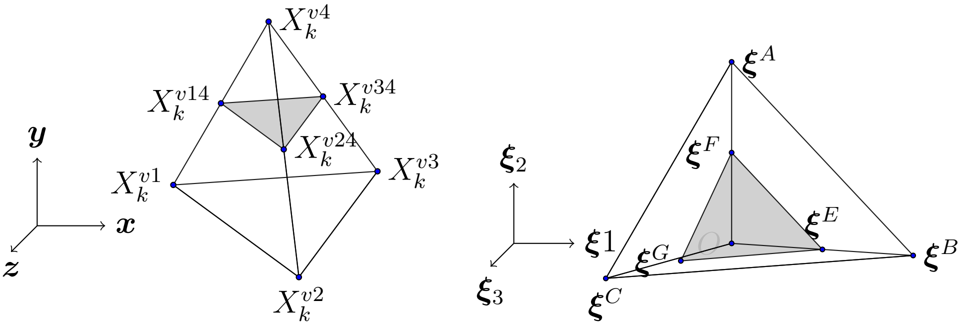
Extension to tetrahedron control volumes

5.1 Introduction

In Chapter 4, we have presented the proposed Mass-Conserving Level-Set method along with numerical results based on 2D triangular mesh. Based on these results it is worthwhile to developed all the ingredients of the proposed method for 3D tetrahedral control volumes. This will pave the way to demonstrate the capabilities of the proposed method for more complex domains. In the light of the objective in this chapter, we have presented a 3D Volume of Fluid function and its inverse function is presented.

5.2 Volume of Fluid function in 3D

Our approach is to use a geometric construction to find the function $\Psi(X_k) = f(\phi_k, \nabla\phi_k)$ that can link VoF to the LS field by an analytic function. The interface LS function ϕ_k is segment of plane. Therefore, $f(\phi_k, \nabla\phi_k)$ is the relative volume of the polygon bounded by the faces of the tetrahedral and the plane $\phi_k = 0$. The polygon can be either tetrahedral or quadrilateral or even reduced to a single point, depending on the intersection of the interface plane with the tetrahedron, Ω_k. An efficient evaluation of the function $f(\phi_k, \nabla\phi_k)$ requires a formulation that considers all possible situations as uniformly as possible.



(a) Control volume (Ω_k).   (b) Barycentric coordinates.

Figure 5.1: Tetrahedron control volume.

Such a formulation is realized when the common vertex of those edges that are intersected by the interface plane is mapped to the origin. This is shown in Fig. 5.1.

This case is considered in Fig. 5.1, $\boldsymbol{X}_k^{v4}$ is this vertex, because it is a common vertex of the edges $\overline{\boldsymbol{X}_k^{v1}\boldsymbol{X}_k^{v4}}$, $\overline{\boldsymbol{X}_k^{v2}\boldsymbol{X}_k^{v4}}$ and $\overline{\boldsymbol{X}_k^{v3}\boldsymbol{X}_k^{v4}}$, as these edges intersect the interface at points $\boldsymbol{X}_k^{v14}$, $\boldsymbol{X}_k^{v24}$ and $\boldsymbol{X}_k^{v34}$, respectively. The barycentric coordinate transformation for the above mapping, assuming $\boldsymbol{X}_k^{v4}$ is the common vertex, can be written as:

$$\boldsymbol{x}(\boldsymbol{\xi}) = (\boldsymbol{X}_k^{v1} - \boldsymbol{X}_k^{v4})\xi_1 + (\boldsymbol{X}_k^{v2} - \boldsymbol{X}_k^{v4})\xi_2 + (\boldsymbol{X}_k^{v3} - \boldsymbol{X}_k^{v4})\xi_3 + \boldsymbol{X}_k^{v4}. \quad (5.1)$$

The linear transformation $\boldsymbol{x}(\boldsymbol{\xi})$ is bijective for a non-degenerate tetrahedron Ω_k: a tetrahedron for which no vertices coincide and/or faces are aligned. Define the (non singular) Jacobian of this mapping as $J = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}$. The linear LS function $\tilde{\phi}(\boldsymbol{\xi})$ can now be formulated as:

$$\tilde{\phi}(\boldsymbol{\xi}) = \Phi(\boldsymbol{x}(\boldsymbol{\xi}_k)) + \left.\frac{\partial \Phi}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=\boldsymbol{\xi}_k} (\boldsymbol{\xi} - \boldsymbol{\xi}_{\boldsymbol{k}}), \qquad \frac{\partial \Phi}{\partial \boldsymbol{\xi}} = J\nabla\Phi, \quad (5.2)$$

where $\boldsymbol{\xi}_k = \boldsymbol{\xi}(\boldsymbol{X}_k)$. Consider Fig. 5.1, the common vertex $\boldsymbol{X}_k^{v4}$ is mapped to the origin and the other three vertices are mapped to points on the axes on logical space. It is important to note that vertices $\boldsymbol{X}_k^{v1}$, $\boldsymbol{X}_k^{v2}$ and $\boldsymbol{X}_k^{v3}$ can be mapped to any of the vertices in logical space, except the origin. The other three vertices in logical space are located at $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. Furthermore, the interface is represented by a plane formed by the points $\boldsymbol{X}_k^{v14}$,$\boldsymbol{X}_k^{v24}$ and $\boldsymbol{X}_k^{v34}$. Our target is to compute the volume of the sub tetrahedron $\blacktriangle\boldsymbol{X}_k^{v14}\boldsymbol{X}_k^{v24}\boldsymbol{X}_k^{v34}\boldsymbol{X}_k^{v4}$ with the help of the linear form of the LS field derived in Equation 5.2.

Let us assume that the vertices $\boldsymbol{X}_k^{v1}$, $\boldsymbol{X}_k^{v2}$ and $\boldsymbol{X}_k^{v3}$ are mapped to $\boldsymbol{\xi}^B = (1,0,0)$, $\boldsymbol{\xi}^A = (0,1,0)$ and $\boldsymbol{\xi}^C = (0,0,1)$ respectively, in a logical space. Therefore, points $\boldsymbol{X}_k^{14}$, $\boldsymbol{X}_k^{24}$, $\boldsymbol{X}_k^{34}$ are mapped to $\boldsymbol{\xi}^E = (\xi_1^E, 0, 0)$, $\boldsymbol{\xi}^F = (0, \xi_2^F, 0)$ and $\boldsymbol{\xi}^G = (0,0,\xi_3^G)$, respectively. This is shown in Fig. 5.1. In order to compute the volume of the region $\boldsymbol{X}_k^{v14}\boldsymbol{X}_k^{v24}\boldsymbol{X}_k^{v34}\boldsymbol{X}_k^{v4}$, the points $\boldsymbol{\xi}^E$, $\boldsymbol{\xi}^F$ and $\boldsymbol{\xi}^G$ should be known.

5.2.1 Coordinates of the point ξ^E

The logical space coordinates of the cell centroid $\boldsymbol{\xi}_k$ are by virtue of the barycentric coordinate transformation always given by $\boldsymbol{\xi}_k = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^T$. This means 5.2 can be written as;

$$\tilde{\phi}(\boldsymbol{\xi}) = \Phi_k + \frac{\partial \Phi}{\partial \xi_1}\left(\xi_1 - \frac{1}{4}\right) + \frac{\partial \Phi}{\partial \xi_2}\left(\xi_2 - \frac{1}{4}\right) + \frac{\partial \Phi}{\partial \xi_3}\left(\xi_3 - \frac{1}{4}\right). \tag{5.3}$$

Substitution of $\boldsymbol{\xi}^E$ in 5.3 leads to:

$$\tilde{\phi}(\boldsymbol{\xi}^E) = \Phi_k + \frac{\partial \Phi}{\partial \xi_1}\left(\xi_1^E - \frac{1}{4}\right) - \left(\frac{1}{4}\right)\frac{\partial \Phi}{\partial \xi_2} - \left(\frac{1}{4}\right)\frac{\partial \Phi}{\partial \xi_3} = 0. \tag{5.4}$$

The right-hand side of 5.4 is zero, because the interface passes through the point $\boldsymbol{\xi}^E$. Evaluation of 5.3 in $\boldsymbol{\xi}(\boldsymbol{X}_k^{v4}) = \mathbf{0}$ gives:

$$\tilde{\phi}(\mathbf{0}) = \Phi_k - \left(\frac{1}{4}\right)\frac{\partial \Phi}{\partial \xi_1} - \left(\frac{1}{4}\right)\frac{\partial \Phi}{\partial \xi_2} - \left(\frac{1}{4}\right)\frac{\partial \Phi}{\partial \xi_3}. \tag{5.5}$$

Solving 5.4 and 5.5 to find ξ_1^E leads to:

$$\xi_1^E = \left| -\frac{\tilde{\phi}(\mathbf{0})}{D_{\xi_1}} \right|, \tag{5.6}$$

where, $D_{\xi_1} = \frac{\partial \Phi}{\partial \xi_1}$. Therefore, the coordinates of the point $\boldsymbol{\xi}^E = \left(\frac{\phi(\mathbf{0})}{D_{\xi_1}}, 0, 0\right)$.

5.2.2 Coordinates of the points ξ^F and ξ^G

Based on the computation for the point $\boldsymbol{\xi}^E$, a similar computation can be done to the other two edges for the points ξ^F and ξ^G, which leads to $\boldsymbol{\xi}^F = \left(0, \frac{\phi(\mathbf{0})}{D_{\xi_2}}, 0\right)$ and $\boldsymbol{\xi}^G = \left(0, 0, \frac{\phi(\mathbf{0})}{D_{\xi_3}}\right)$, respectively.

5.2.3 Evaluation of the VoF from the LS function

We can compute the volume enclosed by the points $\mathbf{0}$, $\boldsymbol{\xi}^E$, $\boldsymbol{\xi}^F$ and $\boldsymbol{\xi}^G$ in logical space. The volume of the enclosed region is denoted by $A_{\boldsymbol{\xi}}$ and is given by:

$$A_{\boldsymbol{\xi}} = \frac{(\tilde{\phi}(\mathbf{0}))^3}{6D_{\xi_1}D_{\xi_2}D_{\xi_3}} \tag{5.7}$$

Equation 5.7 presents the volume of the region enclosed by interface in logical space. The product of $A_{\boldsymbol{\xi}}$ and the Jacobian of the transformation J is equal to the volume enclosed by the points $\boldsymbol{X}_k^{14}$ $\boldsymbol{X}_k^{24}$ $\boldsymbol{X}_k^{34}\boldsymbol{X}_k^{v4}$ in physical space.

The volume of the image of Ω_k in logical space, is $\frac{1}{6}$. Let us call it $A_{\boldsymbol{\xi}}^{\text{total}} = \dfrac{1}{6}$. The VoF fraction in both physical and logical domain can be defined as,

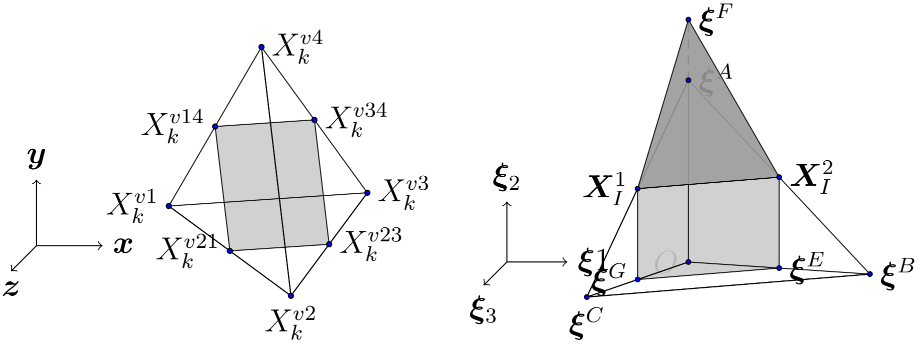
$$\Psi(x_k) \approx f(\phi_k, \nabla\phi_k) = \frac{A_{\boldsymbol{\xi}}}{A_{\boldsymbol{\xi}}^{\text{total}}} = \frac{(\phi(\mathbf{0}))^3}{D_{\xi_1}D_{\xi_2}D_{\xi_3}}. \tag{5.8}$$

This is the volume of fluid fraction formula, based on the LS function and its gradient.

It is important to note by the choice of mapping D_{ξ_1},D_{ξ_2} and D_{ξ_3} cannot be zero. It is possible that the interface is aligned in a computational domain such that it intersect the control volume such that two of its vertices are on one side of the fluid i.e. they have the same sign of the LS value. This situation is shown in the 5.2. In this case a slightly different approach is used to compute teh VoF function and its inverse. This is explained in the following section.

5.2.4 Case: When two vertices are aside

Consider the interface orientation shown in the Figure 5.2. Let us assume that the vertices X_k^{v4} and X_k^{v2} are in the fluid of interest and the other vertices X_k^{v1} and X_k^{v3} are in the other fluid. In this situation, we define a convention that out of the two vertices X_k^{v4} and X_k^{v2} anyone can be mapped to origin provided that it has larger magnitude of the LS and the other one which small magnitude of the LS will be mapped in the $\boldsymbol{\xi_2}$ direction. This convention is only for the simplification and to reduce the other possibilities of the mapping. A similar procedure is used for mapping as done above i.e. for the case where the interface cuts the tetrahedron, such that one vertex is a side and teh rest of three are at the other side. However, the interface intersect the edge in the $\boldsymbol{\xi_2}$ direction, outside the tetrahedron defined in logical mapping. This shown in Figure 5.2. It is quite clear that the volume



(a) Control volume (Ω_k). (b) Barycentric coordinates.

Figure 5.2: Case: where two vertices are aside.

of interest is enclosed by the vertices in logical space, $O, E, F, G, \boldsymbol{\xi}(X_k^{v_2}), X_I^1$ and X_I^2. The points X_I^1 and X_I^2 are the intersection of the interface with the edges formed by $\boldsymbol{\xi}(X_k^{v_2})\boldsymbol{\xi}(X_k^{v_1})$ and $\boldsymbol{\xi}(X_k^{v_2})\boldsymbol{\xi}(X_k^{v_3})$, respectively.

Firstly, the coordinates of the points X_I^1 and X_I^2 are required. To compute these coordinates consider the intersection of the interface with the edges.

5.2.5 For point X_I^1

The linear LS field in the logical space is given as follow,

$$\tilde{\phi}(\boldsymbol{\xi}) = \Phi_k + \frac{\partial \Phi}{\partial \xi_1}\left(\xi_1 - \frac{1}{4}\right) + \frac{\partial \Phi}{\partial \xi_2}\left(\xi_2 - \frac{1}{4}\right) + \frac{\partial \Phi}{\partial \xi_3}\left(\xi_3 - \frac{1}{4}\right). \tag{5.9}$$

The point X_I^1 lies in the $\xi_2\xi_3-$plane. Therefore, $\xi_1 = 0$ and $\xi_2 + \xi_3 = 1$ at the edge. The coordinates of the point $X_I^1(0, \xi_2^{'}, \xi_3^{'})$ and using interface has value zero at the point X_I^1, Equation 5.9 gives,

$$\Phi_k + \frac{\partial \Phi}{\partial \xi_1}\left(-\frac{1}{4}\right) + \frac{\partial \Phi}{\partial \xi_2}\left(\xi_2^{'} - \frac{1}{4}\right) + \frac{\partial \Phi}{\partial \xi_3}\left(\xi_3^{'} - \frac{1}{4}\right) = 0. \tag{5.10}$$

Also, we have $\xi_2^{'} + \xi_3^{'} = 1$ solving these two equation will leads the values of $\xi_2^{'}$ and $\xi_3^{'}$ as follows,

$$\xi_2^{'} = \frac{\tilde{\phi}(\mathbf{0}) + D_{\xi_2}}{D_{\xi_3} - D_{\xi_2}}, \tag{5.11}$$

$$\xi_3^{'} = -\frac{\left(\tilde{\phi}(\mathbf{0}) + D_{\xi_2}\right)}{D_{\xi_3} - D_{\xi_2}}, \tag{5.12}$$

where, $D_{\xi_2} = \dfrac{\partial \Phi}{\partial \xi_2}$, $D_{\xi_3} = \dfrac{\partial \Phi}{\partial \xi_3}$ and

$$\tilde{\phi}(\mathbf{0}) = \Phi_k - \left(\frac{1}{4}\right)\frac{\partial \Phi}{\partial \xi_1} - \left(\frac{1}{4}\right)\frac{\partial \Phi}{\partial \xi_2} - \left(\frac{1}{4}\right)\frac{\partial \Phi}{\partial \xi_3}. \tag{5.13}$$

This defines the coordinates of point X_I^1.

