

# Development of a Finite Volume Solver for Two-phase Incompressible Flows using a Level Set Method

by Tobias Martin

Matr.Nr: 211203603



UNIVERSITÄT ROSTOCK

Master's thesis

Master of Science Ship Technology and Ocean Engineering

At the Chair of Modelling and Simulation

Prof. Dr.-Ing. habil. Nikolai Kornev

Date of submission: 22.09.2016



Task formulation of the Master's thesis  
Development of a two-phase incompressible Navier-Stokes solver using  
a level set method  
of Mr. Tobias Martin

The thesis shall include the following steps:

1. Continuation of the development of discretisation schemes based on WENO reconstructions
  - Handling of arbitrary meshes in two and three dimensions has to be provided as well as parallelisation of the code
  - Efficient code structuring with a partition in preprocessing and runtime steps
  - Integration of the schemes in the framework of OpenFOAM
  - Development of interpolation schemes with bounded solutions and efficient flux evaluation
  - Preparation of additional schemes, such as gradient calculations
2. Development of a mass-conservative level set method using high-order schemes
  - Studying existing level set approaches
  - Derivation of the method and coupling with an appropriate algorithm for solving the incompressible Navier-Stokes equations
  - Minimising the restriction to the CFL-condition
3. Application of the developed solver to incompressible two-phase flows
  - Several typical cases in the field of marine technology should be included
  - Comparison with existing solvers and emphasising the strengths and weaknesses

Ivan Shevchuk, 11.04.2016

## Preface

This thesis aims at extending the development of high-order non-oscillatory schemes of the previous project and the derivation of a new solver for incompressible two-phase flows using a conservative level set method. The WENO reconstruction method handles arbitrary meshes in two and three dimensions. The flux evaluation of the final semi-implicit interpolation scheme is based on the theory of hyperbolic partial differential equations and Riemann problems. In this connection, an alternative perspective of default convection schemes is enabled. The developed reconstruction is additionally used for the creation of gradient schemes with a superior behaviour on unstructured grids. All schemes are embedded in the code structure of OpenFOAM resulting in the applicability to any corresponding term in the high-level programming. Several verification cases show the improved accuracy of the WENO approach, in particular, on unstructured grids due to a mapping of the stencil to a reference space without scaling effects.

The final two-phase flow solver benefits from this superior accuracy and moreover, prevents spurious oscillations of the interface which might occur if compressive volume of fluid methods are applied to simulations with specific mesh topology. For this purpose, a relaxation equation with a novel diffusion coefficient is derived.

An efficiency analysis indicates an increased computational effort of high-order schemes in comparison to common high resolution schemes. Several basic approaches for accuracy and performance improvements are outlined at the end of the thesis.

## **Acknowledgement**

I would like to thank my second supervisor Mr. Ivan Shevchuk for the helpful notes and active discussions. He has advised me since almost three years and always been a patient discussion partner to me. In addition, I want to express my gratitude to the Chair of Modelling and Simulation of Professor Nikolai Kornev for providing the necessary computational resources. My final thanks go to Professor Carl Ollivier-Gooch (University of British Columbia) and Dr. Holger Marschall (Technical University of Darmstadt) for informative email exchanges.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview of Existing Interface Models . . . . .	2
1.2	Outline of the Thesis . . . . .	6
<b>2</b>	<b>Governing Equations and Finite Volume Discretisation</b>	<b>7</b>
2.1	Derivation of the Relevant Equations . . . . .	7
2.2	Continuum Surface Force Model . . . . .	10
2.3	Discretisation of a Transport Equation . . . . .	13
2.3.1	Convection Term . . . . .	14
2.3.2	Diffusion Term . . . . .	25
2.3.3	Source Term . . . . .	26
2.3.4	Temporal Discretisation . . . . .	27
<b>3</b>	<b>Implementation of High-Order Non-Oscillatory Schemes</b>	<b>29</b>
3.1	WENO Reconstruction Method . . . . .	32
3.1.1	Stencil Collection Algorithm . . . . .	36
3.1.2	Generation of Sectoral Stencils . . . . .	40
3.1.3	Solving the System of Equations . . . . .	42
3.1.4	WENO Smoothness Indicator . . . . .	44
3.1.5	Parallelisation . . . . .	46
3.2	Details of Implementation . . . . .	48
3.3	Semi-Implicit WENO Interpolation Scheme for Convective Terms . . . . .	51

3.3.1	Bounded Version . . . . .	54
3.3.2	Application to the Advection Equation . . . . .	56
3.4	WENO Gradient Scheme . . . . .	63
3.4.1	Surface Normal Gradient Extension . . . . .	66
3.5	High-Order Temporal Discretisation Using ADER . . . . .	68
<b>4</b>	<b>A Two-Phase Flow Solver Using a Level Set Method</b>	<b>71</b>
4.1	Derivation of the Interface Model . . . . .	71
4.2	Solution of the Incompressible Navier-Stokes Equations . . . . .	77
4.3	Application and Limitations . . . . .	84
<b>5</b>	<b>Development of a Conservative Two-Phase Flow Solver</b>	<b>89</b>
5.1	Principal Idea . . . . .	89
5.2	New Conservative Level Set Method . . . . .	92
5.2.1	Calculation of the Limiter Factors . . . . .	94
5.2.2	Derivation of the Compression Fluxes . . . . .	97
5.2.3	Definition of the Diffusion Coefficients . . . . .	101
5.3	Application to the Rotation of Zalesak's Disk . . . . .	103
<b>6</b>	<b>Physical Applications and Performance</b>	<b>107</b>
6.1	Motion of Two Superposed Viscous Fluids . . . . .	108
6.2	Two-Dimensional Breaking of a Dam . . . . .	111
6.3	Three-Dimensional Breaking of a Dam . . . . .	115
6.4	Wigley Hull . . . . .	118
6.5	Evaluation of the Performance . . . . .	122
<b>7</b>	<b>Conclusion</b>	<b>127</b>
<b>Bibliography</b>		<b>133</b>

# List of Figures

1.1	Sketch of described interface methods . . . . .	6
2.1	Illustration of the Riemann problem for the linear advection equation . .	19
2.2	Illustration of the TVD regions of Sweby . . . . .	24
3.1	Selection of the vertices for the definition of the reference system . . . .	38
3.2	Layers of cells added at each iteration . . . . .	39
3.3	Definition of sectoral stencils . . . . .	41
3.4	Mapping of a triangular face to the first octant . . . . .	42
3.5	Stencil collection algorithm at processor boundaries . . . . .	48
3.6	File structure of the WENO reconstruction implementation . . . . .	48
3.7	Setup for the two-dimensional transport of prescribed profiles . . . . .	57
3.8	Slice of the <i>ellipse</i> -profile at $x = 0.5 \text{ m}$ . . . . .	58
3.9	Slice of the <i>step</i> -profile at $x = 0.5 \text{ m}$ . . . . .	59
3.10	Setup for the rotation of Zalesak's disk . . . . .	60
3.11	Results for Zalesak's disk on $128 \times 128$ squares . . . . .	61
3.12	Results for Zalesak's disk on $256 \times 256$ squares . . . . .	62
3.13	Results for Zalesak's disk on $128 \times 128$ triangles . . . . .	62
3.14	Results for Zalesak's disk on $256 \times 256$ triangles . . . . .	63
3.15	Results of the gradient calculation in the principal directions . . . . .	66

4.1	Results of the level set method for Zalesak's disk . . . . .	85
4.2	Setup for the collapse of a water column . . . . .	87
4.3	Results for the collapse of a water column on $150 \times 30$ squares . . . . .	88
5.1	Compression effect of the $\alpha$ -profile . . . . .	99
5.2	Results for Zalesak's disk on $128 \times 128$ squares . . . . .	104
5.3	Results for Zalesak's disk on $256 \times 256$ squares . . . . .	105
5.4	Results for Zalesak's disk on $128 \times 128$ triangles . . . . .	105
5.5	Results for Zalesak's disk on $256 \times 256$ triangles . . . . .	106
6.1	Setup for the motion of two superposed viscous fluids . . . . .	109
6.2	Amplitude of the wave at $x = 0\text{ m}$ over time . . . . .	110
6.3	Setup for the two-dimensional breaking of a dam . . . . .	111
6.4	Considered meshes . . . . .	112
6.5	Resulting contours $\alpha = 0.5$ on the mixed grid . . . . .	113
6.6	Resulting contours $\alpha = 0.5$ on the structured grid . . . . .	114
6.7	Setup for the three-dimensional breaking of a dam . . . . .	115
6.8	Slice of the mesh at $y = 0.5\text{ m}$ . . . . .	116
6.9	Measured water column heights over time . . . . .	117
6.10	Setup for the wigley hull case . . . . .	119
6.11	Slice of the mesh with 250,172 cells at $y = 0\text{ m}$ . . . . .	119
6.12	Resulting water surface using <code>interFoam</code> . . . . .	121
6.13	Resulting water surface using <code>clsMULESFoam</code> . . . . .	121
6.14	Longitudinal wave cut along the hull . . . . .	122
6.15	Elapsed time per cell for one time step using <code>interFoam</code> . . . . .	125
6.16	Percentage of single sub-steps of a WENO reconstruction . . . . .	126

# List of Tables

6.1	Overview of applied discretisation schemes . . . . .	107
6.2	$\mathcal{L}_2$ -norm of the errors for the measured data . . . . .	118
6.3	Force coefficients of converged solutions . . . . .	120
6.4	Time measurements on 4 processor for wigley hull and three-dimensional breaking of a dam using <code>clsMULESFoam</code> . . . . .	123
6.5	Time measurements on 4 processor for wigley hull and three-dimensional breaking of a dam using WENO reconstructions . . . . .	124



# List of Abbreviations

<b>ADER</b>	Arbitrary high order schemes using Derivatives
<b>CFL</b>	Courant-Friedrichs-Lowy number
<b>CLS</b>	Conservative Level Set
<b>CSF</b>	Continuum Surface Force model
<b>FDM</b>	Finite Difference Method
<b>FMM</b>	Fast Marching Method
<b>FVM</b>	Finite Volume Method
<b>GAMG</b>	Geometric-Algebraic Multi-Grid method
<b>IC</b>	Initial Condition
<b>LES</b>	Large Eddy Simulation
<b>MPI</b>	Message Passing Interface
<b>MULES</b>	Multidimensional Universal Limiter for Explicit corrected implicit Solutions
<b>PCG</b>	Preconditioned Conjugate Gradient method

<b>PDE</b>	Partial Differential Equation
<b>PISO</b>	Pressure Implicit with Splitting Operators method
<b>RBF</b>	Radial Basis Functions
<b>SVD</b>	Singular Value Decomposition
<b>TVD</b>	Total Variation Diminishing
<b>URANS</b>	Unsteady Reynolds Averaged Navier-Stokes
<b>VOF</b>	Volume Of Fluid
<b>(W)ENO</b>	(Weighted) Essentially Non Oscillatory

# List of Symbols

$a_k$	degrees of freedom [-]
$\tilde{a}_k$	modified degrees of freedom [-]
$c_d$	total drag coefficient [-]
$c_f$	friction coefficient [-]
$c_p$	dynamic pressure coefficient [-]
$\vec{d}$	distance vector between two adjacent cells [m]
$d_m$	linear weight [-]
$\vec{g}$	gravitational acceleration vector $\left[\frac{m}{s^2}\right]$
$\vec{n}_l$	normal vector of a face [-]
$\tilde{\vec{n}}_l$	interface normal vector [-]
$p$	pressure $\left[\frac{kg}{m \cdot s^2}\right]$
$p_i$	polynomial representation of a field variable in a cell [-]
$r$	order of the polynomial [-]
$\vec{u}$	velocity field $\left[\frac{m}{s}\right]$

---

$u_{n_l}$	velocity normal to a face $\left[\frac{m}{s}\right]$
$u_{r,l}$	compression velocity at a face $\left[\frac{m}{s}\right]$
$w_{CD}$	central differencing weight [-]
$\vec{x}$	physical coordinates [-]
$\vec{x}_l$	face centre coordinate [-]
$\vec{x}_{P,N}$	cell centre coordinate of owner and neighbour of a face [-]
$\mathcal{A}$	reconstruction matrix [-]
$\mathcal{A}^+$	Moore-Penrose pseudoinverse [-]
$\mathcal{B}$	oscillation indicator matrix [-]
$\mathcal{D}$	strain rate tensor $\left[\frac{1}{s}\right]$
$F_l$	face of a control volume [-]
$ F_l $	area of the face $[m^2]$
$Fr$	Froude number [-]
$\mathcal{I}$	identity matrix [-]
$\mathcal{I}_S$	WENO smoothness indicator [-]
$\mathcal{J}$	Jacobian matrix [-]
$J_{max}$	number of cells in each stencil [-]
$K$	required number of degrees of freedom [-]
$L_i$	number of faces of a cell [-]

---

$Ma$	Mach number [-]
$N$	total number of cells in domain [-]
$Pe$	Peclet number [-]
$Q_l$	volumetric flow rate $\left[\frac{m^3}{s}\right]$
$Re$	Reynolds number [-]
$S$	stencil [-]
$\vec{S}$	generic source term $\left[\frac{kg}{m^2 \cdot s^2}\right]$
$S_w$	wetted surface $[m^2]$
$V_i$	control volume [-]
$V'_i$	control volume in the reference space [-]
$ V_i $	volume of a cell $[m^3]$
$\vec{X}$	reference space of sectors [-]
$\alpha$	volume fraction [-]
$\tilde{\alpha}$	smoothed volume fraction [-]
$\kappa$	mean surface curvature [-]
$\lambda$	characteristic speed $\left[\frac{m}{s}\right]$
$\mu$	dynamic viscosity $\left[\frac{kg}{m \cdot s}\right]$
$\nu$	kinematic viscosity $\left[\frac{m^2}{s}\right]$
$\vec{\xi}$	reference space of stencils [-]

---

$\vec{\xi}_i$	cell centre coordinate in the reference space [-]
$\rho$	density $\left[ \frac{kg}{m^3} \right]$
$\sigma$	surface tension $\left[ \frac{kg}{s^2} \right]$
$\varphi$	scalar level set [-]
$\omega_m$	WENO weight [-]
$\Gamma$	geometrical surface [-]
$\Sigma$	Cauchy stress tensor $\left[ \frac{kg}{m \cdot s^2} \right]$
$\Phi$	conservative variable [-]
$\Phi_l$	conservative variable interpolated to the face [-]
$\Phi_N$	conservative variable of the neighbour cell [-]
$\Phi_P$	conservative variable of the owner cell [-]
$\bar{\Phi}$	cell-averaged value of a conservative variable [-]
$\Psi_k$	orthogonal basis function [-]
$\Omega_k$	basis function [-]
$\partial V_i$	boundary surface of control volume [-]
$\nabla$	divergence operator [-]
$\otimes$	outer product [-]





# Chapter 1

## Introduction

Multiphase flows frequently arise in nature and industrial systems. The correct simulation of the interfaces separating the different fluids is a key knowledge in mixing processes, atomization and hydrodynamics. Even if restricting to two-phase flows, multiple topological specifications have to be captured by a single interface model. In the field of marine technology and free-surface flows the class of interface phenomena range from large-scale waves of small amplitude to splashing with coalescence and breakup situations. This variety of effects reveals the design of tools for two-phase flow applications as a difficult task [72].

In the recent history, several approaches from different perspectives and with varying intentions were proposed. An overview of their ideas and differences is presented below. The discussion will show that the treatment of two-phase flows is, as most parts of the computational fluid dynamics (CFD), still under development in fundamental issues and needs further attention in the nearer future [72].

The challenge arising from most interface models relates to physical discontinuities of the fluid properties at the interface. Low-order discretisation techniques lead to a large amount of numerical diffusion, whereas high-order methods produce oscillatory and non-physical results. As a consequence, the development of high-order (weighted)

essentially non-oscillatory ((W)ENO) schemes is a major step towards the representation of sharp interfaces in the surroundings of stable solutions.

## 1.1 Overview of Existing Interface Models

The common classification of numerical methods accounting for the presence of the interface in the computational domain distinguishes between explicit and implicit transport of the interface [70].

### Explicit Methods

Explicit interface methods are based on a Lagrangian approach whose equations describe the motion of an individual fluid parcel in space and time. The so-called moving mesh method is the most fundamental representative since the pure Lagrangian transport of the fluid properties as well as the interface grid nodes according to the underlying velocity field are computed. Successful and accurate applications could be shown for the simulation of small amplitude motions of the interface as small waves and weakly deformed bubbles. During less suitable simulations, the meshes can become highly distorted and algorithms for automatic addition or subtraction of grid nodes have to be introduced [70].

A more flexible approach offers the front tracking method which is originally proposed by Univerdi and Tryggvason (see [91]) and the common explicit method today. It combines a fixed Eulerian grid for the fluid flow properties with a separate Lagrangian grid as the interface. This grid consists of a set of massless marker particles typically connected to triangles and transported according to the underlying velocity field. The resulting method is highly accurate due to the transport of a permanent sharp surface and the geometrical calculation of the interface curvature from the mesh [29]. On the contrary, the interface is sensitive to the spacing between the particles which necessi-

tates dynamic re-meshing and affect the conservation of the liquid volume. Further, topological modifications as merging or breakage of interfaces have to be handled by sub-grid models which is a cumbersome task in three dimensions and in parallel [17].

## Implicit Methods

Implicit methods are the more common way of interface handling in CFD today. They avoid the requirement of a Lagrangian grid by describing surfaces as a scalar field advected in the same Eulerian grid as the flow properties. Topological changes are handled automatically and undesired deformation and reparation of surface elements can be circumvented [65]. In addition, the interface is defined as an immersed boundary and taken as the locus of capillary forces and material discontinuities expressed as jump conditions. The treatment of the appearing singularities is provided by additional modifications using the continuum surface force method or the ghost fluid method [70].

The level set method of Osher and Sethian [66] was the first implicit method. It defines the so-called level set as a signed distance function which has a smooth distribution and is zero at the interface. The function is advected using the local fluid velocities and a simple, linear hyperbolic equation called level set equation. The discretisation technique of its convection term attracts special attention due to its importance regarding the mass conservation of the method. The formulation of the level set method is inherently not mass-conservative especially in regions of significant interface deformation [29]. The application of low-order convection schemes introduce too much numerical diffusion resulting in smeared solutions and corruption of the level set to be a signed distance function. Therefore, high-order schemes are proposed for preserve the sharpness of the interface. As it will be explained in the next chapter, such schemes tend to produce oscillatory solutions whose treatment leads to the development of (W)ENO schemes [82]. Unfortunately, even low-dissipative schemes can not guarantee the conservation of the signed distance function for which reason redistancing algorithms are introduced. In

certain cases, particularly if the grid resolution is high, this leads to a mass-conservative solution but also to a shift of the interface if no special correction is applied. Several hybrid approaches were introduced in order to circumvent this drawback but benefit from the simplicity of the principal method at once. A more detailed discussion of these kind of interface models is presented in chapter 4.

In contrast, the volume of fluid method (VOF) of Nichols, Hirt, and Hotchkiss [58] is inherently mass-conservative and therefore, well established. The interface motion is described by the transport of the volume fraction using a similar equation as the level set method. The resulting scalar function has a jump from zero to one at the interface for which reason the solution of advection is prone to numerical instabilities and non-physical oscillations. The sharpness of the profile has to be ensured by applying high-order convection schemes and a geometrical reconstruction or artificial compression techniques [95].

Common geometrical reconstructions are based on the simple line or piecewise linear interface calculation (SLIC and PLIC). SLIC algorithms reconstruct the interface in each cell by a segment aligned with the grid which suffers from at best first-order of accuracy. Therefore, the development of PLIC algorithms was advanced. It reconstructs the interface in each cut cell by a plane and can reach second-order accuracy. The resulting surface is represented by a sequence of polygons with small discontinuities in between [70]. PLIC algorithms are complicated to implement on unstructured, three-dimensional grids even though the generation of a continuous interface is generally unnecessary [29].

Compressive VOF methods were developed in order to reduce the computational costs by avoiding the explicit reconstruction procedure. Instead, the sharpness of the interface is ensured by modified convection schemes or additional, numerical terms. Most approaches are related to the flux-corrected transport method of Boris and Book [5] which adds a limited amount of high-order downwind flux to a low-order upwind flux of the convection term. This anti-diffusion procedure reduces the numerical diffusion

and avoids instabilities of an unlimited compression flux at the same time. However, the performance of the approach is poor for flows with vorticity [29] and the limiter depends on the time step which proves to be cumbersome in implicit calculations.

Most compressive schemes are based on a donor-acceptor approach, such as Ub-bink's compressive interface capturing scheme for arbitrary meshes (CICSAM). In this approach, an upwind and downwind cell are assigned for each face in the domain with respect to the underlying flow field. The scheme spatially blends between a compressive downwind and a diffusive upwind differencing scheme based on the angle between the interface normal vector and the normal vector of the face. This approach often suffers from high numerical diffusion. Denner and Wachem [16] indicate the skewness of the mesh and the extrapolation of the upwind node on unstructured grids as the causing effects. For more details on this topic, see the papers of Gopala and Wachem [29] and Denner and Wachem [16].

In comparison, OpenFOAM's solver for incompressible two-phase flows `interFoam` prevents the necessity of a special scheme by adding an additional compression term to the advection equation. The creator of OpenFOAM, Henry Weller, developed the limiting strategy MULES for the modified equation with the objective of ensuring boundedness of the solution of the volume fraction. It is based on flux-corrected methods and therefore, needs low-order and high-order advection fluxes as well as a compressive fluxes. Gopala and Wachem [29] state a restriction to Courant numbers less than 0.3 for stable solutions. A sub-cycling methodology is introduced for larger time steps which on the other hand leads to higher computational cost due to several advection steps per time step. It will be shown in this thesis that `interFoam` leads to oscillatory interfaces in situations where the compression acts in wrong directions. In this connection, the development of a new interface model in OpenFOAM is desirable which, however, preserve the simplicity of most implicit approaches. The derivation of the model starts with the approach of standard level set methods and intermediate results are presented in order to emphasise the mentioned properties of surface capturing methods. Irrespective to the

chosen model, all implicit methods benefit from high-order non-oscillatory convection schemes for which reason the development of WENO schemes was initialised in the author's previous project [50] and is finished here.

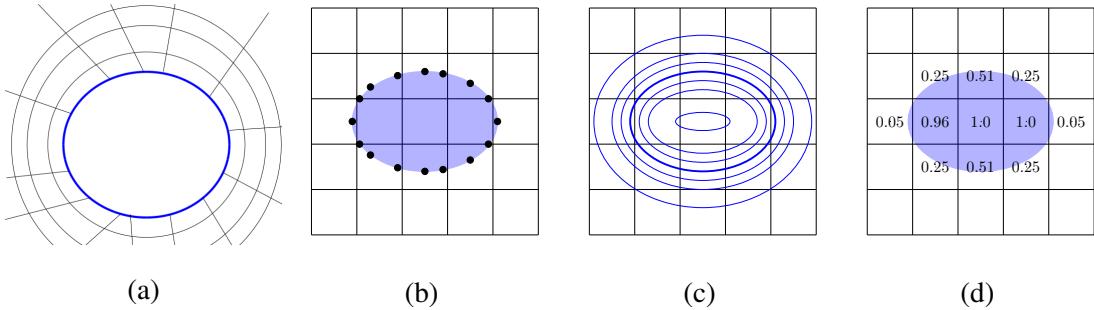


Figure 1.1: Sketch of described interface methods: (a) Moving mesh; (b) Front tracking; (c) Level set; (d) VOF

## 1.2 Outline of the Thesis

The thesis begins with the derivation of the governing equations of incompressible two-phase flows and relevant models (chapter 2). Further, the discretisation techniques of these equations are provided using finite volume methods. A special attention is turned to convection schemes due to the mentioned importance. Chapter 3 deals with the derivation of WENO reconstructions and the implementation of WENO interpolation and gradient schemes. In addition, an extension to ADER schemes will be presented at the end of the chapter. Then, a complete solver using a standard level set method is developed and applied in chapter 4. The poor performance of this approach leads to further developments in chapter 5. The application of the final solver in chapter 6 shows improved results in comparison to OpenFOAM's standard solver but also WENO scheme's lack of efficiency. Final remarks and prospects for further developments are given in chapter 7. In addition, all presented cases and implementations can be found on a disc on the last page of the thesis.

# **Chapter 2**

## **Governing Equations and Finite Volume Discretisation**

The fundamental equations for any viscous, incompressible flow arise from the conservation of mass and momentum. In the following chapter, their derivation is presented under consideration of the presence of an interface. The occurring singularities are treated by the continuum surface force method which is described in chapter 2.2. Finally, chapter 2.3 provides the explanation of the finite volume method for a generic transport equation.

### **2.1 Derivation of the Incompressible Navier-Stokes Equations for Two-Phase Flows**

The derivation of the mass and momentum conservation law is executed under the terms of incompressibility. A fluid is presumed to be incompressible if the density is time-invariant and the Mach number is very small. Further, no phase changes or miscibility of the fluids are considered here.

## Conservation of Mass

The law of conservation of mass states that the rate of increase of mass equals the rate of mass production in an arbitrary volume  $V(t)$  [101]. It results in the expression

$$\int_{V(t)} \left( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) \right) dV = 0. \quad (2.1)$$

The integrand must be zero since (2.1) holds for every  $V(t)$ . It finally yields the continuity equation for incompressible fluids

$$\nabla \cdot \vec{u} = 0. \quad (2.2)$$

In the presence of interfaces, the principle of mass conservation results in a velocity jump condition. The expression is derived under consideration of a geometrical surface  $\Gamma$  in a reference frame where  $\Gamma$  is at rest [75]. Denoting  $\tilde{\vec{n}}$  as the interface normal vector, the normal velocity in the reference frame  $u'_n$  is defined as

$$u'_n = \vec{u} \cdot \tilde{\vec{n}} - u_n, \quad (2.3)$$

with  $u_n$  the interface normal velocity in the global reference system. Taking the Rankine-Hugoniot condition into account, which states that the mass flow  $q$  from phase 1 to phase 2 must equal the opposite mass flow, an alternative expression for the conservation of mass can be written as follows [75]

$$\rho_1 u'_{n,1} = \rho_2 u'_{n,2} = q. \quad (2.4)$$

Introducing (2.3), equation (2.4) becomes in the general frame of reference assuming no phase changes ( $q = 0$ )

$$\rho_1 \cdot (\vec{u}_1 \cdot \tilde{\vec{n}} - u_n) = \rho_2 \cdot (\vec{u}_2 \cdot \tilde{\vec{n}} - u_n) = 0, \quad (2.5)$$

which equals the condition that the normal velocity is continuous across the interface

$$u_n = \vec{u}_1 \cdot \tilde{\vec{n}} = \vec{u}_2 \cdot \tilde{\vec{n}}. \quad (2.6)$$

Furthermore, it is usually assumed that the tangential component of  $\vec{u}$  is also continuous.

The sought continuous condition for the velocity at the interface  $\Gamma$  finally reads

$$[\vec{u}]_\Gamma = 0, \quad (2.7)$$

where  $[x]_\Gamma = x_2 - x_1$  is the jump notation. In contrary, the material properties of the fluid expressed by the density  $\rho$  and the dynamic viscosity  $\mu$ , in general, have a discontinuity at  $\Gamma$  which is further discussed in chapter 2.2.

## Conservation of Momentum

Recalling Newton's second law, stating that the rate of change of momentum in  $V(t)$  equals the sum of forces acting on  $V(t)$ , the conservation law of momentum yields [101]

$$\int_{V(t)} \left( \frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) \right) dV = \int_{V(t)} \vec{f}_s dV + \int_{V(t)} \vec{S} dV, \quad (2.8)$$

with  $\vec{S}$  a volumetric source term described below. The surface forces acting on the volume's boundaries  $\vec{f}_s$  are represented by the Cauchy stress tensor  $\Sigma$ . This tensor can be defined for a Newtonian fluid as [70]

$$\Sigma = 2\mu \mathcal{D} - \left( p + \frac{2}{3}\mu \nabla \cdot \vec{u} \right) \cdot \mathcal{I}, \quad (2.9)$$

with  $\mathcal{I}$  the unit tensor and  $\mathcal{D}$  the strain rate tensor expressed by

$$\mathcal{D} = \frac{1}{2} \cdot \left( \nabla \vec{u} + (\nabla \vec{u})^\top \right). \quad (2.10)$$

The incompressible Navier-Stokes equations are given under consideration of incompressibility and the fact that the equation must hold for every  $V(t)$  by [4, 101]

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) = -\nabla p + \nabla \cdot (2\mu \mathcal{D}) + \vec{S}. \quad (2.11)$$

The source term consists of the gravity as a body force and capillary forces at the interface in case of multiphase flows. It leads to the expression [75]

$$\vec{S} = \rho \vec{g} + 2\sigma \kappa \delta_\Gamma \tilde{\vec{n}}, \quad (2.12)$$

where  $\delta_\Gamma$  denotes a distribution concentrated on the interface and  $\sigma$  is the surface tension which is assumed to be constant over  $\Gamma$ . The mean curvature of the surface  $\kappa$  is calculated according to [75]

$$\kappa = -\frac{1}{2} \nabla_\Gamma \cdot \tilde{\vec{n}}. \quad (2.13)$$

Here,  $\nabla_\Gamma$  corresponds to a gradient operator restricted to  $\Gamma$ . It can be shown that (2.11) is valid for a domain containing multiple phases, also known as the whole-domain formulation. For this purpose, the same moving reference frame as for the continuity equation has to be considered. The integration of the normal component of (2.11) over a infinitesimal small volume enclosing the interface gets [70, 91]

$$[-p\mathcal{I} + 2\mu\mathcal{D}]_\Gamma \cdot \tilde{\vec{n}} = \sigma \kappa \tilde{\vec{n}}, \quad (2.14)$$

since only gradient terms remain. The resulting equation describes the balance of normal stresses and surface tension at interfaces corresponding to the well-known continuity condition of stresses at fluid boundaries [91]. Pringuey [70] indicates that the jump condition (2.14) allows balancing of singularities potentially appearing in terms containing  $\rho$  and  $\mu$  in (2.11). For the sake of completeness, Scardovelli and Zaleski [75] provide the derivation of a continuity condition of the tangential stress components.

## 2.2 Determination of Material Properties Using the Continuum Surface Force Model

In this section the numerical treatment of the capillary forces by the continuum surface force model (CSF) of Brackbill, Kothe, and Zemach [6] in conjunction with the defi-

nition of material properties is presented. The CSF model defines the effect of surface tension as a continuous and three-dimensional phenomenon instead of a local boundary condition on interfaces.

The physical nature of surface tension arises from the imbalance of molecular forces close to the interface. In the midst of the liquid, the intermolecular forces are at equilibrium since each molecule is pulled equally in all directions. At surfaces, the equilibrium is disturbed by unequal or missing attracting forces from beyond the interface. The involved molecules are pulled inwards which is counterbalanced by the resistance forces of the liquid to compression. The resulting equilibrium is reached at a minimal energy state characterized by a configuration of minimum area. Thus, surface tension results in a surface force that arises on molecules at interfaces [6, 26, 70]. The accurate modelling of this balance of normal stresses is given by the Young–Laplace equation describing the difference of capillary pressure  $p_s$  across interfaces [70]. Assuming an incompressible fluid at rest and a constant surface tension, it yields the expression [26]

$$p_s = p_2 - p_1 = \sigma \kappa, \quad (2.15)$$

which is identical to the following surface force on the interface [6]

$$\vec{f}_\Gamma(\vec{x}) = \sigma \kappa(\vec{x}) \tilde{\vec{n}}(\vec{x}). \quad (2.16)$$

In general, the discontinuity between two fluids 1 and 2 can be distinguished by some characteristic function  $\mathcal{H}$  reading

$$\mathcal{H}(\vec{x}) = \begin{cases} \mathcal{H}_1 & \text{in fluid 1} \\ \mathcal{H}_2 & \text{in fluid 2} \\ [\mathcal{H}]_\Gamma = \frac{\mathcal{H}_1 + \mathcal{H}_2}{2} & \text{at interface} \end{cases}$$

The exact definition of  $\mathcal{H}_1$  and  $\mathcal{H}_2$  depends on the considered interface model and is presented below for VOF methods. In the CSF model,  $\mathcal{H}$  is replaced by a mollified function  $\tilde{\mathcal{H}}$  varying from  $\mathcal{H}_1$  to  $\mathcal{H}_2$  smoothly over a finite thickness  $\Delta x$  near the interface.

This approximation leads to the transition of the pressure jump (2.15) to a continuous model including  $\tilde{\mathcal{H}}$ , which can be defined at cell centres and is differentiable. Hence, Brackbill, Kothe, and Zemach [6] were able to derive a volume force  $\vec{f}_\Omega$  reproducing the original surface force per unit area for a decreasing surface thickness  $\Delta x \rightarrow 0$

$$\lim_{\Delta x \rightarrow 0} \int_{\Delta\Omega} \vec{f}_\Omega(\vec{x}) \, d\Omega = \int_{\Delta\Gamma} \vec{f}_\Gamma(\vec{x}) \, d\Gamma. \quad (2.17)$$

Here,  $\Delta\Gamma$  is a portion of the interface and  $\Delta\Omega$  is a small volume of thickness  $\Delta x$  bounding  $\Delta\Gamma$  [70]. Following the derivation [6], the final formulation of the volumetric capillary force gets

$$\vec{f}_{cap} = \sigma \kappa \nabla \tilde{\mathcal{H}}, \quad (2.18)$$

replacing the second term in (2.12). Further, interface normal vectors and the curvature are now calculated from the following formulae [70]

$$\begin{aligned} \tilde{\vec{n}} &= \frac{\nabla \tilde{\mathcal{H}}}{|\nabla \tilde{\mathcal{H}}|}, \\ \kappa &= -\nabla \cdot \left( \frac{\nabla \tilde{\mathcal{H}}}{|\nabla \tilde{\mathcal{H}}|} \right). \end{aligned} \quad (2.19)$$

In the common VOF implementations, the mollified color function  $\tilde{\mathcal{H}}$  is defined as the scalar  $\alpha$  representing the volume fraction in a cell

$$\alpha = \begin{cases} 0 & \text{in fluid 1} \\ 1 & \text{in fluid 2} \\ \frac{1}{2} & \text{at interface} \end{cases}. \quad (2.20)$$

Hence, the global material properties are evaluated as mixed formulations

$$\begin{aligned} \rho &= \alpha \rho_1 + (1 - \alpha) \cdot \rho_2, \\ \mu &= \alpha \mu_1 + (1 - \alpha) \cdot \mu_2. \end{aligned} \quad (2.21)$$

As can be seen in (2.21), the original definition of  $\alpha$  in (2.20) has to be mollified in order to avoid discontinuous material properties. The resulting need for controlling the interface thickness is the main challenge of the CSF model. A thick transition region would be preferable with numerical stability in mind. On the contrary, transition zones generally smear the interface and are not physical. In addition, as it is shown by Harvie, Davidson, and Rudman [33], the CSF model tends to produce non-physical velocities near the interface arising from parasitic currents and generated in fluid regions adjacent to an interface by a discrete imbalance between the surface tension and the associated pressure gradient across the interface [70]. Their magnitude generally increases with the influence of the surface tension, so at low Weber numbers.

As an alternative method, the ghost fluid method was developed by Fedkiw et al. [23]. It captures the jump conditions across the interface with zero transition thickness leading to the simulation of sharp interfaces. For this purpose, each phase has to be extended across the interface, which is a difficult task on unstructured three-dimensional grids. Further details and an exemplary implementation on structured grids are provided by Desjardins, Moureau, and Pitsch [17].

## 2.3 Discretisation of a Transport Equation

In this chapter, an overview of the finite volume discretisation technique is presented. General explanations of this topic can also be read in the books of Ferziger and Perić [24], Hirsch [36] and Versteeg and Malalasekera [93]. Here, the main focus is placed on the discretisation of the convective term due to its importance for the solution's accuracy and the later introduction of WENO convection schemes. In view of the main part of this thesis, a generic transport equation for scalar  $\Phi$  is considered which has the form [41]

$$\frac{\partial \rho \Phi}{\partial t} + \nabla \cdot (\rho \vec{u} \Phi) = \nabla \cdot (\rho \Gamma_\Phi \nabla \Phi) + S_\Phi(\Phi). \quad (2.22)$$

The spatial domain is divided into a finite number  $N$  of control volumes  $V_i$  and the field  $\Phi$  is stored as averaged values  $\bar{\Phi}$  in the centres of mass (cell-centred approach). The finite volume methods (FVM) are applied by integrating (2.22) over  $V_i$  corresponding to [41]

$$\int_{V_i} \frac{\partial \rho \Phi}{\partial t} dV + \int_{V_i} \nabla \cdot (\rho \vec{u} \Phi) dV = \int_{V_i} \nabla \cdot (\rho \Gamma_\Phi \nabla \Phi) dV + \int_{V_i} S_\Phi(\Phi) dV. \quad (2.23)$$

Here, the time integration is neglected but will be described in chapter 2.3.4. Each volume  $V_i$  is constructed as a convex polyhedron with  $L_i$  plane boundary surfaces  $F_l$  each with an outward pointing normal vector  $n_l$ . In addition, the whole framework of OpenFOAM is based on second-order accuracy at most correlating with a linear variation of  $\Phi$  in space and time. Jasak [41] shows the resulting simplification for volume and surface integrations

$$\begin{aligned} \int_{V_i} \Phi dV &= \bar{\Phi}_i |V_i|, \\ \int_{F_l} \Phi dF_l &= \Phi_l |F_l|, \end{aligned} \quad (2.24)$$

with  $\Phi_l$  the value at the centroid of the face. The integral evaluations in (2.24) correspond to the application of an one-point Gaussian quadrature rule [56].

### 2.3.1 Convection Term

The volume integrals in (2.23) containing convection terms are evaluated by transforming them into surface integrals over the boundary of the cells  $\partial V_i$  using Gauss's theorem and splitting them into a sum of integrals over each boundary face of the discretised cell

$$\begin{aligned} \int_{V_i} \nabla \cdot (\rho \vec{u} \Phi) dV &= \int_{\partial V_i} \vec{n} \rho \vec{u} \Phi d\partial V = \sum_{l=1}^{L_i} \int_{F_l} \vec{n}_l \rho \vec{u} \Phi dF_l \\ &= \sum_{l=1}^{L_i} \vec{u}_l \vec{n}_l |F_l| (\rho \Phi)_l = \sum_{l=1}^{L_i} Q_l (\rho \Phi)_l, \end{aligned} \quad (2.25)$$

where  $Q_l$  is the volumetric flow rate through the face or, if multiplied by density, the mass flux. Applying (2.25) to the Navier-Stokes equations leads to a quasi-linear treatment of the non-linear convection term since  $Q_l$  is known from the previous time step (see also chapter 4.2). The remaining task is the determination of the face interpolated value  $(\rho\Phi)_l$  from the known cell-averaged values. In the given framework, all interpolation schemes are derived from a general formulation which is, under consideration of  $\rho$  equals one, defined as [63]

$$(\Phi)_l = \lambda_l \Phi_P + (1 - \lambda_l) \cdot \Phi_N + C_{exp}, \quad (2.26)$$

with  $P$  denoting the owner and  $N$  the neighbour of the face which are two adjacent cells. By definition, the normal vector of  $F_l$  points from the owner to the neighbour. The additional term  $C_{exp}$  stands for an explicit correction term which is involved for obtaining higher orders (see also chapter 3). Besides this correction, each scheme has to provide the weights  $\lambda$  for each face of the domain. Most schemes are further derived from a limiter class returning the weights as [63]

$$\lambda_l = \psi_l \cdot w_{CD,l} + (1 - \psi_l) \cdot pos(Q_l), \quad (2.27)$$

where  $\psi_f$  is the scheme specific limiter and  $w_{CD,l}$  is the central differencing weight given by [63]

$$w_{CD,l} = \frac{\vec{n}_l \cdot (\vec{x}_N - \vec{x}_l)}{\vec{n}_l \cdot (\vec{x}_l - \vec{x}_P) + \vec{n}_l \cdot (\vec{x}_N - \vec{x}_l)}, \quad (2.28)$$

with  $\vec{x}_{P,N}$  the coordinates of the cell centres and  $\vec{x}_l$  the coordinate of the face centre. The expression  $pos(Q_l)$  is determined from

$$pos(Q_l) = \begin{cases} 1 & \text{if } Q_l > 0 \\ 0 & \text{if } Q_l < 0 \end{cases}. \quad (2.29)$$

The performance of the presented convection and interpolation schemes respectively is reviewed on the basis of hyperbolic equations. In this way, the opportunity of comparing the later described WENO schemes with existing schemes arises. The required

equation is generated by applying splitting schemes, treating each term separately, to the discretisation of the scalar transport equation (2.22) [90]. Therefore, equation (2.22) can be rewritten as a hyperbolic conservation law for the convection process which yields the following linear advection equation in terms of physical coordinates

$$\frac{\partial \Phi}{\partial t} + u_x \frac{\partial \Phi}{\partial x} + u_y \frac{\partial \Phi}{\partial y} + u_z \frac{\partial \Phi}{\partial z} = 0. \quad (2.30)$$

The linearity is valid since (2.25) always arises in its linearised form. The finite volume representation of (2.30) gets under consideration of (2.25) [70]

$$\int_{V_i} \frac{\partial \Phi}{\partial t} dV + \sum_{l=1}^{L_i} \int_{F_l} \vec{u} \cdot \vec{n}_l \Phi dF_l = 0. \quad (2.31)$$

It is shown by Toro [90] that (2.31) can be evaluated locally in the normal direction of each face due to its rotational invariance. Due to the hyperbolic character, discontinuities may occur in the solution even if the initial data is smooth. Therefore, the treatment of the convective term is most challenging within the discretisation process and leads to an one-dimensional Riemann problem which has to be solved locally at each face

$$\left. \begin{aligned} PDE : \quad & \frac{\partial \Phi}{\partial t} + u_{n_l} \frac{\partial \Phi}{\partial n_l} = 0, \\ IC : \quad & \Phi(n_l, 0) = \Phi_0(n_l) = \begin{cases} \Phi^- & \text{if } n_l < 0 \\ \Phi^+ & \text{if } n_l > 0 \end{cases} \end{aligned} \right\}, \quad (2.32)$$

with the face flux  $u_{n_l} = \vec{u}_l \cdot \vec{n}_l$ . The initial data  $\Phi^\pm$  represent the values of  $\Phi$  from the adjacent cells at one point of the considered face (see also figure 2.1a). The surface integral in (2.31) is evaluated using a Gaussian quadrature rule of appropriate order. The solution at each Gaussian point is determined from (2.32). In case of second-order accuracy, this procedure simplifies to a single evaluation of the Riemann problem at the face centre and a multiplication of its solution by the face area.

