Compressible multi-phase simulation at extreme conditions using a discontinuous Galerkin scheme

A thesis accepted by the Faculty of Aerospace Engineering and Geodesy of the Universität Stuttgart in partial fulfillment of the requirements for the degree of Doctor of Engineering Sciences (Dr.-Ing.)

by

Stefan Dominik Fechter

born in Leonberg

Main referee:	Prof. Dr. rer. nat. habil Claus-Dieter Munz
Co-referee:	Prof. Dr. Rémi Abgrall
Date of defence:	01.07.2015

Institute of Aerodynamics and Gas Dynamics
University of Stuttgart
2015

Für Birgit, Ingo, Manuel, Kurt und Sibille.

Preface

This thesis gives a résumé of my work as an academic employee at the Institute of Aerodynamics and Gas Dynamics (IAG) of the University of Stuttgart.

I want to thank my doctoral advisor Professor Dr. Claus-Dieter Munz for giving me the opportunity to be part of his research group, for the scientific freedom, and exceptional working atmosphere I found there. Another thanks goes to Professor Dr. Rémi Abgrall for being the co-referee of my thesis.

A big thanks goes to all my colleagues at the Institute, especially to all past and present members of the Numerics Research Group. In particular I want to thank Dr. Andreas Birkefeld, Thomas Bolemann, Prof. Dr. Gregor Gassner, Timon Hitz, Malte Hoffmann, Dr. Felix Jaegle and Matthias Sonntag for the fruitful discussions about the discretization and modeling of multi-phase flows.

The work was financially supported by the German Research Foundation DFG in the scope of the collaborative research council SFB-TRR 75 ("droplet processes at extreme ambient conditions"). This part was investigated within the sub-project A2 ("Development of numerical methods for the simulation of compressible droplet dynamical processes at extreme ambient conditions"). Especially the collaboration within the SFB-TRR75 together with Dr. Veronika Schleper (IANS), Christoph Zeiler (IANS), Prof. Dr. Christian Rohde (IANS), Dirk Dietzel (ITV), Prof. Dr. Andreas Kronenburg (ITV), Prof. Dr. Joachim Groß (ITT), Christoph Kink (ITT) was appreciated. Special thanks go to Dr. Grazia Lamanna (ITLR), Florian Weckenmann (ITLR) and Benjamin Bork (EKT) for the discussions of the numerical results as seen from an experimental point of view.

Last but not least, I want to express my gratitude to everyone who supported me during the last years, especially my family.

Stuttgart, October 2015

Stefan Fechter

Contents

Symbols and Abbreviations

Latin symbols

a	[J/kg]	Specific Helmholtz energy
$\hat{a}$	[-]	Amplitude
A, B	[-]	Parameters for cubic EOS
c	[m/s]	Sound speed
c_v, c_p	[J/(kg K)]	Heat capacity at constant volume/pressure
d	[-]	Dimension of the problem
D	[m^2/s]	Diffusion coefficient
e	[J/kg]	Specific total energy
E	[-]	Reference cube element
f	[-, 1/s]	Flux function or frequency
F	[-]	Flux vector of the Navier-Stokes equations
$\mathcal{F}$	[-]	Flux vector in the reference element
g	[J/kg]	Specific Gibbs free energy
G	[-]	Flux vector of the level-set equations
h	[J/kg]	Specific enthalpy
H	[-]	Numerical Hamiltonian
i	[J/kg]	Kinetic energy
i, j, k, l, m, n	[-]	Summation indices
I	[-]	Unit matrix
$\mathbb{I}$	[-]	Interpolation operator
j	[kg/s]	Mass flux rate
J	[-]	Jacobian of the mapping
$J\mathbf{a}$	[-]	Contravariant basis vectors of the mapping
K	[m/s]	Mass transfer coefficient
L	[m]	Reference length
Ł	[-]	Lax curves
m	[kg]	Mass
$\dot{m}$	[kg/s]	Mass transfer rate at interface

M	[-]	Mach number
$\boldsymbol{n}_{\boldsymbol{\zeta}}$	[-]	Unit normal vector in reference element
N	[-]	Normal vector in physical element
p	[N/m^2]	Fluid pressure
Δp_σ	[Pa]	Surface tension pressure jump
q	[J/(kg m), -]	Heat flux or general tabulated quantity
Q	[-]	Grid cell
r	[-, m]	Eigenvector of the Euler system matrix or radius
R	[J/(kg K)]	Specific gas constant
s	[J/(kg K)]	Specific entropy
$\hat{s}$	[-]	Surface element
s_{PB}	[m/s]	Level-set advection velocity
s_{ext}	[m/s]	Volume extended level-set advection velocity
$\mathbf{S}$	[-]	Source term due to phase transfer modeling
S	[m/s]	Wave speeds in the Riemann problem
T	[K]	Fluid temperature
$\mathbf{U}$	[-]	Vector of the conservation variables
v, v_n	[m/s]	Velocity in normal direction
$\mathbf{v} = (v_1, v_2, v_3)^{\text{T}}$	[m/s]	Velocity vector
$\mathbf{x} = (x, y, z)^{\text{T}}$	[m]	Spatial coordinates
X_v	[-]	Vapor mass fraction
Z	[-]	Compressibility factor

Greek symbols

α	[-]	Gas mass/volume fraction
γ	[-]	Adiabatic coefficient
δ	[-]	Kronecker delta or reduced critical density $= \rho/\rho_c$
Δ	[-]	Difference
ϵ	[J/kg]	Specific inner energy
$\boldsymbol{\zeta} = (\zeta^1, \zeta^2, \zeta^3)$	[-]	Coordinates in reference element
κ	[1/m]	Interface curvature
λ	[W/(kg K)]	Heat transfer coefficient
λ_{vap}	[-]	Vaporization constant

μ	[J/kg, kg/(m s)]	Chemical potential or fluid viscosity
π	[-]	Mathematical constant
ρ	[kg/m^3]	Density
σ	[Pa s]	Surface tension coefficient
τ	[-]	Shear stress tensor, time scale for pseudo time-stepping or reduced inverse temperature $= {}^{T_c}/_{T}$
ϕ	[-]	Polynomial ansatz functions or Helmholtz energy functional
φ	[-]	Basis functions
Φ	[m]	Cropped level-set function
$\bar{\Phi}$	[m]	Distance function or reconstructed level-set polynomial
Ψ	[-]	Test function
ω	[1/s, -]	Angular frequency or Gauss integration weights
Ω	[-]	Computational domain

Subscripts

$(\cdot)_0$	Initial solution
$(\cdot)_{\text{int}}$	Quantity at the interface
$(\cdot)_\infty$	Freestream value of a quantity
$(\cdot)_c$	Value at the critical point
$(\cdot)_{\text{cond}}$	Condensation parameter
$(\cdot)_{\text{evap}}$	Evaporation parameter
$(\cdot)_{\text{liq}}$	Value on the liquid side of the saturation curve
$(\cdot)_L$	Left state of the Riemann solver
$(\cdot)_{\text{LS}}$	Level-set quantity
$(\cdot)_{\text{PB}}$	Value at the phase boundary
$(\cdot)_R$	Right state of the Riemann solver
$(\cdot)_t$	Time derivative
$(\cdot)_{tp}$	Value at the triple point
$(\cdot)_{\text{tait}}$	EOS parameter for the Tait equation of state
$(\cdot)_{\text{vap}}$	Value on the vapor side of the saturation curve

Superscripts

$(\cdot)^-$, $(\cdot)^+$	Left/Right state
$(\cdot)^*$	Material boundary in the Riemann wave fan
$(\cdot)^\sharp$	Phase boundary in the Riemann wave fan
$(\cdot)^0$	Initial solution
$(\cdot)^1, (\cdot)^2, (\cdot)^3$	Component of contravariant quantity
$(\hat{\cdot})$	Polynomial coefficients
$(\tilde{\cdot})$	Value in reference element
$(\dot{\cdot})$	Time derivative
$(\cdot)^T$	Transposed vector

Abbreviations

BR	Bassi-Rebay
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy condition
CJ	Chapman-Jouguet point
cond	Condensation
CSF	Continuous Surface Force
DFL	Diffusion time step constraint
DFT	Density Functional Theory
DG	Discontinuous Galerkin
DGSEM	Discontinuous Galerkin Spectral Element Method
DNS	Direct Numerical Simulation
DOF	Degree of Freedom
ENO	Essentially Non-Oscillatory
EOS	Equation of State
ERS	Exact Riemann solver
evap	Evaporation
FE	Finite Element
FS3D	Free Surface 3D (numerical code)
FV	Finite Volume
GFM	Ghost Fluid Method
HLL	Harten-Lax-van Leer Riemann solver
HLRS	Hochleistungsrechenzentrum Stuttgart
IAG	Institute for Aerodynamics and Gas dynamics

IANS	Institute for Applied Analysis and Numerical Simulation
ITLR	Institute of Aerospace Thermodynamics
LDG	Local discontinuous Galerkin
LF	Lax-Friedrichs flux
LS	Level-Set
LES	Large Eddy simulation
LRS	Linearized Riemann solver
MI	Material interface
NIST	National Institute of Standards and Technology
PB	Phase boundary
PCP-SAFT	Perturbed Chain Statistical Associating Fluid Theory
PDE	Partial Differential Equation
PLIC	Piece-wise Linear Interface Reconstruction
PT	Phase Transition
RK	Runge-Kutta
TVD	Total Variation Diminishing
VOF	Volume of Fluid
WENO	Weighted Essentially Non-Oscillatory

Kurzfassung

Die Simulation von Mehrphasenströmungen bei extremen Umgebungsbedingungen stellt hohe Anforderungen an die numerische Beschreibung und das zugrundeliegende Verfahren. Dies ist einerseits durch die kompressible Behandlung beider Phasen und der zugehörigen numerischen Methoden bedingt, besonders im Hinblick auf die Auflösung der Phasengrenze. Andererseits ist die Auswertung exakter Zustandsgleichungen, die das Fluidverhalten in der Nähe des kritischen Punkt genau beschreiben, aufwändig. Als zusätzliche Schwierigkeit sind Hydrodynamik und Thermodynamik über die Strömungsgleichungen und die Zustandsgleichung eng miteinander gekoppelt, sodass ein thermodynamisch konsistenter numerischer Ansatz gewählt werden muss.

Die numerische Bechreibung von kompressiblen Mehrphasenströmungen beinhaltet die numerische Behandlung von Phänomenen, die in bzw. zwischen verschiedenen Aggregatszuständen vorkommen. Dabei ist die Modellierung dieser Vorgänge an der Phasengrenzfläche, wie beispielsweise Phasenübergang und Oberflächenspannung, von zentraler Bedeutung.

Im Rahmen dieser Arbeit liegt dabei der Fokus auf dem Mehrphasensystem flüssig-gasförmig und dem Fluidverhalten bei extremen Bedingungen, nahe des kritischen Punktes. Unter extremen Bedingungen werden hier hohe Temperaturen und/oder hohe Drücke verstanden, bei denen die Kompressibilität beider Phasen nicht mehr vernachlässigt werden kann. Damit ist auch der häufig verwendete inkompressible Ansatz zur Beschreibung des Mehrphasengebietes nicht mehr ausreichend. Dies ist der Ansatzpunkt der Dissertation, die beschreibt, wie man kompressible Mehrphasenströmungen numerisch beschreiben kann. Der Fokus liegt dabei auf der thermodynamisch konsistenten Modellierung der Prozesse an der Phasengrenze und deren Auswirkungen auf die Strömungen im Tropfen und in der Gasphase.

Für die Simulation von kompressiblen Mehrphasenströmungen wird dabei auf eine hetereogene Mehrskalenmethode zurückgegriffen, mit derer Hilfe es möglich ist, die verschiedenen Skalen im Strömungsgebiet aufzulösen. Dabei treten an der Phasengrenzfläche kleinere Skalen auf, als in den beiden Phasen. Diese Me-

thode bedingt die folgenden Bausteine für das zugrundeliegende numerische Verfahren:

- Berechnungsverfahren zur Auflösung der Skalen in der flüssigen und gasförmigen Phase, das hier als Discontinuous Galerkin (DG) Verfahren hoher Ordnung umgesetzt ist.
- Lokale h-Verfeinerung des Gitters an der Phasengrenze um die Auflösung der Grenzfläche zu verbessern. Dies ist durch ein Finite-Volumen Verfahren in der DG Zelle umgesetzt.
- Berechnung einer lokalen Lösung an der Phasengrenzfläche, die die Oberflächeneffekte wie Oberflächenspannung und Phasenübergang mit auflöst. Die Phasengrenze wird dabei als Unstetigkeit im Strömungsgebiet behandelt.
- Die Phasengrenze wird mit Hilfe eines Level-Set Verfahrens verfolgt. Dieser Ansatz ermöglicht eine genaue Berechnung der Geometrie der Phasengrenze, die für die Berücksichtigung von Oberflächenspannungseffekten notwendig ist.
- Allgemeiner Ansatz zur Approximation der thermodynamischen Zustandsgleichung im numerischen Algorithmus für hochgenaue Zustandsgleichungen, die auch am kritischen Punkt gültig sind.

Diese hetereogene Mehrskalenmethode wurde an generischen Testfällen validiert. Eine Erweiterung hin zu realen Einsatzbedingungen in den Experimenten am ITLR ist Gegenstand der aktuellen Entwicklung.

Abstract

The simulation of compressible multi-phase flows at extreme ambient conditions imposes high demands on the numerical treatment as well as the numerical method. On the one hand, due to the compressible treatment of both fluid phases and their corresponding numerical methods, especially regarding the numerical resolution of the phase interface. On the other hand, the evaluation of equation of states (EOS), that are valid in the vicinity of the critical point, is expensive. As additional challenge are hydrodynamics and thermodynamics coupled closely by the compressible flow equations. This implies that a thermodynamically consistent numerical method has to be chosen.

The numerical description of compressible multi-phase flows implies the numerical treatment of phenomena that occur at or between different phases. The modeling and resolution of these processes at the phase interface is of central importance. These effects include e. g. surface tension and phase transfer.

The focus of this thesis is the multi-phase system liquid-gas and the fluid behavior at extreme ambient conditions. The term "extreme conditions" implies here high temperatures and/or high pressures at which the compressibility of both phases can not be neglected. Thus the commonly used incompressible method is not sufficient any more. This is the entry point of my dissertation that describes how to treat compressible multi-phase flows numerically in a consistent way. The focus is hereby on the thermodynamic consistent modeling of the processes at the phase interface and their effects on the fluid flow inside and outside of the droplet.

For the simulation of compressible multi-phase flows a heterogeneous multiscale method is used. This method allows to resolve the dissimilar scales within the flow field. At the phase interface smaller scales are present compared to the bulk phases. This implies that numerically the following building blocks are needed:

- A numerical scheme for the resolution of the scales in the bulk phases. Here, a Discontinuous Galerkin method is used.

- Local h-refinement at the phase interface to provide a sufficient resolution of the phase interface. This is done using a finite-volume sub-cell method inside the DG cell.
- Calculation of the local solution at the phase interface that takes interface effects like surface tension and phase transfer into account. The phase interface is resolved in a sharp interface manner within the flow field.
- The tracking of the phase interface is done using a level-set method. This approach allows for an accurate estimation of the interface geometry that is needed for the inclusion of surface tension effects.
- General approach for the approximation of the thermodynamic equation of states in the numerical algorithm that is suitable for high-accurate equation of states that are valid at the critical point.

This heterogeneous multi-scale method has been validated at generic test cases. The extension to realistic operating conditions found in experiments at ITLR is subject of the recent development.

1 Introduction

In many technical applications, multi-phase flows meet conditions such as high pressure environments and/or high velocities that prohibit the popular assumption of an incompressible flow field. Important examples for such extreme ambient conditions include fuel injection systems of aeronautical, automotive and rocket engines that are used at high-pressure and high temperature operating conditions. At these extreme ambient conditions, physical effects can be observed that are yet not totally understood. This is essentially true for operating conditions near or at the critical point. In this region, strong gradients are observed in the EOS description. One application of such near-critical conditions is the high-pressure diesel injection. A rough estimation can be gained assuming an incompressible multi-phase flow regime that does not account for the high-pressure effects. This is due to the neglect of compressibility effects in the bulk phases. In this work, the building blocks for the numerical simulation of compressible multi-phase flows, that include physical effects like surface tension and phase transition, are described. Of major interest are the trans-critical processes that occur during the transition from a sub- to a supercritical state or vice versa. For example, a fuel droplet is injected into the combustion chamber at a subcritical state which is then heated up to a supercritical one.

The term multi-phase implies that at least two different phases are present within the computational domain. In the following, the investigation is restricted to the vapor-liquid system with large density ratios in the order of 1000. This means that liquid and vapor phases are coexisting in the domain, e. g. a liquid droplet within a gaseous environment or vice versa. Due to the transition to a supercritical state, the phase boundary may not always be present. In a supercritical state only a diffused interface can be investigated. The compressible simulation implies that all waves within the bulk phases are resolved, also acoustic waves that play a major role in the onset of cavitation. Both phases are treated compressible implying that the density of the liquid and vapor phase is variable. Due to the stiff equation of state (EOS) in the liquid phase, a small density change implies a large pressure jump as it is (nearly)

incompressible. The compressibility of the liquid phase increases towards critical conditions and for supercritical states the interface smears out. Within the vapor phase compressibility effects play a major role. This behavior reflects the dissimilar material behavior of two-phase flows having one stiff liquid phase and a compressible vapor phase at the same time. One important part is the resolution of interface effects at the phase interface. There, physical effects, like phase transition and surface tension, have to be resolved that solely occur there.

The numerical simulation at high pressure and/or high temperature operation points aims to supply the understanding of the related physical process as this regime is accessible only with high-pressure experiments [103]. In these experiments a small portion of the experiment is visible and not all fluid variables can be measured directly. Typically, a two-dimensional section of the droplet is measured by means of shadowgraphy or laser-induced fluorescence. Thereby, information about the droplet behavior is obtained using high-speed cameras. Based on the droplet shape distribution, integral evaporation rates can be estimated. Advanced measurement techniques are employed to measure the mixture state of the droplet based on the Raman spectroscopy providing information about the mixture state inside and at the surface of the evaporating droplet.

To be able to develop a numerical method that is suitable for the investigation of the transcritical processes, a compressible multi-phase method is needed. The compressible treatment of two-phase flows introduces additional difficulties compared to the incompressible treatment that is typically found in multi-phase solvers in practical use today. All conservation equations are coupled via the equation of state (EOS) and have to be solved simultaneously. The decomposition and separate solution of hydrodynamics and thermodynamics as in the incompressible case is no longer possible. Hence, the numerical simulation of compressible two-phase flow introduces additional difficulties to the numerical approach, mainly due to the need of a consistent numerical and thermodynamical approximation. Another difficulty is that established modeling methods in the incompressible context have to be adapted or new methods have to be developed to account for processes at the phase interface as surface tension and phase transition.

In the context of compressible two-phase flows, two elements of the numerical algorithm are crucial: The first one is a method that allows to define the interface geometry and its temporal evolution. The second is a numerical strat-

egy to treat the discontinuous nature of the interface together with the related physical interface processes such as surface tension or phase change. In the compressible case considered here this may include different tabulated EOS on both sides of the phase interface that reveal the dissimilar behavior of the liquid and gaseous phase. The focus of this investigation is on a single fluid, i. e. a liquid and its vapor. Mixture effects of different species are not considered in this investigation but may be included in the general numerical method.
In the context of compressible multi-phase flows, a sharp interface method is developed that consists of the following five building blocks:

1. Macro-scale solver based on the compressible Navier-Stokes equations for the resolution of the bulk phases (see section 1.1).

2. Consistent thermodynamic approximation of the phase interface in a sharp or diffuse interface manner.

3. Micro-scale model at the interface position to resolve locally effects at the phase interface such as surface tension and phase transition (see section 1.2).

4. High-accurate interface tracking method that allows for an estimate of the interface curvature as well as the interface normals (see section 1.3).

5. Suitable thermodynamic equation of state that describes fluid behavior at extreme ambient conditions combined with a method that allows for an efficient evaluation (see section 1.4).

These components are visualized in figure 1.1. In the following the main components are introduced in more detail. A complete description of the building blocks can be found in the sections of the thesis later on.

1.1 Discontinuous Galerkin method

The macro-scale model is based on a discontinuous Galerkin spectral element method (DGSEM) [48, 41] that enables high order of accuracy as well as efficient calculations. The discontinuous Galerkin (DG) formulation, similar to the finite-volume method, allows discontinuities at the element interfaces while having a continuous in-cell resolution. This simplifies the implementation of the sharp interface approach. Solely the numerical fluxes at the position of the

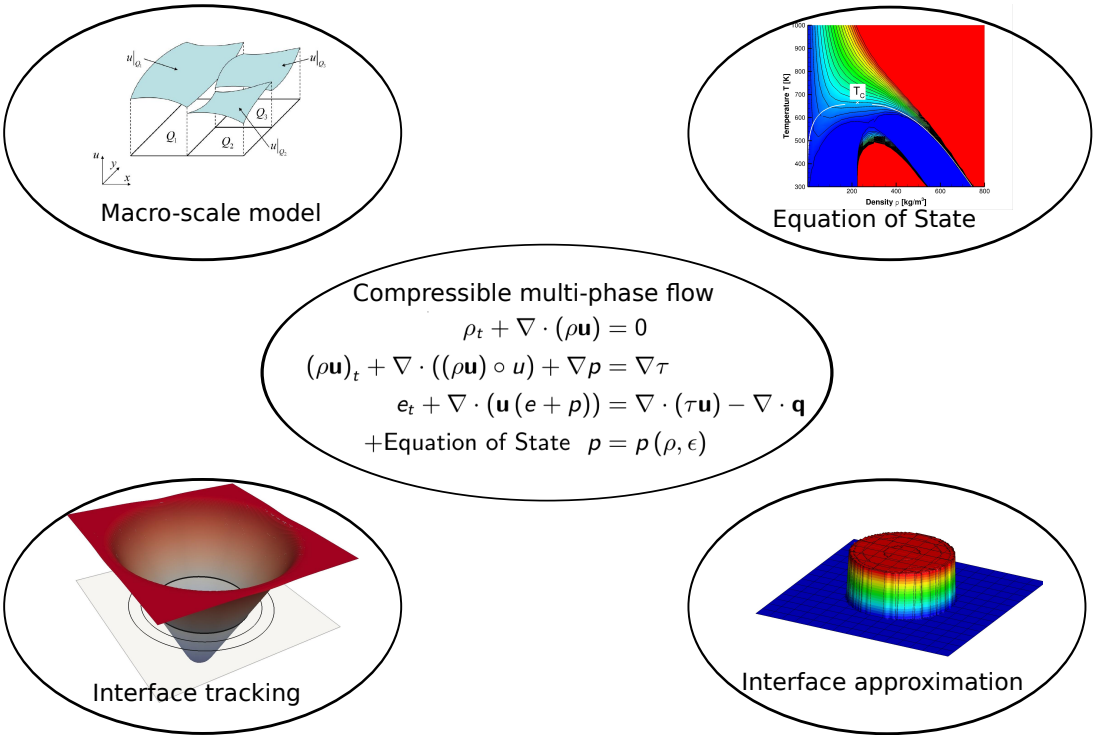


Figure 1.1: Overview of the main components for the simulation of compressible multi-phase flows.

tracked interface have to be adapted. The main advantage is the excellent scalability on supercomputers combined with the high approximation order. Due to the large and complex problem sizes in multi-phase flows, high-performance computing is an essential building block.
The application of high-order discontinuous Galerkin methods is not restricted to multi-phase flows. DG methods are typically applied to the simulation of single-fluid problems within Computational Fluid Dynamics (CFD). The DG methods can be applied to several kind of problems including

- Direct numerical simulation of laminar and turbulent flows.
- Large eddy simulations (LES) of turbulent flows [7] based on implicit LES models.
- Direct acoustic simulation of flow noise [25, 31].
- Maxwell-equations with particle in cell methods for the simulation of space applications [65, 66, 87].

- Magnetohydrodynamics equations (coupled Navier-Stokes equations with magnetic field) [3] for research in fusion reactors and astrophysics.

These key applications reflect the work done within the research group of Prof. Munz at the Institute of Aerodynamics and Gas Dynamics.

1.2 Approximation of the interface

The approximation of the phase interface in the computational domain is needed for the resolution of the phase interface that numerically represents a discontinuity. Additionally, physical effects like surface tension and phase transition may occur at the interface that have to be resolved locally and may introduce additional surface forces and/or source terms.
In literature, two main interface coupling methods are available. The first, the diffuse interface approach, is characterized by a smooth transition region between the bulk phases resulting in a finite interface thickness. Typical representatives are Baer-Nunziato and Navier-Stokes-Korteweg models, see e. g. [81], [4]. The smoothing of the interface is provided by the numerical scheme or explicitly introduced by a phase indicator to allow a stable resolution by the numerical approximation. The challenge is to derive a thermodynamically consistent smoothening of the interface that does not violate the basic thermodynamic laws and provides a realistic interface width. Furthermore, a fine resolution of the interface region is needed to provide a realistic approximation of the interface thickness. The second coupling method is the sharp interface representation that was introduced by Fedkiw et al. [30] in the context of finite-volume methods. This approach avoids any smoothening at the interface, which is a challenge for the coupling of the flow solver in the bulk phases and the interface treatment. The compressible flow solver resolves the macroscopic scales of the flow and necessarily allows discontinuous states at an accurate interface position. Appropriate jump conditions at the interface as well as the interface position are mandatory for sharp-interface models. The coupling of the macroscopic flow solver and the interface tracking must establish the sharp interface treatment. Besides the use of a moving grid (see e. g. [67]) or the introduction of cutted cells, the so called ghost fluid approach has been introduced by Fedkiw et al. [30] for finite volume schemes. This formalism has been extended by Merkle and Rohde [62] and by Rohde and Dressel [18] for piecewise constant approximations in one space dimension and in the multi-dimensional case, re-

spectively. Here, fictitious fluid states are introduced near the interface, similar to the well-known concept of ghost-cells to define boundary conditions. A third method is the molecular dynamics simulation of the fluid interfaces. Hereby, the movement of the single molecules are tracked in time at the interface to investigate the physical effects at the interface. Due to the huge computational effort of this method only small droplets can be resolved in the order of some nanometers. Because of that, this method is typically used for the fundamental research of the transport processes. For larger scale computations, models are derived based on the findings.
In the following, the sharp interface approach based on the ideas of the ghost-fluid method is considered, but without the explicit introduction of ghost cells at the interface. The different phases are considered explicitly in the local solution at the interface and different numerical fluxes are applied for the vapor and liquid phases, respectively. Information about the interface physics and the interface velocity is provided by an approximate local solution, that is estimated by a micro-scale solver. In case phase transition is present, this solvers takes information from the micro-scale into account to provide a unique solution. For all investigated test cases, linear Riemann-type solvers [62, 44, 26] are used in cases without phase transition. In cases where phase transition is considered a special Riemann interface solution is considered [28, 79] that resolves the hyperbolic-elliptic problem. The mass flux over the interface is considered by means of a kinetic relation that ensures a thermodynamic consistent approximation. Both interface Riemann solver approaches are applicable to general EOS.
For reference, a simple diffuse interface model, together with an accurate description of the underlying EOS, is shown. This approach smears the interface and employs a special approximation for the EOS in the multi-phase region to obtain a purely hyperbolic formulation at the interface in case phase transfer effects are considered. This special treatment is necessary to avoid the elliptic region inside the multi-phase region that can not be resolved using a standard hyperbolic flow solver. The approach used here is compared qualitatively to models based on the Baer-Nunziato mixture model that account for phase transfer effects based on a thermo-chemical relaxation solver, as e. g. developed by Saurel et al. [81] and Zein et al. [108].

1.3 Interface tracking

The use of a sharp interface approximation method together with the inclusion of surface tension effects necessitates the use of an interface tracking method.

This method adds an additional conservation equation to the equation system that has to be solved. A more detailed introduction to the here used level-set method is given in chapter 4.
Several methods for an explicit tracking of the interface are available: The first is the volume of fluids (VOF) method as introduced by Hirt and Nichols [42]. Here, the volume fraction is approximated as an additional variable in each grid cell. The volume fraction is allowed to jump between each grid cell and the interface is given by a certain value. Due to the approximation of the volume fraction, the interface geometry has to be reconstructed using a piecewise linear interface reconstruction (PLIC) algorithm. The second method, called the front tracking method, is based on the inclusion of marker particles that move with the fluid velocity for the interface representation as described by Unverdi and Tryggvason [98]. The third algorithm is the level-set method as introduced by Osher and Sethian [69] and further refined by Sussman and coworkers [89] for the simulation of incompressible multi-phase flow. The interface geometry is approximated as zero iso-contour of the continuous level-set function that represents the distance to the interface.
In the scope of this work, the level-set interface representation is chosen as the continuous level-set function is easily discretizable using the high-order discontinuous Galerkin method as well as extensible to compressible flow (with non-zero velocity divergence). The high order discretization is favorable for the estimation of interface curvature. The approximation of the curvature would be much more challenging in case a VOF interface capturing method is used as the PLIC reconstructed interface has to be used as basis for the curvature calculation.

1.4 Consistent approximation of the equation of state

One crucial part is the approximation of the underlying equation of state (EOS) in the algorithm. In contrast to the incompressible treatment, an EOS is mandatory in the compressible description as it couples all conservation equations. The challenge is that the EOS has to describe the dissimilar behavior of multi-phase flows correctly. This implies the resolution of the (weakly) compressible liquid part, that is rather stiff, as well as the compressible gaseous part.

Several EOS are discussed in literature and each description has advantages and disadvantages. For the resolution of phase transfer effects inside the multi-phase region at least a cubic equation of state is needed. Another possibility is to relate the two describing EOS to each other such that the enthalpy jump at the phase interface is reproduced. This can be done by introducing offsets in the stiffened gas EOS as described by LeMétayer et al. [55].
In case the phase transfer effects are modeled or not considered otherwise, simpler EOS descriptions are possible. Several possible approximations are listed here:

1. A rather popular assumption is to use the *stiffened gas equation*, the extension of the ideal gas EOS to multi-phase flows. This EOS has low computational costs as it is analytically solvable for the temperature and pressure. The stiffness for the liquid part is introduced by a stiffness parameter in the pressure law (that is zero in the gaseous region). Formally two EOS are used to the gaseous and liquid part. LeMétayer et al. [55] showed how to regress the stiffened gas parameters to realistic fluid behavior within a certain temperature range and how to introduce offsets for the inner energy to relate the liquid and vapor phases. This fitting of the enthalpy and entropy is needed to be able to reconstruct a physically meaningful connection between two states in different phases in case phase transfer effects are considered. However, this approach is not suitable for the description in the vicinity of the critical point or at high pressure conditions.

2. A first improvement is achieved by using a cubic EOS, such as the van der Waals EOS [100] or Peng-Robinson EOS [73]. Many other variants are possible that have modifications in the description terms (for a more complete overview see Poling et al. [75]). The van der Waals-EOS can be solved analytically while for the Peng-Robinson EOS an iterative method is necessary due to the non-linear temperature term in the pressure law. All these descriptions have in common that the multi-phase region is described by an oscillating characteristic. This characteristic introduces the spinodal region in which a non-defined sound speed can be found. This type of EOS has the advantage of being relatively simple as well as being able to describe fluid behavior quite accurately. A further advantage is that fluid mixtures can be treated easily by means of mixture dependent coefficients in the pressure law.

3. To be able to increase the accuracy of the describing EOS, Setzmann and Wagner [85, 84] introduced a method to regress the EOS behavior up to a given accuracy that is typically less than 1% within a certain temperature and pressure range. These regressed EOS descriptions include up to 53 terms (for water) to be able to accurately capture the fluid behavior. The residual part of the Helmholtz energy functional is parametrized and from this, the EOS behavior can be calculated. Parameters for several fluids are published in literature and various calculation tools, e. g. CoolProp [8] and Refprop [56], are available that use these correlations. These polynominals reflect the correct behavior at the critical point and are sufficiently accurate to describe fluids at extreme ambient conditions. The main drawback of this method is high computational costs for the EOS evaluation.

4. The perturbed-chain polar statistical associating fluid theory (PCP-SAFT) EOS is e. g. described by Gross in [35] and by Johannessen et al. in [46] for pure fluids. The extension to fluid mixtures is described by Tang and Gross in [92]. This method provides an approximation of the residual Helmholtz energy functional from which the basic thermodynamic quantities can be derived by differentiation. However, the derivation history is completely different. In contrast to the method of Wagner the EOS behavior is approximated by means of the molecular composition of the considered fluid. This approach provides a rather general method to describe fluids that is located in-between the molecular simulations and the continuum models

For a more complete list of different EOS see the book of Poling et al. [75]. Using the tabulation approach presented in section 5.3 it is possible to reduce the overall computation time by a factor of 1000 without reducing the accuracy of the underlying EOS. This approach is not limited to a specific EOS.

1.5 Overview of the thesis

This thesis covers the description of a novel numerical method to simulate compressible multi-phase flows. Hereby, the flow in the bulk phases as well as the interface tracking is calculated using a discontinuous Galerkin spectral element method. This description is is advantageous concerning the dispersion and dissipation properties of the high order scheme. Furthermore, it allows

for a direct estimation of the interface curvature. The interface effects in the computational domain are resolved using a generalized Riemann solution at the interface that takes local interface effects like surface tension and phase transition into account. In case phase transition occurs the solution is non-unique and additional constraints from the micro-scale have to be used.
The structure of this dissertation is the following: In chapter 2 the numerical equations and the overall computation method for multi-phase flows is introduced. This description is followed by an introduction of the DG method and the local sub-cell refinement at the interface in chapter 3. The level-set method that is used for interface tracking is described in section 4 together with a description of the numerical method for the approximation of the curvature. Special considerations have to be taken for the choice of the EOS suitable for the description of multi-phase flows that is introduced in section 5. The solution structure of the generalized Riemann problem at the interface that resolves surface tension and phase transfer effects is detailed in section 6. In the following chapter the numerical method is applied to several droplet test cases in chapter 7 showing the capabilities of the numerical method. In the last section a short conclusion and outlook is presented.

2 Heterogeneous Multi-scale Method

In this chapter the Heterogeneous Multi-scale Method (HMM) used for the simulation of compressible multi-phase flow is described. In the first part the balance equations for the bulk flow as well as the level-set based interface tracking method are introduced. In the second part the methodology of the HMM is described in detail as well as the difficulties of compressible multi-phase flows and the coupling concept at the phase interface. The coupling is based on a sharp interface method. Local interface related effects, such as surface tension and phase transfer, are resolved by a generalized Riemann problem at the interface position.

2.1 Balance equations

2.1.1 Compressible Navier-Stokes equations

Inside the bulk phases away from the interface the compressible Navier-Stokes equations are considered. They can be formulated in the conservation form describing the conservation of mass, momentum and energy

$$\mathbf{U}_t + \nabla \cdot \mathbf{F}^A(\mathbf{U}) - \nabla \cdot \mathbf{F}^D(\mathbf{U}, \nabla \mathbf{U}) = \mathbf{S}. \tag{2.1}$$

The vector of the conservative variables $\mathbf{U}$ is defined by $\mathbf{U} = (\rho, \rho v_1, \rho v_2, \rho v_3, \rho e)^{\mathrm{T}}$ and contains the primitive variables density ρ, the velocity vector $\mathbf{v} = (v_1, v_2, v_3)^{\mathrm{T}}$ and the total specific energy e. The right hand side term $\mathbf{S}$ describes any source term in the equations. This source term can be used, for example, for the inclusion of surface tension effects or phase transfer effects.
Using the notation (2.1) one can distinguish two different flux contributions, the advection and the diffusion flux. These flux vectors can be written in flux vector notation $\mathbf{F}^A = (F_1^A, F_2^A, F_3^A)$ and $\mathbf{F}^D = (F_1^D, F_2^D, F_3^D)$ for three-dimensional flow.

The advection flux can be expressed in the following way

$$\mathbf{F}_1^A = \begin{pmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ \rho v_1 v_3 \\ v_1(\rho e + p) \end{pmatrix}, \quad \mathbf{F}_2^A = \begin{pmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + p \\ \rho v_2 v_3 \\ v_2(\rho e + p) \end{pmatrix}, \quad \mathbf{F}_3^A = \begin{pmatrix} \rho v_3 \\ \rho v_1 v_3 \\ \rho v_2 v_3 \\ \rho v_3^2 + p \\ v_3(\rho e + p) \end{pmatrix}, \quad (2.2)$$

while the diffusion flux is defined by

$$\mathbf{F}_1^D = \begin{pmatrix} 0 \\ \tau_{11} \\ \tau_{21} \\ \tau_{31} \\ \sum_j \tau_{1j} v_j - q_1 \end{pmatrix}, \quad \mathbf{F}_2^D = \begin{pmatrix} 0 \\ \tau_{12} \\ \tau_{22} \\ \tau_{32} \\ \sum_j \tau_{2j} v_j - q_2 \end{pmatrix}, \quad \mathbf{F}_3^D = \begin{pmatrix} 0 \\ \tau_{13} \\ \tau_{23} \\ \tau_{33} \\ \sum_j \tau_{3j} v_j - q_3 \end{pmatrix}. \quad (2.3)$$

The term $\underline{\underline{\tau}}$ in the diffusion flux vector is the shear tension matrix including the hypothesis of Stokes

$$\underline{\underline{\tau}} := \mu(\nabla \mathbf{v} + (\nabla \mathbf{v})^T - \frac{2}{3}(\nabla \cdot \mathbf{v})\mathbf{I}). \quad (2.4)$$

The fluid properties for the viscosity μ and the heat transfer coefficient λ are mainly a function of the fluid temperature (e. g. Sutherland viscosity correlation for gases). However, high-pressure corrections include as well density dependent terms.
For the assumption of a thermodynamically ideal gas behavior with constant coefficients, simple models can be found while for general EOS the approximation is not trivial. Typically, extensive models are used to describe the thermodynamic fluid behavior. The pressure law for a general EOS can be written as

$$p = f(\rho, T) = f(\rho, \epsilon) \quad (2.5)$$

describing the pressure evolution dependent on the thermodynamic state that may be given by the density ρ and the temperature T or by the density ρ and the inner energy ϵ. For an ideal gas approximation, this relationship can be solved analytically to

$$p = \rho R T = (\kappa - 1)\rho(e - \frac{1}{2}\mathbf{v} \cdot \mathbf{v}). \quad (2.6)$$

For general EOS this is not possible as typically a temperature iteration step is necessary to express the pressure in terms of the conservative variables density and inner energy. Note that the inner energy ϵ is related to the total energy e by $e = \epsilon + {}^1\!/\!_2\vec{v}^2$. A more detailed description of the EOS-specific terms is given in chapter 5 including a detailed description for several EOS. These issues described become even more complicated in the case when mixing of different fluids is considered.
The last term, the specific heat transfer coefficient λ is for general EOS a function of $\lambda = f(\rho, p, T)$. Using the ideal gas approximation this term can be simplified to $\lambda = \frac{c_p \mu}{Pr}$ including the heat conductivity at constant pressure c_p and the Prandtl number Pr. This leads to the definition of the heat transfer contribution in the fluid

$$q = -\lambda \nabla T \,, \tag{2.7}$$

including the temperature gradient ∇T in the flow field of the bulk phases. Approaches to approximate the spatial temperature gradient for general EOS are discussed in section 5.2.2.

2.1.2 Level-set transport equation

In the context of the HMM an additional conservation equation is solved for the interface tracking. The interface position is needed in the context of the sharp interface method used. Here, the level-set method is chosen as it is easily applicable in the context of high-order DG schemes. For the representation of the interface geometry, a signed distance function Φ is initialized whose zero iso-contour describes the position of the phase interface.
The description of the method starts from the original level-set advection equation as introduced by Sussman et al. in [89]

$$\frac{D\Phi}{Dt} = \frac{\partial \Phi}{\partial t} + \boldsymbol{s}_{\mathrm{PB}} \cdot \nabla \Phi = 0 \text{ in } \Omega \times (0, T) \,, \tag{2.8}$$

which is a conservation equation for an incompressible flow field ($\nabla(\boldsymbol{s}_{\mathrm{PB}}) = 0$) only. Hereby, Φ denotes the continuous level-set function and $\boldsymbol{s}_{\mathrm{PB}}$ the advection velocity field. The transport equation is recast as conservation equation to be able to treat the level-set variable using the DGSEM. This reformulation leads to an additional right-hand-side term

$$\frac{\partial \Phi}{\partial t} + \nabla \cdot (\boldsymbol{s}_{\mathrm{PB}} \Phi) = \Phi \nabla \cdot (\boldsymbol{s}_{\mathrm{PB}}) \,. \tag{2.9}$$

Due to the consideration of the compressible flow equations, the right-hand-side (RHS) term is non-zero and has to be resolved. The level-set equation (2.9) can be formulated in flux formulation similar to equation (2.1)

$$\Phi_t + \nabla \cdot \mathbf{G}^a(\Phi) = \Phi \cdot \nabla(\boldsymbol{s}_{\mathrm{PB}}) \,, \tag{2.10}$$

with the linear advection flux

$$G_1^a = (s_{1,\mathrm{PB}}\Phi) \,, \quad G_2^a = (s_{2,\mathrm{PB}}\Phi) \,, \quad G_3^a = (s_{3,\mathrm{PB}}\Phi) \,, \tag{2.11}$$

and the components of the level-set advection velocity $\boldsymbol{s}_{\mathrm{PB}}$. This velocity is determined in the scope of the HMM by a Riemann-based solver at the phase interface. Alternatively, the fluid velocity can be chosen in case no phase transfer effects are considered.

2.2 Building blocks of the Heterogeneous Multi-scale method

In this section, the building blocks of a multi-scale method suitable for the direct numerical simulation (DNS) of compressible multi-phase flows are described. To be able to include interface phenomena such as surface tension or phase change into the algorithm, a heterogeneous multi-scale method (HMM) is chosen. This method is able to resolve the different length scales present in a multi-phase flow. Within the bulk flow, a lower resolution is sufficient, while at the interface position locally small length scales have to be resolved together with stiff source terms due to the modeling of local interface effects.
In the following subsections the basic building blocks of the multi-phase method are described. A short overview for orientation is given by the listing of the different computation steps that are also visualized in figure 2.1:

Step 1: Computation of the interface curvature κ, whose location is given by a high-order numerical approximation of the level-set equation. To have a smooth and accurate approximation at the interface, the level-set variable is improved by an a posteriori reconstruction prior to the curvature estimation. The curvature estimation method is detailed in section 4.2.

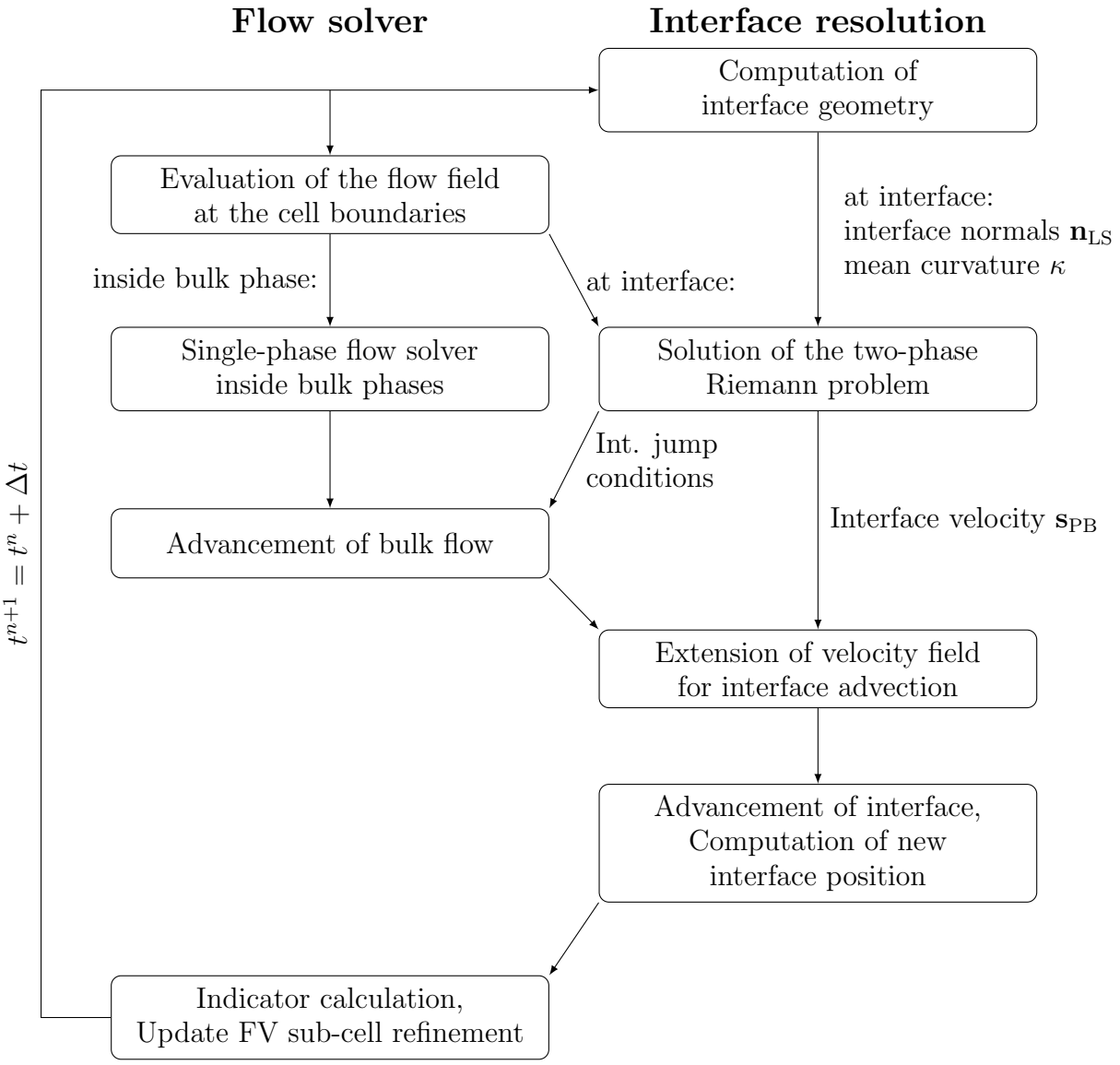


Figure 2.1: HMM algorithm used for the simulation of compressible multi-phase flows.