5.2.6 For point X_I^2

A similar procedure is adopted for the point X_I^2. The point X_I^2 lies in the $\xi_2\xi_1-$plane. Therefore, $\xi_3 = 0$ and $\xi_2 + \xi_1 = 1$ at the edge. The coordinates of the point $X_I^2(\xi_1^{'}, \xi_2^{'}, 0)$ and using the fact that interface has value zero at the point X_I^2, Equation 5.9 gives.

$$\Phi_k + \frac{\partial \Phi}{\partial \xi_1}\left(\xi_1^{'} - \frac{1}{4}\right) + \frac{\partial \Phi}{\partial \xi_2}\left(\xi_2^{'} - \frac{1}{4}\right) + \frac{\partial \Phi}{\partial \xi_3}\left(-\frac{1}{4}\right) = 0. \tag{5.14}$$

Also, we have $\xi_2^{'} + \xi_1^{'} = 1$ solving these two equation leads to the values of $\xi_2^{'}$ and $\xi_1^{'}$ as follows,

$$\xi_2^{'} = \frac{\tilde{\phi}(\mathbf{0}) + D_{\xi_2}}{D_{\xi_1} - D_{\xi_2}}, \tag{5.15}$$

$$\xi_1^{'} = -\frac{\left(\tilde{\phi}(\mathbf{0}) + D_{\xi_2}\right)}{D_{\xi_1} - D_{\xi_2}}, \tag{5.16}$$

where, $D_{\xi_1} = \dfrac{\partial \Phi}{\partial \xi_1}$, $D_{\xi_2} = \dfrac{\partial \Phi}{\partial \xi_2}$ and

$$\tilde{\phi}(\mathbf{0}) = \Phi_k - \left(\frac{1}{4}\right)\frac{\partial \Phi}{\partial \xi_1} - \left(\frac{1}{4}\right)\frac{\partial \Phi}{\partial \xi_2} - \left(\frac{1}{4}\right)\frac{\partial \Phi}{\partial \xi_3}. \tag{5.17}$$

This defines the coordinates of point X_I^2. The points X_I^1 and X_I^2 are used to compute the extra volume, explained in the next section.

5.2.7 Extra volume

The VoF fraction given by the VoF function derived in equation 5.7 gives the volume enclosed by the interface and the tetrahedron and also the extra volume defined by the tetrahedron $\boldsymbol{\xi}^F, \boldsymbol{\xi}(X_k^{v2}), \boldsymbol{\xi}(X_I^1)$ and $\boldsymbol{\xi}(X_I^2)$. Let us call the extra volume V_k^{extra}. In order to get correct volume enclosed by the interface in a cell, the volume V_k^{extra} should be subtracted from the volume obtained by VoF function. This can be noted as follows,

$$V_k = \frac{(\tilde{\phi}(\mathbf{0}))^3}{6D_{\xi_1}D_{\xi_2}D_{\xi_3}} - V_k^{extra}. \tag{5.18}$$

Now, V_k^{extra} is needed to be determined and this can be computed by using the coordinates of the points $\boldsymbol{\xi}^F, \boldsymbol{\xi}(X_k^{v2}), \boldsymbol{\xi}(X_I^1)$, all of these points are known and a simple needs to determine defined by these points. This is leads to,

$$V_k^{extra} = \left| \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^3}{6D_{\xi_2}(D_{\xi_3} - D_{\xi_2})(D_{\xi_1} - D_{\xi_2})} \right|. \tag{5.19}$$

So, the volume fraction can be defined as,

$$\Psi_k = \frac{(\tilde{\phi}(\mathbf{0}))^3}{D_{\xi_1}D_{\xi_2}D_{\xi_3}} - \frac{(\tilde{\phi}(\mathbf{0}) + D_{\xi_2})^3}{(D_{\xi_1} - D_{\xi_2})D_{\xi_2}(D_{\xi_3} - D_{\xi_2})}, \tag{5.20}$$

where, $\tilde{\phi}(\mathbf{0})$ is the value of the LS at the origin in logical space and defined as;

$$\tilde{\phi}(\mathbf{0}) = \Phi_k - \left(\frac{D_{\xi_1} + D_{\xi_2} + D_{\xi_3}}{4} \right). \tag{5.21}$$

The volume fraction equation can be further simplified to a simple cubic polynomial equation for $\tilde{\phi}(\mathbf{0})$. To do this let us define two coefficients, $A = \dfrac{1}{D_{\xi_1}D_{\xi_2}D_{\xi_3}}$ and $B = \dfrac{1}{(D_{\xi_1} - D_{\xi_2})D_{\xi_2}(D_{\xi_3} - D_{\xi_2})}$. This will give us volume fraction function in the simplest form as,

$$\Psi_k = a\tilde{\phi}(\mathbf{0})^3 + b\tilde{\phi}(\mathbf{0})^2 + c\tilde{\phi}(\mathbf{0}) + d, \tag{5.22}$$

where, the coefficients are $a = A - B$, $b = -3BD_{\xi_2}$, $c = -3BD_{\xi_2}^2$ and $d = -BD_{\xi_2}^2$. This cubic polynomial representation of the VoF fraction is very convenient in use and it is also used to derived the inverse function, explained in the next section.

5.3 Inverse function in 3D

In this section derivation of the inverse function is presented $\Phi_k = g(\Psi_k, \Delta\Phi_k)$. This function is used to correct the LS field based on the given VoF fraction. This VoF fraction is obtained by advecting the VoF field parallel to the LS field. In other words the inverse function is the analytic way to coupled the LS field and the VoF field. The main objective of this coupling is to find the cell centroid value of the LS Φ_k^c that corresponds to the given VoF fraction Ψ_k^c.

Consider Equation 5.22 and assume that the interface is corrected in the direction of the normal to the interface. Therefore, we keep $D_{\xi_1}^{n+1}, D_{\xi_2}^{n+1}$ and $D_{\xi_3}^{n+1}$ as they are at time t_{n+1}. Therefore, the inverse function can be defined as,

$$a\tilde{\phi}(\mathbf{0})^3 + b\tilde{\phi}(\mathbf{0})^2 + c\tilde{\phi}(\mathbf{0}) + d - \Psi_k^c = 0. \tag{5.23}$$

Above equation does not define the inverse function completely. First the value of $\tilde{\phi}(\mathbf{0})$ is computed for the given Ψ_k^c by solving cubic equation 5.23. Then the cell centroid value of the LS can be obtained as,

$$\Phi_k^c = \tilde{\phi}(\mathbf{0}) + \left(\frac{D_{\xi_1}^{n+1} + D_{\xi_2}^{n+1} + D_{\xi_3}^{n+1}}{4}\right). \tag{5.24}$$

This will give a corrected value of the LS field at the cell centroid. Here, corrected value of the LS field means that centroid value of the LS now corresponds to the advected VoF fraction. This whole step is simple and straight-forward, except that how to solve equation 5.23. There are two choices either solve it with cubic root finding analytic method or solve it numerically, like Newtons-Raphson iterative method for the non-linear equations. We have selected the iterative method to demonstrate the working of the inverse function, however, any analytic method for finding cubic roots of the polynomial is applicable. In an iterative method initial guess is important and we have observed that, if we select the starting value equal to the lineraized value at the origin based on the non-corrected LS field and applying Newtons-Raphson method it appears to always converge to the correct root value, which makes sure that the interface remains within the element and also corresponds to the correct VoF fraction. Let us assume that the starting value is $\phi^{(1)}$, then the Newton-Raphson algorithm is presented as follows,

Algorithm 1 The inverse function solution, using the Newton-Raphson method

```
Require: $\phi^{(1)}, a, b, c, d, tol, \epsilon$
  while $\epsilon \geq tol$ do
    $\phi^{(i+1)} = \phi^{(i)} - \frac{a(\phi^{(i)})^3 + b(\phi^{(i)})^2 + c\phi^{(i)} + d}{3a(\phi^{(i)})^2 + 2b\phi^{(i)} + c}$
    $\epsilon = |\phi^{(i+1)} - \phi^{(i)}|$
    $\phi^{(i+1)} = \phi^{(i)}$
  end while
```

This has completed our discussion and overview of the MCLS method with respect to the tetrahedron control volumes. In the next section we have only presented the result to demonstrate the working of VoF function and the inverse function for the tetrahedron control volumes.

5.4 Test Case: The 3D back and forth interface reconstruction

To demonstrate the accuracy of the conversion of the LS field to the VoF field and then the VoF field back to the LS field using the inverse function. We have designed a test case in which in first step we convert the LS field into the VoF field, using VoF function, equation 5.22 and in the second step, the converted VoF field is again converted back to the LS set field, using the inverse function 5.23. This is a basic step of the MCLS method to obtained the mass conservative LS field. However, in practice the VoF field in second step is obtained by advecting VoF filed along the LS field. In this test case no advection of the interface is involved. Naturally, inter-conversion of the LS field and the VoF field without loss of mass is a prerequisite for the algorithm to achieve (nearly) exact mass conservation. This test shows that the absolute error (Err.) induced by the inter-conversion is nearly zero. Steps involved in this test can expressed by the following equations,

$$\Phi_k \rightarrow \underbrace{f(\Phi_k, \nabla\Phi_k)}_{\text{Step-1: VOF function}} \rightarrow \Psi_k \rightarrow \underbrace{g(\Psi_k, \nabla\Phi_k)}_{\text{Step-2: Inverse function}} \rightarrow \Phi_k, \tag{5.25}$$

$$Err. = |\Phi_k - \Phi_k| \approx 0.$$

In the light of the above test case, let us consider the interface topology of the shape Zalesak's slotted sphere. The interface is centered at $\boldsymbol{x^c} =$

$(0.5, 0.5, 0.5)$ of the unit cube, i.e. $\Omega_D = [0,1] \times [0,1] \times [0,1]$. The LS field is given as Φ_k, now we have applied the first step-1 and after that step-2.

It is observed that the error between the LS field before and after conversion is zero upto machine precision. In Figure 5.3 the interface contour is plotted after correction. It is observed that the developed VoF function and the inverse function work efficiently and induced no error upto machine precision. This test suggests that apart from the error induced due to the interface advection with the flow field, there is no error due to inter conversion of the LS field. This test is sufficient to conclude that the developed VoF function and the inverse function are conservative, once applied with a complete advection scheme for a 3D tetrahedron.

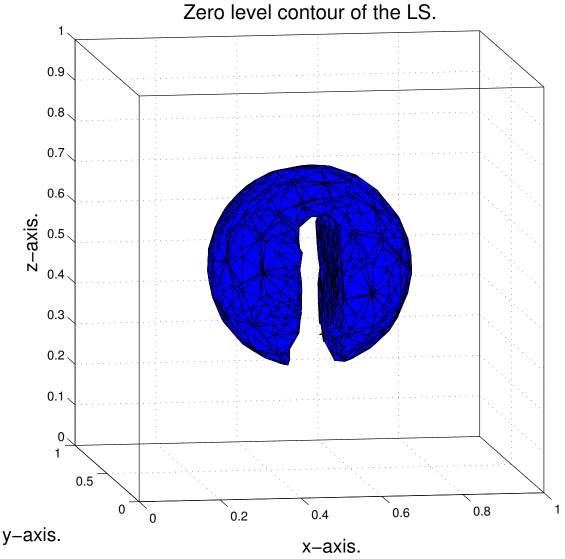


Figure 5.3: The interface zero-contour after correction for the Zalesak's slotted sphere.

5.5 Conclusions

In this chapter the 3D extension of the MCLS method is presented. The developed VoF function and the inverse for the tetrahedron control volumes are mass conservative. The mass-conservation correction to the LS field is very trivial and easy to implement. This seems promising for the complete algorithm once coupled with the advection of the LS field and the VoF field.

CHAPTER 6

The Modified Level-Set method

6.1 Introduction

In the last two chapters the extension of the MCLS for Cartesian control volumes to triangular and tetrahedral control volumes has been presented. The hybrid formulation based on a congruent LS and VoF field combines the advantages of the LS formulation with accurate mass conservation. However, the Eulerian-Lagrangian approach to advect the VoF field is quite involved and will be even more so when applied in three spatial dimensions.

For those cases for which the requirements of very strict mass conservation can be relaxed an alternative approach has become very popular: The modified Level-Set method introduced in the milestone paper by Olsson and Kreis [15, 37, 39].

The concept is based on the use of a level-set function that is C^∞ continuous but closely resembles the color function [add reference to equation where this is defined]. For any continuous function $\phi(\boldsymbol{x},t)$ the 'mass' M in the domain is defined as the area of the region enclosed by the $\phi(\boldsymbol{x},t)=0$ LS:

$$M(t)=\int_\Omega H(\phi(\boldsymbol{x},t))d\Omega. \tag{6.1}$$

Therefore, if the field $\phi(\boldsymbol{x},t)$ is conservatively advanced in time this does not guarantee conservation of mass unless either:

- Mass conservation is explicitly enforced after each advance in time. (e.g. as in the MCLS algorithm),
- $\|\phi(\boldsymbol{x},t) - H(\phi(\boldsymbol{x},t))\| < \epsilon$.

It is the latter approach that is followed in the modified LS method. The 'true' color function is replaced by a mollified color function $\phi_m(\boldsymbol{x},t)$, where the change of $\phi_m(\boldsymbol{x},t)$ from the one to the other value of the color function takes place over a nonzero but small distance. How exactly this change occurs is arbitrary, but smoothness of the transition will improve the accuracy of the interface location and efficiency of the interface capturing model.

Clearly, the closer the modified LS function resembles the true color function, the more accurately mass is conserved. However, simultaneously, accurate discretisation of the conservation equation for such a rapidly changing and highly nonlinear function becomes more and more challenging.

The ability to easily accomodate changes in the topology of the interface necessary to model coalesence and break-up is shared by the LS and the modified LS. Also the evolution of both is described by the same simple linear advection equation, which results in an efficient and robust algorithm. The discretisation of this advection equation is not complicated by the use of an unstructured tesselation of triangular or tetrahedral control volumes, instead of a set of Cartesian control volumes. Curvature and interface normal vector can be reconstructed from the modified LS field in exactly the same way as for the LS field.

Up till recently, the finite volume methods and standard limited convection schemes were used to discretise the modified LS equation. In [41] the use of the discontinuous Galerkin method was introduced in this context. The aim of this research is to investigate the potential of the modified LS method in terms of accuracy of the interface location and mass conservation when state-of-the-art spatial discretisation techniques are used. Additionally, an artificial compressive velocity field is introduced to improve the interface sharpness when it is aligned with the flow direction. This technique is used in some finite volume schemes [51], but is now introduced in the context of a discontinuous Galerkin discretisation of the modified LS equation. To circumvent the occurence of oscillations in the modified LS field in the vicinity of the interface first and second order formulations of the limiter proposed by Kuzmin et al. [31,32] are used. The application of the limiter succesfully suppresses any oscillations, but also leads to smearing out of the interface. The latter can now be corrected by the introduction of the artificial compressive velocity field. The key characteristics of the proposed algorithm can be summarized as:

- The modified LS field is approximated by either a first or second degree piecewise polynomial function on a set of non overlapping triangular control volumes.
- To avoid oscillations in the solution of the LS field near the interface due to the nearly discontinuous behaviour a limiter is applied after the advection of the field.
- To retain an interface with nearly constant thickness an artificial compressive velocity field is introduced next to the physical velocity field.