The main characteristics of suitable convection schemes are besides accuracy, conservativeness and monotonicity. The order of accuracy of a scheme equals the nominal

order of the truncated term of the corresponding Taylor series of  $\Phi$  around the cell centres [93]. Conservativeness is ensured if the scheme is applicable to the conservative form of the equation. As illustrated by Wesseling [101], monotonicity ensures the convergence of conservative schemes to a physically correct solution, also known as entropy solution. It can be expressed as [90]

$$\max_i \left( \Phi_i^{(n+1)} \right) \leq \max_i \left( \Phi_i^{(n)} \right), \quad \min_i \left( \Phi_i^{(n+1)} \right) \geq \min_i \left( \Phi_i^{(n)} \right), \quad i = 1, \dots, N, \quad \forall n, \quad (2.33)$$

for the time step  $n$ . Equation (2.33) implies that all coefficients of the final system of equations around  $V_i$  are of the same sign as the coefficient in  $V_i$  (typically chosen as positive) [90, 93]. As a consequence, the creation of new extrema is prevented which leads to bounded solutions. The monotonicity property is particularly preferable in case of large gradients, as in front propagating problems, where the sharpness of the interface has to be preserved without creating spurious oscillations near these discontinuities. This Gibbs-like phenomenon generates  $\mathcal{O}(1)$  spurious oscillations at points of discontinuity, proportional to the size of the jump [32].

## Upwind Schemes

Godunov [27] developed the first conservative scheme with a monotonic behaviour and stated at the same time that such schemes can not exceed first-order accuracy due to their essential linearity (prove can be found in Toro's book [90]). His first-order upwind scheme is derived by solving the Riemann problem (2.32) exactly with piecewise constant initial data, which is  $\Phi^- = \Phi_P$  and  $\Phi^+ = \Phi_N$ . In figure 2.1a, the initial conditions of (2.32) are shown. The initial data has a discontinuity at the face  $n_l = 0$  resulting from the two different values of the neighbouring cells. Progressing in time, the discontinuity at  $n_l = 0$  propagates with a constant characteristic speed  $u_{n_l}$  indicated by the characteristic curve  $n_l = u_{n_l}t$  which represents the locations without changes in  $\Phi$ . Hence, the

exact solution of (2.32) is given as follows (see figure 2.1b and [90])

$$\Phi(n_l, t) = \Phi_0(n_l - u_{n_l}t) = \begin{cases} \Phi_P & \text{if } n_l - u_{n_l}t < 0 \\ \Phi_N & \text{if } n_l - u_{n_l}t > 0 \end{cases}. \quad (2.34)$$

At any time  $t > 0$ , the interpolated value  $\Phi_l$  yields at  $n_l = 0$

$$\Phi_l = \begin{cases} \Phi_P & \text{if } u_{n_l} > 0 \\ \Phi_N & \text{if } u_{n_l} < 0 \end{cases}. \quad (2.35)$$

Recalling the general interpolation formulae (2.26) and (2.27), the solution of the upwind scheme (2.35) is mimicked as a limited scheme using  $\psi_f = 0$  and  $C_{exp} = 0$ .

The upwind scheme is monotone and the solution, therefore, bounded since all waves of the opened Riemann fan are taken into account. However, the first-order accuracy introduces a huge amount of numerical diffusion resulting in smeared solutions [47]. This diffusion can be reduced by increasing the nominal order of accuracy of the scheme which needs an extension of their stencils in the upwind direction. A common representative is the second-order accurate linear upwind scheme also known as the method of Warming and Beam [99]. A simple way of derivation is the exchange of the constant field distribution in (2.32) by a linear distribution in the cells. Under consideration of the solution of the Riemann problem (2.35), the interpolation reads [47]

$$\Phi_l = \begin{cases} \Phi_P + (\vec{x}_l - \vec{x}_P) \cdot (\nabla \Phi)_P & \text{if } u_{n_l} > 0 \\ \Phi_N + (\vec{x}_l - \vec{x}_N) \cdot (\nabla \Phi)_N & \text{if } u_{n_l} < 0 \end{cases}. \quad (2.36)$$

The determination of the gradients is described in chapter 2.3.2. OpenFOAM introduces the additional term in (2.36) as an explicit correction to the first-order accurate upwind scheme. The stability restriction is given by Toro [90] as Courant number ( $Co$ ) less than 2 which allows applicability to many simulations. However, it is not strictly bounded [90] and rather useful for smooth solutions. Leveque [47] reveals this property

by analysing a modified version of the Beam-Warming scheme. As a result, an additional third-order term occurs which has a dispersive behaviour. In comparison to a dissipative behaviour, as the first-order scheme indicates, dispersion causes less smoothing of the solution but rather a temporal displacement of the oscillations from the main part of the data. The reduction of smearing may lead to the mentioned unbounded solutions.

Hence, the idea of (weighted) essentially non-oscillatory ((W)ENO) schemes arose in order to get rid of these oscillations and achieve higher order of accuracy at once. The implementation of WENO schemes were the subject of the author's previous project [50] and are considered in chapter 3.

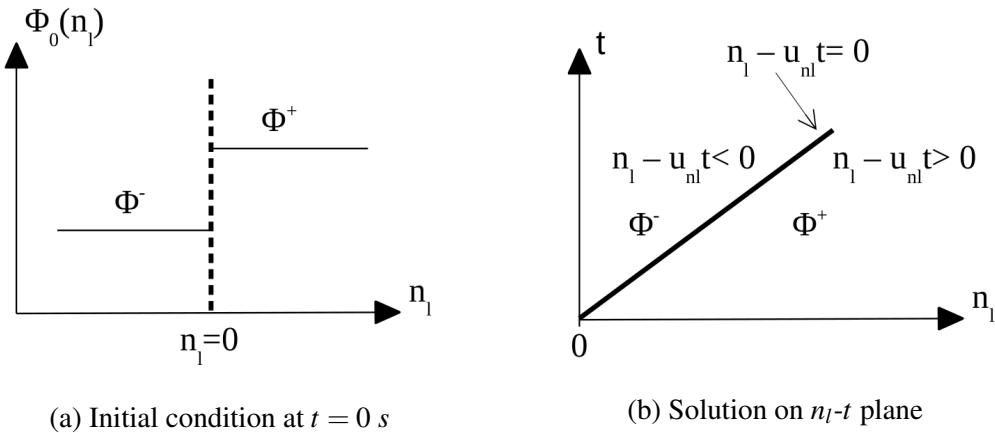


Figure 2.1: Illustration of the Riemann problem for the linear advection equation

### Central Schemes

The central differencing scheme is the simplest central scheme and provides second-order accuracy. In OpenFOAM, this linear interpolation is calculated by setting  $\psi_f = 1$  in (2.27) or rather inserting the central differencing weights (2.28) directly in (2.26) which becomes

$$\Phi_l = w_{CD,l} \Phi_P + (1 - w_{CD,l}) \cdot \Phi_N. \quad (2.37)$$

The solution is not stable at all in case of hyperbolic problems since no characteristics of the underlying Riemann problem are taken into account. However, the transient, incompressible Navier-Stokes equations are parabolic-hyperbolic equations where the dominant type can be evaluated from the Peclet number ( $Pe$ ) [93]. It is defined as the ratio of convective to diffusive fluxes and plays an important role in the later interface model (see chapter 5.2.3). The convection term dominates in case of high  $Pe$  numbers and central differencing produces non-physical over- and undershoots due to the hyperbolic character of the equation. As it is shown by Versteeg and Malalasekera [93], the solution remains bounded if  $Pe < 2$  due to the positive signs of the coefficients. Therefore, the application is restricted to convective terms in diffusion dominant flows at low Reynolds numbers ( $Re$ ), meshes with small grid-spacings [93], and the discretisation of gradients and diffusion terms.

Nevertheless, conservative central schemes can generally be beneficial for hyperbolic equations due to their simple implementation without solving a Riemann problem. The first attempt to stabilise central schemes was the Lax-Friedrichs method of Lax [45] which introduces an additional term to the central differencing scheme arising from the explicit temporal discretisation. Leveque [47] shows that this term has a diffusion-like behaviour resulting in smeared solutions. Furthermore, Hirsch [36] states that the Lax-Friedrichs method is just stable for  $Co \leq 1$  and first-order accurate. Lax and Wendroff [46], therefore, developed the Lax-Wendroff scheme which is based on a second-order central differencing in space and a first-order time discretisation resulting in a second-order accuracy in space and time [36]. As for the linear upwind scheme, Leveque's [47] analysis of a modified equation displays a third-order dispersive term and an extra fourth-order dissipative term resulting in a damping of the oscillations in practice. It might be noticed that this analysis reveals the same dissipation term for the central differencing scheme (with Euler explicit). As it is shown by Toro [90], the Lax-Wendroff scheme is not monotone leading to unbounded solutions. Therefore, Toro

[90] developed the so-called FORCE scheme [89] which evaluates the fluxes as an arithmetic mean of the Lax-Friedrichs and Lax-Wendroff fluxes. In doing so, the numerical diffusion of the Lax-Friedrichs method is reduced by the factor of two [86]. In combination with a WENO reconstruction, the FORCE scheme provides a centred, high-order and non-oscillatory method. All described attempts of stabilising the central approach does not fit in the given framework due to the combination of spatial with temporal discretisation.

### TVD Schemes

The upper discussion confirms Godunov's statement of accuracy restrictions on linear, monotone schemes. It leads to the conclusion that a high-order scheme can just be non-oscillatory near large gradients if it is non-linear even for linear equations [90]. Most effective kind of schemes fulfilling the non-oscillatory property are total variation diminishing (TVD) schemes which can reach second-order accuracy in smooth regions and often fulfil boundedness even on three-dimensional, unstructured grids [107]. For this purpose, a solution-depended, non-linear flux limiter is applied switching between a bounded first-order scheme near high gradients and an unbounded second-order scheme else [41]. By implication, TVD schemes degenerate to first-order accuracy near extrema due to the addition of artificial viscosity [64] and measuring the total variation with the help of the averaged field values in the cell centres [107].

The property of total variation diminishing for any discretised field  $\Phi$  corresponds to the non-increasing of the total variation function  $TV$  in time

$$\begin{aligned} TV(\Phi) &= \sum_l |\Phi_N - \Phi_P|, \\ TV\left(\Phi^{(n+1)}\right) &\leq TV\left(\Phi^{(n)}\right), \quad \forall n, \end{aligned} \tag{2.38}$$

with the index  $l$  running through all faces of the domain. Harten [31] proofed that all monotone schemes are TVD, but all TVD schemes are just monotonicity preserving

which is the preservation of a monotone solution in the course of time and in the same sense [90]. As a conclusion, TVD can just guarantee the convergence to the weak solution of scalar conservation laws which is possibly not the physical one [36]. Most TVD schemes, however, prevented wrong convergence by solving the Riemann problem with Godunov's monotone upwind scheme and computing correction terms as anti-diffusive fluxes on the basis of the waves arising in that Riemann solution [47]. Hence, the face flux is written as a low-order flux and a limited second-order flux [90]. The corresponding interpolation at any point of the face yields

$$\Phi_l = (\Phi_l)^{LO} + \psi_l \cdot ((\Phi_l)^{HO} - (\Phi_l)^{LO}). \quad (2.39)$$

In OpenFOAM, these schemes are implemented in the `limited` class and the comparison of (2.39) with (2.27) shows identity if  $(\Phi_l)^{LO}$  is calculated by upwind differencing and  $(\Phi_l)^{HO}$  by central differencing which correlates with the spatial discretisation of the Lax-Wendroff method [63, 90].

The limiters have to be adapted to the TVD restriction (2.38) in order to ensure non-oscillatory behaviour near discontinuities [47]. Sweby [83] transformed this restriction to several properties for the practise and developed the Sweby diagram illustrating the TVD regions of  $\psi$  out of them. For this purpose,  $\psi_l$  is defined as a dependency on a local smoothness indicator  $r_l$  which equals the ratio of consecutive gradients [83]

$$r_l = \frac{\Phi_P - \Phi_O}{\Phi_N - \Phi_P}, \quad (2.40)$$

with  $O$  the neighbour to the left of  $P$ . In case of  $r_l \approx 1$  a smooth data distribution can be assumed, whereas  $r_l$  is in- or decreasing near discontinuities. The simple definition (2.40) contributes another reason to the potential degeneration near smooth extrema [47]. Under consideration of the sought TVD property (2.38),  $\psi_l$  has to be defined such that [47, 90]

$$0 \leq \psi_l(r_l) \leq 2, \quad 0 \leq \psi_l(r_l) \leq 2\psi_l(r_l), \quad \forall r_l \geq 0. \quad (2.41)$$

Jasak [41] states that the resulting conditions in (2.41) are just valid for  $Co \leq 1$  and on the basis of explicit time integration. Nevertheless, the solutions remain also stable for higher  $Co$  numbers in practice. Sweby [83] derived further properties of  $\psi_l$  in order to prevent excessive compression within smooth regions, as the superbee limiter shows [47], or too much limiting resulting in smeared solutions, as the minmod limiter offers [41]. He recommends a linear combination of the limiters of the Lax-Wendroff scheme ( $\psi_l = 1$ ) and Beam-Warming scheme ( $\psi_l = r_l$ ) besides the reasonable condition for all second-order accurate schemes  $\psi_l(1) = 1$ . The resulting diagram (figure 2.2a) illustrates proper regions of the limiter in grey. An overview of the common definitions can be found in the book of Versteeg and Malalasekera [93]. In this thesis, the two applied limiters are expressed by [63]

$$\begin{aligned} \text{vanLeer : } \psi_l(r_l) &= \frac{r_l + |r_l|}{1 + |r_l|}, \\ \text{limitedLinear : } \psi_l(r_l) &= \max \left[ 0, \min \left( \frac{2}{k + \varepsilon} r_l, 1 \right) \right], \quad 0 \leq k \leq 1, \end{aligned} \quad (2.42)$$

with  $k$  an user defined parameter and  $\varepsilon$  a small number preventing division by zero. As to be seen in figure 2.2b, limitedLinear mimics for  $k = 1$  and  $r_l \leq 1$  the rather compressive superbee limiter and for  $r_l > 1$  the diffusive minmod limiter. It might be noticed, that for  $k < 1$  the limiter does not satisfy the TVD properties at small indicator values. In contrast, van Leer's limiter is always TVD and, according to Leveque [47], full second-order accurate because of its smoothness near  $r_l = 1$ . Further, it fulfils the symmetry condition

$$\psi_l \left( \frac{1}{r_l} \right) = \frac{\psi_l(r_l)}{r_l}, \quad (2.43)$$

which ensures that backward and forward facing gradients are treated in the same fashion [83] as well as a piecewise linear reconstruction of a symmetric field distribution remains symmetrically [47]. The verification of (2.43) for the limitedLinear limiter reveals its asymmetry.

Finally, the calculation of the smoothness indicator (2.40) has to be modified in case of arbitrary meshes where the left neighbour  $\Phi_O$  is generally unknown. The principal derivation was provided by Darwish and Moukalled [14] and starts with the extension of (2.40)

$$r_l = \frac{\Phi_N + (\Phi_P - \Phi_O) - \Phi_N}{\Phi_N - \Phi_P} = \frac{(\Phi_N - \Phi_O) - (\Phi_N - \Phi_P)}{\Phi_N - \Phi_P}. \quad (2.44)$$

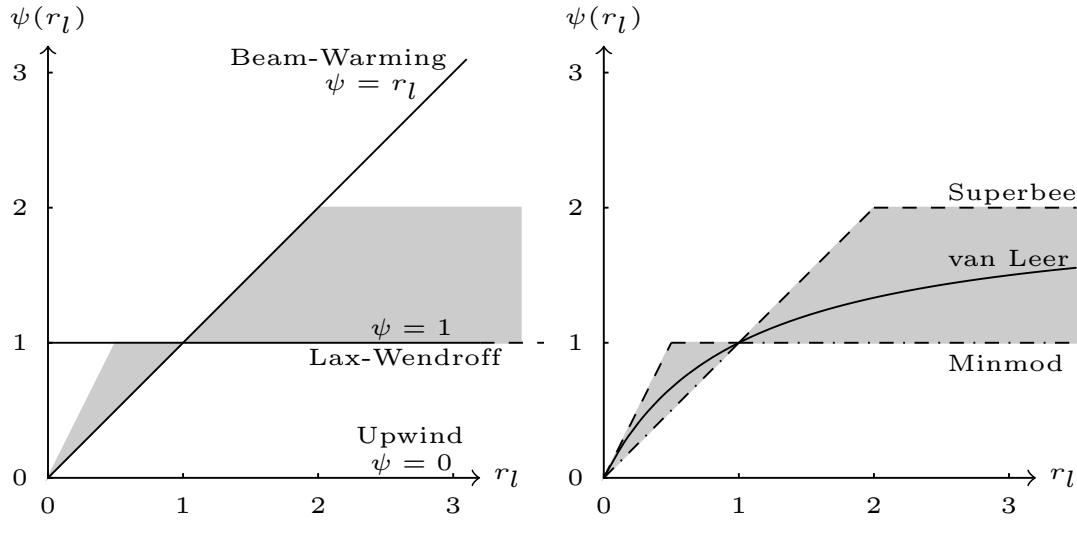
Next, the term involving  $\Phi_O$  is reformulated by defining the position of the virtual  $\vec{x}_O$  at a point on the line through  $\vec{x}_N$  and  $\vec{x}_P$ . At this point the distance to  $\vec{x}_P$  equals the distance between owner and neighbour  $\vec{d}$ . This definition preserves second-order accuracy and reads

$$(\Phi_N - \Phi_O) = \nabla \Phi_P \cdot \vec{d}_{ON} = 2 \nabla \Phi_P \cdot \vec{d}. \quad (2.45)$$

Inserting (2.45) in (2.44), a proper expression for the smoothness indicator becomes

$$r_l = \frac{2 \nabla \Phi_C \cdot \vec{d}}{\Phi_N - \Phi_C} - 1, \quad (2.46)$$

which can be determined without the knowledge of  $\vec{x}_O$ .



(a) Valid regions of TVD schemes

(b) Overview of limiter functions

Figure 2.2: Illustration of the TVD regions of Sweby

### 2.3.2 Diffusion Term

In analogy to convection terms, the discretisation of diffusion terms are based on Gauss's theorem leading to

$$\begin{aligned} \int_{V_i} \nabla \cdot (\rho \Gamma_\Phi \nabla \Phi) dV &= \int_{\partial V_i} \vec{n} \rho \Gamma_\Phi \nabla \Phi d\partial V = \sum_{l=1}^{L_i} \int_{F_l} \vec{n}_l \rho \Gamma_\Phi \nabla \Phi dF_l \\ &= \sum_{l=1}^{L_i} \vec{n}_l |F_l| (\rho \Gamma_\Phi)_l (\nabla \Phi)_l. \end{aligned} \quad (2.47)$$

All described convection schemes can be applied to the interpolation of  $\rho$  and  $\Gamma_\Phi$ . The remaining factors in (2.47) are discretised under consideration of possible non-orthogonality of the mesh which accounts for the non-parallelism of the vector between the cell centres of owner and neighbour  $\vec{d}$  and the normal vector of the face. In order to preserve second-order accuracy, OpenFOAM offers an explicit correction to the usual way of approximating  $(\nabla \Phi)_l$  using finite differences (see Jasak's thesis [41] for a more theoretical treatise) [63]

$$\vec{n}_l (\nabla \Phi)_l = \frac{(\Phi_N - \Phi_P)}{\Delta_l} + (\nabla \Phi)_l \cdot \left( \vec{n}_l - \frac{\vec{d}}{\Delta_l} \right), \quad (2.48)$$

where  $\Delta_l$  is defined as

$$\Delta_l = \max \left( \vec{n}_l \cdot \vec{d}, 0.05 \cdot |\vec{d}| \right). \quad (2.49)$$

The modified calculation of the distance in (2.49) should improve results on meshes of bad quality [63]. Equation (2.48) reduces to the standard gradient calculation in case of an orthogonal mesh expressed by

$$\vec{n}_l (\nabla \Phi)_l = \frac{(\Phi_N - \Phi_P)}{|\vec{d}|}, \quad (2.50)$$

which is handled implicitly. Additionally, the computation of  $(\nabla \Phi)_l$  has to be provided for the correction part. For this purpose, the gradients are calculated in the cell centres

using Gauss's theorem [56]

$$\begin{aligned} (\nabla \Phi)_i &= \frac{1}{|V_i|} \int_{V_i} \nabla \Phi \, dV = \frac{1}{|V_i|} \sum_{l=1}^{L_i} \int_{F_l} \vec{n}_l \Phi \, dF_l \\ &= \frac{1}{|V_i|} \sum_l \vec{n}_l |F_l| \Phi_l, \end{aligned} \quad (2.51)$$

and the face gradients are evaluated as interpolated values from these. It should be noticed, that the discretisation (2.48) is bounded just for orthogonal meshes. Additional limiters help keeping this property up for general meshes which is, however, to the account of accuracy [41].

### 2.3.3 Source Term

The source term in (2.23) has to be rewritten in an appropriate way before discretising it. In the second-order accurate framework of OpenFOAM, a linear representation is preferable. Patankar [68] offers several examples of linearising source terms properly in his book. The result has to be expressible in the following form

$$S_{\Phi,i}(\Phi) = S_u + S_i \Phi_i. \quad (2.52)$$

In terms of FVM, the linearised source term (2.52) reads under consideration of (2.24)

$$\int_{V_i} S_{\Phi}(\Phi) \, dV = S_u |V_i| + S_i \Phi_i |V_i|. \quad (2.53)$$

As suggested by Jasak [41] and Patankar [68], the matrix of the final system of equations should be diagonal dominant in order to guarantee convergence of the iterative solvers. This property is fulfilled, if the magnitude of the diagonal coefficient at least equals the sum of magnitudes of off-diagonal coefficients for each row and additionally, exceeds this sum for at least one row. Therefore, a non-positive linear coefficient  $S_i$  in (2.53) is assigned to the matrix since it leads to an increase of the diagonal coefficient and thereby, improves the diagonal dominance. In contrast, a positive  $S_i$  is written on the right hand side of the system and treated explicitly which has no effect on the matrix.

### 2.3.4 Temporal Discretisation

Besides the spatial discretisation, the transport equation (2.23) has to be integrated over a defined time step  $\Delta t$  for transient problems. For the sake of simplicity, all terms except of the transient term are summarised as a function  $R_i(\Phi)$  resulting in

$$\begin{aligned} \int_t^{t+\Delta t} \left( \int_{V_i} \frac{\partial \rho \Phi}{\partial t} dV \right) dt &= \int_t^{t+\Delta t} R_i dt \\ \int_t^{t+\Delta t} \left( \rho_i |V_i| \frac{\partial \Phi}{\partial t} \right) dt &= \int_t^{t+\Delta t} R_i dt, \end{aligned} \quad (2.54)$$

with the assumption of time-invariant cell volumes and density.

The application of an explicit discretisation leads to the calculation of the solution of time step  $n+1$  from values of the previous time step  $n$ . A first-order accurate formulation for the cell  $V_i$  results in the Euler explicit formula [41]

$$\frac{\Phi_i^{(n+1)} - \Phi_i^{(n)}}{\Delta t} = \frac{1}{\rho_i |V_i|} R_i(\Phi^{(n)}). \quad (2.55)$$

The order of accuracy can be increased by evaluating the function on several sub-steps. The third-order TVD-Runge-Kutta method is a frequent representative and defined by [47]

$$\begin{aligned} \Phi_i^{(n,1)} &= \Phi_i^{(n)} + \frac{\Delta t}{\rho_i |V_i|} \cdot R_i(\Phi^{(n)}) \\ \Phi_i^{(n,2)} &= \frac{3}{4} \Phi_i^{(n)} + \frac{1}{4} \Phi_i^{(n,1)} + \frac{\Delta t}{4\rho_i |V_i|} \cdot R_i(\Phi^{(n,1)}) \\ \Phi_i^{(n+1)} &= \frac{1}{3} \Phi_i^{(n)} + \frac{2}{3} \Phi_i^{(n,2)} + \frac{2\Delta t}{3\rho_i |V_i|} \cdot R_i(\Phi^{(n,2)}). \end{aligned} \quad (2.56)$$

The size of the time step  $\Delta t$  in (2.55) and (2.56) has to be chosen without violating the stability criterion of the  $Co$  number, also known as CFL-condition [12]

$$\text{CFL} = \frac{\vec{u}_f \cdot \vec{d}}{\Delta t} \leq 1. \quad (2.57)$$

The restriction leads to disproportional time steps in steady-state simulations.

In such cases, implicit schemes benefit from their unconditional stability resulting in arbitrary  $\Delta t$  in theory. In return, a system of equations has to be solved in each time step in an iterative, time consuming manner. In analogy to the explicit Euler method, the implicit Euler method is identical to the first-order formulation [41]

$$\frac{\Phi_i^{(n+1)} - \Phi_i^{(n)}}{\Delta t} = \frac{1}{\rho_i |V_i|} R_i(\Phi^{(n+1)}). \quad (2.58)$$

Versteeg and Malalasekera [93] recommend the implicit Euler method also for transient flows because of its stability and boundedness irrespective of the chosen time step. However, high accuracy is just obtained for small  $\Delta t$ . The order of accuracy can be increased by taking several previous solutions into account. The Crank-Nicolson scheme uses one old and the current time step values for calculating the new solution as a central difference. According to Ferziger and Perić [24], this leads to second-order accuracy in time, unconditional stability, but boundedness just for small time steps. Spurious oscillations may occur from negative coefficients of the old time step resulting in less diagonal dominance of the systems matrix. OpenFOAM, therefore, offers a blending with the Euler implicit method for the purpose of stability. As an alternative, Ferziger and Perić [24] recommend the quadratic backward scheme which needs an additional previous time step. The resulting formula is given by ( see [41] for a derivation)

$$\Phi_i^{(n+1)} = \frac{4}{3}\Phi_i^{(n)} - \frac{1}{3}\Phi_i^{(n-1)} + \frac{2\Delta t}{3\rho_i |V_i|} R_i(\Phi^{(n+1)}). \quad (2.59)$$

The scheme tends to less oscillation at larger time steps for which reason it is chosen in the later simulations. However, it is less accurate than the Crank-Nicolson scheme at small  $\Delta t$ .

# **Chapter 3**

## **Implementation of High-Order Non-Oscillatory Schemes**

In this chapter, the development of the weighted essentially non-oscillatory (WENO) reconstruction is presented first. It can be applied to the evaluation of any field in finite volumes using its known cell-averaged values. The algorithm handles convex cells in arbitrary three-dimensional meshes. The efficiency of the implementation is increased by parallelising the code using Message Passing Interface (MPI). The resulting reconstruction method is the basis for the development of high-order interpolation and gradient schemes in chapter 3.3 and 3.4. Further, chapter 3.5 provides the derivation of a high-order temporal discretisation scheme using the ADER approach.

As described in chapter 2.3, the discretisation of convective terms is a challenging task in flows with discontinuities as shocks or free surfaces. Large gradients should be captured as sharp as possible, but at the same time, non-physical solutions have to be prevented. A typical example in multiphase flows are negative density values which can abort the whole simulation. Linear schemes with such monotone behaviour are restricted to first-order accuracy according to Godunov [27]. Therefore, non-linear schemes were developed, such as TVD schemes of chapter 2.3.1 which are based on

non-linear limiting. Their main drawbacks are the degeneration to first-order accuracy near extrema regardless of a smooth peak or discontinuity, and at the most second-order accuracy. This led to the development of (weighted) essentially non-oscillatory schemes. They introduce non-linear weighting for preventing oscillations near discontinuities and simultaneously reach arbitrary order of accuracy in smooth regions. However, the solution is not strictly bounded leading to the necessity of additional limiting as presented in chapter 3.3.1 [107].

The first ENO scheme based on adaptive stencils was developed by Harten et al. [32]. Their scheme searches for the most suitable cells in a dynamic manner in order to obtain a stencil with the smoothest solution in each time step. The application on unstructured grids was provided by the scheme of Abgrall [1] for the first time. The drawbacks of ENO methods are the prevention of convergence in the case of frequent switching of the stencil from one time step to another and costly operations in runtime reducing the overall performance [60]. An alternative scheme, named Quasi-ENO, was introduced by Ollivier-Gooch [60] to overcome these drawbacks. In contrast to classical ENO schemes, it is based on a high-order least-squares reconstruction which fulfils the ENO-property by a pointwise data-dependent weighting in one fixed, central stencil [59]. Classical least-squares reconstructions are able to reach high-orders in smooth regions but fail near discontinuities. A common way of eliminating this problem is introducing a limiter as Wang [98] and McDonald [52] demonstrated. However, the underlying mathematical problem is the inclusion of the data from cells behind a discontinuity into the reconstruction. The Quasi-ENO scheme solves this problem in a more efficient way as the classical ENO scheme and furthermore, offers the possibility of excluding cells of poor quality from the reconstruction. In the author's preceded work [50], it could be shown that Ollivier-Gooch's approach seems to provide accurate solutions as long as the topology allows compact stencils without scaling effects. But, the solutions of the systems of equations often have to be calculated from bad conditioned

matrices, particularly in case of higher order polynomials. The common way of improving the condition numbers by applying preconditioning fails for the ENO weighted reconstruction matrices. Further, the stencils may loose too much information for reaching the nominal order of accuracy near discontinuities which results in a reduction of the order similar to TVD schemes.

Therefore, the implementation of the previous work was extended to a WENO scheme which operates in a reference space without scaling effects in order to prevent bad conditioned matrices and improve the accuracy on irregular grids. The first WENO scheme was introduced by Liu, Osher, and Chan [48] and Jiang and Shu [42]. Later, Friedrich [25] extended it to two-dimensional, unstructured grids. In the recent past, Dumbser and Käser [18] and Tsoutsanis, Antoniadis, and Drikakis [92] accomplished that finite volume WENO schemes are applicable on three-dimensional, unstructured grids with up to arbitrary order of accuracy. Pringuey [70] successfully combined this approach with a level set method in his Ph.D. thesis. In contrast to ENO, WENO schemes compute the solution on several fixed stencils and combine them by a non-linear weighting in order to obtain a final, smooth solution. The weights are calculated from an evaluation of the smoothness in each stencil. This theoretically leads up to  $(2r + 1)^{th}$  order of accuracy by using a polynomial of  $r^{th}$ -order [59]. In general, the step from ENO to WENO schemes seems to be indispensable because they provide a higher efficiency in runtime by computating pseudoinverses during the preprocessing step. Further, they yield better convergence due to smoother numerical fluxes [79]. In return, they need the preparation of several stencils which is computationally quite expensive in preprocessing especially on unstructured meshes in three dimensions [18]. Besides, the reconstruction may fail near critical points, on very coarse grids, or if not enough smooth data is provided [102].

### 3.1 WENO Reconstruction Method

The spatial domain is taken in its discretised form of  $N$  cells with the volumes  $|V_i|$ ,  $i = 1, \dots, N$ , and the data of any scalar variable  $\Phi$  is stored in the cell centres  $\vec{x}_i$  represented by the cell averaged values  $\bar{\Phi}_i$ . In case of a vector or tensor field, each component is evaluated separately because of the later introduced weighting. With the aid of a least-squares reconstruction,  $\Phi$  is replaced by a polynomial representation  $p_i$  in each cell  $V_i$  with the constraint of conservation of the mean value within  $V_i$ . Scaling effects are prevented by mapping the reconstruction process from the physical space  $\vec{x} = \vec{x}(x, y, z)$  into a reference space  $\vec{\xi} = \vec{\xi}(\xi, \eta, \zeta)$  using an affine transformation  $\vec{x} = \vec{x}(\xi, \eta, \zeta)$  and its inversion  $\vec{\xi} = \vec{\xi}(x, y, z)$  respectively. The affinity of the transformation conserves the conservation condition which results in (see [50] for the prove)

$$\bar{\Phi}_i = \frac{1}{|V_i|} \int_{V_i} \Phi(\vec{x}) dx dy dz = \frac{1}{|V'_i|} \int_{V'_i} \Phi(\vec{\xi}) d\xi d\eta d\zeta = \frac{1}{|V'_i|} \int_{V'_i} p_i(\vec{\xi}) d\xi d\eta d\zeta, \quad (3.1)$$

with  $V'_i$  the mapped cell  $V_i$  in its reference space. The polynomials  $p_i$  are expressed by an expansion over local polynomial basis functions  $\Omega_k$  [92]

$$p_i(\vec{\xi}) = \bar{\Phi}_i + \sum_{k=1}^K a_k \Omega_k(\vec{\xi}). \quad (3.2)$$

The number of degrees of freedom  $K$  relates to the order of the polynomial  $r$  in three dimensions according to

$$K = \frac{(r+1)(r+2)(r+3)}{6} - 1, \quad (3.3)$$

respectively in two dimensions to

$$K = \frac{(r+1)(r+2)}{2} - 1. \quad (3.4)$$

The degrees of freedom  $a_k$  are evaluated on a set of stencils  $U_{Si}$  for each cell given by

$$U_{Si} = \bigcup_{m=0}^{N_{Si}} S_m, \quad (3.5)$$

in comparison to the single, central stencil of classical least-squares reconstructions. The number of sector stencils  $N_{Si}$  depends on the cell shape and its closeness to boundaries. By definition, the stencil  $S_0$  is the central stencil. For each  $S_m$ , appropriate neighbouring cells  $V'_j$  of the reference space of  $V'_i$  are collected which yields

$$S_m = \bigcup_{j=0}^{J_{max}} V'_j. \quad (3.6)$$

The choice of  $J_{max}$  is an important parameter which is discussed in chapter 3.1.1. By definition,  $j = 0$  corresponds to the target cell  $V'_i$ . A polynomial  $p_m(\vec{\xi})$  is formulated for each  $S_m$  using (3.2) and afterwards, the WENO reconstructed polynomial is obtained by a non-linear combination of several  $p_m$  according to

$$p_{weno}(\vec{\xi}) = \sum_{m=0}^{N_{Si}} \omega_m p_m(\vec{\xi}), \quad (3.7)$$

with the non-linear weights

$$\omega_m = \frac{\gamma_m}{\sum_{m=0}^{N_{Si}} \gamma_m}, \quad (3.8)$$

and  $\gamma_m$  defined as

$$\gamma_m = \frac{d_m}{(\varepsilon + \mathcal{I}_{S,m})^p}. \quad (3.9)$$

$\varepsilon$  prevents the denominator from becoming zero and is taken as  $10^{-6}$ . The index  $p$  enables the weights of non-smooth regions the possibility of tending to zero faster than the mesh size converges to zero and is usually chosen as  $p = 4$  [34]. The linear weights  $d_m$  assign a higher weight to the central stencil in comparison to the sectoral stencils due to its precision in reconstructing smooth functions. In accordance with Dumbser and Käser [18], they are defined as

$$d_m = \begin{cases} 10^3 & \text{if } m = 0 \\ 1 & \text{else} \end{cases} . \quad (3.10)$$

A lower weight  $d_0$  might be advantageous for problems including many discontinuities. Finally,  $\mathcal{I}_{S,m}$  represents the indicator of the smoothness of the solution in the stencil  $S_m$ . This indicator diminishes in case of smooth solutions and thus, the corresponding weight  $\omega_m$  increases. The matrix expression for  $\mathcal{I}_{S,m}$  is given by Pringuey [70] and reads

$$\mathcal{I}_{S,m} = \sum_{s=1}^K a_s^{(m)} \cdot \left( \sum_{t=1}^K \mathcal{B}_{st} a_t^{(m)} \right), \quad (3.11)$$

where  $\mathcal{B}_{st}$  is an element of the mesh-independent oscillation indicator matrix  $\mathcal{B}$ , which is discussed in chapter 3.1.4. It is convenient to represent the WENO polynomial  $p_{weno}$  in the form of (3.2) by substituting (3.2) in (3.7)

$$p_{weno}(\vec{\xi}) = \sum_{m=0}^{N_{Si}} \omega_m \left( \bar{\Phi}_i + \sum_{k=1}^K a_k^{(m)} \Omega_k(\vec{\xi}) \right). \quad (3.12)$$

In consideration of the partition of unity through the weights and a rearrangement, equation (3.12) finally results in

$$p_{weno}(\vec{\xi}) = \bar{\Phi}_i + \sum_{k=1}^K \left( \sum_{m=0}^{N_{Si}} \omega_m a_k^{(m)} \right) \cdot \Omega_k(\vec{\xi}) = \bar{\Phi}_i + \sum_{k=1}^K \tilde{a}_k \cdot \Omega_k(\vec{\xi}), \quad (3.13)$$

with  $\tilde{a}_k$  denominated as modified degrees of freedom.