Step 2: At the interface position, a two-phase Riemann problem is solved, which provides information about the local normal interface velocity $\boldsymbol{s}_{\mathrm{PB}}$ as well as the states at the interface from the right- and left-hand side. Such a Riemann problem is solved at each surface integration point of the phase interface and takes surface tension and phase transfer effects directly into account. Due to these source terms, it has to be considered as a generalized Riemann problem. The initial states at the interface are provided by an extrapolation of the values from the liquid and gas bulk phases. In case phase transition occurs, additional information from the micro-scale has to be provided to ensure uniqueness of the solution and to account for the mixed hyperbolic-elliptic problem. The local solution at the interface is developed in close collaboration with IANS, University of Stuttgart.

Step 3: The explicit discontinuous Galerkin spectral element method (DGSEM) is used to advance the flow field to the next time level t^{n+1}. The flow solver is a single-phase solver for a general equation of state and is applied either in the gas or in the liquid phase. The appropriate fluxes are calculated with the states given by the interface Riemann solver from step 2. Within the bulk phases, traditional single-phase Riemann solvers are considered.

Step 4: The local normal interface speed $\boldsymbol{s}_{\mathrm{PB}}$ from step 2 at the computational interface is extended along the level-set normals $\mathbf{n}_{\mathrm{LS}}$ to obtain a velocity field for the level-set advection equation. This type of extension is advantageous as the level-set reinitialization can be used infrequently.

Step 5: The new position of the level-set zero iso-contour is used to determine the updated position of the physical and computational interfaces.

2.2.1 Macro-scale model

The Euler and Navier-Stokes equations are considered as macro-scale model in the bulk phases, together with a suitable approximation for the underlying EOS. Similar to the jumps in density and pressure, the viscosity and heat transfer coefficient may be discontinuous across the interface. Note that a sharp interface model is chosen based on the ideas of the Ghost Fluid Method (GFM) as originally developed by Fedkiw [30]. This method implies that in addition to the fluid equations the interface has to be tracked explicitly. Furthermore,

the geometry of the interface is needed for an explicit estimation of the surface tension effects while the interface advection speed is given by the solution of interface Riemann solver. This results in a close interaction between fluid solver and interface tracking.

2.2.2 Interface resolution and coupling

The solution for the macro-scale model is provided by a discontinuous Galerkin spectral element method (DGSEM) explicit solver as developed by Kopriva [48, 50, 41]. For the resolution of the interface effects at the phase interface, a micro-scale solver is chosen. This micro-scale solver may be based on different concepts such as extended classical Riemann solvers or well-resolved one-dimensional simulations. The coupling between the micro- and macro-scale model is done using numerical fluxes at the interface position.
In the present study, several multi-phase Riemann solvers at the interface are applied. Simple linear Riemann solvers as published by Fechter et al. [26] or Hu et al. [44] can be used to resolve surface tension effects at the interface position. These methods are a suitable approximation for problems without phase transfer as a classical Riemann wave fan is assumed (see figure 2.2). This allows the coupling of arbitrary EOS at the phase interface enabling even the use of different EOS for the liquid and gaseous phases respectively. In this case, the material and phase boundary are equal and, hence, a classical Riemann problem is retained. Note that the term *phase boundary* describes the position of the boundary between the gaseous and liquid phase while the term *material boundary* describes the border between different materials.
In case phase transition is present, an additional wave has to be resolved in the local solution at the interface. To be able to resolve this additional wave correctly, constraints are needed such as the second law of thermodynamics or a molecular measure of the entropy dissipated due to phase transition at the interface. The Riemann wave fan with phase transition is visualized in figure 2.2 on the right hand side, that consists of four distinct waves and five states. Note that the phase and material boundary do not coincide any more. This is due to the evaporating liquid leading to a new state between the phase and material boundary that describes the evaporated gas portion. This evaporated gaseous fluid mixes in a second step with the surrounding gas that may be of a different type (multi-component fluid).

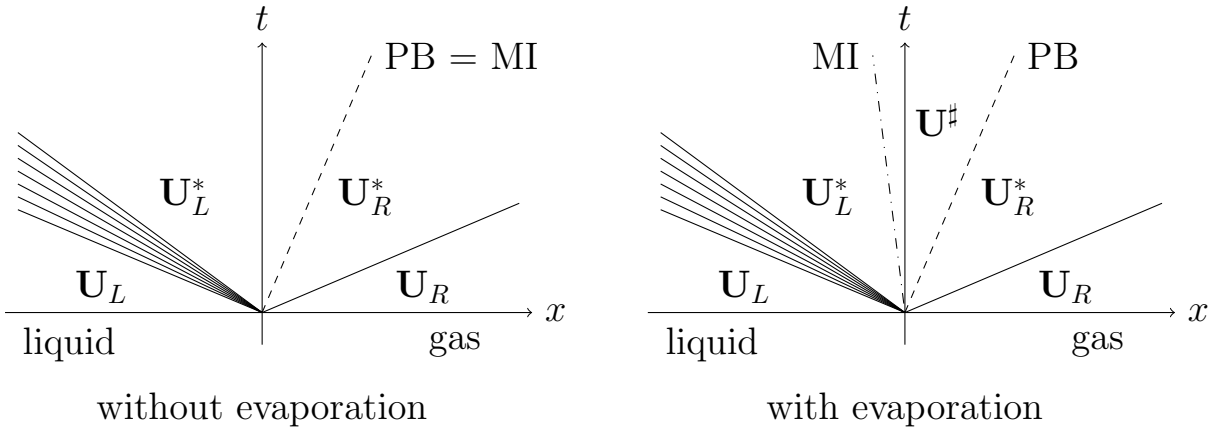


Figure 2.2: Riemann wave fan at the phase interface for cases without phase transition (left) and with phase transition (right). "MI" marks the material interface and "PB" the phase interface. Note that in case phase transition is present, the material and phase interface do not coincide any more.

From the local solution of the Riemann problem into the normal direction of the interface, two types of information are obtained: The velocity of interface propagation $\boldsymbol{s}_{\mathrm{PB}}$ and the states from both sides of the material interface ($\mathbf{U}_L^*$ and $\mathbf{U}_R^*$). The velocity $\boldsymbol{s}_{\mathrm{PB}}$ is needed for the interface tracking and is different to the local fluid velocity in case phase transition is present. The states are then used to calculate the numerical fluxes in the grid cells located at the interface. In the ghost-fluid approach two sets of data are present in an interface grid cell. One set is used to calculate the numerical flux to fluid 1 and the other the numerical flux to fluid 2. The flow solver always operates in a single fluid and the fluxes at the interface are chosen such that the proper waves in the bulk phases are reproduced. Looking at a typical wave structure of the interface Riemann problem as sketched in figure 2.2, proper states for the ghost-cells are the states left and right of the material interface $\mathbf{U}_{\mathrm{L}}^*$ and $\mathbf{U}_{\mathrm{R}}^*$. In our simulations we directly take the fluxes $\mathbf{F}_{\mathrm{L}}^*$ and $\mathbf{F}_{\mathrm{R}}^*$.
This ghost-fluid approach shifts the waves generated in the interface Riemann problem inside the different fluids to the grid cell interface of the corresponding fluid. Defined in this way, the ghost data generate the proper waves in the bulk phases. Due to the different fluxes from the different sides at a grid cell with a material interface, the mixing is avoided, but the conservation can no longer be satisfied locally. A good quality of the determination of the interface velocity is important. The approach considered relies on the original ghost-fluid ideas

of Fedkiw et al. [30] and its extension by Merkle and Rohde [62] and by Rohde and Dressel [18] for piecewise constant approximations in one space dimension and in the multi-dimensional case, respectively.
In the bulk phases away from the interface, the numerical flux of the DGSEM scheme is calculated by a standard formulation for single-phase flow. The transfer of information from the local interface Riemann problem to the flow solver is illustrated in figure 2.3 in a one-dimensional case and in figure 2.4 for the generalization of the approach in two dimensions.
In case the interface position, as defined by the zero iso-contour of the level-set function, has moved across one grid cell, the state in this cell has to be re-defined according to its new fluid state. As no information is available, the state has to be extrapolated using the adjacent grid cells. This is due to the approximation of the interface in the sharp interface context. The surrounding cells are used for the reconstruction of the state in the cell

$$\mathbf{U}_{\text{new}} = \frac{1}{\sum_{i=1}^{\text{nCell}} \delta_{i,\text{new}}} \sum_{i=1}^{\text{nCell}} \delta_{i,\text{new}} \mathbf{U}_i \tag{2.12}$$

with

$$\delta_{i,\text{new}} = \begin{cases} 1 \text{ if fluid(i)} = \text{fluid(new)} \\ 0 \text{ else.} \end{cases} \tag{2.13}$$

This state averaging is done for all surrounding nCell cells of the same fluid phase (in 3D: nCell $\in [1, 6]$). To avoid pressure discontinuities due to the state extrapolation, the average pressure is calculated instead of the average total energy.

2.2.3 Interface tracking and geometry representation

The interface tracking is based on a level-set method [89] that allows for an accurate approximation of secondary interface properties such as interface normals $\mathbf{n}_{\text{LS}}$ or mean interface curvature κ. The level-set approximation is preferable in the context of a high-order discretization as a continuous function can be easily approximated with a high-order method in contrast to the jumping Volume of Fluids (VOF) solution. The level-set method divides the cells such that a cell is either in the gaseous or in the liquid phase.

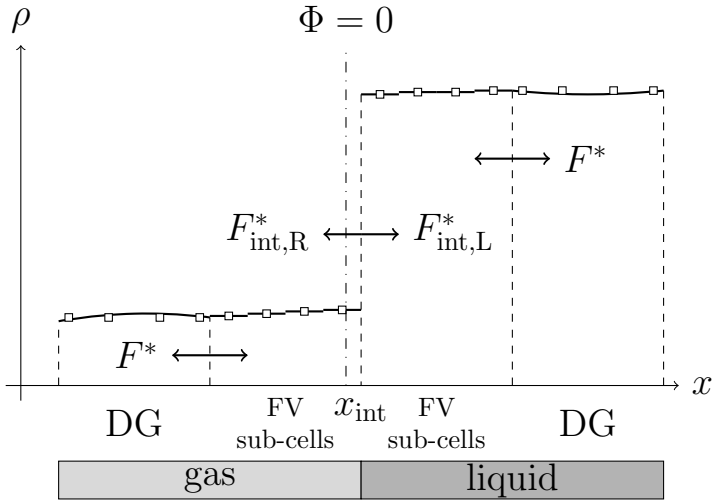


Figure 2.3: One-dimensional treatment of the fluxes at the phase interface. Within the bulk phases the standard single-phase Riemann flux F^* is used. The interface position is marked with $\Phi = 0$. At the interface position the fluxes $F^*_{\mathrm{int,L}}$ and $F^*_{\mathrm{int,R}}$ are applied to the left and right phase.

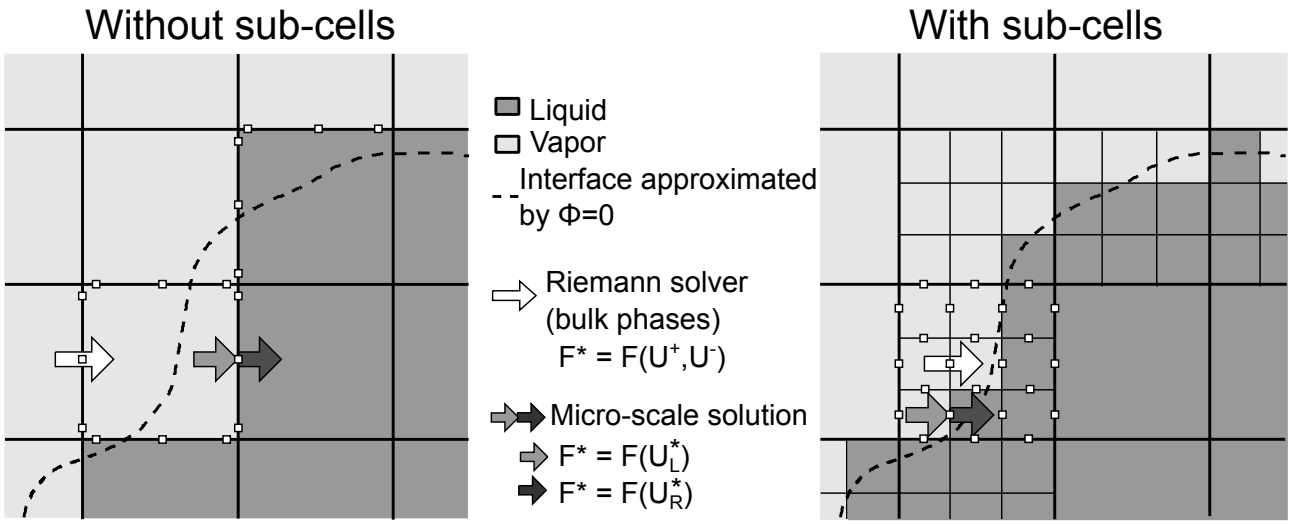


Figure 2.4: Sketch of a typical setting in the two-phase approach for the sharp interface approach: The liquid-vapor interface is approximated by the zero level-set $\Phi = 0$. According to the ghost-fluid approach, the numerical fluxes at the grid cell boundaries are determined either by a standard single-phase Riemann solver (bulk phase) or the interface Riemann solver and the ghost-cell data. The white dots visualize the integration points at the element boundaries.

In the context of the sharp interface approximation, we distinguish two different interfaces. The *tracked interface position* is described by the root of the level-set polynomial while the description of the *computational interface* is given by the surface dividing the cells that are predominately occupied by either the gas or the liquid phase. This approach allows the usage of different EOS in the liquid and gaseous phases respectively, if the coupling at the phase interface is possible. Therefore, a mixing of the bulk phases and the associated difficulty in finding a suitable numerical EOS is avoided.
The secondary interface properties are needed for the inclusion of surface tension effects as described by the Young-Laplace equation, written in the three-dimensional context as

$$(\rho(\mathbf{v}\cdot\mathbf{n}_{\mathrm{LS}}-\mathbf{s}_{\mathrm{PB}}\cdot\mathbf{n}_{\mathrm{LS}})\mathbf{v}+p(\rho,T)\mathbf{n}_{\mathrm{LS}})=2\sigma\kappa\mathbf{n}_{\mathrm{LS}}=\Delta p_{\sigma}\mathbf{n}_{\mathrm{LS}}. \tag{2.14}$$

The surface tension term enters the HMM as micro-scale model. With κ we denote the mean curvature of the interface associated with the orientation of the mean interface normal $\mathbf{n}_{\mathrm{LS}}$ and the interface normal velocity $\mathbf{s}_{\mathrm{PB}}\cdot\mathbf{n}_{\mathrm{LS}}$. For a pure fluid, σ is dependent on the pressure and temperature condition. In the following, the surface tension coefficient σ is assumed to be constant and positive. The inclusion of surface tension effects as well as the thermodynamic consistent coupling of the phases can be handled in a generalized Riemann problem with non-homogeneous jump conditions at the interface.

3 Numerical scheme

This chapter is dedicated to a description of the macro-scale solver used in the heterogeneous multi-scale method (see section 2.2). The solver is applied for the bulk phases away from the position of the phase interface. This method is also applied for the interface tracking method that is described in section 4.1 later on. The level-set method used can be written in conservation formulation similar to the Navier-Stokes equations for the flow solver. The high-order discretization for the level-set is advantageous due to the low numerical dissipation of the DG method.
In the first part of this chapter the underlying discontinuous Galerkin spectral element method (DGSEM) is described in detail. This scheme allows for a high order approximation of the flow field that is advantageous in smooth parts of the fluid. In the second part, the finite-volume sub-cell refinement is introduced that enhances the resolution of the phase interface and increases the stability of the numerical algorithm. Therefore, an integrated, block-structured finite-volume scheme is introduced that uses the large discontinuous Galerkin cells as domain decomposition.

3.1 Discontinuous Galerkin spectral element method

The discontinuous Galerkin method is a combination of the finite-volume (FV) and finite-element (FE) method that allows for a high order approximation together with the ability to resolve discontinuities. Thereby, inside the DG cells an element-locale continuous ansatz polynomial is used (similar to the approximation in the FE context) while jumps between the cells are permitted (similar to the FV approximation). Due to the discontinuous approximation at the cell interfaces, the estimation of the numerical fluxes at the cell edges rely on Riemann-type solvers. The usage of this discontinuous formulation increases the overall stability and allows for the approximation of discontinuities as they can be found, for example, in multi-phase flows or at shock waves.

In a first derivation step, the spatial discretization of the discontinuous Galerkin spectral element method is introduced that is based on the weak formulation of the conservation law. In a second step the temporal discretization based on a high-order Runge-Kutta (RK) method, see e. g. Carpenter [47] for details of different schemes, is outlined. In general any RK time discretization method can be chosen for the time integration. Here, the focus is on third and fourth order accurate RK methods.

3.1.1 Mapping of the equations

The description of the discontinuous Galerkin spectral element method for the simulation of compressible multi-phase flow is an extension of a single-fluid flow solver as described in [41]. The scheme is based on the discontinuous Galerkin spectral element approach as described by [48, 50]. The formulation of the DGSEM method is done for the system of conservation equations in the form

$$\mathbf{U}_t + \nabla \cdot \mathbf{F}(\mathbf{U}, \nabla\mathbf{U}) = \mathbf{0}, \tag{3.1}$$

where $\mathbf{U}$ is the vector of the solution unknowns, $\nabla\mathbf{U}$ is the gradient of the solution unknowns and $\mathbf{F}$ is the corresponding flux. In the following the Navier-Stokes equations are considered that include the advection and diffusion fluxes as introduced in 2.1.1. For the discretization of the gradient terms the BR1 method as described by Bassi and Rebay [6] is used.
In a three-dimensional domain the computational space is subdivided into non-overlapping hexahedral elements. Each element is mapped onto a reference cube element $E := [-1, 1]^3$ by a mapping $\boldsymbol{x}(\boldsymbol{\zeta})$ where $\boldsymbol{\zeta} = (\zeta^1, \zeta^2, \zeta^3)$ is the reference element coordinate vector. The mapping onto the reference element, E, transforms (3.1) to the system

$$J\mathbf{U}_t + \nabla \cdot \mathcal{F}\left(\mathbf{U}, \nabla\mathbf{U}\right) = \mathbf{0}, \tag{3.2}$$

where

$$\tilde{\boldsymbol{U}} = J\boldsymbol{U}, \quad \mathcal{F}^i = J\boldsymbol{a}^i \cdot \mathbf{F} = \sum_{n=1}^{3} \left(Ja_n^i\right) \boldsymbol{F}_n \tag{3.3}$$

are the mapped quantities. $\boldsymbol{a}^i$ are the contravariant basis vectors of the mapping and J the Jacobian. For a hexahedron with plane-parallel element sides the Jacobian and the metric terms $J\boldsymbol{a}^i$ are constant. Kopriva [49] showed how to compute the metric terms to satisfy the free-stream preserving condition.

3.1.2 Discontinuous Galerkin Formulation

The basis of the discontinuous Galerkin approximation is the weak form of the conservation law (3.1) that is obtained by a multiplication with the test function ϕ and an integration over the reference element E

$$\int_E J(\boldsymbol{\zeta})\mathbf{U}_t\phi d\boldsymbol{\zeta} + \int_E \nabla_{\boldsymbol{\zeta}} \cdot \mathcal{F}(\mathbf{U}, \nabla\mathbf{U})\phi d\boldsymbol{\zeta} = \mathbf{0}. \tag{3.4}$$

The weak form is now derived by using integration by parts for the second term

$$\int_E J(\boldsymbol{\zeta})\mathbf{U}_t\phi d\boldsymbol{\zeta} + \int_{\partial E} \left(\mathcal{F}\cdot\hat{\boldsymbol{n}}_\xi\right)^* \phi dS - \int_E \mathcal{F}\cdot\nabla_{\boldsymbol{\zeta}}\phi d\boldsymbol{\zeta} = \mathbf{0}, \tag{3.5}$$

with a suitable test function ϕ. $\boldsymbol{n}_{\boldsymbol{\zeta}}$ is the unit normal vector on the reference element faces. Since a discontinuous solution at the element faces may occur, Riemann solvers are used to approximate the flux on the element surface. The (approximate) Riemann solution is indicated by the superscript *.

3.1.3 Approximation of the solution and the fluxes

The solution of each element is approximated by polynomial tensor product bases with degree N, separately in each space direction

$$\mathbf{U}^N(\boldsymbol{\zeta}, t) \equiv \mathbb{I}^N\mathbf{U} = \sum_{\lambda,\mu,\nu=0}^{N} \hat{\mathbf{U}}_{\lambda\mu\nu}(t)\,\psi_{\lambda\mu\nu}(\boldsymbol{\zeta}), \tag{3.6}$$

where $\mathbb{I}^N$ is the discrete interpolation operator of order N. The basis functions ψ are products of one-dimensional Lagrange polynomials

$$\psi(\boldsymbol{\zeta}) = \psi_{ijk}(\boldsymbol{\zeta}) = \ell_i\left(\zeta^1\right)\ell_j\left(\zeta^2\right)\ell_k\left(\zeta^3\right). \tag{3.7}$$

In ζ^1 direction, the corresponding Lagrange polynomial $\ell_j\left(\zeta^1\right), j = 0, \ldots, N$ is of degree N and is defined by

$$\ell_j(\zeta^1) = \prod_{\substack{i=0 \\ i\neq j}}^{N} \frac{\zeta^1 - \zeta_i^1}{\zeta_j^1 - \zeta_i^1}, \quad j = 0, \ldots, N, \tag{3.8}$$

where ζ_i^1 are given points in $[-1, 1]$. To yield good interpolation properties, a favorable choice are the Gauss or Gauss-Lobatto points.

Each component of the contravariant fluxes is approximated by interpolation at the nodal points

$$\mathcal{F}^i\left(\mathbf{U}\left(\boldsymbol{\zeta}\right)\right) = \sum_{i,j,k=0}^{N} \hat{\mathcal{F}}^m_{ijk}\psi_{ijk}(\boldsymbol{\zeta}). \tag{3.9}$$

The corresponding polynomial coefficients are computed from the transformed physical fluxes

$$\hat{\mathcal{F}}^m_{ijk} = \hat{\mathcal{F}}^m\left(\zeta^1_i, \zeta^2_j, \zeta^3_k\right) = \sum_{d=1}^{3} J\mathbf{a}^m_d\left(\zeta^1_i, \zeta^2_j, \zeta^3_k\right) F_d\left(\zeta^1_i, \zeta^2_j, \zeta^3_k\right). \tag{3.10}$$

3.1.4 Approximation of the integrals

The integrals in (3.5) are approximated by numerical quadrature based on the Gauss-Legendre quadrature rule. Here, the same set of points are used for the interpolation as well as the integration. This choice strongly reduces the computational effort in this spectral element approach. The calculation of the three-dimensional integral can be split into one-dimensional sweeps.

3.1.4.1 Time derivative integral

The first integral in (3.5) is approximated using the ansatz for the solution as defined in (3.6) and choosing the test function as $\phi = \psi_{ijk}$

$$\frac{\partial}{\partial t}\int_E J\mathbf{U}\phi d\zeta = \frac{\partial}{\partial t}\int_E J(\boldsymbol{\zeta})\left(\sum_{l,m,n=0}^{N}\hat{\mathbf{U}}_{lmn}\psi_{lmn}(\boldsymbol{\zeta})\right)\psi_{ijk}(\boldsymbol{\zeta})d\boldsymbol{\zeta}\,. \tag{3.11}$$

In the next step, the integrals are split in each axis direction and each integral is approximated by a Gauss-Legendre quadrature rule

$$\begin{aligned}
\frac{\partial}{\partial t}\int_E J\mathbf{U}\phi d\boldsymbol{\zeta} &= \frac{\partial}{\partial t}\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} J(\boldsymbol{\zeta})\left(\sum_{l,m,n=0}^{N}\hat{\mathbf{U}}_{lmn}\psi_{lmn}(\boldsymbol{\zeta})\right)\psi_{ijk}(\boldsymbol{\zeta})d\zeta^1 d\zeta^2 d\zeta^2 \\
&= \frac{\partial}{\partial t}\sum_{\lambda,\mu,\nu=0}^{N} J\left(\boldsymbol{\zeta}_{\lambda\mu\nu}\right)\left(\sum_{l,m,n=0}^{N}\hat{\mathbf{U}}_{lmn}\underbrace{\ell_l(\zeta^1_\lambda)}_{\delta_{l\lambda}}\underbrace{\ell_m(\zeta^2_\nu)}_{\delta_{m\mu}}\underbrace{\ell_n(\zeta^3_\mu)}_{\delta_{k\nu}}\right)\psi_{ijk}(\boldsymbol{\zeta}_{\lambda\mu\nu})\omega_\lambda\omega_\mu\omega_\nu
\end{aligned}$$

$$= \frac{\partial}{\partial t} \sum_{\lambda,\mu,\nu=0}^{N} J\left(\boldsymbol{\zeta}_{\lambda\mu\nu}\right) \hat{\mathbf{U}}_{\lambda\mu\nu} \underbrace{\ell_i(\zeta_\lambda^1)}_{\delta_{i\lambda}} \underbrace{\ell_j(\zeta_\mu^2)}_{\delta_{j\mu}} \underbrace{\ell_k(\zeta_\nu^3)}_{\delta_{k\nu}} \omega_\lambda \omega_\mu \omega_\nu$$

$$= J\left(\boldsymbol{\zeta}_{ijk}\right) \omega_i \omega_j \omega_k \frac{\partial \hat{\mathbf{U}}_{ijk}}{\partial t}. \tag{3.12}$$

Note that due to the collocation of interpolation and integration as well as the exploit of the Lagrange property, the computational effort for the integral evaluation is significantly reduced. Equation (3.12) is integrated exactly by a Gauss-Legendre quadrature as long as the Jacobian is linear in $\boldsymbol{\zeta}$, thus for linear and trilinear hexahedra.

3.1.4.2 Volume integral

The volume integral is split into the three contravariant flux components

$$\int_E \mathcal{F}(\mathbf{U}) \cdot \nabla_{\boldsymbol{\zeta}} \phi d\boldsymbol{\zeta} = \sum_{d=1}^{3} \int_E \mathcal{F}^d(\mathbf{U}) \frac{\partial \phi}{\partial \zeta^d} d\boldsymbol{\zeta}\,. \tag{3.13}$$

In the following the focus is on the first component of the contravariant flux $\mathcal{F}^1$. As a first step the flux approximation (3.10) is inserted and the test function is set to $\phi = \psi_{ijk}$. Then a Gauss quadrature is applied and inserted in the equation. Similar to the volume integral (3.12) nearly all sums cancel out

$$\begin{aligned}
&\int_E \mathcal{F}^1(\mathbf{U}) \frac{\partial \phi}{\partial \zeta^1} d\boldsymbol{\zeta} \\
&= \int_E \left(\sum_{l,m,n=0}^{N} \mathcal{F}^1_{lmn} \psi_{lmn}(\boldsymbol{\zeta}) \right) \frac{\partial \psi_{ijk}(\boldsymbol{\zeta})}{\partial \zeta^1} d\boldsymbol{\zeta} \\
&= \sum_{\lambda,\mu,\nu=0}^{N} \left(\sum_{l,m,n=0}^{N} \mathcal{F}^1_{lmn} \underbrace{\ell_l(\zeta_\lambda^1)}_{\delta_{l\lambda}} \underbrace{\ell_m(\zeta_\mu^2)}_{\delta_{m\mu}} \underbrace{\ell_n(\zeta_\nu^3)}_{\delta_{n\nu}} \right) \frac{\partial \psi_{ijk}(\boldsymbol{\zeta})}{\partial \zeta^1} \omega_\lambda \omega_\mu \omega_\nu \\
&= \sum_{\lambda,\mu,\nu=0}^{N} \mathcal{F}^1_{\lambda\mu\nu} \left. \frac{d\ell_i(\zeta)}{d\zeta} \right|_{\zeta=\zeta_\lambda^1} \underbrace{\ell_j(\zeta_\mu^2)}_{\delta_{j\mu}} \underbrace{\ell_k(\zeta_\nu^3)}_{\delta_{k\nu}} \omega_\lambda \omega_\mu \omega_\nu \\
&= \omega_j \omega_k \sum_{\lambda=0}^{N} \mathcal{F}^1_{\lambda jk} \left. \frac{d\ell_i(\zeta)}{d\zeta} \right|_{\zeta=\zeta_\lambda^1} \omega_\lambda.
\end{aligned} \tag{3.14}$$

Equation (3.14) can be seen as a one-dimensional matrix vector product with the differentiation matrix of the Lagrange polynomial

$$D_{ij} = \left.\frac{d\ell_i(\zeta)}{d\zeta}\right|_{\zeta=\zeta_j^1}, \quad i,j = 0,\ldots,N. \tag{3.15}$$

Putting all this together reveals the final version of the volume integral for the three-dimensional case

$$\begin{aligned}\int_E \mathcal{F}(\mathbf{U}) \cdot \nabla_{\boldsymbol{\zeta}} \phi d\boldsymbol{\zeta} = \omega_j\omega_k \sum_{\lambda=0}^{N} D_{i\lambda}\hat{\mathcal{F}}^1_{\lambda jk}\omega_\lambda \\ + \omega_i\omega_k \sum_{\mu=0}^{N} D_{j\mu}\hat{\mathcal{F}}^2_{i\mu k}\omega_\mu \\ + \omega_i\omega_j \sum_{\nu=0}^{N} D_{k\nu}\hat{\mathcal{F}}^3_{ij\nu}\omega_\nu \,.\end{aligned} \tag{3.16}$$

3.1.4.3 Surface Integral

Similar to the approach in the derivation of the volume integral, the surface integral is applied dimension by dimension

$$\begin{aligned}\int_{\partial E} (\mathcal{F}\cdot\boldsymbol{n}_{\boldsymbol{\zeta}})^* \,\phi dS = \left[\int_{-1}^{1}\int_{-1}^{1} (\mathcal{F}\cdot\boldsymbol{n}_{\boldsymbol{\zeta}})^* \,\phi d\zeta^2 d\zeta^3\right]_{\zeta^1=-1}^{1} \\ + \left[\int_{-1}^{1}\int_{-1}^{1} (\mathcal{F}\cdot\boldsymbol{n}_{\boldsymbol{\zeta}})^* \,\phi d\zeta^1 d\zeta^3\right]_{\zeta^2=-1}^{1} \\ + \left[\int_{-1}^{1}\int_{-1}^{1} (\mathcal{F}\cdot\boldsymbol{n}_{\boldsymbol{\zeta}})^* \,\phi d\zeta^1 d\zeta^2\right]_{\zeta^3=-1}^{1}.\end{aligned} \tag{3.17}$$

For a demonstration of the structure of the surface integral term, again the ζ^1 direction is chosen. The definition of the contravariant fluxes is given in equation (3.10) which one wants to compute using Riemann solvers. The numerical flux of the Riemann solver is denoted by *. Note that at the numerical phase interface the numerical flux is calculated using a generalized Riemann solver that takes the interface physics into account. This is described in chapter 6. The only difference to the single-phase case is the choice of the numerical flux. The derivation of the surface integral remains identical.

For $\zeta^1 = \pm 1$ the normal vector in reference space is $\boldsymbol{n}_{\boldsymbol{\zeta}} = (\pm 1, 0, 0)^{\mathrm{T}}$ and the flux reduces to

$$\begin{aligned}(\mathcal{F} \cdot \mathbf{n}_{\boldsymbol{\zeta}}) &= \pm \mathcal{F}^1, \\ \mathcal{F}^1 &= \sum_{n=1}^{3} J\mathbf{a}_n^1(\pm 1, \zeta^2, \zeta^3)\mathbf{F}(\pm 1, \zeta^2, \zeta^3) := (\mathbf{F} \cdot \boldsymbol{N}_x)\, \hat{s},\end{aligned} \tag{3.18}$$

with $\boldsymbol{N}_x$ being the normal vector in physical space. This surface normal vector is defined by the metric terms evaluated at the surface. The resulting surface element $\hat{s}$ and the unit normal in physical space $\mathbf{N}_x$ are computed as

$$\hat{s} = \sqrt{\sum_{n=1}^{3} \left(J\mathbf{a}_n^1(\pm 1, \zeta^2, \zeta^3)\right)^2}\,, \tag{3.19}$$

$$N_{x,n} = \frac{\pm J\mathbf{a}_n^1}{\hat{s}}\,. \tag{3.20}$$

The contravariant fluxes can now be replaced with the surface element and the numerical flux function $\mathbf{F}^*$

$$(\mathcal{F} \cdot \boldsymbol{n}_{\boldsymbol{\zeta}})^* = (\mathbf{F} \cdot \boldsymbol{N}_x)^*\, \hat{s} = \mathbf{F}^*(\mathbf{U}^L, \mathbf{U}^R, \boldsymbol{N}_x)\hat{s}. \tag{3.21}$$

This flux function now depends only on the discontinuous states at the element face and the unit normal in physical space. In case Gauss quadrature points are used for the approximation, the state at the element face is extrapolated from the inner Gauss nodes, yielding

$$\mathbf{F}^*(\pm 1, \zeta^2, \zeta^3)\hat{s} = \sum_{j,k=0}^{N} \left[\mathbf{F}^*\hat{s}\right]_{j,k}^{\pm\zeta^1} \ell_j(\zeta^2)\ell_k(\zeta^3)\,. \tag{3.22}$$

Evaluation on the face in positive ζ^1 direction are marked by $+\zeta^1$ and in negative direction by $-\zeta^1$.
Now the numerical flux function and the test function $\phi = \psi_{ijk}$ are inserted into the surface integral for $\zeta^1 = \pm 1$

$$\left[\int_{-1}^{1}\int_{-1}^{1} \mathbf{F}^*(\mathbf{U}^L, \mathbf{U}^R; \boldsymbol{N}_x)\hat{s}\phi d\zeta^2 d\zeta^3\right]_{\zeta^1=-1}^{1}. \tag{3.23}$$

In a second step the integrals are replaced by Gauss-Legendre quadrature resulting in

$$
\begin{aligned}
&= \sum_{\mu,\nu=0}^{N} \left(\sum_{m,n}^{N} [\mathbf{F}^* \hat{s}]_{mn}^{+\zeta^1} \underbrace{\ell_m(\zeta_\mu^2)}_{\delta_{m\mu}} \underbrace{\ell_n(\zeta_\nu^3)}_{\delta_{n\nu}} \right) \ell_i(1)\ell_j(\zeta_\mu^2)\ell_k(\zeta_\nu^3)\omega_\mu\omega_\nu \\
&- \sum_{\mu,\nu=0}^{N} \left(\sum_{m,n}^{N} [\mathbf{F}^* \hat{s}]_{mn}^{-\zeta^1} \underbrace{\ell_m(\zeta_\mu^2)}_{\delta_{m\mu}} \underbrace{\ell_n(\zeta_\nu^3)}_{\delta_{n\nu}} \right) \ell_i(-1)\ell_j(\zeta_\mu^2)\ell_k(\zeta_\nu^3)\omega_\mu\omega_\nu \\
&= \sum_{\mu,\nu=0}^{N} \Big([\mathbf{F}^* \hat{s}]_{jk}^{+\zeta^1} \ell_i(1) - [\mathbf{F}^* \hat{s}]_{jk}^{-\zeta^1} \ell_i(-1) \Big) \underbrace{\ell_j(\zeta_\mu^2)}_{\delta_{j\mu}} \underbrace{\ell_k(\zeta_\nu^3)}_{\delta_{n\nu}} \omega_\mu\omega_\nu \\
&= \Big([\mathbf{F}^* \hat{s}]_{jk}^{+\zeta^1} \ell_i(1) - [\mathbf{F}^* \hat{s}]_{jk}^{-\zeta^1} \ell_i(-1) \Big) \omega_j\omega_k. \qquad (3.24)
\end{aligned}
$$

This procedure can now be applied analogously to the ζ^2 and ζ^3 directions leading to the surface integral contribution

$$
\begin{aligned}
\int_{\partial E} (\mathcal{F} \cdot \boldsymbol{n}_{\boldsymbol{\zeta}})^* \, \phi dS = &\Big([\mathbf{F}^* \hat{s}]_{jk}^{+\zeta^1} \ell_i(1) - [\mathbf{F}^* \hat{s}]_{jk}^{-\zeta^1} \ell_i(-1) \Big) \omega_j\omega_k \\
&+ \Big([\mathbf{F}^* \hat{s}]_{ik}^{+\zeta^2} \ell_j(1) - [\mathbf{F}^* \hat{s}]_{ik}^{-\zeta^2} \ell_j(-1) \Big) \omega_i\omega_k \\
&+ \Big([\mathbf{F}^* \hat{s}]_{ij}^{+\zeta^3} \ell_k(1) - [\mathbf{F}^* \hat{s}]_{ij}^{-\zeta^3} \ell_k(-1) \Big) \omega_i\omega_j \, . \qquad (3.25)
\end{aligned}
$$

3.1.4.4 Semi-discrete formulation

With the definitions of the following precomputed operators

$$
\hat{\ell}_i = \frac{\ell_i}{\omega_i}, \quad \hat{D}_{ij} = -\frac{\omega_i}{\omega_j} D_{ij}, \quad i,j = 0,\ldots,N, \qquad (3.26)
$$

the time integral (3.12), the volume integral (3.16) and the surface integral (3.25) can now be assembled in the weak form (3.5). The time derivative of each DOF on each hexahedron can be estimated by

$$
(\mathbf{U}_{ijk})_t = -\frac{1}{J_{ijk}} \left[\sum_{\lambda=0}^{N} \hat{D}_{i\lambda} \hat{\mathcal{F}}_{\lambda jk}^1 + \sum_{\mu=0}^{N} \hat{D}_{j\mu} \hat{\mathcal{F}}_{i\mu k}^2 + \sum_{\nu=0}^{N} \hat{D}_{k\nu} \hat{\mathcal{F}}_{ij\nu}^3 \right.
$$

$$
\begin{aligned}
&+ \left([\mathbf{F}^* \hat{s}]_{jk}^{+\zeta^1} \, \hat{\ell}_i(1) - [\mathbf{F}^* \hat{s}]_{jk}^{-\zeta^1} \, \hat{\ell}_i(-1) \right) \\
&+ \left([\mathbf{F}^* \hat{s}]_{ik}^{+\zeta^2} \, \hat{\ell}_j(1) - [\mathbf{F}^* \hat{s}]_{ik}^{-\zeta^2} \, \hat{\ell}_j(-1) \right) \\
&+ \left([\mathbf{F}^* \hat{s}]_{ij}^{+\zeta^3} \, \hat{\ell}_k(1) - [\mathbf{F}^* \hat{s}]_{ij}^{-\zeta^3} \, \hat{\ell}_k(-1) \right) \quad \Big] \, . \qquad (3.27)
\end{aligned}
$$

For the time update it is necessary to compute the contravariant fluxes $\mathcal{F}^m$ by means of equation (3.10) for the volume integral. Furthermore, if Gaussian integration and interpolation points are chosen, the solution on the inner Gauss points has to be extrapolated to the element faces to compute the Riemann fluxes $\mathbf{F}^*(\mathbf{U}^+, \mathbf{U}^-, \mathbf{N}_x)$ of the surface integral. The extrapolation is only a 1D scalar product, for example shown in ζ^1 direction

$$
\mathbf{U}_{jk}^{+\zeta^1} = \sum_{i=0}^{N} \mathbf{U}_{ijk} \ell_i(\zeta^1 = 1). \qquad (3.28)
$$

The surface flux contribution $[\mathbf{F}^* \hat{s}]_{jk}^{+\zeta^1}$ is computed once within the bulk phases to ensure conservative fluxes. Away from the phase interface, the contribution for the neighboring element is simply $-[\mathbf{F}^* \hat{s}]_{jk}^{+\zeta^1}$. At the position of the phase interface, two fluxes are calculated as the different phase states have to be taken into account. The focus of chapter 6 is dedicated to an description how to compute the numerical fluxes at the phase interface.
The tensor product structure of the DGSEM operator can now be visualized as in figure 3.1 for the choice of Gauss and Gauss-Lobatto integration points.

3.1.5 Gradient estimation

The gradients of the solution unknowns, that are needed for the estimation of the viscous fluxes as defined in 2.1.1, are computed using the BR1 gradient lifting method as introduced by Bassi and Rebay [6]. The BR1 method is implemented in the DGSEM context. This estimation method is standard and discussed only briefly.
The basic steps of the BR1 gradient lifting are:

1. Reformulation of the second order PDE to a system of first order with additional PDEs for the gradients.

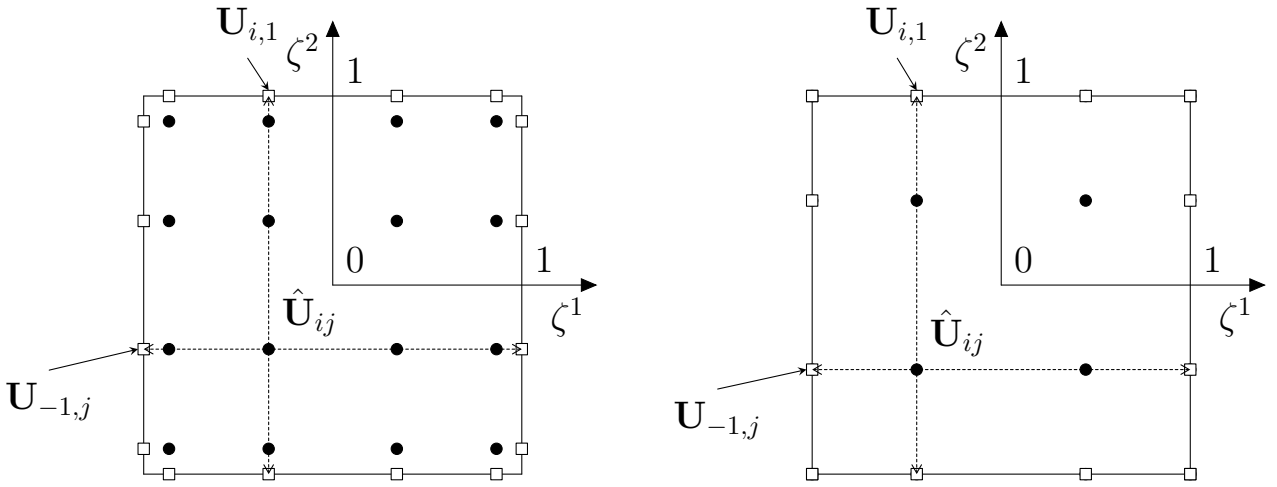


Figure 3.1: Tensor product structure of the DGSEM operator for $N = 3$ with Gauss integration points (left) and Gauss-Lobatto integration points (right). The inner Gauss points are marked by black circles, the surface integration points by rectangles. Note that in the Gauss-Lobatto case integration points on the surfaces are already present.

2. Introduction of a suitable numerical flux for the gradient lifting. In the BR1 case the arithmetic mean is chosen as numerical flux since it is stable due to the parabolic character of the gradient terms.

For a real fluid, the temperature gradient cannot be calculated directly from the conservative gradients, as estimated by the BR1 gradient lifting. To overcome this problem, a temperature lifting is applied. The temperature is added to the conservative variables for the gradient estimation and the numerical procedure is similar to the lifting of the conservative variables $\mathbf{U}$. At each time step the temperature within the flow field has to be estimated from the conservative variables $\mathbf{U}$.

3.1.6 Time integration

The in the last sections described DG method is independent of the time discretization used. Since the numerical flux is dependent on the state on both grid cell edges the system is weakly coupled. For the numerical flux calculation, only the direct neighbor cells are needed. The time evolution can be approximated by a suitable standard time integration method, e. g. Runge-Kutta methods. In the following, explicit third and fourth order Runge-Kutta methods are used for time integration as e. g. described by Kennedy et al. [47].

The explicit time step restriction for the DG scheme is estimated using the CFL-number (named after Courant, Friedrichs and van Lewy)

$$\Delta t \leq \mathrm{CFL}_N \frac{\min(\Delta x, \Delta y, \Delta z)}{\max(|c| + |\mathbf{v}|)}, \tag{3.29}$$

together with a scaling factor CFL_N.
This time step limitation can be estimated using the metric term, that arise from the transformation onto the reference element. Due to the hyperbolic-parabolic character of the Navier-Stokes equations, one has to consider a convective and viscous time step restriction. For the estimation of both constraints, one transforms the conservation equations onto the reference element $[\zeta^1, \zeta^2, \zeta^3] \in [-1; 1]^3$ in analogy to the derivation of the DG operator. For the high-order DG approximation, the scaling factor CFL_N is dependent on the approximation order used in the computation.
For the convective time step, the relevant signal velocity arises from the eigenvalues of the inviscid Euler equations. These eigenvalues are dependent of the fluid velocity $\mathbf{v}$ and the sound speed c

$$\lambda_{\text{euler}} = (v_n - c, v_n, v_n, v_n, v_n + c). \tag{3.30}$$

Using the transformation onto the reference element the eigenvalues can be described by means of the metric terms $J\mathbf{a}_n^i$ in the following way

$$\tilde{\mathbf{v}}^i = \sum_{n=1}^{3} J\mathbf{a}_n^i \mathbf{v}_n^i, \qquad \tilde{c}^i = c\sqrt{\sum_{n=1}^{3} (J\mathbf{a}_n^i)^2}. \tag{3.31}$$

The eigenvalues needed for the CFL time step constraint can now be described by the metric terms. For the maximum eigenvalue $|\tilde{\mathbf{v}}| + \tilde{c}$ in the reference cell one obtains

$$\Delta t \leq \mathrm{CFL}_N \frac{2}{\max\left(\frac{1}{J}\sum_{i=1}^{3}(|\tilde{\mathbf{v}}^i| + \tilde{c}^i)\right)}. \tag{3.32}$$

With this transformation one can describe the time step constraint efficiently in the reference cell (that has always an identical size) and is no longer dependent on the physical dimension of the grid cell. This approach avoids the approximation of the minimum element length, e. g., using incircles.