In the following Sections these characteristics will be discussed in more detail. Finally, computational results obtained with the proposed modified LS algorithm are presented for the test cases discussed in Chapter 2 and Chapter 4. The proposed algorithm will be assessed on the accuracy of it's solutions, it's mass conservation properties and it's ability to maintain an interface region of constant width.

6.2 The Modified Level-Set method

The main idea of the modified Level-Set method is to define an LS function that mimicks the VoF field as close as is practically possible from a computational point of view, to improve the mass conservation property of the former method. A commonly made choice for the modified LS function is a scaled and translated tangent hyperbolic function of the (signed) distance to the interface. Note that this definition of the modified LS function only explicitly appears in the initial condition of the initial boundary value problem for the modified LS function. This initialization is demonstrated for a one-dimensional example.

Assume that the LS field is defined by the signed distance function $\phi(\boldsymbol{x},t)$. The interface is the zero LS of $\phi(\boldsymbol{x},t)$, i.e. the interface $I(\boldsymbol{x},t)$ is defined as

$$I(\boldsymbol{x},t) = \{\boldsymbol{x} \in \Omega \mid \phi(\boldsymbol{x},t) = 0\}. \tag{6.2}$$

To improve the mass conservation property of the LS advection algorithm, the following modified LS field $\phi_m^\epsilon(\boldsymbol{x},t)$ is defined:

$$\phi_m^\epsilon(\boldsymbol{x},t) = F_m^\epsilon(\phi(\boldsymbol{x},t)), \tag{6.3}$$

where

$$F_m^\epsilon(\phi(\boldsymbol{x},t)) = \frac{1}{2}\left(\tanh\left(\frac{\phi(\boldsymbol{x},t)}{2\epsilon}\right)+1\right). \tag{6.4}$$

where ϵ is a parameter that controls the width of the interface region G_I. The definition of the interface in terms of the modified LS field $\phi_m^\epsilon(\boldsymbol{x},t)$ is now given by

$$I(\boldsymbol{x},t) = \{\boldsymbol{x} \;\; \epsilon\; \Omega \mid \phi_m^\epsilon(\boldsymbol{x},t) = \tfrac{1}{2}\}, \tag{6.5}$$

but it seems to be more appropriate to define an interface region G_I as:

$$G_I = \{\boldsymbol{x} \in \mathcal{R}^2 |\; \delta < \phi(\boldsymbol{x},t) < 1-\delta\;\}, \quad 0 < \delta < \tfrac{1}{2}. \tag{6.6}$$

where $\delta \ll 1$. The interface width $w(t)$ is defined in the following way:

$$\left.\begin{array}{l} \boldsymbol{x}^+(\boldsymbol{x}_s, w^+) = \boldsymbol{x}_s(t) + \nabla\phi_m^\epsilon(\boldsymbol{x}_s,t)w^+ \\ \boldsymbol{x}^-(\boldsymbol{x}_s, w^-) = \boldsymbol{x}_s(t) - \nabla\phi_m^\epsilon(\boldsymbol{x}_s,t)w^- \end{array}\right| \; \boldsymbol{x}_s(t) \in I(t). \tag{6.7}$$

$$w(t) = \max_{w^+,w^->0\;;\;\boldsymbol{x}^+,\boldsymbol{x}^-\in G_I(t)} (|\boldsymbol{x}^+ - \boldsymbol{x}^-|) \tag{6.8}$$

It is most important to formulate the algorithm in a way $w(t)$ does not grow over time, because this would prohibit the use of the method for the computation of stationary solutions to the two-phase flow model.

The parameter ϵ has dimension length and has to be related to a characteristic length scale of the numerical method at hand. In many two-phase flow models the interface width is related to a representative mesh width h, e.g. in the Continuous Surface Force model of Brackbill [9]. Here h is defined as $h = \max(\sqrt[d]{|\Omega_k|})$, with d the dimension of the problem and $|\Omega_k|$ is the area of the k^{th} control volume. Smaller values of ϵ bring the modified LS field closer to the VoF field, but these also increase the norm of the gradient of the modified LS field in the vicinity of the interface. The need for an interface region with minimal width to minimize changes in mass has to be balanced with the difficulties presented to the spatial discretisation method by the large gradient of the modified LS field.

It is interesting to consider the influence of ϵ on the width of the interface region (6.6). First of all the interface region has to be defined in terms of the parameter $0 < \delta < \frac{1}{2}$. The influence of the value of δ on the interface width for a fixed value of the dimensionless parameter ϵ/h is shown in Figure 6.1. Clearly, the interface width vanishes independently of the parameter ϵ for $\delta = \frac{1}{2}$. For a given value of ϵ/h the interface width increases with decreasing δ, and becomes infinitely large when δ approaches zero.

Next, the value of the dimensionless parameter ϵ/h is considered in Figure 6.2 and Figure 6.3, for a fixed value of δ. For a given value of δ the interface width scales linearly with ϵ/h. The value of δ is small but arbitrary, and the value is only set to be able to visualize the interface width in terms of the definition (6.7).

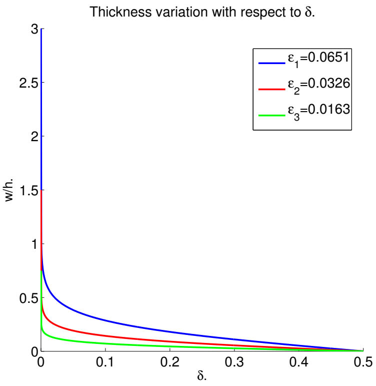


Figure 6.1: The influence of the parameter δ on the width of the interface for a fixed value of the parameter ϵ/h.

In the Continuous Surface Force model for surface tension, the interface region has a width of $3h$, where h is a representative mesh width. For a one dimensional case the relation between the parameters δ, ϵ and $|G_I|$ can be easily determined as:

$$w = 4\epsilon \operatorname{arctanh}(1-2\delta). \tag{6.9}$$

Similar to (6.1) the mass contained in control volume Ω_k is defined as [42]:

$$M_k(t) = \int_\Omega H(\phi^\epsilon_m(\boldsymbol{x},t) - \tfrac{1}{2})d\Omega. \tag{6.10}$$

Conservation of $\sum M_k(t)$ will result in exact mass conservation, i.e. the radius of a circular interface would remain constant while the interface is advected, but would lead back to a discretisation of a VoF type field. Alternatively, define the quantity $M^\epsilon_k(t)$ as:

$$M^\epsilon_k(t) = \int_{\Omega_k} \phi^\epsilon_m(\boldsymbol{x},t)\ d\Omega. \tag{6.11}$$

The field $\phi^\epsilon_m(\boldsymbol{x},t)$ approaches $H(\phi^\epsilon_m(\boldsymbol{x},t) - \frac{1}{2})$ in the limit of vanishing ϵ so an equivalent definition of $M_k(t)$ is given by:

$$M_k(t) = \lim_{\epsilon\to 0} \int_{\Omega_k} \phi^\epsilon_m(\boldsymbol{x},t)\ d\Omega. \tag{6.12}$$

If the algorithm ensures $0 < w(t) < C$, with $C = \mathcal{O}(\epsilon)$ then mass loss $|M(0) - M(t)| = \mathcal{O}(\epsilon)$ for the continuous model. Clearly, for nonzero ϵ the

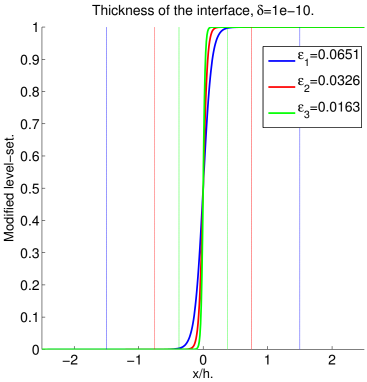


Figure 6.2: The dimensionless width of the interface w/h for a fixed definition of the interface and different values of the dimensionless parameter ϵ/h. The vertical lines indicate the coordinate value $x_{\mathrm{min,max}}/h$ where $|\phi^{\epsilon}_{m}(x_{\mathrm{min,max}}) - (\phi_m)_{\mathrm{min,max}}| < \delta$.

method can guarantee mass conservation only in the sense $|M_\epsilon(0) - M_\epsilon(t)| = 0$, but will conserve mass in the sense $|M(0) - M(t)| = \mathcal{O}(\epsilon)$ much more accurately than the LS method. The introduction of the compressive velocity field will ensure this. The discretization of the conservation equation for the modified LS field is done in a way that differs only in one aspect from the discretisation of the LS field described in Chapter 2. The main difference between the signed-distance function LS field and the modified LS field is that the latter is a highly nonlinear function of the signed distance from the interface, while the former is a linear function of that same distance. The highly nonlinear, near discontinuous behavior near the interface will result in oscillations in this region when high-order spatial discretisations are applied. Naturally, these oscillations are unwanted, but more importantly any modified LS value outside the interval $[0, 1]$ is nonphysical. Furthermore, it will become impossible to accurately approximate the interface curvature from an oscillatory modified LS field. To suppress the oscillations the vertex based limiter proposed by [32] will be utilized. The limiter is very succesful in removing the oscillations but the resulting additional numerical viscosity leads to an increase of the interface width. In the finite volume discretisation of the modified LS equation proposed by [51], the evolution of the modified LS field is split into two separate steps: First the interface, implicitly defined by the modified LS field, is advanced. Next an interface compression step is applied that theoretically leaves the interface position unchanged. In later approaches both are combined in a single step by introducing an artificial compressive velocity field, that is directed

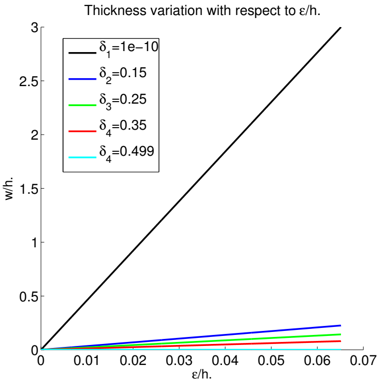


Figure 6.3: LS, Modified LS for different ϵ and VoF

orthogonal to and towards the interface location. Combining both steps improves efficiency, but defining an artificial compressive velocity field that is independent of the problem to be solved and the time step size presents to be a challenge.

The influence of the limiter and the compressive velocity field are studied for a one-dimensional problem. First a baseline solution without limiting is presented. Next the particulars of the limiter are presented, followed by the limited solution of the one dimensional problem. Finally, the compressive velocity field is introduced and shown to have a significant influence on the interface width.

A base line solution can be defined by applying the discontinuous Galerkin discretisation to the conservation equation of the modified LS field without limiting for the one-dimensional problem defined in Chapter 2 of the LS field discretisation. Significant oscillations occur in the direct vicinity of the interface both up- and downstream Fig. 6.5.

6.3 A vertex-based limiter for the modified level-set function

To suppress the oscillations in the solution the vertex-based limiter proposed by Kuzmin [32] is applied. The limiter is an extension of the Barth-Jasper limiter [25, 32] with enhanced accuracy and especially developed to be applied to a solution of a discontinuous Galerkin discretisation. The limiter is guaranteed to bound the solution, but not necessarily within the interval $[0, 1]$. However, if the initial condition is within this interval, so will be the solution at later times. The vertex-limiter can be formulated for application to a polynomial expansion of the solution of arbitrary order. For the modified LS the first or second degree formulation will be utilized. The coefficients in the limiter for an expansion of the solution of degree p are recursively defined in terms of the coefficients of the limiter for a degree $p-1$ expansion. It is important to stress that the vertex-based limiter is to the best of the author's knowledge the only limiter that can be applied to a discontinous Galerkin discretisation on a set of unstructured triangular control volumes and can be extended to be used to limit a discrete solution of arbitrary polynomial order. However, other limiters do exist for application in the context of a discontinous Galerkin discretisation, but only for one-dimensional problems or multi-dimensional problems utilizing (block) structured meshes [30, 31, 33, 62]. No theoretical results for the expected order of convergence of the limited solution are presented in [32].

6.3.1 Limiter for linear polynomial expansion of the modified level-set field

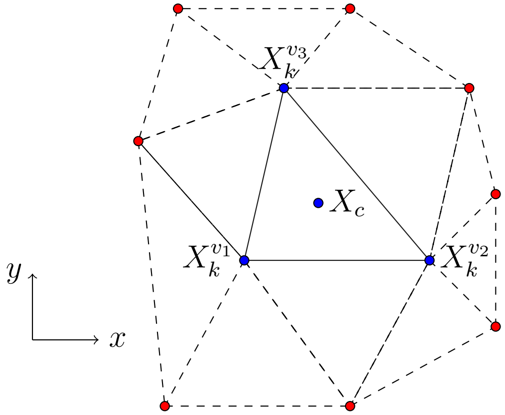


Figure 6.4: Control volume Ω_k and it's neighbouring elements (dashed edges) that contribute to the limited solution.

Consider the piecewise linear discontinuous Galerkin approximation of the modified LS field on a tessellation of the domain $\Omega \subset \mathbb{R}^2$ consisting of the set W_Ω of nonoverlapping triangular control volumes Ω_k, $k = 1..N$, as shown in Fig. 6.4. The value of the solution at the cell centroid $\boldsymbol{X}_k^c$ is denoted by ϕ_k^c . Note that the solution at the vertices is not uniquely defined, because the discrete solution is discontinuous across the boundary of the elements. For each vertex $\boldsymbol{X}_l$ a set $N_b(\boldsymbol{X}_l)$ of triangular elements can be defined, for which $\boldsymbol{X}_l$ is a common vertex:

$$N_b(\boldsymbol{X}_l) = \left\{ j \in \mathbb{N} : \left\{ \boldsymbol{X}_j^{v1}, \boldsymbol{X}_j^{v2}, \boldsymbol{X}_j^{v3} \right\} \cap \{\boldsymbol{X}_l\} \neq \emptyset \right\}. \tag{6.13}$$

Unique values $\phi_{k\,\max}$ and $\phi_{k\,\min}$ can be defined as:

$$\phi_{k\,\max} = \max_{j \in N_b(\boldsymbol{X}_k)} \phi_j^c, \quad \phi_{k\,\min} = \min_{j \in N_b(\boldsymbol{X}_k)} \phi_j^c. \tag{6.14}$$

Also define a mapping $\mathcal{M}_k,\ \{1,2,3\} \to \mathbb{N}$ that maps the local labels of the three vertices to the global labels of the vertices in the tesselation in the following way:

$$\boldsymbol{X}_k^{v_i} = \boldsymbol{X}_{\mathcal{M}_k(i)},\ i = 1,2,3. \tag{6.15}$$

The first order Taylor expansion of the modified LS field with respect to the cell centroid and with *controlled* gradient is defined as (dropping the ϵ superscript to improve readability):

$$\phi_k^h(\boldsymbol{x}) = \phi_k^c + \alpha_k \left.\nabla\phi\right|_{\boldsymbol{X}_k^c} (\boldsymbol{x} - \boldsymbol{X}_k^c), \quad 0 \leqslant \alpha_k \leqslant 1, \tag{6.16}$$

where α_k is an (elementwise defined) parameter to restrict the value of the gradient to limit the flux. For the vertex based limiter it's value is defined as:

$$\alpha_k = \min_{i \in \{1,2,3\}} \begin{cases} \min\left\{ \dfrac{\phi_{\mathcal{M}_k(i)\,\max} - \phi_k^c}{\phi_i - \phi_k^c}, 1 \right\} & \text{if } \phi_i - \phi_k^c > 0, \\ 1 & \text{if } \phi_i - \phi_k^c = 0, \\ \min\left\{ \dfrac{\phi_{\mathcal{M}_k(i)\,\min} - \phi_k^c}{\phi_i - \phi_k^c}, 1 \right\} & \text{if } \phi_i^c - \phi_k^c < 0. \end{cases}$$

Here, ϕ_i is the unconstrained solution in element Ω_k at vertex $X_k^{v_i}$, defined as:

$$\phi^i = \phi_k^c + \left.\nabla\phi\right|_{\boldsymbol{X}_k^c} (\boldsymbol{X}_k^{v_i} - \boldsymbol{X}_c). \tag{6.17}$$

This generic formulation of the vertex-based limiter guarantees no new local extrema will occur in the solution for $t > 0$. Therefore, in this case it ensures the modified LS field will remain contained within the interval $[0,1]$.