The basis functions  $\Omega_k$  have to be chosen with the constraint of satisfying (3.1), equivalent to a zero mean value over  $V'_i$ . The condition can be satisfied analytically by an appropriate definition of the basis functions

$$\Omega_k(\vec{\xi}) = \Psi_k(\vec{\xi}) - \frac{1}{|V'_i|} \int_{V'_i} \psi_k(\vec{\xi}) d\xi d\eta d\zeta, \quad k = 1, \dots, K, \quad (3.14)$$

with arbitrary orthogonal polynomial basis functions  $\Psi_k$ . In accordance with Ollivier-Gooch [60] and Friedrich [25], a Taylor series expansion around the centre of  $V_i$  is applied and defined as

$$\Psi_k(\vec{\xi}) = (\xi - \xi_i)^n (\eta - \eta_i)^m (\zeta - \zeta_i)^o, \quad (3.15)$$

where  $k$  corresponds to one combination of  $n, m, o$  such that  $0 \leq n + m + o \leq r$ . The final system of equations is computed with the aim of preserving the averaged values  $\bar{\Phi}_j$  in all cells  $V'_j$  of the stencil  $S_m$  by the corresponding cell averages of  $p_m$ . It can be expressed by

$$\bar{\Phi}_j = \frac{1}{|V'_j|} \int_{V'_j} p_m(\vec{\xi}) d\xi d\eta d\zeta, \quad j = 1, \dots, J_{max}. \quad (3.16)$$

Substituting (3.2) in (3.16) yields the final system of equations

$$\bar{\Phi}_j - \bar{\Phi}_i = \sum_{k=1}^K a_k^{(m)} \cdot \left( \frac{1}{|V'_j|} \int_{V'_j} \Omega_k(\vec{\xi}) d\xi d\eta d\zeta \right), \quad j = 1, \dots, J_{max}, \quad (3.17)$$

$$b_j^{(m)} = \sum_{k=1}^K \mathcal{A}_{jk} a_k^{(m)}, \quad j = 1, \dots, J_{max}, \quad (3.18)$$

where  $\mathcal{A}_{jk}$  can be calculated under consideration of (3.2), (3.14) and (3.17) as

$$\mathcal{A}_{jk} = \frac{1}{|V'_j|} \int_{V'_j} \Psi_k d\xi d\eta d\zeta - \frac{1}{|V'_i|} \int_{V'_i} \Psi_k d\xi d\eta d\zeta. \quad (3.19)$$

The volume integrals would have to be evaluated for each combination  $V'_j$  since  $\Psi_k$  depends on the centre of  $V'_i$ . In order to avoid this, the Taylor series is transformed by replacing e.g.  $(\xi - \xi_i)$  by  $(\xi - \xi_j) + (\xi_j - \xi_i)$  with  $\vec{\xi}_j$  the cell centre of  $V'_j$ . Inserting these expressions in (3.19), the final computation of  $\mathcal{A}_{jk}$  gets

$$\begin{aligned} \mathcal{A}_{jk} &= \sum_{p=0}^l \sum_{q=0}^m \sum_{g=0}^n \binom{l}{p} \binom{m}{q} \binom{n}{g} (\xi_j - \xi_i)^p (\eta_j - \eta_i)^q (\zeta_j - \zeta_i)^g \cdot \\ &\quad \frac{1}{|V'_j|} \int_{V'_j} (\xi - \xi_j)^{(l-p)} (\eta - \eta_j)^{(m-q)} (\zeta - \zeta_j)^{(n-g)} d\xi d\eta d\zeta \\ &\quad - \frac{1}{|V'_i|} \int_{V'_i} (\xi - \xi_i)^l (\eta - \eta_i)^m (\zeta - \zeta_i)^n d\xi d\eta d\zeta, \end{aligned} \quad (3.20)$$

A more detailed derivation is provided by Ollivier-Gooch [60]. The volume integrations in (3.20) have to be computed for each cell and its appropriate stencil separately due to the dependency of  $V'_j$  on the coordinate system of  $V'_i$ . For this purpose, the volume

integrals are transformed into surface integrals by using the divergence theorem [55]. Denoting  $\vec{n}'$  as the outward normal vector in the reference space, it yields

$$\int_{V'_i} \nabla \cdot \Psi_k \, dV'_i = \int_{\partial V'_i} \Psi_k \vec{n}' \cdot \vec{e}_n \, d\partial V'_i. \quad (3.21)$$

The right-hand side can be further written as a sum of surface integrals over the  $L_i$  faces  $F'_l$  of  $V'_i$ . Here, the normal vector of each face  $\vec{n}'_l$  is constant and can be taken out of the integrals. The desired volume integrals of the monomials are obtained from (3.21) by integrating  $\Psi_k(\vec{\xi})$  analytically over one of the coordinates. Under consideration of the definition of  $\Psi_k$  in (3.15), the evaluation can be written as

$$\begin{aligned} \int_{V'_i} (\xi - \xi_j)^n (\eta - \eta_j)^m (\zeta - \zeta_j)^o \, dV'_i = \\ \sum_{l=1}^{L_i} \frac{n'_{l,\xi}}{n+1} \int_{F'_l} (\xi - \xi_j)^{n+1} (\eta - \eta_j)^m (\zeta - \zeta_j)^o \, dF'_l. \end{aligned} \quad (3.22)$$

The surface integrals are computed by decomposing the faces into triangles and transforming each of them to a standard triangle using linear mapping [15]. OpenFOAM's `polyMeshTetDecomposition`, which splits any plane polygon into a minimum number of triangles, is applied for this decomposition. The experiences show accurate results for arbitrary cell shapes as long as convexity and planarity is ensured. They are evaluated using a Gaussian quadrature rule for the standard triangle with the appropriate coordinates of the Gaussian points prepared by Deng [15]. It might be noticed, that the computation of the volume integral of a fourth-order polynomial needs surface integrations over fifth-order monomials which results in seven Gaussian points per triangle.

### 3.1.1 Stencil Collection Algorithm

The modified degrees of freedom are calculated from weighted solutions in several stencils. As mentioned above, these solutions are evaluated in one central and several sectoral stencils which cover all spatial directions of the target cell. In the case of cells near

boundaries, some stencils could be too small and have to be discarded. In contrast to classical ENO schemes, the WENO scheme computes the solutions on time-invariant stencils. Thus, the time consuming collection part simplifies and has to be executed just once during the preprocessing step. The most important requirement for stencils to obtain an accurate solution is compactness. On isotropic, uniform meshes it is simple preserved by adding the nearest neighbours iteratively. However, on unstructured meshes and in regions with highly anisotropic cells this procedure can not ensure compactness. The selection of the stencils in physical space may lead to a loss of information near walls along the boundary layer region, in particular. Therefore, the stencil is transformed to the reference system  $\vec{\xi} = (\xi, \eta, \zeta)$  where no scaling effects from increasing grid resolution or deformed cells occur. In the following, the used mapping of Dumbser and Käser [18] is explained in detail.

Denoting any vertex of  $V_i$  as  $P_0 = (x_0, y_0, z_0)$ , a basis of the reference frame can be spanned by this point and three other vertices of the cell  $P_1 = (x_1, y_1, z_1), \dots, P_3 = (x_3, y_3, z_3)$  which are connected to  $P_0$  through an edge (see figure 3.1). This leads to the following mapping  $\vec{x} = \vec{x}(\xi, \eta, \zeta)$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \mathcal{J} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix}, \quad (3.23)$$

with  $\mathcal{J}$  the Jacobian matrix defined as

$$\mathcal{J} = \begin{pmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{pmatrix}. \quad (3.24)$$

Any point in the stencil  $P_{\vec{x}} = (x_P, y_P, z_P)$  can then be mapped to its equivalent in the new

space  $P_{\vec{\xi}} = (\xi_P, \eta_P, \zeta_P)$  by applying

$$\begin{pmatrix} \xi_P \\ \eta_P \\ \zeta_P \end{pmatrix} = \mathcal{J}^{-1} \begin{pmatrix} x_P - x_0 \\ y_P - y_0 \\ z_P - z_0 \end{pmatrix}. \quad (3.25)$$

The transformed cell volume  $|V'_i|$  results from

$$|V'_i| = |\det(\mathcal{J}^{-1})| |V_i|. \quad (3.26)$$

The inverse Jacobians are precomputed for each target cell analytically which reduces the costs of mapping.

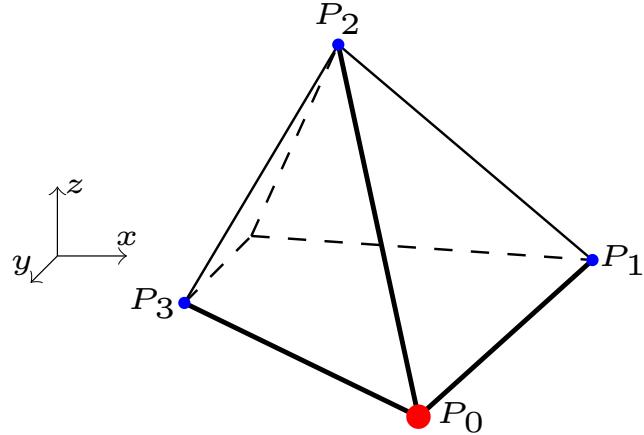


Figure 3.1: Selection of the vertices for the definition of the reference system  $\vec{\xi}$

On arbitrary mapped meshes, the most compact stencil could be collected by using point-neighbour information. On the contrary, the use of face-neighbours extends the dependent data further into the mesh which reduces the redundancy of data on anisotropic, structured meshes [70]. The affine transformation preserves the principal connections between the cells for which reason the gathering can be performed using existing owner-neighbour lists. The most efficient way of collecting cells on unstructured meshes is based on adding the neighbours of the target cell iteratively until each

stencil  $S_m$  has the sought size of  $J_{max}$  entries. By collecting a surplus of possible stencil cells, the algorithm is independent of the starting point of an iteration and provides complete layers of new neighbours at a time as it is shown in figure 3.2. It may happen that more layers have to be considered until the necessary size is reached near boundaries. At this point, the iterative implementation is straightforward and advantageous. Once the lists are completed, all candidates are sorted by centre to centre distances to the target cell in  $\vec{\xi}$  and the nearest  $J_{max}$  cells are stored.

The system of equations (3.18) provides a solution if the matrix  $\mathcal{A}$  is at least squared resulting in the condition  $J_{max} \geq K$ . As Tsoutsanis, Antoniadis, and Drikakis [92] state, choosing  $J_{max} = K$  leads to unstable solutions or eventually ill-conditioned systems. Therefore,  $J_{max}$  should be approximately  $2 K$  for three-dimensional problems and  $1.5 K$  in 2D.

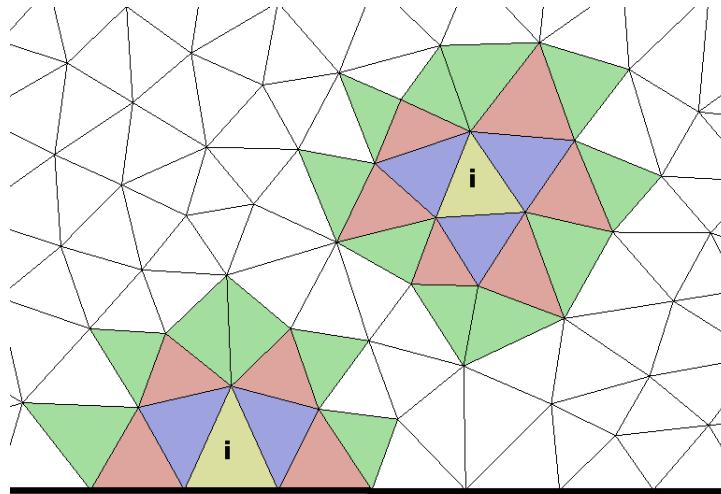


Figure 3.2: Layers of cells added at each iteration: in purple, initial neighbours; in red, the cells added at the first iteration; in green, the cells added at the second iteration - two-dimensional example

### 3.1.2 Generation of Sectoral Stencils

The sectoral stencils are constructed under consideration of several aspects. All stencils have to be compact in a topological sense which is automatically fulfilled by the above procedure. Further, the union of all sectors has to cover the whole space around the target cell like one big central stencil on condition that they are not allowed to share another cell than the target cell [70]. A simple but nonetheless brilliant way of taking this term into account is provided by Pringuey [70]. Following his development, the selection of sectors starts from one large central stencil list where the cells are already sorted by distance as described in chapter 3.1.1. The size of the list  $N_U$  relates to the number of internal faces of the target cell  $L_i$  according to

$$N_U = (L_i + 1) \cdot J_{max}, \quad (3.27)$$

which offers a small surplus due to the described collection of layers and in case of convoluted boundaries [70]. After the selection of sectoral stencils, the central stencil is simply obtained by cutting it to the necessary size  $J_{max}$ . In general, the number of sectors equals the number of internal faces of  $V'_i$ . However, if some sector does not provide enough cells it is not taken into account for the runtime operations.

Next, each sector is spanned as a cone with the cell centre of the target cell  $P_0 = (x_0, y_0, z_0)$  as the apex and the contour of the related face as the base. The  $N_U$  cells are assigned to the sectors according to the position of their cell centres which results in a distribution as can be seen in figure 3.3. In order to check the assignment of the cells, each sector is mapped to the first octant of another transformation space  $\vec{X} = (X, Y, Z)$  where the relevant cell centres have positive coordinates. Under consideration of the cell centre  $P_0 = (x_0, y_0, z_0)$ , such a mapping is given by (see also figure 3.4)

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \mathcal{J}_C \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \quad (3.28)$$

with  $\mathcal{J}_C$  the Jacobian matrix built from the base of the cone  $P_1, P_2, P_3$  (compare (3.24)). Equation (3.28) is only applicable to triangles for which reason each internal face is divided into a minimum number of triangles  $N_T$  first. Thus, the sector of a face  $S_l$  is a union of several sub-sectors  $S_{lj}$

$$S_l = \bigcup_{j=1}^{N_T} S_{lj}. \quad (3.29)$$

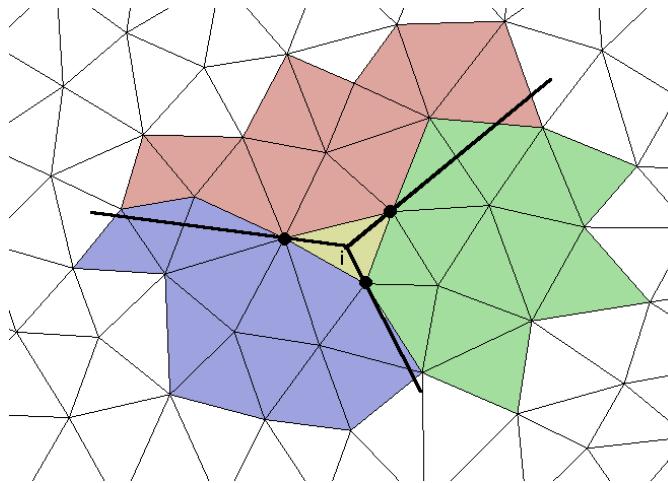


Figure 3.3: Definition of the three sectoral stencils for a triangular cell - two-dimensional example

As a final step, the cell centre of each of the  $N_U$  cells is transformed to the reference space  $\vec{X}$  of each  $S_{lj}$  and added to the sectoral stencil  $S_l$  if:

- All coordinates in the reference space are positive.
- The cell is not already a member of another stencil which may happen on Cartesian grids where the centre lies on the boundary of two adjunct sectors.
- The target stencil list contains no more than  $J_{max}$  cells.

The first and last condition are simple requests while the second condition is prevented by using a dynamic list from which cells are removed after being assigned to a sector. The obtained stencils are compact by itself since the central lists are presorted and scanned from the nearest to the farthest cells.

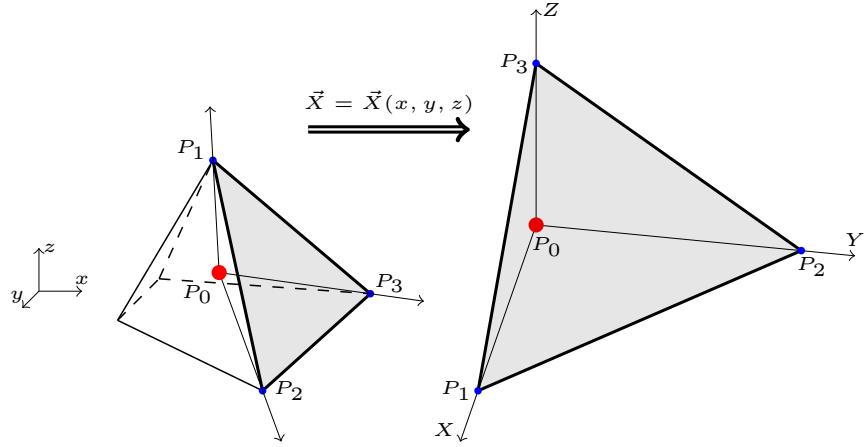


Figure 3.4: Mapping of a triangular face to the first octant

### 3.1.3 Solving the System of Equations

In this chapter, the procedure for solving the resulting system of equations (3.18) is described. As stated in chapter 3.1.1, the robustness of the system is improved by considering more equations than unknowns resulting in an overdetermined, linear least-squares problem. Physically, it corresponds to the minimization of the  $\mathcal{L}_2$ -norm of the error in predicting the averaged values of the polynomial in all cells of the stencil [59]

$$\min \|\mathcal{A} \cdot a - b\|_2, \quad \mathcal{A} \in \mathbb{R}^{m \times n}, \quad b \in \mathbb{R}^m, \quad m \geq n, \quad (3.30)$$

with  $a$  the solution vector containing the degrees of freedom. The easiest way of solving (3.30) is using normal equations which is written as

$$\begin{aligned} \mathcal{A}^T \mathcal{A} a &= \mathcal{A}^T b \\ a &= (\mathcal{A}^T \mathcal{A})^{-1} \mathcal{A}^T b = \mathcal{A}^+ b, \end{aligned} \quad (3.31)$$

with  $\mathcal{A}^+$  the Moore-Penrose pseudoinverse. According to Golub and Loan [28], the accuracy of the solution from (3.31) depends on the square of the condition number of  $\mathcal{A}$ . In WENO computations, the resulting condition numbers are rather large, particularly for higher order polynomials. Furthermore, rank-deficient matrices may occur caused by nearly linear-dependent lines in  $\mathcal{A}$  which arise if several cells of a stencil lie on a straight line on structured grids [18]. The deficiencies are indicated by very small singular values resulting in large condition numbers as well (see below). One solution is the addition of more cells to the stencil [59]. However, the condition number is linear dependent on  $J_{max}$  for which reason the accuracy of the solution could decrease. Following Lawson and Hanson [44], the better solution is the negligence of the problem causing information from reconstructions which needs alternative methods.

In the implementation of the Quasi-ENO scheme, the author [50] indicated  $QR$ -decomposition with Householder transformations as the preferable method for solving rank-deficient least-squares methods due to greatly improved efficiency in comparison to the common singular value decomposition (SVD). The algorithm mainly benefits from an implicit decomposition which is, however, lost if pseudoinverses have to be provided explicitly. Therefore, SVD is a suitable alternative for calculating  $\mathcal{A}^+$  due to a more stable, straightforward computation of the decomposition. In addition, the tolerance parameter, used for the exclusion of columns from the solution, is no more a loose parameter but a more comprehensible singular value.

In SVD, the  $m \times n$  matrix  $\mathcal{A}$  is represented by its decomposition

$$\mathcal{A} = \mathcal{U}\mathcal{S}\mathcal{V}^T, \quad (3.32)$$

with  $\mathcal{U}$  a  $m \times m$  orthogonal matrix of the orthonormalised eigenvectors of  $\mathcal{A}\mathcal{A}^T$  and  $\mathcal{V}$  a  $n \times n$  orthogonal matrix of the orthonormalised eigenvectors of  $\mathcal{A}^T\mathcal{A}$ . Lastly,  $\mathcal{S}$  represents a  $m \times n$  diagonal matrix containing the singular values  $\sigma$  of  $\mathcal{A}$ . For a detailed description of efficient decompositions see the book of Lawson and Hanson [44].

As indicated above, rank-deficient matrices may occur corresponding to small values

in  $\mathcal{S}$ . In this case, it is preferable to neglect the causing columns from the further solution procedure since the condition number  $\kappa$  is influenced by singular values of  $\mathcal{A}$  according to

$$\kappa(\mathcal{A}) = \frac{\sigma_{\max}(\mathcal{A})}{\sigma_{\min}(\mathcal{A})}. \quad (3.33)$$

A small  $\sigma_{\min}$  increases  $\kappa$  and reduces the accuracy of the solution. The tolerance parameter  $\tau$  is introduced representing the minimum allowed singular value included in the solution procedure. Thereby, the diagonal matrix  $\mathcal{S}$  is replaced by the modified matrix  $\mathcal{S}_\tau$  which reads

$$\mathcal{S}_\tau = \text{diag}(\sigma_{\tau,n}), \quad \sigma_{\tau,n} = \begin{cases} \sigma_n & \text{if } \sigma_n > \tau \\ 0 & \text{otherwise} \end{cases}. \quad (3.34)$$

In this thesis,  $\tau$  is set to  $10^{-5}$  for all calculations. Once decomposed, the least-squares problem can be solved in each time step by inserting the decomposition (3.32) in (3.30) leading to

$$\begin{aligned} (\mathcal{U}\mathcal{S}\mathcal{V}^T) \cdot a &= b \\ a &= (\mathcal{V}\mathcal{S}_\tau^+\mathcal{U}^T) \cdot b = \mathcal{A}^+b. \end{aligned} \quad (3.35)$$

As proven by Lawson and Hanson [44], equation (3.35) calculates the unique solution of minimum length in any case. The pseudoinverses have to be computed for each stencil in preprocessing only once since  $\mathcal{A}$  is solution independent.

### 3.1.4 WENO Smoothness Indicator

After the degrees of freedom are calculated for each stencil, the final coefficients are computed as a non-linear weighted combination using (3.13). The weights have to determine to what extend the solution in a stencil provides a qualitative contribution to a smooth solution of the polynomial reconstruction in the target cell. The key element

is here the calculation of the smoothness indicator  $\mathcal{I}_S$ . The aim of the indicator is the minimization of the total variation of the sum of the  $\mathcal{L}_2$ -norms of all derivatives of the polynomial, which is comparable to the TVD property of other convection schemes [42].

Recalling the definition of  $\mathcal{I}_S$  in (3.11), this property has to be satisfied by the oscillation indicator matrix  $\mathcal{B}$  whose elements  $\mathcal{B}_{st}$  are, therefore, defined as

$$\mathcal{B}_{st} = \sum_{\lambda=1}^r \sum_{\alpha=0}^{\lambda} \sum_{\beta=0}^{\lambda-\alpha} \int_{V'_i} \frac{\partial^\lambda}{\partial \xi^\alpha \partial \eta^\beta \partial \zeta^\gamma} \Omega_s(\vec{\xi}) \cdot \frac{\partial^\lambda}{\partial \xi^\alpha \partial \eta^\beta \partial \zeta^\gamma} \Omega_t(\vec{\xi}) d\xi d\eta d\zeta, \quad (3.36)$$

with  $\gamma = \lambda - \alpha - \beta$  and  $r$  the polynomial order. As can be seen from (3.36), the matrix is solution independent and may be precomputed. Additionally, it is mesh independent through the calculation in the reference space. Under consideration of the definitions for the basis functions (3.14), equation (3.36) can be further simplified resulting in

$$\mathcal{B}_{st} = \sum_{\lambda=1}^r \sum_{\alpha=0}^{\lambda} \sum_{\beta=0}^{\lambda-\alpha} \int_{V'_i} \frac{\partial^\lambda}{\partial \xi^\alpha \partial \eta^\beta \partial \zeta^\gamma} \Psi_s(\vec{\xi}) \cdot \frac{\partial^\lambda}{\partial \xi^\alpha \partial \eta^\beta \partial \zeta^\gamma} \Psi_t(\vec{\xi}) d\xi d\eta d\zeta. \quad (3.37)$$

The monomials are expressed by their orthogonal basis functions as

$$\begin{aligned} \Psi_s(\vec{\xi}) &= (\xi - \xi_i)^{A_1} (\eta - \eta_i)^{B_1} (\zeta - \zeta_i)^{C_1}, \\ \Psi_t(\vec{\xi}) &= (\xi - \xi_i)^{A_2} (\eta - \eta_i)^{B_2} (\zeta - \zeta_i)^{C_2}, \end{aligned} \quad (3.38)$$

with the condition  $(A_1, B_1, C_1, A_2, B_2, C_2) \in [0, r]^6$  and  $\{1 \leq A_i + B_i + C_i \leq r, i = 1, 2\}$ .

Applying the partial derivatives to (3.38), equation (3.37) yields

$$\mathcal{B}_{st} = \sum_{\lambda=1}^r \sum_{\alpha=0}^{\lambda} \sum_{\beta=0}^{\lambda-\alpha} \int_{V'_i} K \cdot \xi^{(A_1+A_2-\alpha)} \eta^{(B_1+B_2-\beta)} \zeta^{(C_1+C_2-\gamma)} d\xi d\eta d\zeta. \quad (3.39)$$

As it is shown in Pringuey's thesis [70],  $K$  is identical to

$$\begin{aligned} K &= \begin{cases} K_1 & \text{if } \forall_{i=1,2} : (A_i \geq \alpha) \wedge (B_i \geq \beta) \wedge (C_i \geq \gamma) \\ 0 & \text{else} \end{cases}, \\ K_1 &= \frac{A_1!}{(A_1 - \alpha)!} \cdot \frac{A_2!}{(A_2 - \alpha)!} \cdot \frac{B_1!}{(B_1 - \beta)!} \cdot \frac{B_2!}{(B_2 - \beta)!} \cdot \frac{C_1!}{(C_1 - \gamma)!} \cdot \frac{C_2!}{(C_2 - \gamma)!}. \end{aligned} \quad (3.40)$$

The evaluation of the volume integrals in (3.39) is carried out by transforming them into surface integrals and integrating over the faces of each cell as described in the beginning of this chapter. Here, the highest order to consider is  $2r - 2$  since  $\lambda_{min} = 1$ . All volume integrals are calculated before the triple sum is evaluated using (3.39) and (3.40) for the sake of efficiency.

### 3.1.5 Parallelisation

In this chapter an outline of the parallelisation of the code is given. It is a crucial step due to the time consuming reconstruction process in runtime. By default, OpenFOAM uses a 0-halo approach which divides the domain into several non-overlapping regions and Message passing interface (MPI) to transmit the information between the inter-processor boundaries. This leads to at best second-order accurate solutions at such boundaries [70]. In contrary, the stencils of a high-order (W)ENO scheme near processor patches need the geometrical and physical data from several layers of the neighbouring domain. Consequently, a  $n$ -halo approach with several overlapping sub-domains would be the proper choice. The implicit handling of the Navier-Stokes equations leads to algebraic systems of equations which are solved by linear, iterative solvers in OpenFOAM. Since these solvers only work for 0-halo approaches, a  $n$ -halo approach is discarded. Instead, the solution is the virtual extension of the sub-domains by collecting halo cells from neighbouring processors in additional lists. Then, the field values of the halo cells are updated in the beginning of each runtime step which is computed on non-overlapping domains.

The initialization of parallelisation starts in the preprocessing step after the local stencils are collected. At this point, several stencils with a deformed shape exist near processor boundaries such as the blue framed stencil in figure 3.5a. Appropriate halo cells from other processors provide the necessary correction of the stencils. It is noticeable that all possible stencils with a deformed shape and therefore, acceptors for halo

cells, are included in the stencils of target cells next to processor boundaries. It implies that these acceptor cells are vice versa the only possible halo cells for stencils of other processors. Hence, the halo cells do not have to be collected separately but can just be taken from the prepared local stencil lists. This leads to the following modification for the central stencil collection algorithm in the case of using several processors:

1. For each sub-domain  $D_i$ , all cells from the stencils of target cells next to a processor boundary are gathered in a list of halo cells together with the information of the target processor. Beyond, the stencils of these cells are marked as possible acceptors for halo cells from other sub-domains. In figure 3.5a, acceptor cells of the sub-domain  $D_1$  are coloured green while its halo cells from sub-domain  $D_2$  are coloured red and vice versa the green cells are the halo cells from sub-domain  $D_1$  for the red acceptor cells of  $D_2$ .
2. The lists of halo cells are further prepared by assigning them a new ID and additionally, storing their cell centre coordinates and the coordinates of the triangles from the triangulation of the cell's boundaries. Afterwards, the lists are transmitted to the appropriate target processor using MPI.
3. The required halo cells for each marked central stencil  $S_{c,i}$  are determined by a geometrical selection due to missing face neighbour information beyond processor boundaries. For this purpose, a sphere is spanned around the target cell  $V'_i$  of  $S_{c,i}$  with the distance from the centre of  $V'_i$  to the most outer cell centre in the local stencil as the radius. All halo cells whose cell centres are located within this sphere are added to a new global stencil. In figure 3.5b, this geometrical selection results in the yellow coloured global stencil for the blue framed stencil in figure 3.5a.
4. The final stencils are attained from sorting the global stencils by distance and pick the nearest  $J_{max}$  cells. In figure 3.5b, the new stencil is framed in blue.

Additional lists with the information of the origin processor of each cell in a stencil are generated in order to transmit the field data between processors before the local reconstruction starts in each runtime step.

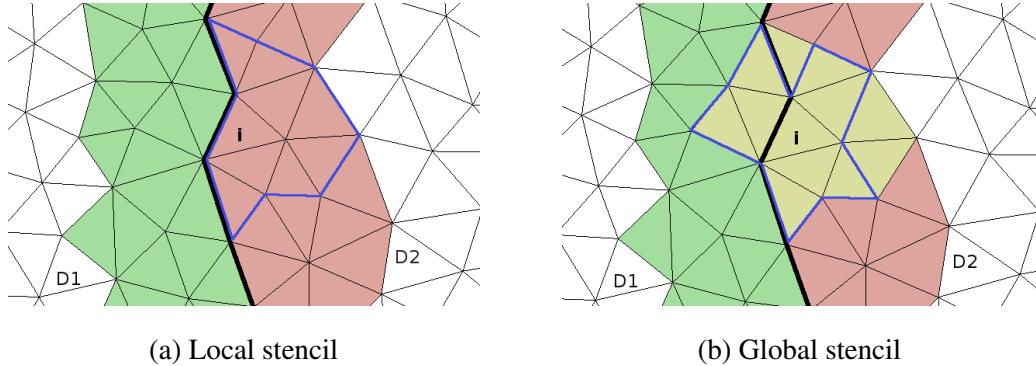


Figure 3.5: Stencil collection algorithm at processor boundaries

## 3.2 Details of Implementation

The reconstruction of any function in multiple dimensions and on unstructured meshes can be computed using the above presented WENO reconstruction method. It is successfully implemented in the framework of OpenFOAM 2.3.x. In the following, several details of implementation shall be given in order to make future developments with this code possible. The description is oriented chronological, so divided in preprocessing and runtime. Figure 3.6 shows the file structure of the implementation. A detailed documentation of the source code is provided on the enclosed disc.

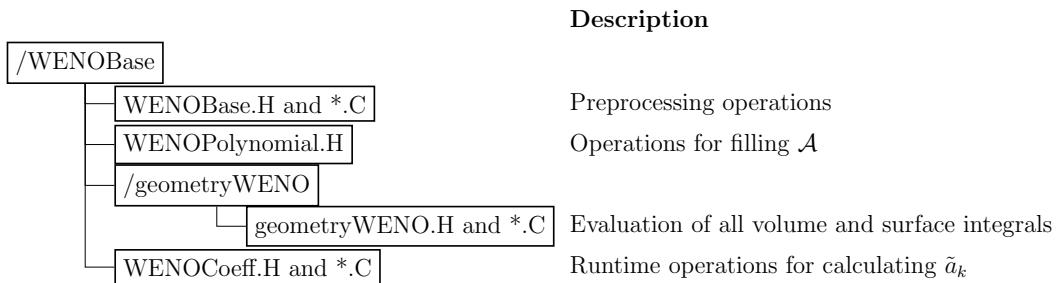


Figure 3.6: File structure of the WENO reconstruction implementation

## Preprocessing

The preprocessing of high-order WENO schemes is a time consuming procedure including the extended stencil collection, calculation of multiple smoothness and reconstruction matrices, and several SVD computations for each cell. Assuming a time-invariant mesh, all steps just have to be computed once before the first runtime reconstruction is performed. Therefore, a singleton pattern is implemented which restricts the instantiation of the preprocessing class, called `WENOBase`, to a single object [2]. To that effect, the preprocessing functions can just be called once in runtime. Here, one object is created which holds all necessary lists. In all following time steps, the singleton pattern not only prevents the creation of further instances but also provides global access to the held lists of the unique instance. The executed preprocessing steps are under consideration of chapter 3.1:

1. Generation of one large central stencil list for each control volume  $V'_i$  in the transformed space and sorting it by distance as described in chapter 3.1.1.
2. Calculation and storing of all volume integrals of the basis functions of  $V'_i$  using triangularization of the faces and a Gaussian quadrature rule of appropriate order.
3. If several processors are involved, halo cells are collected. Afterwards, appropriate cell coordinates and triangulated face coordinates are transmitted and the global stencils are gathered using the procedure of chapter 3.1.5.
4. Generation of the sectoral stencils and the final central stencil in accordance with the algorithm described in chapter 3.1.2.
5. Determination of the reconstruction matrix for each sectoral stencil of  $V'_i$  (see (3.20)). For this purpose, volume integrals of the basis functions for the cells in the stencil are calculated in the space of  $V'_i$ . Finally, the pseudoinverse  $\mathcal{A}^+$  is computed using SVD from chapter 3.1.3.
6. Calculation of the oscillation indicator matrix  $\mathcal{B}$  for each  $V'_i$  using (3.39).

The efficiency of the preprocessing is further improved by storing the lists of the most time-consuming processes as files in the constant folder of the considered case. During a restart of the simulation, the `WENOBase` class searches for these lists and read them, if available.

## Runtimes

The runtime operations of the reconstruction are performed in the `WENOCoeff` class. The input data is any variable  $\Phi$  represented by the averaged values of each cell  $\bar{\Phi}_i$ . The first step consists of calling the `WENOBase` class in order to check whether an object already exists. The constructor of the preprocessing class returns the object directly or after the preprocessed lists are newly created and read from file respectively. The necessary lists are obtained by calling a designated function using the call by reference method. Afterwards, the following runtime steps are executed:

1. Collection and transmission of the field values of the halo cells in case of parallel computing.
2. For each stencil of  $V'_i$ , generation of the vector  $b$  as the right hand side of (3.18) using  $\bar{\Phi}$ . The degrees of freedom  $a_k^{(m)}$  are computed directly from the matrix vector product in (3.35).
3. Insertion of the coefficients in (3.11) and evaluation of the smoothness indicator afterwards.
4. Calculation of the WENO weights using (3.8) and computation of the modified degrees of freedom. Under consideration of (3.13), this evaluation can be written as

$$\tilde{a}_k = \sum_{m=0}^{N_{Si}} \omega_m a_k^{(m)}. \quad (3.41)$$

All derived schemes obtain the necessary polynomials by this procedure which facilitates the addition of new schemes by users easily. The order of accuracy of the schemes are not restricted by the presented algorithm for which reason the polynomial order  $r$  is an user defined parameter. For the sake of convenience,  $r$  is defined in the default file for the selection of all schemes (`fvSchemes`). An exemplary entry looks like

```
div(phi, alpha)      Gauss WENOUpwindFit 2 0;
```

It corresponds to the discretisation of a convection term using Gauss's theorem and the later introduced `WENOUpwindFit` interpolation scheme based on a second-order polynomial and a third-order accurate scheme respectively. The last entry is related to limiting which is described in chapter 3.3.1.

### 3.3 Semi-Implicit WENO Interpolation Scheme for Convective Terms

Recalling the derivation of the discretised Navier-Stokes equations, the finite volume form of the convective term yields

$$\int_{V_i} \nabla \cdot (\rho \vec{u} \Phi) \, dV = \sum_l^{L_i} \int_{F_l} \vec{n}_l \vec{u} \Phi \, dF_l, \quad (3.42)$$

on the assumption that  $\rho = 1 \frac{kg}{m^3}$ . In case of a second-order accurate discretisation, the surface integrals in (3.42) are further simplified as derived in (2.25). In order to preserve the higher order of the interpolation scheme, the integrals have to be evaluated with a higher order of accuracy too. The velocity at the face can be taken out of the integration since (3.42) is always treated in a linearised form. Inserting the polynomial expressions

of  $\Phi$  from (3.13) in (3.42) yields at any face  $F_l$  of cell  $V_i$

$$\begin{aligned} \int_{F_l} \vec{n}_l \vec{u} \Phi \, dF_l &= \vec{u}_l \cdot \frac{\vec{n}_l |F_l|}{|F'_l|} \int_{F'_l} \Phi \, dF'_l = \frac{Q_l}{|F'_l|} \int_{F'_l} p_i \, dF'_l \\ &= Q_l \cdot \left[ \bar{\Phi}_i + \frac{1}{|F'_l|} \cdot \left( \sum_{k=1}^K \tilde{a}_{k,i} \cdot \int_{F'_l} \Omega_{k,i}(\vec{\xi}) \, dF'_l \right) \right], \end{aligned} \quad (3.43)$$

with  $Q_l$  the volumetric flow rate. The remaining surface integrals in (3.43) are solution independent and can be precomputed. Under consideration of (3.14), they can be expressed as

$$\int_{F'_l} \Omega_{k,i} \, dF'_l = \int_{F'_l} \Psi_k \, dF'_l - \frac{|F'_l|}{|V'_i|} \int_{V'_i} \Psi_k \, dV'_i. \quad (3.44)$$

The volume integrals in (3.44) are already computed during the reconstruction procedure. The surface integrals over the basis functions are evaluated in a similar way by decomposing the faces into triangles and using Gaussian quadrature rules of appropriate order. Hence, the evaluation of linearised convective terms reduces to a sum of scalar products in runtime.

It is shown in chapter 2.3.1 that the flux evaluation of linearised convective terms corresponds to the Riemann problem of the linear advection equation. Its exact solution is provided by the Riemann solver (2.35) and computed very efficiently. Therefore, this solver is appropriate for creating high-order interpolation methods based on WENO reconstructions. The so-called WENOUpwindFit arises as a high-order non-oscillatory upwind scheme extending Godunov's first-order version by higher order terms [88]. For this purpose, the constant values  $\Phi_P$  and  $\Phi_N$  are exchanged by the two reconstructed face values  $\Phi^-$  and  $\Phi^+$  (compare (3.13))

$$\begin{aligned} \Phi^-(\vec{\xi}^-) &= \bar{\Phi}_i + \sum_{k=1}^K \tilde{a}_{k,i} \cdot \Omega_{k,i}(\vec{\xi}^-), \\ \Phi^+(\vec{\xi}^+) &= \bar{\Phi}_{jl} + \sum_{k=1}^K \tilde{a}_{k,jl} \cdot \Omega_{k,jl}(\vec{\xi}^+). \end{aligned} \quad (3.45)$$

Here, the index  $i$  represents the owner and  $jl$  represents the neighbour cell of the face. The correlating reference spaces are denoted by  $\vec{\xi}^\pm$ . The resulting scheme is obtained by inserting (3.45) in the solution of the Riemann problem (2.35) which leads to

$$\int_{F_l} \vec{n}_l \vec{u} \Phi \, dF_l = \begin{cases} \int_{F_l} \vec{n}_l \vec{u} \Phi^- \, dF_l & \text{if } u_{n_l} > 0 \\ \int_{F_l} \vec{n}_l \vec{u} \Phi^+ \, dF_l & \text{if } u_{n_l} < 0 \end{cases}. \quad (3.46)$$

The surface integrals (3.46) are evaluated using (3.43) in the proper reference space.

The implementation of the `WENOUpwindFit` scheme is based on the deferred correction method of Khosla and Rubin [43] which combines an implicit first-order upwind scheme with an explicit high-order correction term. It equals the described decompositions in the Beam-Warming scheme or OpenFOAM's non-orthogonal treatment of diffusion terms. The first-order part ensures convergence due to its monotonicity and diagonal dominance. Unfortunately, as it is shown by Zhang and Shu [106, 107], WENO schemes are not strictly bounded. The correction term can, therefore, be unbounded and still influence the solution's physical reliability. This claims for additional attention as it is presented in the next chapter. Alternatively, a complete implicit scheme could be more advantageous by reason of stability at high  $Co$  numbers. However, the large stencils of high-order schemes result in many inter-cell dependencies and off-diagonal coefficients reducing the convergence ratio of the system's solution [100]. The missing monotone behaviour of WENO schemes could further influence the boundedness of the solution.

For the sake of completeness, WENO reconstructions can also be used for the implementation of central schemes. Analogue to the upwind scheme, the `WENOCentredFit` scheme is the combination of an implicit central-differencing discretisation and a central high-order correction term. The formulation equals (2.37) by exchanging the constant cell values with the polynomials (3.45). The resulting scheme is not monotone and may lead to divergence of the solution as it is remarked for the second-order formulation. Other central approaches from chapter 2.3.1, as the Lax-Friedrichs or FORCE scheme, are also possible building-blocks for central WENO schemes.