The estimation of the viscous time step restriction is similar, only an additional approximation is necessary for the flux term. This results in the following formula for the viscous time step of an ideal gas with polytropic coefficient γ

$$\Delta t \leq \mathrm{DFL}_N \frac{4}{\max\left(\frac{\mu\gamma}{\rho \mathrm{Pr}} \sum_{n=1}^{3} \left(\frac{J\mathbf{a}_n^i}{J}\right)^2\right)}. \tag{3.33}$$

and an scaling factor DFL_N.
In case of a cartesian grid the above equation can be simplified as the sum only has one term

$$\Delta t \leq \mathrm{DFL} \frac{4}{\max\left(\frac{\mu\gamma}{\rho \mathrm{Pr}} \left(\frac{2}{\Delta x} + \frac{2}{\Delta y} + \frac{2}{\Delta z}\right)\right)} \tag{3.34}$$

using a scaling factor DFL_N for the parabolic part (similar to the advection part). The estimation for general EOS is similar but the diffusive eigenvalues of the Navier-Stokes equations are needed.
With this estimation the convective and diffusive time steps are estimated and scaled with a scaling factor that is dependent of the polynomial degree of the approximation. The dependence is e. g. described in Gassner [32]. The global time step of the scheme is the minimum of both time steps. In order to adapt the time step to the high approximation order of the DG scheme, the $\mathrm{CFL}_N/\mathrm{DFL}_N$ number is scaled with the polynomial degree of the ansatz functions. This results in smaller time steps for high order approximations.

3.2 Sub-cell refinement

In this section the basics of the finite-volume sub-cell refinement are described that is used locally in the vicinity of the phase interface to enlarge the numerical resolution of the phase interface. Another field of application is the shock capturing in case strong shocks are present in the computational domain (see e. g. the shock-droplet interaction test case in section 7.1.3). Based on an indicator value the DG cell is refined to multiple finite-volume cells to provide numerical stability.
Due to the chosen sharp interface approach, which arises from the Ghost-Fluid method [30] for the numerical treatment of the phase interface, jump terms at the phase boundary have to be resolved. These jump terms are provided by the micro-solver that is described in section 6. A difficulty arising with the used

high-order macro-scale method is that the numerical resolution of the phase interface is coarse and does not reflect the physical approximation of the phase boundary very well. This is a direct consequence of the DG approximation as a continuous in-cell resolution is assumed. Using high-order DG methods the grid cells get larger and larger (if the total DOF are kept equal) due to the high-order ansatz functions inside the DG cell. Similar to the finite-volume approach, discontinuities are only allowed at element boundaries, but due to the continuous in-cell resolution a DG cell has $(N+1)^3$ degrees of freedom instead of 1^3 in the finite-volume (FV) discretization within a three-dimensional discretization. This leads to a dilemma that one would like to exploit high-order approximations for the estimation of interface curvature terms as well as a fine interface resolution. On the other hand the numerical resolution of the phase boundary limits the usable order of approximation.

One approach to overcome this disadvantage of the DG scheme is to use a finite-volume sub-cell refinement within one DG cell. A direct refinement using multiple DG cells would have a massive impact on the global time step restriction and would be difficult to handle in the data structure. The advantage of the DG method is that different schemes can be easily coupled using the numerical flux at the element faces. This idea is extended and the DG mesh is refined in the vicinity of the phase interface using multiple equidistant FV cells such that the total degrees of freedom remain constant in the domain. This implies that one DG cell is refined into $(N+1)^3$ FV cells. Therefore, the massive time step restriction is circumvented that would be imposed if the DG cells would be refined, as the DG time step is much more restricted than the FV time step.

In the following sections, the basics of the finite-volume scheme are introduced and its application within a DG macro-cell. In the first part the description of the FV-scheme is written down. In the second part the application of the FV scheme inside a DG macro cell is introduced. The FV scheme can be introduced into the DG method without difficulties as both methods use numerical fluxes based on Riemann solvers at the grid cell boundaries. The only difference is the choice of the ansatz function inside the grid cell. In the finite-volume context integral cell mean values are used while in the DG context polynominal basis functions are applied that lead to a high approximation order.

3.2.1 Finite-Volume scheme

The finite-volume (FV) scheme can be derived starting from a general conservation law of the type

$$\mathbf{U}_t + \nabla \cdot \boldsymbol{F}(\mathbf{U}, \nabla\mathbf{U}) = \mathbf{0}\,. \tag{3.35}$$

Similar to the DG scheme the equation is mapped to the reference element as defined in (3.3) leading to the transformed conservation law

$$J\mathbf{U}_t + \nabla \cdot \mathcal{F}(\mathbf{U}, \nabla\mathbf{U}) = \mathbf{0}\,, \tag{3.36}$$

compare section 3.1.
The occurrence of physical discontinuities like shock wave prohibits the direct usage of this equation for the construction of numerical scheme. Instead, the weak formulation of equation (3.36)

$$\int_{E_{\mathrm{FV}}} J\mathbf{U}_t dE_{\mathrm{FV}} + \int_{E_{\mathrm{FV}}} \nabla \cdot \mathcal{F}(\mathbf{U}, \nabla\mathbf{U}) dE_{\mathrm{FV}} = 0 \tag{3.37}$$

on the finite-volume reference element E_{FV} is used. The reference element E_{FV} is defined for $(\zeta^1, \zeta^2, \zeta^3) \in [-1, 1]^3$. The right volume integral can now be transformed to a surface integral using the Gauss theorem. The volume integral vanishes due to the central finite-volume approximation of integral mean values in each cell. The final FV formulation on the reference element can be obtained to

$$\int_{E_{\mathrm{FV}}} J\mathbf{U}_t dE_{\mathrm{FV}} + \oint_{\partial E_{\mathrm{FV}}} \left(\mathcal{F}(\mathbf{U}, \nabla\mathbf{U}) \cdot \mathbf{n}_{\boldsymbol{\zeta}}\right)^* \, dS_{\mathrm{FV}} = 0\,. \tag{3.38}$$

For the approximation of the numerical fluxes at the element boundaries Riemann solvers are used. The physical flux $\mathbf{F}$ is mapped to the contravariant flux $\mathcal{F}$ as defined in equation (3.21). Within the bulk phases standard Riemann solvers are used, as described e. g. in Toro [96]. The approach for the calculation of the numerical fluxes at the phase interface is detailed in chapter 6. The only difference at the phase interface is that non-conservative fluxes are used that take the different equations of states on both sides of the interface into account. Due to the approximation of integral mean values, a second order (or higher order ENO or WENO) scheme has to consider neighbor information for the higher order reconstruction. Here, a TVD reconstruction limiter is applied to achieve a second order finite volume scheme.
In the next section the method to integrate the FV scheme into the DG context described in section 3.1 based on a sub-cell refinement.

3.2.2 Finite-volume sub-cell ansatz inside DG macro-cell

The finite-volume sub-cell approach can be efficiently included into the DGSEM description as coupling of the grid cells is solely done using the numerical fluxes at the element boundaries. In this context, one DG cell is replaced by $(N+1)^3$ second-order finite-volume cells for a three-dimensional discretization. This approach is done to retain the total number of degrees of freedom in the computational domain and restrict the negative impact on the time step restriction due to the refinement.
The finite-volume sub-cell ansatz enters the DGSEM description in terms of a modified volume integral. Instead of the continuous DG volume integral the sum of surface contributions for the equidistant sub-cell FV cells is calculated. In the following the surfaces of the FV sub-cells are marked with ∂E_{FV} to distinguish between the DG macro-cells and FV sub-cells. The modified volume integral for the block-structured FV-scheme can be written in the following way

$$\int_E \mathcal{F} \cdot \nabla_{\boldsymbol{\zeta}} \phi_{ijk} dE = \sum_{i,j,k=0}^{N} \int_{\partial E_{FV} \setminus \partial E_{\mathrm{DG}}} (\mathcal{F} \cdot \boldsymbol{n}_{\boldsymbol{\zeta}})^* \, dS, \tag{3.39}$$

where ∂E_{FV} is the surface area of the sub-cell FV-cell V_{FV}. The additional sum with the indices i, j and k adds the contributions of all FV-cells inside the DG macro-cell. i describes thereby the FV-cells in the local ζ^1, j in ζ^2 and k in ζ^3-direction. This approach allows discontinuities between each sub-cell as no continuity constraint is enforced. This approach is advantageous for the interface resolution as it allows a numerically finer resolution without increasing the overall degrees of freedom. Of course, the disadvantage is that the order of the scheme is reduced locally. However, due to the jump terms and discontinuities, a high-order non-oscillating approximation is difficult to achieve (or even impossible).
At the border between DG and FV sub-cells the state and flux have to be interpolated. This is needed due to the different surface integration points in the FV and DG cells. In the FV sub-cells equidistantly distributed points are used while in the DG cell uses Gaussian distributed ones. This conversion between the different points is implemented using a two-dimensional change-basis routine. The basis of the conversion is the conservative $\mathbb{L}_2$ projection

$$\int_E (\boldsymbol{U}_{ijk} \psi_{ijk})_{\mathrm{DG}} \; d\boldsymbol{\zeta} = \sum_{i,j,k=0}^{N} \int_{E_{\mathrm{FV}}} (\boldsymbol{U}_{ijk} \cdot 1)_{\mathrm{FV}} \, d\boldsymbol{\zeta} = \sum_{i,j,k=0}^{N} (\boldsymbol{U}_{ijk})_{\mathrm{FV}} \tag{3.40}$$

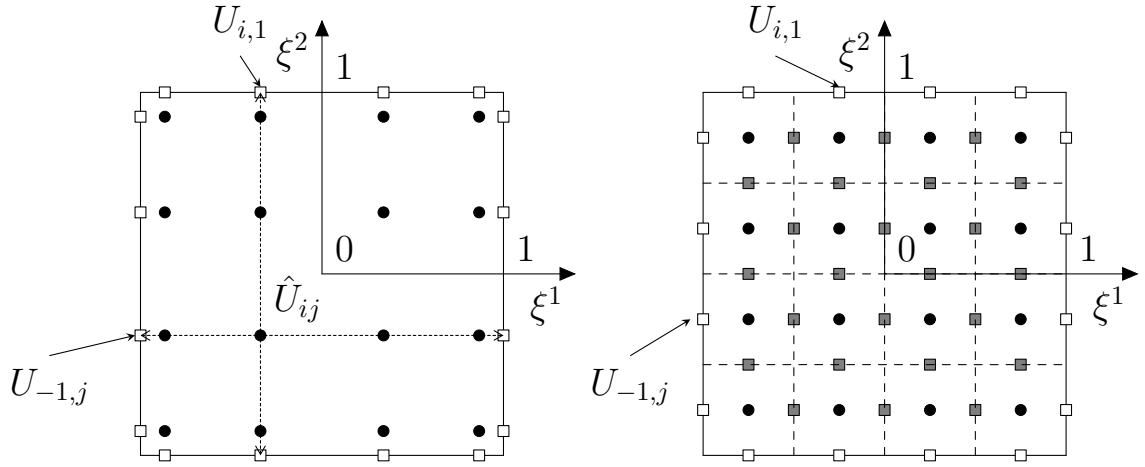


Figure 3.2: Comparison between a DG cell (left) with interior points according to the Gaussian integration rule and a refined DG cell (right). In the refined case, the DG cell is replaced by multiple FV sub-cells, that are indicated by dashed lines. The gray rectangles denote the additional flux evaluation points of the numerical flux inside the DG macro cell.

and the cell-locale reconstruction that is the inverse of the $\mathbb{L}_2$ projection. Note that in the FV method constant basis functions are applied. The overall estimation of the surface integral in (3.27) remains the same. The sub-cell partition is done such that the number of the degrees of freedom remains constant. This allows to switch from the spectral representation of DGSEM directly to a piecewise constant representation and vice versa.

In figure 3.2 the numerical approaches for a high-order DG cell and a refined DG cell are presented. In the DG case, the volume integral is approximated using a high-order quadrature based on the Gauss points. This approach assumes that the function is continuous within the grid cell. In contrast, a discontinuous flux function can be approximated in the case with multiple FV sub-cells. The advantage is that a discontinuous state, as it can be found at a phase interface, can be resolved better using the FV sub-cell approach. Due to the second-order TVD approach, an oscillating solution is prohibited and the interface resolution is increased.

The possibility of the inclusion of jump terms within one DG cell through the presented FV-sub-cell approach significantly increases the resolution of the phase interface. This enhancement is visualized in figure 3.3 showing the density iso-contour of the liquid phase (corresponding to the numerically approximated

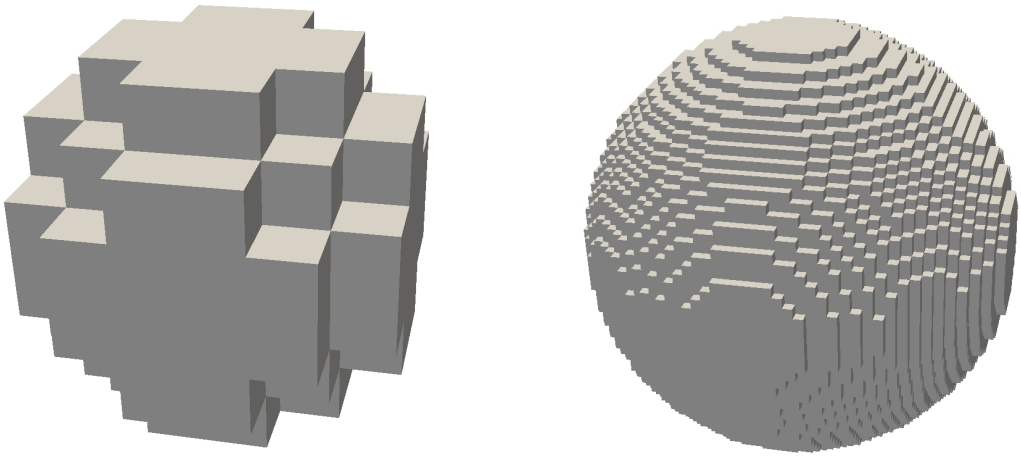


Figure 3.3: Comparison of the phase boundary approximation for a DG scheme of order 6 using the density iso-contour of the liquid phase. **Left:** Approximation using a pure DG scheme. **Right:** Approximation using the adaptive mesh refinement at the phase boundary.

interface). For this study a sixth-order DG scheme (polynomial degree $N = 5$) was used. This sub-cell approach combines both the high order resolution of a DG scheme in smooth regions as well as the fine resolution of discontinuities of a finite volume scheme. In order to limit the impact of the underlying finite-volume scheme on the global solution (the underlying sub-cell FV-scheme has larger dissipation and dispersion errors), the sub-cell approach is limited to a small region next to the physical phase boundary.
The staircase approximation of the interface in the flow solver is a large simplification in the chosen approach. This is done to simplify the treatment of the interface inside the flow solver and is similar to immersed boundary methods. The orientation and the shape of the interface are considered in an additional step in the local solution at the interface (described in chapter 6). However, this method allows a flexible treatment of the interface that may change the topology with time.

3.3 Indicator calculation

As a last step, one needs the information where to apply the DG and the FV scheme within the computational domain. This decision is based on the calculation of indicators that estimate either the oscillation degree (limiter for shock capturing) or the distance to the phase interface. The indicators are

calculated for each DG cell in every time step and depending on the indicator value, a cell is "refined" using finite-volume sub-cells or "coarsened" to the DG scheme.
The first indicator that is considered here is an indicator for the detection of the phase interface, while the Persson indicator is one possible shock shock indicator. Many other indicators exist for shock capturing purposes in literature.

3.3.1 Level-set indicator

The simple level-set indicator determines whether a cell is located in the vicinity of the interface or not. The indicator value is the cell-averaged level-set value that can be calculated as

$$\mathrm{Ind}_\Phi = \frac{\sum_{i,j,k=0}^{N} \Phi_{ijk}\omega_i\omega_j\omega_k}{\sum_{i,j,k=0}^{N} \omega_i\omega_j\omega_k} = \frac{\sum_{i,j,k=0}^{N} \Phi_{ijk}\omega_i\omega_j\omega_k}{8}\,, \tag{3.41}$$

with ω being the Gauss integration weights and Φ_{ijk} the level-set value at position i, j, k inside the DG cell. In case the absolute value of the indicator $|\mathrm{Ind}_\Phi|$ is lower than a specific value, the cell is calculated using the sub-cell FV scheme. For larger indicator values, the DG scheme is used for the calculation of the time update.

3.3.2 Persson indicator

The calculation of this indicator is based on the description of Persson and Peraire in [74] for the high-order DG scheme. The indicator was initially proposed by Neumann and Richtmyer [99] and is based on the principle that locally artificial viscosity is added to be able to resolve the strong gradients without widening of the shock.
The idea behind this indicator is to compare the polynomial coefficients of a modal hierarchical basis to each other. For smooth problems the polynomial coefficients are expected to decay very quickly. Thus, a large influence of the highest mode indicates an oscillating solution.
As a first step, the solution $\mathbf{U}$ inside one DG cell is transformed to a hierarchical modal basis using the modal basis functions Ψ

$$\mathbf{U} = \sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{k=0}^{N} \mathbf{U}_{ijk}\Psi_{ijk}\,. \tag{3.42}$$

In addition, a truncated interpretation of the same solution is considered $\bar{\mathbf{U}}$ that now containing all modes except the highest mode

$$\bar{\mathbf{U}} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \mathbf{U}_{ijk} \Psi_{ijk} \,. \tag{3.43}$$

The last step is now the calculation of the smoothness indicator that is defined within each DG cell

$$\mathrm{Ind}_{\mathrm{persson}} = \log \left\{ \frac{(\mathbf{U} - \bar{\mathbf{U}}, \mathbf{U} - \bar{\mathbf{U}})}{(\mathbf{U}, \mathbf{U})} \right\} , \tag{3.44}$$

where $(\cdot, \cdot)$ is the standard inner product in $\mathbb{L}_2$. Hence, the indicator measures the influence of the highest modal mode of the solution. A large influence suggests an oscillatory solution.

4 Interface tracking method

One difficulty arising in the computation of multi-phase flows with a sharp-interface approximation is that the phase interface has to be tracked. An estimate of the interface geometry is needed for the inclusion of surface tension effects as well as for the numerical resolution of the phase interface. Within the sharp-interface context, the interface geometry is needed for the application of the generalized Riemann problems at the interface position. In order to include surface tension effects, the interface curvature has to be approximated. That is the second derivative of the interface geometry. Thus, an approximation implies the usage of a high order discretization or an alternative computation method that avoids the explicit calculation of the curvature. All methods have in common that an additional transport equation for the interface position has to be solved. The interface is defined by an iso-contour or special marker particles depending on the interface tracking method. The function or particles are transported by a velocity field that is here defined by the interface Riemann solver. Another choice is to use the velocity field in case no phase transfer effects are resolved.
Different methods are found in literature that approximate the interface geometry. In incompressible multi-phase solvers often Volume of Fluid (VOF) methods (Hirt and Nichols [42]) are used as they can be formulated in a mass-conservative way for a divergence-free velocity field. This method approximates the volume fraction of one grid cell. The resulting interface can be reconstructed using the approximated volume fraction and the definition of an iso-level, typically $\alpha = 0.5$. The main drawback is that the interface approximation is piecewise constant. A higher order reconstruction of the interface is possible, however, the approximation of the piecewise constant volume fraction is suited for the use in finite-volume schemes, but not for high-order DG schemes. The high-order treatment on the other hand is useful for the calculation of secondary interface values, such as the interface curvature (for more information see section 4.2).

Another type is the front tracking methods as developed by Unverdi and Tryggvason [98] and Terashima and Tryggvason [94]. In this method the interface is represented by marker particles that move with the fluid velocities. At each time step, the position of the marker particles is updated using the time integration scheme of the flow solver. The interface is then defined by the connection of the marker particles. They use a bilinear interpolation to interpolate the velocity to the front.

The last interface tracking method discussed here[1] is the level-set method. This method was originally introduced by Osher and Sethian [69] in which they introduce the approximation of the interface based on a signed continuous distance function. This method was further refined by Sussman and coworkers [89] for the simulation of incompressible multi-phase flow. A summary of level-set methods and its application is given in the review article of Sethian and Smereka [83]. In contrast to the VOF method, the phase interface is defined by the zero iso-contour of the continuous level-set function that is initialized as a signed distance function. This interface approximation simplifies the discretization using a high-order method as well as the calculation of secondary interface quantities. Because of this, the level-set method is chosen in the following for the interface approximation.

A short remark about the mass-conservation of the interface tracking methods: Numerically this is a huge problem and due to numerical inaccuracies mass might get lost. In contrast to the incompressible case, there the VOF-method can be formulated such that it is mass-conservative, none of the methods is mass-conservative in a compressible flow field. In order to limit the numerical diffusion for the level-set transport, the level-set function is reconstructed to a (cropped) distance function. This process is called level-set reinitialization. For the level-set method, the mass loss correlates with the fineness of the interface resolution and the shape of the level-set function. The finer the interface resolution, the smaller the mass loss that is mainly introduced by the level-set reinitialization method (see section 4.3). One strategy is therefore to avoid the level-set reinitialization totally (or partly) with the use of extension velocities according to Peng et al. [72]. They showed that through the use of a special velocity extrapolation strategy the reinitialization can be avoided or at least its frequency can be reduced drastically. This strategy is outlined in section 4.4 and adapted to the use in the compressible framework.

[1] Many more different types exist, but these are the most popular ones. Also combinations if the described tracking methods exist, e. g. a combination of the level-set and VOF method.

The extrapolation strategy for the interface velocity is essential in case phase transition is considered. In this case the fluid velocity at the interface is not equal to the interface velocity as a velocity jump term is present. The advection velocity of the interface is solely given at the interface position as it is determined by the interface Riemann solver (see section 6) as phase velocity. Thus, a numerical method is needed to extrapolate the advection velocity into the volume. In order to avoid negative impacts on the level-set distribution, the extrapolation strategy according to Peng et al. [72] is used. They extrapolate the velocity field along the level-set normals.
In the last section of this chapter two validation test cases of the level-set implementation are presented, Zalesak's rotating sliced disk and the deformation of the level-set according to a prescribed velocity field as induced by vortices. Additionally, the formal order of convergence for the level-set transport and reinitialization implementation is shown using generic test cases.

4.1 Numerical discretization of the level-set method

The level-set function Φ is initialized as (cropped) signed distance function whose zero iso-contour describes the location of the phase interface. The value of Φ is a measure for the distance of a point in the computational domain to the interface. As the interface position may vary during the computation, the level-set function has to be updated in each time step. Because of that an additional transport equation for the level-set function Φ is solved. The level-set function is advected using a special level-set advection velocity field $\boldsymbol{s}_{\mathrm{PB}}$. This velocity field is needed due to the resolution of phase transfer effects (this implies a non-unique interface velocity). The construction of $\boldsymbol{s}_{\mathrm{PB}}$ is detailed in section 4.4.
Starting from the original level-set advection equation as introduced by Sussman [89] and presented in equation (2.8), the equation is recast as a conservation equation. This step is needed due to the level-set discretization using the DGSEM method (see section 3.1). The reformulation adds an additional term that is non-zero for compressible flow. This leads to the first level-set formulation

$$\frac{D\Phi}{Dt} = \frac{\partial \Phi}{\partial t} + \nabla \cdot (\boldsymbol{s}_{\mathrm{PB}}\Phi) = \Phi \nabla \cdot (\boldsymbol{s}_{\mathrm{PB}}) \, . \tag{4.1}$$

Because of the RHS-term (source term) in equation (4.1), the gradient of the

level-set advection velocity has to be estimated. This gradient of the interface velocity can be approximated using a DG lifting procedure. Similar to the Navier-Stokes equations the level-set transport equation is discretized using the DGSEM method.
A simplified implementation of the numerical method is obtained by including the right hand side source term in a modified volume integral. This is done through a reformulation of (4.1). The DG discretized version of the level-set transport equation can be obtained from (3.5) using the test function ϕ and is formulated as

$$\int_E J(\boldsymbol{\zeta})\Phi_t\phi d\boldsymbol{\zeta} + \int_{\partial E} \left(\boldsymbol{s}_{\mathrm{PB}}\Phi \cdot \hat{\boldsymbol{n}}_\zeta\right)^* \phi dS - \int_E \left(\boldsymbol{s}_{\mathrm{PB}}\Phi\right)\cdot\nabla\phi d\boldsymbol{\zeta} = \int_E \left(\nabla\cdot\boldsymbol{s}_{\mathrm{PB}}\right)\Phi\phi d\boldsymbol{\zeta}\,. \tag{4.2}$$

This formulation can now be rearranged by means of a modified volume integral

$$\int_E J(\boldsymbol{\zeta})\Phi_t\phi d\boldsymbol{\zeta} + \int_{\partial E} \left(\boldsymbol{s}_{\mathrm{PB}}\Phi \cdot \hat{\boldsymbol{n}}_\zeta\right)^* \phi dS - \underbrace{\int_E \left(\left(\boldsymbol{s}_{\mathrm{PB}}\Phi\right)\cdot\nabla\phi + \left(\nabla\cdot\boldsymbol{s}_{\mathrm{PB}}\right)\Phi\phi\right) d\boldsymbol{\zeta}}_{\text{modified volume integral}} = 0\,, \tag{4.3}$$

that is a convenient formulation for the implementation as the right hand side term is included in the approximation of a modified volume integral. Numerically, both formulations are identical.
Other authors, including Grooss and Hesthaven [34], and Marchandise and Remacle [60], derived a DG formulation for the level-set advection equation, but neglected the RHS term as they considered an incompressible flow field. They showed the suitability and stability of the DG discretization for the level-set transport and reported promising results using the DG discretization.
The level-set distribution is only important within a small region around the interface where the geometry and the secondary interface quantities are defined. Outside this region the absolute value is not important. The sign of Φ determines the affiliation to the liquid or gaseous phase. To facilitate the numerical treatment concerning the boundary conditions and to reduce the computational costs, the peaks of the initial distance function $\bar{\Phi}$ are removed using the following smooth cropping procedure

$$\Phi = \begin{cases} \bar{\Phi} & \text{if } |\bar{\Phi}| \leq \Phi_{\max}, \\ p(\bar{\Phi}) & \text{if } |\bar{\Phi}| > \Phi_{\max}\,, \end{cases} \tag{4.4}$$

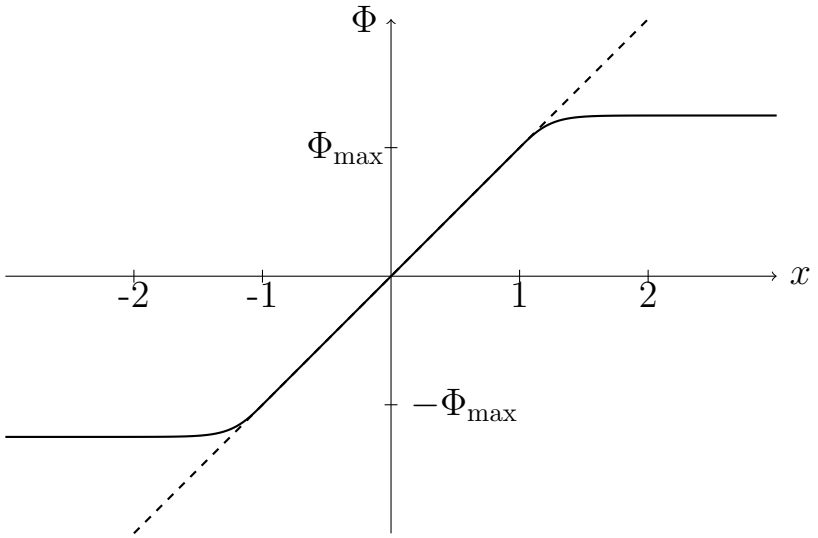


Figure 4.1: Smooth level-set cropping. The solid line is the cropped, the dashed line is the uncropped level-set function.

with the continuous cropping function

$$p(\bar{\Phi}) = \frac{1}{4}\Phi_{\max} \tanh\left(\frac{4}{\Phi_{\max}}\left(\bar{\Phi} - \operatorname{sign}(\bar{\Phi})\Phi_{\max}\right)\right) + \operatorname{sign}(\bar{\Phi})\Phi_{\max}\,. \tag{4.5}$$

The cropping function is visualized in figure 4.1. The level-set cropping level $\Phi_{\max}$ is a user-defined parameter, which is dependent of the droplet size and controls the width of the narrow band level-set approach, i. e. the number of DG cells inside the not cropped part and thereby the quality of the curvature estimation. To be able to achieve a reliable curvature estimation, at least 3–5 DG cells should be in the part that is not cropped. Marchandise and Remacle [60] reported that a direct evaluation of the curvature from the DG polynomials may lead to a "noisy" curvature field with an oscillating characteristic. This is due to creation of small jumps in the numerical approximation at the element boundaries that stabilize the scheme. However, due to the gradient estimation these disturbances are amplified. They proposed to use a least-square reconstruction of the curvature field to obtain a suitable approximation. A similar reconstruction method, that also increases the order of the curvature estimation, is described here based on the $\mathrm{P}_N\mathrm{P}_M$-method in section 4.2.1.
The smooth cropping applied here is chosen to maintain a smooth level-set distribution and reduces the negative impact of the cropping onto the curvature estimation. One should note that during the estimation of derivatives, small oscillations are amplified and may disturb the derived functions.

This cropping has two advantages: First, the level-set function is further smoothened and it is hence easier to handle using a high-order method, especially at the domain boundaries. Thus, the high-order polynomial tend to be smoother and it is advantageous for the estimation of the derivatives at the interface. Second, the cropping facilitates the handling of the narrow-band approach and the implementation of the boundary conditions. It is easily possible to include additional elements into the narrow band if the level-set function has moved and the level-set advection equation only has to be solved within a narrow band at the phase interface.

4.2 Approximation of secondary interface quantities: normals and curvature

Several approaches to estimate the secondary interface quantities at the position of the phase interface are available ranging from a direct curvature estimation using the well-known DG lifting procedure to approaches based on a robust WENO reconstruction of the curvature or a $\mathrm{P}_N\mathrm{P}_M$-reconstruction of the level-set. These approaches increase the overall accuracy of the curvature estimation and damp existing oscillations in the DG polynomial. The different approaches are compared in section 4.2.4.
The main differences are robustness and accuracy of the curvature estimation. A robust curvature estimation approach is e. g. needed in case two droplet collide with each other or getting very close (within one or two DG cells). In this case the level-set function may not be smooth and the $\mathrm{P}_N\mathrm{P}_M$-reconstruction and the DG lifting fail. The only choice to estimate a meaningful curvature value is to use a WENO reconstruction. In the following these three approaches to estimate the secondary interface quantities are presented and compared.

4.2.1 $\mathrm{P}_N\mathrm{P}_M$-reconstruction

To be able to get smooth approximations (the piecewise polynomial DG approximation may include discontinuities at the element boundaries) for the secondary interface values (such as the interface curvature κ), a $\mathrm{P}_N\mathrm{P}_M$-reconstruction [20, 21] up to the order $M = 3N_{\mathrm{DG}} + 2$ is used for the level-set polynomial. A central reconstruction stencil on a cartesian grid is used. The reconstructed polynomial has the property that it is smooth at the element

boundaries and the gradients can be evaluated locally for each cell. Due to the higher reconstruction order it is possible to enhance accuracy in the gradient estimation significantly, the polynomial is smoothed considerably and the jumps at the element boundaries have less influence. This facilitates the gradient estimation as a direct differentiation of the reconstructed polynomial is possible without using a numerical lifting procedure.
The $\mathrm{P}_N\mathrm{P}_M$ reconstruction operator for the level-set polynomial can be written as inverse of a $\mathbb{L}_2$ projection. The focus here is on the symmetric reconstruction with a reconstruction stencil of three grid cells in each axis direction with $k = -1, \ldots, 1$. Due to the choice of $M = 3N_{\mathrm{DG}} + 2$ the resulting projection matrix is quadratic and invertible. For choices of $N_{\mathrm{DG}} < M < 3N_{\mathrm{DG}} + 2$ a least square approximation is needed for the inversion of the matrix. Exemplary, the reconstruction is written for the ζ^1-direction in the reference element

$$\int_{-1}^{1} \overline{\Phi}\left(\zeta^1 + 2k\right) \varphi_i(\zeta^1) d\zeta^1 = \int_{-1}^{1} \Phi^k(\zeta^1)\varphi_i(\zeta^1) d\zeta^1, \quad i = 0, \ldots, N. \tag{4.6}$$

For the reconstruction of the level-set polynomial Φ the unknown function $\overline{\Phi}$ (the reconstructed level-set polynomial) is approximated; the polynomial data $\Phi^k(\zeta^1)$ is given. For the linear level-set function, the reconstruction operator can be written as reconstruction matrix-vector formulation

$$\mathbb{C}^k \hat{\overline{\Phi}} = \hat{\Phi}^k \tag{4.7}$$

with

$$\mathbb{C}^k_{ij} = \int_{-1}^{1} \overline{\varphi}_j(\zeta^1 + 2k)\varphi_i(\zeta^1) d\zeta^1 \,. \tag{4.8}$$

Here, $\varphi_i(\zeta^1)$ are the one-dimensional Lagrange basis functions in $[-1, 1]$. The reconstruction of the DOF is then given by the inverse of the matrix $\mathbb{C}$

$$\overline{\Phi} := \mathbb{C}^{-1}\Phi = \sum_{k=-1}^{1} \mathbb{C}_k \Phi^k \,. \tag{4.9}$$

Note that due to the linear level-set transport equation, the reconstruction matrix $\mathbb{C}$ is computed once in a preprocessing step. The reconstruction is only a matrix-vector operation.

Due to the reconstruction, the accuracy of the gradient estimation is significantly enhanced. The continuous reconstruction facilitates the gradient estimation as the jumps at the element boundaries have less influence and a direct differentiation of the reconstructed polynomial is possible.

4.2.2 Direct lifting of the gradients

Other approaches to estimate the interface curvature include the BR1-scheme of Bassi and Rebay [6] or the local DG (LDG) method of Cockburn and Shu [15]. These approaches calculate the gradients using a DG lifting procedure together with a parabolic flux at the DG cell interfaces. These approaches take the local discontinuous DG polynomial for the lifting into account and ensure a global coupling of the derivative calculation. Since the level-set polynomial is not reconstructed, the accuracy of the curvature is sufficiently lower and the formal order of convergence is two orders lower than the order of the ansatz polynomials.
The advantage of the this method is that an additional level-set reconstruction step is not needed. However, as the level-set polynomials tend to have jumps at the element boundaries the direct curvature estimation may tend to have oscillations.

4.2.3 WENO reconstruction of the curvature

Two different approaches to reconstruct the curvature are possible within the Weighted Essentially Non-Oscillatory (WENO) method. The first is the calculation of the curvature at each point using a finite-difference approximation and then a smooth WENO reconstruction of the curvature. The third method is the WENO reconstruction of the level-set function Φ where the polynomial WENO approximation for the curvature is used.
The WENO method for the high-order curvature estimation is the method of choice for complex level-set geometries as this is the most robust method of the three curvature estimation methods presented. In the following the description is for a 5^{th} order WENO scheme (as introduced in the context of the level-set reinitialization in section 4.3). The calculation of the polynomial smoothness coefficients and the reconstruction is based on the PyWENO framework [23]. The smooth WENO reconstruction is used to provide smoothness for the curvature estimation for cases where the level-set function may have kinks or dis-

continuities. Within the WENo method we restrict us on linear grids and do not consider curved geometries.
The reconstruction step is applied on the grid in the unit element $E \in [-1, 1]$. Thus, it is possible to construct this step on a regular grid without having to consider geometry information. All geometry information enters the description in terms of the Jacobian of the transformation.
Both approaches are described in the following:

1. For the first method of the WENO curvature reconstruction, as a first step the second derivatives of the level-set polynomial within the reconstruction stencil are estimated by a second-order finite difference formula

$$\Phi_{xx} = \frac{\Phi_{i-1} - 2\Phi_i + \Phi_{i+1}}{(\Delta x)^2}\,, \tag{4.10}$$

   with Δx being the interface width of the finite volume sub-cells and Φ_i the level-set value in the respective FV sub-cell. Note that due to the structured grid, the reconstruction is done for each direction separately.

   In a next step the curvature polynomial based on the three ENO stencils is reconstructed, the smoothness coefficients of the polynomials are computed and their value at the barycenter of the center cell in the reconstruction stencil is evaluated. This approach ensures a smooth curvature distribution with a convergence order that coincides with the formal convergence order of the WENO scheme. Due to the finite-volume approximation of the curvature, the reconstruction stencil is enlarged by two points compared to traditional WENO methods. Another difficulty is that the initial curvature estimation based on the finite-difference formula already has a stencil of three points. This stencil can lead to problems in case kinks or discontinuities appear in the level-set solution. Most of these difficulties are avoided due to the use of the level-set reinitialization (see section 4.3) but in case of colliding droplets such kinks may occur that lead to a non-physical curvature estimation.

2. The second method of estimating the interface curvature uses the inverse procedure. Here, the level-set polynomial Φ is reconstructed using the WENO method. This means that the smoothness coefficients and weights of the WENO polynomial are estimated based on Φ. The interface curvature κ is then approximated by the second derivative of the

WENO polynomial using the weights obtained from the reconstruction of the level-set polynomial.

This method ensures a smooth reconstruction of the level-set polynomial in all cases. The difficulty is that the derivatives of the WENO polynomial may not be smooth in all cases. Another problem is the lower accuracy compared to the first WENO reconstruction approach as the formal convergence order is lower due to the analytical derivation of the polynomial. To achieve a similar accuracy compared to the first approach a larger stencil for the WENO reconstruction can be used (higher WENO reconstruction). This approach can handle kinks in the level-set function Φ more easily compared to the first WENO reconstruction approach due to the direct reconstruction of Φ (and not its second derivative).

Altogether, the WENO curvature estimation method is a robust method to calculate the curvature. However, in some cases this method is not robust enough to ensure a reliable curvature estimation, e. g. in case of colliding droplets. Then, special reconstruction methods or special methods have to be used to ensure a reliable curvature estimation. This is, e. g., discussed by Tanguy et al. in [93]. This method is based on the introduction of additional cells at the level-set kink for the curvature estimation.

4.2.4 Comparison of the curvature estimation methods

In table 4.1 the accuracy of the curvature estimation for the sample case of a spherical droplet is investigated. The $\mathbb{L}_2$ error of the DG approximation is compared to the analytical curvature in the vicinity of the interface. For the case with $\mathrm{P}_N\mathrm{P}_M$ reconstruction a higher convergence order of the curvature is reached than one would gain using direct differentiation of the level-set ansatz polynomials. In the case here investigated with a polynomial degree $N = 3$ one would expect a second order convergence for the curvature. The $\mathrm{P}_N\mathrm{P}_M$ reconstruction reduces the overall error in the curvature estimation considerably due to the higher convergence order.
In case of the coarsest grid, the discretization is too coarse to resolve the level-set function and, hence, the discretization errors are rather large for the curvature. A minimum of 20 grid cells in each axis direction is necessary to resolve the curvature properly.
In the following, the $\mathrm{P}_N\mathrm{P}_M$ reconstruction method is chosen for the computation of the curvature. This method provides an accurate, nearly oscillation-

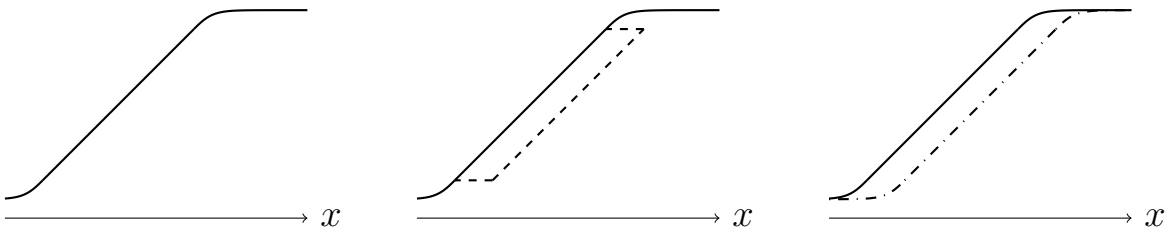
# Grid cells	$P_N P_M$		BR1		WENO	
	$\mathbb{L}_2$ error	order	$\mathbb{L}_2$ error	order	$\mathbb{L}_2$ error	order
10	3.923 E-01	-	4.878 E-00	-	4.770 E-01	-
20	5.164 E-02	2.92	1.586 E-01	4.94	7.731 E-02	2.60
40	1.228 E-03	5.39	1.895 E-02	3.06	5.021 E-03	3.94
80	5.254 E-05	4.55	2.099 E-03	3.17	6.881 E-03	0.45

Table 4.1: $\mathbb{L}_2$ error of the curvature approximation (second derivative of the level-set function Φ) and the corresponding approximation order for a spherical droplet with radius 0.4 and a fourth order DG scheme.

free approximation of the curvature for "typical" applications within the field of droplet dynamics. For the case of droplet collisions and coalescence locally more robust approaches for the curvature estimation, e. g. the described WENO method, have to be used.

4.3 Level-set reinitialization

In order to keep the level-set function close to a signed distance function in the vicinity of the phase interface, a reinitialization procedure is needed. In principle, two mechanisms necessitate the use of a reinitialization procedure: The first is a numerical consideration as it is numerically advisable to have a level-set function that is close to a signed distance function to limit the numerical diffusion. A further advantage is that this aides the accurate computation of the derivatives, especially for the curvature calculation. Due to the consideration of a compressible flow field, physical phenomena like shocks may be present in the flow field that can severely deform the level-set function. This deformation introduces additional oscillations that may lead to unwanted additional roots of the level-set polynomial and, thereby, to unphysical new locations of the phase interface. Another source of level-set disturbances is the non-constant advection velocity field used for the level-set transport. This introduces small jumps in the level-set function as visualized in figure 4.2. Already small disturbances in the level-set polynomial may lead to large inaccuracies in the curvature calculation, as the errors are amplified by the estimation of the secondary interface properties (e. g. curvature calculation).



(a) $t = 0$ (b) $t = 1$ (c) $t = 1$ (after reinit)

Figure 4.2: Visualization of the level-set advection using a non-constant velocity field. The initial smooth level-set function is advected until $t = 1$. Due to the advection disturbances next to the level-set cropping limit, e. g. discontinuities, jumps, are introduced that are removed by the level-set reinitialization. The solid line is the initial level-set distribution, the dashed line the level-set distribution at $t = 1$ before the reinitialization and the dash-dot line the level-set at $t = 1$ after the reinitialization.

This mechanism is circumvented through the use of a frequent reinitialization procedure that resets the level-set function to a preferable shape (distance function next to the interface). The step is needed for consideration of colliding or merging bubbles and even for the simple case of advected bubbles it is not guaranteed that the level-set function remains numerically advantageous for all times. This becomes even more crucial if the interface curvature, which is the second derivative of the level-set function, is considered. Typically, the reinitialization procedure is called approximately every 100 time steps depending on the test case. Due to the use of an extrapolated velocity field along the level-set normals the deformation of Φ is minimized. This is advantageous as mass loss is typically accompanied with frequent reinitialization. Approaches to correct the leading order term of the mass loss are discussed in section 4.3.4.

For the reinitialization of the level-set function two different methods are available. The first is an iterative root finding algorithm that uses an adaptive refinement in the vicinity of the level-set root. The refinement is needed to provide the interface location up to a specified accuracy. With the knowledge of the phase interface position in space, it is possible to calculate the updated distance function. The second method is an iterative reinitialization procedure as introduced by Sussman et al. [89]. In the following the focus is on the iterative reinitialization procedure.

4.3.1 Iterative reinitialization procedure

This iterative procedure, as introduced by Sussman et al. [89], is based on the solution of the following Hamilton-Jacobi equation

$$\begin{aligned} \Phi_t &= S_\epsilon\left(\Phi_0\right)\left(|\nabla\Phi| - 1\right), \\ \Phi\left(\mathbf{x}, 0\right) &= \Phi_0\left(\mathbf{x}\right), \end{aligned} \tag{4.11}$$

where S_ϵ represents a sign function. It has been shown by several authors, e. g. in [89], to be advantageous to use a numerically smeared sign function instead of the sharp sign function. This formulation avoids the explicit search for the root of the level-set function and the reset of the level-set function close to the phase front, which requires $\mathrm{O}(N_{\mathrm{DOF}}^3)$ operations. The numerical sign function is defined in the following way

$$S_\epsilon\left(\Phi_0\right) = \frac{\Phi_0}{\sqrt{\Phi_0^2 + \epsilon}} \tag{4.12}$$

and is only dependent on the initial condition Φ_0 of the pseudo-iteration. The smoothening term ϵ is chosen according to the minimum grid length in the computational domain. Equation (4.11) is iterated until the pseudo-time τ reaches the level-set crop limit $\Phi_{\max}$ ensuring that Φ is a distance function in the interval $[-\Phi_{\max}, \Phi_{\max}]$. The reconstructed level-set function ensures that the zero-value remains unchanged as the root of the level-set function represents physically the position of the phase interface.
In order to maintain the level-set restriction introduced in (4.4) the level-set reinitialization equation is modified in the following way

$$\Phi_t = S_\epsilon\left(\Phi_0\right)\left(|\nabla\Phi| - l\left(\Phi\right)\right), \tag{4.13}$$

with the l-function defined as follows

$$l(\Phi) = \begin{cases} 1 & \text{if } |\Phi| < \Phi_{\max}, \\ 0 & \text{if } |\Phi| > {}^5\!/_4\,\Phi_{\max}, \\ p^*(\Phi) & \text{in between.} \end{cases} \tag{4.14}$$

Again, $\Phi_{\max}$ is the cropping limit of the level-set function. This function $l(\Phi)$ is chosen such that it efficiently forces the level-set gradient in equation (4.11) to be 0 if $|\Phi| > {}^5\!/_4\Phi_{\max}$ and to be 1 if $|\Phi| < \Phi_{\max}$ with a smooth transition in between. This smooth transition is provided by the p^*-function

$$p^*(\Phi) = -64\Phi^3 + 192\Phi^2 - 192\Phi + 65 \tag{4.15}$$

that ensures a two times differentiable transition region. It would also be possible to use the original reinitialization equation and apply the smoothening described in (4.4). However, this method has the disadvantage that the extrema get smaller if the reinitialization is not carried out to the full extent until the end of the computational domain.
In the here considered DG solver framework, the solution is first projected in the whole domain onto the equidistant sub-cell grid that is already used for the interface resolution. Then, a scheme to iterate the modified reinitialization equation (4.13) to steady state resulting in a piecewise constant level-set description on the sub-cell grid is used. The final solution is then reconstructed onto the DG grid with piecewise polynomial description. The challenge is to tune the scheme such that also the second derivative at the interface is smooth, as this value is needed for the inclusion of surface tension effects. Thus, the interface curvature has a direct effect on the flow solution.