6.3.2 Limiter for degree two polynomial expansion of the level-set field

More challenging is to guarantee the modified LS field remains contained within the interval $[0,1]$, when a higher order discontinous Galerkin discretisation is applied to the conservation equation for the modified LS field.

The formulation of the vertex based limiter for a linear approximation of the solution presented in the previous section can be extended to be applied to an approximation of arbitrary order [30,31].

This solution can be expanded in a Taylor polynomial of degree two with respect to the cell center $\boldsymbol{X}_k^c$ of Ω_k as,

$$\phi_h = \phi_c + \nabla\phi|_{\boldsymbol{X}_k^c}\left(\boldsymbol{x} - \boldsymbol{X}_k^c\right) + \left(\boldsymbol{x} - \boldsymbol{X}_k^c\right)^T H(\phi)|_{\boldsymbol{X}_k^c}\left(\boldsymbol{x} - \boldsymbol{X}_k^c\right), \tag{6.18}$$

$$H(\phi) = \tfrac{1}{2}\begin{pmatrix} \phi_{,11} & \phi_{,12} \\ \phi_{,21} & \phi_{,22} \end{pmatrix}.$$

Introducing volume averaged quantities, (6.18) can be transformed to [32]:

$$\phi_h = \overline{\phi}_c + \nabla\phi|_{\boldsymbol{X}_k^c}\left(\boldsymbol{x} - \boldsymbol{X}_k^c\right) + \left(\boldsymbol{x} - \boldsymbol{X}_k^c\right)^T H(\phi)|_{\boldsymbol{X}_k^c}\left(\boldsymbol{x} - \boldsymbol{X}_k^c\right) - H(\phi)|_{\boldsymbol{X}_k^c} : I_k, \tag{6.19}$$

$$I_k = \tfrac{1}{2}\begin{pmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{pmatrix}.$$

The volume average $\overline{\phi}_c$ and tensor I_k are defined as:

$$\overline{\phi}_c = \frac{1}{|\Omega_k|}\int_{\Omega_k} \phi_h d\Omega, \qquad I_{11} = \frac{1}{|\Omega_k|}\int_{\Omega_k} \left(x_1 - x_1^c\right)^2 d\Omega, \tag{6.20}$$

$$I_{22} = \frac{1}{|\Omega_k|}\int_{\Omega_k} \left(x_2 - x_2^c\right)^2 d\Omega, \qquad I_{12} = I_{21} = \frac{1}{|\Omega_k|}\int_{\Omega_k} \left(x_1 - x_1^c\right)\left(x_2 - x_2^c\right) d\Omega. \tag{6.21}$$

I_{11}, I_{22} and I_{12} are the second area moments of the control volume with respect to the center of the element. These can be pre-computed for the complete mesh in advance of the solution of the time-stepping sequence. Both formulations (6.18) and (6.19) are equivalent approximations of the solution, i.e. they have the same order of accuracy (add additional discussion of the alternative formulation based on averaging). Now both the first and second degree contributions are limited by a factor $0 \le \alpha^{(\beta)} \le 1$:

$$\phi_h = \overline{\phi}_c + \alpha_k^{(1)}\left(\boldsymbol{x} - \boldsymbol{X}_k^c\right)^T H(\phi)|_{\boldsymbol{X}_k^c}\left(\boldsymbol{x} - \boldsymbol{X}_k^c\right) - \alpha_k^{(2)} H(\phi)|_{\boldsymbol{X}_k^c} : I_k. \tag{6.22}$$

The factor $\alpha_k^{(2)}$ defined as

$$\alpha_k^{(2)} = \min\{\alpha_{k\ x_1}^{(2)}\ ,\ \alpha_{k\ x_2}^{(2)}\}, \tag{6.23}$$

where $\alpha_{k\ x_1}^{(2)}$ and $\alpha_{k\ x_2}^{(2)}$ are used to limit corrections to the individual components of the gradient vector:

$$\phi_{,1}^{(2)} = \phi_{,1}|_{\boldsymbol{X}_k^c} + \alpha_{k\ x_1}^{(2)}\left\{\phi_{,11}|_{\boldsymbol{X}_k^c}\left(x_1 - (\boldsymbol{X}_k^c)_1\right) + \phi_{,12}|_{\boldsymbol{X}_k^c}\left(x_2 - (\boldsymbol{X}_k^c)_2\right)\right\}, \tag{6.24}$$

$$\phi_{,2}^{(2)} = \phi_{,2}|_{\boldsymbol{X}_k^c} + \alpha_{k\ x_2}^{(2)} \left\{ \phi_{,21}|_{\boldsymbol{X}_k^c} (x_1 - (\boldsymbol{X}_k^c)_1) + \phi_{,22}|_{\boldsymbol{X}_k^c} (x_2 - (\boldsymbol{X}_k^c)_2) \right\}. \tag{6.25}$$

The coefficients $\alpha_k^{(2)}$ are computed using the vertex-based limiter, described in the previous section but based on a linear reconstruction of the first derivatives in the vertices. The procedure for evaluation of $\alpha_k^{(1)}$ starts with the procedure followed in the linear case, but is followed by a comparison between $\alpha_k^{(1)}$ and $\alpha_k^{(2)}$, that resets $\alpha_k^{(1)}$ as:

$$\alpha_k^{(1)} := \max(\alpha_k^{(1)}, \alpha_k^{(2)}). \tag{6.26}$$

Once the coefficients $\alpha_k^{(\beta)}$ are known the limited solution can be constructed in each element prior to advancing to the next time step. A more detailed description of the limiting process is presented in [30,31,33,54,62].

6.4 Compressive velocity formulation

In order to keep the width of the interface region constant the use of a compressive, artificial, velocity field is proposed. The limiter will suppress oscillations in the modified LS field in the vicinity of the interface but this will simultaneously cause the width of the interface region to grow. The compressive velocity field will reduce the latter effect.

Although, this is a commonly applied augmentation of a finite volume discretisation of the modified LS equation, it has not been introduced in the context of a discontinous Galerkin discretisation, but for finite volume discretisation it has been reported to improve the resolution of the interface. Additionally, in the implementation presented in [51], the compressive velocity field is scaled in a (weakly) problem specific way.

The equation that describes the evolution of the modified LS field is given by:

$$\frac{\partial \phi_m(\boldsymbol{x})}{\partial t} + \left((u_\alpha(\boldsymbol{x}) + u_\alpha^c(\phi_m(\boldsymbol{x}))) \right)_{,\alpha} = 0, \ \boldsymbol{x} \in \Omega. \tag{6.27}$$

where u_α^c denotes the artificial compressive velocity field defined as:

$$u_\alpha^c(\phi_m(\boldsymbol{x})) = K_c \phi_m(\boldsymbol{x})(0.5 - \phi_m(\boldsymbol{x}))(1 - \phi_m(\boldsymbol{x}))\ \frac{\phi_{m,\alpha}(\boldsymbol{x})}{|\nabla \phi_m(\boldsymbol{x})|}. \tag{6.28}$$

The formulation of the compressive velocity field is chosen to accomplish the following:

- The compressive velocity is only nonzero in the direct vicinity an on both sides of the interface.
- The inclusion of the compressive velocity field is chosen such that an additional artificial viscosity $\epsilon_a \leq 0$ is added of the form:

$$\epsilon_a = -K_c \phi_m(\boldsymbol{x})(0.5 - \phi_m(\boldsymbol{x}))(1 - \phi_m(\boldsymbol{x})) \frac{1}{|\nabla \phi_m(\boldsymbol{x})|}. \qquad (6.29)$$

- The compressive velocity field is directed orthogonal to the interface, and towards the interface. The sign of the compressive velocity field is different on both sides of the interface, to apply compression and leave the interface location unaffected.

The coefficient K_c has the dimension of m/s and to have the appropriate scaling behaviour should be related to the appropriate problem length and time scale.

Clearly, (6.27) is a nonlinear equation. However, evaluation of the compressive velocity field at the previous time step restores linearity for the discretized equation. It is very important that the compressive velocity field is evaluated at the edges of the control volumes, to ensure conservation of the modified LS field. The fact that the compressive velocity field is not solenoidal is of no concern for the conservation properties of the equation. The choice of the appropriate value of K_c is experimentally determined.

6.5 Test Cases

The key characteristics of the modified LS method that need to be assessed are the mass conservation property, the accuracy of the modified LS field and the thickness of the interface. To accomplish this, three test cases are selected based on the properties of the imposed velocity field: constant, linear or nonlinear. The test cases chosen are:

- Translation of the interface in $\Omega_D \subset \mathbb{R}$,
- Solid body rotation of a circular interface in $\Omega_D \subset \mathbb{R}^2$,
- The *reversed vortex* test case [34, 56, 60].

In all cases the problem is nondimensionalized by introducing a reference length and reference time scale $L = 1\ m$ and $T = 1\ s$, respectively.

The expansion of the solution can be done for polynomial basis functions of a user defined degree, without any difficulty. However, the number of nodal points increases with the degree of the basis functions and hence the number of degrees of freedom of the problem. Because the vertex-based limiter discussed in Section 6.3 is derived for an expansion in linear and quadratic basis functions, only those bases will be used in the experiments. However, the frame work of the vertex based limiter permits to extend it to any arbitray order polynomial.

6.5.1 Translation of the interface in one spatial dimension

In the first test case the modified LS field is a scaled and mollified rectangular function, that is translated without any deformation. The LS field is initialized in domain $\Omega_D = [0, 2]$ as:

$$\phi(\boldsymbol{x}, 0) = |\boldsymbol{x} - \boldsymbol{x^c}(\mathbf{0})| - R, \tag{6.30}$$

where $\boldsymbol{x^c}(\mathbf{0}) = \frac{1}{2}$ and $R = \frac{1}{5}$. A uniform mesh of size $h = 2/K$ is used, where K is the number of control volumes. The interface thickness parameter is set as $\epsilon = \frac{3}{10}h$. The initial modified LS field can be given as:

$$\phi_m(\boldsymbol{x}, 0) = \frac{1}{2}\left(\tanh\left(\frac{\phi(\boldsymbol{x}, 0)}{2\epsilon}\right) + 1\right). \tag{6.31}$$

The velocity $u_1 = 1$ and the final time for the solution is set to $T = 1$. This means the solution for the LS field at the final time T is given as:

$$\phi(\boldsymbol{x}, T) = |\boldsymbol{x} - \boldsymbol{x}^c(T)| - R, \tag{6.32}$$

where $\boldsymbol{x}^c(T) = 1.5$. The corresponding modified LS field at the final time is given by:

$$\phi_m^{\epsilon}(\boldsymbol{x}, T) = \frac{1}{2}\left(\tanh\left(\frac{\phi(\boldsymbol{x}, T)}{2\epsilon}\right) + 1\right). \tag{6.33}$$

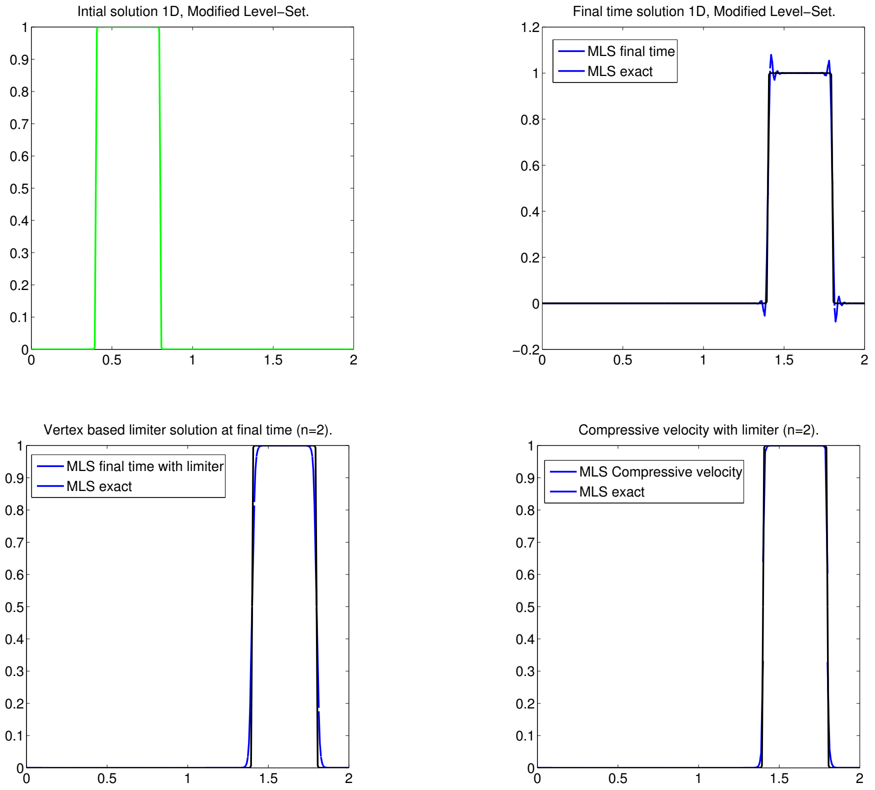


Figure 6.5: Translation of a circular region in $\Omega_D \subset \mathbb{R}$. The solution is shown at the initial and final time. Clockwise from the top left corner: Exact solution, numerical solution without application of a limiter, numerical solution based on the application of the vertex-based limiter, numerical solution based on the application of the vertex-based limiter and a compressive velocity field.

Three different approaches are compared for this one-dimensional test case:

- discretisation using either linear or quadratic expansion of the modified LS field, without application of the vertex-based limiter,
- discretisation using either linear or quadratic expansion of the modified LS field, with application of the vertex-based limiter,
- discretisation using either linear or quadratic expansion of the modified LS field, with application of the vertex-based limiter and incorporation of a compressive velocity field.