### 3.3.1 Bounded Version

In this chapter the limiting strategy of Zhang and Shu [107] is presented which can be applied to any high-order finite volume scheme in order to satisfy the maximum-principle for scalar conservation laws. This property implies that a time step's solution is bounded by the cell centred values of the previous step. Hence, it is important for the convergence to the entropy solution. All monotone and TVD schemes fulfil the maximum-principle but loose accuracy at smooth extrema due to the measuring of the total variation using the cell centred values (see also chapter 2.3.1). In contrast, the new limiter is evaluated from the maximum and minimum of the reconstruction polynomials in each cell which preserves the accuracy.

Zhang and Shu [107] proofed that the limited WENO polynomial  $\tilde{p}_i(\vec{\xi})$  fulfils the maximum-principle by applying a linear scaling limiter  $\theta_i \in [0, 1]$  to the polynomial  $p_i(\vec{\xi})$  of cell  $V'_i$  in accordance to

$$\begin{aligned}\tilde{p}_i &= \theta_i \cdot p_i + (1 - \theta_i) \cdot \bar{\Phi}_i, \\ \theta_i &= \min \left( \left| \frac{M - \bar{\Phi}_i}{M_i - \bar{\Phi}_i} \right|, \left| \frac{m - \bar{\Phi}_i}{m_i - \bar{\Phi}_i} \right|, 1 \right).\end{aligned}\quad (3.47)$$

Here,  $M$  and  $m$  are defined as the upper and lower global bounds of  $\bar{\Phi}$ . The local minimum and maximum is calculated as

$$\begin{aligned}M_i(\vec{\xi}) &= \max_{\vec{\xi} \in V'_i} p_i(\vec{\xi}), \\ m_i(\vec{\xi}) &= \min_{\vec{\xi} \in V'_i} p_i(\vec{\xi}).\end{aligned}\quad (3.48)$$

The values in (3.48) should be evaluated from the polynomials at all used Gaussian points of the later integration. In the presented interpolation scheme, the surface integration is precomputed for which reason this handling would be inefficient. It is, therefore, decided to take the surface integrated values of  $p_i$  into account instead. This decision is towards the underlying mathematics, but on the other hand, the values of  $M$

and  $m$  are also just available at the cell centres in OpenFOAM. Hence, equation (3.48) is evaluated as

$$\begin{aligned} M_i(\vec{\xi}) &= \max_l \left( \frac{1}{|F'_l|} \int_{F'_l} p_i(\vec{\xi}) \, dF'_l \right), \\ m_i(\vec{\xi}) &= \min_l \left( \frac{1}{|F'_l|} \int_{F'_l} p_i(\vec{\xi}) \, dF'_l \right), \quad l = 1, \dots, L_i. \end{aligned} \quad (3.49)$$

By limiting the integrated polynomials, the interpolation scheme can be rewritten as a sum of a first-order upwind scheme and a limited high-order correction, similar to (3.43).

For the sake of practicability, another user specified parameter  $\theta_2 \in [0, 1]$  is introduced which provides the switch between a limited ( $\theta_2 = 1$ ) and unlimited ( $\theta_2 = 0$ ) computation. The final fluxes of the `WENOUpwindFit` scheme become

$$\begin{aligned} \int_{F_l} \vec{n}_l \vec{u} \Phi \, dF_l &= Q_l \cdot \left[ \theta_2 \cdot \left\{ \bar{\Phi}_i + \frac{\theta_i}{|F'_l|} \cdot \left( \sum_{k=1}^K \tilde{a}_{k,i} \cdot \int_{F'_l} \Omega_{k,i}(\vec{\xi}) \, dF'_l \right) \right\} + \right. \\ &\quad \left. (1 - \theta_2) \cdot \left\{ \bar{\Phi}_i + \frac{1}{|F'_l|} \cdot \left( \sum_{k=1}^K \tilde{a}_{k,i} \cdot \int_{F'_l} \Omega_{k,i}(\vec{\xi}) \, dF'_l \right) \right\} \right]. \end{aligned} \quad (3.50)$$

The value of  $\theta_2$  is specified as the last entry behind the name of the scheme in `fvSchemes` (see also chapter 3.2).

An alternative approach could be the so-called WENO-TVD scheme based on a high-order TVD formulation for fluxes. Titarev and Toro [87] proved its TVD properties and at the same time the reduction of the typical drawbacks of second-order TVD schemes. The presented results are improved in comparison to unlimited WENO schemes but the efficiency is even worse. The implementation in the given framework would be cumbersome since TVD schemes are based on a linear interpolation as written in (2.39) which is contrary to the upwind approach of the WENO-TVD scheme. Further, it is based on an unsuitable combination of spatial and temporal discretisation.

### 3.3.2 Application to the Advection Equation

The principal working of the described WENO interpolation scheme is presented for the advection equation which arises in the later introduced conservative level set method. Further, the presented limiter strategy is verified. For a more detailed validation process of the WENO reconstruction method, see also the author's previous project [50].

The advection equation (2.30) is introduced in chapter 2.3.1 for describing different discretisation techniques for linearised convective terms. Here, the spatial discretisation of its finite volume formulation (2.31) is handled by the `WENOUpwindFit` interpolation scheme. The temporal discretisation is performed explicitly using the TVD third-order Runge-Kutta method (2.56) from chapter 2.3.4. The time step has to be chosen without violating the stability of WENO which is a restriction on the CFL-condition to  $CFL \leq \frac{1}{3}$  in three dimensions [86]. In the following computations, this restriction is hold although they are two-dimensional.

#### Transport of Prescribed Inlet Profiles

First, the performance for two convection test cases of Jasak [41], which are generated to investigate the numerical diffusion of different convection schemes, are evaluated. For this purpose, the left boundary of a squared domain ( $1\text{ m} \times 1\text{ m}$ ) is taken as an inlet with a predefined profile as shown in figure 3.7. The two considered profiles are an *ellipse*-profile

$$\Phi(x=0, y) = \begin{cases} \sqrt{1 - \left(\frac{y - \frac{1}{3}}{\frac{1}{6}}\right)^2} & \text{if } \frac{1}{6} \leq y \leq \frac{1}{2}, \\ 0 & \text{else} \end{cases}, \quad (3.51)$$

and a *step*-profile resulting in the transport of  $\Phi = 0.5$  as a physical surface

$$\Phi(x=0, y) = \begin{cases} 0 & \text{if } \frac{1}{6} \leq y \leq 1 \\ 1 & \text{else} \end{cases}. \quad (3.52)$$

In the whole domain, the velocity is prescribed as  $\vec{u} = (\frac{\sqrt{3}}{2} \frac{m}{s}, \frac{1}{2} \frac{m}{s})$  leading to a transport of  $\Phi$  from the left to the right with an angle of  $30^\circ$ . The considered meshes are two-dimensional and consist of 2,500 quadratic and 2,832 uniform triangular cells respectively. The simulations are completed after steady-state solutions are obtained. The evaluation of the results is carried out graphically by plotting the steady-state solution of  $\bar{\Phi}$  along the y-axis at  $x = 0.5 \text{ m}$  and a comparison with the analytical results from (3.51) and (3.52).

The same test cases were already performed in the previous project [50]. The focus was on the comparison of ENO and WENO upwinding schemes with different order of accuracy. Thereby, additional two- and three-dimensional profiles were investigated. Here, the focus is on the comparison of the fourth-order accurate WENOUpwindFit scheme in its unlimited and limited version, denoted as WENO3, with the TVD scheme with van Leer's limiter, denoted as TVD-vanLeer.

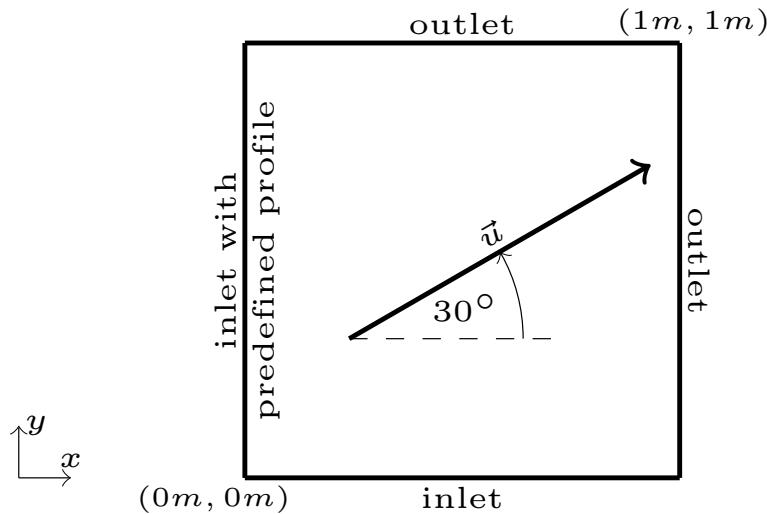


Figure 3.7: Setup for the two-dimensional transport of prescribed profiles

The simulation results are presented in figure 3.8 and 3.9. On the structured grids, both convection schemes predict similar profiles. Obviously, the second-order accurate

scheme benefits from the high mesh resolution resulting in less improvement by applying even higher orders. On the unstructured grids, the transformations into a reference space leads to improved results using the WENO scheme. The missing collinear cells in the stencils lead to lower condition numbers for the reconstruction matrices and thereby, a more accurate solution, even though the mesh resolution is lower than on the structured grid. The results for the TVD-vanLeer scheme are similar to its results on the structured grid which is less accurate than WENO schemes. For both simulations, limiting the WENO polynomials would not be necessary since the results are as bounded as of the TVD scheme. But, it is noticeable that the limited version loses no accuracy corresponding to the theoretically intended effect.

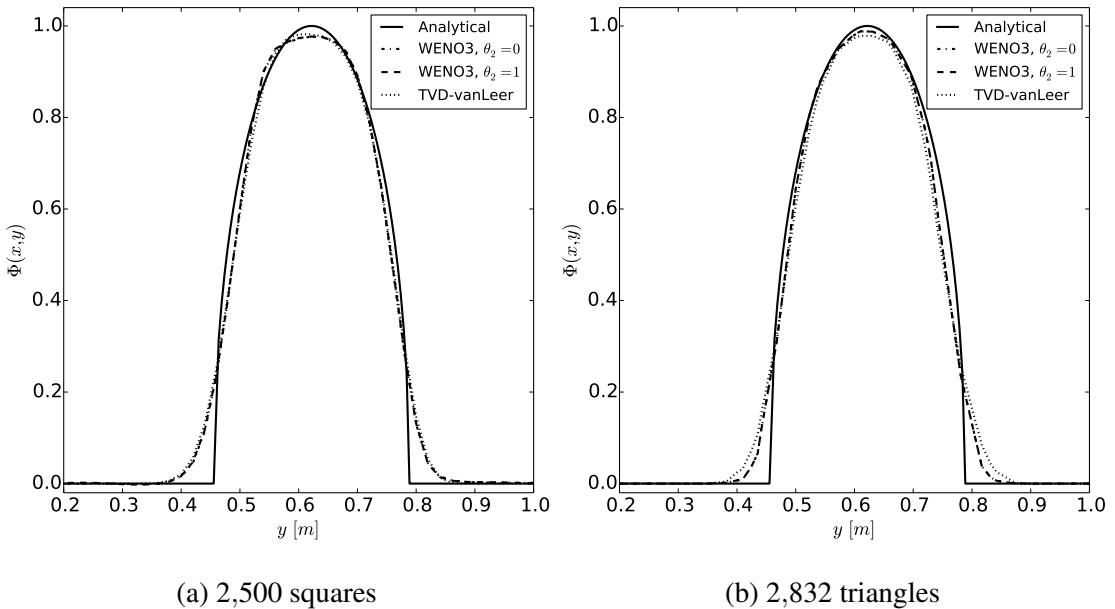


Figure 3.8: Slice of the *ellipse*-profile at  $x = 0.5 \text{ m}$

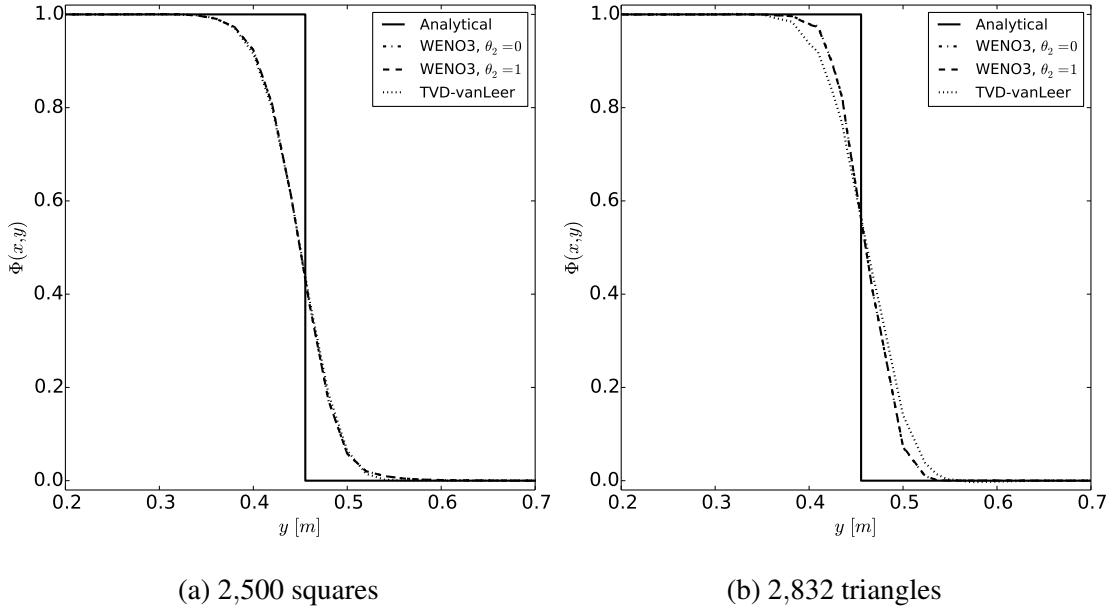


Figure 3.9: Slice of the *step*-profile at  $x = 0.5 \text{ m}$

### Rotation of Zalesak's Disk

The next test case is the rotation of a slotted disk designed by Zalesak [105] in order to verify the performance of his flux-corrected transport scheme. It is a classical case for the comparison of different convection schemes due to its complex shape. As it can be seen in figure 3.10, a disk with diameter  $0.6 \text{ m}$  is centred on  $(0 \text{ m}, 0.5 \text{ m})$  in a square, two-dimensional domain of  $2 \text{ m} \times 2 \text{ m}$  delimited by the points  $(-1 \text{ m}, -1 \text{ m})$  and  $(1 \text{ m}, 1 \text{ m})$ . The difficulties of the shape arise from subtracting a vertical rectangle of  $0.1 \text{ m} \times 0.5 \text{ m}$  from the disk resulting in a slotted disk with sharp corners and thin slot [70]. The initial field is set to be  $\Phi = 1$  in the disk and  $\Phi = 0$  in the rest of the domain. The velocity field is defined as  $\vec{u} = (-2\pi y \frac{\text{m}}{\text{s}}, 2\pi x \frac{\text{m}}{\text{s}})$  which leads to an off-centred rotation of the disk. At the end of the simulation, after  $t = 1 \text{ s}$ , the profile is rotated back into the initial position. The considered meshes are two-dimensional and consist of quadratic and uniform triangular cells respectively. The number of cells is fixed to

$128 \times 128$  and  $256 \times 256$ . The results are presented graphically as a comparison of the contours  $\Phi = 0.5$  and slices of the convected profile at  $y = 0.6 \text{ m}$  with the corresponding analytical results.

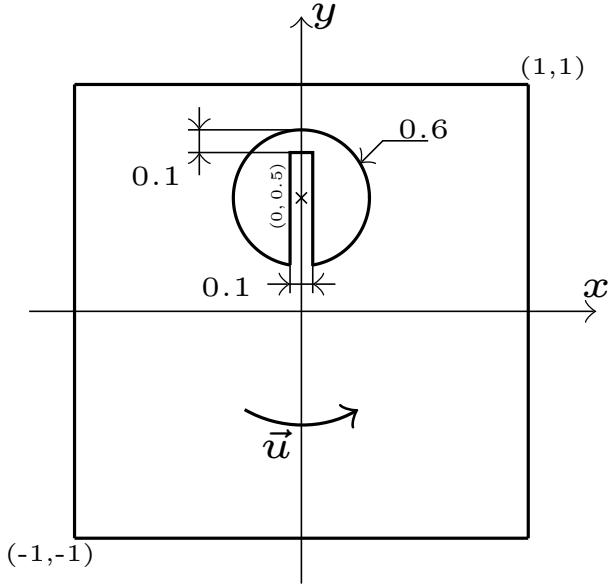


Figure 3.10: Setup for the rotation of Zalesak's disk (all measurements in metres)

The results on the coarse structured grid in figure 3.11 show superior performance of the WENO schemes to the TVD scheme. The contour's sharp corners are approximated more accurate and the slice shows an improved approximation of the slot due to less numerical diffusion. The TVD results are smeared because of the degeneration to first-order accuracy near the extrema. Further, the overshoots of the unlimited WENO scheme are removed by the limiter, whereas the remaining distribution is preserved. Similar results arise on the finer mesh in figure 3.12 which shows a great accordance of the WENO computations with the analytical results. The mesh resolution yields an improvement of the TVD scheme too. It is, however, still worse than the fourth-order accurate WENO scheme especially in the slot. The refinement also reduces the overshoots of the unlimited WENO scheme and they disappear after limiting.

In comparison, the calculations on the unstructured grids are generally improved as it is already stated for the previous case. On the coarse grid (figure 3.11) the overshoots of the WENO scheme are about 6% and removed after limiting the polynomials. The WENO profiles are less smeared than the profiles of the TVD-vanLeer scheme. The refinement of the mesh renders limiting almost unnecessary.

In conclusion, all simulations of the rotation of Zalesak's disk indicate the improved performance of high-order WENO schemes in comparison to TVD schemes. The lower order scheme need an increase of the mesh resolution in order to obtain similar accuracy of the results. Unfortunately, this deficit is counterbalanced by the increased computation time which is about 60 times higher if WENO reconstructions are applied (see also chapter 6.5).

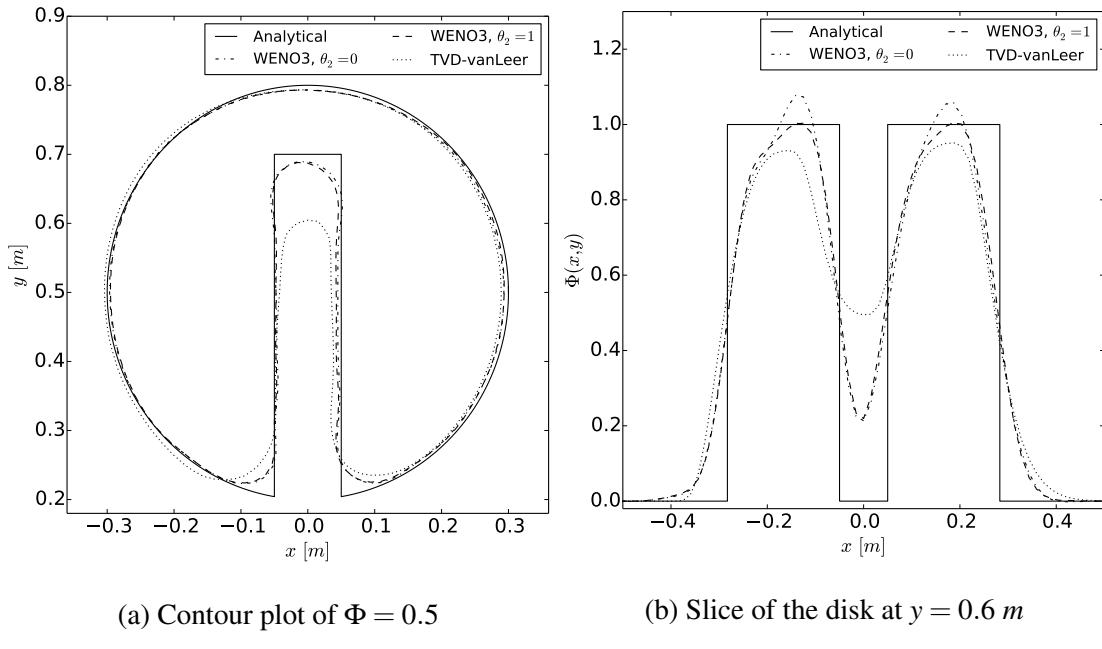
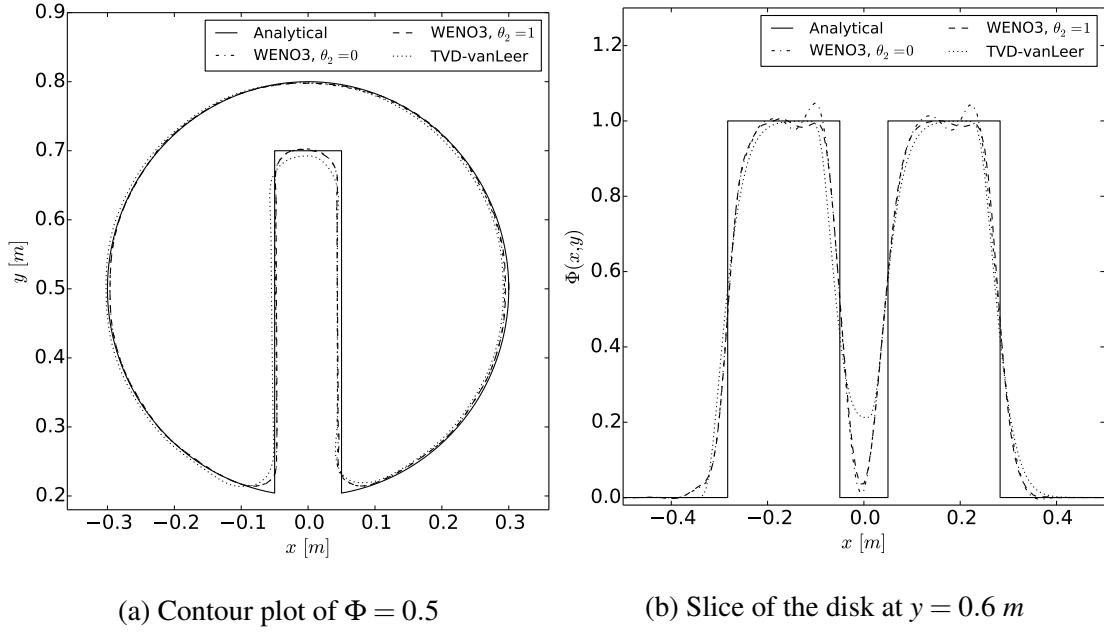
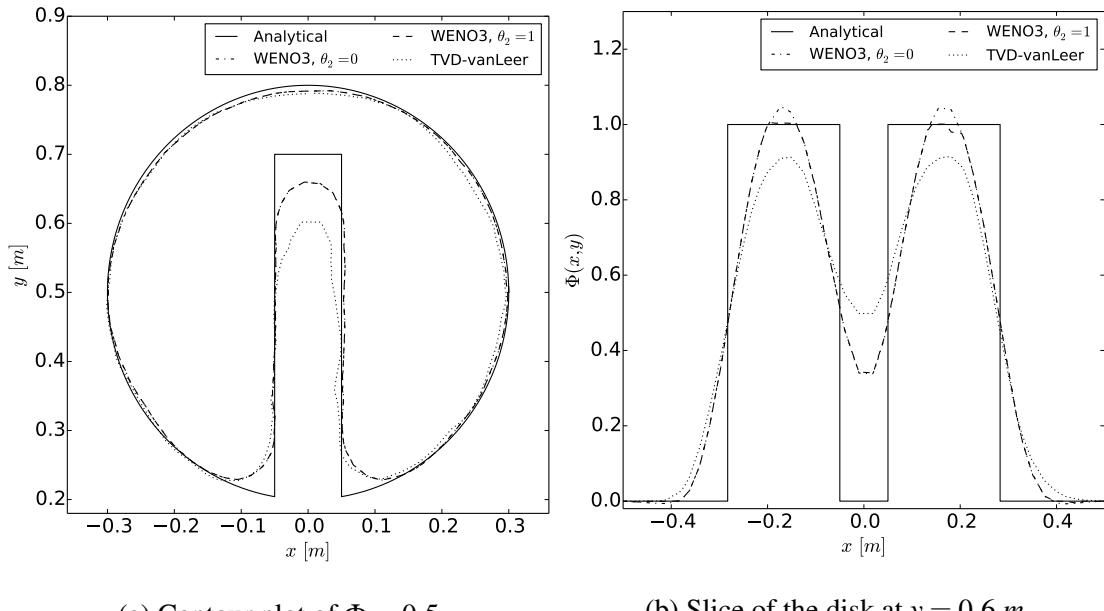


Figure 3.11: Results of convection schemes for Zalesak's disk on  $128 \times 128$  squares

Figure 3.12: Results of convection schemes for Zalesak's disk on  $256 \times 256$  squaresFigure 3.13: Results of convection schemes for Zalesak's disk on  $128 \times 128$  triangles

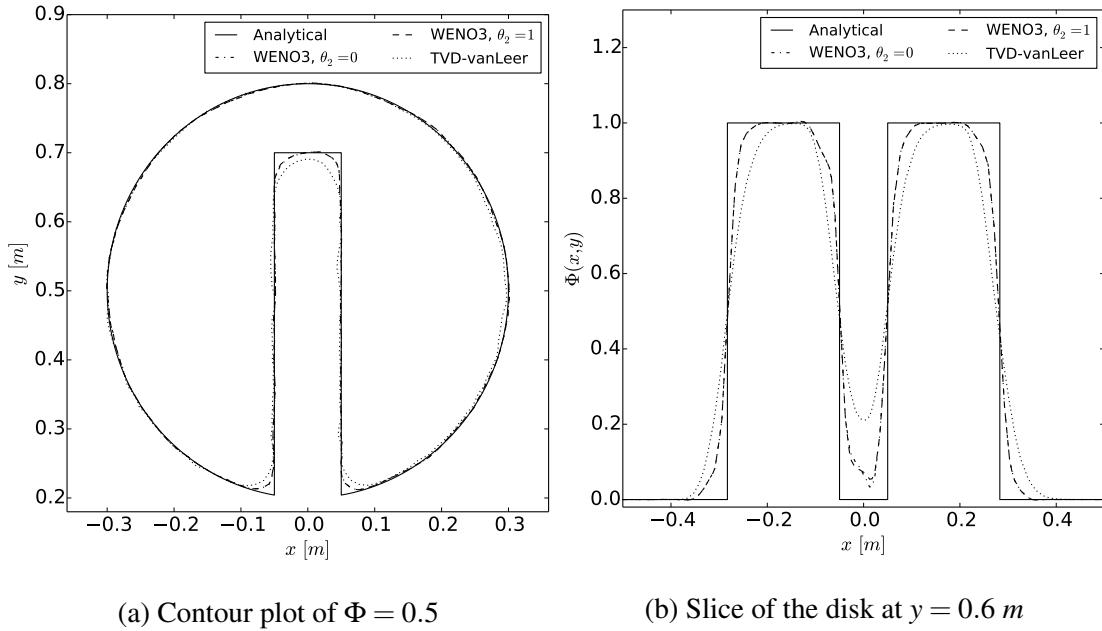


Figure 3.14: Results of convection schemes for Zalesak's disk on  $256 \times 256$  triangles

## 3.4 WENO Gradient Scheme

The calculation of gradients in the cell centres is a frequent operation in FVM. For multiphase flows, the accuracy of the calculations of interface normal vectors is mainly influenced by the gradient of the color function. This gradient is large near the interface which can lead to oscillatory solutions. The common gradient calculation is based on Gauss's theorem and linear interpolations as described in chapter 2.3.2. It was already stated that central differencing does not provide any control over a non-oscillatory solution for which reason the creation of a WENO gradient scheme is advantageous.

The starting point of the gradient calculation in the cell centre of  $V_i$  is its finite volume formulation

$$(\nabla\Phi)_i = \frac{1}{|V_i|} \int_{V_i} \nabla\Phi \, dV. \quad (3.53)$$

The volume integral in (3.53) can be evaluated with a high-order accuracy in two ways. One opportunity is the transformation into surface integrals as executed in (2.51) and a high-order interpolation of  $\bar{\Phi}$  to the boundaries using e.g. the WENOCentredFit scheme. This method is applicable without any further implementations. A more efficient computation is the direct evaluation of the volume integral by replacing the gradient of  $\Phi$  by its polynomial representation using WENO reconstructions. Here, the difficulty is the correct definition of the Gaussian points in arbitrary shaped volumes which could be obtained by decomposing each cell in tetrahedra where the Gaussian points are known. The coordinates and weights could be stored in the preprocessing resulting in higher efficiency in runtime. However, the additional tetrahedralization is a time-consuming computation and should be avoided. If the non-oscillatory behaviour of the scheme is in the first place and the theoretical order of accuracy is less important, a more efficient gradient scheme can be generated by replacing the gradient of  $\Phi$  by its polynomial representation and evaluating the volume integral with second order of accuracy. It corresponds to the evaluation of the gradient at the centre  $\vec{x}_i$  of a cell according to

$$\frac{1}{|V_i|} \int_{V_i} \nabla \Phi(\vec{x}) \, dV \approx \nabla \Phi|_{\vec{x}=\vec{x}_i}. \quad (3.54)$$

The resulting gradients are a practical compromise between accuracy, stability and efficiency due to its simple evaluation without any integrations. They have to be further transformed into the reference space in accordance with the polynomials of the WENO reconstruction. It is accomplished by writing [70]

$$\nabla_{\vec{x}} \Phi(\vec{x}) = (\mathcal{J}^{-1})_i^T \cdot \nabla_{\vec{\xi}} \Phi(\vec{\xi}), \quad (3.55)$$

with  $\nabla_{\vec{\xi}}$  the gradients in the principal directions of the reference space of  $V'_i$ . The gradient at a cell centre yields due to the affine transformation

$$\nabla_{\vec{x}} \Phi|_{\vec{x}=\vec{x}_i} = (\mathcal{J}^{-1})_i^T \cdot \nabla_{\vec{\xi}} \Phi|_{\vec{\xi}=\vec{\xi}_i}. \quad (3.56)$$

Without presenting the detailed derivation, the next steps are the replacement of  $\Phi$  by its polynomial formulation (3.13), taking the derivatives and inserting  $\vec{\xi} = \vec{\xi}_i$ . The formula can then be simplified due to the cell centred orthogonal basis functions. The remaining non-zero terms are related to the coefficients of the first-order terms of the polynomial. Finally, a simple expression can be found which reads as follows

$$\nabla_{\vec{x}} \Phi|_{\vec{x}=\vec{x}_i} = (\mathcal{J}^{-1})_i^T \cdot \begin{pmatrix} \tilde{a}_3 \\ \tilde{a}_2 \\ \tilde{a}_1 \end{pmatrix}_i. \quad (3.57)$$

The inverse Jacobian matrices are already calculated in the preprocessing step. The resulting gradient scheme, called `WENOGrad`, can be applied to any gradient computation in OpenFOAM due to its explicit treatment in any case.

The performance of the scheme shall be presented for a smooth, three-dimensional function with large gradients in all three coordinate directions. For this purpose, the fourth-order accurate `WENOGrad` scheme and OpenFoam's standard method, selectable as `Gauss linear`, are applied to a harmonic function which is defined as

$$\Phi_{exact}(x, y, z) = y \cdot \cos(4x) + z \cdot \sin(10y) + x \cdot \cos(3z). \quad (3.58)$$

The solution of the gradients of (3.58) is computed on a cubic domain with an edge length of 1 m and centred on  $x = y = z = 0.5$  m. The considered meshes are a structured grid with 8,000 hexahedra and an unstructured grid with 7,982 tetrahedra. The results are represented by a plot of the gradients of  $\bar{\Phi}$  along the diagonal from (0 m, 0 m, 0 m) to (1 m, 1 m, 1 m) and can be seen in figure 3.15. Here, all three gradients are pictured in a single figure for each mesh.

The superior results of the WENO gradient scheme reproduce almost the analytical solution regardless of the considered direction or mesh. In comparison, the linear scheme has several inaccuracies, particularly on the unstructured grid and near boundaries.

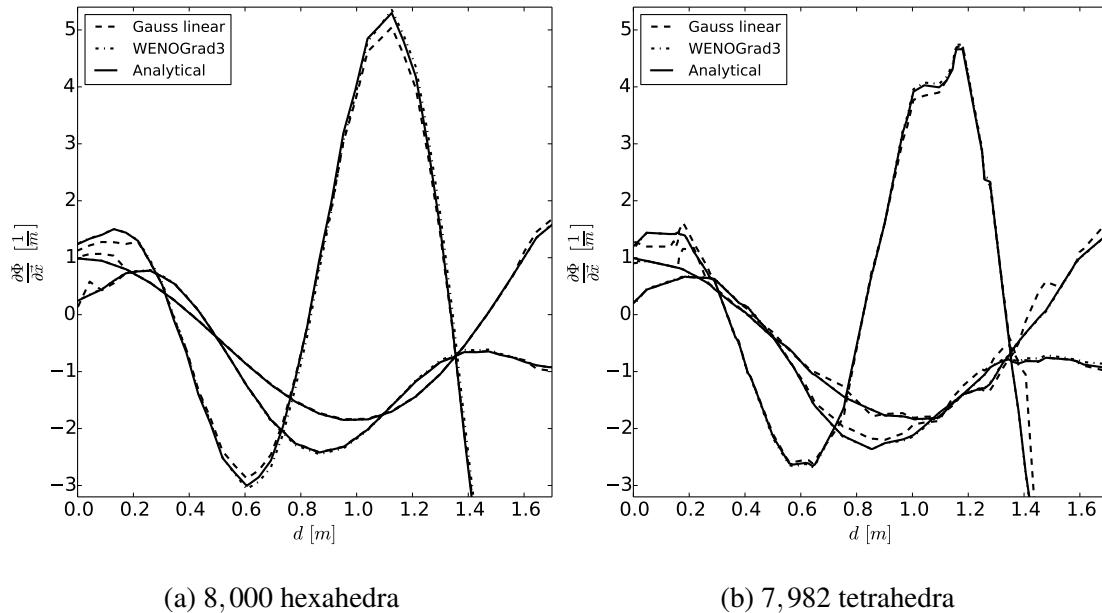


Figure 3.15: Results of the gradient calculation in the principal directions along the diagonal  $d$

### 3.4.1 Surface Normal Gradient Extension

The standard evaluation of surface normal gradients is explained in chapter 2.3.2 and given by (2.48). Alternative schemes can be implemented by providing the necessary data for the following expression which is similar to the default interpolation formula (2.26)

$$\vec{n}_l (\nabla \Phi)_l = \Delta'_l (\Phi_N - \Phi_P) + C_{exp}, \quad (3.59)$$

with  $\Delta'_l$  as coefficients defined as  $\frac{1}{\Delta_l}$  for the standard scheme and  $C_{exp}$  an explicit correction term.

The only possibility of increasing the order of accuracy of this operator using WENO reconstructions is to set  $\Delta'_l = 0$  and insert the whole computation in the correction term. This procedure is arguable as long as the surface normal gradient is treated explicitly as

in some interface models for the calculation of the surface tension forces or in the PISO algorithm for the pressure-velocity coupling. Here, the explicit term is taken on the right hand side of the system of equations and therefore, does not influence the structure of the matrix. It is further noticeable that the resulting scheme is, in general, not recommended for the discretisation of diffusion terms due to its missing implicit part. It is, however, conceivable that (semi-)implicit WENO schemes would reduce the need of the common non-orthogonal correction steps due to the bigger numerical molecule and the validity of a polynomial in the whole cell irrespective of its shape. Unfortunately, the improper formulation (3.59) prevents its implementation and leads to the requirement of reformulating the surface normal gradient and diffusion operator. In order to stay in the default framework, WENO can just be taken into account for diffusion terms by replacing (2.51) with the WENOGrad scheme in the course of determining the gradients of the non-orthogonal correction term.

Under the consideration of the derivation of the WENO gradient scheme in the previous chapter, the surface normal gradient is calculated using FVM and WENO reconstructions as

$$\begin{aligned}\vec{n}_l (\nabla \Phi)_l &= \frac{\vec{n}_l}{F_l} \int_{F_l} \nabla_{\vec{x}} \Phi(\vec{x}) \, dF_l = \frac{\vec{n}_l}{F'_l} \int_{F'_l} \nabla_{\vec{x}} \Phi(\vec{x}) \, dF'_l \\ &= \frac{\vec{n}_l}{F'_l} \cdot (\mathcal{J}^{-1})_i^T \int_{F'_l} \nabla_{\vec{\xi}} \Phi(\vec{\xi}) \, dF'_l.\end{aligned}\quad (3.60)$$

The gradient of  $\Phi$  is expressed by the gradient of its polynomial formulation (3.13) which yields for  $V'_i$

$$\nabla_{\vec{\xi}} \Phi(\vec{\xi}) = \nabla_{\vec{\xi}} \left( \bar{\Phi}_i + \sum_{k=1}^K \tilde{a}_k \cdot \Omega_k(\vec{\xi}) \right) = \sum_{k=1}^K \tilde{a}_k \nabla_{\vec{\xi}} \Omega_k(\vec{\xi}) = \sum_{k=1}^K \tilde{a}_k \nabla_{\vec{\xi}} \Psi_k(\vec{\xi}), \quad (3.61)$$

since all constant terms vanish. Inserting (3.61) in (3.60) results in the final formulation of the so-called WENO<sub>n</sub>Grad scheme

$$\vec{n}_l (\nabla \Phi)_l = \vec{n}_l \cdot \sum_{k=1}^K \tilde{a}_k \cdot \left( \frac{(\mathcal{J}^{-1})_i^T}{F'_l} \int_{F'_l} \nabla_{\vec{\xi}} \Psi_k(\vec{\xi}) \, dF'_l \right). \quad (3.62)$$

The bracket term is solution independent and can be precomputed. Here, the surface integrals can be evaluated during the preprocessing of the WENO interpolation schemes without additional computational cost. The final solution at a face is taken as a linear combination of the surface normal gradients of the two adjacent cells. Alternative combinations could be implemented if necessary.

### 3.5 High-Order Temporal Discretisation Using ADER

The temporal discretisation of the explicit solution of the advection equation using the third-order accurate TVD-Runge-Kutta method needs three sub-steps at each time step. Runge-Kutta methods of even higher order in combination with WENO schemes are usually not considered due to the time-consuming reconstruction which has to be executed in each sub step. Therefore, Titarev and Toro [85] developed arbitrary high order schemes using derivatives (ADER) in order to reach increased order of accuracy in space and time without a drastic decrease of efficiency.