4.3.2 Numerical method

In contrast to the Navier-Stokes equations, the reinitialization procedure is of Hamilton-Jacobi type. Due to the different numerical solution strategy for this type of equations the DGSEM method can not be applied directly. Numerical methods for the Hamilton-Jacbobi equation are based on the solution of conservation equations for the derivatives as well as the application of a suitable numerical flux that is based on the estimation of a numerical Hamiltonian. These are the two building blocks for the discretization of the level-set reinitialization.
Equation (4.11) is discretized using a 5th order WENO scheme developed by Jiang and Peng [45]. A DG based version for discretization of (4.11) as proposed by Yan and Osher [105] was not stable for the reinitialization, possibly because of the exhibition of discontinuous derivatives.

4.3.2.1 WENO approximation

Numerical methods for the Hamilton-Jacobi are based on the conservation of the gradients and a numerical Hamiltonian H for the time-update of the pseudo-iteration (4.11) is introduced. This equation can be formulated using a general

numerical Hamiltonian $H(\boldsymbol{x}, t)$

$$\begin{aligned} \Phi_t + H\left(\boldsymbol{x}, t, \Phi, \nabla\Phi\right) &= 0 \\ \Phi(x,0) &= \Phi_0(x), \end{aligned} \tag{4.16}$$

which can be reduced to the viscosity solution of (4.16) (as determined by Crandall and Lions [16]). They introduce an important class of monotone schemes with a simplified form

$$\Phi_t + H\left(\nabla\Phi\right) = 0 \tag{4.17}$$

and proved that this scheme converges to the viscosity solution. Note that the solution of the Hamilton-Jacobi equation is smooth but may exhibit discontinuous derivatives. Different fluxes can be used for the discretization including a simple Lax-Friedrichs diffusion flux and Godunov-type fluxes. All these fluxes have in common that an approximation of the gradient on the left element side Φ^- and on the right element side Φ^+ is needed. The aim of a numerical method is hence to approximate these derivatives using a numerical method.
For the high-order WENO reconstruction, a left-biased stencil x_k, $k = i-3, \ldots, i+2$ and a right-biased stencil $x_k, k = i-2, \ldots, i+3$ is defined as shown in figure 4.3. In the following the reconstruction is described in a 1D manner, but this is sufficient as only structured grids are considered here. The estimation of the derivatives is taken from Jiang and Peng [45].
The derivative using the left-biased stencil $\Phi^-_{x,i}$ can be written using a WENO-limiter function Ψ_{WENO} as

$$\begin{aligned} \Phi^-_{x,i} = \frac{1}{12}\left(-\frac{\Delta^+\Phi_{i-2}}{\Delta x} + 7\frac{\Delta^+\Phi_{i-1}}{\Delta x} + 7\frac{\Delta^+\Phi_i}{\Delta x} - \frac{\Delta^+\Phi_{i+1}}{\Delta x}\right) \\ - \Psi^{WENO}\left(\frac{\Delta^-\Delta^+\Phi_{i-2}}{\Delta x}, \frac{\Delta^-\Delta^+\Phi_{i-1}}{\Delta x}, \frac{\Delta^-\Delta^+\Phi_i}{\Delta x}, \frac{\Delta^-\Delta^+\Phi_{i+1}}{\Delta x}\right) \end{aligned} \tag{4.18}$$

and using the right-biased stencil

$$\begin{aligned} \Phi^-_{x,i} = \frac{1}{12}\left(-\frac{\Delta^+\Phi_{i-2}}{\Delta x} + 7\frac{\Delta^+\Phi_{i-1}}{\Delta x} + 7\frac{\Delta^+\Phi_i}{\Delta x} - \frac{\Delta^+\Phi_{i+1}}{\Delta x}\right) \\ + \Psi^{WENO}\left(\frac{\Delta^-\Delta^+\Phi_{i-1}}{\Delta x}, \frac{\Delta^-\Delta^+\Phi_i}{\Delta x}, \frac{\Delta^-\Delta^+\Phi_{i+1}}{\Delta x}, \frac{\Delta^-\Delta^+\Phi_{i+2}}{\Delta x}\right). \end{aligned} \tag{4.19}$$

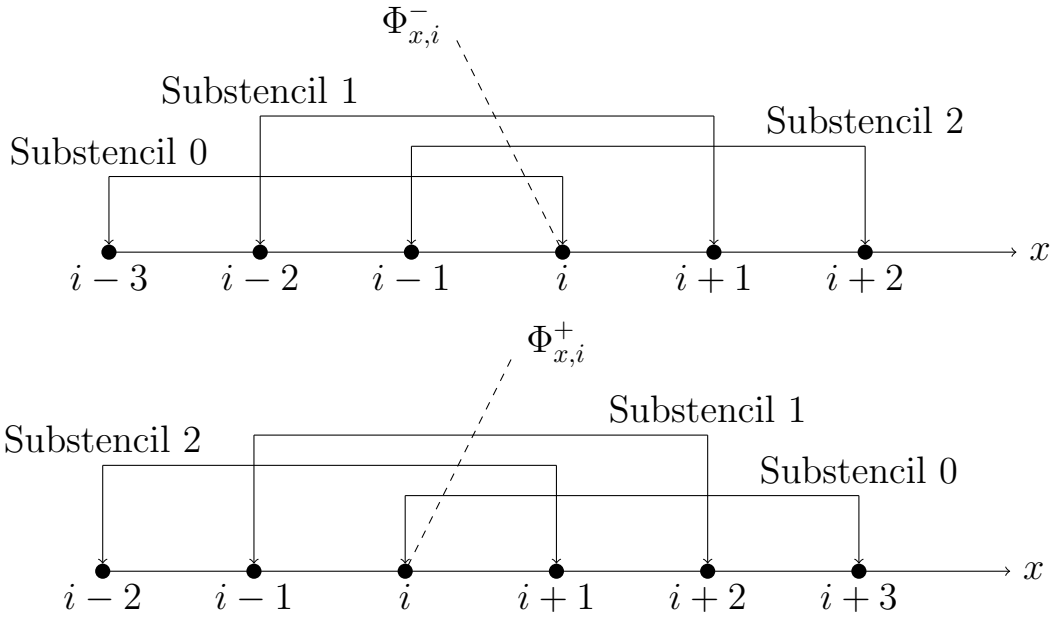


Figure 4.3: Left- and right-biased stencil for the WENO scheme.

Here, the abbreviations

$$\Phi_k = \Phi(x_k), \qquad \Delta^+\Phi_k = \Phi_{k+1} - \Phi_k, \qquad \Delta^-\Phi_k = \Phi_k - \Phi_{k-1},$$

are used.
The WENO limiter can now be defined as

$$\Psi^{WENO}(a,b,c,d) = \frac{1}{3}\omega_0(a-2b+c) + \frac{1}{6}\left(\omega_2 - \frac{1}{2}\right)(b-2c+d) \tag{4.20}$$

with the following weights defined as

$$\omega_0 = \frac{\alpha_0}{\alpha_0+\alpha_1+\alpha_2}, \quad \omega_2 = \frac{\alpha_2}{\alpha_0+\alpha_1+\alpha_2};$$

$$\alpha_0 = \frac{1}{(\epsilon+IS_0)^2}, \quad \alpha_1 = \frac{6}{(\epsilon+IS_1)^2}, \quad \alpha_2 = \frac{3}{(\epsilon+IS_2)^2};$$

$$\begin{aligned} IS_0 &= 13(a-b)^2 + 3(a-3b)^2, \\ IS_1 &= 13(b-c)^2 + 3(b-c)^2, \\ IS_2 &= 13(c-d)^2 + 3(3c-d)^2. \end{aligned}$$

Analog, the approximations in y-direction Φ^+_y and Φ^-_y as well as in z-direction Φ^+_z and Φ^-_z can be obtained.

4.3.2.2 Numerical Fluxes

Based on the approximation of the derivatives Φ^+ and Φ^- in the respective axis directions a numerical flux for the Hamilton-Jacobi equation can be defined. There are several flux choices available, including Lax-Friedrichs and Godunov-type fluxes. For the level-set reinitialization equation, the numerical Hamiltonian H can be written as

$$H\left(\Phi, \Phi_x, \Phi_y, \Phi_z\right) = S_\epsilon\left(\sqrt{\Phi_x^2 + \Phi_y^2 + \Phi_z^2} - l(\Phi)\right). \tag{4.21}$$

In the following an overview of common flux approximations is presented:

1. The Lax-Friedrichs (LF) flux

$$\begin{aligned} H_{LF}(\Phi_x^+, \Phi_x^-, \Phi_y^+, \Phi_y^-, \Phi_z^+, \Phi_z^-) = H&\left(\frac{\Phi_x^+ + \Phi_x^-}{2}, \frac{\Phi_y^+ + \Phi_y^-}{2}, \frac{\Phi_z^+ + \Phi_z^-}{2}\right) \\ &- \max|H_x(\Phi_x^+, \Phi_x^-)|\frac{\Phi_x^+ - \Phi_x^-}{2} \\ &- \max|H_y(\Phi_y^+, \Phi_y^-)|\frac{\Phi_y^+ - \Phi_y^-}{2} \\ &- \max|H_z(\Phi_z^+, \Phi_z^-)|\frac{\Phi_z^+ - \Phi_z^-}{2}, \end{aligned} \tag{4.22}$$

   with H_d being the partial derivative of H with respect to Φ_d with $d \in \{x, y, z\}$. The global Lax-Friedrichs scheme can be obtained if the global maxima of the H_d terms are chosen. Correspondingly, the local Lax-Friedrichs scheme is derived if only the local values are taken into account. The global Lax-Friedrichs scheme is very dissipative and leads to a large mass loss during reinitialization, while the local Lax-Friedrichs flux has considerably lower dissipation properties.

2. Godunov-type flux

$$\begin{aligned} &H^G(\Phi_x^+, \Phi_x^-, \Phi_y^+, \Phi_y^-, \Phi_z^+, \Phi_z^-) = \\ &S_\epsilon\begin{cases} \{[\max((\Phi_x^+)^-, (\Phi_x^-)^+)]^2 + [\max((\Phi_y^+)^-, (\Phi_y^-)^+)]^2 + \\ \quad [\max((\Phi_z^+)^-, (\Phi_z^-)^+)]^2\}^{0.5} - l(\Phi) & \text{if } \Phi > 0 \\ \{[\max((\Phi_x^+)^+, (\Phi_x^-)^-)]^2 + [\max((\Phi_y^+)^+, (\Phi_y^-)^-)]^2 + \\ \quad [\max((\Phi_z^+)^+, (\Phi_z^-)^-)]^2\}^{0.5} - l(\Phi) & \text{otherwise.} \end{cases} \end{aligned} \tag{4.23}$$

Hereby, the following abbreviations are used: $(a)^+ = \max(a, 0)$ and $(a^-) = -\min(a, 0)$. Note the Godunov-type flux is often complicated for general Hamilton-Jacobi equations [5, 70]. Just for the level-set reinitialization equations, it takes the above simple form.

4.3.3 Time stepping

For the iteration, as presented in equation (4.11), a time-stepping scheme is needed. The time-accurate resolution of the reinitialization process is not important, a simple first order time stepping scheme is used that helps to damp out occurring oscillations.
The time step for the pseudo-iteration is constant (due to the linearity of the reinitialization) and is estimated as follows. It is based on the DG time step estimation, as presented in section 3.1.6, but the advection velocities are set constant due to the linear equation. The maximum advection velocity is set to $v_{\text{reinit}} = 1$ in all axis directions as well as the sound speed $c_{\text{reinit}} = 1$. Note that this is a conservative approximation of the time step.
This leads to the approximation of the maximum speed

$$\lambda_{\text{reinit}} = v_{\text{reinit}} + c_{\text{reinit}} = 2\,. \tag{4.24}$$

After the transformation in the unit element, the maximum speed can be written using the metric terms $J\mathbf{a}_n^i$ in the following way

$$\tilde{v}^i_{\text{reinit}} = \sum_{n=1}^{3} J\mathbf{a}_n^i \mathbf{v}^i_{n,\text{reinit}}, \qquad \tilde{c}^i_{\text{reinit}} = c_{\text{reinit}} \sqrt{\sum_{n=1}^{3} (J\mathbf{a}_n^i)^2}. \tag{4.25}$$

The metric terms $J\mathbf{a}_n^i$ describe the transformation from the unit cell to the physical cell. The eigenvalues needed for the CFL time step restriction can be estimated using this metric terms. For the eigenvalue λ_{reinit} in the reference element one gets

$$\Delta t = \frac{2}{\frac{1}{J}\sum_{i=1}^{3} \lambda_{\text{reinit}}} = \frac{J}{3}. \tag{4.26}$$

4.3.4 Level-set correction

Approaches of Sussman and Fatemi [88] and Hartmann et al. [39] are available in literature to correct the leading order term of the mass loss associated with

the level-set reinitialization procedure. Both approaches are based on a correction of the reinitialization flux such that the leading order term of the mass loss is eliminated during reinitialization.
The disadvantage of this approach is that the flux modification, applied once every reinitialization pseudo time step, is disturbing the curvature distribution in the flow field. Therefore, the level-set correction approaches are not considered in the following. The advantage of the high-order discretization is that the mass loss accompanied with level-set transport and reinitialization is low. Due to the choice of the Godunov-type flux for the reinitialization scheme, the mass loss is significantly reduced compared to the Lax-Friedrichs flux.

4.4 Creation of level-set advection velocities

The movement of the level-set function represents the deformation of the physical droplet. Therefore, the level-set function is advected by a velocity field $\boldsymbol{s}_{\mathrm{PB}}$ that is determined by the micro-scale solver (see section 6). The main difficulty is that this velocity field is only defined at the phase interface, the level-set advection velocity though is needed in the volume. This is visualized in figure 4.4. The objective is now to construct an interface advection velocity field such that the level-set function remains close to a signed distance function.
Malladi et al. [59] introduced the idea of the creation of a level-set advection velocity. They used the closest point on the front and extrapolated the velocity to the volume in the context of shape sedimentation. This approach was then generalized to a method to create extension velocities using an iterative method by Chen et al. [14] and Adalsteinsson and Sethian [2].
A level-set advection velocity can be obtained by extending the local interface velocity constant in normal direction of the level-set function. The extended velocity field $\mathbf{s}_{\mathrm{ext}}$ has to fulfill

$$\left(\nabla \mathbf{s}_{\mathrm{ext}}\right)^{\mathrm{T}} \mathbf{n}_{\mathrm{LS}} = \left(\nabla \mathbf{s}_{\mathrm{ext}}\right)^{\mathrm{T}} \nabla \frac{\Phi}{|\nabla \Phi|} = 0\,. \tag{4.27}$$

One way to create these advection velocities is to use the following pseudo-iteration for the advection velocity field $\mathbf{s}_{\mathrm{ext}}$

$$\frac{\partial \mathbf{s}_{\mathrm{ext}}}{\partial \tau} + \nabla \mathbf{s}_{\mathrm{ext}} \nabla \mathbf{n}_{\mathrm{LS}} = 0 \tag{4.28}$$

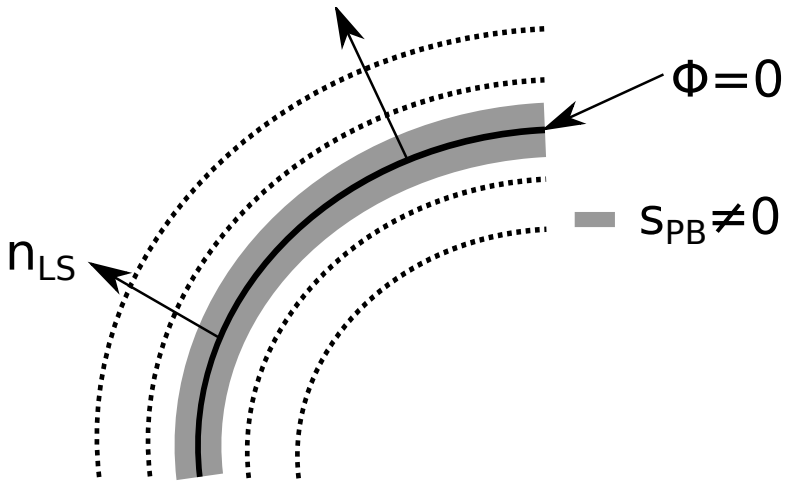


Figure 4.4: Schematic view of the creation of the level-set advection velocity field. The gray region represents the area next to the interface position in which the level-set velocity is defined by the interface Riemann problem. Starting from this narrow region the velocity field is extrapolated along the level-set normals to provide minimal disturbances to the level-set function.

assuming constant level-set normals $\mathbf{n}_{\mathrm{LS}} = \nabla \frac{\Phi}{|\nabla \Phi|}$. Similar to the reinitialization procedure (see section 4.3), a pseudo time stepping with the pseudo-time τ is used to gain a global velocity field. The extension is done separately for each velocity component of $\mathbf{s}_{\mathrm{PB}}$. A local Lax-Friedrichs flux is used for the underlying Hamilton-Jacbobi equation (4.28).

The numerical scheme used is identical to the scheme for the level-set reinitialization procedure (see section 4.3). The only difference is the used numerical flux and that the creation of the extension velocity is done separately for each velocity component. The numerical Hamilton H_{ext} can be written in terms of the level-set gradients $\mathbf{n}_{\mathrm{LS}}$ and the i-the component of the extension velocity $F_{\mathrm{ext,i}}$,

$$H_{\mathrm{ext},i}(\mathbf{s}_{\mathrm{ext},i,x}, \mathbf{s}_{\mathrm{ext},i,y}, \mathbf{s}_{\mathrm{ext},i,z}) = S_\epsilon \left(\mathbf{s}_{\mathrm{ext},i,x} n_{x,\mathrm{LS}} + \mathbf{s}_{\mathrm{ext},i,y} n_{y,\mathrm{LS}} + \mathbf{s}_{\mathrm{ext},i,z} n_{z,\mathrm{LS}} \right) . \tag{4.29}$$

With the choice of the numerical Hamiltonian, the WENO scheme for level-set reinitialization (see section 4.3) is used to obtain the velocity field.

4.5 Level-set validation examples

The implementation of the level-set is validated using two test cases taken from literature, Zalesak's disk [107] and the deformation of the interface according to a prescribed velocity field (LeVeque [58]). In both cases the velocity field is prescribed by an analytical function such that the velocity extrapolation step (see chapter 4.4) is not necessary. Both test cases incorporate a three-dimensional transport of the level-set function. The numerical dissipation of the DG level-set method is assessed by comparing the volume loss and the smearing of an initially sharp interface edges.
In all test cases, a fourth order DG method is used to discretize the test cases. The number of elements is varied to show the grid independence of the solution. The two-dimensional examples were calculated by extending the grid in z-direction with one grid cell and apply symmetry boundary conditions. The area/volume has been estimated using a first order accurate method in ParaView. Hence, the accuracy of the mass loss calculation is not very high, but has has been increased by applying a super-sampling of the DG solution for the rendering in ParaView.

4.5.1 Convergence test

In this test case the method of manufactured solutions is used to investigate the order of convergence for the level-set method. This is done in a separate test case as the investigated level-set test cases have a non-smooth level-set distribution such that the full convergence order can not be seen.
An analytical solution is calculated for a planar wave example with periodic boundary conditions. The level-set function is transported using the fluid velocity. An sinusoidal density and level-set distribution is assumed that is advected with a constant velocity of 1. Using the mathematical software Maple a source term was calculated to achieve a manufactured solution.
The initial solution is set to

$$\begin{aligned} \rho &= 2 + \hat{a}\sin(\omega(x_1 + x_2 + x_3) - 2v_{\text{adv}}\pi t)\,, \\ v_1 &= 1\,, \\ v_2 &= 1\,, \\ v_3 &= 1\,, \\ e &= 2 + \hat{a}\sin(\omega(x_1 + x_2 + x_3) - 2v_{\text{adv}}\pi t)\,, \\ \Phi &= 2 + \hat{a}\sin(\omega(x_1 + x_2 + x_3) - 2v_{\text{adv}}\pi t)\,, \end{aligned} \tag{4.30}$$

DG cells	DOF	ρ		Φ	
		$\mathbb{L}_2$ error	Order	$\mathbb{L}_2$ error	Order
10	40	1.5218509E-04	–	1.0986066E-04	–
20	80	6.4248290E-06	4.56	7.1766720E-06	3.93
40	160	3.5962203E-07	4.16	4.4471995E-07	4.01

Table 4.2: $\mathbb{L}_2$-Errors and convergence order for the flow solver with level-set advection equation using the method of manifactured solutions at $t = 1$ for a polynomial degree of $N = 3$.

with $\hat{a}$ being the amplitude of the planar wave and v_{adv} the advection velocity. The analysis of the approximation errors is done using a higher polynomial degree. The results of the convergence analysis are presented in table 4.2. The level-set implementation does get the optimal order of convergence for the DG scheme.

4.5.2 Rigid body rotation of Zalesak's disk

This test case, as introduced by Zalesak [107], describes the rigid body rotation of a slotted disk in a prescribed constant velocity field and assesses the numerical dissipation of the level-set transport. The exact solution for this transport problem is known.
In the first part the original test case is considered that is then extended to a 3D test case by considering a slotted sphere instead of a slotted disk.

4.5.2.1 2D rotation

Initially, the sphere has a radius of 0.15 and is located at (0.5,0.75) within the computational domain of $[0, 1]^2$. The slot has a width of 0.05. Due to the prescribed velocity field, the slotted disk rotates in the x-y-plane with a constant velocity. The velocity field is given by

$$\begin{aligned} v_1(x, y) &= \pi\,(0.5 - y)\ , \\ v_2(x, y) &= \pi\,(x - 0.5)\ , \end{aligned} \tag{4.31}$$

such that one rotation is completed in two time units. A measure of the dissipation of the numerical scheme is the smearing of the interface and the associated

DG cells	DOF	Area	% area loss	$\mathbb{L}_2$ error	Order
16	64	0.0494792	14.98	1.0984064E-03	–
32	128	0.0542535	6.78	3.5957768E-04	1.61
64	256	0.0561659	3.50	1.2713903E-04	3.11

Table 4.3: Errors and mass loss after one complete rotation of 2D Zalesak's disk compared to the approximated volume at $t = 0$ for a DG scheme with $N = 3$.

mass loss due to the transport of the interface. Due to numerical dissipation the initially sharp edges of the slotted disk are rounded as the disk is rotated. The errors and area losses after one complete rotation are shown in table 4.3. Note however that the mass loss is estimated using a first order method. Thus, the estimation may be too large. In figure 4.5 the interface position is shown after 1, 5 and 10 complete rotations of the disk highlighting the initially sharp edges of the disk. Note that the DG order is not recovered due to an initially non-smooth level-set distribution that is needed to describe the geometry of the slotted disk. The non-smooth part is located in the region of the slot, especially in the sharp end region of the slot.

4.5.2.2 3D rotation

For a complete 3D test case, the slotted disk of the two-dimensional test case is extended to a 3D slotted sphere. The rotation velocity in z-direction is set to zero resulting in a rotation of the slotted sphere in the x-y-plane.
Initially, the sphere has a radius of 0.15 and is placed at (0.5,0.75,0.5) within the computational domain of $[0, 1]^3$. The slot has a width of 0.05. Due to the prescribed velocity field, the sphere rotates along the parallel to the z-axis through (0.5,0.5,0). The constant velocity field is given as

$$\begin{aligned} v_1(x, y, z) &= \pi\,(0.5 - y)\ , \\ v_2(x, y, z) &= \pi\,(x - 0.5)\ , \\ v_3(x, y, z) &= 0\,, \end{aligned} \tag{4.32}$$

similar to the two-dimensional case. A measure of the dissipation of the numerical scheme is the smearing of the interface and the associated mass loss due to the transport of the interface. The errors for different grid sizes after one

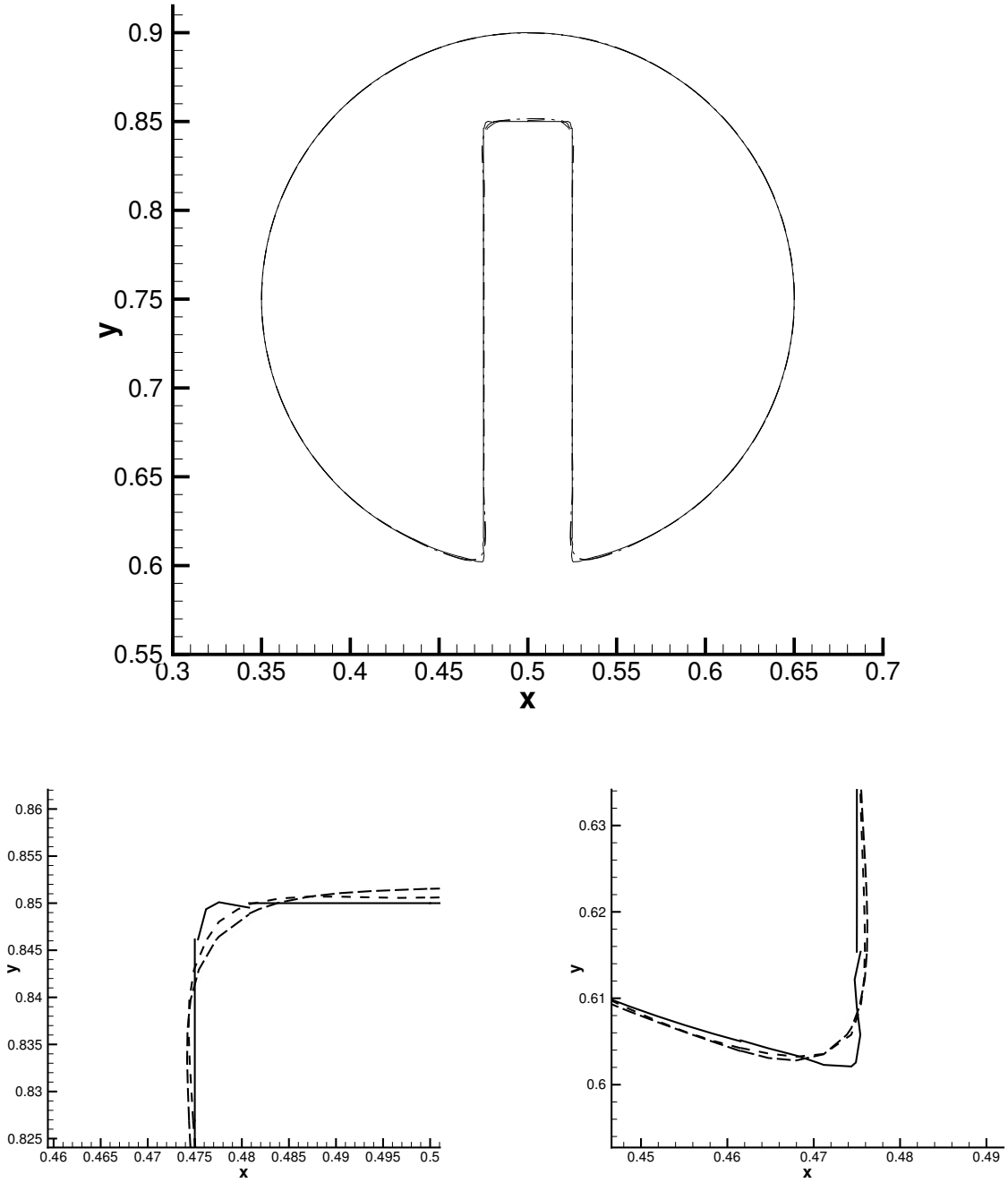


Figure 4.5: 2D Zalesak's disk test problem at $t = 0$ (solid line) after 1 (dashed), 5 (long dash) and 10 (dotted) complete rotations for a high grid resolution of 64 DG grid cells (= 256 DOF). Visualized is the zero iso-contour of the level-set function, representing the position of the interface. Close up views of the disk corners are provided to show the numerical smearing.

DG cells	DOF	Volume	$\mathbb{L}_2$ error	Order
16	64	0.0083168	8.2989356E-04	–
32	128	0.00987803	2.8639988E-04	1.53
64	256	0.010383	1.0388646E-04	1.46

Table 4.4: Errors and mass loss after one complete rotation of the 3D Zalesak's slotted sphere for different grid resolutions, compared to the approximated volume at $t = 0$. A fourth order DG scheme was used.

Rotations	Volume	% Volume loss	$\mathbb{L}_2$ error
0	0.00986912	–	8.1975975E-05
1	0.00987803	-0.09	2.8639988E-04
2	0.00984798	+0.21	3.1926325E-04
3	0.00981431	+0.55	3.4159698E-04
4	0.00976256	+1.08	3.5994490E-04
5	0.00971721	+1.54	3.7610407E-04

Table 4.5: Errors and mass loss of Zalesak's disk with increasing number of rotations. Shown is the mid-size resolution of 32 DG cells in each axis direction.

complete rotation are shown in table 4.4. The errors during several rotations of the slotted sphere are shown in table 4.5 for up to 5 rotations of the disk on the mid-sized grid.
In figure 4.6 one complete rotation of the Zalesak's sphere is shown in terms of the interface position with time. Similar to the two-dimensional case, numerical smearing occurs at the sharp edges of the slot leading to a local rounding.

4.5.3 Vortex in a box

This test case was introduced by LeVeque [58] as a demonstration of the accuracy of level-set methods. The test case is divided into a 2D and a 3D investigation. In both cases a pair of vortices is used to stretch the level-set function in time.
At $t = T/2$ the flow is reverted and, hence, the first and the last time instances are identical. The difference between the initial and last time instance can be seen as numerical errors and inaccuracies during the calculation.

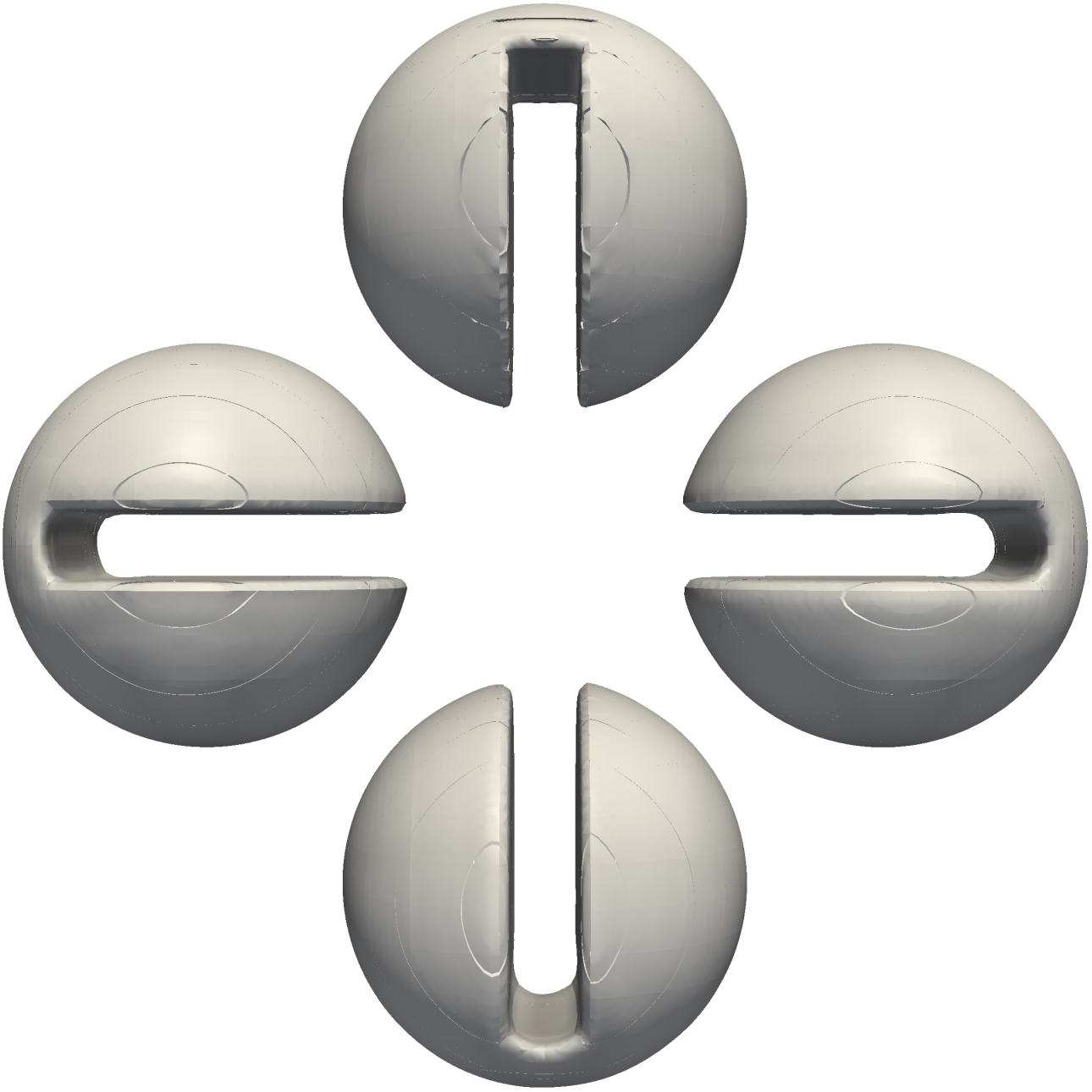


Figure 4.6: Zalesak's disk under one complete rotation (left: $t = 0.5$, bottom: $t = 1$, right: $t = 1.5$, top: $t = 2$) computed on a 32^3 grid cells and a fourth order DG scheme. Visualized is the zero iso-contour of the level-set function, representing the position of the interface.

DG cells	DOF	Area	% area loss	$\mathbb{L}_2$ error	order
exact	–	0.0707	–	–	–
16	64	0.0573	15.4	1.303 E-02	–
32	128	0.0704	3.5	3.218 E-03	2.0
64	256	0.0692	2.8	1.055 E-03	1.6

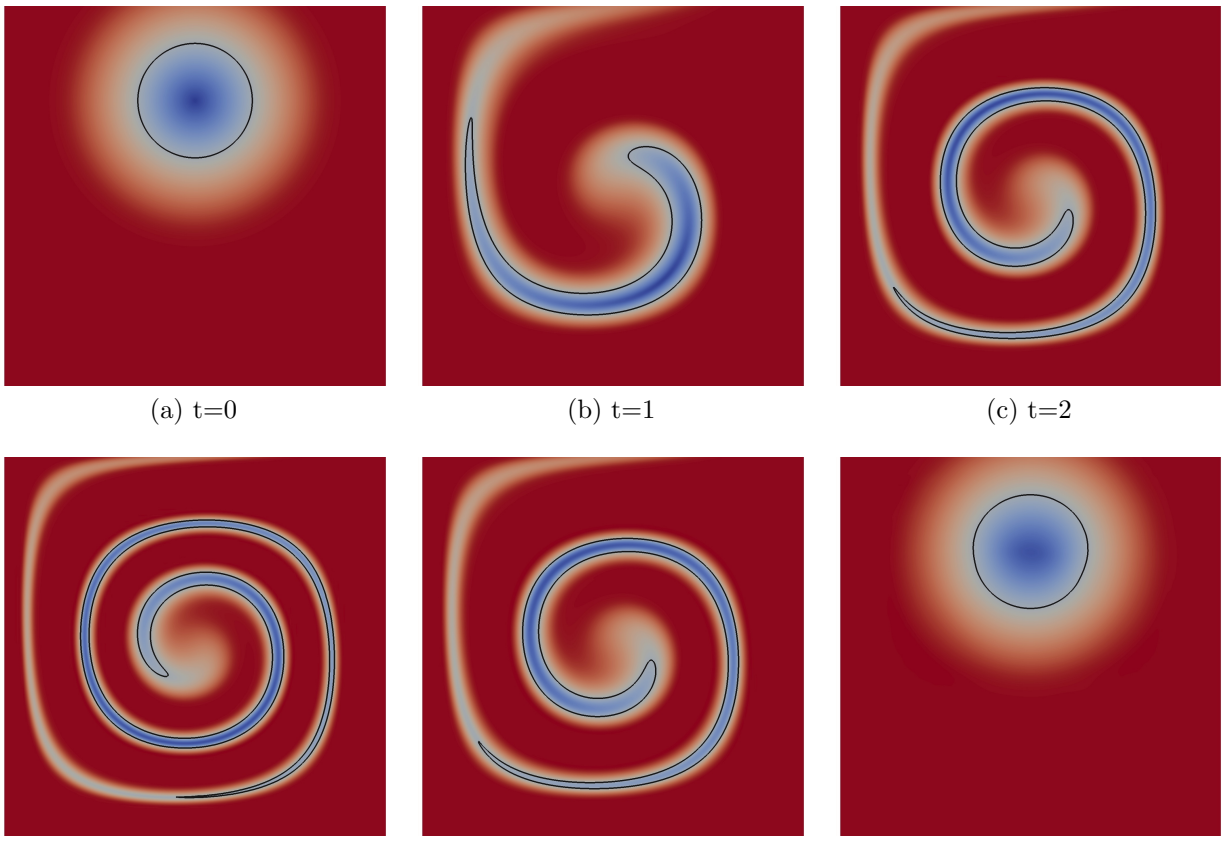
Table 4.6: Errors and mass loss for the final time $T = 8$ of the two-dimensional LeVeque test case using a fourth order DG scheme in space.

4.5.3.1 2D deformation field

The level-set is deformed using a time-dependent velocity field that is given by

$$\begin{aligned} v_1(x, y, z) &= 2\sin^2(\pi x)\sin(2\pi y)\,, \\ v_2(x, y, z) &= -\sin(2\pi x)\sin^2(\pi y)\,, \end{aligned} \tag{4.33}$$

Additionally, the velocity field is modulated in time with a period of $T = 8$ such that the velocity field is reverted at $T/2 = 4$. This implies that the maximum deformation is occurring at $T = 4$. For the final time, the initial and final solution should be identical. The analytical velocity function describes two stretching vortices that stretch the level-set function. Parts of the level-set function get very thin and are difficult to resolve.
A circle of radius $R = 0.15$ is initialized at position (0.5, 0.75) within a unit computational domain. The test cases are discretized using different mesh resolutions. The mesh sizes as well as the discretization errors at the final simulation time $t = 8$ are given in table 4.6. The level-set distribution is shown for various time instances in figure 4.7.
The initial and final solution agree very well for this level-set deformation test case. The circular geometry is nearly recovered for the $t = 8$. The small deviations are related to inaccuracies in the DG scheme in resolving the small structures within the stretched level-set function.



(a) t=0 (b) t=1 (c) t=2

(d) t=4 (e) t=6 (f) t=8

Figure 4.7: Deformation of the level-set in a two-dimensional domain for the Leveque test case at several time instances using 64^2 grid cells. The thick black line indicates the position of the zero iso-contour.

4.5.3.2 3D deformation field

The level-set is deformed using a time-dependent velocity field that is given by

$$\begin{aligned} v_1(x,y,z) &= 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z), \\ v_2(x,y,z) &= -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z), \\ v_3(x,y,z) &= -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z). \end{aligned} \tag{4.34}$$

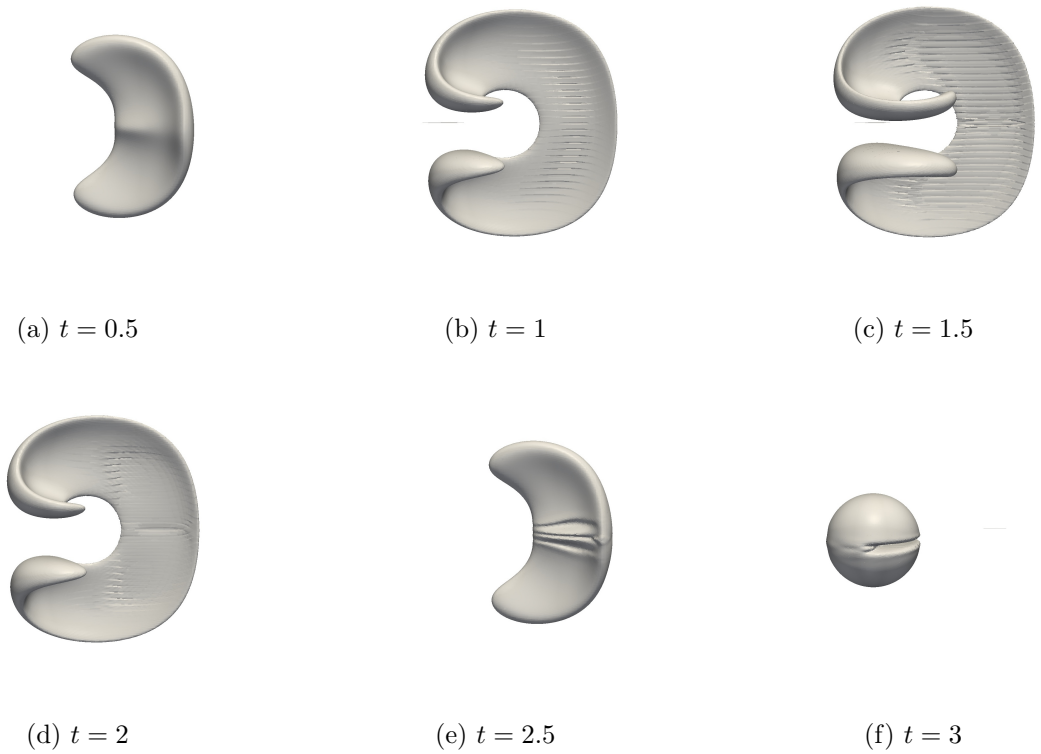
Additonally, the velocity field is modulated in time with a period of $T = 3$ such that the velocity field is reverted at $T = 1.5$. This implies that the maximum deformation is occurring at $T = 1.5$. For the final time, the initial and final solution should be identical.
A sphere of radius $R = 0.15$ is initialized at position (0.35, 0.35, 0.35) within a unit cube computational domain. The test cases are discretized using different mesh resolutions. The mesh sizes as well as the discretization errors for the final simulation time are given in table 4.7.
The prescribed analytical velocity function describes two stretching vortices that stretch the level-set function. Parts of the level-set function get very thin and are difficult to resolve. The thin level-set structures around $t = 1.5$, the time for the maximum stretching, result in approximation errors that lead to the slotted sphere at the final simulation time.
A better approximation would be gained by locally enlarging the mesh resolution in regions with strong level-set gradients or by the use of a higher mesh resolution. The overall mass conservativity and dissipation properties are very good compared to finite-volume approaches. Despite the artificial slot the sphere geometry at the final time, that is a common numerical artifact in the solution [60, 24], the total mass is conserved during the stretching. The gained DG results are similar to the Particle Level-set approach as proposed by Enright et al. [24]. Similar results were also reported by Marchandise and Remacle [60] for a DG based level-set advection scheme.

4.5.4 Level-set reinitialization order

This test case is taken from Russo and Smereka [80] and is also presented in Hartmann et al. [38]. It shows the applicability of the level-set reinitialization to a difficult problem with large and small scales present in the computational domain. As the exact solution of the final level-set distribution is known, the accuracy of the reinitialization scheme can be investigated.

Grid points	DOF	Volume	% volume loss	$\mathbb{L}_2$ error	Order
exact	–	0.01414	–	–	
$1/16$	64	0.00929	-34.29	1.0099703E-02	–
$1/32$	128	0.01234	-12.73	6.0997712E-03	0.72
$1/64$	256	0.01325	-6.29	3.2212047E-03	0.92

Table 4.7: Errors and volume loss for the final time $T = 3$ of the three-dimensional LeVeque test case.



(a) $t = 0.5$ (b) $t = 1$ (c) $t = 1.5$

(d) $t = 2$ (e) $t = 2.5$ (f) $t = 3$

Figure 4.8: Deformation of the level-set in a three-dimensional domain for the Leveque test case at several time instances using a grid of 32^3 grid cells and a fourth order DG scheme.

Δx	$\mathbb{L}_1$ error	Order	$\mathbb{L}_2$ error	order
64	1.741 E-1	–	4.172 E-1	–
128	1.977 E-2	3.13	1.406 E-1	1.56
256	8.113 E-4	4.60	2.848 E-2	2.30
512	8.812 E-7	9.85	9.387 E-4	4.92
1024	1.063 E-7	3.05	1.031 E-4	3.18

Table 4.8: Approximation errors for the reinitialized solution for the reinitialization test case.

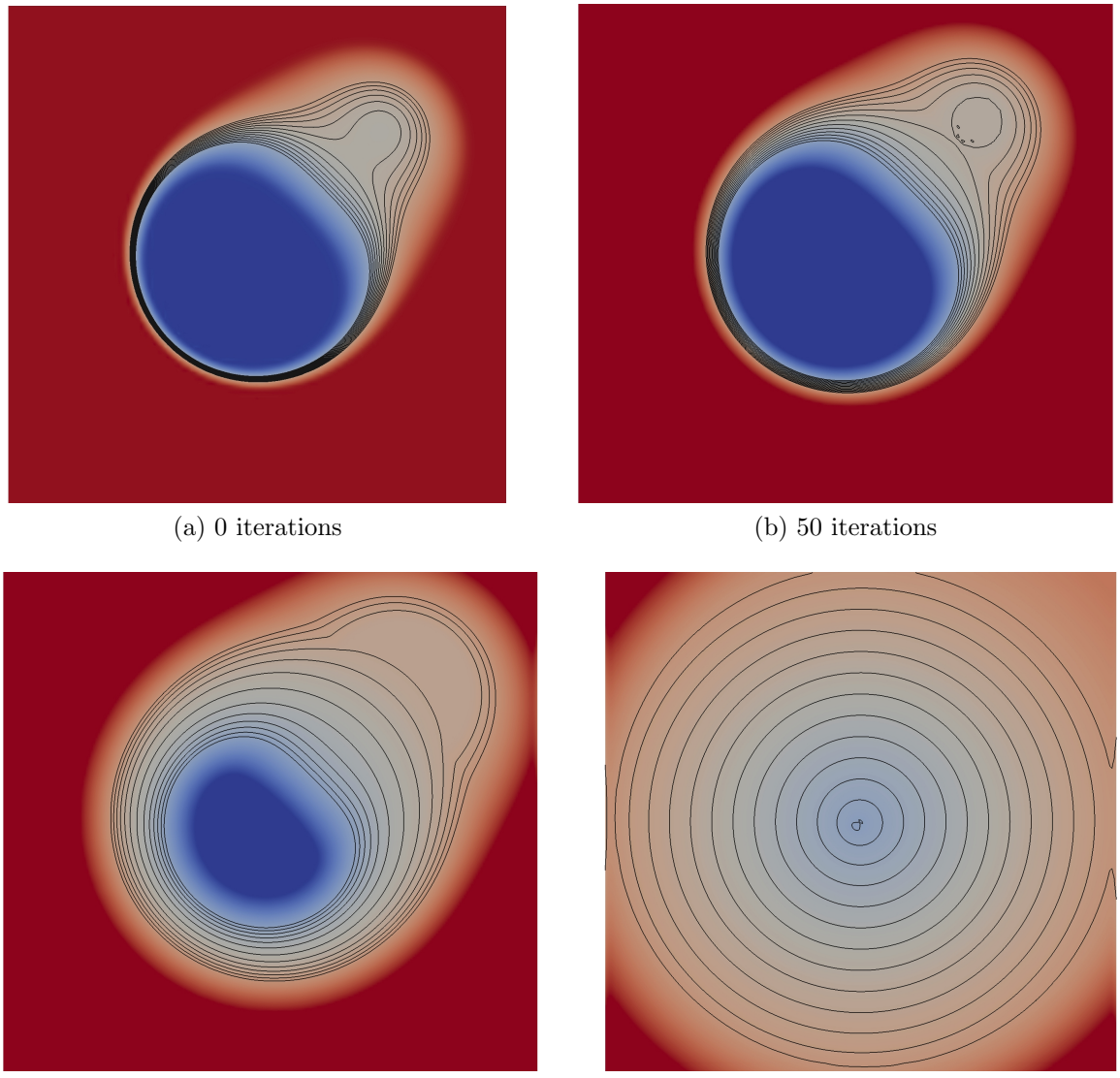
The initial level-set function is disturbed by a function $g(\mathbf{x})$ and is given by

$$\phi(\mathbf{x}) = g(\mathbf{x}) \left(r - \sqrt{x^2 + y^2} \right), \tag{4.35}$$

with a disturbance function $g(\mathbf{x})$ that introduces small and large gradients in the flow field. The radius of the level-set approximation is chosen to $r = 3$ located in the center of the domain. Note that for $g(\mathbf{x}) = 1$ the undisturbed level-set function is retained in 2D featuring unlimited concentric circles. Here, the disturbance function is chosen in accordance to Hartmann et al. [38] to

$$g(\mathbf{x}) = 0.1 + (x - r)^2 + (y - r)^2. \tag{4.36}$$

The extent of the computational domain is chosen to $[-5, -5] \times [5, 5]$. In table 4.8 the discretization errors of the reinitialization scheme are summarized for different discretizations. Note that Δx describes here the number of WENO cells in the axis directions. The level-set reinitialization evolution is depicted in figure 4.9 visualizing the steps towards the signed distance function.