In Figure 6.5 and Tables 6.1, 6.2, 6.3 and 6.4 the results of the different approaches are shown. Clearly, without application of the limiter the solution shows unacceptable oscillations near the interface location. The limiter succesfully suppresses these oscillations and from the results in Tables 2.3, 6.2, 6.3 and 6.4 the application of the limiter has no significant influence on the asymptotic order of convergence. Kuzmin [32] does not present an assessment of the influence of the accuracy of the vertex-based limiter, but the observed behavior is in line with the results presented in [30,31]. Inclusion of a small compressive velocity improves the resolution of the interface and again has no significant influence on the asymptotic rate of convergence. Addtional experiments show the order of convergence is affected to some extend by the choice of ϵ/h. When this parameter is chosen too small, the asymptotic order of convergence is reduced. The mass conservation properties can be assessed using either the 'strict' requirement based on the difference $|M(t)-M(0)|$ or the 'loose' requirement $|M(t)^{\epsilon}-M(0)^{\epsilon}|$. Results for the 'loose' requirement are not shown because these are near machine precision in all experiments. The average 'strict' mass error is shown in Tables 2.3, 6.3 and converges, although not monotonically, upon grid refinement. Figure shows how the error $|M(t)-M(0)|$ evolves over time for the mesh with $h=2/256$. The maximum error in the mass does not grow significantly over time, but is significantly larger than for the MCLS. For those applications where the requirements on mass conservation can be relaxed the modified LS method can be a viable alternative for the costly ALE VoF advection algorithm proposed in the MCLS.

h	**N=1**		
	$\|\boldsymbol{error}\|_{L_2}$	**Order**	$\|M(T)-M(0)\|$
2/32	5.473e-02	—	8.042e-03
2/64	2.276e-02	1.265	4.365e-04
2/128	9.768e-03	1.220	2.236e-04
2/256	2.122e-03	2.202	1.270e-05
2/512	3.309e-04	2.681	5.231e-06

Table 6.1: Convection of the interface in one spatial dimension, mass error ($|M(T)-M(0)|$), L_2-error of the modified LS field and estimated convergence rate without compressive velocity field ($K_c = 0$) for expansion in linear polynomial basis $N = 1$.

The chosen values for the compressive velocity field and the interface thickness parameter ϵ/h of this one-dimensional test case are used for all two-dimensional test cases.

h	**N=2**		
	$\|\|error\|\|_{L_2}$	**Order**	$\|M(T) - M(0)\|$
2/32	1.223e-02	—	2.332e-02
2/64	3.612e-03	1.760	3.493e-03
2/128	3.058e-04	3.562	4.816e-03
2/256	1.882e-05	4.02	7.907e-04
2/512	1.923e-06	3.291	1.174e-03

Table 6.2: Convection of the interface in one spatial dimension, mass error ($|M(T) - M(0)|$), L_2-error of the modified LS field and estimated convergence rate without compressive velocity field ($K_c = 0$) for expansion in quadratic polynomial basis $N = 2$.

h	**N=1**		
	$\|\|error\|\|_{L_2}$	**Order**	$\|M(T) - M(0)\|$
2/32	5.473e-02	—	8.041e-03
2/64	2.279e-02	1.264	4.334e-04
2/128	9.771e-03	1.221	2.273e-04
2/256	2.120e-03	2.204	1.685e-05
2/512	3.284e-04	2.690	4.338e-07

Table 6.3: Convection of the interface in one spatial dimension, mass error ($|M(T) - M(0)|$), L_2-error and estimated convergence rate of the modified LS field with compressive velocity field ($K_c = 1e-5$) for expansion in linear polynomial basis $N = 1$.

h	**N=2**		
	$\|\|error\|\|_{L_2}$	**Order**	$\|M(T) - M(0)\|$
2/32	1.223e-02	—	2.332e-02
2/64	3.611e-03	1.760	3.491e-03
2/128	3.034e-04	3.572	4.818e-03
2/256	1.831e-05	4.050	7.933e-04
2/512	6.654e-06	1.460	1.171e-03

Table 6.4: Convection of the interface in one spatial dimension, mass error ($|M(T) - M(0)|$), L_2-error and estimated convergence rate of the modified LS field with compressive velocity field ($K_c = 1e - 5$) for expansion in quadratic polynomial basis $N = 2$.

6.5.2 Rotation of a circular interface in two spatial dimensions

The second test case is related to the rotation of circular interface in a domain $\Omega_D = [0, 1] \times [0, 1]$. Initially, the centre of the circle is located at $\boldsymbol{x}^c(0) = (\frac{1}{2}, \frac{3}{4})^T$. The initial condition for the LS field is defined as:

$$\phi(\boldsymbol{x}, 0) = |\boldsymbol{x} - \boldsymbol{x}^{\boldsymbol{c}}\boldsymbol{(0)}| - R, \tag{6.34}$$

where $R = \frac{5}{20}$. The corresponding intial modified LS field is given as,

$$\phi_m^\epsilon(\boldsymbol{x}, 0) = \frac{1}{2}\left(\tanh\left(\frac{\phi_l(\boldsymbol{x}, 0)}{2\epsilon}\right) + 1\right). \tag{6.35}$$

The circular interface is advected with a linear velocity field $\boldsymbol{u} = (x_1 - \frac{1}{2}, -x_2 + \frac{1}{2})^T$ using a time step $\Delta t = \frac{1}{100}$. The final time is $T = 2\pi$. This initial setup is shown in Figure 6.6. Because the velocity field is solenoidal, mass should be conserved up to machine precision and the velocity field will return the interface to its initial position at $t = T$. Hence, $\phi_m^\epsilon(\boldsymbol{x}, T) = \phi_m^\epsilon(\boldsymbol{x}, 0)$ as shown in Figure 6.6.

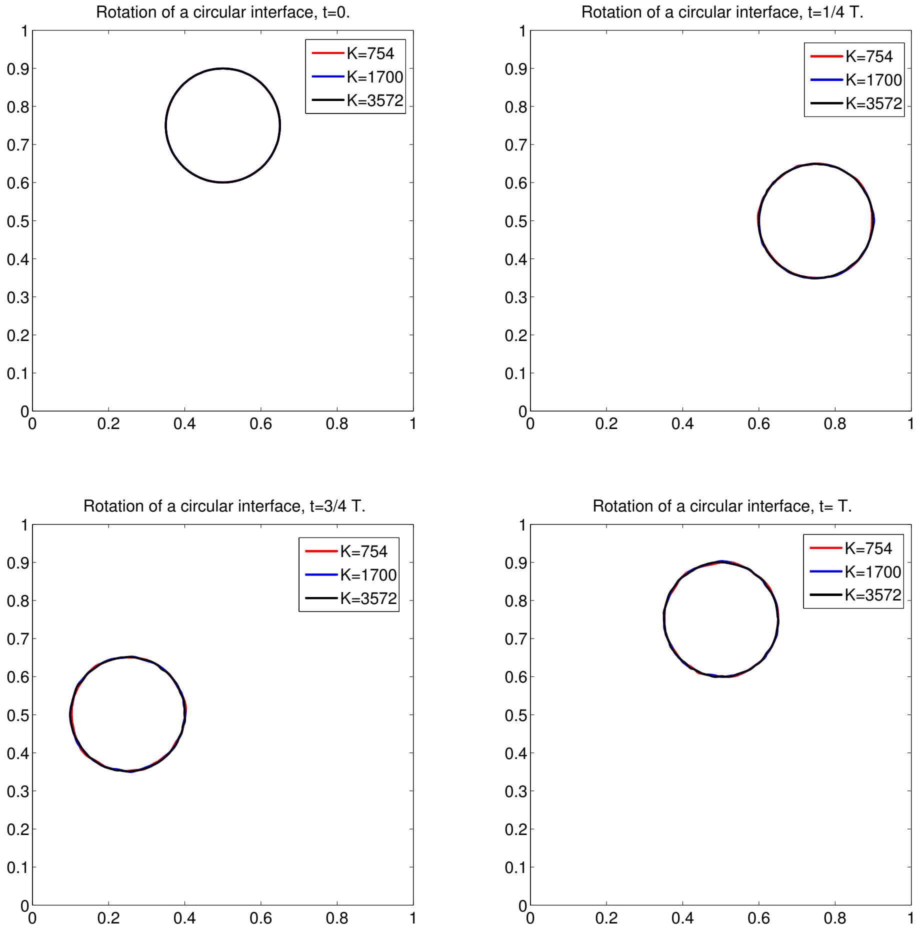


Figure 6.6: The interface $\phi_m^\epsilon(\boldsymbol{x}, t) = 0.5$ for the rotation test case at time levels $t = 0,\ t = \frac{1}{4}T,\ t = \frac{3}{4}T$ and $t = T$.

In Figure 6.7 the mass ratio $M^\epsilon(t)/M^\epsilon(0)$ is shown for three different mesh sizes, for degree $n = 1$ and $n = 2$. The graph shows mass is conserved during the rotation up to machine precision in the 'loose' sense. It is not straightforward to assess the mass conservation properties in the two-dimensional case in the 'strict' sense. Assessment of the mass-conservation properties in the 'strict' sense is necessary but left for future work.

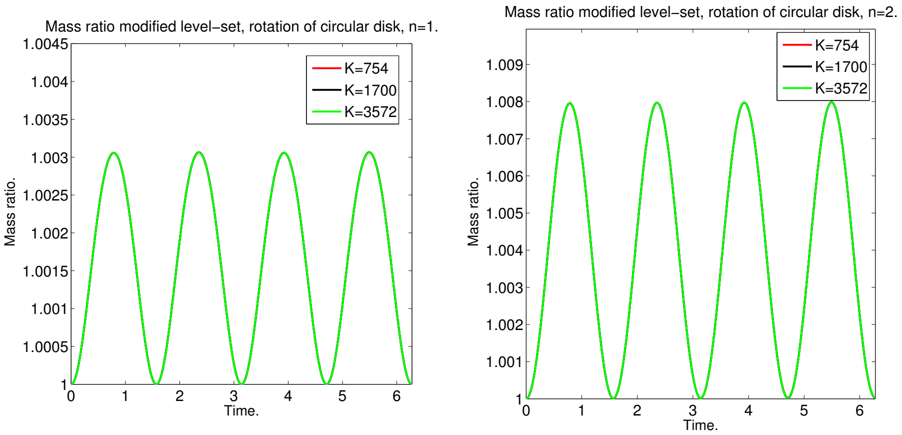


Figure 6.7: Evolution of $M^\epsilon(t)/M^\epsilon(0)$ for the rotation of the MLS representation of a circular interface.

In Table 6.5 the average mass change $\overline{|M^\epsilon(t) - M^\epsilon(0)|}$ and the L_2 error in the solution at the final time $\|\phi^\epsilon_m(\boldsymbol{x},T) - \phi^\epsilon_m(\boldsymbol{x},T)\|_{L_2}$ are presented for three different mesh sizes, together with the order of convergence estimate of the latter quantity. $M^\epsilon(t)$ is not constant due to a nonzero mass flux through the boundary.

K	**N=1**			**N=2**		
	Avg.Mass	**Error**	**Order**	**Avg.Mass**	**Error**	**Order**
754	1.286472e-04	1.2554e-02	—	7.209225e-04	1.4617e-03	—
1700	1.286473e-04	6.2274e-03	1.729	7.209225e-04	6.6431e-04	1.945
3572	1.286477e-04	2.3077e-03	2.701	7.209225e-04	3.5112e-04	1.735

Table 6.5: Average mass error and L_2 error of the MLS field for the rotation of circular test case.

6.5.3 The reversed vortex test case

The *reversed vortex* or *single reversed vortex* test case is one of the more challenging tests for models of two-phase immiscible flow. The reason for this is the severe stretching and deformation of the convected interface caused by the nonlinear velocity field during the advection which makes accurate mass conservation challenging.

Domain Ω and initial condition are identical as for the linear translation test case. The circular region is advected with a divergence free nonlinear velocity field given by:

$$\boldsymbol{u} = \cos(\pi t/T)(\sin^2(2\pi x_1)\sin(2\pi x_2), -\sin^2(2\pi x_2)\sin(2\pi x_2))^T. \qquad (6.36)$$

The $\cos(\pi t/T)$ will cause the velocity field to reverse direction at time $t = T/2$. This implies the interface will return to its original position at time $t = T$. In this test case $T = 2$ is used. The initial setup is shown in Figure 6.8. Like in the other test cases the mass (area) enclosed by the interface is the quantity of interest. This is computed after every time step and compared with the initial quantity to give the amount of mass loss or gain during advection. Also, at the final time the corrected LS field is compared with the initial condition as the circular interface should return to its initial position and shape at time $t = T$.

In Figure 6.8 the interface is shown at different time levels for three different mesh sizes.

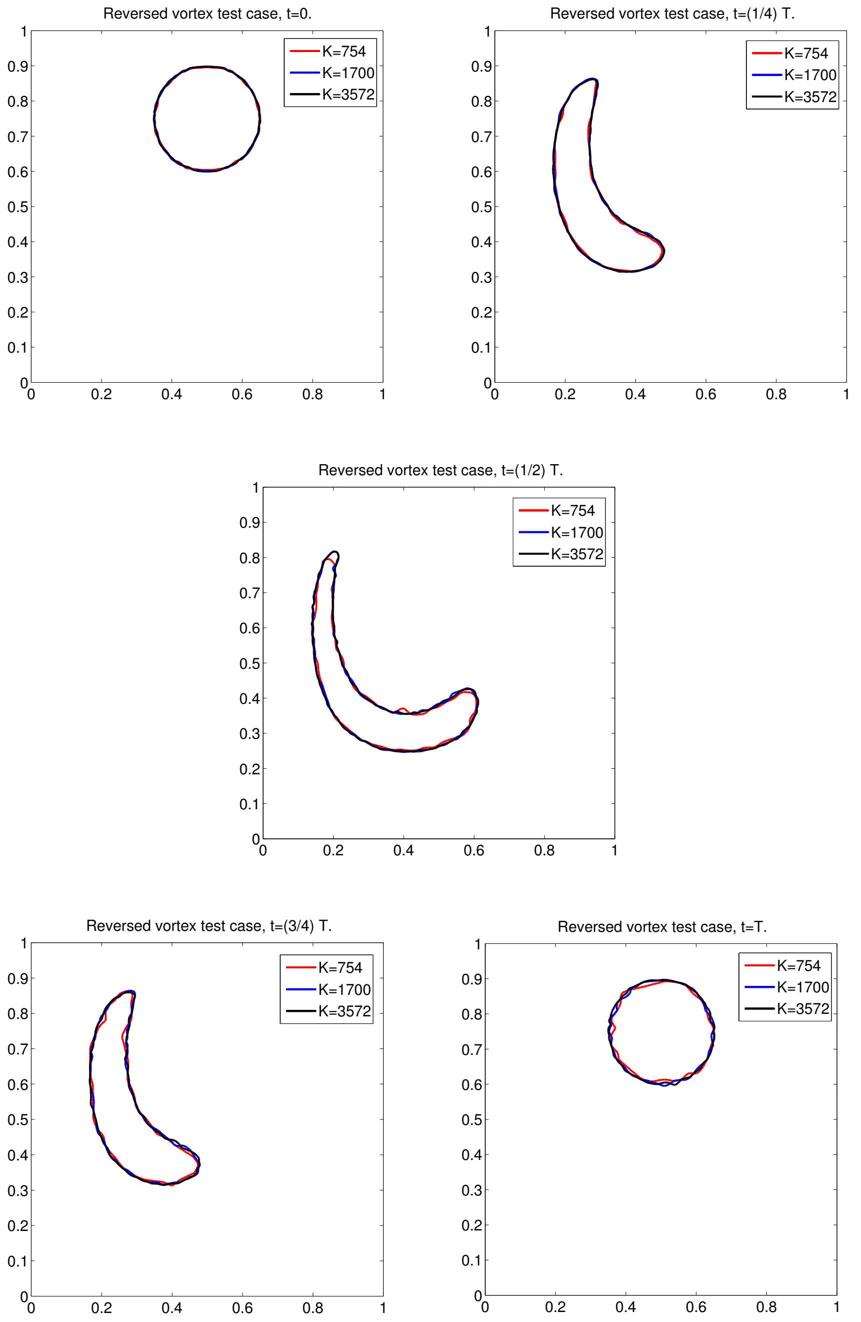


Figure 6.8: The interface $\phi_m(\boldsymbol{x}, t) = 0$ for the reversed vortex test case at time levels $t = 0,\ t = \frac{1}{4}T,\ t = \frac{1}{2}T,\ t = \frac{3}{4}T$ and $t = T$.