An one step formulation of the advection equation is taken into account which can be written under consideration of (2.54) as

$$\begin{aligned}\Phi_i^{(n+1)} &= \Phi_i^{(n)} + \frac{\Delta t}{|V_i|} \cdot \sum_l^{L_i} \frac{1}{\Delta t} \int_t^{t+\Delta t} \left( \int_{F_l} \vec{n}_l \vec{u} \Phi \, dF_l \right) dt \\ &= \Phi_i^{(n)} + \frac{\Delta t}{|V_i|} \cdot \sum_l^{L_i} \tilde{F}_l.\end{aligned}\quad (3.63)$$

The task is the evaluation of the fluxes  $\tilde{F}_l$  in space and time with a higher order of accuracy. For this purpose, the ADER approach solves generalised Riemann problems with the initial condition consisting of two smooth polynomials [90]. Here, the presented WENO reconstructions opens up as an appropriate method. The generalisation of the classical Riemann problem consists of replacing the constant initial values in (2.32) by the polynomials as for the WENO interpolation scheme. The problem can once again be solved in the direction normal to the face. For the time integration, a numerical flux

has to be calculated at each face as a function of time. In accordance with Titarev and Toro [85], a truncated Taylor expansion of the interface state of  $\Phi$  in time is used. It can be written for any point at the face as

$$\Phi_l(t) = \Phi_l(0+) + \sum_{m=1}^r \frac{\partial^m \Phi_l(0+)}{\partial t^m} \frac{t^m}{m!}. \quad (3.64)$$

The leading term  $\Phi_l(0+)$  accounts for the first-order interaction of the boundary extrapolated values corresponding to the classical Riemann problem with Godunov's upwind scheme as the solution for the flux. Further, all temporal derivatives in (3.64) are replaced by spatial derivatives using the Cauchy–Kowalewski procedure (see the book of Evans [22] for details) which results in

$$\frac{\partial^m \Phi_l}{\partial t^m} = (-u_{n_l})^m \cdot \frac{\partial^m \Phi_l}{\partial n_l^m}. \quad (3.65)$$

Next, an evolution equation has to be found for each spatial derivative and a classical Riemann problem has to be solved for its linearised version. In case of the linear advection equation, Toro [90] showed that this step simplifies to the same classical Riemann problems as for the leading term. Hence, the sought fluxes are calculated from the solution of one simple Riemann problem (compare (3.46))

$$\tilde{F}_l = \frac{1}{\Delta t} \cdot \int_t^{t+\Delta t} \left\{ \begin{array}{ll} \int_{F_l} \vec{n}_l \vec{u} \Phi^- dF_l + \sum_{m=1}^r (-u_{n_l})^m \int_{F_l} \frac{\partial^m \vec{n}_l \vec{u} \Phi^-}{\partial n_l^m} \frac{t^m}{m!} dF_l & \text{if } u_{n_l} > 0 \\ \int_{F_l} \vec{n}_l \vec{u} \Phi^+ dF_l + \sum_{m=1}^r (-u_{n_l})^m \int_{F_l} \frac{\partial^m \vec{n}_l \vec{u} \Phi^+}{\partial n_l^m} \frac{t^m}{m!} dF_l & \text{if } u_{n_l} < 0 \end{array} \right\} \cdot dt. \quad (3.66)$$

Here,  $\Phi^\pm$  are the polynomials from (3.45). The time integration can be provided by applying a Gaussian quadrature rule or using the flux-expansion approach [84], in which a truncated Taylor time expansion of the physical flux is sought similar to (3.64). The final flux expression can be written under consideration of the linearity as

$$\tilde{F}_l = u_{n_l} \cdot \left\{ \begin{array}{ll} \int_{F_l} \Phi^- dF_l + \sum_{m=1}^r (-u_{n_l})^m \int_{F_l} \frac{\partial^m \Phi^-}{\partial n_l^m} dF_l \frac{(\Delta t)^m}{(m+1)!} & \text{if } u_{n_l} > 0 \\ \int_{F_l} \Phi^+ dF_l + \sum_{m=1}^r (-u_{n_l})^m \int_{F_l} \frac{\partial^m \Phi^+}{\partial n_l^m} dF_l \frac{(\Delta t)^m}{(m+1)!} & \text{if } u_{n_l} < 0 \end{array} \right.. \quad (3.67)$$

The computation of the surface integrals of the leading term is already discussed for interpolation schemes (see (3.43)). The remaining integrals are evaluated as

$$\begin{aligned} \int_{F_l} \frac{\partial^m \Phi^-}{\partial n_l^m} dF_l &= \frac{|F_l|}{|F'_l|} \int_{F'_l} \frac{\partial^m}{\partial n_l^m} \left( \bar{\Phi}_i + \sum_{k=1}^K \tilde{a}_k \Omega_k \right) dF'_l \\ &= \frac{|F_l|}{|F'_l|} \sum_{k=1}^K \tilde{a}_k \int_{F'_l} \left( \frac{\partial^m \Psi_k}{\partial n_l^m} \right) dF'_l. \end{aligned} \quad (3.68)$$

The integrals over the basis functions are precomputed similar to the volume integrals of the smoothness indicator matrices in chapter 3.1.4.

The resulting scheme, called `ADERUpwindFit`, can only be applied to the linear advection equation as it is executed for the level set function in the two-phase flow solver of the next chapter. The accuracy of the scheme is comparable to the WENO scheme with Runge-Kutta time integration. However, the efficiency is improved in runtime since only one reconstruction per time step has to be computed and the most costly parts are shifted to preprocessing. The overall computational time including preprocessing is decreased by e.g. one third for the test cases of chapter 3.3.2.

# Chapter 4

## A Two-Phase Flow Solver Using a Level Set Method

This chapter aims at developing a Navier-Stokes solver for incompressible two-phase flows. For this purpose, the presented Navier-Stokes equations (chapter 2.1) are combined with the CSF model (chapter 2.2) and the level set interface model of Osher and Sethian [66]. The general discretisation methods of chapter 2.3 are applied to the equations and the modified PISO-algorithm of Weller is used in order to obtain a linear, algebraic system of equations which is solved iteratively. It will be shown that the resulting `lsFoam` solver suffers the common problem of loss of mass due to the non-conservative formulation of the standard level set method. Further, the problematic on arbitrary meshes will be discussed in detail.

### 4.1 Derivation of the Interface Model

In the level set method, the interface between two fluids  $\Gamma$  is described by an implicit function  $\varphi(\vec{x})$ . Osher and Sethian [66] defined this so-called level set function as a smooth distance function which equals zero for  $\vec{x} \in \Gamma$  or the shortest distance  $\min(\vec{x} -$

$\vec{x}_\Gamma$ ) elsewhere. The introduction of signs leads to positive distances in one fluid and negative distances in the other one and hence, avoids large gradients and discontinuities in the function [82]. The definition of the level set to be a signed distance function reads

$$|\nabla \varphi| = 1. \quad (4.1)$$

The implicit representation of the surface prevents the problems arising in front tracking methods, which describes the surface explicitly and evolves it using the kinematic boundary condition for interfaces in a Lagrangian formulation [65] (see also chapter 1.1). In contrast, the motion of  $\varphi$  and thereby, the evolution of the interface is calculated in an Eulerian grid using the advection equation (2.30). This level set equation is characterised as a linear conservation law and yields in its conservative form

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot (\vec{u} \varphi) = 0. \quad (4.2)$$

Assuming incompressible fluids, the discretisation procedure of this equation is already described and verified in the previous chapters. A transport of  $\varphi$  using low-order convection schemes leads to a smeared solution and loss of mass due to missing counteractions. Therefore, the ADERUpwindFit scheme of chapter 3.5 is applied here due to its high-order accuracy and improved efficiency in comparison to WENO schemes with Runge-Kutta methods. This choice, however, limits the time step to  $Co \leq \frac{1}{3}$  due to the explicit discretisation. Further, even high-order convection schemes can not conserve the level set as a signed distance function due to numerical diffusion and merging, spreading and jumping of the interface from high velocity gradients [70, 82]. As a result, a redistancing algorithm has to be introduced after each time step to keep  $\varphi$  satisfying (4.1).

Sussman, Smereka, and Osher [82] developed a partial differential equation whose steady-state solution converges to (4.1). Taking  $\varphi_0$  as the solution of the advection step,

the corrected level set  $\varphi$  arises from the equation

$$\begin{aligned} \text{PDE: } \frac{\partial \varphi}{\partial \tau} &= S_\varepsilon(\varphi_0) \cdot (1 - |\nabla \varphi|), \\ \text{IC: } \varphi(\tau = 0) &= \varphi_0, \end{aligned} \quad (4.3)$$

with  $\tau$  the artificial time and  $S$  the sign function which is modified as

$$S_\varepsilon(\varphi_0) = \frac{\varphi_0}{\sqrt{\varphi_0^2 + \varepsilon}}, \quad \varepsilon \approx 10^{-5}, \quad (4.4)$$

for the sake of numerical stability. The nature of the PDE is highlighted by inserting the identity

$$|\nabla \varphi| = \frac{|\nabla \varphi|^2}{|\nabla \varphi|} = \frac{\nabla \varphi}{|\nabla \varphi|} \cdot \nabla \varphi, \quad (4.5)$$

in (4.3). It yields after rearranging

$$\frac{\partial \varphi}{\partial \tau} + \vec{w} \cdot \nabla \varphi = S_\varepsilon(\varphi_0), \quad (4.6)$$

with  $\vec{w}$  the unit normal given by

$$\vec{w} = S_\varepsilon(\varphi_0) \cdot \frac{\nabla \varphi}{|\nabla \varphi|}, \quad (4.7)$$

and always pointing outward from the interface. The formulation (4.6) indicates the redistancing equation as a non-linear, non-conservative hyperbolic equation with a source term and the characteristics  $\vec{w}$  propagating in the direction normal to the interface with speed one [80]. The redistancing of  $\varphi$  evolves from the zero level set and in each solving step, another layer of cells is completed (if  $Co = 1$ ). Here, a narrow band of corrected level set around the interface is already sufficient for a stable simulation. As an additional condition, the zero level set should be fixed during the redistancing which is not fulfilled in any case of discretising (4.6). The shift of the interface is a common problem in level set methods and the main reason for loss of mass. Many researchers tried to

overcome this problem as Sussman and Fatemi [80] with a gradient projection method or Russo and Smereka [73] with a sub-cell fix whose formula is given below.

The obvious idea of discretising (4.6) would be based on FVM in order to apply available methods and preserve the arbitrary mesh handling of OpenFOAM. The characteristic speed of the hyperbolic equation  $\vec{w}$  depends on the solution which produces distortions, such as shock and rarefaction waves, during the transport process of  $\varphi$  [90]. So far, all presented convection schemes were described as convergent to the physical reliable, entropy solution if they are monotone. But generally, this convergence also depends on an additional entropy condition which is related to the speed of discontinuities. Linear conservation laws fulfil this condition automatically due to their constant characteristic speed for which reason it was not considered before. On the contrary, non-linear conservation laws have to take this condition into account at discontinuities. As it is shown in chapter 5.2.2, a corresponding algebraic expression can be derived from the integral form of the conservation law.

Unfortunately, equation (4.6) is a non-conservative formulation which results in a so-called generalised Rankine-Hugoniot condition at the discontinuity. An a-priori unknown path has to be determined connecting the two states on both sides of the discontinuity in the phase-space. The advisable theory was developed by Maso, LeFloch, and Murat [51]. Based on this, Parés [67] introduced path-conservative finite volume methods which are generalizations of the usual concept of conservative methods for conservation laws. Recently, Dumbser et al. [19] extended this concept to three-dimensional tetrahedral meshes using ADER schemes. The application to arbitrary meshes is to the authors knowledge not derived yet. Its development is beyond the scope of this thesis, but most of the implemented routines are components of a possible starting point.

Therefore, alternative methods have to be taken into account. One possibility is the application of finite difference methods (FDM) by identifying (4.6) as a Hamilton-Jacobi equation. Osher and Fedkiw [65] depicted the connection between these equa-

tions and conservation laws in an one-dimensional case as follows: the solution  $\varphi$  of a Hamilton-Jacobi equation equals the integral over a solution of a conservation law of the derivative  $\frac{\partial \varphi}{\partial x}$ . As a consequence, the solution can be found from the monotone, first-order accurate Godunov scheme and by splitting the treatment of convection and source term. In an one-dimensional case, the expression for the new time step reads [82]

$$\varphi_i^{(n+1)} = \varphi_i^{(n)} - \Delta t S_\varepsilon(\varphi_{i,0}) \cdot G(\varphi_i),$$

$$G(\varphi_i) = \begin{cases} \sqrt{\max(\max(D_x^- \varphi_i, 0)^2, \min(D_x^+ \varphi_i, 0)^2)} - 1 & \text{if } \varphi_i > 0 \\ \sqrt{\max(\min(D_x^- \varphi_i, 0)^2, \max(D_x^+ \varphi_i, 0)^2)} - 1 & \text{if } \varphi_i < 0 \end{cases}, \quad (4.8)$$

with the finite differences

$$D_x^- \varphi_i = \frac{\varphi_i - \varphi_{i-1}}{\Delta x},$$

$$D_x^+ \varphi_i = \frac{\varphi_{i+1} - \varphi_i}{\Delta x}. \quad (4.9)$$

In two- and three-dimensional simulations, this procedure can be extended to each dimension separately. Further, the order of accuracy can be increased using HJ-ENO or HJ-WENO schemes which are derived in the book of Osher and Fedkiw [65]. The discretisation (4.8) leads to a significant shift of the interface due to the violation of upwind differencing across the interface. Russo and Smereka [73] fixed this problem by calculating the differences near the interface strictly one-sided. The resulting modified version of (4.8) is given by

$$\varphi_i^{(n+1)} = \begin{cases} \varphi_i^{(n)} - \Delta t \cdot \left( S_\varepsilon(\varphi_{i,0}) |\varphi_i^{(n)}| - D(\varphi_0) \right) & \text{if } \varphi_{i,0} \varphi_{i+1,0} < 0, \\ & \text{or } \varphi_{i,0} \varphi_{i-1,0} < 0 \\ \varphi_i^{(n)} - \Delta t S_\varepsilon(\varphi_{i,0}) \cdot G(\varphi_i) & \text{else} \end{cases}, \quad (4.10)$$

with the definition of  $G(\varphi_i)$  from (4.8) and  $D(\varphi_{i,0})$  calculated as

$$D(\varphi_0) = \frac{2\varphi_{i,0}}{|\varphi_{i+1,0} - \varphi_{i-1,0}|}. \quad (4.11)$$

A similar but more sophisticated fix is derived by Colicchio, Landrini, and Chaplin [10] for simulations including high curvatures of the interface. Both methods improve the mass conservation significantly but the principal disadvantage as a finite difference discretisation is the assumption of uniform structured grids which excludes the application to complex problems. A circumvention for complex geometries could be the immersed boundary method of Peskin [69] which describes the boundaries by a Lagrangian formulation.

Alternatively, Edwards [20] proposes the idea of solving (4.6) as an one-dimensional problem in the direction normal to the interface on unstructured grids. Here, the polynomials of the implemented WENO reconstruction could be used for representing the derivatives (4.9) in the surroundings of the each cell. First numerical experiments show that the determination of the virtual boundary, where the derivatives are evaluated, is critical for arbitrary shaped cells and has to be investigated further. In addition, the necessary number of WENO reconstructions for one redistancing procedure would drastically reduce the overall efficiency of the method. A more promising approach is based on multidimensional splines with radial basis functions (RBF) which has been successfully coupled with ENO properties by Cecil, Qian, and Osher [7]. This unstructured finite difference method is more efficient than FVM-WENO reconstructions due to smaller stencils resulting in linear systems of equations and missing evaluations of volume integrals. RBF-FDM is part of recent developments in OpenFOAM [49] and could, thereby, benefit from the implemented stencil collection algorithm. The question of the prevention of interface shifts on unstructured grids is without reply so far.

The third possibility of redistancing the level set function is the fast marching method (FMM) of Sethian and Vladimirsky [77]. It is based on solving (4.1) as an Eikonal equation and connected to Huygen's principle since its construction involves expanding wavefronts [78, 104]. The main idea is to sweep the front ahead in an upwind manner starting from a narrow band around the interface where the level set remains a signed

distance function. In each loop, the nearest point to the actual front has to be searched iteratively and all neighbours of this point are then updated if they are on the downwind side. For more details and efficient algorithms on structured grids see Sethian's book [76]. Sethian and Vladimirskey [78] extended the original approach to unstructured grids but also stated that it is a cumbersome implementation for three-dimensional problems. Mut, Buscaglia, and Dari [57] developed a mass-conservative formulation of the FFM on tetrahedral grids in three dimensions which, however, is just conservative if the grid resolution is high. Further, the FFM has to be solved on the whole domain in each time step which is quite inefficient [20].

In conclusion, it has to be stated that all presented redistancing methods are not generally applicable in the given framework. By nature, methods based on finite differences are most efficient but unable to handle unstructured grids. In contrast, FVM struggles with the characteristics of the equation and efficiency. Therefore, it is decided to implement the simple FDM approach of Sussman with Russo's sub-cell fix (4.10) in order to verify the level set method at all. The method is then extended to unstructured grids by modifying the level set definition in chapter 5.

## 4.2 Solution of the Incompressible Navier-Stokes Equations

In the beginning of each time step, the material properties have to be determined irrespective of the choice of the solution algorithm. For this purpose, the level set  $\varphi$  is first advected using the advection equation (4.2) and afterwards redistanced by solving (4.8) to steady-state. The material properties are updated using the CSF model (see chapter 2.2 and (2.21)). In order to prevent numerical instabilities due to discontinuous density

and viscosity functions, the expression for  $\alpha$  is found from  $\varphi$  in accordance to

$$\alpha = \begin{cases} 0 & \text{if } \varphi < -\varepsilon \\ \frac{1}{2} \cdot (1 + \frac{\varphi}{\varepsilon} + \frac{1}{\pi} \sin(\frac{\pi\varphi}{\varepsilon})) & \text{if } |\varphi| \leq \varepsilon \\ 1 & \text{if } \varphi > \varepsilon \end{cases}, \quad (4.12)$$

resulting in an interface thickness of approximately  $2\varepsilon$  with  $\varepsilon$  an user defined value. A smoother representation of  $\alpha$  can be defined as [95]

$$\alpha = \frac{1}{2} \cdot \left( 1 + \tanh\left(\frac{\varphi}{2\varepsilon}\right) \right), \quad (4.13)$$

which is the selected computation in this thesis.

The resulting system of equations describing the incompressible Navier-Stokes equations for two-phase flows yields under consideration of chapter 2.1 and 2.2

$$\begin{aligned} \nabla \cdot \vec{u} &= 0, \\ \frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) &= -\nabla p + \nabla \cdot (2\mu \mathcal{D}) + \rho \vec{g} + \sigma \kappa \nabla \alpha. \end{aligned} \quad (4.14)$$

The main difficulty of the system is the missing pressure information in the continuity equation. No transport equation is available for obtaining the unknown pressure which arises the necessity of special solution algorithms for coupling pressure and velocity [93]. Furthermore, the non-linearity in the convective term has to be treated properly.

A simple class of solution algorithms is based on the projection method of Chorin [8]. Its theoretical background is the Helmholtz–Hodge decomposition stating that a velocity field  $\vec{u}$  can be decomposed in a divergence-free part  $\vec{u}_{df}$  and an irrotational part [9]

$$\vec{u} = \vec{u}_{df} + \nabla p, \quad (4.15)$$

with  $p$  a scalar function in general and denoting pressure here. Thus, the sought divergence-free velocity field  $\vec{u}^{(n+1)}$  of (4.14) is determined from the orthogonal projection of an

intermediate field  $\vec{u}^{(n+1/2)}$  on its divergence-free part as follows [9]

$$\vec{u}^{(n+1)} = \vec{u}^{(n+1/2)} - \frac{\Delta t}{\rho} \cdot \nabla p^{(n+1)}. \quad (4.16)$$

The intermediate velocity is the result of an explicit discretisation of the momentum equation without the pressure term which overcomes the non-linearity automatically. The pressure  $p^{(n+1)}$  is computed implicitly from a Poisson's equation which arises from (4.16) by applying the divergence operator and assuming  $\nabla \cdot \vec{u}^{(n+1)} = 0$

$$\nabla \cdot (\nabla p^{(n+1)}) = \frac{\rho}{\Delta t} \cdot (\nabla \cdot \vec{u}^{(n+1/2)}). \quad (4.17)$$

Both, the implementation of the projection method and the discretisation process is straightforward and can be applied on collocated, unstructured grids [35]. Amongst others, Vuorinen et al. [96] implemented it in OpenFOAM using a high-order Runge-Kutta time discretisation and showed improved efficiency in comparison to the available PISO algorithm for one-phase flows. Colicchio, Landrini, and Chaplin [10] developed an accurate two-phase flow solver on structured grids based on the projection method. The explicit calculation of the intermediate velocity could benefit from the implemented WENO schemes without applying it as a defered correction. However, it also restricts the time steps of the whole computation to the CFL-condition which is cumbersome for stiff, elliptic equations as in steady-state calculations.

Therefore, more implicit algorithms come to the fore such as Issa's [39] pressure-implicit with splitting operators (PISO) method handling the pressure-velocity coupling through several predictor and corrector steps. The PISO algorithm provides approximations to the exact solution of  $\vec{u}$  and  $p$  without the need of iterations. The accuracy of these solutions depend on the number of PISO loops and the chosen time stepping. The step size is not strongly restricted to the CFL-condition in comparison to the projection method. Each PISO loop consists of solving a pressure equation and correcting velocities explicitly. In addition, a momentum predictor step can be executed ahead as a first

guess of the velocity. The following description is oriented to the existing algorithm of Weller since his implementation has some improving modifications inside.

At first, the viscous stress term in (4.14) is reformulated because its numerical evaluation is more efficient if written as [72]

$$\nabla \cdot (2\mu \mathcal{D}) = \nabla \cdot (\mu(\nabla \vec{u} + (\nabla \vec{u})^T)) = \nabla \cdot (\mu \nabla \vec{u}) + (\nabla \vec{u}) \cdot \nabla \mu. \quad (4.18)$$

Further, the pressure is replaced by a modified pressure

$$p_{rgh} = p - \rho \vec{g} \cdot \vec{x}, \quad (4.19)$$

which is the pressure minus the hydrostatic pressure. This definition leads to a simpler specification of the pressure boundary conditions, since  $p_{rgh}$  is e.g. zero in the far fields of motionless flows [72]. Taking the gradient of (4.19) yields

$$\nabla p_{rgh} = \nabla p - \rho \vec{g} - (\vec{g} \cdot \vec{x}) \nabla \rho, \quad (4.20)$$

which can be inserted in (4.14). Under consideration of (4.18), the new formulation of the momentum equations reads as follows

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) = -\nabla p_{rgh} + \nabla \cdot (\mu \nabla \vec{u}) + (\nabla \vec{u}) \cdot \nabla \mu - (\vec{g} \cdot \vec{x}) \nabla \rho + \sigma \kappa \nabla \alpha. \quad (4.21)$$

## Momentum Predictor

For the predictor step, equation (4.21) is discretised in a semi-implicit manner as

$$\begin{aligned} \left[ \left[ \frac{\partial \rho [\vec{u}]}{\partial t} \right] \right] + \left[ \left[ \nabla \cdot (\rho \vec{u} \otimes [\vec{u}]) \right] \right] - \left[ \left[ \nabla \cdot (\mu \nabla [\vec{u}]) \right] \right] - (\nabla \vec{u}) \cdot \nabla \mu &= -\nabla p_{rgh} - (\vec{g} \cdot \vec{x}) \nabla \rho \\ &\quad + \sigma \kappa \nabla \alpha, \end{aligned} \quad (4.22)$$

where terms in double brackets are treated implicitly and the single bracket denotes the solution variable. The detailed discretisation of the terms using FVM is described in chapter 2.3. In this procedure, the key steps are the linearisation of the convection

term using (2.25) and the introduction of  $Q_l$  as the volumetric flow rate through the face. All explicit terms in (4.22) are reconstructed from face interpolated values in order to be consistent with the later correction step. The reconstruction methodology is described at the end of this section.  $Q_l$  is stored at the face centres and updated during the correction step in order to satisfy the continuity equation. In doing so, the checkerboard pressure oscillations are removed in the spirit of the Rhie and Chow interpolation since the velocities are obtained from reconstructed fluxes as a secondary variable [70, 72]. Further,  $Q_l$  and  $p$  are taken from the previous time step. The PISO algorithm possibly converges to an accurate solution without a predictor step for which reason it is often skipped on efficiency grounds.

## Solving a Pressure Equation

The first step of the actual PISO loop is solving a pressure equation. Its derivation starts from the semi-implicit discretised momentum equation (4.22) without the right-hand side which decouples pressure and velocity. Irrespective of the chosen schemes, the reduced equation can be generally written as an algebraic expression for each cell  $V_i$

$$a_i \vec{u}_i^{(n)} + \sum_j a_j \vec{u}_j^{(n)} = \vec{r}_i, \quad i = 1, \dots, N, \quad (4.23)$$

with  $j$  indicating the neighbouring cells of  $V_i$ ,  $a$  the interpolation coefficients and  $\vec{r}$  the source term. Assembling this representation to all cells of the domain produces an algebraic system of equations due to the dependency of  $\vec{u}_i$  on  $\vec{u}_j$ . It yields in matrix vector notation

$$\mathcal{A} \cdot \mathcal{U} = \mathcal{R}, \quad (4.24)$$

where  $\mathcal{A}$  is a  $N \times N$  scalar matrix with the coefficients  $a_i$  on the main diagonal and  $a_j$  as the off-diagonal elements.  $\mathcal{U}$  and  $\mathcal{R}$  are vectors of vectors of length  $N$  with each

line corresponding to the solution and source terms of one cell respectively. For a semi-implicit discretisation procedure,  $\mathcal{A}$  is split in the sum of a diagonal matrix  $\mathcal{A}_D$  and a matrix  $\mathcal{A}_{OD}$  containing the off-diagonal coefficients. The decomposition results in

$$\begin{aligned} (\mathcal{A}_D + \mathcal{A}_{OD}) \cdot \mathcal{U} &= \mathcal{R}, \\ \mathcal{A}_D \cdot \mathcal{U} &= \mathcal{R} - \mathcal{A}_{OD} \cdot \mathcal{U} = \mathcal{A}_H, \end{aligned} \quad (4.25)$$

with  $\mathcal{A}_H$  a matrix containing the off-diagonal elements of  $\mathcal{A}$ , the previous time step solutions from the transient term and explicit correction terms of high-order convection and diffusion term discretisations. In the following,  $\mathcal{A}_D$  is represented as a scalar field  $a_D$  which contains the main diagonal elements of  $\mathcal{A}$  and  $\mathcal{A}_H$  as a vector field  $\vec{a}_H$ .

Introducing this decomposition in (4.22) yields the following semi-discretised formulation of the momentum equation for the sought divergence-free velocity [70]

$$\vec{u} = \frac{1}{a_D} \cdot (\vec{a}_H - (\vec{g} \cdot \vec{x}) \nabla \rho + \sigma \kappa \nabla \alpha) - \frac{1}{a_D} \cdot \nabla p_{rgh}, \quad (4.26)$$

where the pressure gradient term is not discretised at this stage due to the Rhie and Chow procedure [41]. The first term on the right-hand side of (4.26), further denoted as  $\vec{u}^p$ , is similar to a velocity but does not satisfy the momentum equation due to the pressure term [70]. For the later calculation of the face fluxes, equation (4.26) is interpolated to the faces in accordance to

$$\vec{u}_l = \vec{u}_l^p - \left( \frac{1}{a_D} \right)_l \cdot (\nabla p_{rgh})_l. \quad (4.27)$$

A Poisson's equation for the pressure is now generated by inserting (4.26) in the continuity equation on the assumption that  $\vec{u}$  is divergence-free. The final expression reads as follows

$$\nabla \cdot \left( \frac{1}{a_D} \cdot \nabla [p_{rgh}] \right) = \nabla \cdot \vec{u}^p. \quad (4.28)$$

Equation (4.28) is discretised using the techniques of FVM resulting in

$$\sum_l^{L_i} \vec{n}_l |F_l| \cdot \left( \left( \frac{1}{a_D} \right)_l \cdot (\nabla [p_{rgh}])_l \right) = \sum_l^{L_i} Q_l^p, \quad (4.29)$$

with  $Q_l^p$  the volumetric flow rate through the face corresponding to  $\vec{u}_l^p$ . In accordance with (4.26), its computation is given by

$$Q_l^p = \left( \frac{1}{a_D} \right)_l \cdot \vec{n}_l \cdot (|F_l|(\vec{a}_H)_l - |F_l|(\vec{g} \cdot \vec{x})_l(\nabla p)_l + (\sigma \kappa)_l(\nabla \alpha)_l). \quad (4.30)$$

The pressure equation (4.28) is solved several times due to the non-orthogonal correction of diffusion terms (see (2.47)). In each loop, an algebraic systems of equations is solved using iterative algorithms as the preconditioned conjugate gradient (PCG) or the geometric-algebraic multi-grid (GAMG) methods. Their descriptions are not in the scope of this thesis, instead, it is referred to Saad's book [74] for derivations and applications of several solvers. The choice of appropriate methods is essentially since solving the pressure equation usually dictates the period of the whole time step.

## Explicit Velocity Correction

Once a converged solution for the pressure has been found, the volumetric flow rate can be updated in an explicit manner in order to satisfy the continuity equation. Under consideration of (4.27),  $Q_l$  reads

$$Q_l = Q_l^p - \vec{n}_l |F_l| \cdot \left( \left( \frac{1}{a_D} \right)_l \cdot (\nabla p_{rhs})_l \right). \quad (4.31)$$

The divergence-free velocities are finally reconstructed from  $Q_l$  using

$$\vec{u} = \left( \sum_{l=1}^{L_i} \vec{n}_l \otimes \vec{n}_l |F_l| \right)^{-1} \cdot \left( \sum_{l=1}^{L_i} \vec{n}_l Q_l \right). \quad (4.32)$$

In his thesis, Pringuey [70] derives this formula starting from the assumption

$$\sum_{l=1}^{L_i} \vec{n}_l \cdot (\vec{n}_l |F_l| \cdot \vec{u}) = \sum_{l=1}^{L_i} \vec{n}_l \cdot (\vec{n}_l |F_l| \cdot \vec{u}_l), \quad (4.33)$$

where the right-hand side is a face weighted averaging operator. The term on the left-hand side can be transformed as follows

$$\sum_{l=1}^{L_i} \vec{n}_l \cdot (\vec{n}_l |F_l| \cdot \vec{u}) = \sum_{l=1}^{L_i} (\vec{n}_l \otimes \vec{n}_l |F_l|) \cdot \vec{u}, \quad (4.34)$$

which corresponds to a mapping of  $\vec{u}$  onto a vector parallel to  $\vec{n}_l$  with length  $|\vec{n}_l| \cdot (\vec{n}_l |F_l| \cdot \vec{u}) \approx Q_l$ . Inserting  $Q_l = \vec{n}_l |F_l| \cdot \vec{u}$  in the right-hand side of (4.33) and rearranging the terms yields the reconstruction formula (4.32). The averaging operator preserves the second-order accuracy of the overall algorithm. If a complete high-order finite-volume method is sought, all Gaussian points of the flux evaluation should be included in this operation. The whole PISO loop is repeated by an user defined number at which Issa [39] proposes at least two loops. Each of them takes the velocity and pressure values from the previous step as initial conditions.

For the sake of completeness, in case of turbulent simulations the conservation equations have to be filtered temporally (URANS) or spatially (LES). Thereby, an additional term occurs which has to be modelled separately. In URANS, most models are based on the Boussinesq approach which replaces the arising Reynolds stresses by an expression depending on the turbulent viscosity  $v_t$  and turbulent kinetic energy  $k$ . In this thesis, the  $k$ - $\varepsilon$  model and Menter's  $k$ - $\omega$ -SST model [53] are applied for computing these unknowns. Therefore, two additional transport equations for  $k$  and  $\varepsilon$  and  $\omega$  respectively are solved after the final PISO loop. For further details on this topic, see e.g. the books of Versteeg and Malalasekera [93] and Ferziger and Perić [24].

### 4.3 Application and Limitations

The performance of the `lsFoam` solver is presented for the simulation of a collapsing water column. Before, the level set algorithm is verified without solving physics for the rotation of Zalesak's disk. All results confirm the described problems of mass conservation.

## Rotation of Zalesak's Disk

This case is introduced for the verification of the WENO interpolation scheme in chapter 3.3.2. The considered meshes are uniform structured due to the limitations of the redistancing algorithm. The number of cells quadruples for each refinement step resulting in grids with  $64 \times 64$ ,  $128 \times 128$  and  $256 \times 256$  squares. In comparison to the simulations in chapter 3.3.2, the smooth signed distance function is convected which should be less prone to numerical instabilities. The fourth-order accurate ADERUpwindFit scheme is applied together with Sussman's first-order accurate redistancing algorithm and Russo's sub-cell fix.

In order to initialise the signed distance function,  $\varphi$  is set to be  $+1$  in one fluid and  $-1$  in the other. This field converges to a proper function by solving the redistance equation several times as it can be seen in figure 4.1a.

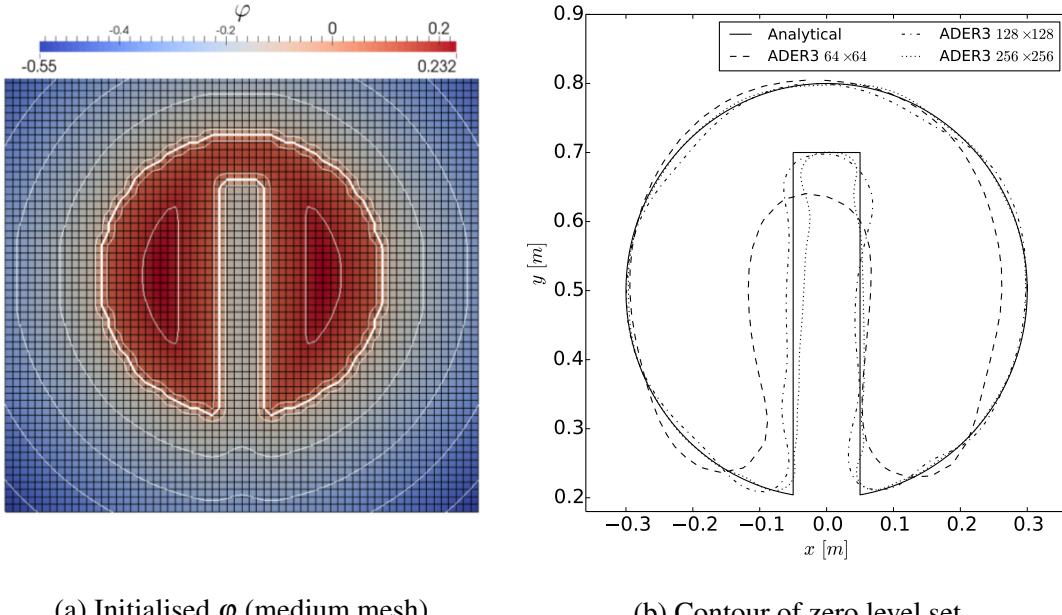


Figure 4.1: Results of the level set method for Zalesak's disk for three uniform structured meshes

The resulting contours of the zero level set after one revolution are shown in figure 4.1b. The accuracy is increasing with an increasing number of cells indicating the principal convergence of the solver. The results are similar to the WENO/Runge-Kutta contour on the finest mesh in figure 3.12a which correlates with less diffusion of the low order redistancing algorithm due to the high grid resolution. In contrary, this algorithm degrades the accuracy of the solution on coarse grids resulting in a deformed disk and a significant loss of area. The disk loses 0.9% of its area on the coarse grid and about 0.2% for the highest grid resolution. This shows the principle working of the sub-cell fix as long as the mesh is not too coarse.

## Collapse of a Water Column

As the actual verification case, the `lsFoam` solver is tested for the collapse of a two-dimensional water column designed by Yang [103]. As to be seen in figure 4.2, a squared box of length  $0.5\text{ m} \times 0.5\text{ m}$  filled with water stands in the left corner of a rectangular domain of length  $5\text{ m} \times 1\text{ m}$  as the initial position. At  $t = 0\text{ s}$ , the water column starts collapsing and a surge front moves in positive  $x$ -direction since all boundaries are fixed walls. The domain is decomposed in  $150 \times 30$  uniform structured cells which corresponds to a lower grid resolution than for Zalesak's disk.

The case is simulated as laminar, thus, no turbulence model is considered here. The transport properties are defined as water and air at room temperature which is  $v_l = 10^{-6}\frac{\text{m}^2}{\text{s}}$  and  $\rho_l = 1000\frac{\text{kg}}{\text{m}^3}$ . The discretisation schemes for the `lsFoam` solver are mainly based on WENO reconstructions and common second-order schemes. The fourth-order accurate `ADERUpwindFit` scheme is applied for the convection of  $\varphi$  while the convection term of the momentum equation is discretised using the fourth-order accurate `WENOUpwindFit` scheme. The gradients are evaluated with the described method of chapter 2.3.2 and central differencing since neither large gradients nor highly unstructured grids are considered. The time discretisation is computed using the second-order

accurate backward scheme. The  $Co$  number is fixed to 0.3. OpenFOAM's incompressible, two-phase flow solver `interFoam` based on a compressive VOF method is considered for comparison. The same discretisation schemes as for `lsFoam` are applied but with TVD convection instead of WENO schemes.

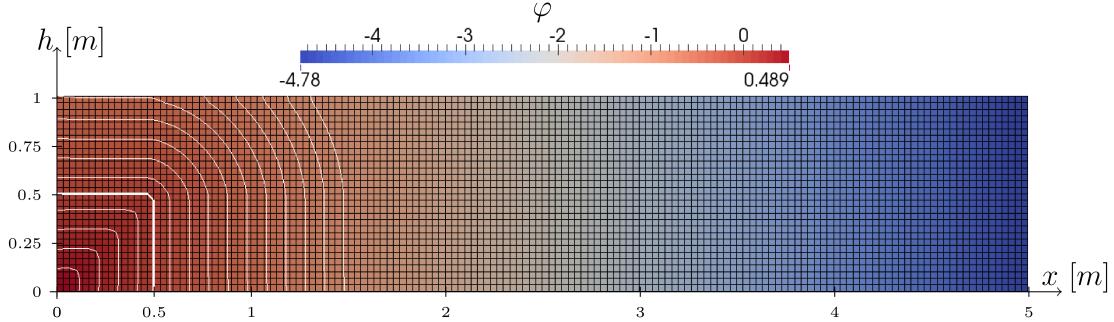
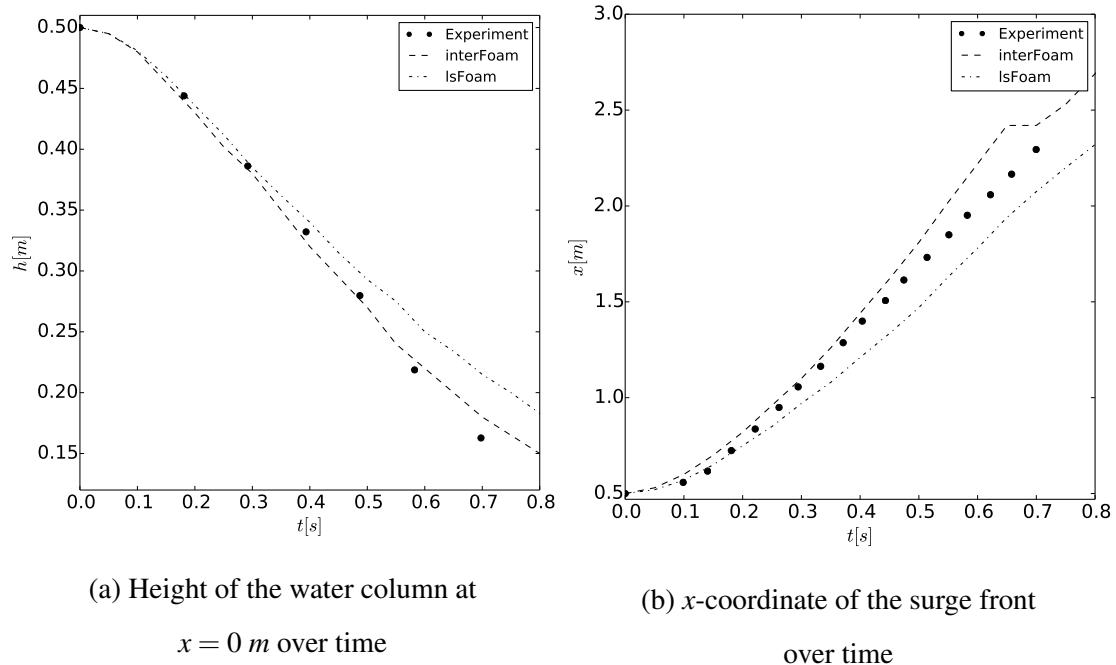


Figure 4.2: Setup for the collapse of a water column

The results are represented by the surge front as well as remaining water height at  $x = 0 \text{ m}$  over time and a comparison with the experimental data of Yang [103]. The height over time is presented in figure 4.3a. At the beginning of the simulation, both solvers confirm with the experiment. However, the level set solver diverges from the measured data increasingly resulting in a deviation of +30% at  $t = 0.7 \text{ s}$ . Here, `interFoam` deviates just +10%. The slower sag of the water correlates with a slower progression of the surge front as it can be seen in figure 4.3b. The deviation of `lsFoam` is about 9% at the same time which is satisfying if the poor grid resolution is taken into account. But, this resolution leads to a significant loss of mass of 5% which is unacceptable, particularly in comparison to the VOF method whose mass loss is less than 0.001%. A higher grid resolution would reduce the mass loss of the level set methods but the efficiency of the simulation as well. Therefore, an improvement of the mass conservation is aspired and leads to a new solver which is described in the next chapter.