(a) 0 iterations

(b) 50 iterations

(c) 200 iterations

(d) 700 iterations

Figure 4.9: Reinitialization of the level set function initialized by eq. (4.35) into a signed distance function using the WENO5 scheme in a computational domain $\Omega : [-5, -5] \times [5, 5]$ discretized by 128^2 WENO cells: (a) before the reinitialization; (b) after 50 iterations; (c) after 200 iterations; (d) after 700 iterations. Level-set contours are plotted as black lines and are evenly spaced by 0.5 in the range $\Phi = -3, \ldots 3$. The colors correspond to the value of the level-set function.

5 Equation of state

An equation of state suitable for the simulation of multi-phase flow has to fulfill several properties: First it has to reflect the dissimilar nature of the liquid and vapor phase respectively. This implies that the EOS accurately predicts the rather stiff liquid part as well as the compressible gaseous part. This property requires an at least cubic EOS if only one EOS is chosen to cover both phases. Other strategies use different EOS for the different phases that are coupled consistently at the phase interface e. g. by offsets for the enthalpy and the entropy [55]. Moreover, one tries to use rather simplified EOS descriptions as the EOS evaluation can become a major time-consuming factor during the simulation. Because of that, the ideal gas approximation is wide-spread in computational fluid dynamics for single-phase flows today.
The basic relations describing the thermodynamic state of a fluid can be summarized as follows

Specific inner energy	ϵ,
Specific enthalpy	$h = \epsilon + \frac{p}{\rho}$,
Specific Gibb's free energy	$g = h - Ts$,
Specific Helmholtz energy	$a = \epsilon - Ts = h - \frac{p}{\rho} - Ts$.

The EOS can be divided into several regimes, the gaseous, liquid and solid regimes as well as the supercritical regime above the critical point. In figure 5.1 the different flow regions are visualized in a p-T-diagram.
A closer look at the relevant EOS behavior in vapor-liquid flows is obtained by the p-ρ-diagram presented in figure 5.2. In this diagram the multi-phase region between the liquid and vapor saturation lines can be visualized. There the liquid and vapor phase are coexisting. For the description of phase transfer effects, this region is of major interest, as a wave has to be constructed through the mixture region.
After a general introduction to realistic EOS and their thermodynamic description, several EOS suitable for multi-phase flows are presented. This is followed by a description of two methods to obtain the spatial temperature gradient

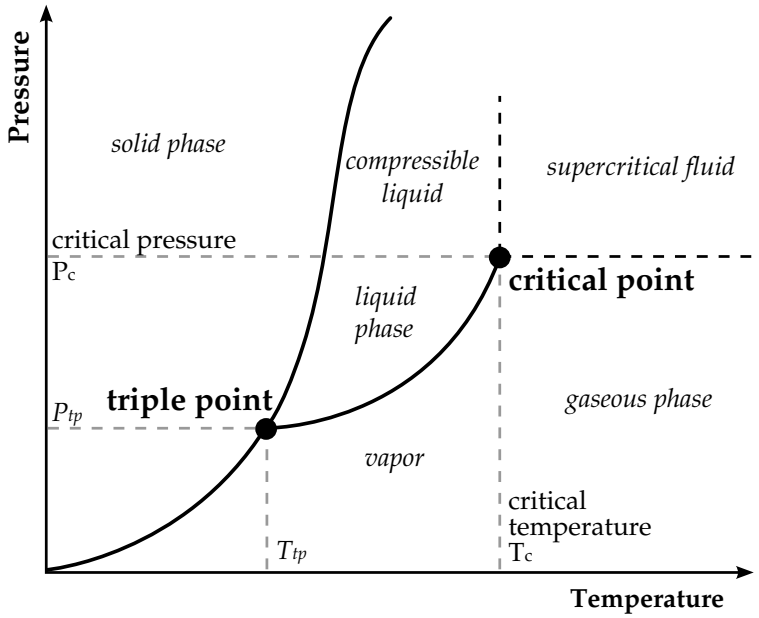


Figure 5.1: Schematic view of a phase diagram adapted from [43] for a pure substance with a description of the different regimes.

for general EOS. In the last section a numerical method to overcome the main disadvantage of realistic EOS, the high computation times, is presented.

5.1 Equations of state for multi-phase flows

For an estimation of volumetric behavior of fluids, extensive databases such as described in Poling et al. [75] and given in the NIST database [56] are available. The departure functions for the state calculation can typically be divided into an ideal and residual part. The ideal part is a function of the chosen fluid and only a function of the temperature. For the ideal part general polynomial approximations are available that are e. g. published in Poling et al. [75]. The residual part is dependent on the density ρ and the temperature T of the state. This part is heavily dependent on the chosen EOS formulation. Therefore, the ideal part is introduced in this section while the residual part is described in the following sections for different EOS. In the following the ideal part is marked with $(\cdot)^{ig}$ while the residual part is marked with $(\cdot)^{res}$. The sum of ideal and residual part is the estimated thermodynamic state.
In the following, the reference state definition of the International Institute of Refrigeration (IIR) is chosen. The reference state values for the enthalpy of a saturated liquid at $T_0 = 273.15\text{K}$ is set to $h_0^0 = 200\,{}^{kJ}/_{kg}$ and the entropy to

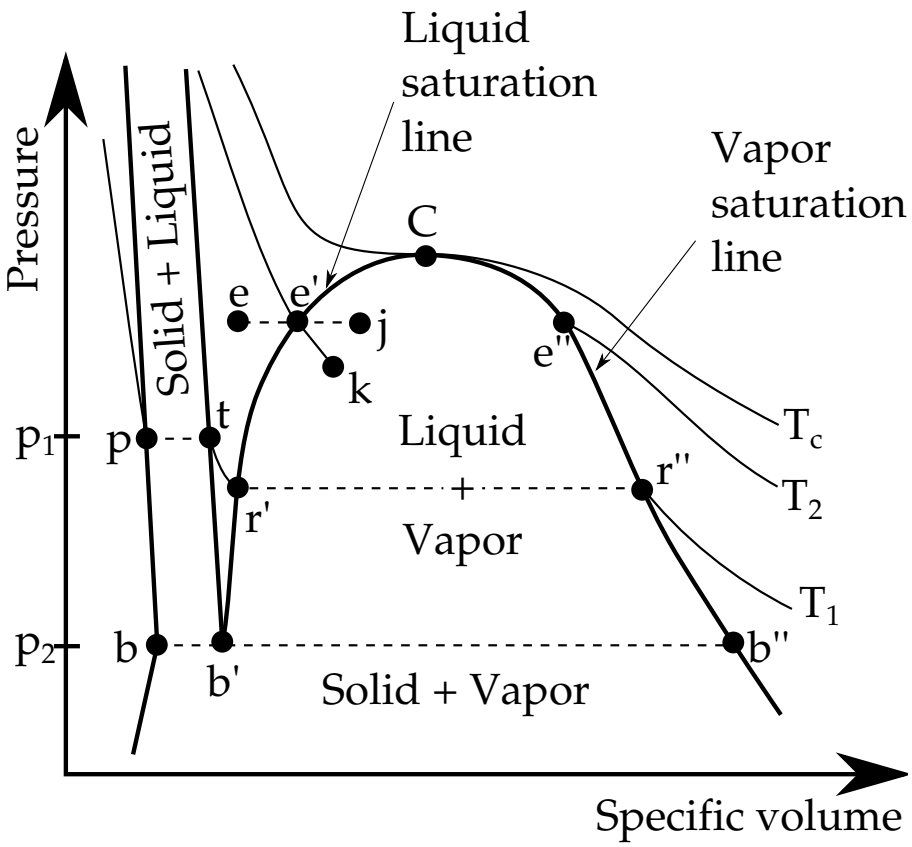


Figure 5.2: Schematic pressure-volume projection of the phase diagram for a pure substance adapted from Debenedetti [17]. b, b' and b'' are the coexisting solid, liquid and vapor phases at the triple point. p and t are the coexisting solid and liquid phases, and r' and r'' the coexisting liquid and vapor phases, at a temperature T_1. e is a stable liquid, and e' and e'' the saturated liquid and vapor phases at the same pressure as e, and at a temperature T_2. j and k are metastable, single-phase superheated states obtained by isobaric heating and isothermal decompression, respectively. C is the critical point and T_c the critical temperature ($T_c > T_2 > T_1$).

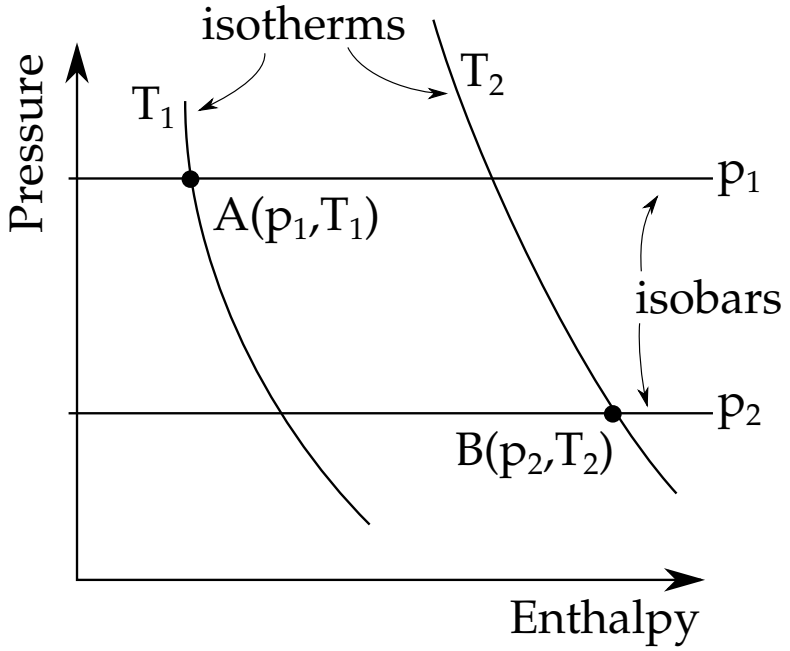


Figure 5.3: Schematic diagram for the evaluation of changes in enthalpy with changes in pressure p, and temperature T. Any thermodynamic path between A and B can be chosen.

$s_0^0 = 1\,{}^{kJ}/_{kg\,K}$. Note that the definition of the reference state is exchangeable, as only differences in the state variables are needed to describe the thermodynamic processes.
The enthalpy change between two states A and B (as visualized in the p-h-diagram in figure 5.3) can be estimated in the following way, including an ideal and residual part

$$\begin{aligned} h(T_2,p_2) - h(T_1,p_1) = &- \left[h^{ig}\left(T_2,p_2\right) - h\left(T_2,p_2\right)\right] \\ &+ \left[h^{ig}\left(T_2,p_2\right) - h^{ig}\left(T_1,p_1\right)\right] \\ &- \left[h\left(T_1,p_1\right) - h^{ig}\left(T_1,p_1\right)\right] \\ = &- \int_0^{p_2} \left(\frac{1}{\rho} - T\left(\frac{\partial\left({}^1/_\rho\right)}{\partial T}\right)_p\right)_{T=T_2} dP \\ &+ \int_{T_1}^{T_2} c_p^0 dT \\ &- \int_0^{p_1} \left(\frac{1}{\rho} - T\left(\frac{\partial\left({}^1/_\rho\right)}{\partial T}\right)_p\right)_{T=T_1} dP\,. \end{aligned} \tag{5.1}$$

For the entropy s the relation is

$$\begin{aligned} s(T_2,p_2) - s(T_1,p_1) = &- \left[s^{ig}\left(T_2,p_2\right) - s\left(T_2,p_2\right)\right] \\ &+ \left[s^{ig}\left(T_2,p_2\right) - s^{ig}\left(T_1,p_1\right)\right] \\ &- \left[s\left(T_1,p_1\right) - s^{ig}\left(T_1,p_1\right)\right] \\ = &- \int_0^{p_2} \left(\frac{R}{p} - \left(\frac{\partial\left(1/\rho\right)}{\partial T}\right)_p\right)_{T=T_2} dP \\ &+ \int_{T_1}^{T_2} \frac{c_p^0}{T} dT - R\ln\frac{p_2}{p_1} \\ &- \int_0^{p_1} \left(\frac{R}{p} - \left(\frac{\partial\left(1/\rho\right)}{\partial T}\right)_p\right)_{T=T_1} dP\,. \end{aligned} \tag{5.2}$$

The change of the ideal part needs only an approximation of the ideal gas heat capacity c_p^{ig} which can vary significantly over temperature. This temperature dependence can be approximated using polynomials, e. g. as published by Poling [75]

$$\frac{c_p^{ig}}{R} = \sum_{i=0}^{4} \hat{c}_{p,i}^0 T^i\ , \tag{5.3}$$

with the tabulated fluid-specific coefficients $\hat{c}_{p,i}$ valid for a wide temperature range of typically 200–1000 K. This temperature validity range differs for different fluids.

The compressibility factor Z is used to display the departure of a fluid from the ideal description. It is defined as

$$Z = \frac{p}{\rho R T}. \tag{5.4}$$

Note that the compressibility factor of an ideal gas equals to one. This implies that the ideal gas behavior can be described by the ideal part and a vanishing residual part. Often, c_p is assumed to be constant over the temperature range, leading to a simple algebraic EOS formulation for an ideal gas that is widely used in CFD tools today.

5.1.1 Generic EOS: Ideal gas approximation and Tait equation

The simplest compressible EOS, one can think of, is the ideal gas approximation describing the behavior of a diluted gas (without considering the molecular volume of the gas and the attraction forces). In this case the residual part is zero and the pressure is only a function of the temperature (and independent of the specific volume $\nu = 1/\rho$). This choice does not reveal the dissimilar nature of the multi-phase flow (especially for the liquid phase) and is only a suitable description for the gas part. Assuming a constant, temperature-independent heat capacity c_p behavior, the pressure law, the internal energy and the sound speed can be written as follows

$$p = \rho R T, \tag{5.5}$$

$$\epsilon = \frac{p}{(\gamma - 1)\rho}, \tag{5.6}$$

$$c = \sqrt{\gamma \frac{p}{\rho}}, \tag{5.7}$$

with the gas constant R and the adiabatic coefficient γ.
A suitable extension to be able to describe two-phase flow is to combine the ideal gas approximation with a simple model for the nearly incompressible liquid phase: the Tait EOS. This EOS linearizes the EOS behavior around a reference state given by $(\rho_{0,\text{tait}}, p_{0,\text{tait}})$ with material specific constants $k_{0,\text{tait}}$ and $\gamma_{0,\text{tait}}$ leading to the following relations in the liquid phase

$$p = k_{0,\text{tait}} \left(\left(\frac{\rho}{\rho_{0,\text{tait}}} \right)^{\gamma_{\text{tait}}} - 1 \right) + p_{0,\text{tait}}, \tag{5.8}$$

$$c = \sqrt{\frac{\partial p}{\partial \rho}} = \sqrt{\gamma_{\text{tait}} k_{0,\text{tait}} \frac{\rho^{\gamma_{\text{tait}}-1}}{\rho_{0,\text{tait}}^{\gamma_{\text{tait}}}}}\,. \tag{5.9}$$

As the pressure is only dependent on the density, no energy equation has to be solved. As in this case two different EOS are used within the computational domain, one has to ensure a consistent coupling at the phase interface.
A possibility to combine both EOS described above in one equation is the stiffened gas equation of state. The corresponding pressure law is

$$p = \rho\,(\gamma - 1)\,\epsilon - \gamma p_\infty \tag{5.10}$$

that can be used to switch between the ideal gas behavior ($p_\infty = 0$) and the Tait EOS ($p_\infty = k_0 - p_0$). In the following the formulation with two distinct EOS is chosen.
All these generic EOS have the advantage that they are computationally cheap, as direct analytical expressions for the conversion between primitive and conservative variables can be found. The biggest drawback is that these equations can not be used to describe phase transition effects without introducing a generic switching point between the distinct EOS for the vapor and liquid phase.

5.1.2 Van der Waals EOS

The van der Waals equation of state [100] is an extension of the ideal gas approximation to include the molecular volume of the gas as well as attraction forces between the molecules. The molecular volume is accounted for by the introduction of a parameter B in the equation. Also, the pressure is reduced by a term $A\rho^2$ to account for the attraction force of the molecules. This formulation is the simplest formulation of a cubic EOS.
Altogether, the van der Waals equation of state can be written

$$p = \frac{RT}{1/\rho - B} - A\rho^2, \tag{5.11}$$

with temperature independent fluid-specific constants A and B that can be calculated using the critical state of the fluid to

$$A = \frac{3p_c}{\rho_c^2}, \tag{5.12}$$

$$B = \frac{1}{3\rho_c}. \tag{5.13}$$

A reformulation in terms of the density ρ shows that this EOS is cubic.
Due to the temperature independent van der Waals constants, this EOS can be solved analytically for the enthalpy h and entropy s. For this EOS the departure functions (residual part of the EOS) can be written as follows in terms of the compressibility factor Z

$$\frac{h^{ig} - h}{RT} = -\rho + 1 - Z \ , \tag{5.14}$$

$$\frac{s^{ig} - s}{R} = -\ln\left[Z\left(1 - B\rho\right)\right] \ . \tag{5.15}$$

From this departure functions the inner energy and the sound speed can be calculated to (assuming a constant c_v over the considered temperature range)

$$\epsilon = c_v T + A\rho, \tag{5.16}$$

$$c = \sqrt{\frac{RT}{1-B\rho} + \frac{RT\rho B}{(1-B\rho)^2} - 2A\rho}. \tag{5.17}$$

Note that the sound speed is not defined in the spinodal region in which $c^2 < 0$.

5.1.3 Peng-Robinson EOS

The Peng-Robinson equation of state [73] resolves both the liquid and vapor states for fluids with sufficient accuracy and was initially developed to describe the behavior of alkanes. The advantages of this cubic EOS are that the densities can be computed analytically in case pressure and temperature of the state are known. Furthermore, this EOS can be easily extended to include material behavior due to mixing effects through an adaption of the Peng-Robinson coefficients A and B. Note that in contrast to the van der Waals EOS (see section 5.1.2) the coefficient A is now temperature dependent which implies that an iterative solution is needed in case the temperature is not known.
The Peng-Robinson pressure function can be expressed as follows

$$p = \frac{RT}{1/\rho - B} - \frac{A(T)}{(1/\rho)^2 + 2B1/\rho - B^2}, \tag{5.18}$$

with the temperature dependent constant $A(T)$ and the temperature independent constant B that are defined as follows

$$A(T) = \frac{(RT_c)^2}{p_c}\left(1 + f_\omega\left(1 - \sqrt{\frac{T}{T_c}}\right)\right)^2, \tag{5.19}$$

$$B = \frac{RT_c}{p_c}. \tag{5.20}$$

For definition of the constants the fluid state at the critical point (p_c, T_c) is used as well as a correlation based on the accentric factor ω of the fluid for the f_ω-term

$$f_\omega = 0.37464 + 1.52226\omega - 0.2699\omega^2. \tag{5.21}$$

Note that equation (5.18) can be written as third order polynomial in term of the compressibility factor Z

$$Z^3 + (B' - 1)Z^2 + \left[\Theta' + \epsilon' - 2B'(B' + 1)\right] Z - \left[\epsilon'(B' + 1) + \Theta' B'\right] = 0 \quad (5.22)$$

with the dimensionless parameters

$$B' = \frac{Bp}{RT}, \qquad \Theta' = \frac{A(T)p}{(RT)^2}, \qquad \epsilon' = -B^2 \left(\frac{p}{RT}\right)^2 . \quad (5.23)$$

Calculation of state variables

Using the definition of (5.18) the residual terms for enthalpy h and entropy s can be defined to

$$\frac{h^{ig} - h}{RT} = \frac{A(T) - T\frac{dA(T)}{dT}}{\sqrt{8}RTB} \ln\left[\frac{\frac{1}{\rho} + (1 - \sqrt{2})B}{\frac{1}{\rho} + (1 + \sqrt{2})B}\right] + 1 - Z \ , \quad (5.24)$$

$$\frac{s^{ig} - s}{R} = \frac{\frac{dA(T)}{dT}}{\sqrt{8}RB} \ln\left[\frac{\frac{1}{\rho} + (1 - \sqrt{2})B}{\frac{1}{\rho} + (1 + \sqrt{2})B}\right] - \ln\left[Z\left(1 - B\rho\right)\right] \ . \quad (5.25)$$

Using these definitions all other state variables can be calculated, in particular the inner energy u, the Gibb's free energy g and the Helmholtz energy a.
Based on the definition of the adiabatic sound speed c

$$c^2 = \left(\frac{1}{\rho}\right)^2 \left[\left(\frac{\partial p}{\partial T}\right)_\rho \frac{T}{c_v} - \left(\frac{\partial p}{\partial {}^1\!/\!_\rho}\right)_T\right] \quad (5.26)$$

the necessary derivatives can be calculated using equation (5.18) leading to the following expression

$$\begin{aligned} c^2 = {} & \frac{({}^R\!/\!_\rho)^2}{(B - {}^1\!/\!_\rho)^2}\frac{T}{c_v} + \frac{RT\left({}^1\!/\!_\rho\right)^2}{(B - {}^1\!/\!_\rho)^2} + \frac{2A(T)(B + {}^1\!/\!_\rho)\left({}^1\!/\!_\rho\right)^2}{(({}^1\!/\!_\rho)^2 + {}^{2B}\!/\!_\rho - B^2)^2} \\ & + \left(\frac{\frac{dA(T)}{dT}\left({}^1\!/\!_\rho\right)^2}{B^2 - {}^{2B}\!/\!_\rho - \left({}^1\!/\!_\rho\right)^2}\right)^2 \frac{T}{c_v} + \frac{2R\left({}^1\!/\!_\rho\right)^2}{(B - {}^1\!/\!_\rho)} \frac{\frac{dA(T)}{dT}}{(B^2 - {}^{2B}\!/\!_\rho - \left({}^1\!/\!_\rho\right)^2)} \frac{T}{c_v} . \end{aligned} \quad (5.27)$$

Estimation of liquid densities

Using equation (5.18) and a suitable approximation of the saturation pressure (for more information about how to approximate the saturation pressure see Poling et al. [75]), the liquid and vapor densities can be estimated. However, the calculated liquid and vapor pressure do not fit to experimentally determined values, such that correction procedures are needed.
Approaches to correct the estimation of the liquid vapor densities is a volume shift as e. g. described by Peneloux et al. [71] using a constant offset or e. g. by Monnery et al. [64] using a temperature dependent correction term.

5.1.4 EOS based on the approximation of the Helmholtz energy functional

High fidelity and high accuracy EOS are often based on a non-analytic formulation of the residual Helmholtz energy. The parameters of such formulations are based on optimization models to acquire highly accurate EOS. This is described by Setzmann and Wagner [85, 84]. In this case, the describing polynomials of the underlying EOS are fitted to experimental data based on an optimization strategy.
The residual Helmholtz energy functional ϕ^{res} is written as a function of the ideal gas functional ϕ^{ig} for a thermodynamic state defined by the density ρ and the temperature T

$$\frac{\phi^{res}}{RT} = \frac{\left[\phi(T,\rho) - \phi^{ig}(T)\right]}{RT}. \tag{5.28}$$

The descriptions for polar and non-polar substances differ in terms of the used exponents and variables. Lemmon and Span [57] published a high-reliability EOS for industrial fluids. Extensive descriptions for various fluids exist in literature. The residual term for the Helmholtz energy functional ϕ^{res} can be expressed by a sum of regressed polynomial functions

$$\phi^{res}(\tau,\delta) = \sum_k N_k \delta^{i_k} \tau^{j_k} + \sum_k N_k \delta^{i_k} \tau^{j_k} \exp\left(-\delta^{l_k}\right) \tag{5.29}$$

with the inverse reduced temperature $\tau = {}^{T_c}/_{T}$ and non-dimensional density $\delta = {}^{\rho}/_{\rho_c}$ as parameters. A detailed descriptions of the parameter N_k and the respective exponents i_k, j_k, l_k can be found in literature. The acceptable error for fluid properties is less than 1% in this formulation. For the fluid acetone

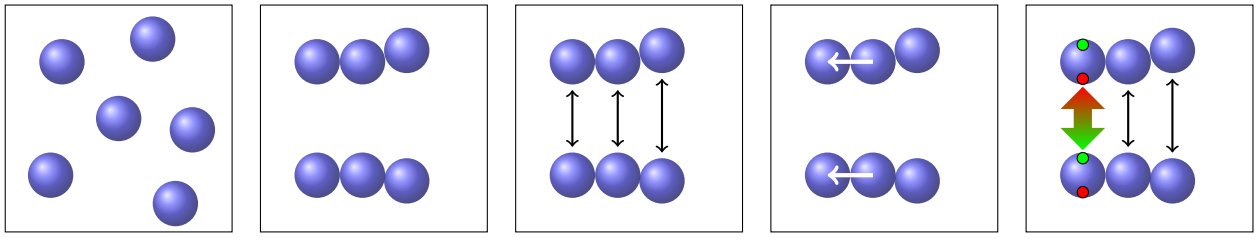
12 terms are necessary to approximate the residual Helmholtz energy potential. The number of terms varies for each substance, e. g. up to 52 terms are necessary for an accurate description of water.
Starting from the correlated Helmholtz energy functional, all state variables can be determined by analytic differentiation with respect to the density and temperature. This is done in terms of the non-dimensional parameters τ and δ. Several libraries are available to estimate the fluid behavior of fluids as e. g. CoolProp [8], FPROPS [76] and Refprop [56]. The main disadvantage of this approach is the massive computation times for the EOS evaluation that can easily take over 90% of the total computation time.

5.1.5 Density-functional theory

A different approach to estimate fluid behavior of real gases is the density-functional theory (DFT). In this theory, the Helmholtz energy functional is approximated based on the perturbed-chain polar statistical associating fluid theory (PCP-SAFT) for vapor-liquid interfaces, as e. g. described by Gross in [35] and by Johannessen et al. in [46]. Here, only a short overview is presented. An extension for binary mixtures is described in [92] and the estimation of surface tension in [91].
The DFT is a molecular theory that calculates average probability densities for different fluids. This is a completely different approach compared to the fitting to experimental data (see section 5.1.4). This theory is located in-between the molecular simulations and the continuum models. It allows for a prediction of static interface properties (e. g. surface tension coefficients) as well as vapor-liquid phase equilibria and fluid properties. These interface properties play a significant role in the simulation of flows with interfaces.
The underlying molecular description consists of the representation of the single molecules in terms of connected spherical segments. In this model the Helmholtz energy functional for the chain formulation and the hard sphere reference fluid can be developed. The complete functional is given by the sum of all contributions

$$\phi(\rho_i) = \phi^{id}(\rho_i) + \phi^{\text{hard chain}} + \phi^{\text{disp. chain}} + \phi^{\text{polar}} + \phi^{\text{association}} . \tag{5.30}$$

In figure 5.4 the different contributions are visualized.
This formulation has the advantage that it can be applied to various types of fluids and allows the simultaneous estimation of interface properties. Even a further generalization to the description of mixtures is possible in terms of a modified interaction functional.



$$\phi(\rho_i) = \phi^{id}(\rho_i) + \phi^{\text{hard chain}} + \phi^{\text{disp. chain}} + \phi^{\text{polar}} + \phi^{\text{association}}$$

Figure 5.4: Chain contribution for the DFT with visualization of the physical effects.

5.2 Estimation of the temperature gradient

One problem arises in the approximation of the viscous part of the Navier-Stokes equations. In the energy balance equation the spatial temperature gradient $\partial T/\partial \mathbf{x}$ is needed for the estimation of the heat transfer term. The standard DG lifting procedure (see section 3.1.5) provides derivatives of the conservation variables but not for the primitive variables. The estimation of the derivatives of the velocity field is straightforward as no EOS description is involved and the velocity gradients can be calculated using the chain rule. For the estimation of the temperature gradient the estimation gets a bit tricky as the EOS comes into play.
For an ideal gas approximation, the temperature gradient can be calculated directly from the gradient of the total energy, as the temperature can be expressed as an analytical function of the inner energy. For a general EOS this is getting more difficult, as the total energy e is a function of the density ρ and the temperature T. The total energy e consists of the inner energy ϵ and the kinetic energy i

$$e = \epsilon + i = \epsilon + \frac{1}{2}v^2. \tag{5.31}$$

To be able to estimate the temperature gradient, that is needed for the heat conduction term in the Navier-Stokes equations, two different methods are available. The first is to exploit the structure of the underlying Helmholtz energy functional and use its second derivatives to estimate the temperature gradient. This procedure is a generalization of the procedure used for an ideal gas. An additional term enters the formulation derived in section 5.2.1 as the inner energy is additionally a function of the density.

The second approach is to employ a DG lifting procedure for the additional variable temperature to provide an estimate for the temperature gradient. The approach for this method is described in section 5.2.2 and is based on the BR1 method of Bassi and Rebay [6]. This method is straightforward in the implementation and applicable to any kind of EOS as the differentiation is done in the DG context.

5.2.1 Direct evaluation using derivatives of the Helmholtz energy functional

The first possibility is to use the second derivatives of the Helmholtz energy functional for the calculation of the temperature gradient. Here, the description is restricted to EOS that can be expressed in terms of the Helmholtz energy functional ϕ (see section 5.1.4). In this description the inner energy ϵ can be expressed as

$$\epsilon = RT_c \left(\frac{\partial \phi^{ig}}{\partial \tau} + \frac{\partial \phi^{res}}{\partial \tau} \right), \tag{5.32}$$

using again the non-dimensional parameters $\tau = {}^{T_c}/_{T}$ and $\delta = {}^{\rho}/_{\rho_c}$. Accordingly the total derivative of the inner energy can be written as

$$d\epsilon = \left(\frac{\partial \epsilon}{\partial \delta} \right)_{\tau=\text{const}} d\delta + \left(\frac{\partial \epsilon}{\partial \tau} \right)_{\delta=\text{const}} d\tau. \tag{5.33}$$

Up to now only the non-dimensional derivatives are approximated. In dimensional space the derivative of the inner energy is given by

$$d\epsilon = \frac{1}{\rho_c} \left(\frac{\partial \epsilon}{\partial \rho} \right)_{T=\text{const}} d\rho + T_c \left(\frac{\partial \epsilon}{\partial T} \right)_{\rho=\text{const}} dT \tag{5.34}$$

with the two derivatives that can be calculated using the non-dimensional inverse reduced temperature τ and the non-dimensional density δ to

$$\begin{aligned} \left(\frac{\partial \epsilon}{\partial \rho} \right)_{T=\text{const}} &= \frac{RT_c}{\rho_c} \left(\frac{\partial^2 \phi^{res}}{\partial \delta \partial \tau} \right) && := c^*(\rho, T) \\ \left(\frac{\partial \epsilon}{\partial T} \right)_{\rho=\text{const}} &= R\tau^2 \left(\frac{\partial^2 \phi^{ig}}{\partial \tau^2} + \frac{\partial^2 \phi^{res}}{\partial \tau^2} \right) && := c_v(\rho, T) \end{aligned} \tag{5.35}$$

The second derivatives of the Helmholtz functional can be obtained analytically from the EOS evaluation tools, e. g. from CoolProp.

Using these expressions, the spatial gradient of the inner energy $^{du}/_{d\mathbf{x}}$ can be written as

$$\begin{aligned}\frac{du}{d\mathbf{x}} &= \frac{d}{d\mathbf{x}}\left[\left(\frac{\partial\epsilon}{\partial\rho}\right)_{T=\text{const}} d\rho + \left(\frac{\partial\epsilon}{\partial T}\right)_{\rho=\text{const}} dT\right] \\ &= c^*(\rho,T)\frac{d\rho}{d\mathbf{x}} + c_v(\rho,T)\frac{dT}{d\mathbf{x}}\,. \end{aligned} \tag{5.36}$$

This can be reformulated to get the spatial derivative of the temperature

$$\frac{dT}{d\mathbf{x}} = \left[\frac{d\epsilon}{d\mathbf{x}} - c^*(\rho,T)\frac{d\rho}{d\mathbf{x}}\right]\frac{1}{c_v(\rho,T)}\,. \tag{5.37}$$

Note that the derivative of the density is already known due to the lifting of the conservative variables.
If the total energy is considered instead of the inner energy, the following expression for the spatial temperature gradient is obtained

$$\begin{aligned}\frac{dT}{d\mathbf{x}} &= \left[\frac{de}{d\mathbf{x}} - \mathbf{v}^{\mathrm{T}}\frac{d\mathbf{v}}{d\mathbf{x}} - c^*(\rho,T)\frac{d\rho}{d\mathbf{x}}\right]\frac{1}{c_v(\rho,T)} \\ &= \left[\frac{d\epsilon}{d\mathbf{x}} - c^*(\rho,T)\frac{d\rho}{d\mathbf{x}}\right]\frac{1}{c_v(\rho,T)}\,. \end{aligned} \tag{5.38}$$

For the special case of an ideal gas ($\phi^{res} = 0$), the well known expression is obtained

$$\frac{dT}{d\mathbf{x}} = \frac{1}{c_v(T)}\frac{d\epsilon}{d\mathbf{x}} = \frac{\gamma-1}{R}\frac{d\epsilon}{d\mathbf{x}}. \tag{5.39}$$

With this modified equation (5.38) it is now possible to calculate the temperature gradient from the gradient of the inner energy and the velocity gradients.

5.2.2 Temperature lifting

The second alternative is to employ an additional lifting procedure for the primitive variable temperature. The lifting procedure is necessary to account for the cell-locale DG approximation. It is lifted as additional variable to provide an estimate for the spatial temperature gradients needed in the flux approximation. Here, the temperature lifting approach is based on the BR1 scheme of Bassi and Rebay [6] to account for discontinuous solutions at the cell boundaries. This

procedure is necessary as it is not easily possible to calculate the temperature gradient from the gradient of the total energy density.
The temperature lifting approach is applicable to any kind of EOS and avoids the evaluation of the second derivative terms of the Helmholtz energy functional. Instead the temperature gradient is approximated by the DG ansatz polynomials. This approximation is computationally cheaper as no additional EOS gradient terms have to be approximated and stored. A further advantage is that this approach can be applied to any kind of EOS regardless if an analytic description is available or not.

5.2.3 Comparison for an ideal gas

Here, the two previously described approaches for the estimation of the temperature gradient are compared for the special case of an ideal gas. The temperature gradient is needed for the evaluation of the heat conduction term in equation 2.1.1 and can be obtained either by a temperature lifting procedure (see section 5.2.2) or by the use of the second derivatives of the Helmholtz energy functional (see section 5.2.1). The second approach is trivial for an ideal gas as an analytical expression for the calculation of the temperature from the inner energy is available.
The temperature lifting approach is validated by means of a convergence test for the compressible Navier-Stokes equations using an ideal gas approximation with the applied lifting of the temperature. For general EOS, the construction of a solution test case with an exact solution is getting difficult (or even impossible). The chosen test case is a planar sine-wave with a source term according to the method of manufactured solution, with a constant viscosity and heat transfer coefficient.
The corresponding error norms and numerical order are listed in table 5.1 for a fourth and fifth order accurate DG scheme. In the table the comparison to the direct evaluation approach is shown. The convergence is similar in both cases, only the discretization errors are slightly different.

5.2.4 Conclusion

In the following, the temperature lifting approach is applied as this is the faster and more general approach to estimate the spatial temperature gradient. The additional computation time for the estimation of temperature as well as the

$N = 3$:

	T-lifting				direct evaluation			
# Elems	$\mathbb{L}_2$	$\mathcal{O}_{\mathbb{L}_2}$	$\mathbb{L}_\infty$	$\mathcal{O}_{\mathbb{L}_\infty}$	$\mathbb{L}_2$	$\mathcal{O}_{\mathbb{L}_2}$	$\mathbb{L}_\infty$	$\mathcal{O}_{\mathbb{L}_\infty}$
4	9.6e-03	-	1.9e-02	-	9.6e-03	-	1.9e-02	-
8	2.3634e-03	2.03	8.8690e-03	1.06	2.3600e-03	2.03	9.0196e-03	1.05
16	1.1994e-04	4.30	4.5026e-04	4.30	1.2002e-04	4.30	4.3373e-04	4.38
32	8.5889e-06	3.80	3.8116e-05	3.56	8.6886e-06	3.79	3.6323e-05	3.58
64	5.9732e-07	3.85	2.1619e-06	4.14	6.1099e-07	3.83	2.2551e-06	4.01
128	3.6960e-08	4.01	1.2040e-07	4.17	3.8273e-08	4.00	1.3543e-07	4.06

$N = 4$:

	T-lifting				direct evaluation			
# Elems	$\mathbb{L}_2$	$\mathcal{O}_{\mathbb{L}_2}$	$\mathbb{L}_\infty$	$\mathcal{O}_{\mathbb{L}_\infty}$	$\mathbb{L}_2$	$\mathcal{O}_{\mathbb{L}_2}$	$\mathbb{L}_\infty$	$\mathcal{O}_{\mathbb{L}_\infty}$
4	9.4e-03	-	4.1e-02	-	9.1e-03	-	3.9e-02	-
8	5.8152e-05	7.34	2.6630e-04	7.26	4.8097e-05	7.57	1.1130e-04	8.43
16	8.6964e-06	2.74	3.5416e-05	2.91	9.9561e-06	2.27	4.0038e-05	1.48
32	2.5547e-07	5.09	1.4995e-06	4.56	2.7193e-07	5.19	1.7843e-06	4.49
64	8.0628e-09	4.99	5.0596e-08	4.89	8.1631e-09	5.06	5.3775e-08	5.05
128	2.6719e-10	4.92	1.8494e-09	4.77	2.6764e-10	4.93	1.8664e-09	4.85

Table 5.1: Numerical convergence results for a fourth and fifth order accurate DGSEM scheme with temperature lifting compared to the direct evaluation of the temperature gradient for an ideal gas. Shown is the error for the variable ρ. The other variables have a similar tendency.

numerical calculation of the gradient is negligible compared to the estimation of the two additional terms in (5.38). Efficient implementations would require an additional tabulation approach, as introduced in the next section, for these values and, thus, an enlarged tabular size of the approximation.

5.3 Approximation of EOS behavior

The use of realistic EOS for the simulation of compressible multi-phase flows introduces the difficulty of very long computation times due to the frequent EOS evaluation. This EOS evaluation is computationally cheap as long as the pressure p and temperature T can be expressed analytically by means of the density ρ and inner energy ϵ. This is e. g. the case for an ideal gas or the stiffened gas formulation. However, these formulations describe only a simplified EOS behavior and they are not an accurate approximation for near-critical fluid approximations. Due to the low computational effort to solve the ideal gas approximation, this EOS is widely used in CFD. In most single-phase applications the assumption of an ideal gas is adequate as it describes the behavior of air at moderate ambient conditions quite well.

In case more complex EOS models are needed, e. g. to describe the fluid behavior in the vicinity of the critical point, the computational costs enlarge enormously. This is on the one hand related to the evaluation of the more complex EOS formulation, as up to 53 terms are needed for the approximation of the Helmholtz energy functional. On the other side, a non-linear Newton iteration for the temperature is needed to be able to express the EOS in terms of the density and the inner energy. For some important EOS, e. g. water, also the reverse formulations are available. This iteration is needed as the Helmholtz energy functional is expressed in terms of the density and the temperature. Both factors account for a massively enlarged computation time due to the use of realistic EOS. Nevertheless, such complex EOS are needed to describe flow effects at extreme ambient conditions.

Note that for a Godunov-type numerical scheme, the EOS evaluation is needed in every time step and for every grid point due to the estimation of the numerical fluxes. Thus, the EOS evaluation is a very wallclocktime-critical process. In this chapter a method is described that allows for a massive reduction of the computation time compared to the direct evaluation of the underlying EOS.

5.3.1 Basic idea of the EOS tabulation

The basic idea is to compute and store the EOS behavior prior the simulation as introduced by Dumbser et al. in [22] based on a general AMR tabulation for the Euler equations. The extension to the compressible viscous Navier-Stokes equations is described by Bolemann et al. [9]. In a step prior to the simulation, the EOS behavior is tabulated by means of the density ρ and the inner energy ϵ, that are the variables needed for the conversion of the conservative to the primitive variables during the simulation. The output of the EOS evaluation is then stored in an adaptive quadtree-based table with p-refined DG polynomial to provide minimal storage size together with a chosen accuracy. The approximation granularity needed depends on the region in which the EOS behavior is tabulated. Typically, a much finer resolution is needed in the vicinity of the critical point as well as the saturation lines.
The main advantage of the method is that it skips the non-linear temperature iteration during the simulation time because the variables as needed in the flow solver are given directly. Furthermore, the EOS evaluation is replaced by the interpolation of the solution within the EOS table. This provides a robust and efficient approach for the EOS evaluation as the interpolation of the table is typically faster compared to the temperature iteration.
For a simulation within the framework of a Godunov-type CFD code based on the Navier-Stokes equations, the evaluation of pressure and sound speed is needed for the flux calculation. An estimate of the viscosity and the heat transfer coefficient is necessary to account for the viscous contributions. These approximations are based on correlations and are given as a function of the fluid temperature and density (primarily of the fluid temperature). Some of the correlations are given in [75]. These expressions can now be evaluated efficiently with the known temperature or tabulated as additional variables within the tabulation approach.

5.3.2 EOS library

The evaluation and the generation of the EOS table is done within the EOS library that is developed at IAG. This library supports several EOS evaluation programs as backends and allows for an easy implementation of real EOS in the CFD code using a single code interface. Here is a list of the supported EOS evaluation programs that are based on the approximation of the Helmholtz energy functional ϕ:

1. ideal gas (for validation purposes only),
2. Fluidcal (Wagner routines) [101],
3. CoolProp [8],
4. Ascend Fprops [76],
5. Refprop [56],
6. PCP-SAFT EOS [46] and section 5.1.5,
7. Cubic Peng-Robinson EOS [73] and section 5.1.3.

The tabulation of an ideal gas is included to be able to access the computational overhead due to the use of a tabulated EOS compared to a direct implementation of the EOS in the code. The EOS library enables the usage of a variety of different EOS with a single call to the library. The tabulation approach overcomes the most significant disadvantage of accurate EOS, the long computation times for the EOS evaluation during the simulation. Furthermore, it enables the usage of multiple EOS within the computational domain such that different EOS can be chosen for liquid and gaseous bulk phases.

5.3.2.1 Building step

In a first step, the EOS is tabulated using the variables density ρ and inner energy ϵ to avoid non-linear iterations during the simulation. The lower and upper boundaries of the tabulation region are chosen according to the limits of the underlying EOS and the expected range needed during the simulation. The polynomial coefficients of the EOS approximation are adaptively stored within an quadtree-based data structure that allows for locale h-p-refinement through the choice of a suitable approximation order within the cell and the refinement level of the quadtree data structure. This approach allows for an adaptive storage with minimal memory requirement. The minimal memory requirement is essential as the table has to fit into the RAM of the computation nodes.
In the building step that is done prior to the simulation, the adaptive tabulation is started using a quadtree grid with initially four cells. Similar to the DG approximation Lagrangian interpolation polynomials are used to approximate the EOS behavior. This results in a nodal set of basis functions and a diagonal mass matrix. Here the two-dimensional nodal coordinates are given by ξ and

η that span ρ-ϵ-space. Since the interpolation points are the Gauss-Legendre quadrature points the resulting polynomial basis is by construction also orthogonal. This implies that the element mass-matrix M_{km} is diagonal and can be computed as

$$M_{km} = \int_0^1 \int_0^1 \varphi_k(\xi,\eta)\varphi_m(\xi,\eta)d\xi d\eta = \begin{cases} \chi_m & \text{if } k = m \\ 0 & \text{else}\,. \end{cases} \tag{5.40}$$

The polynomial coefficients for the tabulated quantity q are computed using a $\mathbb{L}_2$ projection within each cell of the quadtree approximation

$$\int_0^1 \int_0^1 \varphi_k(\xi,\eta)q(\rho,\epsilon)d\xi d\eta = M_{km}\hat{q}_m\,, \tag{5.41}$$

with the polynomial coefficients $\hat{x}_m$ for the tabulated quantity q using the element mass-matrix M_{km}
The integrals on the left side are evaluated using Gauss-Legendre quadrature rules. Note that the EOS table can be used several times if the same fluid is used. The different EOS are included in the EOS-libraries in an object-oriented manner. If the $\mathbb{L}_\infty$-Error of the polynomial approximation for the quantity q satisfies the following criterion

$$\max\left(\frac{q_h(\rho,\epsilon) - q(\rho,\epsilon)}{q(\rho,\epsilon)}\right)_\infty > \epsilon, \tag{5.42}$$

the tabulation is sufficient and a further refined approximation is not necessary.