In Table 6.6 the mass error $|M^\epsilon(T) - M^\epsilon(0)|$ and the L_2 error of the modified LS field at the final time $\|\phi_m(\boldsymbol{x}, T) - \phi_m^h(\boldsymbol{x}, T)\|_{L_2}$ are presented for three different mesh sizes. The asymptotic order of convergence is close to the theoretical value that can be expected on a general unstructured triangular mesh.

K	**N=1**			**N=2**		
	Avg.Mass	**Error**	**Order**	**Avg.Mass**	**Error**	**Order**
754	1.004e-16	2.284e-02	—	3.743e-19	3.440e-03	—
1700	4.889e-17	1.166e-02	1.657	9.865e-17	1.622e-03	1.826
3572	1.009e-16	6.862e-03	1.444	3.122e-17	8.279e-04	1.842

Table 6.6: Final mass error and L_2 error of the MLS field for the reversed vortex test case.

6.6 Conclusions

The modified LS method is designed to combine the mass-conserving properties of the VoF method with the efficieny and simplicity of a LS method. Oscillations in the modified LS field that are occur in the vicinity of the interface when the rapidly changing modified LS field is projected on a higher order polynomial basis are effectively suppressed by the vertex-based limiter of Kuzmin [32]. With the introduction of an additional nonphysical compressive velocity field, the interface width can be diminished without affecting the asymptotic convergence rate of the discretisation. Despite the sharpness of the interface, one dimensional numerical experiments indicate the current realization of the modified LS method can not match the mass conservation properties of the MCLS method. If the mass conservation requirements can be relaxed, the method can be used as an alternative for the costly VoF advection scheme that is used in the MCLS method.

CHAPTER 7

Comparision of the LS, MCLS and MLS method

7.1 Introduction

So far three different methods to simulate immcisible two-phase flows have been presented, namely the Level-Set method (LS), Mass-Conserving Level-set method (MCLS) and the modified Level-Set method (MLS). Each method has been discussed individually so far and the following key observations have been made:

- The computational cost of the LS method is the lower of the three, but the method does not conserve mass accurately.
- The proposed MCLS method is exactly mass-conserving but it is computationally intensive as a result of the Eulerian-Lagrangian volume of fluid advection.
- The MLS method is more computationally intensive than the LS method, due to the incorporation of a limiter, but less than the MCLS.

The convergence rate of the solutions obtained with the MLS method depend on the values of the interface width parameter and the magnitude of the compressive velocity. This means the MLS model depends on problem specific parameters, contrary to the LS and MCLS method.

Based on these observations, the LS method is discarded as a viable alternative of the MCLS but the MLS requires further assessment. The interest in the MLS method is dual:

- As a stand-alone method that can challenge the MCLS method, because of the relatively low computational cost of the former.
- As a replacement of the computationally intensive VoF advection module of the MCLS.

Therefore, we decided to dedicate one more chapter for further comparison using slightly more difficult test cases, in which the interface representation is more challenging. These test cases consist of;

- Advection of a bubble with a lens shaped cross section.
- *Zalesak's* rotating disc.

Although the LS method is not considered a viable alternative to the MCLS, results for the former method are also included to put the results of the MLS and MCLS in perspective. Both testcases will be computed on a sequence of three meshes to determine the order of convergence of the solution and the convergence rate of the mass error if this is more than machine precision.

7.1.1 Advection of a bubble with a lens shaped cross section

A very commonly studied test case for models for immiscible, incompressible two-phase flow is the rise of an initially circular bubble. Depending on the material properties of the liquid phase and the density ratio between both phases, the bubble will change shape during its ascend to finally reach its terminal spatial form. Accurate prediction of the rise speed is particularly challenging when the terminal shape is lens-like with a very strong variation of the interface curvature: very low curvature near the symmetry line of the bubble and very high curvature near the extremities of the bubble, corresponding to the skirt of the bubble in the three dimensional case. Because the coupling between a flow solver and the interface model is outside the scope of this thesis, a model problem is studied where an interface, that closely resembles the shape of a skirted bubble is convected with a constant velocity.

This solution is obtained in a domain $\Omega = [0,1] \times [0,1]$. The initial condition for the LS field is defined as:

$$\phi_l(\boldsymbol{x},0) = \max(\phi_1(\boldsymbol{x},0), -\phi_2(\boldsymbol{x},0)), \tag{7.1}$$

where $\phi_1(\boldsymbol{x},0)$ and $\phi_2(\boldsymbol{x},0)$ are defined as:

$$\phi_1(\boldsymbol{x},0) = |\boldsymbol{x} - \boldsymbol{x_1^c(0)}| - R, \tag{7.2}$$

$$\phi_2(\boldsymbol{x},0) = |\boldsymbol{x} - \boldsymbol{x_2^c(0)}| - R. \tag{7.3}$$

Here, $\boldsymbol{x}_1^c(0) = (0.5, 0.2)^T$, $\boldsymbol{x}_2^c(0) = (0.5, 0.35)^T$ and $R = 0.15$. The lens shaped interface is advected with a constant velocity field $\boldsymbol{u} = (0, \frac{1}{10})^T$. The final time is $T = 3$. This initial condition for the LS field is shown in Figure 7.1.

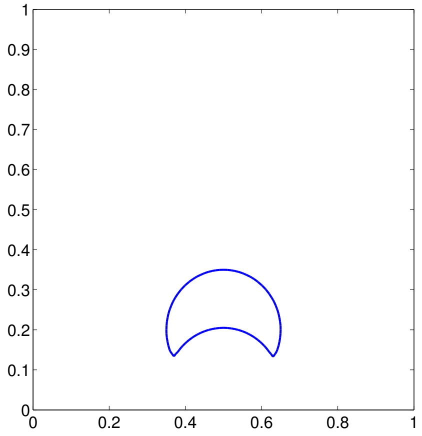


Figure 7.1: Initial position of the bubble with a lens shaped interface.

At the final time i.e. $T = 3$, the final position is explicitly computed to compare the advected LS field and modified field. It is defined as;

$$\phi_l(\boldsymbol{x},T) = \max(\phi_1(\boldsymbol{x},T), -\phi_2(\boldsymbol{x},T)), \tag{7.4}$$

where the LS fields for the two intersecting circular interfaces $\phi_1(\boldsymbol{x},T)$ and $\phi_2(\boldsymbol{x},T)$ are defined as:

$$\phi_1(\boldsymbol{x},T) = |\boldsymbol{x} - \boldsymbol{x_1^c(T)}| - R, \tag{7.5}$$

$$\phi_2(\boldsymbol{x},T) = |\boldsymbol{x} - \boldsymbol{x_2^c(T)}| - R, \tag{7.6}$$

where $\boldsymbol{x}_i^c(T) = \boldsymbol{x}_i^c(0) + T\boldsymbol{u}$. The LS field is converted to the corresponding MLS field as:

$$\phi_m(\boldsymbol{x}, t) = \frac{1}{2}\left(\tanh\left(\frac{\phi_l(\boldsymbol{x}, t)}{2\epsilon}\right) + 1\right). \tag{7.7}$$

where the interface thickness parameter is chosen as $\epsilon = 0.4\max(\sqrt{|\Omega_k})$. The LS field is expanded in a polynomial basis of degree one for the MCLS method and of degree one and two for the LS and MLS method. In Figure 7.2 the mass ratio is shown for three different mesh sizes for the LS, MCLS and MLS method.

In Figure 7.3 the final position of the interface is presented for three different mesh sizes for the LS, MCLS and MLS. The mass error in the solution of the MCLS is close to machine precision. For the LS and MLS the mass error can only be determined in the 'strict' sense of (reference to strict mass conservation equation in chapter on MLS) for an expansion of the solution in polynomials of degree one, because in that case the interface is a line segment.

Both the MLS and MCLS have a tendency to round off the extremities of the bubble, but show superior mass conservation properties. The interface approximation of the MCLS on the coarser mesh is not comparable to the approximations with the other two methods.

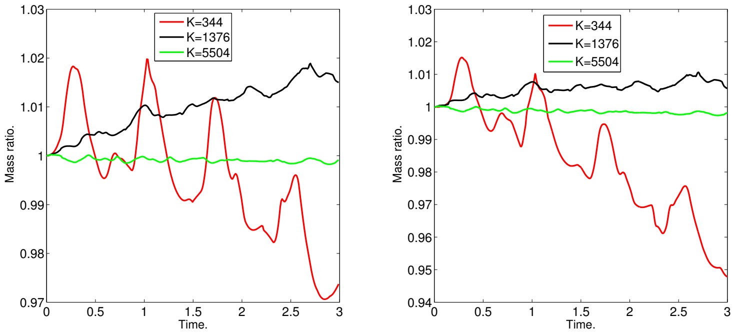


Figure 7.2: Evolution of the mass ratio for the lens shaped interface of LS and MLS for $n = 1$.

In Tables 7.1, 7.2 and 7.3 the average mass error $M_{\text{average}} = \overline{|M(t) - M(0)|}$ and the L_2 error in the solution at the final time $\|error\|_{L_2} = \|\phi(\boldsymbol{x}, T) - \phi^h(\boldsymbol{x}, T)\|_{L_2}$ are presented for three different mesh sizes, together with the estimated order of convergence for the bubble shaped interface. In all experiments the LS and the MLS are conserved to machine precision, but as stated earlier, conservation of these quantities does not imply mass conservation.

K	**MCLS**		
	n=1		
	M.avg.	$\|error\|_{L_2}$	**Order**
344	9.5104e-16	1.0835e-02	—
1376	3.0591e-17	3.2529e-03	1.7359
5504	2.2902e-16	1.1668e-03	1.4791

Table 7.1: Averaged mass error and L2-norm error for the MCLS field for $n = 1$.

K	**LS**					
	n=1			**n=2**		
	M.avg.	$\|error\|_{L_2}$	**Order**	**M.avg.**	$\|error\|_{L_2}$	**Order**
344	1.6086e-04	8.7099e-03	—	—	2.3531e-03	—
1376	4.0969e-04	3.2191e-03	1.4359	—	6.3009e-04	1.9009
5504	3.7851e-05	1.1554e-03	1.4782	—	2.6139e-04	1.2693

Table 7.2: Averaged mass error and L2-norm error for the LS field for $n = 1$ and $n = 2$.

K	**MLS**					
	n=1			**n=2**		
	M.avg.	$\|error\|_{L_2}$	**Order**	**M.avg**	$\|error\|_{L_2}$	**Order**
344	7.1425e-04	1.8541e-02	—	—	5.0323e-03	—
1376	2.3264e-04	6.7263e-03	1.4625	—	1.4751e-03	1.7703
5504	5.8344e-05	2.7912e-03	1.2681	—	5.7920e-04	1.3486

Table 7.3: Averaged mass error and L2-norm error for the MLS field for $n = 1$ and $n = 2$.

The orders of convergence for the LS and MLS method are close to the expected theoretical order of convergence on a general unstructured grid, e.g. the solution obtained with the LS converges with order close to $(n + \frac{1}{2})$.

The order of convergence of the MCLS method is also as expected. This means the convergence rate is not significantly influenced by the fact that the extremities of the interface are clipped off to some extend. The average mass errors in the solutions obtained with the MLS method are comparable to those obtained with the LS method. This means that from this point of view the MLS offers no significant advantage over the LS method: the additional complexity of handling a highly nonlinear function to describe the interface does not pay off.

7.1.2 Zalesak's rotating disc

In this second test case, commonly referred to as the Zalesak's rotating disc [61], the same domain is considered as in the previous test case. The initial condition for the LS field is defined as:

$$\phi_l(\boldsymbol{x},0) = \max(\phi_1(\boldsymbol{x},0), -\phi_2(\boldsymbol{x},0)), \tag{7.8}$$

where, $\phi_1(\boldsymbol{x},0)$ and $\phi_2(\boldsymbol{x},0)$ are defined as follow,

$$\phi_1(\boldsymbol{x},0) = |\boldsymbol{x} - \boldsymbol{x^c(0)}| - R. \tag{7.9}$$

$$\phi_2(\boldsymbol{x},0) = \max(|(x_1 - x_1^c(0))| - w, |x_2 - x_2^c(0) + 2w| - l). \tag{7.10}$$

Zalesak's rotating disc is a challenging test case because the interface is only C^0 continuous and has both convex and concave regions. Numerical methods are assessed on their ability to preserve the sharp corners of the interface. In [52] the Cartesian MCLS is used to compute solutions to this test case on grids with $(50 \times 50), (100 \times 100), (150 \times 150)$ and (200×200) control volumes. For a resolution of (100×100) the shape of the interface is qualitatively correct.

Here, $\boldsymbol{x}^c(0) = (\frac{1}{2}, \frac{3}{4})^T$ and $R = \frac{3}{20}$. It is clear that $\phi_2(\boldsymbol{x},0)$ is a rectangular region of width $w = \dfrac{R}{6}$ and length $l = R$. The slotted disc is rotated around the centre of the domain by a (linear) velocity field, defined as $\boldsymbol{u} = (-x_2 + \frac{1}{2}, x_1 - \frac{1}{2})$. The velocity field is divergence free and has non-zero magnitude at the boundary of the domain. One full rotation around the centre of the domain is considered. Therefore, the final time is $T = 2\pi$. The initial setup is shown in Figure 7.4.

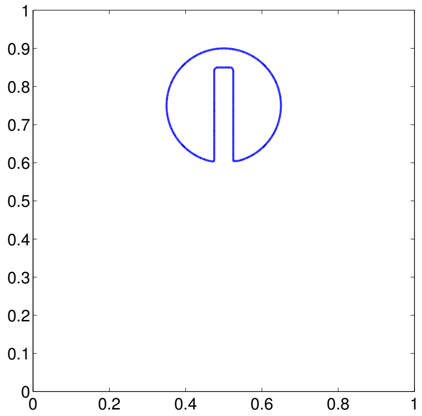


Figure 7.4: Initial position of the interface in Zalesak's rotating disc.

At the end of the rotation, i.e. at the final time, the position of the interface should coincide with the initial position. The accuracy of the method is assessed by comparison of the advected field at the final time. For the MLS method, the initial LS field is first converted to using (7.7).

In Table 7.4 the average mass error $M_{\text{average}} = \overline{|M(t) - M(0)|}$ and the L_2 error in the solution at the final time $\|error\|_{L_2} = \|\phi(\boldsymbol{x}, T) - \phi^h(\boldsymbol{x}, T)\|_{L_2}$ are presented for three different mesh sizes, together with the estimated order of convergence of the LS field. For an expansion in linear polynomials the mass can be exactly computed based on the position of the (linear) interface. For the $n = 2$ case, this computation is not straightforward, although theoretically possible. Because of its large complexity, analysis of the mass conservation properties of degree two representation of the LS and MLS field is left for future research and only results for $n = 1$ are presented.

	LS			**MCLS**			**MLS**		
	M. Avg.	$\|error\|_{L_2}$	**Order**	**M. Avg.**	$\|error\|_{L_2}$	**Order**	**M. Avg.**	$\|error\|_{L_2}$	**Order**
2984	1.8490e-03	8.8117e-03	—	5.4416e-16	8.9715e-03	—	2.9987e-04	2.6429e-02	—
11936	2.8268e-04	3.4650e-03	1.3465	6.4045e-16	3.4613e-03	1.3740	3.4767e-4	8.0843e-03	1.7089
47744	1.0029e-04	1.4077e-03	1.2995	2.6886e-15	1.4109e-03	1.2947	6.4457e-5	2.8736e-03	1.4925

Table 7.4: Zalesak's slotted disc test, average mass error and L2-norm error for the LS, MCLS and MLS field for $n = 1$.