Figure 4.3: Results for the collapse of a water column on  $150 \times 30$  squares

# **Chapter 5**

## **Development of a Conservative Two-Phase Flow Solver**

The application of the two-phase flow solver using level set methods reveals the known difficulties with mass conservation due to the non-conservative formulation of the redistancing equation and the inability to control the mass enclosed by the zero level set [81]. Irrespective of the conservation issue, the presented development is limited to uniform structured grids excluding most applications in the field of marine technology. Therefore, a new interface model based on a conservative level set method and OpenFOAM's limiting strategy MULES is derived in this chapter.

### **5.1 Principal Idea**

Most attempts to construct a conservative level set method (CLS) couple the standard level set method with another conservative interface model. Sussman and Puckett [81] developed the CLS-VOF method which applies the level set formulation to the accurate calculation of the interface properties, such as normal vectors and curvature, and a VOF method for a mass-conservative advection. Enright et al. [21] exchanged the VOF

method by a particle method whereby lagrangian mass-free marker particles rebuild the level set function in under-resolved regions. Although these methods serve their purpose, the computation is expensive since two complete algorithms have to be executed in each time step.

Therefore, the CLS method of Olsson and Kreiss [62] provides an advanced solution by recovering the simplicity of the original level set method. They propose to advect a smeared out Heaviside function, as it is given in (4.13), instead of the original signed distance function. Hence, the new level set function is 0 in one fluid and 1 in the other, whereas the transition region is smooth and contains the interface as 0.5 level set implicitly. The choice of the thickness of the transition region is a crucial factor and decides on the stability of the method. If the thickness tends to zero, the calculation of the interface normal vectors and curvature gets unstable and distorted surfaces may occur.

The interface area is conserved as long as monotone convection schemes are applied and the profile is accurately predicted [62]. In practice, even high-order TVD or WENO schemes can not provide this condition for which reason a reinitialisation step has to be performed after a specific number of advection steps. A compression flux in the normal direction of the interface is applied during the reinitialisation as well as an amount of diffusion to prevent the thickness from getting zero. The equation is solved to steady-state in artificial time which theoretically has the original profile of constant thickness as its solution. The complete building block of the CLS method can be expressed in the following form [70]

$$\begin{aligned} \frac{\partial \alpha}{\partial t} + \nabla \cdot (\vec{u} \alpha) &= 0, \\ \frac{\partial \alpha}{\partial \tau} + \nabla \cdot (\tilde{\vec{n}} \cdot (\alpha(1-\alpha) - \epsilon(\nabla \alpha \cdot \tilde{\vec{n}}))) &= 0, \end{aligned} \quad (5.1)$$

with  $\alpha$  the smeared Heaviside function and  $\tilde{\vec{n}}$  the interface normal vectors defined as

$$\begin{aligned}\alpha &= \frac{1}{2} \cdot \left( 1 + \tanh\left(\frac{\varphi}{2\varepsilon}\right) \right), \\ \tilde{\vec{n}} &= \frac{\nabla\alpha}{|\nabla\alpha|}.\end{aligned}\tag{5.2}$$

The reinitialisation equation in (5.1) can be characterised as a non-linear hyperbolic conservation law on the condition that the gradient is assumed to be constant as Pringuey [70] proposes. In doing so, he also provides a possible discretisation of (5.1) using FVM. The time steps  $\Delta t$  and  $\Delta\tau$  have to be chosen under consideration of the CFL-condition since explicit discretisation is usually applied.

The proper definition of the interface thickness  $\varepsilon$  is discussed in the paper of Olsson and Kreiss [62]. As already stated above, the method is very sensitive to this choice which is a priori and dependent on a characteristic cell size  $\Delta x$ . Pringuey [70] derives the criterion  $\varepsilon \geq 0.5\Delta x$  from an one-dimensional stability analysis. He further proposes to assume a single  $\Delta x$  for the whole domain on unstructured grids for reasons of safety. But, this would lead to smeared interfaces in case of typical external flows with high cell size ratios as ship flows.

In addition, Desjardins, Moureau, and Pitsch [17] suggested that the non-oscillatory behaviour of TVD and WENO schemes is strongly coupled to divergence-free velocity fields. This property is desired but not ensured in complex, turbulent simulations for which reason unbounded solutions for  $\alpha$  may occur. In order to reduce the dependency on  $\varepsilon$  and avoid spurious oscillations, they combined a standard level set method for calculating interface normal vectors accurately with the CLS method for conserving mass. This coupled method shows promising results but is less efficient. Further, a redistancing algorithm of the standard level set has to be provided. A similar approach was presented by Zhao et al. [108] who called this coupling improved conservative level set method (ICLS). Recently, Walker and Müller [97] analysed the convergence of the reinitialisation equation and concluded that the initial interface could deform with de-

creasing residual or even small residuals could not be reached at all. The proper number of advection steps before reinitialising as well as the steady-state criterion seem not to be general but have to be determined for each case separately, as own investigations confirm. Finally, the computation of the progress in time of the reinitialisation equation is quite inefficient due to the requirement of several WENO reconstructions per artificial time step.

This study of the standard CLS method indicates several disadvantageous which are tried to overcome in the following development of the new conservative level set method.

## 5.2 New Conservative Level Set Method

Olsson and Kreiss [62] stated that the mass conservation of their method is improved with decreasing profile thickness  $\varepsilon$ . Hence, there is no need of a preserved profile as long as an accurate calculation of  $\tilde{n}$  is ensured. For this purpose, a smoothed version of  $\alpha$  is considered instead of the introduction of coupled methods. This idea was first presented by Brackbill, Kothe, and Zemach [6] in conjunction with their CSF model. Rusche [72] proposes an elliptic relaxation equation for  $\alpha$  which is used here in a modified version. A general form of this equation can be written as

$$\left[ \nabla \cdot (\gamma_\alpha \nabla [\tilde{\alpha}]) \right] = [\tilde{\alpha}] - \alpha, \quad (5.3)$$

with  $\tilde{\alpha}$  the smoothed level set function. The first source term is treated explicitly since its slope is always positive and the diagonal dominance of the matrix would be reduced, as described in chapter 2.3.3. The determination of the diffusion coefficient  $\gamma_\alpha$  will be discussed in chapter 5.2.3. The updated formula for the calculation of the interface normal vectors is (compare (5.2))

$$\tilde{n} = \frac{\nabla \tilde{\alpha}}{|\nabla \tilde{\alpha}| + \frac{10^{-8}}{\Delta x}}. \quad (5.4)$$

The second term in the denominator stabilise the calculation in the far field of the interface and  $\Delta x$  is some characteristic cell size calculated as  $|V_i|^{1/3}$ .

The diffusion term of the reinitialisation equation becomes redundant now and the new expression reduces to

$$\frac{\partial \alpha}{\partial \tau} + \nabla \cdot (\tilde{\vec{n}} \cdot (\alpha(1-\alpha))) = 0, \quad (5.5)$$

with  $\alpha$  the advected Heaviside function. As stated above, the boundedness of the solution has to be ensured during solving (5.5) iteratively which could be provided using WENO reconstructions with the limiting strategy of chapter 3.3.1. Other mentioned difficulties, as the restriction to the CFL-condition and inefficiency of the method if WENO reconstructions are applied, are still valid. Therefore, OpenFOAM's multidimensional universal limiter for explicit corrected implicit solutions (MULES) is applied since it is not strongly restricted to  $Co$  numbers and provides a more efficient way of bounding the solution due to less WENO reconstructions per time step.

MULES evolves the level set under consideration of an advection flux of an implicit advection step using monotone, first-order accurate schemes and a limited portion of high-order advection and compression flux. It replaces the previous solution procedure of computing an advection step and compressing the resulting  $\alpha$ -field afterwards iteratively. The limiters ensure  $\alpha$  to be bounded by 0 and 1 for which reason distorted interfaces from excessive compression are mostly prevented. In analogy to the described PISO algorithm, which results in divergence-free volumetric flow rates  $Q_l$ , MULES is applied to the flux of  $\alpha$  instead to  $\alpha$  itself. In analogy to  $Q_l$ , this flux is defined as

$$Q_{\alpha,l} = \vec{u}_l \vec{n}_l |F_l| \alpha_l = Q_l \alpha_l. \quad (5.6)$$

Its limited version  $\hat{Q}_{\alpha,l}$  reads

$$\hat{Q}_{\alpha,l} = Q_{\alpha,l}^{(UD)} + \hat{Q}_{\alpha,l}^{(Corr)}, \quad (5.7)$$

with  $Q_{\alpha,l}^{(UD)}$  the advection flux of an implicit calculation using an upwind discretisation in space and the implicit Euler scheme for the temporal discretisation. The limited high-order correction flux  $\hat{Q}_{\alpha,l}^{(Corr)}$  is given by

$$\hat{Q}_{\alpha,l}^{(Corr)} = \theta_l \cdot Q_{\alpha,l}^{(Corr)} = \theta_l \cdot \left( Q_{\alpha,l}^{(HO)} + Q_{\alpha,l}^{(C)} - Q_{\alpha,l}^{(UD)} \right). \quad (5.8)$$

Here,  $Q_{\alpha,l}^{(HO)}$  denotes the high-order convection flux and  $Q_{\alpha,l}^{(C)}$  the compression flux. After the limiter  $\theta_l$  is determined, the level set of the new time step  $n+1$  is calculated from the implicitly advected value  $\alpha^{(n+1/2)}$  and the limited correction flux using an explicit Euler scheme as

$$\alpha_i^{(n+1)} = \alpha_i^{(n+1/2)} - \frac{\Delta t}{|V_i|} \sum_{l=1}^{L_i} \hat{Q}_{\alpha,l}^{(Corr)}, \quad i = 1, \dots, N. \quad (5.9)$$

As a consequence,  $\alpha$  is obtained in a semi-implicit manner for which reason  $\Delta t$  is not strongly restricted but also not arbitrary. The calculation of  $\theta_l$  is presented in detail in chapter 5.2.1, whereas the compression term is discussed in chapter 5.2.2.

After the limited fluxes and  $\alpha$  are computed from (5.7) and (5.9), the interface and material properties can be calculated using the CSF model and (2.21). The presented PISO algorithm from chapter 4.2 is still applicable for solving the Navier-Stokes equations. As mentioned above, the continuity equation is mainly satisfied by  $Q_l$  for which reason the density in the convective term of the momentum equations should be determined at the faces and not interpolated from cell centres. For this purpose, the volumetric flow rate is replaced by the mass flow rate  $Q_{l,\rho}$ . In accordance with the definition of material properties in (2.21), it is calculated as a dependency on  $\alpha$  which leads to

$$Q_{l,\rho} = \hat{Q}_{\alpha,l}\rho_1 + (Q_l - \hat{Q}_{\alpha,l}) \cdot \rho_2. \quad (5.10)$$

### 5.2.1 Calculation of the Limiter Factors

The development of MULES was advanced by Weller who never published an official paper regarding his idea. Therefore, the following description is based on the code of

his semi-implicit implementation [63] and a description of Pringuey [70].

The aim of the limiting strategy is to guarantee a bounded  $\alpha$ -field in the interval  $[0, 1]$  resulting in physical density values. For this purpose, the  $\alpha$ -values of time step  $n + 1$  should not exceed the extreme values of the neighbouring cells of the intermediate time step  $n + 1/2$  which can be written as

$$\min_{j_L} (\alpha_{j_L}^{(n+1/2)}) \leq \alpha_i^{(n+1)} \leq \max_{j_L} (\alpha_{j_L}^{(n+1/2)}), \quad j_L = 1, \dots, L_i. \quad (5.11)$$

If (5.11) holds, the solution is strictly bounded since the intermediate results are obtained by applying monotone schemes. In terms of fluxes, the bounds in (5.11) can be expressed by the maximum temporal variation of correcting  $\alpha$  in  $V_i$  [70]. It corresponds to the maximum increase and decrease of level set through the maximum inflow and outflow correction fluxes respectively which yields

$$\left( \frac{\Delta\alpha}{\Delta t} \right)_i^{out,max} \leq \hat{Q}_{\alpha,l}^{(Corr)} \leq \left( \frac{\Delta\alpha}{\Delta t} \right)_i^{in,max}, \quad (5.12)$$

with the bounds defined as

$$\begin{aligned} \left( \frac{\Delta\alpha}{\Delta t} \right)_i^{out,max} &= \frac{|V_i|}{\Delta t} \cdot (\alpha_i^{(n+1/2)} - \alpha_{i,min}^{(n+1/2)}), \\ \left( \frac{\Delta\alpha}{\Delta t} \right)_i^{in,max} &= \frac{|V_i|}{\Delta t} \cdot (\alpha_{i,max}^{(n+1/2)} - \alpha_i^{(n+1/2)}), \end{aligned} \quad (5.13)$$

and the extreme values given by

$$\begin{aligned} \alpha_{i,min}^{(n+1/2)} &= \max \left( \min_{j_L} (\alpha_{j_L}^{(n+1/2)}), 0 \right), \\ \alpha_{i,max}^{(n+1/2)} &= \min \left( \max_{j_L} (\alpha_{j_L}^{(n+1/2)}), 1 \right), \quad j_L = 1, \dots, L_i. \end{aligned} \quad (5.14)$$

Inserting (5.13) in (5.9) shows that this definition results in the required limits in (5.11). Hence, the sought  $\theta_l$  are found from the condition (5.13) for the correction fluxes. The procedure is executed for each face separately and in an iterative manner over  $k$  loops. At the beginning, the limiter is set to be  $\lambda^{(0)} = 1$  and in each further iteration, it is reduced gradually corresponding to reducing the influence of the high-order fluxes.

Weller searches for an averaged limiter for all faces of a cell supporting an outflow correction flux  $\theta_i^{out}$  and another one for the remaining faces  $\theta_i^{in}$  [70]. In order to obtain conditional equations for these limiters, the same splitting is obtained for the fluxes by introducing an unlimited correction flux operator  $Q_{\alpha,i}^{(Corr)}$  for each cell which equals the sum of inflow and outflow fluxes

$$Q_{\alpha,i}^{(Corr)} = \sum_l Q_{\alpha,l}^{(Corr)} = \sum_{l,out} |Q_{\alpha,l}^{(Corr)}| - \sum_{l,in} |Q_{\alpha,l}^{(Corr)}|, \quad l = 1, \dots, L_i. \quad (5.15)$$

The splitting has to be executed once before the first iteration starts. The same procedure is used for the definition of a limited correction flux operator  $\hat{Q}_{\alpha,i}^{(Corr)}$  for each cell which yields in the  $k^{th}$  iteration

$$\hat{Q}_{\alpha,i}^{(Corr),(k)} = \sum_{l,out} (\theta_l)^{(k)} |Q_{\alpha,l}^{(Corr)}| - \sum_{l,in} (\theta_l)^{(k)} |\hat{Q}_{\alpha,l}^{(Corr)}|, \quad l = 1, \dots, L_i. \quad (5.16)$$

Under consideration of (5.15) and (5.16), the bounds in (5.12) can be replaced by sums of limited inflow and outflow fluxes

$$\begin{aligned} \left( \frac{\Delta\alpha}{\Delta t} \right)_i^{out,max} &= (\theta_i^{out})^{(k)} \sum_{l,out} |Q_{\alpha,l}^{(Corr)}| - \sum_{l,in} (\theta_l)^{(k-1)} |Q_{\alpha,l}^{(Corr)}|, \\ \left( \frac{\Delta\alpha}{\Delta t} \right)_i^{in,max} &= (\theta_i^{in})^{(k)} \sum_{l,in} |Q_{\alpha,l}^{(Corr)}| - \sum_{l,out} (\theta_l)^{(k-1)} |Q_{\alpha,l}^{(Corr)}|. \end{aligned} \quad (5.17)$$

Both equations are decoupled by combining a limited version of (5.15) with (5.16). Thus, the averaged limiters can be computed since the terms on the left hand side are known from (5.13). Rearranging (5.17) and introducing additional, physical bounds yields

$$\begin{aligned} (\theta_i^{out})^{(k)} &= \max \left[ \min \left( \frac{\left( \frac{\Delta\alpha}{\Delta t} \right)_i^{out,max} + \sum_{l,in} (\theta_l)^{(k-1)} |Q_{\alpha,l}^{(Corr)}|}{\sum_{l,out} |Q_{\alpha,l}^{(Corr)}| + \varepsilon}, 1 \right), 0 \right], \\ (\theta_i^{in})^{(k)} &= \max \left[ \min \left( \frac{\left( \frac{\Delta\alpha}{\Delta t} \right)_i^{in,max} + \sum_{l,out} (\theta_l)^{(k-1)} |Q_{\alpha,l}^{(Corr)}|}{\sum_{l,in} |Q_{\alpha,l}^{(Corr)}| + \varepsilon}, 1 \right), 0 \right]. \end{aligned} \quad (5.18)$$

As a result, two averaged limiters from the adjacent cells are available at each face. The final limiter at a face is evaluated as a dependency on the direction of the correction flux

$$\theta_l^{(k)} = \begin{cases} \min\left((\theta_l)^{(k-1)}, (\theta_P^{out})^{(k)}, (\theta_N^{in})^{(k)}\right) & \text{if } Q_{\alpha,l}^{(Corr)} > 0 \\ \min\left((\theta_l)^{(k-1)}, (\theta_P^{in})^{(k)}, (\theta_N^{out})^{(k)}\right) & \text{if } Q_{\alpha,l}^{(Corr)} < 0 \end{cases}, \quad (5.19)$$

where index  $P$  denotes the owner and  $N$  the neighbouring cell. The number of iterations is user specified but should be greater than one due to the averaging in (5.17).

### 5.2.2 Derivation of the Compression Fluxes

The correction fluxes in MULES contain a high-order convection and compression flux. The convection flux is calculated using any high-order scheme as WENOUpwindFit or TVD schemes. The compression flux has to be treated differently due to its non-linear formulation. In the CLS method of Olsson and Kreiss [62], the compression is defined in the normal direction of the interface and independent of the  $Co$  number. In contrast, Weller's algorithm introduces a compression velocity  $\vec{u}_{r,l}$  which also acts in the direction normal to the interface but with a local, velocity scaled magnitude. It is given by

$$\vec{u}_{r,l} = \left| \frac{Q_l}{|F_l|} \right| \cdot \tilde{\vec{n}}_l, \quad (5.20)$$

with  $Q_l$  the volumetric flow rate defined in (2.25). In view of this definition, the standard CLS method applies the compression velocity  $\vec{u}_{r,l} = \tilde{\vec{n}}_l$  with magnitude one. The additional factor leads to an adaptation to different flow regions. It increases the compression at faces of high velocity and therefore, counteracts the higher numerical diffusion of the advection step to a certain extent. Further, it prevents compression of static interfaces where no diffusion occurs since  $Q_l = 0$ . These advantages are the reason for taking (5.20) into account in this thesis.

Under consideration of the definition of the fluxes in (5.6), the face-integrated com-

pression fluxes  $Q_{\alpha,l}^{(C)}$  can be written as

$$Q_{\alpha,l}^{(C)} = \int_{F_l} \vec{n}_l \cdot \vec{u}_{r,l}(\alpha(1-\alpha)) \, dF_l = \int_{F_l} \tilde{Q}_{\alpha,l}^{(C)} \, dF_l. \quad (5.21)$$

The evaluation of  $\tilde{Q}_{\alpha,l}^{(C)}$  arises from the spatial discretisation of the compression equation which yields in accordance to (5.5)

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\vec{u}_{r,l} \cdot (\alpha(1-\alpha))) = 0. \quad (5.22)$$

The equation is characterised as a non-linear hyperbolic conservation law for which reason the fluxes are obtained by solving the underlying Riemann problem. The values of  $\tilde{Q}_{\alpha,l}^{(C)}$  have to be computed at each Gaussian point of the face separately due to the non-linearity in contrast to the convection fluxes of linear conservation laws (see chapter 2.3.1 and 3.3). Therefore, equation (5.21) gets for each face using the same triangulation as for WENO reconstructions

$$\int_{F_l} \tilde{Q}_{\alpha,l}^{(C)} \, dF_l = \sum_{t=1}^{N_t} \sum_{k=1}^{N_k} w_k |F_{l,k}| \cdot \tilde{Q}_{\alpha,l,k}^{(C)}(\alpha^\pm), \quad (5.23)$$

with  $N_t$  the number of triangles and  $N_k$  the number of Gaussian points for each of them. This procedure is time-consuming even though the coordinates and weights can be pre-computed. As an alternative, Weller's second-order accurate algorithm linearises the compression term and thus, computes two convection fluxes consecutively using any common discretisation scheme.

If higher order of accuracy is sought, a local Riemann problem has to be solved at each Gaussian point. For this purpose, the compression equation (5.22) is evaluated in the direction normal to the face due to its rotational invariance resulting in the one-dimensional equation

$$\frac{\partial \alpha}{\partial t} + \frac{\partial Q_k}{\partial n_l} = 0, \quad (5.24)$$

with  $Q_k$  denoting  $\tilde{Q}_{\alpha,l,k}^{(C)}(\alpha^\pm)$  for the sake of clarity. Under consideration of the characteristic speed

$$\lambda(\alpha) = \frac{\partial Q_k}{\partial \alpha} = \vec{n}_l \vec{u}_{r,l} \cdot (1 - 2\alpha), \quad (5.25)$$

equation (5.24) results in the following Riemann problem

$$\left. \begin{array}{ll} \text{PDE :} & \frac{\partial \alpha}{\partial t} + \lambda(\alpha) \frac{\partial \alpha}{\partial n_l} = 0, \\ \text{IC :} & \alpha(n_l, 0) = \alpha_0(n_l) = \begin{cases} \alpha^- & \text{if } n_l < 0 \\ \alpha^+ & \text{if } n_l > 0 \end{cases} \end{array} \right\}. \quad (5.26)$$

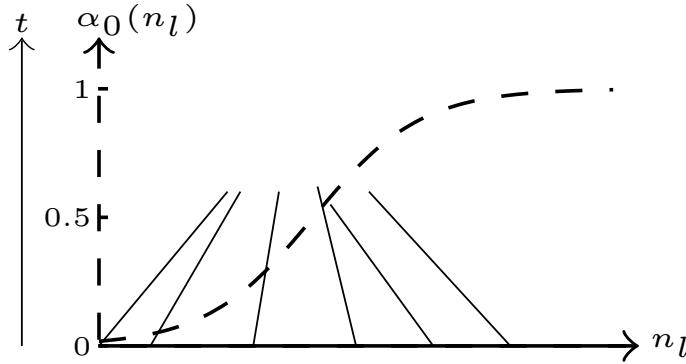


Figure 5.1: Compression effect of the  $\alpha$ -profile (dashed line). Characteristics on  $n_l$ - $t$  plane are represented as thin black lines

The initial values  $\alpha^\pm$  are the cell averaged values in case of a first-order accurate Godunov scheme or polynomials, evaluated at the considered Gaussian point, in case of WENO reconstructions. In comparison to the linear case,  $\lambda$  is solution depended for which reason the initial data is translated with distortions [90]. A typical smeared profile of  $\alpha$  is shown in figure 5.1 together with the characteristics carrying the constant initial values  $\alpha_0$ . The compression effect occurs due to the different signs of the characteristic's slopes which equal  $\lambda$ . In case of  $\alpha_0 < 0.5$ , the characteristic speed is greater than

zero and increases with decreasing  $\alpha_0$ . Hence, smaller initial data is transported faster than larger one. In case of  $\alpha_0 > 0.5$ ,  $\lambda$  gets negative and increases with increasing  $\alpha_0$  resulting in a transport in negative  $n_l$ -direction. At the interface,  $\alpha_0 = 0.5$ , the speed equals zero and no flux is transported resulting in steepening of the profile at this point.

As already indicated in chapter 4.1, the exact solution of the Riemann problem (5.26) can be found under consideration of the Rankine Hugoniot condition. It yields an algebraic expression which relates the jump of the fluxes to the jump of  $\alpha$  at the discontinuity

$$Q_k^- - Q_k^+ = S \cdot (\alpha^+ - \alpha^-), \quad (5.27)$$

with  $S$  the speed of the discontinuity [90]. Here, it can be calculated from (5.27) as

$$S = \frac{\Delta Q_k}{\Delta \alpha} = \vec{n}_l \vec{u}_{r,l} \cdot (1 - (\alpha^- + \alpha^+)). \quad (5.28)$$

Moreover, solutions of non-linear hyperbolic equations are not physical in any case which finally leads to the following constraint, also known as entropy condition [90]

$$\lambda(\alpha^-) > S > \lambda(\alpha^+). \quad (5.29)$$

Taking this condition into account, the exact solution of (5.26) can now be determined as a dependency of the signs of  $\alpha^\pm$ . If  $\alpha^- > \alpha^+$ , the solution develops a shock wave and  $\alpha^- < \alpha^+$  leads to a rarefaction wave. For more details on the different wave types and possible unstable solutions, see the book of Toro [90]. The final solution reads as follows

$$\begin{aligned} \text{if } \alpha^- > \alpha^+ : \alpha(n_l, t) &= \begin{cases} \alpha^- & \text{if } n_l - St < 0 \\ \alpha^+ & \text{if } n_l - St > 0 \end{cases}, \\ \text{if } \alpha^- > \alpha^+ : \begin{cases} \alpha(n_l, t) = \alpha^- & \text{if } \frac{n_l}{t} \leq \lambda(\alpha^-) \\ \lambda(\alpha) = \frac{n_l}{t} & \text{if } \lambda(\alpha^-) < \frac{n_l}{t} < \lambda(\alpha^+) \\ \alpha(n_l, t) = \alpha^+ & \text{if } \frac{n_l}{t} \geq \lambda(\alpha^+) \end{cases} &. \end{aligned} \quad (5.30)$$

The flux across the face at any Gaussian point is evaluated from the Riemann solver (5.30) by inserting  $n_l = 0$  and assuming  $t > 0$  [70]

$$\begin{aligned} \text{if } \alpha^- > \alpha^+ : Q_k &= \begin{cases} Q_k^- & \text{if } S > 0 \\ Q_k^+ & \text{if } S < 0 \end{cases}, \\ \text{if } \alpha^- > \alpha^+ : Q_k &= \begin{cases} Q_k^- & \text{if } 0 \leq \lambda(\alpha^-) \\ \vec{n}_l \vec{u}_{r,l} \cdot \frac{1}{4} & \text{if } \lambda(\alpha^-) < 0 < \lambda(\alpha^+) \\ Q_k^+ & \text{if } 0 \geq \lambda(\alpha^+) \end{cases}. \end{aligned} \quad (5.31)$$

Once the solution of  $Q_k$  is obtained, the overall flux of each face can be summed up using (5.23). The final non-linear Riemann solver is implemented as a generally applicable interpolation scheme in OpenFOAM, called `WENOCompressionFit`.

### 5.2.3 Definition of the Diffusion Coefficients

The main advantage of the new interface model is the avoidance of iterative reinitialisation algorithms by applying MULES and a smoothed  $\alpha$  from (5.3) for the determination of the interface normal vectors. The modified  $\alpha$  is calculated from an elliptic relaxation equation using the implicit advected level set, so an intermediate, smeared interface. The necessity of the relaxation arises from the study of compressive VOF methods where no mollified  $\alpha$  is considered. In several applications, an improper compression of the interface leads to distorted interfaces and jumps in the density function which increase with the density ratio of the fluids. In the PISO algorithm, the discontinuity of the density is transferred to the fluxes and often leads to divergence of the solution of the pressure equation, in particular at high  $Co$  numbers. The causing situations are near cells characterised by high velocities and small sizes corresponding to high compression but little numerical diffusion. Numerical experiments show that this effect vanishes if a proper calculation of the interface normal vectors is ensured. In this case, the direction of compression is precisely computed and spurious oscillation of

the interface are prevented. The success of the relaxation (5.3) depends on the proper determination of the diffusion coefficients  $\gamma_\alpha$ . Rusche [72] proposes a quadratic dependency on the cell size in conjunction with a free parameter. In contrast, a more physical reliable approach is used in this thesis.

It is derived from the assumption that a known, finite amount of diffusion  $\gamma_{t,l}$  exists for each face which ensures a proper calculation of the interface normal vectors. This value equals the sum of diffusive influences on  $\alpha$  which are the numerical convection and temporal diffusion  $\gamma_{n,l}$  and the additional relaxation using  $\gamma_{\alpha,l}$

$$\gamma_{t,l} = \gamma_{\alpha,l} + \gamma_{n,l}. \quad (5.32)$$

A possible determination of the numerical diffusion on unstructured grids using FVM is provided by Jasak [41]. He obtains numerical diffusion tensors from an evaluation of the difference between the considered convection and the less diffusive central differencing scheme. It is, however, advisable to define  $\gamma_{\alpha,l}$  as a scalar due to stability reasons. Hence,  $\gamma_{n,l}$  can not be determined in an efficient way on arbitrary grids and is, therefore, neglected here.

The total amount of diffusion  $\gamma_{t,l}$  is calculated from the  $Pe$  number since it relates the relaxation to the velocity and grid size. This dimensionless number expresses the ratio of convective to diffusive fluxes and is written as

$$Pe_l = \frac{|Q_l| \vec{n}_l \vec{d}_l}{|F_l| \Gamma_l}, \quad (5.33)$$

at the face. Taking the neglect of the numerical diffusion into account, the transport of  $\alpha$  in the advection step is pure convective and the diffusion coefficient  $\Gamma_l$  in (5.33) can be identified as the sought  $\gamma_{t,l}$  and  $\gamma_{\alpha,l}$  respectively. Hence, the relaxation coefficients are determined from (5.33) after a rearrangement

$$\gamma_{\alpha,l} = \frac{|Q_l| \vec{n}_l \vec{d}_l}{|F_l| \cdot Pe}, \quad (5.34)$$

with  $Q_l$  from the previous time step and  $Pe$  as a free parameter here. Practical investigations show that the restriction of  $Pe$  on the central differencing scheme from chapter 2.3.1 fits quite well for which reason  $Pe = 2$  is the default setting. Higher numbers can also be adequate which is mainly caused by the neglected numerical diffusion.

A secondary effect of choosing  $\gamma_{\alpha,l}$  from (5.34) is the interaction with the definition of the compression flux since both parameters depend on the local normal velocity magnitude. Both modifications are inactive near motionless interfaces and thus, no information is lost by applying the described smoothing.

### 5.3 Application to the Rotation of Zalesak's Disk

The final solver, called `clsMULESFoam`, is the combination of a new CLS method and MULES limiting strategy. It might be noticed that the presented method can also be interpret as a modified version of OpenFOAM's compressive VOF method due to the same definition of level set and volume fraction respectively. However, the addition of the relaxation effects the prevention of spurious oscillations in many applications as it will be shown in chapter 6.

Before, the new interface model, called `clsMULES`, is verified for the rotation of Zalesak's disk as described in chapter 3.3.2. The accuracy of the disk's contour and distribution at the slice  $y = 0.6 \text{ m}$  is checked for the `clsMULES` model using the fourth-order accurate `WENOUpwindFit` convection and `WENOCompressionFit` compression scheme. OpenFOAM's VOF method based on MULES in conjunction with the TVD-vanLeer scheme is taken as a reference here.

As it can be seen in the resulting figures 5.2 - 5.5, the reference method predicts oscillatory and asymmetric contours. In comparison, `clsMULES` flattens these effects successfully. The smoothing expectedly leads to a loss of accuracy at the corners of the contour. This drawback might be acceptable since most applications lack in such sharp

transitions. A possibly problematic case could be breaking waves which are, however, not in the scope of this thesis. The distribution of the profiles at the slice  $y = 0.6 \text{ m}$  look similar for both methods. On unstructured grids, the `clsMULES` method benefits from the superior WENO reconstructions in reference spaces. Further, the principal working of MULES can be confirmed since all fields are bounded. The compression term leads to broader distributions at the extrema which predict the analytical profiles more accurate than using simple advection as it was shown in the figures 3.11b - 3.14b.

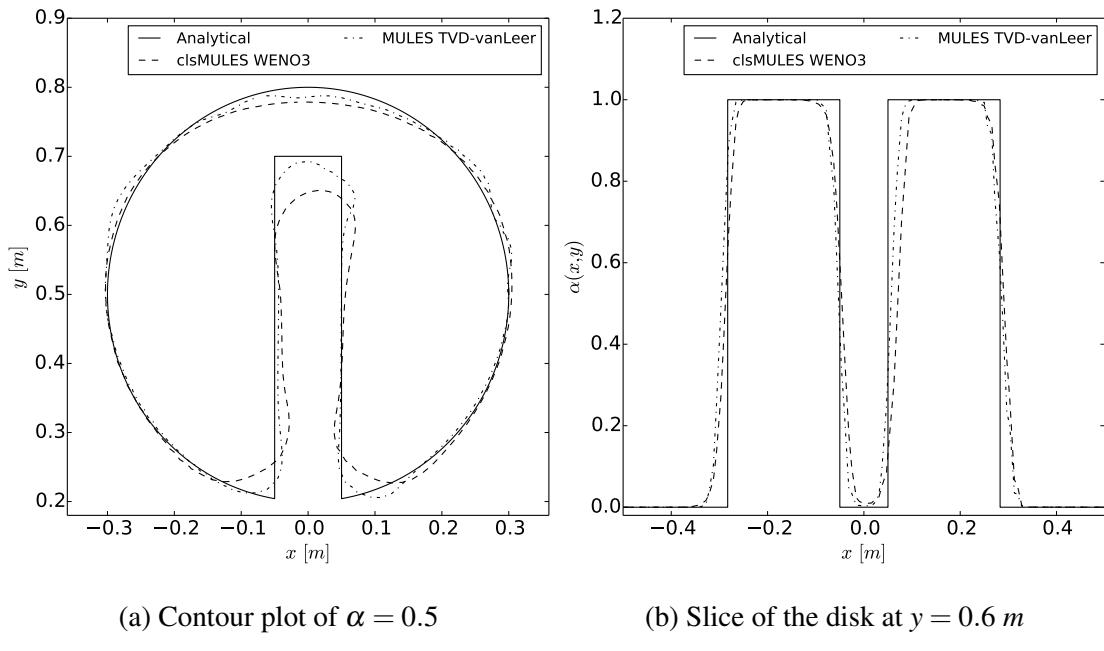


Figure 5.2: Results of the interface models for Zalesak's disk on  $128 \times 128$  squares

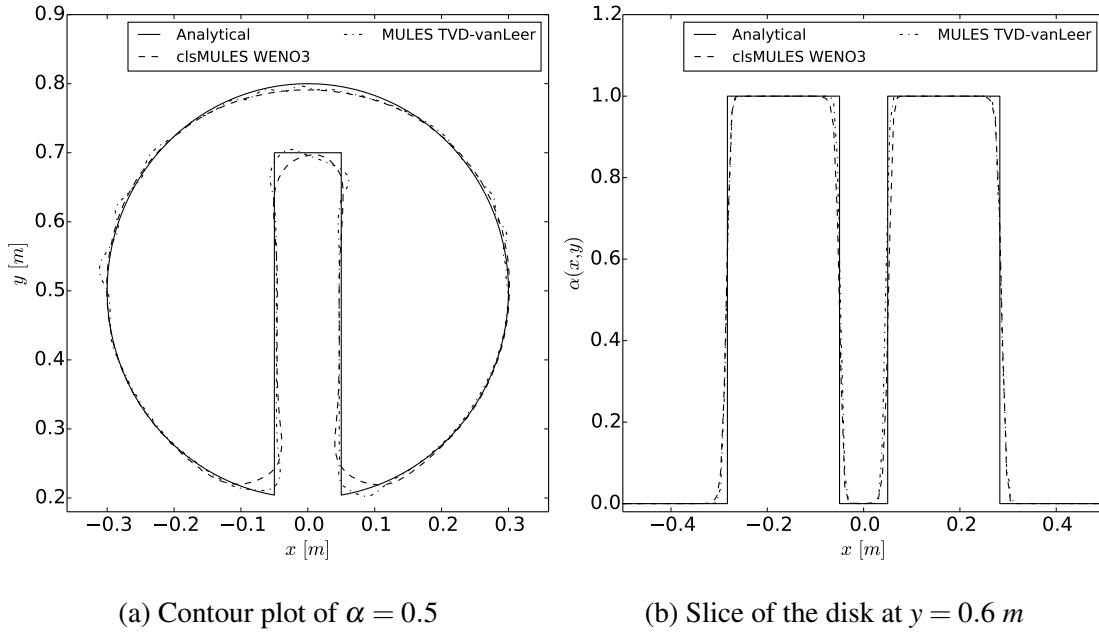


Figure 5.3: Results of the interface models for Zalesak's disk on  $256 \times 256$  squares

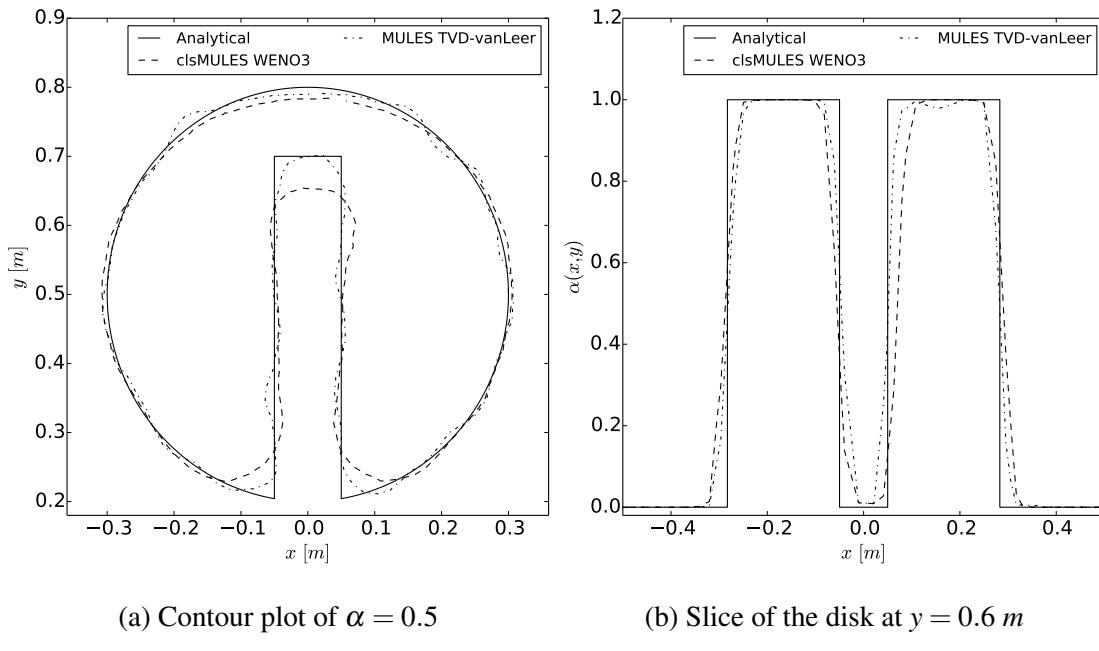
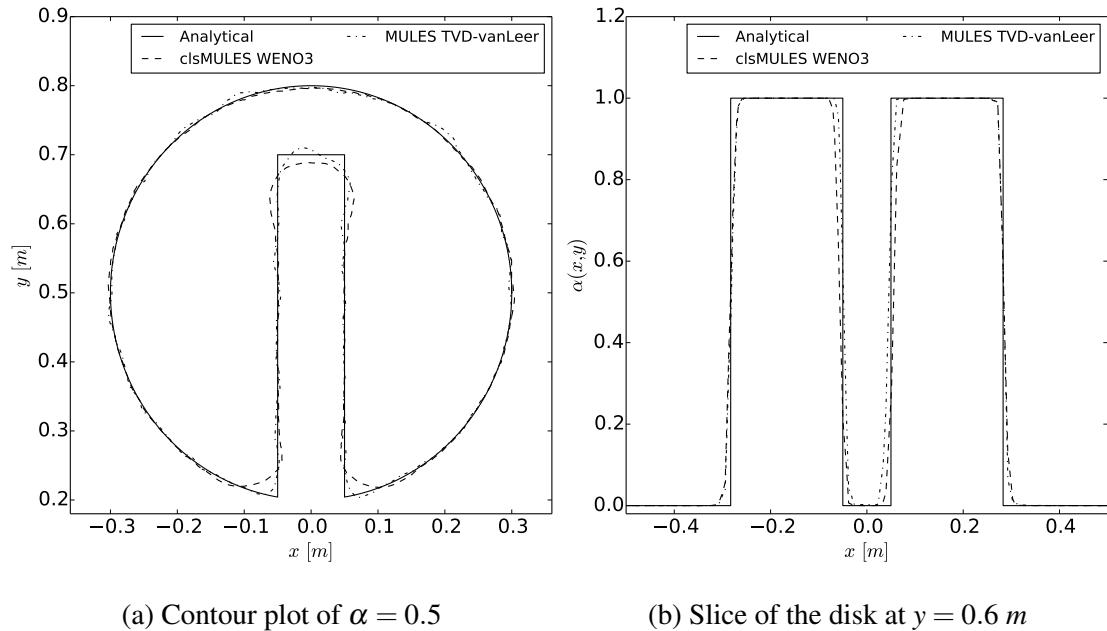


Figure 5.4: Results for Zalesak's disk on  $128 \times 128$  triangles

(a) Contour plot of  $\alpha = 0.5$ (b) Slice of the disk at  $y = 0.6 \text{ m}$ Figure 5.5: Results for Zalesak's disk on  $256 \times 256$  triangles

# Chapter 6

## Physical Applications and Performance

The application of the `clsMULESFoam` solver to several test cases is presented in this chapter. It includes structured and unstructured grids in two and three dimensions. The meshes are generated with the objective of producing oscillatory results for the standard VOF method in order to show the superior behaviour of the new solver. All cases are computed using similar schemes and system solvers in order to ensure comparability. The discretisation schemes are listed in table 6.1. The verification is completed by an evaluation of the performance of `clsMULESFoam` and WENO schemes in chapter 6.5.