5.3.2.2 Simulation step

During the simulation the iteration of the quantity q (e. g. temperature) is replaced by an interpolation of the polynomial basis as defined by

$$q_h(\rho,\epsilon) = \sum_{m=1}^{(q+1)^2} \varphi_m(\xi,\eta)\hat{q}_m := \varphi_m\hat{q}_m \tag{5.43}$$

in the cell containing the solution with the chosen accuracy. The valid cell for the EOS interpolation is found by a grid traversal through the quadtree until the lowest level is reached that fulfills the requirement concerning the approximation accuracy. In case the tabulation is not accurate enough, the

EOS backend is evaluated with this state. Another approach is to extrapolate the state using the nearest valid state in the table. For a fine table with a good approximation of the EOS this backend is not used often.
An example visualization of a EOS table can be seen in figure 5.5 for the fluid dodecane. In this example the thermodynamic and hydrodynamic equilibrium assumption is used within the multi-phase region (see section 6.4.5 for a complete description). Plotted are the EOS discretization based on element boundaries of the underlying quadtree data structure by black lines. Additionally, the isotherms and the pressure solution within the a specified range is plotted. As the visualization of the EOS table is based on the plot of the element boundaries, the visualization is a bit coarse in some regions.

5.3.3 Adaptive storage of fluid data

The EOS data is stored adaptively within a quadtree data structure that allows for a minimal storage size. For each cell an adaptive polynomial degree up to a selected maximal degree is chosen. The error is evaluated for each polynomial degree and the lowest possible polynomial degree that satisfies the error is chosen as approximation.
In case the refinement of the polynomial degree is not sufficient, another refinement level of quadtrees is added to the data structure. In figure 5.5 the refinement of the EOS table at the critical point and at the saturation lines can be seen clearly. In regions away from these regions a rather coarse approximation of the EOS behavior is sufficient.

5.3.4 Refinement at saturation curve

In case a model for the states inside the multi-phase region is needed, strong gradients occur at the liquid-vapor saturation line. To be able to efficiently resolve these gradients, a cartesian cut-cell method is implemented to approximate the kinks efficiently. The use of cartesian cut-cells reduces the storage size drastically as less approximation levels are needed to obtain the same overall discretization error at the liquid-vapor saturation line. The use of cut-cells typically reduces the maximum quadtree levels needed for a given maximum error by 2 to 3.
The kinks in the EOS derivatives originate from the EOS modeling inside the multi-phase region in which, in the simplest case, an isothermal approximation

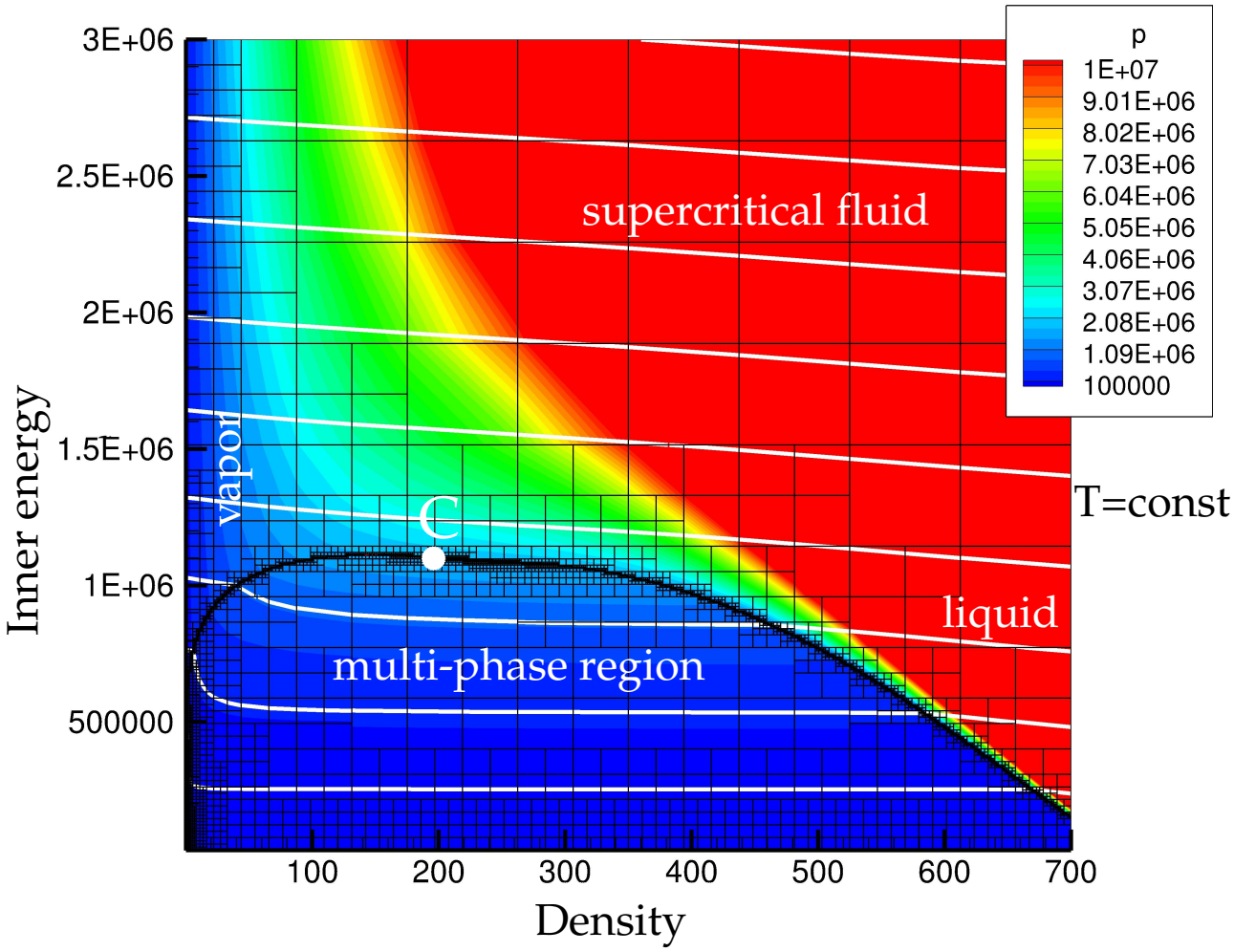


Figure 5.5: View of a sample EOS table for fluid dodecane. The white lines show the isotherms and C marks the critical point. The different fluid regions are marked in the figure. The black lines display the adaptive quadtree grid including the refinement at the liquid and vapor saturation line.

Polynomial degree	Runtime [s] Direct evaluation	EOS tabulation	Overhead
2	77.5	107.5	38,7 %
3	178.6	272.0	52.7 %
4	533.4	750.1	40.6 %

Table 5.2: Assessment of the numerical overhead due to the use of a tabulated EOS in comparison to a direct implementation of an ideal gas for compressible viscous flow.

is chosen (see chapter 6.4.5 for a complete description of the approach). This modification of the original EOS polynomial introduces kinks in the thermodynamic properties. As these kinks are difficult to discretize using a continuous polynomial description, one way is to apply cutcells locally in this region to improve the accuracy and reduce the overall storage size of the EOS table.

5.3.5 Overhead due to use of tabulated EOS

The computational overhead due to the interpolation of the EOS table is assessed by means of a convergence test case in 3D. In this case a simple ideal gas approximation is considered as this is known to be the fastest EOS approximation for the computation of a compressible flow field. The computation time for the direct evaluation is compared to the computation time with the use of tabulated ideal gas.

The numerical overhead as shown in table 5.2 is due to the grid traversal and interpolation of the DG solution in the EOS table as well as the additional calls to specific EOS subroutines. Note that the numerical overhead may change for finer tables due to the increased computation time for the grid traversal. An investigation of the main time consuming components revealed that the main computation time is due to the interpolation of the EOS solution.

The numerical overhead due to the EOS evaluation and the temperature lifting is about 40% compared to the direct evaluation for an ideal gas. This overhead is quite low if one keeps in mind that the quality of the EOS description is increased significantly. Furthermore, the new method is independent of the choice which EOS is used.

5.3.6 Speedup

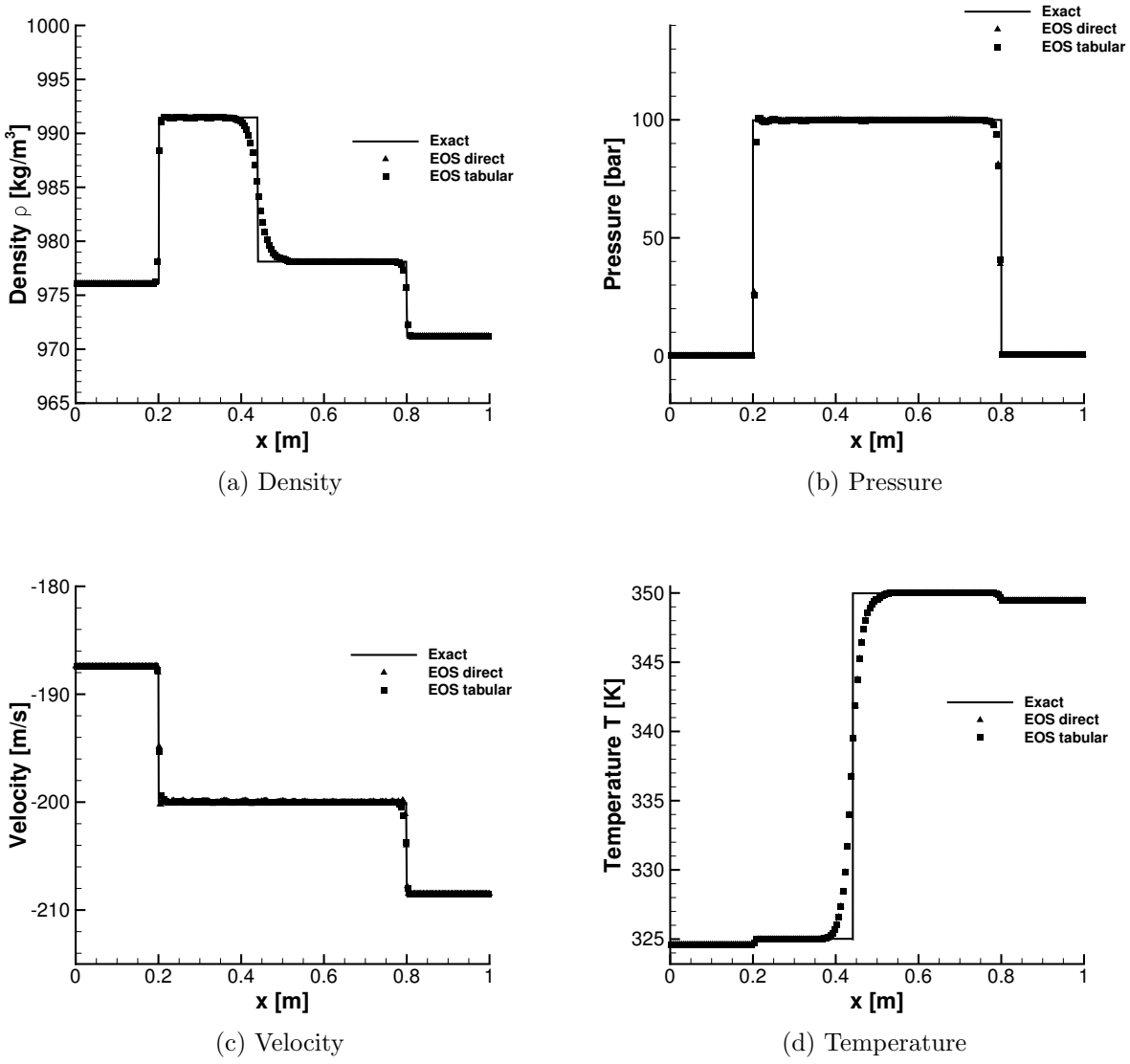
The speedup of the tabulation test case is demonstrated by means of an one-dimensional test case as proposed by Dumbser et al. in [22]. Here, the test case RP-W3 taken is used for the speedup of the tabulation approach. The EOS description is based on the IAPWS-I97 standard for water and steam. We compare here the computation time for the different EOS evaluation programs that are available to the EOS tabulation approach using a target tabulation accuracy of 10^{-5}. A minimum tabulation accuracy of 10^{-3} was chosen in case the target accuracy can not be reached with the maximum number of refinement levels. A Riemann problem in the liquid portion of the EOS is chosen as this part is rather stiff and minor deviations in the EOS behavior become more evident. A simple HLL-Riemann solver is chosen within the whole computational domain. The initial conditions for the water shock-tube problem are defined as

$$(\rho, u, p) = \begin{cases} (976.0968\,{}^{kg}/_{m^3}, -187.40908\,{}^{m}/_{s}, 13279.97\,Pa) \text{ if } x < 0.5 \\ (971.2215\,{}^{kg}/_{m^3}, -208.48529\,{}^{m}/_{s}, 40776.45\,Pa) \text{ else}\,, \end{cases} \tag{5.44}$$

and the computational domain [0,1] is discretized using 50 DG cells with a polynomial degree of $N = 3$, total 200 DOFs. Note that all initial computational states lie in the multi-steam region. This test case is very stringent as pressure jumps of several orders of magnitudes occur together with phase change effects from wet steam to wet vapor. The drastic pressure jumps are due to the stiff EOS for the liquid phase.
The solution obtained with the use of the EOS table is compared to the one with the direct evaluation of the EOS. This comparison is presented in figure 5.6 for a water shock-tube problem. Additionally, the exact solution taken from Dumbser et al. [22] is plotted. The fluctuations in the pressure solution for the EOS table are related to the tabulation accuracy. For this problem, an overall accuracy of 10^{-3} is chosen, that is not enough for the representation of this stiff liquid part of the EOS. For all other variables, the agreement between the direct iteration and the tabulation approach is excellent. The main advantage is the massive saving in the computation time as presented in table 5.3.

5.3.7 Conclusion

The adaptive tabulation method together with the EOS library provides an extensive and simple to use method for the EOS description. The tabulation



(a) Density

(b) Pressure

(c) Velocity

(d) Temperature

Figure 5.6: Riemann validation test case at $t = 3.0 \cdot 10^{-4}$ s. Comparison of the exact solution compared to the solution with direct evaluation of the EOS and the use of the EOS tabulation approach. Note that all EOS evaluation programs use the IAPWS-IF97 standard for water [101] and produce identical results. Thus, only the result for the simulation with CoolProp is shown here.

	fluidcal	CoolProp	FPROPS	REFPROP	EOS table
CPU time [s]	5539.8	2204.3	2891.6	2826.72	1.3
Speedup [-]	4261	1695	2224	2174	–

Table 5.3: Speedup of the tabulation approach for the Riemann problem compared to the direct evaluation of the EOS using different EOS evaluation programs.

of the EOS behavior in terms of the density and the inner energy reduces the total computation time. Due to the high-order polynomial approximation an efficient approximation in smooth regions of the EOS is obtained. Close to kinks or discontinuities the quadtree-based refinement algorithm together with the cartesian cut-cell method provide a good local approximation. The impact of this EOS approximation is rather small as seen in the test case and provides a massive speedup by three order of magnitudes. Thus, this method is essential for the investigation of multi-dimensional test cases.

6 Numerical resolution of the physics at the phase interface

One crucial part of a multi-phase solver is the resolution of the phase interface within the flow field as most of the relevant physical effects occur at or in the vicinity of the phase interface. In the coarse interface approximation considered here, the phase interface represents a discontinuity within the flow field that has to be resolved and kept sharp to represent the typical behavior of multi-phase flow. For a complete resolution of the phase interface the numerical workload would be too high, as length scales of some Angström (10^{-10} m) have to be resolved. This would imply a very fine local grid resolution at the interface combined with tiny time steps for the whole simulation.

One type of this diffuse interface method are phase-field models, as e. g. represented by the Navier-Stokes Korteweg equations [4] or the Baer-Nunziato model [81]. Another approach is to solely resolve the macro-scopic bulk phases and apply jump conditions at the interface. These jump conditions, respective the fluxes that contain the jump conditions, at the interface have to be provided to the macro-scale flow solver by means of special numerical solvers. This can be done by additional ghost cells at the interface as proposed by Fedkiw et al. [30] or with the use of special interface Riemann solver as considered here. A general description of the Riemann problems for real materials is given in Menikoff and Plohr [61].

An overview of a typical phase interface is sketched in figure 6.1. The interface is divided in different parts: Away from the phase interface both bulk phases can be found assuming constant states and a perfectly mixed state. In the vicinity of the interface, a small interface transition region is developing with directional fluid properties [46, 35]. Within these small region, a typical width of the interface is some nanometers, a smeared out density profile is present with micro-scale jump conditions. Such a density profile is sketched in figure 6.1 with the corresponding micro- and macro-scale jump conditions. Note that these jump conditions are different as the macro-scale model does not resolve

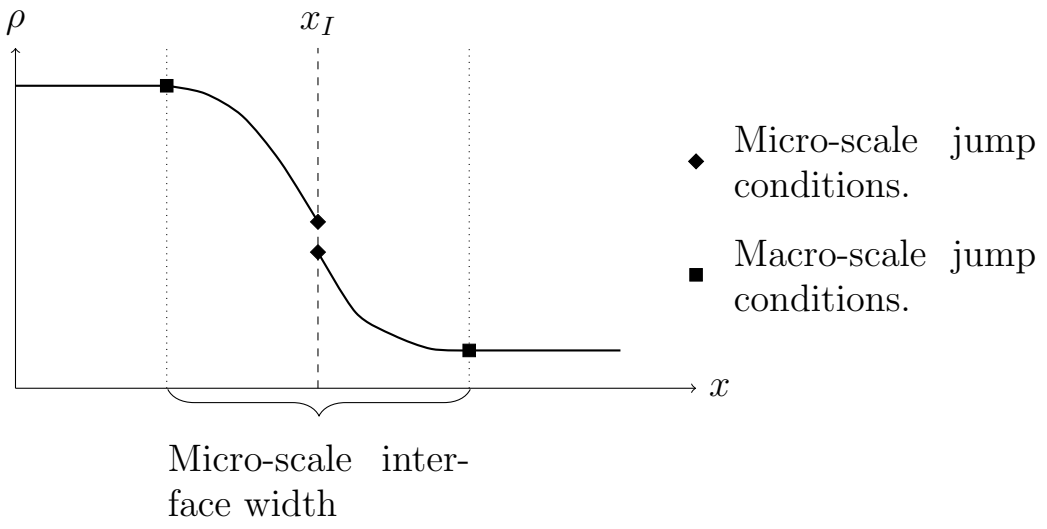


Figure 6.1: Sketch of a typical interface with the macro- and micro-scale jump conditions. x_I is the micro-scale interface position. Note that the typical micro-scale interface width is several nanometers and is not resolved by the macro-scale model. Therefore, the macro-scale jump conditions are applied at the interface.

the small transition region between the bulk phases. In the following we will only consider the case of the mixed bulk phases together with suitable macro-scale jump conditions at the interface.

In the Ghost-Fluid method chosen for the resolution of the multi-phase flows, jump conditions at the phase interface have to be resolved explicitly (jumps in pressure due to surface tension effects and density due to the different fluid phases). These jump conditions ensure that the numerical scheme correctly resolves the discontinuous nature of multi-phase flows including interface related physical effects of surface tension and phase transition that introduce additional difficulties compared to the single-phase investigations.

Starting with a description of the jump conditions at the interface and the coupling procedure, it is shown how to approximate these jump conditions in the context of a Riemann-solver based approach. The simplest case is the interface resolution in case no phase transfer is present as in this case the structure of the Riemann solution is similar to the one in the single-fluid case. After that the description is extended to more complex cases. Several possibilities to include phase transfer effects in the heterogeneous multi-scale algorithm are discussed, ranging from the simple use of an analytical evaporation model (as used in incompressible multi-phase solvers) to a description of phase transfer effects

by an interface Riemann solver. The limits of the considered approaches are discussed with focus on subsonic phase transfer effects. In the last section the different approaches are compared to each other.
The focus of the investigations here are the inviscid Euler equations as this introduces all relevant difficulties especially the jump terms at the interface. The extension to the viscous Navier-Stokes equations is then straightforward as the viscous terms tend to diminish jumps. However, a suitable method to cope with jumping viscosity and heat transfer coefficients at the interface is needed as they may jump across the interface by one or two orders of magnitude.

6.1 Resolution of interface effects at the phase interface

As discussed earlier the phase interface can be resolved using different methods. In the sharp interface approximation considered here based on the Ghost Fluid approximation as introduced by Fedkiw et al. [30], special care has to be taken to the resolution or estimation of the jump conditions at the location of the phase interface. These jump conditions include density jumps due to different phases and pressure jumps related to surface tension as well as phase transfer effects. These jump terms can be considered by a direct resolution of the effects at the interface (e. g. by a direct consideration of the surface forces within the framework of generalized Riemann problems) or by means of volume source terms in the vicinity of the interface.
The magnitude of the density jump is described by the EOS. Typically, one tries to connect the gaseous and liquid states on the saturation lines by a suitable approach. In case two different fluids are coupled at the interface (e. g. acetone droplet within a nitrogen environment), even two different EOS can be used.
Additionally to the density jump at the phase interface, the physical effects of surface tension and phase transition have to be resolved at the interface. These effects are introduced here shortly:

Surface Tension: The magnitude of the pressure jump due to surface tension effects at the interface is described by the Young-Laplace law

$$(\rho(\mathbf{v}\cdot\mathbf{n}_{\mathrm{LS}}-\mathbf{s}_{\mathrm{PB}}\cdot\mathbf{n}_{\mathrm{LS}})\mathbf{v}+p(\rho,T)\mathbf{n}_{\mathrm{LS}})=2\sigma\kappa\mathbf{n}_{\mathrm{LS}}=\Delta p_{\sigma}\mathbf{n}_{\mathrm{LS}} \tag{6.1}$$

and includes the velocity of the phase interface $\mathbf{s}_{\mathrm{PB}}$. The pressure jump $\Delta p_{\sigma}=2\sigma\kappa\mathbf{n}_{\mathrm{LS}}$ is dependent on the interface curvature κ and the surface

tension coefficient σ of the fluid between the gaseous and liquid phase. The surface tension coefficient is dependent on the fluid as well as the temperature of the phase interface and decreases with increasing temperature towards critical conditions. At supercritical conditions a diffuse interface can be investigated and surface tension effects are no longer present.

Different methods are available to include surface tension effects into the multi-phase flow solver. A popular approach is the Continuous Surface Force (CSF) method as introduced by Brackbill [10] that includes the surface tension as a volume source term at the interface. Here, surface tension forces are resolved as surface force within the interface Riemann solver that is applied at the phase interface.

Phase transfer: Similarly to the surface tension effects, additional jump terms for the inclusion of phase transfer effects are present. The magnitude of the phase transfer jump terms is dependent on the local mass transfer rate at the phase interface. In contrast to the surface tension modeling, jump terms have to be considered for all conservation equations. This imposes the additional difficulty that an unique interface velocity has to be estimated that is used for the transport of the interface.

6.2 Coupling to the phase interface

In this section the methodology is described how the physics at the interface are coupled to the flow solver in the bulk phases. This is established by a ghost fluid approach based on the solution of a local Riemann problem at the interface. The solution of the Riemann problem at the interface is introduced in section 6.3 for the case without phase transition and is generalized in section 6.4 for cases with phase transition.
From the local solution of the Riemann problem in the normal direction of the interface two types of information are obtained: The velocity of interface propagation $\boldsymbol{s}_{\mathrm{PB}}$ and the states from both sides of the material interface ($\mathbf{U}_L^*$ and $\mathbf{U}_R^*$). The velocity $\boldsymbol{s}_{\mathrm{PB}}$ in normal direction is needed for the level-set equation. The states are then used to calculate the numerical fluxes in grid cells with an interface. According to the ghost-fluid approach, two sets of data are present in an interface grid cell. One set is used to calculate the numerical flux to fluid 1 and the other the numerical flux to fluid 2. The macro-scale flow solver

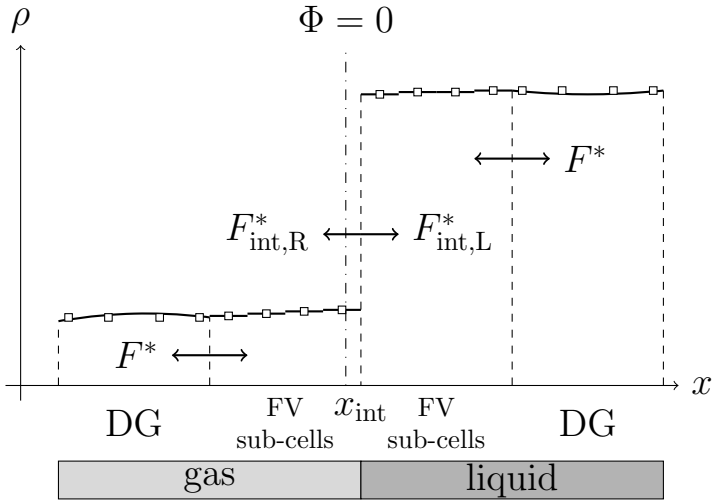


Figure 6.2: One-dimensional treatment of the fluxes at the phase interface. Within the bulk phases the standard single-phases Riemann flux F^* is used. The interface position is marked with $\Phi = 0$. At the interface position the fluxes $F^*_{\text{int,L}}$ and $F^*_{\text{int,R}}$ are applied to the left and right phase.

always operates in a single fluid. The typical wave structures of the Riemann problem are introduced in section 6.3 and 6.4 for the cases with and without phase transition. In the simulations the fluxes directly at the corresponding phase interface are chosen for the flux balance.
The ghost-fluid approach shifts the waves generated in the interface Riemann problem in the different fluids to the grid cell interface of the corresponding fluid. Defined in this way the ghost data generates the proper waves in the bulk phases. Due to the different fluxes from the different sides at a grid cell with a material interface the mixing is avoided, but the conservation can no longer be satisfied locally. A good quality of the determination of the interface velocity is important. The approach considered relies on the original ghost-fluid ideas of Fedkiw et al. [30] and its extension by Merkle and Rohde [62] and by Rohde and Dressel [18] for piecewise constant approximations in one space dimension and in the multi-dimensional case, respectively. In the bulk phases away from the interface, the numerical flux of the DGSEM scheme is calculated by a standard one. The transfer of information from the local interface Riemann problem to the flow solver is illustrated in figure 6.2 in a one-dimensional case and in figure 6.3 for the two-dimensional case.

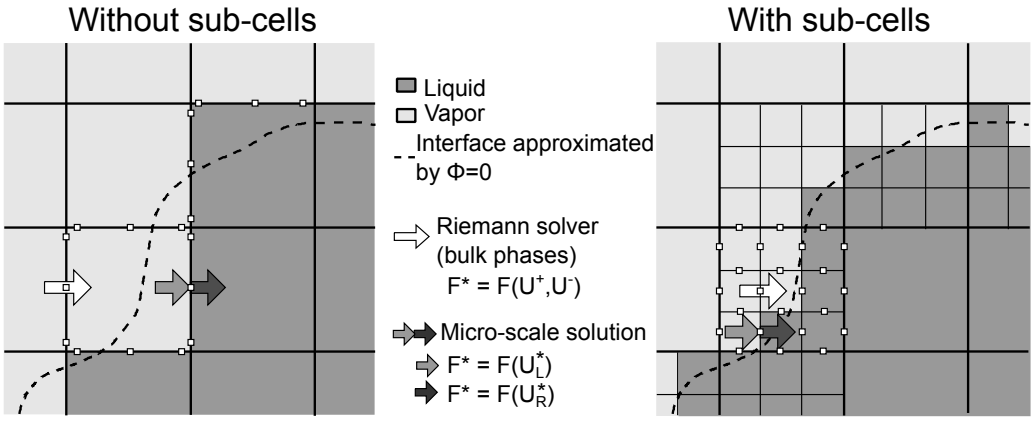


Figure 6.3: Schematic diagram of a typical setting in the two-phase approach for the sharp interface approach: The liquid-vapor interface is approximated by the zero level-set $\Phi = 0$. According to the ghost-fluid approach, the numerical fluxes at the grid cell boundaries are determined either by a standard Riemann solver (bulk phase) or the interface Riemann solver and the ghost-cell data. The white dots visualize the integration points at the element boundaries.

In figure 6.3, $\mathbf{U}_L := \mathbf{U}^-$ and $\mathbf{U}_G := \mathbf{U}^+$ denote the respective states in the two different phases in some quadrature point at the numerical interface. These states are the input for the interface Riemann solver together with the mean interface curvature, if surface tension effects are considered. The multi-phase Riemann solver provides a liquid state $\mathbf{U}_L^*$ and a gaseous state $\mathbf{U}_G^*$ at each quadrature point. Surface tension is resolved as surface force at the interface. In case no sharp interface resolution is chosen, the corresponding interface effects are taken into account by volume source terms at the phase interface. The DGSEM now uses the flux evaluations $\boldsymbol{F}^* = \boldsymbol{F}(\mathbf{U}_L^*)$ for the computation on the liquid element and $\boldsymbol{F}^* = \boldsymbol{F}(\mathbf{U}_G^*)$ on the gaseous element. Inside the finite volume sub-cells the numerical fluxes are used on each face of the sub-cells.
In case the interface position, as defined by the zero iso-contour of the level-set function, has moved across one grid cell, the state in this cell has to be re-defined according to its new fluid state. As no information about the state in the cell is available, this state has to be extrapolated using the adjacent grid cells. This is due to the approximation of the interface in the sharp interface context. The state in this cell is approximated by the mean value of the surrounding cells that are in the same bulk phase

$$\mathbf{U}_{\text{new}} = \frac{1}{\sum_{i=1}^{\text{nCell}} \delta_{i,\text{new}}} \sum_{i=1}^{\text{nCell}} \delta_{i,\text{new}} \mathbf{U}_i \tag{6.2}$$

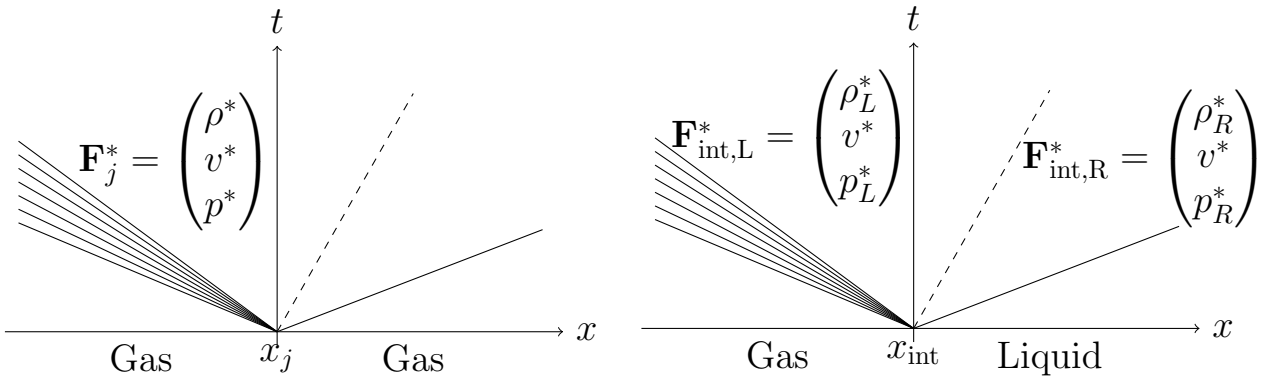


Figure 6.4: Comparison of the Riemann wave pattern. Left: Traditional single-phase Riemann problem. Right: Riemann problem at the phase interface without phase transfer effects.

with

$$\delta_{i,\text{new}} = \begin{cases} 1 \text{ if fluid(i)} = \text{fluid(new)} \\ 0 \text{ else.} \end{cases} \tag{6.3}$$

This state averaging is done for all surrounding nCell cells of the same fluid phase (in 3D: nCell $\in [1, 6]$). To avoid pressure discontinuities due to the state extrapolation, the average pressure is calculated instead of the average total energy.

6.3 Interface resolution without mass transfer

In case no phase transfer effects are considered, the Riemann problem at the phase interface is a standard Riemann problem with 3 traditional waves (shock, contact discontinuity and rarefaction wave). The jump conditions can be calculated using the Rankine Hugoniot jump conditions, but special care has to be taken due to the dissimilar EOS for the liquid and vapor phase. As simple examples for EOS, suitable for an easy simulation of compressible multi-phase flow, the stiffened gas EOS can be named. The aim of these special interface Riemann solvers is to preserve the jump conditions (density and pressure jump) at the phase interface. Here, this is done using non-conservative numerical fluxes at the interface.
Figure 6.4 visualizes one suitable resolution of the phase interface within the traditional Riemann wave fan. The contact discontinuity resolves the phase interface which is a suitable approximation for the case without mass transfer.

In contrast to well known Riemann solution for single phase flows, for an overview of different single-phase Riemann problems see [96], special care has to be taken with the pressure function for multi-phase Riemann solvers. The pressure function may be discontinuous at the phase interface in case two different EOS are coupled there. This coupling approach is often suitable to describe the dissimilar behavior of multi-phase flows for liquid and gaseous phases. Several approaches are introduced in the following, starting from a linearization of the Lax-curves as used in [26] and an adaption of the HLLC Riemann solver as described in [44] combined with a special approximation of the wave speeds.

6.3.1 Lin-Lax interface Riemann solver

The linearized Lax-curves interface Riemann solver, as described by Fechter et al. in [26], is suitable for general EOS, only an estimate of the sound speed is necessary. This makes this Riemann solver also suitable for the application to tabulated EOS as described in section 5.3. In this paper also an exact interface Riemann solver is described which has a considerably higher computational demand due to the non-linear fix-point iteration. In the following, an approximate interface Riemann solver based on the linearization of the Lax-curves is described. The approach is published in Fechter et al. [26] and the numerical derivation of the linearization of the Lax curves is not detailed here.
At the discontinuities of the pressure law, which can be found at phase interfaces, a thorough investigation of the wave structure is needed in order to construct physically correct (entropy) solutions. This will be done by generalizing the standard approach presented, for example in [96].
The solution of the Riemann problem at the interface is given in the linearized case by the equations

$$v^0_{n,L} - \frac{c^0_L}{\rho^0_L}\xi_- = v^0_{n,R} + \frac{c^0_R}{\rho^0_R}\xi_+ \,, \tag{6.4}$$

$$p^0_L + \left(c^0_L\right)^2 \xi_- = p^0_R + \left(c^0_R\right)^2 \xi_+ + 2\sigma\kappa. \tag{6.5}$$

Since $c_{L/R} > 0$ by assumption, the system can be uniquely solved for (ξ_-, ξ_+) yielding

$$\begin{pmatrix}\xi_-\\ \xi_+\end{pmatrix} = \frac{1}{c^0_L c^0_R\left(\frac{c^0_R}{\rho^0_L} + \frac{c^0_L}{\rho^0_R}\right)} \begin{pmatrix}\left(c^0_R\right)^2 & -\frac{c^0_R}{\rho^0_R}\\ \left(c^0_L\right)^2 & \frac{c^0_L}{\rho^0_L}\end{pmatrix} \cdot \begin{pmatrix}\Delta v^0\\ \Delta p^0\end{pmatrix}, \tag{6.6}$$

where $\Delta v^0_n = v^0_{n,L} - v^0_{n,R}$ and $\Delta p^0 = p^0_L - p^0_R - 2\sigma\kappa$ for a three-dimensional flow field.

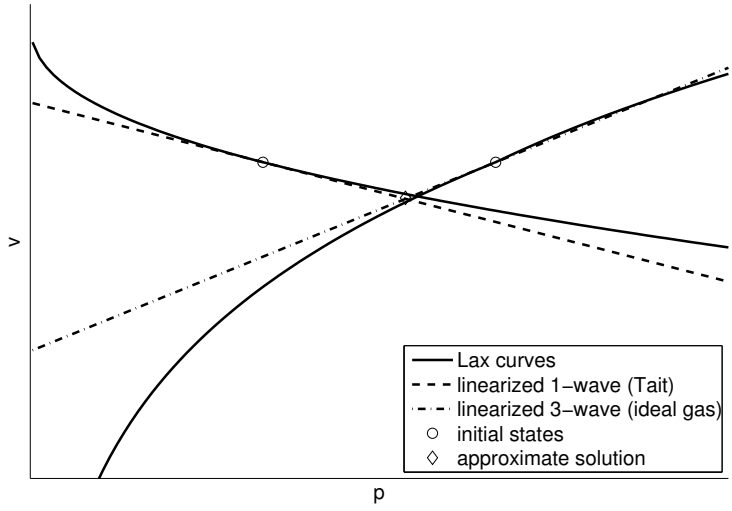


Figure 6.5: Illustration of the approximate solution of the Riemann problem at a planar interface ($\kappa = 0$) using linearized wave curves. $p_1 = p_{Tait}$ and $p_2 = p_{ideal}$. Graph is taken from [26].

Geometrically this approach means that the solution to the Riemann problem at the interface is approximated by the intersection point of the projected tangent curves in the (v_n, p)-plane, see figure 6.5. Note that the intersection point of the tangent curves does not lie on the wave curves. The solution of the approximate Riemann solver is therefore not connected to the initial states by a wave of the first or third family respectively.
Obviously, the error of the above approximation strategy increases with the magnitude of the initial discontinuities in the Riemann problem (densities, pressures, velocities). Furthermore, it depends on the pressure law (i. e. the shape of the wave curves in figure 6.5). Concerning the computational effort, the approximate solution of the Riemann problem at the interface can be obtained by a single evaluation of equation (6.6). Another major advantage of the linearization approach is the fact that only the initial values of pressures and sound speeds are needed. It is therefore possible to use this Riemann solver also for very general pressure laws without any additional computational effort, as long as the corresponding speed of sound is known. In each time step the computational costs are essentially reduced to one evaluation of each pressure function, one evaluation of the corresponding speeds of sound and the solution of a linear 2×2-system. The solver does not require additional evaluations of pressure or sound speed functions at intermediate states.

6.3.2 HLLC interface Riemann solver with corrected wave speeds

A similar ansatz for the interface Riemann solver was done by Hu et al. [44] by extending the HLLC Riemann solver with a suitable approximation for the wavespeeds in the star-region. First the HLLC description is introduced that is then extended for the use at the phase interface. Two different methods for the approximation of the wave speeds are presented.
The original description of Hu et al. [44] does not take the surface tension forces into account for the approximation of the HLL wave speeds. The second approximation, based on the linearized Lax-curves (see section 6.3.1), does take the pressure jump into account.

6.3.2.1 HLLC Riemann solver with surface forces

First the HLLC Riemann solver description is recalled following the description given in Toro [96] to fix the notation used in the following description. The HLLC Riemann solver is chosen as it can be applied to a large variety of problems and is quite robust. It is assumed that the solution to the Riemann problem with left state $\mathbf{U}_L$ and right state $\mathbf{U}_R$ is given by

$$\mathbf{U}_{HLLC} = \begin{cases} \mathbf{U}_L & \text{if } x < S_L t \\ \mathbf{U}_L^* & \text{if } S_L t \leq x < S_M t \\ \mathbf{U}_R^* & \text{if } S_M t \leq x < S_R t \\ \mathbf{U}_R & \text{if } x \geq S_R t \end{cases}. \tag{6.7}$$

The corresponding interface flux at $x = 0$ is then given by

$$\mathbf{F}_{HLLC} = \begin{cases} \mathbf{F}(\mathbf{U}_L) & \text{if } S_L > 0 \\ \mathbf{F}(\mathbf{U}_L^*) & \text{if } S_L \leq 0 < S_M \\ \mathbf{F}(\mathbf{U}_R^*) & \text{if } S_M \leq 0 < S_R \\ \mathbf{F}(\mathbf{U}_R) & \text{if } S_R \leq 0 \end{cases}. \tag{6.8}$$

Due to the two-shock approximation, it is known that $\mathbf{U}_L^*$ and $\mathbf{U}_R^*$ have to fulfill the Rankine-Hugoniot jump conditions, i. e.

$$\mathbf{F}(\mathbf{U}_L^*) - \mathbf{F}(\mathbf{U}_L) = S_L(\mathbf{U}_L^* - \mathbf{U}_L), \tag{6.9a}$$
$$\mathbf{F}(\mathbf{U}_R^*) - \mathbf{F}(\mathbf{U}_R) = S_R(\mathbf{U}_R^* - \mathbf{U}_R). \tag{6.9b}$$

Due to the surface forces, it is known furthermore that the solution must fulfill the following jump conditions at the contact line

$$v_L^* = v_R^* = S_M, \tag{6.10a}$$
$$p_L^* - p_R^* = \Delta p_\sigma, \tag{6.10b}$$

where Δp_σ denotes a jump condition associated with the modeling of the surface forces. As an example in the case of two-fluid flow with surface tension, Δp_σ is given by $\Delta p_\sigma = 2\sigma\kappa$ for three-dimensional flow, where κ is the local mean curvature at the interface, $\sigma > 0$ is the constant surface tension coefficient. Note that for general surface forces, Δp_σ is allowed to depend on $\mathbf{U}_L$ and $\mathbf{U}_R$ as well as on geometric variables (such as position, normal vector or curvature), but does not depend on the intermediate states $\mathbf{U}_L^*$ and $\mathbf{U}_R^*$.
Inserting (6.10a) in (6.9),

$$\rho_L^* S_M - \rho_L v_L = S_L(\rho_L^* - \rho_L), \tag{6.11a}$$
$$\rho_L^* S_M^2 + p_L^* - \rho_L v_L^2 - p_l = S_L(\rho_L^* S_M - \rho_L v_L), \tag{6.11b}$$
$$S_M(E_L^* + p_L^*) - v_L(E_L + p_L) = S_L(E_L^* - E_L). \tag{6.11c}$$

Solving equation (6.11a) for ρ_L^*, one gets

$$\rho_L^* = \rho_L \frac{S_L - v_L}{S_L - S_M}. \tag{6.12a}$$

Now, solving (6.11b) for p_L^* as well as (6.11c) for E_L^* and inserting (6.12a),

$$p_L^* = p_L + \rho_L S_M(S_L - S_M) - \rho_L v_L(S_L - v_L), \tag{6.12b}$$
$$E_L^* = \frac{E_L(S_L - v_L) - v_L p_L + S_M p_L^*}{S_L - S_M}. \tag{6.12c}$$

Analogously,

$$\rho_R^* = \rho_R \frac{S_R - v_R}{S_R - S_M}, \tag{6.13a}$$
$$p_R^* = p_R + \rho_R S_M(S_R - S_M) - \rho_R v_R(S_R - v_R), \tag{6.13b}$$
$$E_R^* = \frac{E_R(S_R - v_R) - v_R p_R + S_M p_R^*}{S_R - S_M}. \tag{6.13c}$$

Inserting (6.12b) and (6.13b) in (6.10b), the speed of the contact line is given by

$$S_M = \frac{\rho_R v_R(S_R - v_R) - \rho_L v_L(S_L - v_L) + p_L - p_R - \Delta p_\sigma}{\rho_R(S_R - v_R) - \rho_L(S_L - v_L)}. \tag{6.14}$$

Once appropriate estimates for the wave speeds S_L and S_R are available, the intercell flux can be determined by (6.8). As for the standard HLLC solver, the wave speeds are chosen to

$$S_L = \min[v_L - c_L, \tilde{v}_L - \tilde{c}_L], \tag{6.15a}$$
$$S_R = \max[v_R + c_R, \tilde{v}_R + \tilde{c}_R], \tag{6.15b}$$

where $\tilde{v}_{L/R}$ and $\tilde{c}_{L/R}$ are estimates for the velocity and sound speed of the intermediate states $\mathbf{U}^*_{L/R}$ respectively. In the following several approaches for the determination of these quantities are introduced.

6.3.2.2 Roe linearization based wave-speed estimate

In the paper of Hu et al. [44] the wave speeds for the intermediate states are approximated using a generalized Roe average based on the approach of Glaister [33] and Roe [78]. The approach is summarized here for completeness based on the paper of Hu.
At the interface, the sound speed can be written as follows

$$c^2 = \Psi + \Gamma \left(\frac{p}{\rho}\right), \tag{6.16}$$

with the approximations of

$$\Psi = \left.\frac{\partial p}{\partial \rho}\right|_e, \tag{6.17a}$$
$$\Gamma = \left.\frac{1}{\rho}\frac{\partial p}{\partial e}\right|_\rho. \tag{6.17b}$$

In the following the Roe-averaged values are denoted by $\tilde{}$. They can be estimated as follows

$$\tilde{\rho} = \sqrt{\rho_L \rho_R} \tag{6.18}$$
$$\tilde{f} = \mu(f) = \frac{\sqrt{\rho_L} f_L + \sqrt{\rho_R} f_R}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad f = v,\, H\,. \tag{6.19}$$

Similarly the estimate for p/ρ-term is given by

$$\widetilde{\left(\frac{p}{\rho}\right)} = \mu\left(\frac{p}{\rho}\right) + \frac{1}{2}\left(\frac{v_R - v_L}{\sqrt{\rho_L} + \sqrt{\rho_R}}\right)^2. \tag{6.20}$$

In the following only the approach for the two-material approximation is considered, as this introduces the difficulty to approximate a jumping pressure function. With this information, the unknown functions $\tilde{\Psi}$ and $\tilde{\Gamma}$ can be approximated as

$$\tilde{\Psi} = \mu(\Psi) \qquad \text{and} \qquad \tilde{\Gamma} = \mu(\Gamma). \tag{6.21}$$

Now the sound speed can be estimated according to equation 6.16 for the wave speed estimate in the HLLC solver. Specifically, for an ideal gas, $p = \Gamma\rho\epsilon$ with $\Gamma = \gamma - 1$. Eq. (21) gives $\tilde{\Gamma} = \Gamma$ and $\tilde{\Psi} = \Gamma\mu(e)$.
Due to the linearization used at the interface, this Riemann interface solver is dependent on the used EOS. This implies that the Riemann solver has to be derived every time a different EOS is used. The linearization is easy to do for linear EOS, like the ideal gas approximation or the stiffened gas equation, but is getting more difficult for advanced, more realistic EOS.