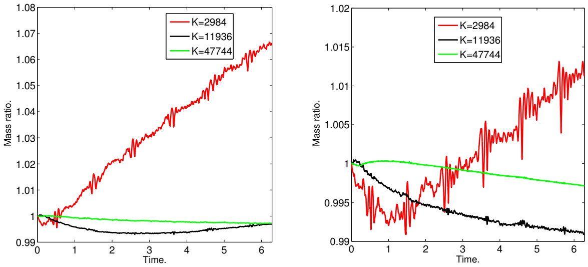


Figure 7.8: Evolution of the mass ratio $M(t)/M(0)$ for Zalesak's rotating disc test, for the LS (left) and MLS (right) field for $n = 1$.

The accuracy of the solutions of LS, MCLS and MLS is comparable.The orders of convergence for the LS and MLS method are close to the expected theoretical order of convergence on a general unstructured grid, e.g. the solution obtained with the LS converges with order close to $(n + \frac{1}{2})$. The mass conserving correction seems to negatively impact the smoothness of the interface, but not the accuracy. Comparing the solution of the extension of the MCLS algorithm with the results for the original algorithm presented in [52] the accuracy of comparable, or even better. This can be attributed to the use of a piecewise constant approximation of the LS field in the original MCLS method, as opposed to the piecewise linear approximation in the proposed algorithm. The use of the MLS offers no advantage over the LS method.

7.2 Conclusions

More challenging test cases show the MLS method is not superior to the LS method with respect to mass conservation, when a more strict definition of mass conservation is imposed. The error in the mass conservation in both methods will reduce with refinement of the mesh, but is not comparable to the MCLS. Imposing the mass conserving correction to the LS field negatively impacts on the smoothness of the interface. The accuracy of the solution obtained with the MCLS method is comparable to the accuracy of the solutions obtained with either the LS and MLS method, but with superior mass conservation.

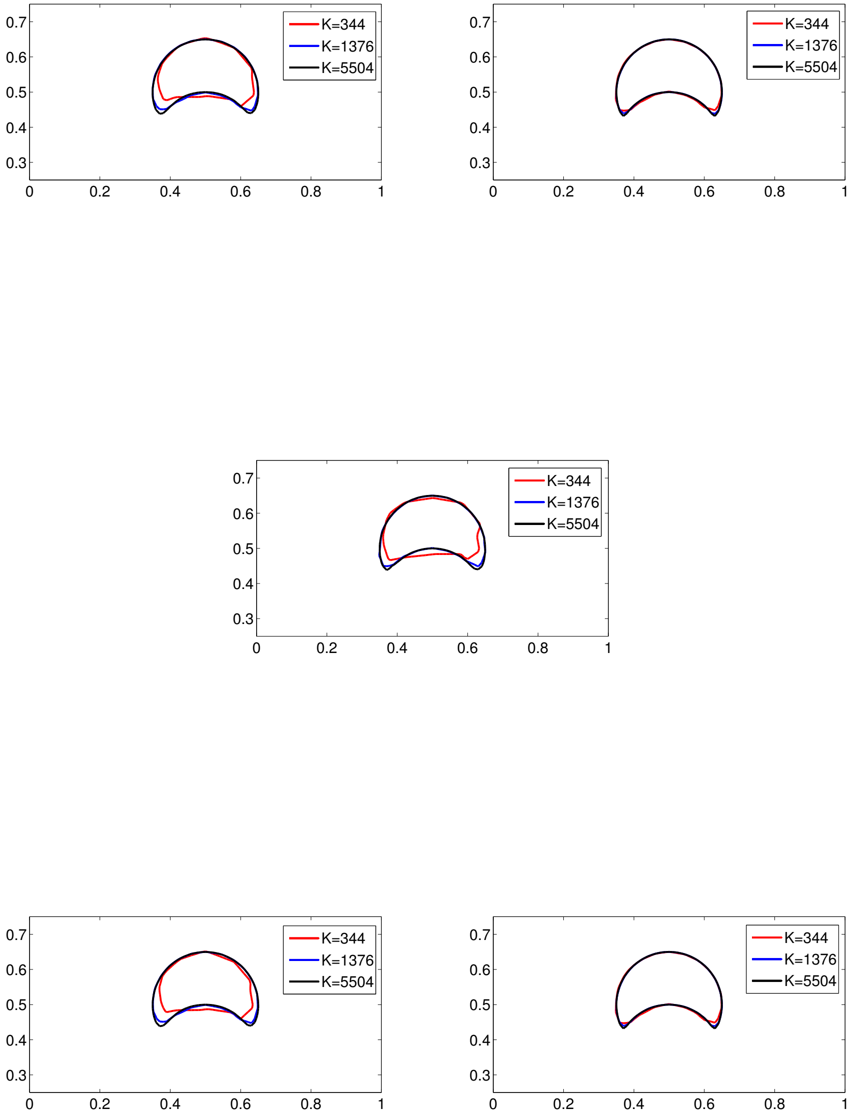


Figure 7.3: The lens shaped interface at the final time $t = T$, for the LS and MLS for $n = 1$ (left) and $n = 2$ (right) and the MCLS for $n = 1$.

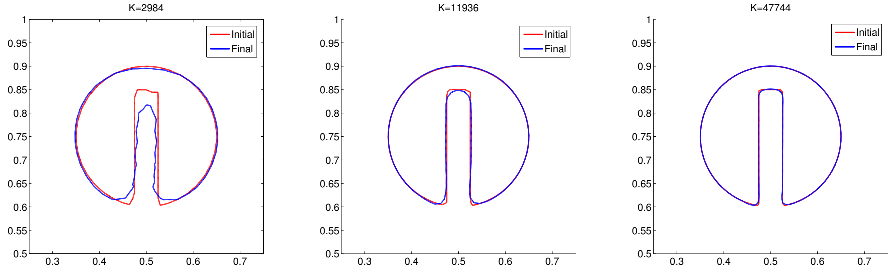


Figure 7.5: Zalesak's slotted disc at $t = T$, for the LS field, using $n = 1$.

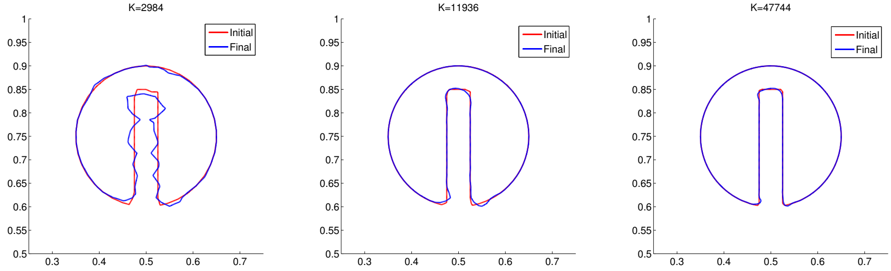


Figure 7.6: Zalesak's slotted disc at $t = T$, for the MCLS field, using $n = 1$.

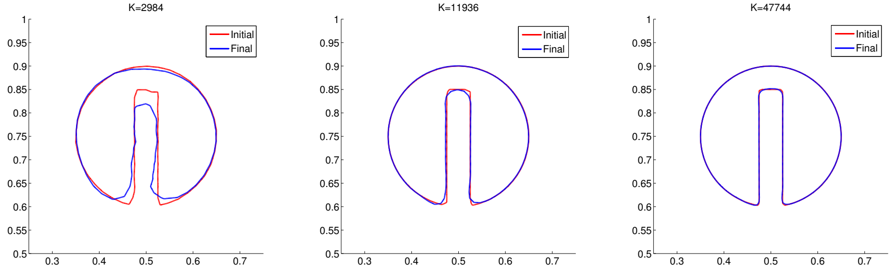


Figure 7.7: Zalesak's slotted disc at $t = T$, for the MLS field, using $n = 1$.

CHAPTER 8

Conclusions and Future Prospects

8.1 Conclusions

The two-dimensional Mass Conserving Level-Set Method, a hybrid Level-Set Volume-of-Fluid method, that combines the advantages of the latter two approaches without combining their computational cost can be extended from a discretisation using structured Cartesian control volumes to general, unstructured control volumes.

In the original method, both the equations for the Level-Set field and the Volume-of-Fluid field were advanced in time, followed by the imposition of a mass-conserving correction on the Level-Set field based on the Volume-of-Fluid field. Contrary to what is done in the original method, the correction can be applied locally in each control volume, without considering the Level Set field in neighbouring control volumes. This new approach makes the new algorithm much more computationally efficient than the old one. Application of a local correction is possible because the Level-Set field is expanded in a set of piecewise linear polynomials, in such a way the discretised field is discontinuous at the interfaces between the control volumes.

For the advection of the color function, which is the main component in the evolution equation for the Volume-of-Fluid field, an Eulerian-Lagrangian algorithm is proposed that conserves the color function, and therefore the area enclosed within the interface up to machine precision. Although the algorithm is computationally costly, the complexity of the construction of (partially overlapping) donating regions is avoided.

The *modified* Level-Set method, as introduced by Olsson and Kreiss [36, 38], is often claimed to have better mass conservation properties than the standard Level-Set method that is based on a signed-distance function formulation. The use of this modified Level-Set method as an alternative for the Volume-of-Fluid module in the Mass Conserving Level Set method, and as an interface capturing approach that can challenge the latter method has been investigated.

Using a state-of-the-art limiter [32] specifically designed for discretisation of hyperbolic conservation laws on unstructured triangular grids, and an additional compressive velocity field the transport equation for the modified Level-Set method can be discretised in such a way the interface width remains nearly constant over time and any nonphysical over- and undershoots in the modified Level-Set field are suppressed. However, it was found that the modified Level-Set method's mass conservation properties are not significantly better than those of the original Level-Set method, hence making this not a viable approach. Furthermore, contrary to the Mass Conserving Level Set method, the properties of the modified Level-Set method strongly depend on problem specific parameters.

8.1.1 Mass conservation

The mass conservation properties of the modified Level-Set methods have to be assessed in the 'strict sense' that the integral over the Heaviside function of the shifted modified Level-Set field has to be constant when there is no net influx of mass and not by assessing the change of the integral over the modified Level-Set field itself.

For an expansion of these fields in a set of piecewise linear polynomials this was found to be relatively straightforward, because within each control volume the interface is a straight line, the intersection of which with boundary of this control volume can be explicitly computed. For an expansion in piecewise degree two or higher polynomials, the reconstruction of the interface is possible too, but extremely complicated. Only by reconstructing the interface the mass conservation can be accurately assessed.

The mass conservation properties of the original Level-Set and modified Level-Set method are comparable. In both cases the area enclosed by the interface fluctuates over time and the mass error increases over the duration of the simulation. Either effect can be reduced to some extend by increasing the mesh resolution, but not eliminated. In all applications of the Mass Conserving Level-Set method the area enclosed by the interface contour is conserved to machine precision.

For those applications for which requirements on mass-conservation can be relaxed, the computational efficiency of the modified Level-Set method remains attractive.

8.1.2 The Mass-Conserving Level-Set Method in relation to other hybrid interface capturing algorithms

The Mass Conserving Level-Set (MCLS) method introduced by Van der Pijl [52, 53] is not unique. The most well known competitors are the CLSVOF method of Sussmann et al. [48, 49] and the VOSET method of Sun [46]. The main difference between CLSVOF and VOSET on one side and the original MCLS is the use of a very efficient algorithm to establish congruence between the Level-Set field and the Volume-of-Fluid field, without explicit interface reconstruction. A simple function relates the two fields within a single Cartesian quadrilateral (2D) or hexahedral (3D) control volume. In the current thesis a similar relation has been defined for triangular control volumes. It is this key difference that makes the (extension of the) MCLS so competitive.

8.1.3 The modified Level-Set method in comparison to the Level-Set method

The near-discontinuous behaviour of the modified Level-Set field in the vicinity of the interface makes accurate discretisation of the transport equation for this field much more challenging and costly than for the Level-Set field. The strong non-linearity of the field will also negatively impact the accuracy of the computation of the interface normal vector and the interface curvature. For these reasons, and the fact that mass conservation is not significantly better than for the Level-Set method, the modified Level-Set method seems to offer little advantage over the original Level-Set method.

8.2 Outlook

A start has been made with the development of a model for two-dimensional immiscible incompressible two-phase flow in geometrically intricate domains. To complete the model the following steps are recommended to be taken.

- The extension of the Mass Conserving Level-Set method has to be embedded in a numerical method that can solve the incompressible Navier-Stokes equations on the same type of control volumes utilized by the proposed interface algorithm. A natural candidate would be a discontinuous Galerkin method, using operator splitting, as proposed by Hesthaven [23].

- To model the effects of surface tension, the local curvature of the interface and the interface normal vector have to be determined from the Level-Set field. It remains to be investigated how this is most accurately done for the piecewise continuous Level-Set field.

- To have even more flexibility for automatic tessellations of geometrically complicated domains, polygonal (2D) and polyhedral (3D) control volumes are preferred. It is worth-while to investigate the possibility of extending the Mass-Conserving Level-Set method for a discretisation using such control volumes, either directly or by using a subdivision of the polyhedral control volumes in a set of tetrahedral control volumes. At least for convex polyhedral this subdivision is always possible.

- Finally, it is recommended that alternatives to the computationally intensive Volume-of-Fluid advection scheme are investigated.

Bibliography

[1] https://en.wikipedia.org/wiki/Bubble_column_reactor.

[2] https://en.wikipedia.org/wiki/Fluidized_bed_reactor.

[3] https://en.wikipedia.org/wiki/Fluidized_bed_reactor.

[4] http://synthoglasspiperepair.com/index.php/ct-menu-item-3.html.

[5] Anthony Anderson, Xiaoming Zheng, and Vittorio Cristini. Adaptive unstructured volume remeshing-I: The method. *Journal of Computational Physics*, 208(2):616 – 625, 2005.

[6] Douglas N. Arnold, Franco Brezzi, Bernardo Cockburn, and L. Donatella Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002.

[7] Blanca AyusodeDios and Ludmil Zikatanov. Uniformly convergent iterative methods fordiscontinuous galerkin discretizations. *Journal of Scientific Computing*, 40(1-3):4–36, 2009.

[8] Néstor Balcázar, Lluís Jofre, Oriol Lehmkuhl, Jesús Castro, and Joaquim Rigola. A finite-volume/level-set method for simulating two-phase flows on unstructured grids. *International journal of multiphase flow*, 64:55–72, 2014.

[9] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.*, 100(2):335–354, June 1992.

[10] John C. Butcher. A multistep generalization of runge-kutta methods with four or five stages. *J. ACM*, 14(1):84–99, January 1967.

[11] Anurag Chakraborty. An extension of weiler-atherton algorithm to cope with the self-intersecting polygon. *CoRR*, abs/1403.0917, 2014.

[12] J Chiva, J Ventosa, O Lehmkuhl, and CD Pérez-Segarra. Modelization of the low-mach navier stokes equations in unstructured meshes. *Proceedings of the 7th International conference on computational heat and mass transfer*, 2011.

[13] Bernardo Cockburn and Chi-Wang Shu. The runge–kutta discontinuous galerkin method for conservation laws v: multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, 1998.

[14] Olivier Desjardins, Vincent Moureau, and Heinz Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, 227(18):8395 – 8416, 2008.

[15] Olivier Desjardins, Vincent Moureau, and Heinz Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, 227(18):8395–8416, 2008.

[16] A Runge-Kutta Discontinuous-Galerkin. Level-set method for unsteady com-pressible two-fluid flow.

[17] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83 – 116, 2002.

[18] Allen Van Gelder. Efficient computation of polygon area and polyhedron volume. In Alan W. Paeth, editor, *Graphics Gems V*, pages 35–41. Academic Press, 1995.

[19] FX Giraldo, Jan S. Hesthaven, and T. Warburton. Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *Journal of Computational Physics*, 181(2):499–525, 2002.

[20] J. Grooss and J.S. Hesthaven. A level set discontinuous galerkin method for free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 195(25-28):3406–3429, 2006.