All provided cases including meshes, initial conditions and other settings can be found on the enclosed disc.

Table 6.1: Overview of applied discretisation schemes

Term	<code>clsMULESFoam</code>	<code>interFoam</code>
temporal	quadratic backward	quadratic backward
gradients	<code>WENOGrad 3</code>	Gauss theorem and linear interpolation
convection	<code>WENOUpwindFit 3 0</code>	<code>TVD-vanLeer</code> and <code>TVD-limitedLinear</code>
compression	<code>WENOCompressionFit 3</code>	blend of linear and upwind
diffusion	linear with non-orthogonal correction	linear with non-orthogonal correction

## 6.1 Motion of Two Superposed Viscous Fluids

The first case is proposed by Damián [13] to compare different temporal discretisation schemes for the application of the `interFoam` solver. It consists of two fluids with the same viscosity which are superposed in a tank with slip walls similar to sloshing problems. The initial position of the free surface is shown in figure 6.1b and expressed by

$$\zeta(x) = 1.5 \text{ m} + \zeta_0 \cdot \sin\left(\pi \cdot \left(\frac{1}{2} - x\right)\right), \quad (6.1)$$

with  $\zeta_0 = 0.1 \text{ m}$  the initial amplitude of the perturbation. The analytical solution for the linearised problem was derived by Prosperetti [71] and yields

$$\begin{aligned} \zeta(t) = & \frac{4 \cdot (1 - 4\beta) \cdot v^2 k^4}{8 \cdot (1 - 4\beta) \cdot v^2 k^4 + \omega_0^2} \cdot a_0 \cdot \operatorname{erfc}(vk^2 t)^{\frac{1}{2}} \\ & + \sum_{i=1}^4 \frac{z_i}{Z_i} \cdot \left( \frac{\omega_0^2 \zeta_0}{z_i^2 - vk^2} \right) \cdot \exp((z_i^2 - vk^2) \cdot t) \cdot \operatorname{erfc}(z_i t^{\frac{1}{2}}), \end{aligned} \quad (6.2)$$

with  $z_i$  the roots of the following equation

$$\begin{aligned} z^4 - 4\beta \cdot (k^2 v)^{\frac{1}{2}} \cdot z^3 + 2 \cdot (1 - 6\beta) \cdot k^2 v z^2 + \\ 4 \cdot (1 - 3\beta) \cdot (k^2 v)^{\frac{3}{2}} \cdot z + (1 - 4\beta) \cdot v^2 k^4 + \omega_0^2 = 0, \end{aligned} \quad (6.3)$$

and  $Z_i$  determined from

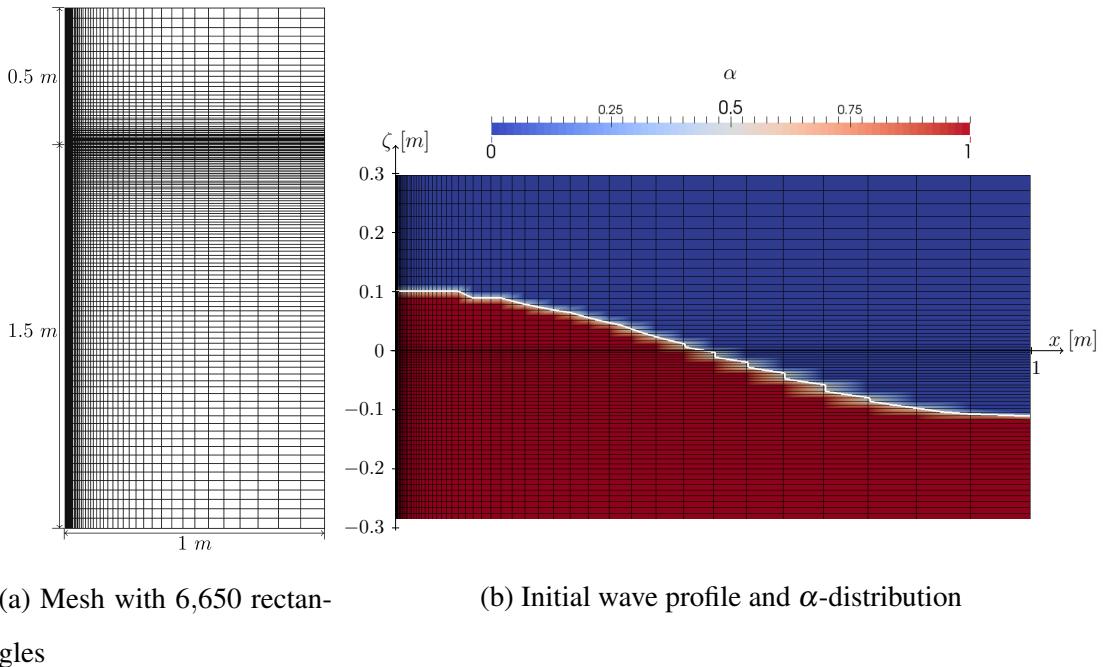
$$Z_1 = (z_2 - z_1) \cdot (z_3 - z_1) \cdot (z_4 - z_1). \quad (6.4)$$

Other  $Z_i$  are obtained by circular permutation of the index. Further,  $\operatorname{erfc}$  is the complex error function and  $k$  the wave number. The remaining parameters are calculated as

$$\begin{aligned} \omega_0 &= \sqrt{\frac{\rho_l - \rho_g}{\rho_l + \rho_g} \cdot gk + \frac{\zeta_0}{\rho_l + \rho_g} \cdot k^3}, \\ \beta &= \frac{\rho_l \rho_g}{(\rho_l + \rho_g)^2}. \end{aligned} \quad (6.5)$$

The fluid properties are chosen as  $\nu = 0.01 \frac{m^2}{s}$ ,  $g = 9.81 \frac{m}{s^2}$ ,  $\rho_l = 1000 \frac{kg}{m^3}$ , and  $\rho_g = 1 \frac{kg}{m^3}$ . Surface tension effects and turbulence are neglected in accordance with Damián [13]. Thus, the sinusoidal motion of the profile is driven by gravitational forces and damped by viscous stresses.

The grid is shown in figure 6.1a and consists of 6,650 rectangles with gradings at the undisturbed interface and near the left boundary. The second grading raises difficulties for the VOF method in case of high  $Co$  numbers. The simulations are evaluated by recording the amplitude of the profile at  $x = 0 m$  over a time period of 7 s and comparing it with the analytical progress using (6.2). Under consideration of Damian's demonstrations [13], the first computation is performed with a fixed time step of  $\Delta t = 10^{-4} s$ . Afterwards, a computation with a fixed  $Co$  number of 0.1, corresponding to three times higher time steps, is tried to execute.



(a) Mesh with 6,650 rectangles

(b) Initial wave profile and  $\alpha$ -distribution

Figure 6.1: Setup for the motion of two superposed viscous fluids

All results are presented in figure 6.2. For the version with a fixed time step, the results of `clsMULESFoam` and `interFoam` are almost identical. All extrema are predicted as  $\approx 20\%$  smaller than the analytical results except of the first maximum which overshoots the theoretical amplitude. The numerical period is minimum longer than the theoretical one but generally confirms the theory.

In case of increasing the time step by fixing the  $Co$  number, `clsMULESFoam` predicts still stable solutions but the period increases. In contrast, `interFoam` stops the calculation since no converged solution of the pressure equation could be determined. This is caused by an insufficient interface thickness near the left boundary due to high compression and  $Co$  numbers resulting in density and pressure discontinuities (compare chapter 5.2.3 for an explanation).

Both simulations indicate that the new solver does not introduce extra diffusion into the simulation since no additional damping occurs. Furthermore, it obviously enables higher  $Co$  numbers corresponding to an increase of efficiency in practise.

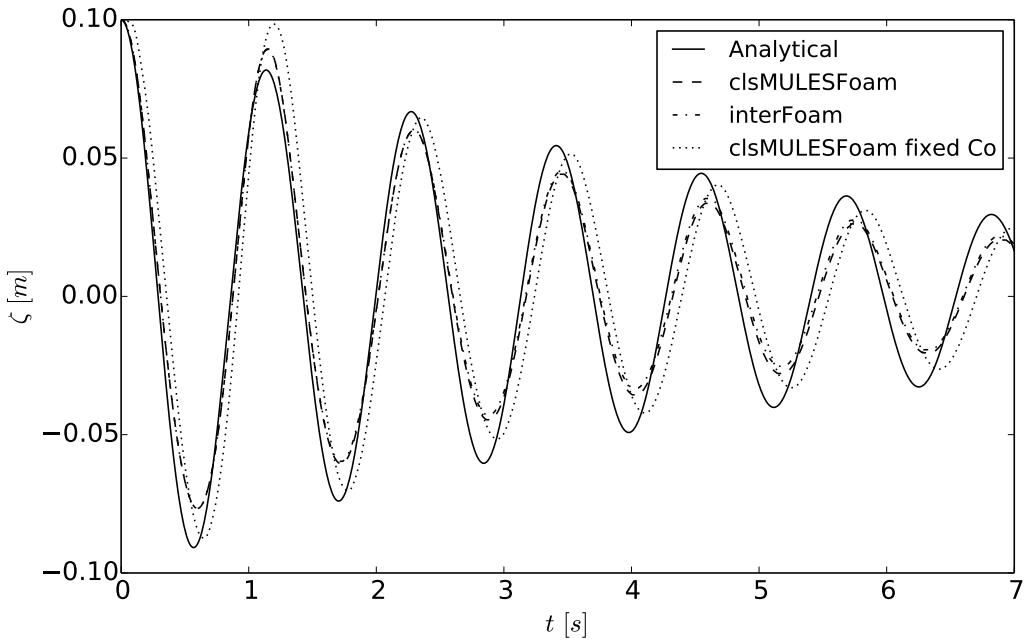


Figure 6.2: Amplitude of the wave at  $x = 0 \text{ m}$  over time

## 6.2 Two-Dimensional Breaking of a Dam

The next verification case is the two-dimensional breaking of a dam which is a classical application in multiphase flows including large deformations of the interface as well as high speed regions. The domain, as represented in figure 6.3, is a square box of  $1\text{ m} \times 1\text{ m}$  with a rectangular obstacle placed on the centre of the bottom. As the initial condition, a water column of  $0.25\text{ m} \times 0.5\text{ m}$  is placed in the left corner and starts collapsing at  $t = 0\text{ s}$ . The fluid properties of water are fixed as  $\rho_l = 1000 \frac{\text{kg}}{\text{m}^3}$  and  $v_l = 10^{-6} \frac{\text{m}^2}{\text{s}}$ , while the rest of the domain is filled with air whose properties are determined as  $\rho_g = 1 \frac{\text{kg}}{\text{m}^3}$  and  $v_g = 1.7 \cdot 10^{-5} \frac{\text{m}^2}{\text{s}}$ .

Two different types of meshes are considered. The first mesh consists of a mix of hexahedra and prisms (figure 6.4b) and the second mesh is structured with a grading at the right boundary (figure 6.4a). It results in similar behaviours of the `interFoam` solver as seen before.

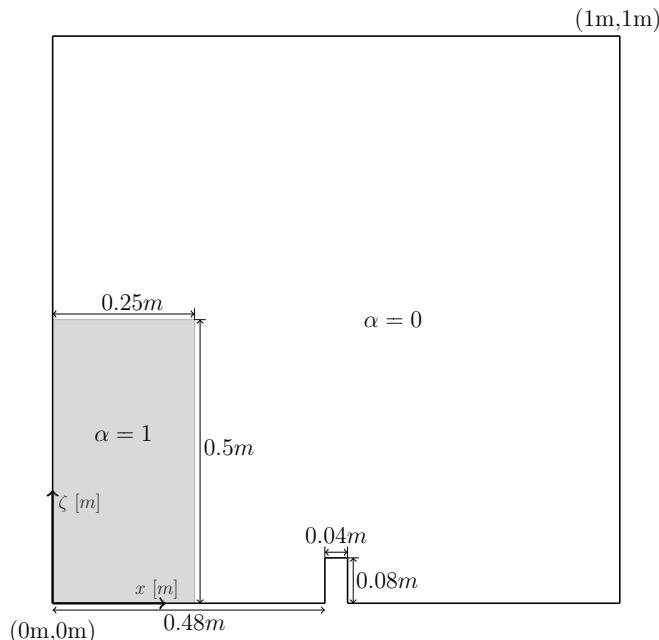


Figure 6.3: Setup for the two-dimensional breaking of a dam

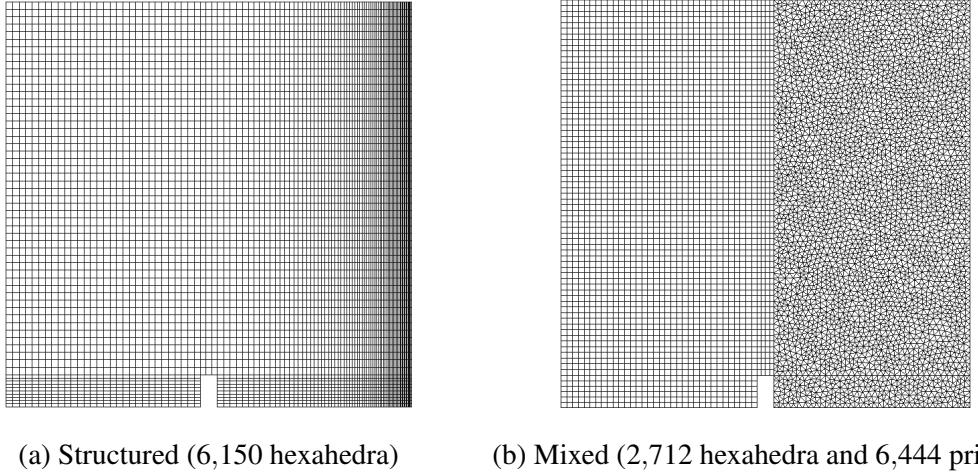
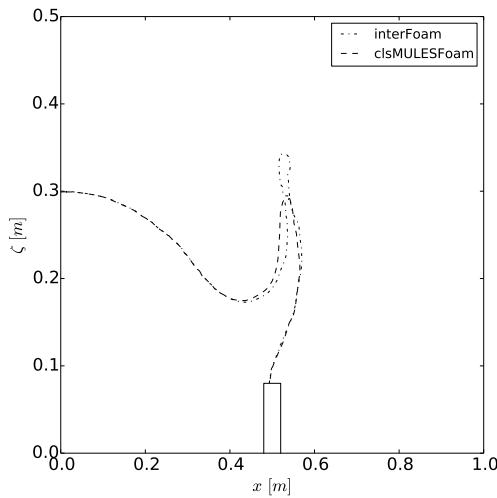
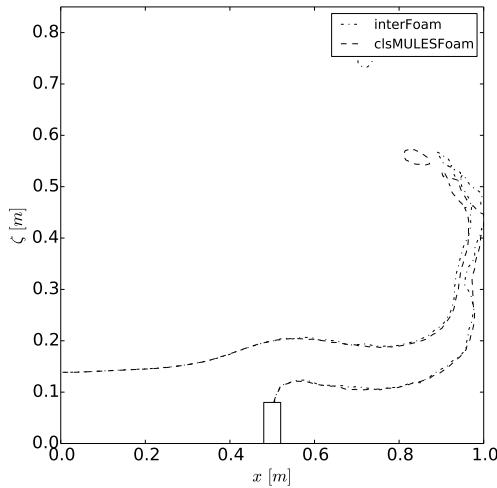
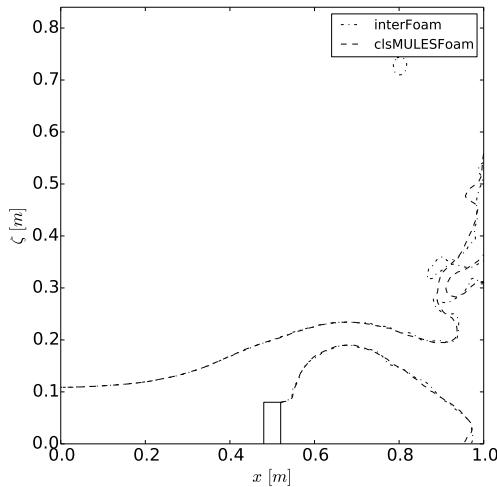
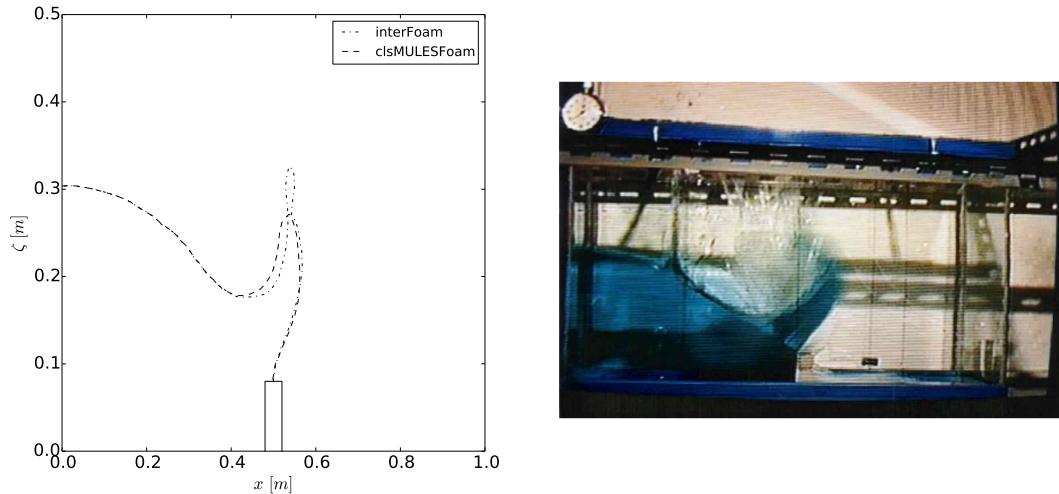
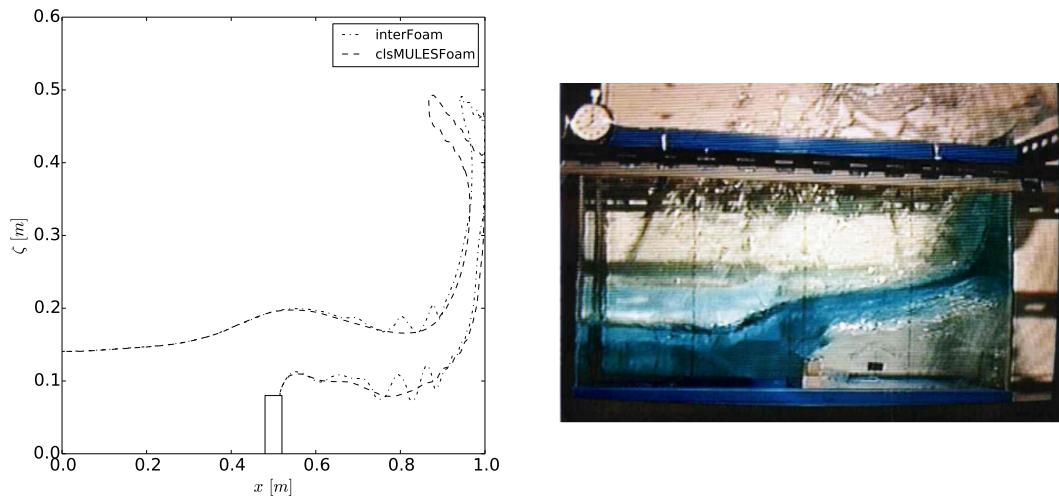
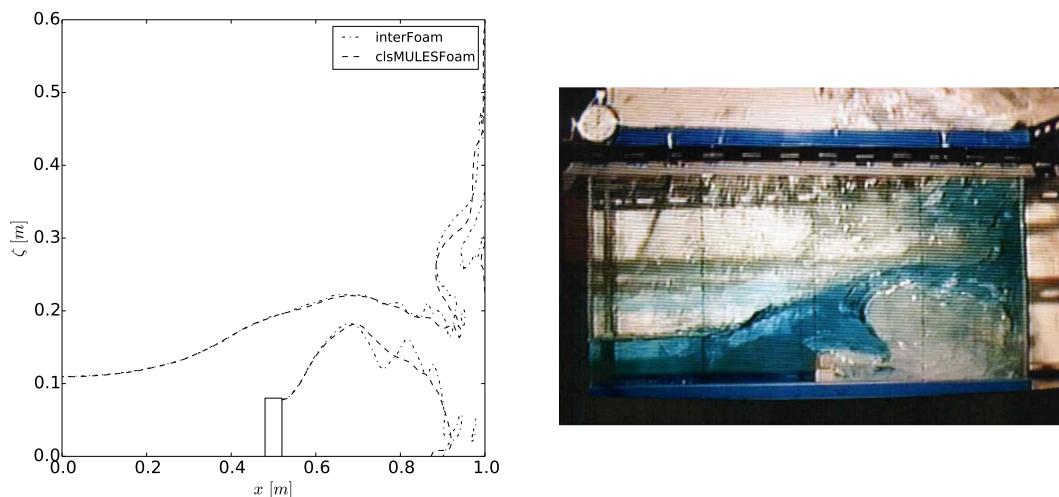


Figure 6.4: Considered meshes

The case is simulated with a specified  $Co$  number of 0.3. All domain boundaries are declared as walls with no-slip conditions. Unfortunately, no validation data is available except of pictures of an experiment. It is, therefore, decided to oppose the numerically predicted interface contours with these pictures as proposed by Greaves [30]. The comparison is executed at  $t = 0.258\text{ s}$ ,  $0.516\text{ s}$ , and  $0.645\text{ s}$  and presented in figure 6.5 and 6.6 respectively.

On the mixed mesh, both solvers produce similar results and compute the change in the mesh topology smoothly. `clsMULESFoam` tends to thicker jets and therefore, a later detachment of bubbles. In comparison to the experiment, the spray could not be resolved due to the grid resolution and principal disadvantages of these methods in case of large geometrical deformations and mixing. Furthermore, at  $t = 0.645\text{ s}$ , the lower water arm is not predicted. This detail also misses on the structured grid where `interFoam` fails due to the missing relaxation as expected. At  $x = 0.8\text{ m}$ , in figure 6.6b and 6.6c, non-physical oscillations occur which vanish, if `clsMULESFoam` is applied. In the rest of the domain both solvers calculate similar contours. However, `clsMULESFoam` tends to slightly deferred contours depending on the chosen  $Pe$  number.

(a) Numerical (left) and experimental result (right) at  $t = 0.258\text{ s}$ (b) Numerical (left) and experimental result (right) at  $t = 0.516\text{ s}$ (c) Numerical (left) and experimental result (right) at  $t = 0.645\text{ s}$ Figure 6.5: Resulting contours  $\alpha = 0.5$  on the mixed grid

(a) Numerical (left) and experimental result (right) at  $t = 0.258\text{ s}$ (b) Numerical (left) and experimental result (right) at  $t = 0.516\text{ s}$ (c) Numerical (left) and experimental result (right) at  $t = 0.645\text{ s}$ Figure 6.6: Resulting contours  $\alpha = 0.5$  on the structured grid

The extension of the two-dimensional breaking of a dam by a third dimension results in a new validation case with even bigger challenges as before. The arising three-dimensional breaking of a dam was documented by Issa and Violeau [38] and is depicted in figure 6.7. In comparison to the previous case, some fluid can avoid the obstacle resulting in a three-dimensional, chaotic impact at the left boundary of the domain. Further, the turbulence is modelled using the standard  $k-\varepsilon$  model. Other characteristics as boundary conditions, material properties, and simulation settings are kept constant.

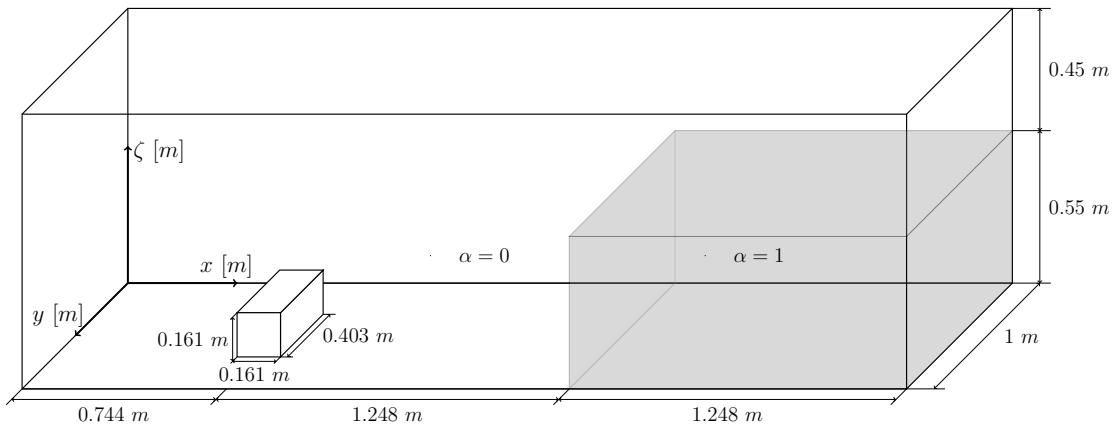


Figure 6.7: Setup for the three-dimensional breaking of a dam

The considered mesh consists of  $\approx 300,000$  tetrahedra and is shown in figure 6.8. The principal mesh topology does not change in  $y$ -direction for which reason just a slice of the mesh is presented. Two refinement levels are introduced, whereby the region around the initial water column and near the obstacle are better resolved. Further, the mesh is decomposed into several sub-domains using standard OpenFOAM tools in order to run the simulations on several processor. A detailed investigation regarding the parallel performance of the solvers can be found in chapter 6.5.

The evaluation is carried out by tracking the computed water column height at four different measuring points  $H_1 - H_4$  over time and comparing the distributions to ex-

perimental results provided by Arnold [3]. All measuring points are located on the symmetry plane  $y = 0.5 \text{ m}$  and their  $x$ -coordinates can be taken from figure 6.8. The resulting distributions are presented in figure 6.9.

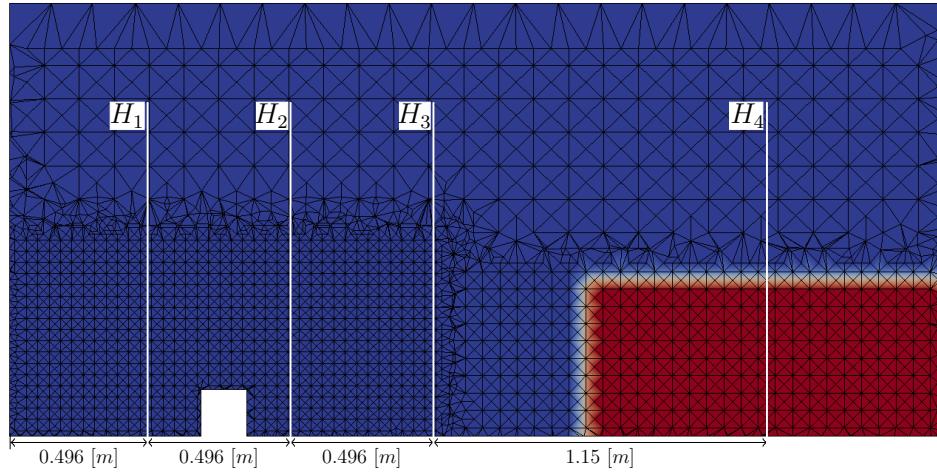


Figure 6.8: Slice of the mesh at  $y = 0.5 \text{ m}$  and locations of the four measuring points

At point  $H_1$ , both solvers have difficulties to predict the experimental distribution in the time interval  $t = 1.2 \text{ s}$  to  $2.8 \text{ s}$ . It should be, however, noticed that at this time the water behind the obstacle is strongly fluctuating which makes the correct determination of the actual columns height difficult. In case of calmer situations, the numerical results coincide much better with the experiment. The distribution of `interFoam` is generally more fluctuating than of the `clsMULESFoam` solver which might be a consequence of the additional diffusion.

The quality of the results can be measured by evaluating the  $\mathcal{L}_2$ -norm of errors at the different points due to the existence of experimental distributions. The norms are defined as follows

$$\mathcal{L}_2 = \sqrt{\int |\zeta_{exp} - \zeta_{num}|^2 dt}. \quad (6.6)$$

The integrals are computed numerically using the trapezoidal method which is sufficient

since no nominal order of accuracy is determined. The resulting norms are listed in table 6.2. At all measuring points, `clsMULESFoam` outperforms `interFoam` considerably due to the application of WENO reconstructions. They benefit from their transformation of the cells in a reference space without scaling effects and the missing collocated cells on tetrahedral meshes. Consequentially, using WENO schemes counteracts the usual degrading of accuracy on unstructured grids and improves the applicability of such meshes. The generation of tetrahedral meshes is less complicated than hexahedral meshes for which reasons a time consuming part can be omitted here.

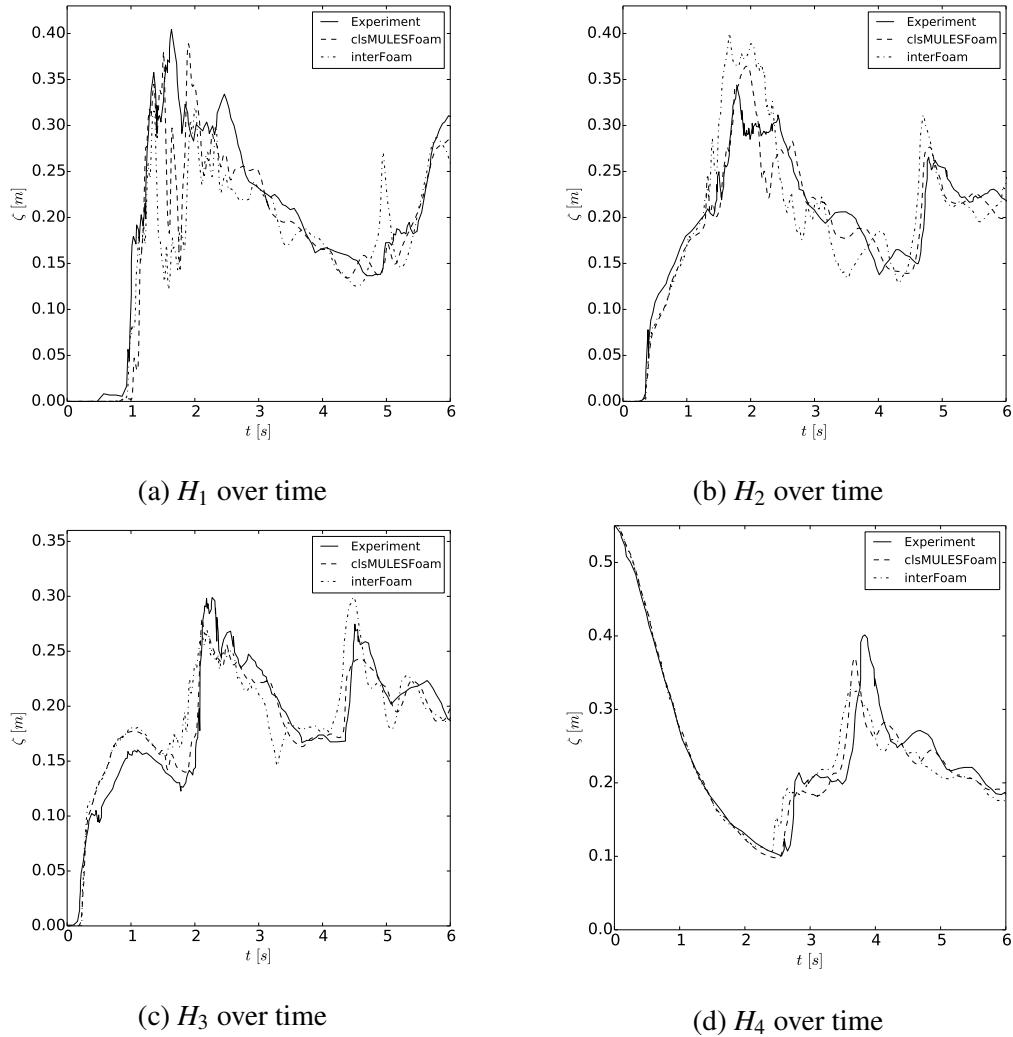


Figure 6.9: Measured water column heights over time

Table 6.2:  $\mathcal{L}_2$ -norm of the errors for the measured data

Solver	$H_1$	$H_2$	$H_3$	$H_4$
clsMULESFoam	0.110	0.057	0.042	0.071
interFoam	0.153	0.107	0.075	0.079

## 6.4 Wigley Hull

As a final example the free surface flow around the wigley hull shall be presented. Wigley hull represents a simplified ship model which has no bulbous bow or transom stern but is symmetric with respect to the midship section. Hence, its form can be described as the following analytical function

$$y(x, z) = \frac{B}{2} \cdot \left(1 - \left(\frac{2x}{L_{pp}}\right)^2\right) \cdot \left(1 - \left(\frac{z}{D}\right)^2\right), \quad (6.7)$$

with  $B = \frac{L_{pp}}{10}$  the maximum beam and  $D = \frac{L_{pp}}{16}$  the maximum draft.  $L_{pp}$  is chosen as 1 m and just a half model is considered in view of the symmetry and aim of the simulation which is the analysis of the free surface.

The mesh, shown in figure 6.11, is created using StarCCM+. A refinement box around the free surface increases the grid resolution in the  $\zeta$ -direction and several prism layers are generated next to the hull in order to guarantee  $y^+ \approx 100$ . This choice is fine enough for demonstrating the advantages of the new solver. The mesh includes several cells of poor quality (curved faces, concavity) leading to wrong decompositions of the affected faces in WENO reconstructions. As it was shown in the previous work [50] these problems reduce the theoretical order of accuracy but do not prevent the general application. The domain is delimited under consideration of the ITTC guidelines [40] and the resultant limits can be found in figure 6.10.

The inlet velocity  $v_\infty$  and kinematic viscosity of the water  $\nu_l$  are calculated from the

considered flow properties  $Fr = 0.316$  and  $Re = 4.4 \cdot 10^6$  in accordance to

$$\begin{aligned} v_\infty &= Fr \cdot \sqrt{gL_{pp}} = 0.989 \frac{m}{s}, \\ v_l &= \frac{v_\infty L_{pp}}{Re} = 2.25 \cdot 10^{-7} \frac{m^2}{s}. \end{aligned} \quad (6.8)$$

The further properties are  $\rho_l = 998 \frac{kg}{m^3}$ ,  $\rho_g = 1 \frac{kg}{m^3}$ , and  $v_g = 1.48 \cdot 10^{-5} \frac{m^2}{s}$ . The surface tension effects are neglected due to their minor effects on the solution. The turbulence is captured using the  $k-\omega$ -SST model in combination with wall functions at the hull.

In order to accelerate a convergence of the solution, the `LTSInterFoam` solver is applied for calculating an initial position of the surface. This solver allows local time stepping leading to non-physical but still accurate results. Afterwards, the `clsMULESFoam` and `interFoam` solver are applied with a fixed  $Co$  number of 0.2.

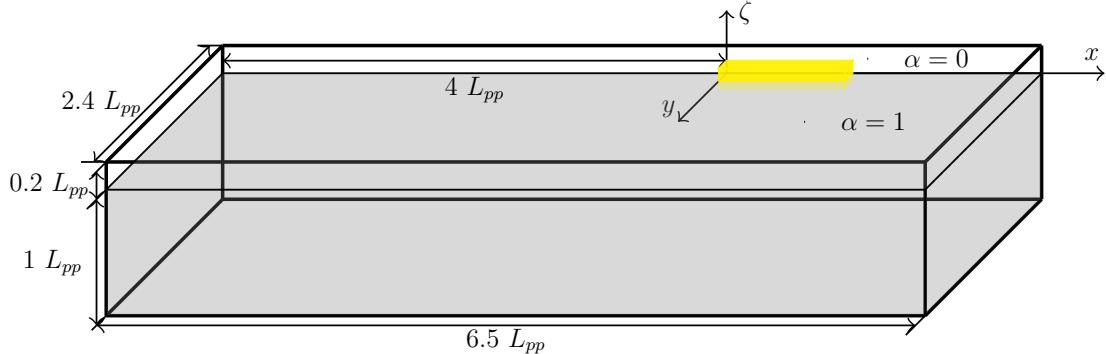


Figure 6.10: Setup for the wigley hull case

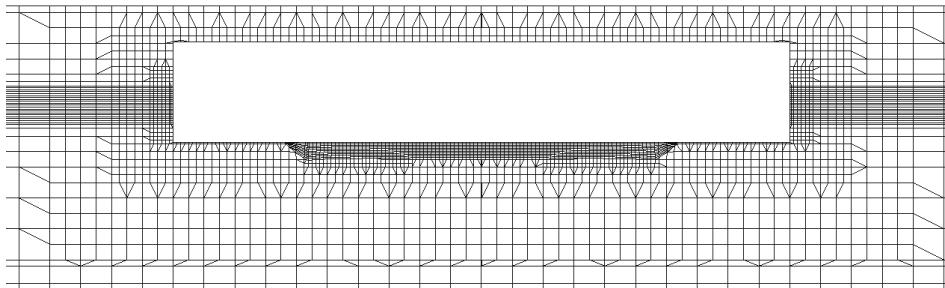


Figure 6.11: Slice of the mesh with 250,172 cells at  $y = 0 \text{ m}$

The convergence of the solution is checked by tracking the total drag coefficient  $c_d$  over time. It is calculated as the sum of pressure coefficient  $c_p$  and friction coefficient  $c_f$  which yields

$$c_d = c_p + c_f = \frac{p_d}{\frac{\rho}{2} \cdot v_\infty^2 S_w} + \frac{\tau_w}{\frac{\rho}{2} \cdot v_\infty^2 S_w}, \quad (6.9)$$

with  $p_d$  the dynamic pressure,  $\tau_w$  the  $x$ -component of the wall shear stress, and  $S_w$  the wetted surface given by [37]

$$S_w = C_s L_{pp} \cdot (2D + B), \quad C_s = 0.661. \quad (6.10)$$

The resulting coefficients are listed in table 6.3 together with their deviation from the experimental value. The difference of the predicted forces is small which indicates the general affinity of the solvers. The difference of about 12% to the experimental data is mainly caused by increased friction forces resulting from wall functions and high  $y^+$  values.

Table 6.3: Force coefficients of converged solutions

Solver	$c_p$	$c_f$	$c_d$	error $c_d$ [%]
<code>clsMULESFoam</code>	$1.06 \cdot 10^{-3}$	$4.72 \cdot 10^{-3}$	$5.78 \cdot 10^{-3}$	+12.02
<code>interFoam</code>	$1.03 \cdot 10^{-3}$	$4.76 \cdot 10^{-3}$	$5.79 \cdot 10^{-3}$	+12.21

In figure 6.12 and 6.13 the contour around the ship is presented first. The improper calculation of the interface normal vectors of `interFoam` leads to an oscillatory surface near the hull where elongated, flat cells occur. In comparison, `clsMULESFoam` damps these effects without loosing accuracy at other wave crests and troughs.