6.3.2.3 Lin-Lax based wave-speed approximation

Another possibility to determine the approximate wave-speeds that are needed for the HLLC Riemann solver is to use an approximation based on the linearized Lax-curves. The approximate quantities $\mathbf{U}_L$ and $\mathbf{U}_R$ are determined by a linearization of the Lax-curves, see [26] and section 6.3.1 for more details. This yields

$$\xi_L = \frac{c_R\rho_R(v_L - v_R) - (p_L - p_R - \Delta p_\sigma)}{c_L\left(c_R\frac{\rho_R}{\rho_L} + c_L\right)}, \quad \xi_R = \frac{c_L\rho_L(v_L - v_R) + (p_L - p_R - \Delta p_\sigma)}{c_R\left(c_R + c_L\frac{\rho_L}{\rho_R}\right)}, \tag{6.22a}$$

$$\tilde{\rho}_L = \rho_L + \xi_L, \qquad \tilde{\rho}_R = \rho_R + \xi_R, \tag{6.22b}$$

$$\tilde{v}_L = v_L - \frac{c_L}{\rho_L}\xi_L, \qquad \tilde{v}_R = v_R + \frac{c_R}{\rho_R}\xi_R, \tag{6.22c}$$

$$\tilde{p}_L = p_L + c_L^2\xi_L, \qquad \tilde{p}_R = p_R + c_R^2\xi_R. \tag{6.22d}$$

To recover the total energy $\tilde{E}_{L/R}$, the equation of state $p(\rho, \epsilon)$ is used to determine $\tilde{\epsilon}_{L/R}$ and thus $\tilde{E}_{L/R} = \tilde{\rho}_{L/R}\tilde{\epsilon}_{L/R} + \frac{1}{2}\tilde{\rho}_{L/R}\tilde{v}_{L/R}^2$. With this information it is possible to calculate the sound speed estimate $\tilde{c}$ and, hence, the values for the signal speeds in the HLLC solver framework.

6.3.3 One-dimensional validation test cases

To show the applicability of the described linear interface Riemann solvers without phase transition, several one-dimensional test cases (Riemann problems) are considered that include two-material interactions without considering phase transfer effects. As not noted otherwise, the ideal gas approximation is used for the gaseous phase and the Tait EOS for the liquid phase are chosen as EOS.
Here, only a small portion of investigated Riemann solvers is shown. For more examples and further Riemann solvers, that are based on a relaxation method, the reader is referred to the paper of Fechter et al. [26].

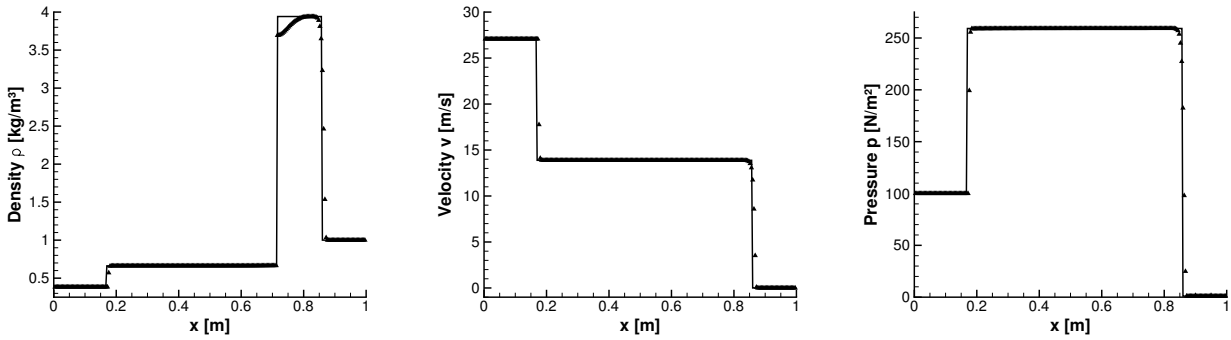
6.3.3.1 Strong transmitted shock without surface forces

The first test case is taken from [44] and is denoted there as case II-A. The impact of a shock wave on the material interface results in a strong transmitted shock wave. Both fluids are modeled by the ideal gas equation of state with coefficient $\gamma = 1.667$. As only gases are considered, no surface tension effects are considered. The initial conditions are given by

$$(\rho, v, p, \gamma) = \begin{cases} (0.384, 27.077, 100., 1.667) & \text{if } 0 \leq x \leq 0.3 \\ (1.0, 0.0, 1.0, 1.667) & \text{else.} \end{cases} \tag{6.23}$$

The purpose of this test case is to compare the performance of the new wave speed estimates for the HLLC interface solver to the well known HLLC interface solver with Roe-type wave estimate as presented in [44]. Since the HLLC solver with Roe wave estimate does not provide a possibility to include surface forces, this test case considers the impact of a shock onto a planar material interface. Figure 6.6 shows the resulting density, velocity and pressure profiles for the HLLC interface solver with three different wave speed estimates at time $t = 0.03$. The results of the top row were obtained using the linearized Riemann solver (LRS) to estimate S_L and S_R. The second row shows the results using the classical HLLC interface solver with Roe type wave estimate.
One can clearly see that there is no difference between the classical HLLC solver with Roe wave speed estimate and the HLLC solver with linear wave speed reconstruction. Especially, both solvers fulfill the required equality of pressure and velocity at the material interface. However, note that the test case is chosen such that the linear interface solver LRS (see section 6.3.1) used alone will not be able to resolve the presented test case setting. Nevertheless, their performance as subsolver within the framework of a HLLC-solver is satisfactory.

HLLC with LRS wave estimate:



HLLC with Roe wave estimate [44]:

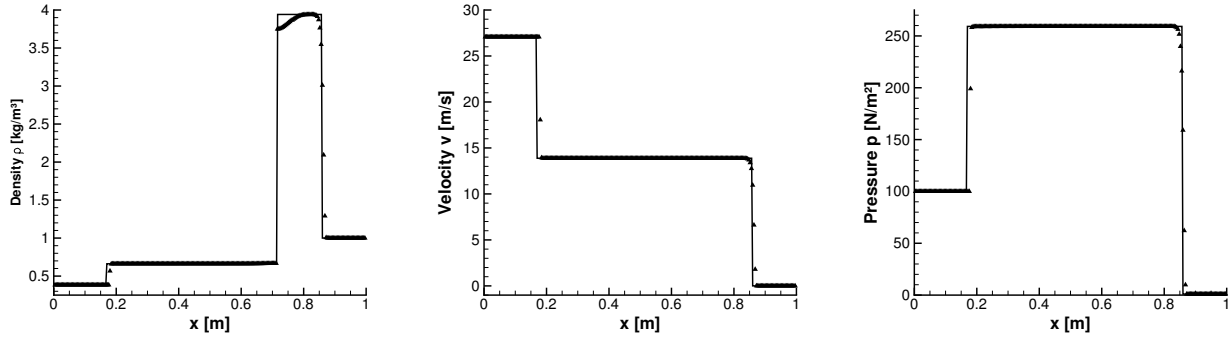


Figure 6.6: Results of the first one-dimensional shock-droplet interaction test case for the HLLC two-phase Riemann solver with two wave speed estimates. Left: density, middle: velocity, right: pressure. The solid line is the exact solution.

6.3.3.2 Multi-phase test case with surface tension

The second test case is chosen in analogy to the shock-droplet interaction problem presented in section 7.1.3. A Mach 3 shock in the gas phase is impinging onto a liquid droplet. Additionally, strong surface tension forces are present at the interface in this test case. The surface tension effect is modeled by a pressure jump of $\Delta p_\sigma = 1$ at the interface position in the one-dimensional framework. For the gas phase, an ideal gas approximation is used while for the liquid phase the Tait EOS is used.
The initial conditions are given by

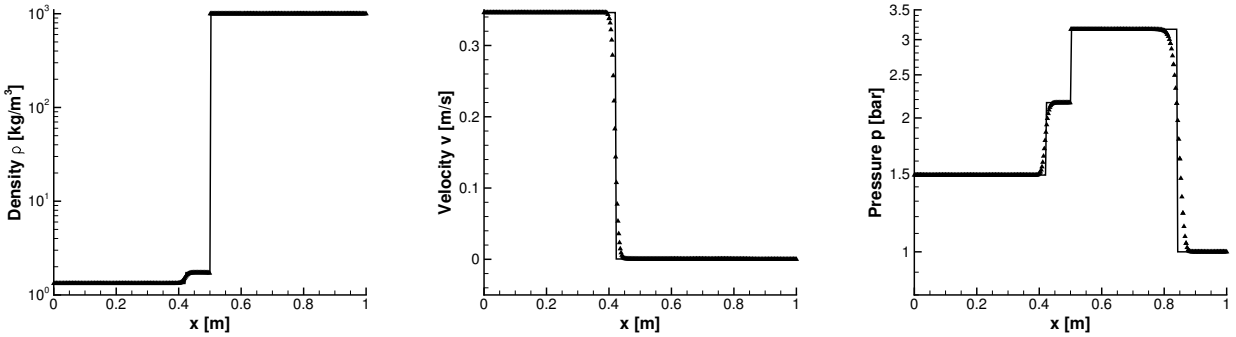
$$(\rho, v, p, \gamma) = \begin{cases} (1.32575, 0.246215, 1.48783, 1.4) & \text{if } 0 \leq x \leq 0.5 \\ (1000, 0.0, 1.0, 7.15) & \text{else.} \end{cases}$$
$$\Delta p_\sigma = 1 \tag{6.24}$$

The non-dimensional parameters of the Tait EOS are chosen to $B = 3310$, $A = 1$ and $\rho_0 = 1000$. The final simulation time is $t = 0.07$. In figure 6.7 the plots for density, pressure and velocity at the final simulation time are shown in comparison to the exact solution. In the top row, the results were calculated using the exact Riemann solver (ERS). In the second row the linearized Riemann solver (LRS) is applied directly to the problem and in the third row the LRS solver is used as wave speed estimate for the HLLC solver. In this way the surface tension effects are included in the wave speed estimate for the HLLC solver.
All three Riemann solvers reproduce the exact solution of the Riemann problem including the surface tension related pressure jump at the interface. The solution is oscillation free despite the large pressure jump at the interface. The HLLC with LRS wave estimate is the most robust approximation.

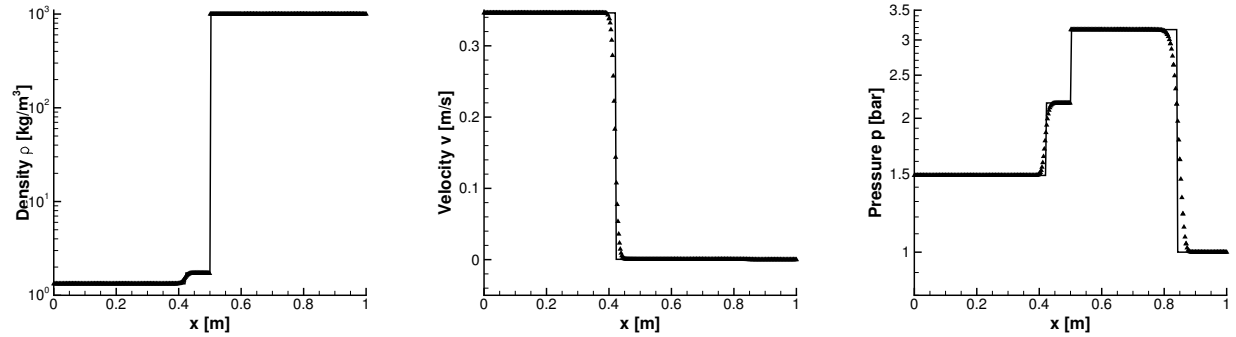
6.3.3.3 Multi-phase test case with strong shock

The third example is test case III-B taken from [44] without surface forces. Here, it is assumed to have a liquid phase to the left of the interface, modeled by the Tait equation of state with $\gamma_l = 7.15$, $k = 3310$ and $p_0 = 1$. The parameter $\rho_0 = \rho \left(\frac{p-p_0}{k} + 1\right)^{-\frac{1}{\gamma_l}}$ is determined by the initial conditions of the left state. The right fluid is assumed to be an ideal gas with $\gamma_g = 1.4$. The

Exact interface Riemann solver (ERS):



Approximate Riemann solver based on linearization of Lax curves (LRS):



HLLC with LRS wave estimate:

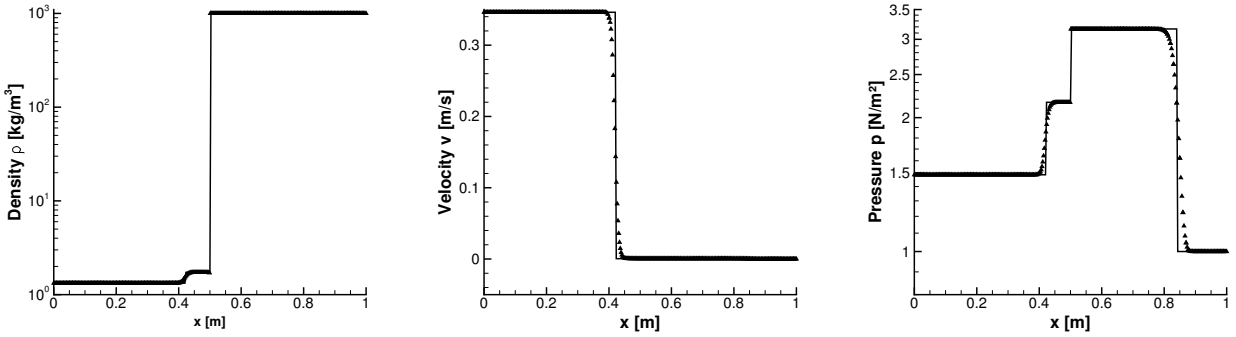


Figure 6.7: Results of the second one-dimensional shock-droplet interaction test case for three two-phase Riemann solvers. Left: density, middle: velocity, right: pressure. The solid line represents the exact solution and the triangles the numerical approximation.

initial conditions for this test case are given by

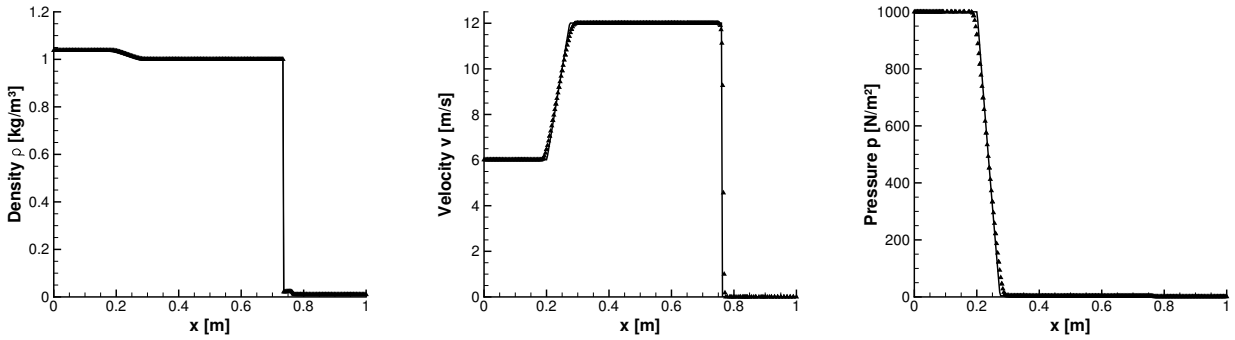
$$(\rho, v, p, \gamma) = \begin{cases} (1.0376, 6.0151, 1000, 7.15) & \text{if } 0 \leq x \leq 0.7 \\ (0.001, 0.0, 1.0, 1.4) & \text{else.} \end{cases} \tag{6.25}$$

For this test case, the interface HLLC solver with surface forces is used as presented in section 6.3.2. To estimate the wave speeds S_L and S_R, on the one hand the Roe-type estimate of Hu et. al. [44] without consideration of the pressure jump at the interface is used. On the other hand, the linear wave speed estimate LRS, capable to deal with the pressure jump due to the surface forces, is used.
Figure 6.8 shows the resulting density, velocity and pressure profiles for the HLLC interface solver with three different wave speed estimates at time $t = 0.003$. The results of the top row were obtained using the linearized Riemann solver (LRS) directly. In the second row the combination of linearized Riemann solver (LRS) and HLLC with surface forces is chosen. Finally, the third row shows the results using the extended HLLC interface solver together with the Roe type wave estimate of [44]. It can be observed that for all variants the predictions are comparable and all Riemann solvers reproduce the exact solution.

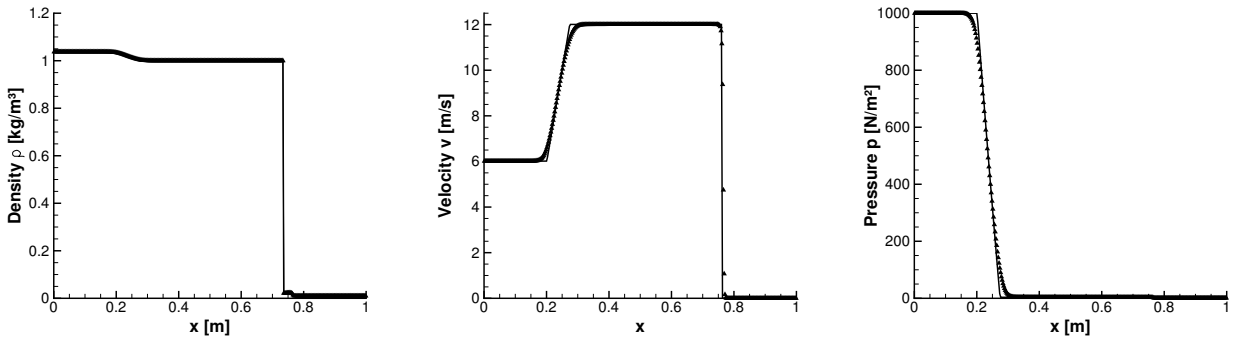
6.3.4 Conclusion

The here interface Riemann solvers described here provide a simple and robust approximation of the phase interface for surface tension driven flows without phase transition. The simple linear Riemann solver, based on the linearization of the Lax curves, fails in case strong shocks are present. However, the LRS solver can be used as sub-solver for the wave speeds within the HLLC framework to improve the accuracy of the HLLC solver. This is advantageous in case surface tension effects are resolved as this sub-solver takes the pressure jump into account and the estimation of the HLL wave speeds is more accurate.
These approaches can be applied to general EOS providing a powerful tool for the resolution of the phase interface. Due to the neglect of phase transfer effects, the traditional structure of single-phase Riemann problems is retained. In the case without phase transfer effects, the material and phase boundary coincide.

Approximate Riemann solver based on linearization of Lax curves (LRS):



HLLC with LRS wave estimate:



HLLC with Roe wave estimate:

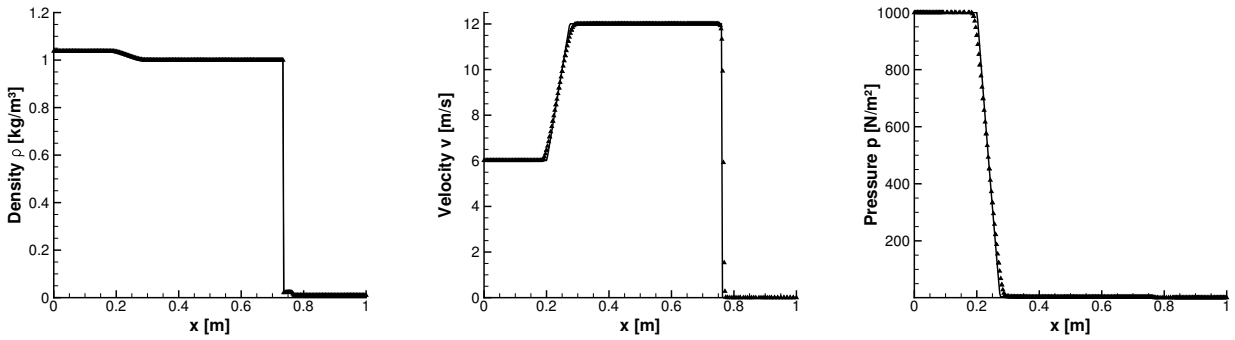


Figure 6.8: Results of the third one-dimensional shock-droplet interaction test case for three two-phase Riemann solvers. Left: density, middle: velocity, right: pressure. The solid line represents the exact solution and the triangles the numerical approximation.

6.4 Interface resolution with mass transfer

One major difficulty in the context of compressible multi-phase flow is to find a thermodynamically consistent way to describe phase transfer effects. In incompressible multi-phase solvers, this problem is circumvented, as no EOS has to be used (as the hydrodynamics are decoupled from the thermodynamics). In this case simple analytical models are used to estimate the phase transfer mass flux rate that e. g. go back to Schrage [82]. The estimated mass transfer rate is then applied in the flow field using source terms in both bulk phases, similar to the surface tension modeling according to the CSF model [42]. This is the usual numerical strategy to include phase transfer effects in the context of incompressible multi-phase solvers.
The situation at the interface can be compared (to some extent) to the occurrence of shock waves in compressible fluid mechanics. The analogy is shown in figure 6.9 comparing the occurrence of shock waves in fluid mechanics with the phase transfer in the scope of the evaporation wave theory. At a shock wave, the supersonic flow is compressed to a higher density and a subsonic fluid velocity. There, material is transported/compressed into the state with a higher pressure. A similarity can be seen at evaporation/condensation waves. Due to diffusion processes, mass is transported from the liquid to the gaseous phase (or vice versa). In contrast to the occurrence at a shock wave, the complete evaporation process is purely subsonic, as the maximum evaporation rate is given by the Chapman-Jouguet (CJ) condition. This maximum evaporation rate induces clearly subsonic velocities in both phases (supersonic evaporation/condensation is not possible due to CJ condition). This is in contrast to the sub- and supersonic velocities at a shock wave.
With a given maximum evaporation rate according to the Chapman-Jouguet condition (see section 6.4.2), it is clear that mass transfer is a (comparatively) slow process that is driven by diffusion between the bulk phases. The rate of mass transfer can be expressed dimensionless using the Sherwood number defined as

$$Sh = \frac{KL}{D} = \frac{\text{Convective mass transfer coefficient}}{\text{Diffusive mass transfer coefficient}}\,, \tag{6.26}$$

that is typically done for evaporation into an environment consisting of a different species. In this context, K denotes the mass transfer coefficient, D the mass diffusivity of the evaporated species in the surrounding species and L a length scale of the problem.

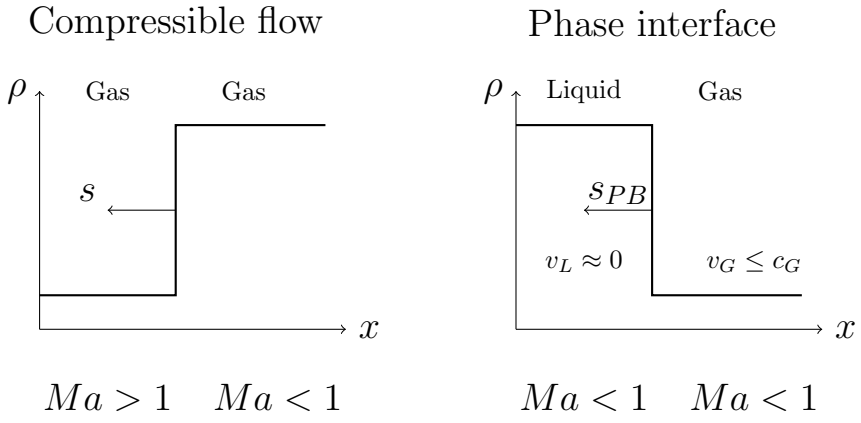


Figure 6.9: Analogy between a shock wave in compressible fluid mechanics (left) and at evaporation waves (right).

6.4.1 Evaporation modeling concept and Riemann wave structure

Figure 6.10 visualizes the evaporation modeling concept used in the sharp interface context. The liquid evaporates and causes a mass flux from the liquid phase to a thin gaseous layer surrounding the droplet. This layer consists of newly evaporated vapor before it mixes with the surrounding gas (that can be another species). This additional layer of newly evaporated vapor causes a formal splitting of the *phase boundary* (interface between gaseous and liquid phase) and *material boundary* (interface between evaporated and surrounding fluid) in the Riemann wave fan. The phase boundary coincides with the intersection between liquid and gaseous bulk phase. The material boundary is the boundary between the layer of newly evaporated liquid and the surrounding gas before mixing processes begins. This necessitates an additional wave and state in the Riemann solution at the interface.
The structure of the resulting Riemann problem is visualized in figure 6.11. An additional evaporation wave (marked with PB) is included in the Riemann problem describing the phase boundary while the wave MI describes the material boundary (note that in the case without phase transfer effects, the material and phase boundary coincide). The additional evaporation wave necessitates the resolution of an additional state $\mathbf{U}^\sharp$ in the Riemann wave fan, whose estimation will be the main difficulty in the description of the resulting Riemann problem. Another difficulty is the simultaneous estimation of the wave speed of the evaporation wave S^* that has to be approximated using theoretical or numerical models that guarantees a thermodynamic suitable solution.

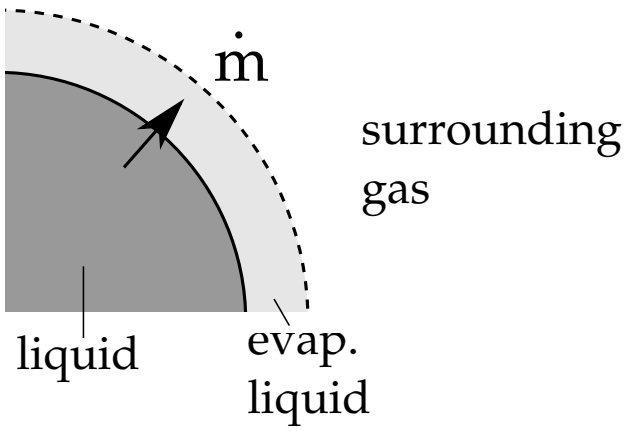


Figure 6.10: Modeling concept for the evaporation wave theory. The liquid evaporates and causes a thin layer of evaporated gas around the droplet. Later on this evaporated gas mixes with the surrounding gas that can be a different fluid. The solid line visualizes the phase boundary and the dashed line the material boundary. These different states can be identified in the Riemann wave structure, see figure 6.11.

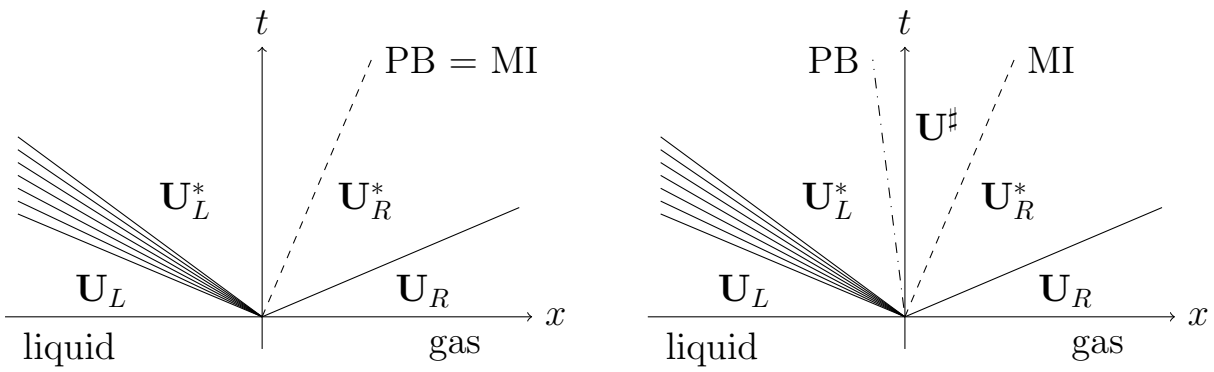


Figure 6.11: Comparison of the Riemann wave pattern for multi-phase problems with and without phase transfer effects. "MI" marks the material interface, "PB" the phase interface. Left: Multi-phase Riemann wave pattern without phase transfer effects. Right: Multi-phase Riemann wave pattern with phase transfer effects. Note that the material and phase boundary do not coincide any more for cases with phase transition.

Different approaches are compared for an investigation of the best suited way to add phase transfer effects into a sharp interface model that is suitable for compressible flow. All evaporation modeling approaches are compared to the exact modeling approach using an exact kinetic solver as described by Zeiler and Rohde [28]. The main numerical problem is that in case of phase transition the strictly hyperbolic character of the fluid equations is lost, as for the resolution of phase transfer the elliptical region inside the multi-phase region of the EOS description has to be considered. In this region the sound speed c is not defined. The major difficulty is to approximate this region in a thermodynamic consistent way such that the basic thermodynamic laws are not violated.

In the following, different methods to resolve the mass transfer at the phase interface are presented. The first method is a simple modeling of mass transfer by a source term distribution next to the interface. The mass transfer rate is approximated by an analytical evaporation model or a kinetic relation. The second method is a relaxation approach for the mixture Baer-Nunziato equations as proposed by Saurel et al. [81]. It is assumed that the state inside the multi-phase region can be described by a mixture of the liquid and vapor state. The third method is to use basically no model for the evaporation process and use the underlying EOS to describe phase transfer effects. Thermodynamic equilibrium is thereby assumed inside the multi-phase region. Through this assumption a hyperbolic EOS description is obtained and standard hyperbolic numerical methods can be applied (similar to single-phase flows). The fourth approach is to apply a micro-solver at the interface that takes the non-convex pressure function in the multi-phase region into account. This approach allows for an exact solution of the interface Riemann problem with phase transfer. Because of that, this case is seen as reference case. An approximate solution for the wave fan is obtained by introducing an additional evaporation wave inside the HLL wave fan. Note that this is not a complete list of all phase transfer modeling approaches. Many more exist in literature. For some diffuse interface approaches, e. g. phase field models, no explicit modeling of phase transfer effects is necessary. However, these approaches require a fine resolution of the phase interface to be able to reproduce a realistic interface thickness.

In the following only evaporation phenomena will be discussed in detail. The extension to condensation phenomena is straightforward as the structure of the evaporation modeling remains the same. The only difference is that the mass flux is directed in the opposite direction and the relative position of the phase transfer and contact discontinuity wave changes.

6.4.2 Physical limits of mass transfer

The maximum possible mass flux at the interface is given by the Chapman-Jouguet condition [13] that describes the physical limit of mass transfer. This condition for the maximum evaporation velocity is equivalent to the condition of maximum entropy production and describes physically the state in which a choked flow is obtained on the gaseous side of the interface.
This condition can be visualized in the p-$1/\rho$-diagram as shown in figure 6.12. In the figure the point A describes the starting point while point B describes the end point of the Maxwell line. The Maxwell line describes the phase equilibrium, i. e. the points with no phase transfer. The other extreme, the maximum phase transfer, is marked by a connection of the point A with the CJ point (Chapman-Jouguet point). This point of maximum mass transfer is defined by the condition that the detonation adiabat touches the Rayleigh line that connects the states within the multi-phase region. Physically, all phase transfer solutions lie between these two limits. The gradient of the connection inside the multi-phase region determines the amount of mass transfer involved in the process. The starting and end points, A and B, do not necessarily have to lie on the saturation lines.
Another interpretation of the Chapman-Jouguet condition is that this condition provides a choked flow on the gaseous side of the phase interface. This means that it is not possible to enlarge the amount of evaporation any more. This condition can be seen by a sonic velocity on the gaseous side of the interface.
The maximum mass transfer rate, i. e. the mass transfer rate that reaches sonic conditions on the gaseous side, can be estimated by the smaller slope of the Rayleigh line. The slope can be expressed in a coordinate system moving with the interface by

$$-j^2 = \frac{p_2 - p_1}{\frac{1}{\rho_2} - \frac{1}{\rho_1}}\,, \tag{6.27}$$

using the two initial states in the bulk phases left and right of the interface.

6.4.3 Linear interface solver combined with theoretical evaporation model

The simplest model is to combine two EOS, that are not capable to describe the multi-phase region, with an analytical evaporation model that determines the amount of phase transition. The phase transfer effects are then modeled

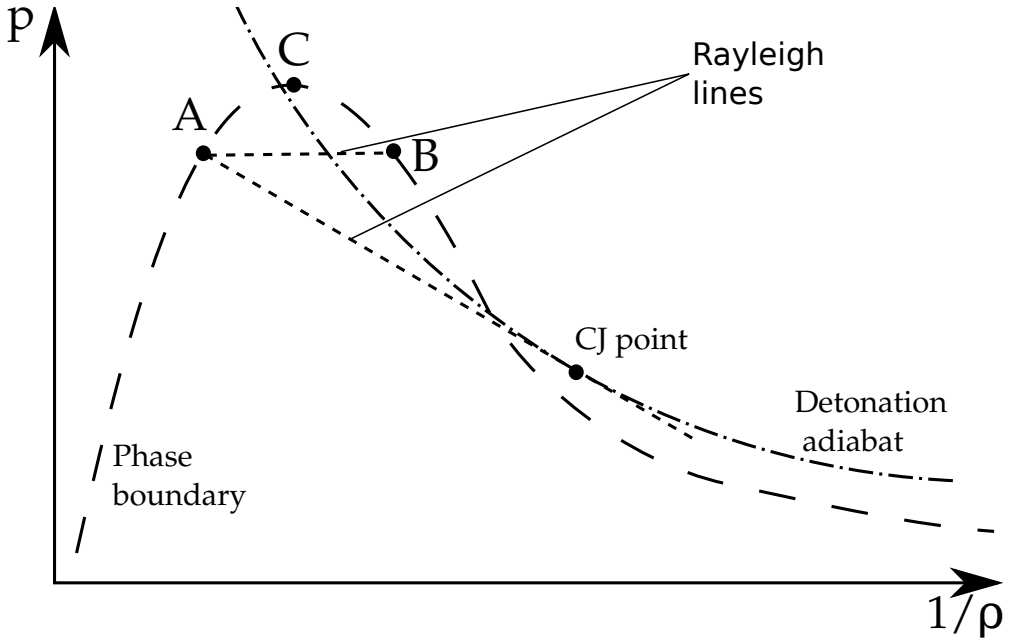


Figure 6.12: Visualization of the Chapman-Jouguet condition for maximum mass transfer. Point A is the starting point, point B visualizes the Maxwell line and the CJ point is the point of maximum mass transfer. C denotes the critical point.

by source terms in the liquid and vapor phase respectively. This method is similar to the mass transfer modeling usually applied in incompressible multi-phase solvers, where source terms for the conservation and energy conservation equations are used. The main difference is that in incompressible solvers an EOS is not necessary for the description of the flow field. This implies that any model for the phase change can be used that physically describes mass transfer. Already within the incompressible description, phase change modeling is crucial because high mass transfer rates have a negative impact on the stability of the numerical scheme. Hence, numerical smearing algorithms (see e. g. Hardt and Wondra [36]) are used to provide sufficient stability for the phase transfer modeling approach.
In the context of the compressible multi-phase solver considered here, the capabilities of such a simple evaporation model are investigated using the stiffened gas EOS (see section 5.1.1). This model represents the fluid behavior of a (nearly) incompressible fluid approximation as considered in the incompressible model.
The centerpiece of the evaporation model is the estimation of the mass transfer rate. This process is driven by thermodynamic diffusion and is a slow process.

Schrage [82] approximated the mass transfer rate to

$$\dot{m} = \frac{\lambda_{\text{vap}}}{\sqrt{2\pi R_{\text{vap}}}} \left(\frac{p_{\text{sat}}(T_{\text{liq}})}{\sqrt{T_{\text{liq}}}} - \frac{p_{\text{vap}}}{\sqrt{T_{\text{vap}}}} \right). \tag{6.28}$$

He assumed a Maxwellian velocity distribution within both bulk phases and analytically estimated on this basis the amount of molecules that change the phase. The temperature change in the liquid phase is neglected and it is assumed that the liquid state remains at saturation conditions. In his model the mass transfer rate is dependent on the pressure and temperature in the vapor and liquid phase as well as the saturation pressure for the liquid phase $p_{\text{sat}}(T_{\text{liq}})$. As shown by Tanasawa in [90], the mass transfer term can be reformulated using the Clausius-Clapeyron equation. An additional evaporation parameter λ_{vap} is introduced to fit the model to experimentally measured mass transfer rates. For water, the evaporation parameter λ_{vap} is between 0 and 1. In the model it is assumed that the temperature of the liquid phase is constant.
Hardt and Wondra [36] showed the application of such an evaporation model in the incompressible context. They formulated the effects of mass transfer as source term in the vicinity of the interface and numerically smeared the mass transfer source term to increase stability. Lauer et al. [52] extended the evaporation model to compressible flow. Based on equation (6.28), the velocity of the evaporation wave v_{PT} is approximated to

$$v_{\text{PT}} = \frac{\dot{m}}{\rho_{\text{liq}}} \tag{6.29}$$

using the density on the liquid side ρ_{liq} and the estimated mass transfer rate $\dot{m}$. The resulting source term uses directly the primitive variables as determined by the Riemann solver without phase transition (see section 6.3) and can be written as follows.
In case of evaporation ($\dot{m} \geq 0$)

$$S_{\text{evap}} = \begin{pmatrix} \dot{m} \\ \dot{m}\mathbf{v}_{\text{liq}} \cdot \mathbf{n} \\ \dot{m}\left(\epsilon_{\text{liq}} - 1/2|\mathbf{v}_{\text{liq}}|^2\right) + p_I v_{\text{PT}} \end{pmatrix} \tag{6.30}$$

using the velocity v_{liq} in the liquid phase, the inner energy u_{vap} of the vapor phase and the pressure at the interface p_I. Analogously the source term can be

formulated for the case of condensation ($\dot{m} \leq 0$)

$$S_{\text{cond}} = \begin{pmatrix} \dot{m} \\ \dot{m}\mathbf{v}_{\text{vap}} \cdot \mathbf{n} \\ \dot{m}\left(\epsilon_{\text{vap}} - 1/2|\mathbf{v}_{\text{vap}}|^2\right) + p_I v_{\text{PT}} \end{pmatrix}. \tag{6.31}$$

In this case the velocity of the vapor phase is used for the calculation of the source term.
The estimated source term is applied at the interface position and is then integrated in the non-conservative fluxes at the phase interface. Thereby, it is easily possible to include this evaporation model in the context of the heterogeneous multi-scale method as described in section 2.2.
The model of Schrage introduced here is only one choice for an analytical evaporation model. Several equilibrium and non-equilibrium models are compared by Miller et al. in [63].

6.4.4 Relaxation approach according to Saurel

Another approach to handle phase transition is proposed by Saurel et al. in [81]. They applied a Baer-Nunziato mixture model together with a stiff thermochemical solver at the phase interface that handles phase transfer effects. Due to the usage of the Baer-Nunziato model, a diffuse interface is obtained. The fluid properties in the multi-phase region are handled based on a mixture approach using the EOS of the vapor and liquid phase. Therefore, an additional PDE is solved that describes the vapor content of the mixture. Separate mass conservation equations with source terms due to the phase transfer modeling are solved in the model. A mixture approach is considered for the approximation of the final state that is based on both bulk phases.
Another open question is, whether the fluid in the multi-phase region can be described by a mixture approach. Because of that, this approach is only mentioned here for completeness and the results are taken from the publication of Saurel et al. [81] and Zein et al. [108]. The latter published an different version of the phase transfer modeling approach.
A stiffened gas equation of state is used to describe the fluid behavior. This EOS has been regressed within a certain temperature range to represent realistic fluid behavior, for details see LeMétayer et al. [55]. The aim of the regression is to fit the sound speed of the simple linear EOS to the sound speed of the realistic fluid. Parameters for water and dodecane that are used in the test

cases are given in [81]. Offset parameters for the equation of states in both phases are introduced to relate the caloric EOS behavior to a fixed reference point. This is needed for the blending in the phase transition approach as the relaxation approach combines both phases.
An different mass transfer model is introduced by Zein et al. [108]. They used the numerical approach of Saurel combined with a new kinetic relation for the phase transfer modeling and obtained realistic evaporation rates. This new phase transfer model resulted in improved mass transfer rates for the measured dodecane shock-tube test cases that are experimentally investigated by Simoes-Moreira and Shepherd in [86], compared to the phase transfer modeling approaches presented by Saurel and Zein.

6.4.5 Diffuse interface modeling using high-accuracy EOS

The main idea behind this approach is that the EOS is able to describe phase transfer effects and, hence, no explicit evaporation model is needed. Evaporation effects are included through the special EOS description in multi-phase region based on the thermodynamic equilibrium assumption. The multi-phase region is bounded by the liquid and vapor saturation lines. The challenge is to develop a consistent EOS model in this region as this region is essential for the description of mass transfer. There, the EOS has physically only one dependent variable (pressure p or temperature T) that describes the fluid behavior. In this region, two coexisting phases (vapor and liquid) are present that reduce the hydrodynamic degrees of freedom. This fact complicates the estimation of fluid states in this region as for usage in the fluid solver a unique mixture state is needed.
Another problem is related to the construction of the EOS. To be able to fit the values on the saturation curves, the EOS description oscillates in this region that does not reflect the physical behavior of the fluid. Within the multi-phase region the spinodal region can be found, for which no approximation of the sound speed is possible.
The way how the EOS is described is outlined in section 5.1.4. Typically, the description includes several correlation parameters for the residual Helmholtz functional that are different for polar and non-polar fluids. The focus here is the approximation in the multi-phase region. The description in the bulk phases remains identical.

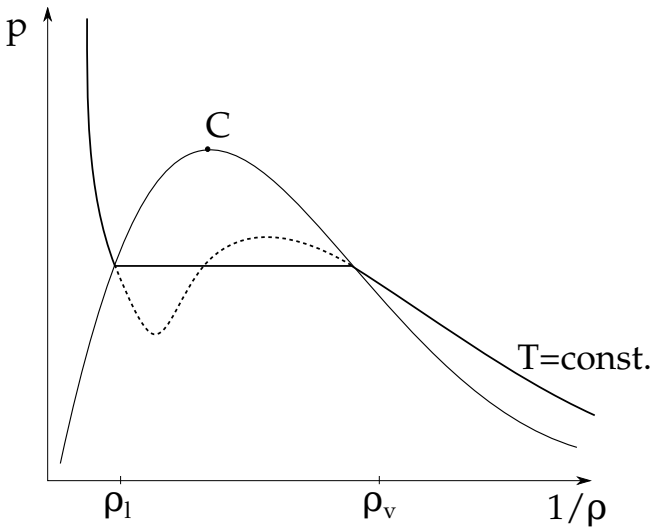


Figure 6.13: Approach to construct a suitable EOS approximation inside the multi-phase region. The EOS description predicts an oscillating pressure dependence (dashed line) that is corrected to the straight line inside the multi-phase region (Maxwell line construction).

One suitable way to approximate the fluid behavior in the multi-phase region is to reconstruct the physical states using the idea of the Maxwell line construction (see figure 6.13 for a visualization). In this reconstruction, the physically described states on the saturation lines are taken as reference. This approach assumes implicitly thermodynamic and hydrodynamic equilibrium at any point in the multi-phase region. The state inside the multi-phase region is described as a linear interpolation between the saturation states. First, the mass fraction of the state is calculated

$$X_v = \frac{\rho - \rho_{\text{vap}}}{\rho_{\text{liq}} - \rho_{\text{vap}}}\,, \tag{6.32}$$

using the states on the saturation curves for the liquid ρ_{liq} and the vapor phase ρ_{vap}. Based on the mass fraction it is possible to approximate the inner energy ϵ of the fluid at any point within the coexistence curves by

$$\epsilon = X_v\,\epsilon_{\text{vap}} + (1 - X_v)\epsilon_{\text{liq}}\,, \tag{6.33}$$

using again the states on the coexistence curves for the evaluation of the inner energies. This approach is used as well for the viscosity μ and the heat transfer coefficient λ. The temperature T and pressure p are set equal to the saturation conditions as they are constant within the multi-phase region. The sound speed c inside the multi-phase region is approximated by the Wood sound speed c_w

[104] to be able to account for the low sound speeds in the mixture between liquid and vapor. This sound speed is defined using again the saturation states by

$$\frac{1}{\rho c_w} = \frac{X_v}{\rho_{\text{vap}} c_{\text{vap}}} + \frac{1 - X_v}{\rho_{\text{liq}} c_{\text{liq}}}. \tag{6.34}$$

This approach assumes implicitly that the evaporation process is performed at thermodynamic equilibrium due to the assumption of a constant temperature within the multi-phase region. Due to the smearing of the interface a stable numerical approximation is needed, as the density jump may be large. The inclusion of surface tension effects in this method is crucial, as no distinct interface position is available. This can be circumvented by including the pressure force as volumetric source term (CSF approach). The main advantage of the model is that only single-phase Riemann problems are used within the computational domain. Furthermore, no special treatment of the interface is needed, not even a numerical interface tracking algorithm in case no surface tension effects are considered.
A similar method is used by the group of Pfitzner and Adams [68] for the simulation of rocket combustion chambers. In the used diffuse interface approach the valid root of the underlying Peng-Robinson EOS is chosen according to its fugacity in case more than one root is present.