[21] Francis H. Harlow and J. Eddie Welch. Numerical calculation of timedependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(12), 1965.

[22] Daniel Hartmann, Matthias Meinke, and Wolfgang Schrder. Differential equation based constrained reinitialization for level set methods. *Journal of Computational Physics*, 227(14):6821 – 6845, 2008.

[23] J. S. Hesthaven and T. Warburton. Discontinuous Galerkin methods for the time-domain Maxwell's equations. *ACES Newsletter*, 19:10–29, 2004.

[24] C.W Hirt and B.D Nichols. Volume of fluid VoF method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201 – 225, 1981.

[25] Barth T. J. and Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA Paper*, 89-0366, 1989.

[26] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202 – 228, 1996.

[27] Lluıs Jofre, Oriol Lehmkuhl, Jesús Castro, and Assensi Oliva. A plic-vof implementation on parallel 3d unstructured meshes. *Proc. of V European Conference on Computational Fluid Dynamics, ECCOMAS CFD*, pages 1–15, 2010.

[28] Lluís Jofre, Oriol Lehmkuhl, Jesús Castro, and Assensi Oliva. A 3-d volume-of-fluid advection method based on cell-vertex velocities for unstructured meshes. *Computers & Fluids*, 94:14–29, 2014.

[29] Ron Kimmel. The osher-sethian level set method. In *Numerical Geometry of Images*, pages 50–60. Springer New York, 2004.

[30] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, and J.E. Flaherty. Shock detection and limiting with discontinuous galerkin methods for hyperbolic conservation laws. *Applied Numerical Mathematics*, 48(34):323 – 338, 2004. Workshop on Innovative Time Integrators for {PDEs}.

[31] Lilia Krivodonova. Limiters for high-order discontinuous galerkin methods. *Journal of Computational Physics*, 226(1):879 – 896, 2007.

[32] Dmitri Kuzmin. A vertex-based hierarchical slope limiter for -adaptive discontinuous galerkin methods. *Journal of Computational and Applied Mathematics*, 233(12):3077 – 3085, 2010. Finite Element Methods in Engineering and Science (FEMTEC 2009).

[33] Hong Luo, Joseph D. Baum, and Rainald Lhner. A discontinuous galerkin method based on a taylor basis for the compressible flows on arbitrary grids. *Journal of Computational Physics*, 227(20):8875 – 8893, 2008.

[34] Emilie Marchandise, Jean-Francois Remacle, and Nicolas Chevaugeon. A quadrature-free discontinuous galerkin method for the level set equation. *Journal of Computational Physics*, 212(1):338 – 357, 2006.

[35] Jeremy O. Mccaslin and Olivier Desjardins. A localized re-initialization equation for the conservative level set method. *J. Comput. Phys.*, 262:408–426, April 2014.

[36] Elin Olsson and Gunilla Kreiss. A conservative level set method for two phase flow. *Journal of Computational Physics*, 210(1):225 – 246, 2005.

[37] Elin Olsson and Gunilla Kreiss. A conservative level set method for two phase flow. *J. Comput. Phys.*, 210(1):225–246, November 2005.

[38] Elin Olsson, Gunilla Kreiss, and Sara Zahedi. A conservative level set method for two phase flow {II}. *Journal of Computational Physics*, 225(1):785 – 807, 2007.

[39] Elin Olsson, Gunilla Kreiss, and Sara Zahedi. A conservative level set method for two phase flow ii. *J. Comput. Phys.*, 225(1):785–807, July 2007.

[40] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.

[41] Mark Owkes and Olivier Desjardins. A discontinuous galerkin conservative level set scheme for interface capturing in multiphase flows. *J. Comput. Phys.*, 249:275–302, September 2013.

[42] Mark Owkes and Olivier Desjardins. A discontinuous galerkin conservative level set scheme for interface capturing in multiphase flows. *J. Comput. Phys.*, 249:275–302, September 2013.

[43] F. Raees, D. R. van der Heul, and C. Vuik. A mass-conserving level-set method for simulation of multiphase flow in geometrically complicated domains. *International Journal for Numerical Methods in Fluids*, pages n/a–n/a, 2015. fld.4188.

[44] Thomas Révész. Clipping polygons with sutherland-hodgman's algorithm. *C Users J.*, 11(8):23–34, August 1993.

[45] W. Rider and D. Kothe. A marker particle method for interface tracking. *Proceedings of the Sixth International Symposium on Computational Fluid Dynamics*, pages 976–981, 1995.

[46] D.L. Sun and W.Q. Tao. A coupled volume-of-fluid and level set (voset) method for computing incompressible two-phase flows. *International Journal of Heat and Mass Transfer*, 53(4):645 – 655, 2010.

[47] Daniel Sunday. Fast polygon area and newell normal computation. *Journal of Graphics Tools*, 7(2):9–13, 2002.

[48] Mark Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187(1):110 – 136, 2003.

[49] Mark Sussman and Elbridge Gerry Puckett. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162(2):301–337, 2000.

[50] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169(2):708 – 759, 2001.

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

[52] S.P. van der Pijl. Computation of bubbly flows with a mass-conserving level-set method. *PhD Thesis, TU Delft, The Netherlands*, 2005.

[53] S.P. van der Pijl, A. Segal, C. Vuik, and P. Wesseling. Computing three-dimensional two-phase flows with a mass-conserving level-set method. *Comput. Visual Sci.*, 11:221–235, 2008.

[54] V. Venkatakrishnan. Convergence to steady state solutions of the euler equations on unstructured grids with limiters. *Journal of Computational Physics*, 118(1):120 – 130, 1995.

[55] Tomasz Wacawczyk. A consistent solution of the re-initialization equation in the conservative level-set method. *Journal of Computational Physics*, 299:487 – 525, 2015.

[56] Y. Wang, S. Simakhina, and M. Sussman. A hybrid level set-volume constraint method for incompressible two-phase flow. *Journal of Computational Physics*, 231(19):6438 – 6471, 2012.

[57] Chi wang Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, volume 1697 of *Lecture Notes in Mathematics*, pages 325–432. Springer, 1998.

[58] G.D. Weymouth and D.K.-P. Yue. Conservative volume-of-fluid method for free-surface simulations on Cartesian-grids. *Journal of Computational Physics*, 229(8):2853 – 2865, 2010.

[59] Nazeeruddin Yaacob and Bahrom Sanugi. A fifth -order five-stage rk-method based on harmonic mean. In E. van Groesen and E. Soewono, editors, *Differential Equations Theory, Numerics and Applications*, pages 381–389. Springer Netherlands, 1997.

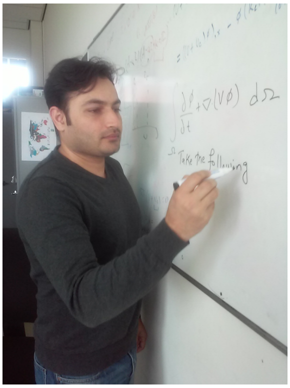
[60] Xiaofeng Yang, Ashley J. James, John Lowengrub, Xiaoming Zheng, and Vittorio Cristini. An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *Journal of Computational Physics*, 217(2):364 – 394, 2006.

[61] Steven T Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31(3):335 – 362, 1979.

[62] H.W. Zheng. A new limiter for hybrid grid. *Procedia Engineering*, 67:430 – 437, 2013. 7th Asian-Pacific Conference on Aerospace Technology and Science, 7th {APCATS} 2013.

[63] Xiaoming Zheng, John Lowengrub, Anthony Anderson, and Vittorio Cristini. Adaptive unstructured volume remeshing-II: Application to two- and three-dimensional level-set simulations of multiphase flow. *Journal of Computational Physics*, 208(2):626 – 650, 2005.

About the Author



Fahim Raees was born on 2nd December 1984 in Karachi, Pakistan. He obtained his High School Certification with Pre-Engineering in 2002 from D.J. Sindh Government Science college, Karachi. After that, he joined the Department of Mathematics of University of Karachi, Pakistan to pursue B.Sc. Honors in Mathematics. He graduated in 2005 with the first division. After that, he completed his M.Sc. in applied Mathematics from the same university and graduated in 2006.

He started working as a Management Trainee Officer in one of the leading banks in Pakistan. In the beginning of 2008, he joined NED University of Engineering and Technology, Karachi, as a lecturer in the department of Mathematics and Basic Sciences, now known as Department of Mathematics. He worked there for three years and taught different undergraduate courses. He was awarded a fully funded faculty development scholarship to pursue Ph.D. studies abroad.

He joined Delft University of Technology, Delft, the Netherlands in December 2010. He started his doctoral research in the Numerical Analysis group of Delft Institute of Applied Mathematics under the supervision of Prof. dr. ir. Kees Vuik and assistant prof. dr. ir. Duncan van der Heul. He worked on the development of the efficient and robust method for the simulation of the two-phase flow problems over complex domains. He selected hybrid approach to combined the characteristics of the Level-Set and the Volume of Fluid method. He got many opportunities to present his Ph.D. research work on international conferences like ECCOMAS conference, 2012 in Vienna, Austria and the CFD meeting of APS, 2013 in Pittsburgh, Pennsylvania, USA.

List of publications

Journal Publications

- F. Raees, D. R. van der Heul, and C. Vuik, A mass-conserving level-set method for simulation of multiphase flow in geometrically complicated domains. *International Journal for Numerical Methods in Fluids*, 2015. fld.4188.

Abstract and Conference Proceedings

- F. Raees, D. R. van der Heul, and C. Vuik, Extension of the Mass-Conserving Level-Set method to unstructured polyhedral control volumes for two-phase flows, *Bulletin of the American Physical Society, 66th Annual Meeting of the APS Division of Fluid Dynamics*, Volume 58, Number 18, 2013. Pittsburgh, Pennsylvania, USA.
- F. Raees, D. R. van der Heul, and C. Vuik, The first step towards extension of the Mass-Conserving Level-Set method to discretisations using general polyhedral control volumes,*Proceedings of the 6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012)*, Vienna, Austria.

Technical Reports

- Evaluation of the interface-capturing algorithm of OpenFOAM for the simulation of incompressible immiscible two-phase flow, *Technical* *report no. 11-07, DIAM, 2011, TU Delft.*

Presentations

- Research work presented at the 66th American Physical Society division of Computational Fluid Dynamics meeting, Pittsburgh, Pennsylvania, USA, 2013.
- Invited speaker at the Contactgroup Computational Fluid Dynamics meeting, Deltares, Delft, The Netherlands, 2013.
- Research work presented at the 6th ECCOMAS 2012 conference, University of Vienna, Vienna, Austria.
- Tea Talks, Numerical Analysis group of DIAM, TU Delft.
- Poster presented at 36th, 37th, 38th, 39th Woudschoten Conference, held in Zeist, (2011-2014), The Netherlands.

Acknowledgements

First of all, I thank Allah (S.W.T) for having me this opportunity and guiding me in my low times during the course of this Ph.D. thesis.

This thesis is not possible without the help of many people; some of them are directly involved in the research and some are in the form of good company. I would like to use this opportunity to thank all of you. However, I think it is not fair not to mention the name of those who have had helped, encouraged and guided me, during the period of my Ph.D. research.

I thank my promotor prof.dr.ir. Kees Vuik for his guidance. He always advised me to use a simple approach to sorting out/debug the big problem. I have found him really hard working and sincere person. I have learned a lot from him, including, how to do research and the importance of the research network.

I am also lucky to have a co-promotor dr.ir. Duncan Roy Van der Heul. He always guided me in the right research direction. The discussion with him over the research problems always leads to some new inside and provides a better understanding of the problem. I thank him for his help and also critically reviewing my manuscripts. I have learned a lot from him about the scientific writing.

I highly acknowledge NED University, Karachi, Pakistan for awarding me faculty development scholarship, without this scholarship this thesis is impossible to complete and I also thank TU Delft for partial funding my project.

I acknowledge my former office mates, Dr. Ibrahim Daud and Dr. Abdul Hanan Sheikh and current office mates Renaldo Astudillo, Behrouz Raftari and Baljinnyam Sereeter for providing me lively and friendly office environment. I will never forget the memorable time, we have spent in our office.

I am greatful to have an opportunity to work with nice and hard working people of the Numerical Analysis group. I like to thank Guido Oud, Manuel van Baumann, Menel Rehrah, Dr. Denis de Ouden, Dr. Jing Zhao, Dr. Xiaozhou Li, Dr. Rohit Gupta, Dr. Yue Qiu, Peiyao Luo, Muhamed Al Abbassi, Luis Antonio, Diaz Gabriela, Zaza van der Have, Fei Cong, Lisa Wobbes, Virginia Roosi, Dr. Faraz Khatami and Thea Vuik for there sincere cooperation and nice working environment. I specially thank to Thea Vuik for translating my propositions into Dutch. I also like to thank my friend from the Mathematical Analysis group, TU Delft, Rajab Ali Malookani, Dr. Sajad Sandilo, Dr. Haider Shah and Nick Gaiko.

I want to say special thank to my colleague and friend Joost van Zwieten for his help during the research and also pursuing me to use Linux. I always have a good discussion with you over research problems and also say my thank to Leonie Kah for having me over dinner.

I have enjoyed my stay in Delft due to the joys company of my friends from Pakistan. I used to have BBQ with them and play cricket. Their company always keep me connected with my homeland. I like to thank, first the former Pakistani colleagues; Dr. M Faisal Nadeem, Dr. Seyab Khan, Dr. Fakhar Anjum, Dr. Shah Muhammad, Dr. Hamayun Zai, Dr. Iftikhar Faraz, Dr. Shahzad Gishkori, Dr. Adeel Javaid and Kazmi Sahab for providing me great company and guidance. I am also grateful to have young Pakistani friends to whom with, I enjoyed delicious food and play cricket a lot; Dr. Imran Ashraf, Dr. Waqas Hassan, Dr. Hamid Mushtaq, Nauman Ahmed, A Nadeem Tabish, Qasim Lone, Hussam Khaliq, Muhammad Farooq, Muneeb Yousaf, Muhammad Irfan, Saad Jahangir, Salman Khan, Noor Nabi Mujahid.

I cannot forget the memorable time spent with Sadar Usama Malik and Umar Ijaz Chaudrey. Thank you friends for your memorable company.

I thank my ex-house mate and best friend Devender Maheshwari and his wife Rita Maheshwari for providing a good company away from my home and special thanks to Rita for cooking nice and delicious vegetarian food.

I want to thank secretary of our group Deborah Dongor for her help in administrative matters. You have helped me a lot. I also thank Franca Post from CICAT for her help in organizing my initial stay in Delft and administrative matters. I also acknowledge the time to time help of Kees Lemmens for ICT support. You have a good command on Linux.

I also like to thank faculty members of my group Prof.dr.ir Martin van

Gijzen, Dr.ir. Fred Vermolen, Dr. Jennifer Ryan, Dr.ir. Domenico Lahaye, Dr. N.V. Budko, Ir. A. Segal and Ir. P. Sonneveld for their lively conversation over a range of topics.

I thank my friends Charlie van Schoor and Anna, Albert, JP, Rubin and Andrew for having a good company over BBQ many times at my place. I also thank my current house mates for maintaining a friendly atmosphere in our student house, Gijs, Jorn, Sharat, Tim, Micheal and Ingeborg.

I am grateful to have NED University colleagues, Tauqeer Ahmed Hashmi, Sohail Ahmed, Zakir Khan, Umar Faryaz, Muhammad Yousuf Tufail and Dr. Bilal Ahmed Usmani. I especially convey my thank to Prof.dr. Mirza Mahmood Baig for their administrative support during my stay aboard.

In last, I like to thank my parents for their continuous encouragement and motivation during my stay aboard. I cannot be able to do anything without your best wishes and prayers, thank you very much. May Allah bless you all!