This could be also seen from the comparison of the resulting wave pattern along the ship with experimental data of H.Kajitani et al. [37] in figure 6.14. The computed surface using `interFoam` can in principal reproduce the experimental result, but in the interval  $x = 0.2 \text{ m}$  to  $0.75 \text{ m}$ , the solution oscillates non-physically. `clsMULESFoam` predicts a similar but non-oscillatory surface with a slightly more accurate reproduction of

the first crest. However, the theoretically increased order of accuracy using WENO reconstructions does not increase the overall accuracy in contrast to the three-dimensional breaking of a dam case. A possible reason could be the mesh topology consisting of many collocated hexahedral cells which results in bad conditioned reconstruction matrices. This makes the recommendation of WENO schemes to ship flows difficult, in particular under consideration of the efficiency as it will be shown next.

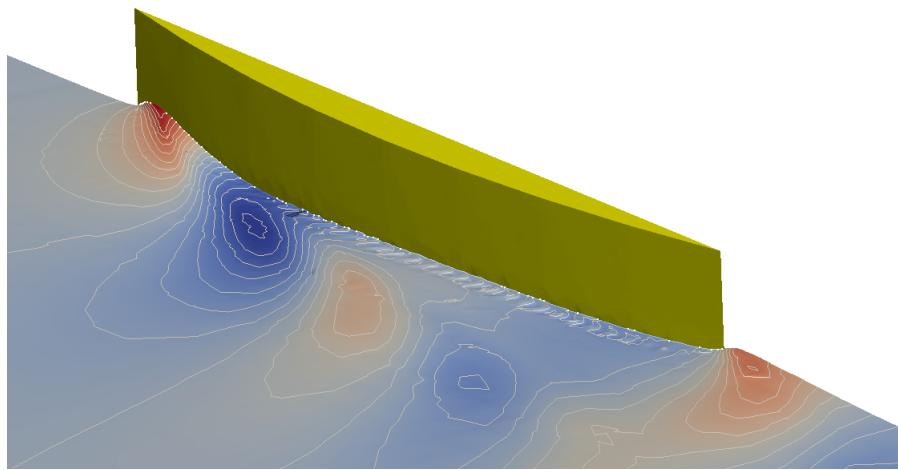


Figure 6.12: Resulting water surface using `interFoam`

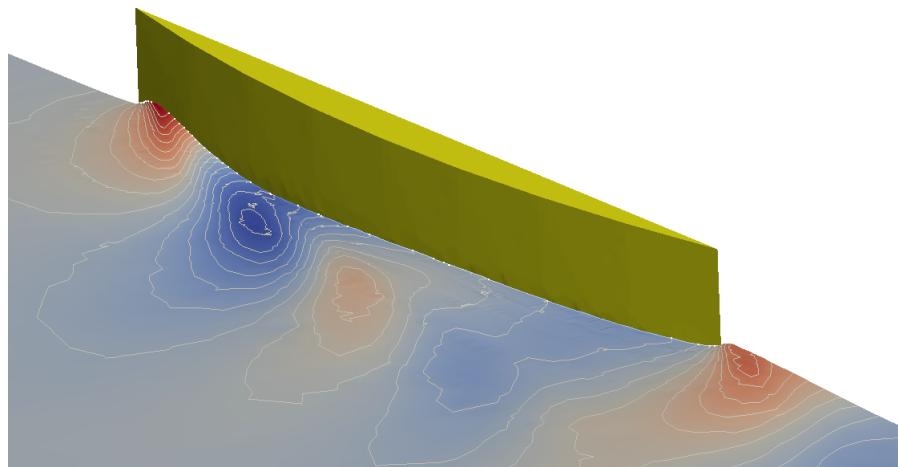


Figure 6.13: Resulting water surface using `clsMULESFoam`

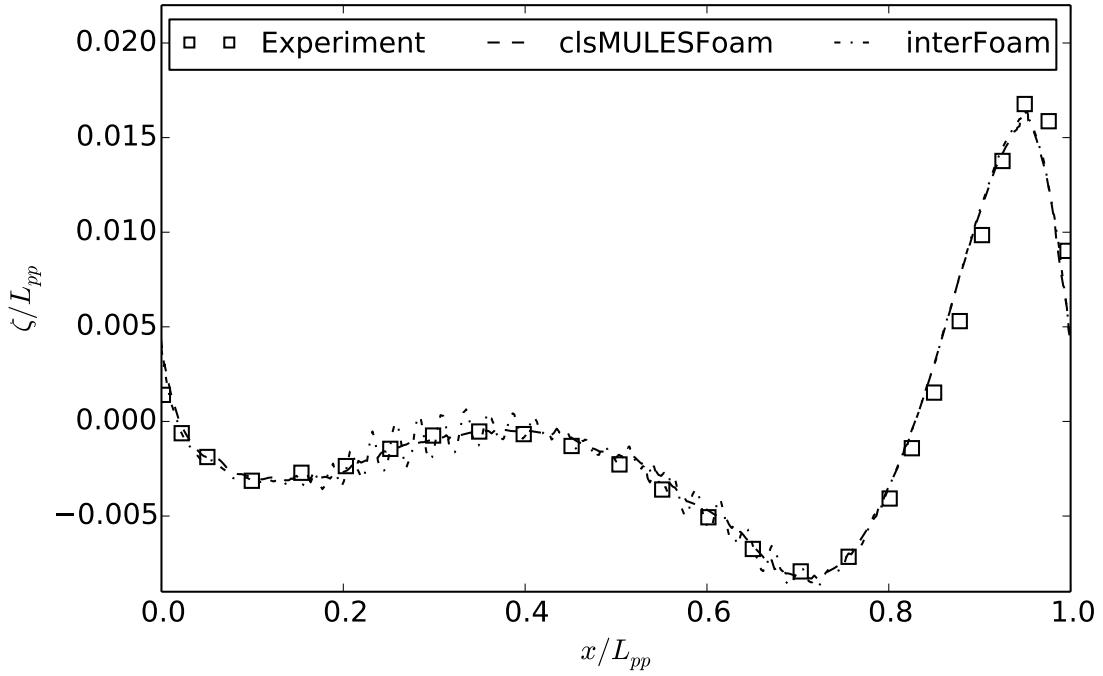


Figure 6.14: Longitudinal wave cut along the hull. Stern at  $x = 0\text{ m}$ , bow at  $x = L_{pp}$ .

## 6.5 Evaluation of the Performance

The previous verification cases showed that `clsMULESFoam` improves the results in situations in which `interFoam` produces oscillatory interfaces. However, secondary results as force coefficients are not influenced by these instabilities. Accuracy improvements of the overall computation could just be stated on unstructured grids and are rather reached by the applied WENO schemes than by the modified surface model. The sense of using high-order schemes and relaxation in the surface model respectively is a question of performance of the resulting tool. For this purpose, an impression of the implementation's efficiency is provided in this chapter. All presented computations are executed on the local Neptun cluster with about 2,700 processors (AMD 6172, 2.2 GHz) and each of them contains 12 cores each with 2 GB of RAM.

As a first step, the efficiency reduction by applying the additional diffusion steps of `clsMULESFoam` in comparison to `interFoam` is presented in table 6.4. The comparability is ensured by executing both solvers with the same schemes which are the standard schemes of the right column in table 6.1. The time increase is between 10% and 15% in case of calm situations and small diffusion coefficients. This value is nearly doubled if the diffusion coefficients increase due to acceleration of the flow as in the impact situations of the breaking of a dam case.

Table 6.4: Time measurements on 4 processor for two test cases using the same standard schemes. Values are percentage time increases per relaxation step of `clsMULESFoam` in comparison to `interFoam`.

Wigley hull (calm situation)	3D dam-break (calm situation)	3D dam-break (impact situation)
10.1	15.4	32.3

In comparison, the efficiency of the method is significantly decreased by switching from standard to WENO schemes. Table 6.5 shows the time increase per WENO reconstruction of `clsMULESFoam` in comparison to the same computations with TVD schemes. Thus, the application of the fourth-order accurate `WENOUpwindFit` scheme in MULES increases the overall period of a time step by a factor of 2.3. The influence of the relaxation equation vanishes here. The increased computational effort is of course the main disadvantage of all high-order methods, in particular, in FVM. Two approaches of reducing computational time for these reconstructions are conceivably and both will be investigated below.

Table 6.5: Time measurements on 4 processor for two test cases using WENO reconstructions. Values are percentaged time increases per WENO reconstruction using `clsMULESFoam` in comparison to `clsMULESFoam` with standard scheme.

$r$	Wigley hull	3D dam-break	3D dam-break
		(calm situation)	(impact situation)
1	17	16	17
2	88	78	79
3	128	139	142

The first way of counteracting higher computational effort is the decomposition of the domain into more processors. However, using more processors also means more inter-processor communication for which reason the overall efficiency is limited to a maximum number of processors. In figure 6.15 the elapsed time per cell for one time step for the three-dimensional breaking of a dam case is shown. The comparability is ensured by applying `interFoam` with different convection schemes in MULES. The reference is given by the TVD-vanLeer scheme (black line) whose time period almost reaches the minimum time step size at 16 processors or about 20,000 cells per processor. In comparison, the `WENOUpwindFit` scheme reaches smaller periods by increasing the number of processors further. It should be noticed that all WENO schemes reach the same time as the TVD scheme in case of 32 processors or 10,000 cells per processor. In more common decompositions with about 80,000 cells per processors, it is readable that similar time periods could be reached by the fourth-order accurate `WENOUpwindFit` scheme as by TVD schemes by executing the computation on 3 times more processors. This number would further increase in view of the possible application of several WENO reconstructions per time step. It would be, therefore, preferable to improve the efficiency of the implementation next. General starting points for code improvements are given as a final remark in chapter 7.

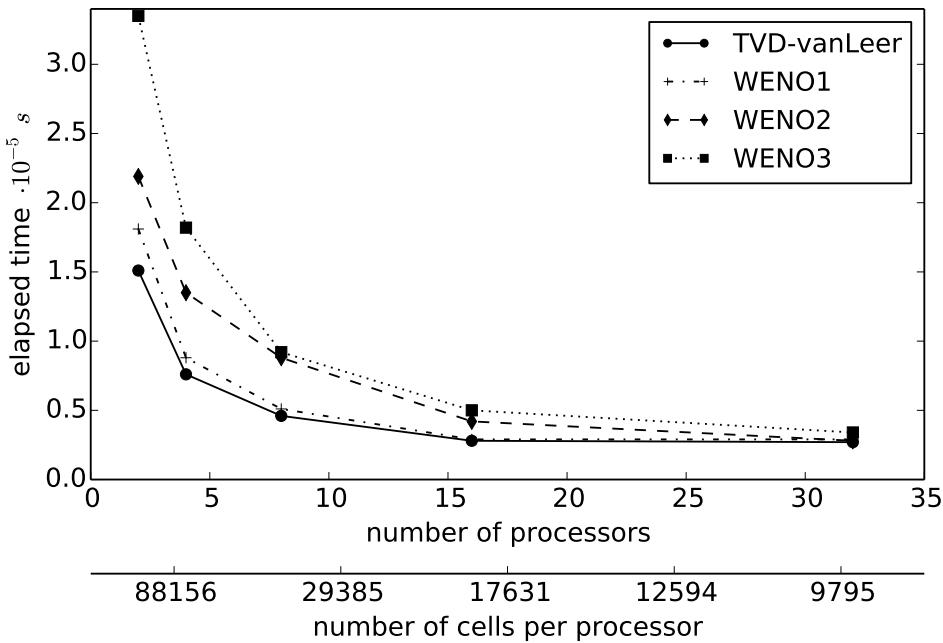


Figure 6.15: Elapsed time per cell for one time step and different number of processors. All computations are executed using `interFoam` and one WENO reconstruction at most.

In order to get an overview of the reconstruction's performance a detailed listing of the sub-steps of a WENO reconstruction is given in figure 6.16. It shows the percentage partition of the total reconstruction time per cell with an explanation of the single steps in the caption of the figure. At first, the impression of figure 6.15 is confirmed since the percentage of the inter-processor communication is generally small even on 32 processors. The disadvantage of more inter-processor communication vanishes in contrast to the remaining calculation. The maximum percentage is expectable for the fourth-order accurate scheme which is caused by the extended stencils and to that effect more halo cells. The by far biggest percentage is related to the calculation of the degrees of freedom in each stencil of a cell. This step takes between 60% and 80% of the overall time which is, however, obvious since the weighting, as the remaining step of the calculation, has to be executed just once.

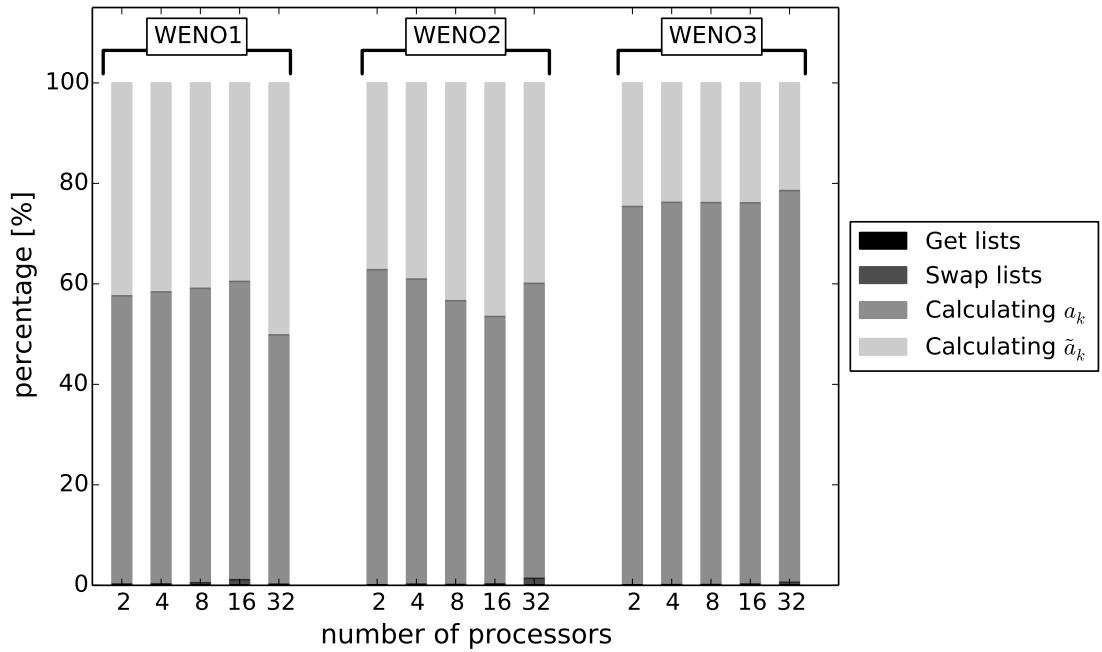


Figure 6.16: Percentage of single sub-steps of a WENO reconstruction. Explanation for the legend: Get lists - receive necessary lists from WENOBase, Swap lists - interprocessor communication and transmission of data of halo cells, Calculating  $a_k$  - matrix vector multiplication for obtaining coefficients of sub-stencils, Calculating  $\tilde{a}_k$  - weighting  $a_k$  and summing up

# **Chapter 7**

## **Conclusion**

The main accomplishments of this thesis consist in the generation of a high-order non-oscillatory framework in OpenFOAM and the development of a new conservative level set method for incompressible two-phase flows. In doing so, all implementations are embedded in the existing code for which reason the schemes and models can be arbitrarily selected by the user. Several applications provide an overview of advantages and possible problems using the presented tools which, in addition, accentuate the need for further improvements.

## **Interface Modelling**

The preliminary investigation of standard level set methods shows the typical strengths and weaknesses of the principal approach. The application of ADER schemes for the advection of the scalar level set offers high spatial and temporal accuracy as well as efficiency at once. However, the existing redistancing algorithms are not suitable for the given framework and the improvement of mass conservation is a cumbersome task on unstructured grids. Irrespective of these drawbacks, a combination with the mentioned projection method could provide a suitable high-order solution algorithm for the Navier-

Stokes equations in two-phase flows. The resulting solver could improve the standard solvers of OpenFOAM, in particular in applications with geometrically complex interfaces as breaking waves, but would be restricted to the CFL-condition.

Therefore, the newly-created `clsMULESFoam` solver was successfully developed in order to prevent oscillatory results which may occur in `interFoam` and loose the restriction on the CFL-condition at the same time. Weller's limiting strategy MULES turns out to be a confident method to obtain bounded solutions even for very high-order fluxes. It should be noticed, that MULES reduces the nominal order of accuracy due to the first guess using low-order schemes and the Euler explicit time integration for the correction. In contrast, an accuracy preserving limiter is provided and successfully implemented. The new limiter is calculated very efficiently due to missing iterations. A possible application in `clsMULESFoam` is prevented by the resulting restriction of the solver to the usage of WENO schemes.

The principal working of the proposed level set relaxation method with a new definition of the diffusion coefficients could be demonstrated. So far, the dependency on the  $Pe$  number leads to loss of accuracy at sharp kinks. It might be possible to add the dependency on the local curvature of the interface to the coefficients since the compressive VOF method preserve accurate contours in these regions. Thereby, the diffusion could be reduced near kinks and reproduce `interFoam`'s results.

The new solver is based on OpenFOAM's PISO algorithm for coupling pressure and velocity in a semi-implicit manner. The advantage of applying WENO schemes in this algorithm is possibly small since the implementation is based on the at most second-order accurate framework of OpenFOAM. A further problem of the two-phase PISO algorithm is the occurrence of spurious air velocities which could influence the interface model negatively through improper compression and diffusion terms. According to Vukčević and Jasak [94], the unresolved coupling of pressure and density gradients in the CSF model effects their occurrence. This phenomenon was already stated in chapter

2.2 where (2.14) expresses the balance between pressure and density. Besides the alternative treatment of the singularities by applying the ghost fluid method, Vukčević and Jasak [94] propose to drop the CSF model for density and rather take the pressure jump condition including density into account for interpolations of the pressure near interfaces. Thereby, an implementation on arbitrary unstructured grids is a straightforward task in comparison to the ghost fluid method.

## WENO Reconstructions

The basis for the presented high-order schemes is the WENO reconstruction whose development was mainly advanced in the previous project work [50]. The method is derived from the approaches of Dumbser and Käser [18] and Pringuey [70], and handles two- and three-dimensional polyhedral meshes in parallel. Most computational work is moved to the preprocessing in order to receive more efficient schemes in runtime. The implementations are integrated in the given interpolation and gradient classes which simplifies further developments with the code.

The `WENOUpwindFit` scheme extends the class of high-order upwind schemes in OpenFOAM and shows superior results in convection dominated flows. The evaluation of non-linear compression fluxes by the `WENOCompressionFit` scheme indicates the simplicity of applying the WENO reconstructions to any conservative convection terms and solving Riemann problems within the default interpolation class. In addition, the derived gradient scheme `WENOGrad` outperforms the standard linear gradient scheme and the same is expectable for the surface normal gradient scheme `WENOSnGrad`. It does not fit in the given formulas and is, therefore, just applicable to explicit discretisations of transport equations and not to diffusion terms, in general.

The presented efficiency analysis exposes no obvious room for improvement for the existing implementation. However, the presented reconstruction can also be taken as a

starting point for more efficient high-order schemes:

- Limited least-squares schemes as proposed by Michalak and Ollivier-Gooch [54]. They just need one central stencil and no weighting of the coefficients. The limiter could be implemented within the interpolation class similarly to the presented one. Hence, the final degrees of freedom would be calculated about four times faster than using WENO reconstructions. But, the scheme looses accuracy near extrema and discontinuities as TVD schemes.
- Adaptive WENO scheme of Costa and Don [11] which diminishes to a low-order upwind scheme in smooth regions and applies WENO in regions of high gradients. Thereby, a fast algorithm could be developed especially for ship flows where the influence of surfaces is just active in a narrow band of the domain. The difficulty is the definition of a proper criterion for distinguishing between the different regions and the preservation of the accuracy of the scheme.

Irrespective to the efficiency, the accuracy of the existing reconstruction could be increased by the following suggestions:

- The face decomposition algorithm of OpenFOAM is restricted to planar boundaries. Complex meshes can not guarantee this property which may lead to defective reconstructions. This problem could be minimised by generating a tetrahedralization of each cell and evaluate volume instead of surface integrals [70]. It would result in quadrature free reconstructions, as Dumbser and Käser [18] proposes, due to the possibility of mapping each tetrahedra on a standard tetrahedron which provides analytical results for all volume integrals. Further, it would enable the generation of more accurate gradient schemes due to the possibility of defining proper coordinates of Gaussian points of a cell using its included mapped tetrahedra (compare chapter 3.4).

- OpenFOAM's boundary conditions are designed for second-order accuracy at most. Hence, high-order schemes loose accuracy near boundaries if the formulas are not modified. Theoretically, boundary conditions could be included into the reconstruction as it was shown in the project work [50]. It would, however, be very inefficient since a new preprocessing step would have to be provided for each new variable.



# Bibliography

- [1] R. Abgrall. “On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation”. In: *Journal of Computational Physics* 114 (1) (1994), pp. 45–58.
- [2] A. Alexandrescu. *Modern C++ Design: Generic Programming and Design Patterns Applied*. Addison Wesley, 2001.
- [3] P. Arnold. *Validation of a 3D Dam Breaking Problem*. URL: <https://www.flow3d.com/validation-flow-3d-3d-dam-breaking-problem> (Accessed: 08/10/2016).
- [4] G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 2007.
- [5] J.P. Boris and D.L. Book. “Flux-corrected transport. I: SHASTA, a fluid transport algorithm that workss”. In: *Journal of Computational Physics* 11 (1) (1973), pp. 38–69.
- [6] J.U. Brackbill, D.B. Kothe, and C. Zemach. “A Continuum Method for Modeling Surface Tension”. In: *Journal of Computational Physics* 100 (2) (1992), pp. 335–354.
- [7] T. Cecil, J. Qian, and S. Osher. “Numerical methods for high dimensional Hamilton–Jacobi equations using radial basis functions”. In: *Journal of Computational Physics* 196 (1) (2004), pp. 327–347.

- [8] A.J. Chorin. “A Numerical Method for Solving Incompressible Viscous Flow Problems”. In: *Journal of Computational Physics* 135 (2) (1997), pp. 118–125.
- [9] A.J. Chorin and J.E. Marsden. *A Mathematical Introduction to Fluid Mechanics*. Springer Verlag, 2000.
- [10] G. Colicchio, M. Landrini, and J. Chaplin. *Level-set modelling of the two-phase flow generated by a surface piercing body*. 18th International Workshop on Water Waves and Floating Bodies. (Accessed: 07/30/2016). 2003.
- [11] B. Costa and W.S. Don. “High order Hybrid central—WENO finite difference scheme for conservation laws”. In: *Journal of Computational and Applied Mathematics* 204 (2) (2007), pp. 209–218.
- [12] R. Courant, H. Lewy, and K. Friedrichs. “Über die partiellen Differenzengleichungen der mathematischen Physik”. In: *Mathematische Annalen* 100 (1928), pp. 32–74.
- [13] S.M. Damián. *Description and utilization of interFoam multiphase solver*. URL: <http://infofich.unl.edu/upload/3be0e16065026527477b4b948c4caa7523c8ea52.pdf> (Accessed: 08/10/2016).
- [14] M.S. Darwish and F. Moukalled. “TVD Schemes for Unstructured Grids”. In: *International Journal of Heat and Mass Transfer* 46 (4) (2003), pp. 599–611.
- [15] S. Deng. *Quadrature formulas in two dimensions, Lecture Math 5172 - Finite element method*. URL: [http://math2.uncc.edu/~shaodeng/TEACHING/math5172/Lectures/Lect\\_15.PDF](http://math2.uncc.edu/~shaodeng/TEACHING/math5172/Lectures/Lect_15.PDF) (Accessed: 07/30/2016).
- [16] F. Denner and B.G.M. van Wachem. “Compressive VOF method with skewness correction to capture sharp interfaces on arbitrary meshes”. In: *Journal of Computational Physics* 279 (1) (2014), pp. 127–144.

- [17] O. Desjardins, V. Moureau, and H. Pitsch. “An accurate conservative level set / ghost fluid method for simulating turbulent atomization”. In: *Journal of Computational Physics* 227 (18) (2008), pp. 8395–8416.
- [18] M. Dumbser and M. Käser. “Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems”. In: *Journal of Computational Physics* 221 (3) (2007), pp. 693–723.
- [19] M. Dumbser et al. “ADER schemes on unstructured meshes for nonconservative hyperbolic systems: Applications to geophysical flows”. In: *Computers & Fluids* 38 (2009), pp. 1731–1748.
- [20] W.V. Edwards. *Towards a Level Set Reinitialisation Method for Unstructured Grids*. PH.D.-thesis, Institute for Energy Systems, University of Edinburgh. (Accessed: 07/30/2016). 2011.
- [21] D. Enright et al. “A Hybrid Particle Level Set Method for Improved Interface Capturing”. In: *Journal of Computational Physics* 183 (1) (2002), pp. 83–116.
- [22] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 2010.
- [23] R.P. Fedkiw et al. “A Non-oscillatory Eulerian Approach to Interfaces in Multi-material Flows (the Ghost Fluid Method)”. In: *Journal of Computational Physics* 152 (2) (1999), pp. 457–492.
- [24] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer Verlag, 2002.
- [25] O. Friedrich. “Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids”. In: *Journal of Computational Physics* 114 (1) (1998), pp. 194–212.

- [26] P.-G. de Gennes, F. Brochard-Wyart, and D. Quéré. *Capillarity and Wetting Phenomena - Drops, Bubbles, Pearls, Waves*. Springer Verlag, 2004.
- [27] S.K. Godunov. “A Finite Difference Method for the Computation of Discontinuous Solutions of the Equations of Fluid Dynamics”. In: *Matematicheskii Sbornik* 47 (1959), pp. 357–393.
- [28] G.H. Golub and C.F. van Loan. *Matrix computations*. John Hopkins University Press, 1996.
- [29] V.R. Gopala and B.G.M. van Wachem. “Volume of fluid methods for immiscible-fluid and free-surface flows”. In: *Chemical Engineering Journal* 141 (2008), pp. 204–221.
- [30] D.M. Greaves. “Simulation of viscous water column collapse using adapting hierarchical grids”. In: *Int. J. Numer. Meth. Fluids* 50 (2006), pp. 693–711.
- [31] A. Harten. “High Resolution Schemes for Hyperbolic Conservation Laws”. In: *Journal of Computational Physics* 49 (3) (1983), pp. 357–393.
- [32] A. Harten et al. “Uniformly high order accurate essentially non-oscillatory schemes, III”. In: *Journal of Computational Physics* 131 (1) (1997), pp. 3–47.
- [33] D.J.E. Harvie, M.R. Davidson, and M. Rudman. “An analysis of parasitic current generation in Volume of Fluid simulations”. In: *Applied Mathematical Modelling* 30 (10) (2006), pp. 1056–1066.
- [34] A.K. Henrick, T.D. Aslam, and J.M. Powers. “Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points”. In: *Journal of Computational Physics* 207 (2005), pp. 542–567.
- [35] C. Hirsch. *Numerical Computation of Internal and External Flows - Volume 1*. John Wiley & Sons Ltd., 2007.

- [36] C. Hirsch. *Numerical Computation of Internal and External Flows - Volume 2.* John Wiley & Sons Ltd., 2007.
- [37] H.Kajitani et al. *The Summary the Cooperative Experiment on Wigley Parabolic Model Japan.* URL: <http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2/p003037.pdf> (Accessed: 08/10/2016).
- [38] R. Issa and D. Violeau. *ERCOFTAC, SPHERIC SIG test-case 2: 3D dambreaking.* URL: [http://app.spheric-sph.org/sites/spheric/files/SPHERIC\\_Test2\\_v1p1.pdf](http://app.spheric-sph.org/sites/spheric/files/SPHERIC_Test2_v1p1.pdf) (Accessed: 08/10/2016).
- [39] R.I. Issa. “Solution of the Implicitly Discretised Fluid Flow Equations by Operator-Splitting”. In: *Journal of Computational Physics* 62 (1) (1986), pp. 40–65.
- [40] ITTC. *Practical Guidelines for Ship CFD Applications.* International Towing Tank Conference. (Accessed: 07/30/2016). 2011.
- [41] H. Jasak. *Error analysis and estimation for the finite volume method with applications to fluid flows.* PH.D.-thesis, Department of Mechanical Engineering, Imperial College of Science, London. (Accessed: 07/30/2016). 1996.
- [42] G.-S. Jiang and C.-W. Shu. “Efficient implementation of weighted ENO schemes”. In: *Journal of Computational Physics* 126 (1) (1996), pp. 202–228.
- [43] P.K. Khosla and S.G. Rubin. “A Diagonally Dominant Second-Order Accurate Implicit Scheme”. In: *Computers & Fluids* 2 (1974), pp. 207–209.
- [44] C.L. Lawson and R.J. Hanson. *Solving least squares problems.* SIAM, 1995.
- [45] P. Lax. “Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation”. In: *Comm. Pure Appl. Math.* 7 (1954), pp. 159–193.
- [46] P. Lax and B. Wendroff. “Systems of Conservation Laws”. In: *Comm. Pure Appl. Math.* 13 (1960), pp. 217–237.

- [47] R.J. Leveque. *Finite-volume methods for hyperbolic problems*. Cambridge University Press, 2004.
- [48] X. Liu, S. Osher, and T. Chan. “Weighted essentially non-oscillatory schemes”. In: *Journal of Computational Physics* 115 (1) (1994), pp. 200–212.
- [49] H. Marschall. Private communication about level set methods. University of Darmstadt. 2016.
- [50] T. Martin. *Solving the level set equation using high-order non-oscillatory reconstruction*. Study research project, Chair of Modelling and Simulation, University of Rostock. 2016.
- [51] G. Dal Maso, P.G. LeFloch, and F. Murat. “Definition and weak stability of nonconservative products”. In: *J. Math. Pures Appl.* 74 (6) (1995), pp. 483–548.
- [52] S. McDonald. *Development of a high-order finite-volume method for unstructured meshes*. Master’s thesis, University of Toronto. (Accessed: 07/30/2016). 2011.
- [53] F.R. Menter. “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications”. In: *AIAA Journal* 32 (8) (1994), pp. 1598–1605.
- [54] C. Michalak and C.F. Ollivier-Gooch. “Accuracy preserving limiter for the high-order accurate solution of the Euler equations”. In: *Journal of Computational Physics* 228 (2009), pp. 8693–8711.
- [55] B. Mirtich. “Fast and accurate computation of polyhedral mass properties”. In: *Journal of graphics tools* 1 (2) (1996), pp. 31–50.
- [56] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics*. Springer Verlag, 2016.

- [57] F. Mut, G.C. Buscaglia, and E.A. Dari. “New Mass-Conserving Algorithm for Level Set Redistancing on Unstructured Meshes”. In: *J. Appl. Mech.* 73 (6) (2006), pp. 1011–1016.
- [58] B.D. Nichols, C.W. Hirt, and R.S. Hotchkiss. “A Fractional Volume of Fluid Method for Free Boundary Dynamics”. In: *Lecture Notes in Physics - Springer Verlag* 141 (1981), pp. 304–309.
- [59] C.F. Ollivier-Gooch. *Higher-order least-squares reconstruction for turbulent aerodynamic flows*. 11th World Congress on Computational Mechanics. (Accessed: 07/30/2016). 2003.
- [60] C.F. Ollivier-Gooch. “Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction”. In: *Journal of Computational Physics* 133 (1) (1997), pp. 6–17.
- [61] C.F. Ollivier-Gooch and M. van Altena. “A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation”. In: *Journal of Computational Physics* 118 (1994), pp. 729–752.
- [62] E. Olsson and G. Kreiss. “A conservative level set method for two phase flow”. In: *Journal of Computational Physics* 210 (1) (2005), pp. 225–246.
- [63] OpenFOAM. *OpenFOAM C++ documentation - version 2.3.x*. URL: <http://foam.sourceforge.net/docs/cpp/index.html> (Accessed: 08/10/2016).
- [64] S. Osher and S. Chakravarthy. “High resolution schemes and the entropy condition”. In: *SIAM Journal on Numerical Analysis* 21 (5) (1984), pp. 955–984.
- [65] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2000.

- [66] S. Osher and J.A. Sethian. “Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations”. In: *Journal of Computational Physics* 79 (1) (1988), pp. 12–49.
- [67] C. Parés. “Numerical Methods for Nonconservative Hyperbolic Systems: A Theoretical Framework”. In: *SIAM Journal on Numerical Analysis* 44 (1) (2006), pp. 300–321.
- [68] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation, 1980.
- [69] C.S. Peskin. “The immersed boundary method”. In: *Acta Numerica* 11 (2002), pp. 1–39.
- [70] T. Pringuey. *Large eddy simulation of primary liqiud-sheet breakup*. PH.D.-thesis, Hopkinson Laboratory, University of Cambridge. (Accessed: 07/30/2016). 2012.
- [71] A. Prosperetti. “Motion of two superposed viscous fluids”. In: *Physics of Fluids* 24 (1981), pp. 1217–1223.
- [72] H. Rusche. *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions*. PH.D.-thesis, Department of Mechanical Engineering, Imperial College of Science, London. (Accessed: 07/30/2016). 2002.
- [73] G. Russo and P. Smereka. “A Remark on Computing Distance Functions”. In: *Journal of Computational Physics* 163 (1) (2000), pp. 51–67.
- [74] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003.
- [75] R. Scardovelli and S. Zaleski. “Direct numerical simulation of free-surface and interfacial flow”. In: *Annual Review of Fluid Mechanics* 31 (1999), pp. 567–603.

- [76] J.A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2005.
- [77] J.A. Sethian and A. Vladimirsky. “A fast marching level set method for monotonically advancing fronts”. In: *Proceedings of the National Academy of Sciences of the United States of America* 93 (4) (1996), pp. 1591–1595.
- [78] J.A. Sethian and A. Vladimirsky. “Fast Methods for the Eikonal and Related Hamilton-Jacobi Equations on Unstructured Meshes”. In: *Proceedings of the National Academy of Sciences of the United States of America* 97 (11) (2000), pp. 5699–5703.
- [79] C.-W. Shu. *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*. Institute for Computer Applications in Science and Engineering, NASA/CR-97-206253, ICASE Report No. 97-65. (Accessed: 07/30/2016). 1997.
- [80] M. Sussman and E. Fatemi. “An Efficient, Interface Preserving Level Set Re-Distancing Algorithm And Its Application To Interfacial Incompressible Fluid Flow”. In: *SIAM Journal on Scientific Computing* 20 (4) (1999), pp. 1165–1191.
- [81] M. Sussman and E.G. Puckett. “A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows”. In: *Journal of Computational Physics* 162 (2) (2000), pp. 301–337.
- [82] M. Sussman, P. Smereka, and S. Osher. “A level set approach for computing solutions to incompressible two-phase flow”. In: *Journal of Computational Physics* 114 (1) (1994), pp. 146–159.
- [83] P.K. Sweby. “High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws”. In: *SIAM Journal on Numerical Analysis* 21 (5) (1984), pp. 995–1011.

- [84] Y. Takakura and E.F. Toro. “Arbitrarily accurate non-oscillatory schemes for a nonlinear conservation law”. In: *Comput. Fluid Dyn. J.* 11 (1) (2002), pp. 7–18.
- [85] V.A. Titarev and E.F. Toro. “ADER schemes for three-dimensional non-linear hyperbolic systems”. In: *Journal of Computational Physics* 204 (2) (2005), pp. 715–736.
- [86] V.A. Titarev and E.F. Toro. “Finite-volume WENO schemes for three-dimensional conservation laws”. In: *Journal of Computational Physics* 201 (1) (2004), pp. 238–260.
- [87] V.A. Titarev and E.F. Toro. “WENO schemes based on upwind and centred TVD fluxes”. In: *Computers & Fluids* 34 (2005), pp. 705–720.
- [88] E.F. Toro. “MUSTA: A multi-stage numerical flux”. In: *Applied Numerical Mathematics* 56 (10-11) (2006), pp. 1464–1479.
- [89] E.F. Toro. *On Glimm-related schemes for conservation laws*. Technical Report MMU-9602, Department of Mathematics and Physics, Manchester Metropolitan University, UK. (Accessed: 07/30/2016). 1996.
- [90] E.F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer Verlag, 2009.
- [91] G. Tryggvason et al. “A Front-Tracking Method for the Computations of Multiphase Flow”. In: *Journal of Computational Physics* 169 (2) (2001), pp. 708–759.
- [92] P. Tsoutsanis, A.F. Antoniadis, and D. Drikakis. “WENO schemes on arbitrary unstructured meshes for laminar, transitional and turbulent flows”. In: *Journal of Computational Physics* 256 (1) (2014), pp. 254–276.
- [93] H.K Versteeg and W. Malalasekera. *Computational Fluid Dynamics - The Finite Volume Method*. Pearson Education Limited, 2007.

- [94] V. Vukčević and H. Jasak. *Interface Jump Conditions: Eliminating Spurious Velocities in Free Surface Flows*. 10th OpenFOAM Workshop 2015, University of Michigan. URL: <http://ukri-openfoam.ex.ac.uk/docs/Tuesday/posters/NavalHydro/jumpConditions-OFW10-poster.pdf> (Accessed: 07/30/2016).
- [95] V. Vukčević, H. Jasak, and Š. Malenica. “Decomposition model for naval hydrodynamic applications, Part I: Computational method”. In: *Ocean Engineering* 121 (2016), pp. 37–46.
- [96] V. Vuorinen et al. “On the implementation of low-dissipative Runge–Kutta projection methods for time dependent flows using OpenFOAM”. In: *Computers & Fluids* 93 (2014), pp. 153–163.
- [97] C. Walker and B. Müller. *Constrained Reinitialisation of the Conservative Level Set Method*. 8th International Conference on CFD in Oil & Gas, Metallurgical and Process Industries, Trondheim. (Accessed: 07/30/2016). 2011.
- [98] Z.J. Wang. *Adaptive high-order methods in computational fluid dynamics, Vol.2*. World Scientific Publishing, 2011.
- [99] R.F. Warming and R.M. Beam. “Upwind Second-Order Difference Schemes and Applications in Aerodynamic Flows”. In: *AIAA Journal* 14 (9) (1976), pp. 1241–1249.
- [100] H. Weller and H.G. Weller. “A high-order arbitrarily unstructured finite-volume model of the global atmosphere: tests solving the shallow-water equations”. In: *Int. J. Numer. Meth. Fluids* 56 (8) (2008), pp. 1589–1596.
- [101] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer Verlag, 1991.
- [102] X. Wu, J. Liang, and Y. Zhao. “A new smoothness indicator for third-order WENO scheme”. In: *Int. J. Numer. Meth. Fluids* 81 (7) (2016), pp. 451–459.

- [103] J. Yang. *Validation Cases - 2D Dam-Break Problem*. URL: <http://user.engr.iowa.edu/~yng/cases/dambreak/dambreak.html> (Accessed: 08/10/2016).
- [104] W. Yue, C.-L. Lin, and V.C. Patel. *Numerical investigations of turbulent free surface flows using level set method and large eddy simulation*. IHHR Report, No. 435. (Accessed: 07/30/2016). 2003.
- [105] S.T. Zalesak. “Fully Multidimensional Flux-Corrected Algorithms for Fluids”. In: *Journal of Computational Physics* 31 (3) (1979), pp. 335–362.
- [106] X. Zhang and C.-W. Shu. “Linear instability of the fifth-order WENO method”. In: *SIAM Journal on Numerical Analysis* 45 (5) (2007), pp. 1871–1901.
- [107] X. Zhang and C.-W. Shu. “On maximum-principle-satisfying high order schemes for scalar conservation laws”. In: *Journal of Computational Physics* 229 (9) (2010), pp. 3091–3120.
- [108] L. Zhao et al. “Improved conservative level set method”. In: *Int. J. Numer. Meth. Fluids* 75 (2014), pp. 575–590.

Ich versichere eidesstattlich durch eigenhändige Unterschrift, dass ich die Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, habe ich als solche kenntlich gemacht. Ich weiß, dass bei Abgabe einer falschen Versicherung die Prüfung als nicht bestanden zu gelten hat.