6.4.6 Mathematical model of an interface Riemann solver with resolved phase transfer effects in the sharp-interface context

In the previously described approaches for the resolution of phase transfer effects, the evaporation wave in the Riemann wave fan is not resolved. Hence, it was accounted for by means of source terms at the interface or by an implicit modeling of the multi-phase region. In the special Riemann type solver introduced here, the evaporation wave in the Riemann fan is resolved. Thus states insides the spinodal region are avoided by the special construction of the solution. Numerically the wave is approximated as undercompressive wave and the mass transfer rate is iterated until a physical solution for both bulk states is obtained. In case one state is not admissible, the step size of the iterative method is decreased. For more information about the concept of undercompressive shock waves see LeFloch and Shearer [54], Chalons et al. [11], and

Merkle and Rohde [62]. Note that now the material and phase interface are not located at the same position anymore and an additional state in the Riemann wave fan $\mathbf{U}^\sharp$ has to be resolved.
The Riemann solver with kinematics and nucleation is the solution of the Riemann problem with initial data $\mathbf{U}_L$ and $\mathbf{U}_R$ that satisfies the entropy inequality, the kinetic relation and the nucleation criterion employed in the model. Note that the solution of this Riemann problem is unique through the choice of an additional constraint, the evaporation model. These constraints can be formulated in terms of a kinetic relation or mass transfer models.
In this model for the Riemann problem with phase transfer effects, all jump conditions for the classical waves (shock, contact discontinuity, rarefaction wave) are described by the (one-dimensional) Rankine Hugoniot jump conditions. These conditions are outlined here for $k = \{L, R\}$

$$\begin{aligned} \rho_k^* v_k^* - \rho_k v_k &= S_k(\rho_k^* - \rho_k)\,, \\ \rho_k^*(v_k^*)^2 + p_k^* - \rho_k(v_k)^2 - p_k &= S_k(\rho_k^* v_k^* - \rho_k v_k)\,, \\ v_k^*\left(\rho_k^* e_k^* + p_k^*\right) - v_k\left(\rho_k e_k + p_k\right) &= S_k(\rho_k^* e_k^* - \rho_k e_k)\,. \end{aligned} \tag{6.35}$$

A special treatment is necessary at the evaporation wave ($S^\sharp$), that is modeled as an undercompressive shock wave with additional source terms due to surface tension effects. There, the following jump conditions normal to the phase interface are applied

$$\begin{aligned} -S^\sharp[\![\rho]\!] + [\![\rho v_n]\!] &= 0, \\ -S^\sharp[\![\rho v_n]\!] + [\![\rho v_n^2 + p]\!] &= 2\sigma\kappa, \\ -S^\sharp[\![\rho e]\!] + [\![(\rho e + p)v_n + q]\!] &= 2\sigma\kappa S^\sharp \end{aligned} \tag{6.36}$$

The used jump operator is defined as

$$[\![f]\!] = \lim_{\epsilon \to 0} f(\mathbf{x} + \epsilon\mathbf{n}) - f(\mathbf{x} - \epsilon\mathbf{n})\,.$$

The heat flux in (6.36) is modeled by means of the latent heat of vaporization

$$[\![q]\!] = \Delta h_{\text{lat}} = h_{\text{gas}}(T_m) - h_{\text{liq}}(T_m)\,, \tag{6.37}$$

evaluated at the median temperature $T_m = 1/2(T_L + T_R)$ of the initial states. Surface tension effects are included as surface force at the evaporation wave.
The crucial part is to estimate the wave speed of the evaporation wave that is one of the unknowns in the above jump conditions. Therefore, the kinetic

relation comes into play that estimates the massflux j at the phase interface based on the chemical potential μ and the velocities in both phases. This relation controls explicitly the amount of entropy that is produced at the phase interface and is a model for the nucleation in the multi-phase region. It can be expressed by means of a model function $K(j) = c_s j$ together with a modeling constant c_s. This leads to the final kinetic relation based on the investigations of Dreyer [19] and LeFloch [53]

$$\left[\!\left[\mu + \frac{1}{2}\left(v_n - S^\sharp\right)^2\right]\!\right] = -K(j) = c_s j \tag{6.38}$$

The chemical potential μ is a good measure for the departure of the phase interface from equilibrium. In thermodynamic and hydrodynamic equilibrium, the difference in chemical potential $[\![\mu]\!]$ is equal to zero resulting in the trivial case of no mass flux at the interface. This constraint is a requirement for the estimation of the multiphase region in the EOS chart. The right hand side term $K(j)$ is an entropy production term at the interface that is modeled based on an entropy production coefficient c_s. This constant c_s is approximated using molecular theory [102] and controls the entropy production/dissipation at the interface. This kinetic relation ensures a solution that is compatible with the second law of thermodynamics. However, the approximation of this term is not trivial and the magnitude is not known for many fluids. Some estimates based on molecular simulations and the DFT are introduced in [102] for an alkane and an inert gas.
The physical background of this modeling is that due to phase transition processes entropy is produced at the interface. One method to model this is model an additional jump term for the entropy at the interface that is proportional to the mass flux rate j.
Having an estimate for the speed of the evaporation wave $S^\sharp$, the evaporation mass flux can be calculated to

$$j = \rho_L(S^\sharp - v_L) = \rho_R(S^\sharp - v_R)\,. \tag{6.39}$$

In case the maximum mass transfer rate is reached and the left and right state can not be connected using the wave fan directly, attached waves (shocks or rarefaction) at the phase boundary have to be introduced. These attached waves travel with the evaporation wave and ensure the Chapman-Jouguet condition at the phase boundary.

This mathematical model for the Riemann problem with phase transfer is valid normal to the phase interface. Due to the non-linearity of the model, especially regarding the wave speed approximation of the phase transition wave, the Riemann solver has to be applied normal to the phase interface. This direction is determined by the level-set normals whose estimation is described in section 4.2. Note that the level-set normals are a by-product of the curvature estimation.
The next subsections are dedicated to a short introduction of solution strategies for this kind of Riemann problem with phase transfer. All Riemann solvers solve the above described mathematical model for phase transfer with surface tension. The first method is an implementation of an exact Riemann solver that is, in a second step, simplified to approximate Riemann solvers. These approximate solvers are based on the HLL approximation of the Riemann problem with one or two additional waves depending on the type of abstraction. Additionally to the HLL wave fan, a phase transition wave and a contact discontinuity can be considered.

6.4.6.1 Exact micro-solver with mass transfer and surface tension

An exact implementation of a Riemann solver based on the above stated equations is described by Rohde and Zeiler in [79] for the isothermal case and its generalization to the full Euler equations in [28]. All jump conditions (as defined above) as well as the kinetic relation are solved simultaneously by a multidimensional Newton method. A special method is used to construct the evaporation wave in the wave fan that is approximated by an undercompressive shock wave. In total a non-linear iteration of 13 equations is necessary to resolve the Riemann wave fan at the phase boundary. Because of that this Riemann solver is rather slow but provides an exact solution with evaporation and condensation effects at the phase boundary.
The resulting Riemann wave fan is visualized in figure 6.14 exemplary for a configuration with evaporation. For the flux calculation on the liquid side the state $\mathbf{U}_L^*$ is chosen, on the gaseous side the state $\mathbf{U}^\sharp$. This solver is non-linear due to the exact resolution of rarefaction waves and the non-linear kinetic relation that is used to estimate the wave speed of the evaporation wave. Due to the non-linearity this solver has to be applied normal to the tracked phase interface as determined by the level-set method.
This exact solver for the Riemann problem at the phase interface provides reference solutions shown in the investigations of the one-dimensional shock tube problems (see section 6.4.7.1). The exact solver is called “micro-solver” in

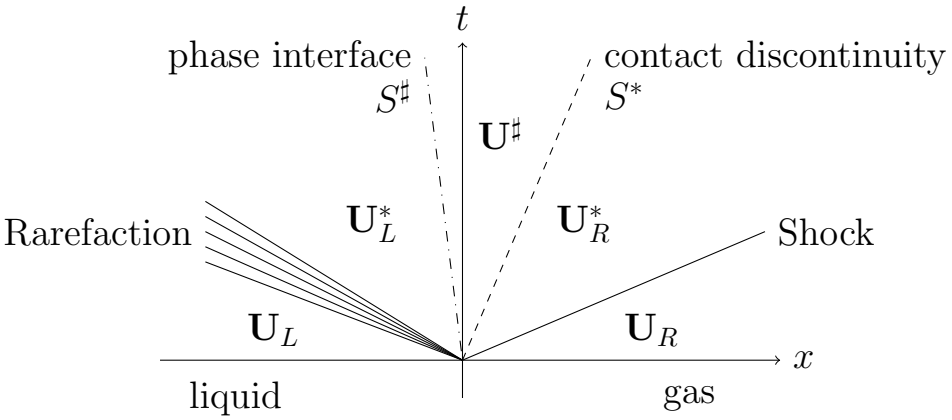


Figure 6.14: Riemann wave structure of the exact Riemann solver at the phase interface with phase transition wave and contact discontinuity. Here, an evaporation case is shown as the contact discontinuity is right of the evaporation wave.

the following investigations.

6.4.6.2 HLLCP: Approximate micro-scale solver with mass transfer and surface tension

The driving force behind the development of approximate Riemann solvers for the case with phase transfer is the massive computation time for the exact solver. The idea is to provide a simple and fast approximate solver similarly to the approximate solver for single-fluid flows as e. g. discussed in the book of Toro [96].
Here, a Harten-Lax-van Leer (HLL) Riemann solver [37] is chosen as basis for the further development as the wave structure within the outer waves can be easily extended. This was e. g. done by Toro et al. [97] to restore the contact discontinuity as additional wave to improve its resolution. In first case with phase transition considered here two additional (linear) waves have to be inserted into the HLL wave fan. This is done to resolve all waves that are present in the Riemann wave fan. The only approximation is the linearization of the rarefaction wave. The methodology for this linearization can be taken directly from investigations of the interface Riemann solvers without phase transfer that already considers the pressure jump due to surface tension forces.
The first additional wave is the contact discontinuity that is introduced similarly to the derivation of the HLLC Riemann solver. This wave now describes

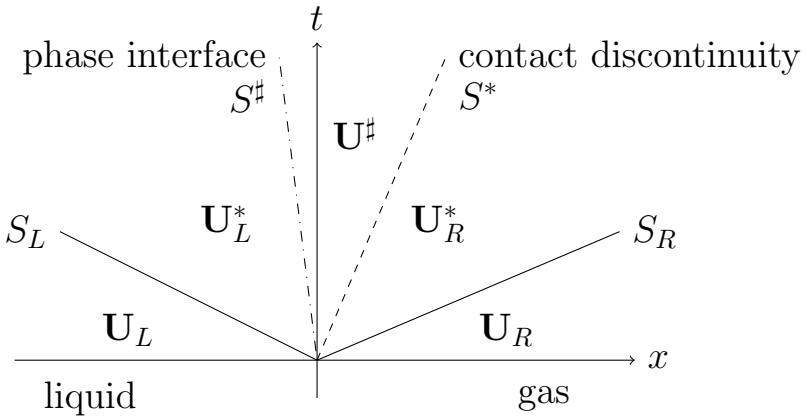


Figure 6.15: Riemann wave structure of the approximate HLLCP Riemann solver with phase transition wave and contact discontinuity. Here, an evaporation case is shown.

the location of the material boundary. An additional wave is introduced that describes the phase boundary and approximates the evaporation wave. The so called phase transition wave is approximated as undercompressive shock wave and the mass flux rate is estimated using the kinetic relation. In the following this special HLL Riemann solver is called HLLCP solver with the two additional waves: contact discontinuity (C) and evaporation wave (P for phase transition). The wave structure of the HLLCP solver is visualized in figure 6.15. The states left and right of the transition wave are used for the HLL flux approximation. Similarly to the single-phase HLL solver, the wave speed of the outer waves are approximated such that pressure jumps due to surface tension and phase transition are taken into account (see section 6.3.2). The pressure jump term due to surface tension is approximated based on an initial guess provided by the kinetic relation.
The resulting equation system that has to be solved consists of 13 equations in total. Rankine-Hugoniot jump conditions (6.35) are applied at the two outer waves (S_L and S_R) as well as at the contact discontinuity (S^*). The evaporation wave ($S^\sharp$) is modeled as undercompressive shock wave and the resulting jump conditions are described in (6.36). The phase velocity of the evaporation wave is determined by a kinetic relation (6.38) that controls the entropy produced at the interface. The resulting non-linear equation system was solved using Maple for a variable speed of the transition wave (i. e. variable mass flux rate) and a variable pressure $p^\sharp$. This implies that a two-dimensional Newton iteration is necessary in this case.

The solution strategy is the following:

1. Based on an initial evaluation of the kinetic relation and the interface curvature, the pressure jump and the minimal/maximal wave speeds for the HLL solver are approximated. This approximation is based on the description in section 6.3.2.3.

2. An iterative method is used to solve the resulting equation system. For the iteration, the equation system was reduced to an iteration of the sharp-state pressure $p^\sharp$ and the velocity of the evaporation wave $S^\sharp$.

3. In a last step the states on each side of the evaporation wave are inserted into the HLL flux approximation. These fluxes are then used for the flux balance in the code.

6.4.6.3 HLLP: More simplified approximate Riemann solver with surface tension and mass transfer

An even more simplified approach is to consider only one additional wave in the HLL wave fan. The choice is here to resolve the phase interface as the jump conditions at the interface have to be resolved within the sharp-interface approximation. The (typically small) jumps in density and inner energy at the contact are for most cases negligible. This approximation is based on the HLL modeling strategy for single-phase flows by Harten et al. [37]. The overall solution strategy is similar to the HLLCP solver described in section 6.4.6.2. In this case only a reduced equation system of 10 equations has to be solved that allows for a more efficient and more robust approximation of the underlying wave structure. The Riemann wave structure of the HLLP solver [29] is shown in figure 6.16. The approach for the wave speed estimate as described for the HLLCP solver is identical to the one used in the HLLP context as only the wave speeds of the S_L and S_R wave have to be approximated.
The main simplification from the HLLCP to the HLLP solver is that the contact discontinuity (material interface in the model of the Riemann problem at the interface) is not resolved any more. This imposes the difficulty that an overdetermined equation system is obtained as 9 Rankine-Hugoniot jump equations are available for only 8 unknowns. To be able to resolve the wave fan, the following approximation is introduced: The energy coupling in the gaseous phase is dropped. This simplification allows for an efficient estimation as solely an iteration of the kinetic relation is necessary. For cases with low mass transfer rates the Newton iteration converges typically within 6 iterations.

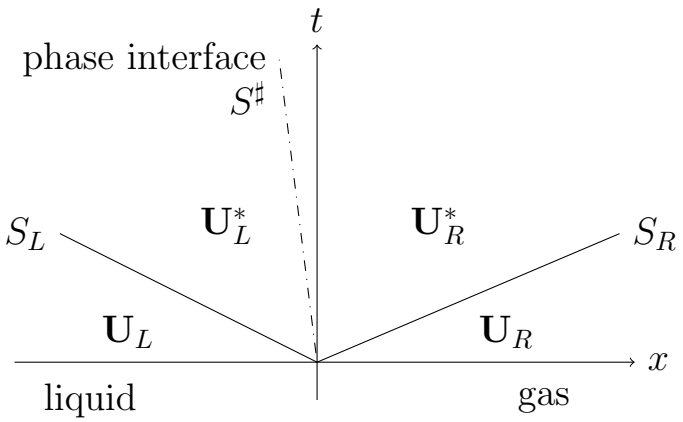


Figure 6.16: Riemann wave structure of the approximate HLLP Riemann solver with phase transition wave.

The reason for the neglect of the energy coupling on the gas side is the following: Due to here considered evaporation effects, the internal energy of the state on the gaseous side of the Riemann problem is dominated by the evaporation. For a condensation process the opposite has to be employed. The validity of this approximation is investigated in section 6.4.7.2 in comparison to the exact solution provided by the exact micro-solver described in section 6.4.6.1.

The choice of the phase transfer estimation method is arbitrary in the context of the approximate HLLP solver. The last remaining iteration is due to the considered implicit kinematic relation (see equation (6.38)). Different choices can be made regarding the estimation of the evaporation/condensation mass flux. Two simple choices are the following:

- Instead of the kinetic relation, the mass flux estimate according to Schrage (see equation (6.28)) can be used within the HLLP solver. The explicit dependence of the mass flux estimation from the solution avoids the needed iteration. This mass flux estimates considers thermodynamic non-equilibrium effects for the mass flux. A review of different evaporation models can be found in Miller et al. [63]. The model of Schrage is just one model that estimates the mass transfer rate.

- A different choice is to take maximum phase transfer, as defined by (6.27). In this case always the maximum mass transfer rate is triggered at the interface leading to a choked flow on the gaseous side.

6.4.7 One-dimensional validation of phase transfer effects

The influence of the mass transfer modeling is compared by means of one-dimensional test cases, also called Riemann problems. In all test cases, phase transfer effects are included but surface tension effects are not considered in the one-dimensional setting. For all test cases a constant entropy production coefficient $c_s = 1.4\,{}^{m^4}/_{g\,s}$ is applied for dodecane. This coefficient has been estimated for the fluid model-octane by Waibel [102]. Due to the similarity in the molecular structure of octane and dodecane this constant coefficient is used as a rough estimate. For improved mass transfer rates this coefficient has to be adapted to the investigated fluid. Also a temperature-dependent approximation of the coefficient might be needed. For larger c_s the interface entropy production increases and the mass flux rates decrease. For lower values of c_s the mass flux rates increase.

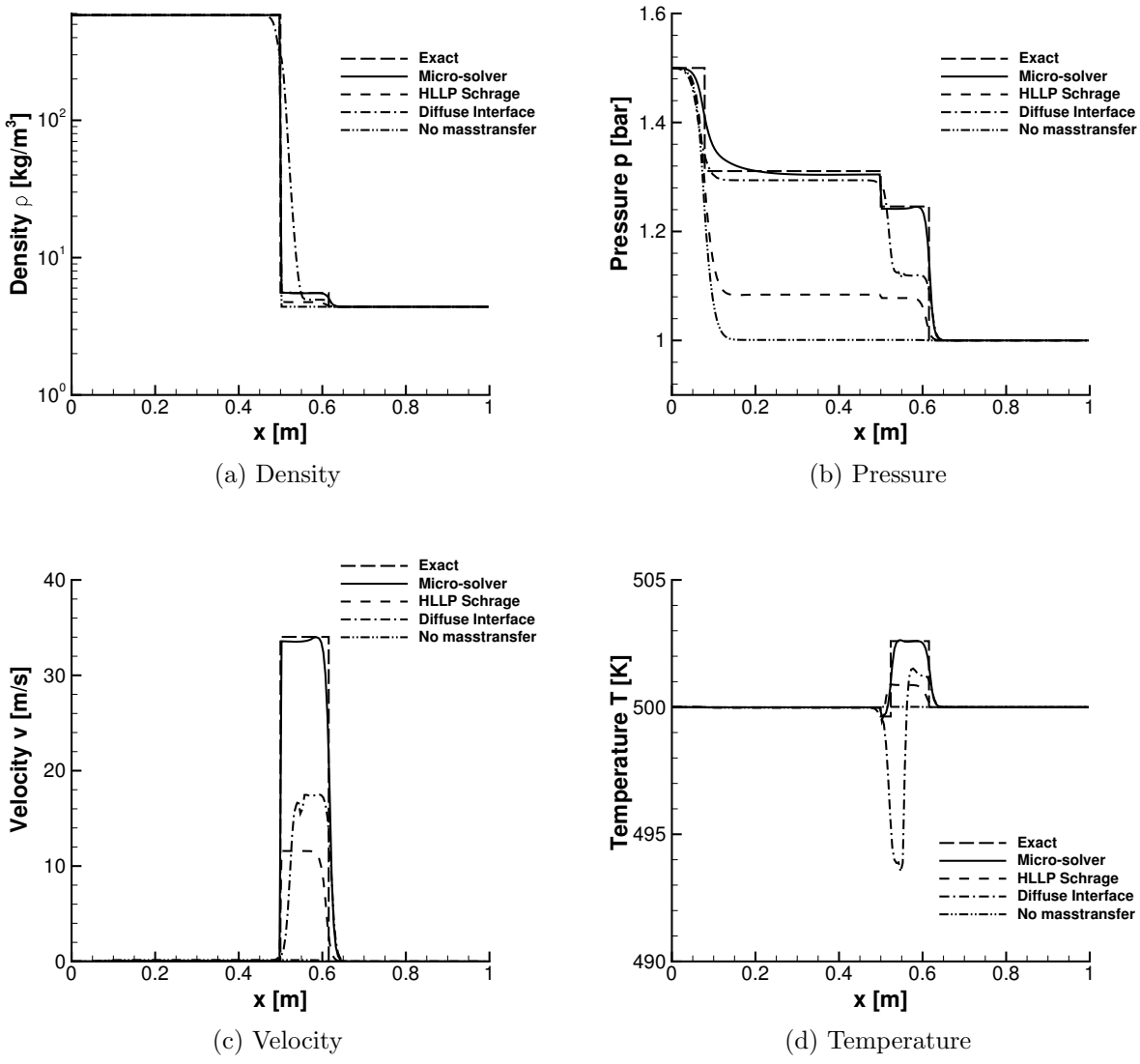
6.4.7.1 Dodecane shock tube problems

In the following, two generic Riemann problems including phase transfer effects, are compared. In this test case a dodecane fluid is considered for the description of the fluid properties, similar to the phase transfer investigations by Saurel et al. [81]. Dodecane is a retrograde fluid for which adiabatic evaporation waves can be investigated, see e. g. the experiments of Simoes-Moreira and Shepherd [86]. Initially, the phase interface is located at a position $x = 0.5$ within a unit computational domain. The liquid phase is located on the left side and the vapor phase on the right side. Two evaporation test cases with different mass flux rates at the interface are considered in the investigation. One with a medium mass transfer rate at low superheat and one with a large evaporation rate, close to the limit (CJ-point).

Medium evaporation rate This test case introduces a n-dodecane shock tube problem with a weak shock and medium evaporation rates at the interface. The initial conditions are

$$(\rho, v, p, T) = \begin{cases} (584.08\,{}^{kg}/_{m^3}, 0\,{}^{m}/_{s}, 1.5\,\text{bar}, 500\,K) & \text{if } x \leq 0.5, \\ (4.38\,{}^{kg}/_{m^3}, 0\,{}^{m}/_{s}, 1.0\,\text{bar}, 500\,K) & \text{else.} \end{cases}$$
$$p_{\text{sat}}(T = 500\,K) = 1.29\,\text{bar} \tag{6.40}$$

A pressure jump of 0.5 bar is applied to trigger the evaporation process. Initially, a constant temperature of 500 K is applied, away from the critical temperature



(a) Density

(b) Pressure

(c) Velocity

(d) Temperature

Figure 6.17: Results for the first Riemann problem test case with a medium evaporation rate at time $t = 0.7\,\mathrm{ms}$.

of dodecane with 608 K. The initial states induce an initial overheat of approximately 7 K on the liquid side and an overheat of 11 K on the gaseous side. The minor deviations of the numerical solution from the exact solution are related to the modeling of the latent heat. For the exact solution a constant latent heat is assumed while for the micro-solver a temperature dependent approximation, based on the median temperature, is used. As the temperature differences are small in this case, this effect does not have a major influence but is visible in the pressure plot due to the stiff liquid branch of the EOS.

The maximum evaporation mass flux (as defined by the Chapman-Jouguet condition) is not reached, due to the chosen low pressure ratio at the interface. With an increasing pressure ratio, the mass flux across the interface is increased until the physical mass flow limit is reached. The results are visualized in figure 6.17 for a simulation time of 0.7 ms comparing the two interface resolution methods. The long-dashed line visualizes the exact solution of the underlying interface Riemann problem with phase transfer (see section 6.4.6.1). The solid line shows the result using the solution of the exact Riemann solver at the phase boundary for the full Euler equations, representing the sharp interface approximation. Here, the micro-solver (see section 6.4.6.1) is used as interface Riemann solver with a varying median temperature. The mass transfer correlation of Schrage [82] is used as estimate for the mass transfer rate in the context of the approximative HLLP Riemann solver (see section 6.4.6.3) for the solution of the test case. For the analytical evaporation model an evaporation parameter $\lambda_{\mathrm{vap}} = 1$ is considered, the maximum mass transfer rate of the model. Hereby, the mass transfer rates of the analytical model are used as input to the HLLP solver. The dash-dot line is the approximation using the diffuse interface model (see section 6.4.5). For reference, the solution without phase transfer is shown as dash-dotdot line using the HLLC solver described in section 6.3.2.

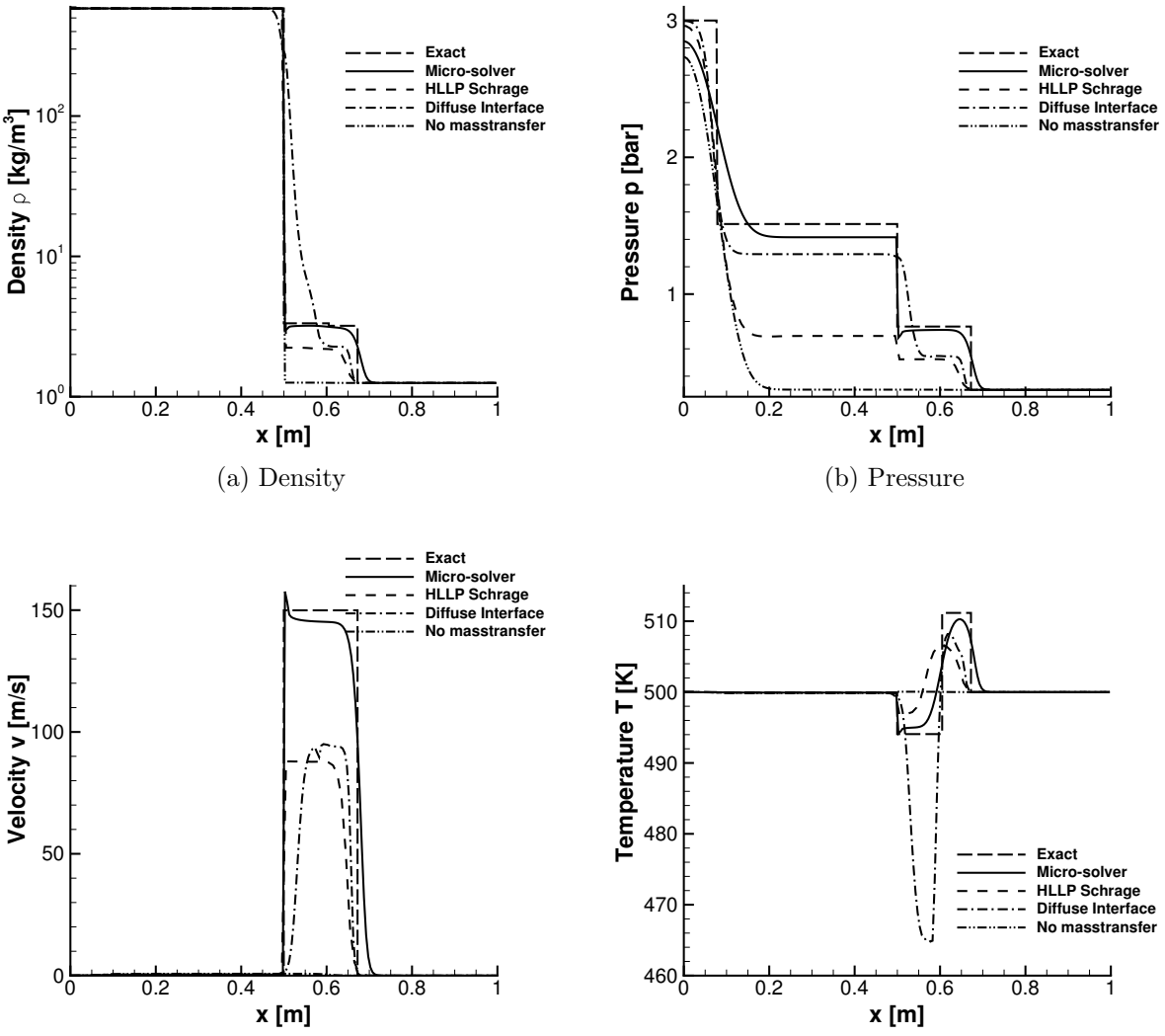
In the solution the distinct states in the Riemann solution can be distinguished; the contact discontinuity wave can hardly be identified due to the low temperature differences at the phase boundary (but the contact discontinuity is present in the solution). A fast moving rarefaction wave can be seen in the liquid part due to the initially higher pressure. At the phase boundary a pressure jump is present due to the evaporation mass flux applied there. In the gaseous phase a shock wave is traveling downstream, away from the interface. Due to the consideration of evaporation effects, the fluid speed in the gaseous part is increased with a factor of about 100. Furthermore, the fluid velocities in the gaseous and liquid phases are not equal any more. Both interface approximation approaches

reproduce the characteristics of the evaporation process as the velocities at the phase boundary are much larger compared to the case without phase transfer. Now the results for the three different approximation approaches are compared: In the context of the sharp interface with the solution of the kinetic relation, the approximation of the velocities in the gaseous phase are sufficiently larger by almost a factor of two. Due to the applied smearing at the interface, probably a lot of numerical diffusion is introduced. Another approximation applied is that the process in the multi-phase region is always at thermodynamic equilibrium which is an implicit result of the Maxwell line construction. This results in a different model for the phase change. Due to the lower velocity level also the overall pressure level after the shock is sufficiently lower. Another major difference is the position of the phase interface: In the diffuse interface approach the interface advection speed is too large as the diffuse interface is transported with the local velocity (another information about the interface speed is not available). This implies physically that the evaporation rates are not estimated correctly. The minor deviations between the exact solution and the use of the micro-solver are related to the use of a non-constant reference temperature. For the exact solution, a constant reference temperature is assumed while for the micro-solver solution the average fluid temperature of both bulk phases is used. Additionally, the results for the analytical evaporation model of Schrage [82] are plotted. The approximate HLLP solver was used with the mass transfer rates as predicted by the analytical formula (6.28). In this case the predicted mass transfer rates are sufficiently lower, compared to the estimation of the kinetic relation (approximately by a factor of 4).

High evaporation rate This test case introduces a shock tube problem with a stronger shock, now coming close to the maximum mass transfer rate as defined by the Chapman-Jouguet condition. This can be identified thereby as the velocity in the vapor phase is close to the sonic velocity of the gaseous dodecane ($c \approx 150\,{}^{m}/_{s}$). The initial conditions

$$(\rho, v, p, T) = \begin{cases} (584.56\,{}^{kg}/_{m^3}, 0\,{}^{m}/_{s}, 3.0\,\mathrm{bar}, 500\,K) & \text{if } x \leq 0.5, \\ (1.25\,{}^{kg}/_{m^3}, 0\,{}^{m}/_{s}, 0.3\,\mathrm{bar}, 500\,K) & \text{else,} \end{cases} \tag{6.41}$$

are chosen similar to the first test case, except that a larger pressure jump is considered to achieve larger evaporation rates and a much more difficult evaporation problem. The initial overheat is increased in both faces. Now an overheat of 40 K is considered on the liquid side and one of 55 K on the vapor



(a) Density

(b) Pressure

(c) Velocity

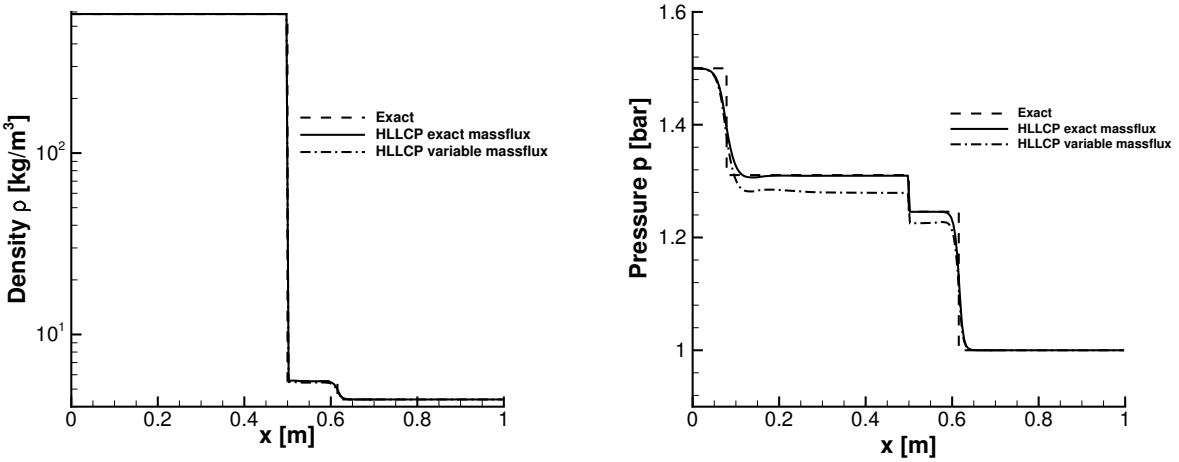
(d) Temperature

Figure 6.18: Results for the second Riemann problem test case with a high evaporation rate at time $t = 0.7\,\mathrm{ms}$.

side. The maximum mass transfer rate, as defined by the Chapman-Jouguet condition, is 530 $^{kg}/_{s}$ while the numerical approximation for the micro-solver is 501 $^{kg}/_{s}$. Hence, the mass transfer rates reached are about 5% below the maximum mass transfer rates.
Again, the comparison for the different approaches is shown in figure 6.18. The wave structure of the solution is similar to the one in the first test case. Compared to the first test case, the induced velocities at the phase interface nearly reach sonic condition. This is a clear indication that the Chapman-Jouguet point is nearly reached. An implication of the larger phase transfer rates are the increased temperature differences at the interface. The increased amount of evaporated liquid cools the surrounding gas more quickly.

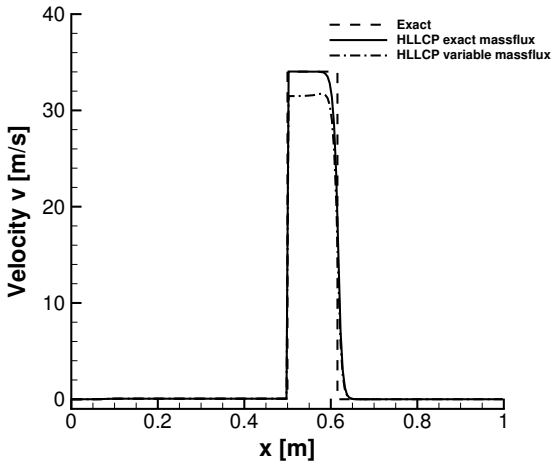
6.4.7.2 Validation of the approximate HLL-based Riemann solver

In a second step the approximative Riemann solvers, described in section 6.4.6.2 and 6.4.6.3, are validated against the exact solution as given by the exact micro-solver. This is possible as both the exact and the approximative Riemann solvers use the same kinetic model. The only approximation is due to the linearization of the outer waves in the Riemann wave fan. For validation of the approximate Riemann solvers, the test case with medium evaporation rates as described in section 6.4.7.1 is chosen for comparison. The simplifications considered here are only valid for low to medium mass transfer rates.
In figure 6.19 the results for the HLLCP Riemann solver and in figure 6.20 the ones for the HLLP solver are compared to the exact solution. Two different cases are considered: the first uses the constant prescribed mass flow rate that was determined using the exact solver. In this case the solution of the approximate Riemann solver converges to the exact solution. This is the case if no kinetic relation is considered. In the second comparison the full solver with kinetic relation is compared to the exact solution. Now the approximate Riemann solvers converge to a different mass flux rate and, hence, the solution differs slightly. The error in the mass flux estimation is about 10% which is negligible due to the large uncertainties in the estimation of the mass flow rate. No large differences between the HLLCP and HLLP Riemann solver could be found.

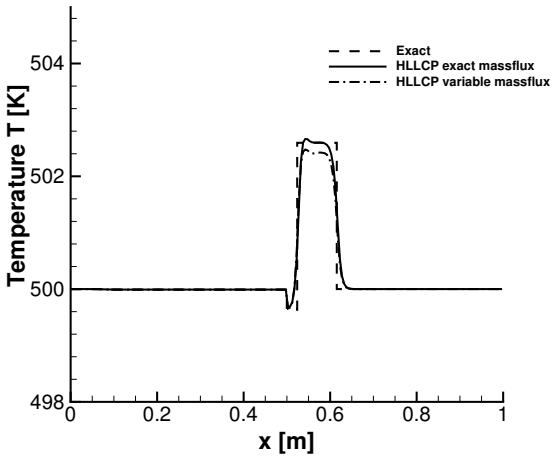


(a) Density

(b) Pressure

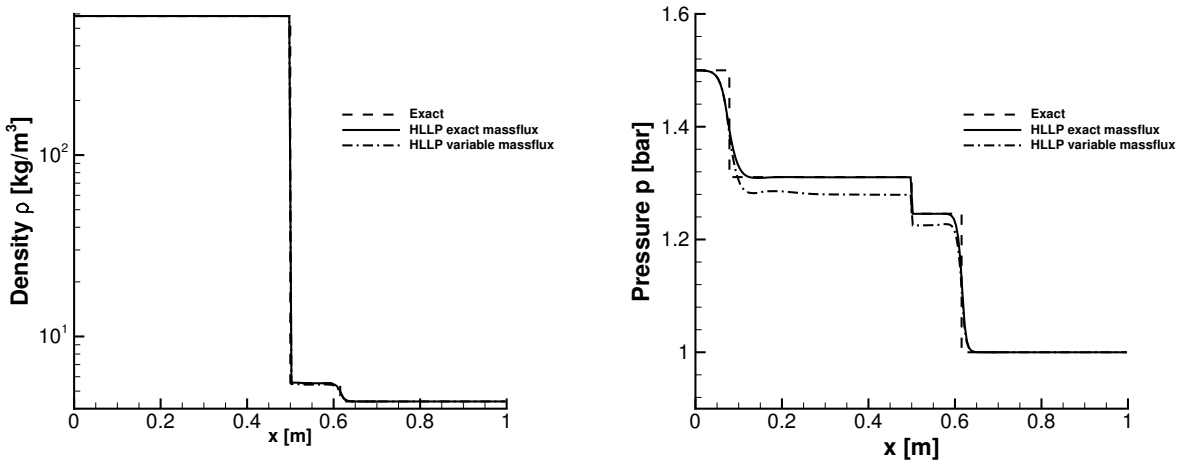


(c) Velocity



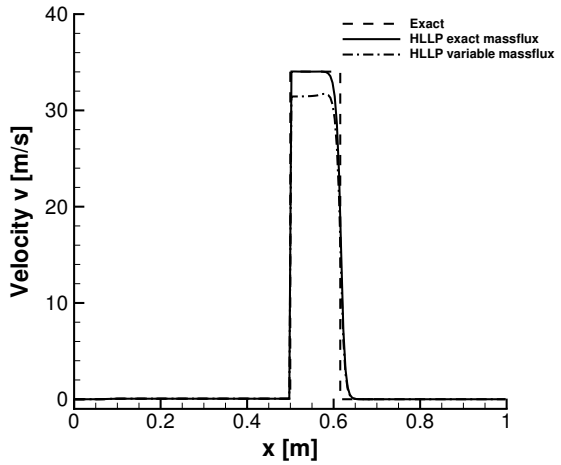
(d) Temperature

Figure 6.19: Comparison of the approximate HLLCP Riemann solver using the first Riemann problem test case at $t = 0.7\,\mathrm{ms}$.

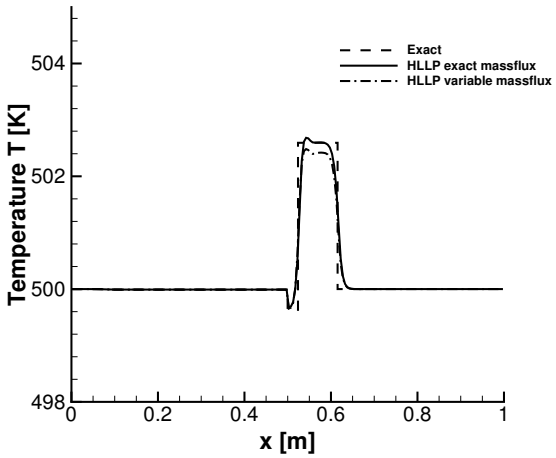


(a) Density

(b) Pressure



(c) Velocity



(d) Temperature

Figure 6.20: Comparison of the approximate HLLP Riemann solver using the first Riemann problem test case at $t = 0.7\,\mathrm{ms}$.

6.4.7.3 Comparison to Baer-Nunziato evaporation model

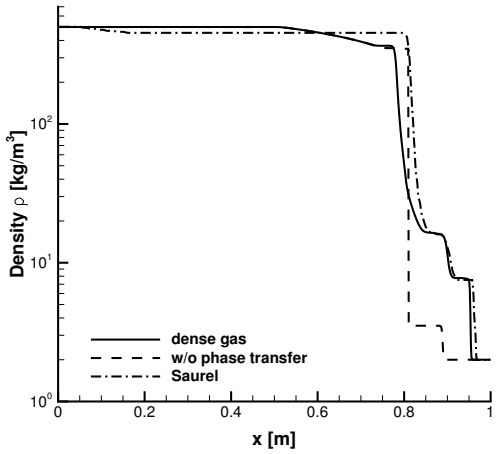
The present interface Riemann solver is now compared to a published literature shock-tube problem obtained with the evaporation model of Saurel et al. [81] and its improvement by Zein et al. [108]. For comparability to the reference solutions, a special stiffened gas EOS is used with fitted constants. The fitting of the EOS constants was introduced by LeMétayer et al. [55].

In this test case the left part of the shock tube is filled with liquid dodecane at a high pressure of 1000 bars and a density of $500\,{}^{kg}/_{m^3}$. The right side of the tube is filled with dodecane vapor at atmospheric pressure and a density of $2\,{}^{kg}/_{m^3}$. Initially, the interface is located at $x = 0.75$ within a unit computational domain. These initial conditions are summarized in table 6.1. Surface tension effects are not considered in this test case and the stiffened gas EOS, as introduced in LeMétayer [55], is used for the simulation.

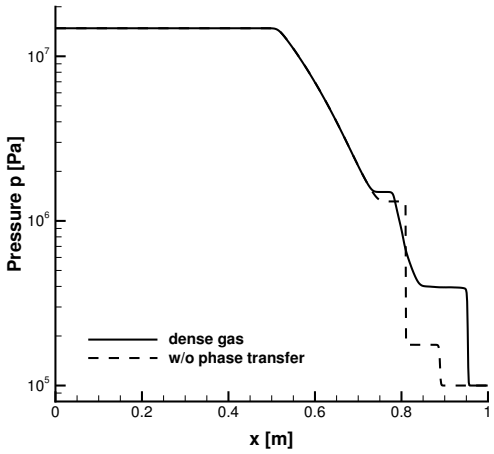
The results for this test case at time $t = 473\,\mu s$ are shown in figure 6.21 and are compared to reference results of Saurel and Zein. Despite the totally different methods to treat the compressible multi-phase flow field (mixture EOS in the Baer-Nunziato approach and a here considered sharp interface approach), the results are quite similar. The characteristics of the shock tube test with evaporation are reproduced correctly by the HLLP Riemann solver. All four waves present in the Riemann wave fan can be seen in the results. Furthermore, the influence of phase transition is clearly visible by means of the characteristic velocity peak in the gaseous part. Due to the left rarefaction wave superheated liquid dodecane is produced that evaporates quickly.

The main differences in the solution are related to the differently approximated mass transfer rates at the interface as well as the interface approximation. The model of Saurel estimates the smallest mass transfer rate, followed by the kinetic model in the HLLP solver. The model of Zein has the largest mass transfer rate that can be seen in a higher gas velocity. This evaporation rate influences mainly the states on the gas side. This is the main reason for the differences in density at the phase interface. The HLLP Riemann solver estimates an evaporation rate between both reference solutions.

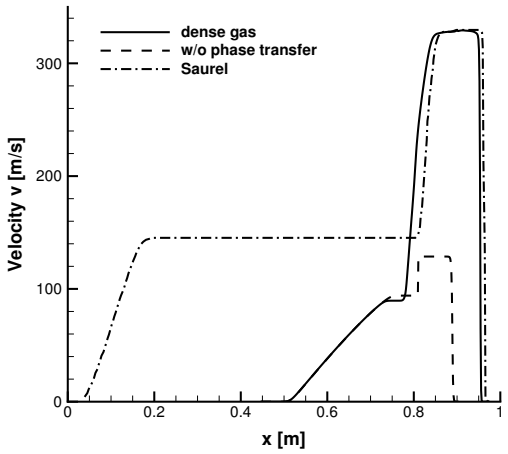
The position of the phase interface is approximately the same for all solutions. In the sharp-interface approximation considered here, the interface is approximated by a discontinuity that can be clearly seen in the plots. In the Baer-Nunziato based models the velocity and density jump at the interface is smeared out over several grid cells.



(a) Density



(b) Pressure



(c) Velocity

Figure 6.21: Comparison of the sharp interface approach to the solution of the evaporation shock-tube problem calculated with the Baer-Nunziato mixture approach. Shown is the comparison with evaporation effects at a time $t = 473\,\mu s$.

Initial conditions:

		left (liquid)	right (gas)
ρ	$[kg/m^3]$	500.0	2.0
p	$[Pa]$	1.0e8	1.0e5
v	$[m/s]$	0	0

Parameters for the stiffened gas EOS:

	$p_\infty[Pa]$	c_P $[J/kg\,K]$	c_V $[J/kg\,K]$	γ	q $[J/kg]$	q' $[J/kg\,K]$
liquid	4 E08	2534	1077	2.35	-755 E03	0
vapor	0	2005	1956	1.025	-237 E03	-24 E03

Table 6.1: Initial conditions for the Saurel dodecane two-phase shock tube problem and parameters for the fitted stiffened gas EOS for dodecane with offsets for enthalpy (q) and entropy (q').

6.4.7.4 Comparison to experiments with rapid depressurization

The adiabatic occurrence of evaporation waves in rapid depressurization of superheated liquid dodecane has been investigated experimentally by Simoes-Moreira and Shepherd [86]. Similar studies for other fluids were conducted by Hill and Sturtevant [40] (only refrigerants) and Reinke and Yadigaroglu [77] (including water and propane).

In all experiments a superheated liquid is rapidly depressurized by expanding the liquid into (nearly) vacuum. For some configurations stable evaporation waves could be observed and the phase velocities were determined experimentally in the studies cited above. In the following, the focus will be the comparison to the experiments of Simoes-Moreira [86] for liquid dodecane. These experiments have already been used by Saurel et al. [81] and Zein et al. [108] for the validation of their evaporation model based on the Baer-Nunziato model. In both studies a numerical mixture model based on the Baer-Nunziato model with a stiffened gas EOS is used to describe the compressible two-phase flow. In this study we use the high-accuracy EOS description to approximate the EOS behavior of dodecane while in the studies of Zein and Saurel a simple stiffened gas EOS was used.

For this study, the experimental configuration is reproduced starting from saturated liquid state for the reported temperature. A low density and pressure is chosen on the right side to be able to reproduce the experimental configuration of an expansion into vacuum. The pressure on the gaseous side of the evaporation wave is fitted to the experimental measurement of p_e as provided in the experimental results. This approach has already been used by Zein to fit the numerical results to the experimental ones. However, Zein uses an increased initial pressure in the liquid phase that is not used here. The front velocities are computed as wave velocities of the evaporation front in the Riemann wave fan. In the model of the kinetic relation the entropy production coefficient c_s is estimated using DFT based on Waibel [102]. The coefficient is evaluated at the initial temperature of the liquid phase and is within the range $c_s = 7.2\,{}^{m^4}/_{kg\,s}$ at 573 K to $c_s = 77\,{}^{m^4}/_{kg\,s}$ at 453 K.
The results are shown in figure 6.22 for the temperature range investigated experimentally. The initially saturated liquid dodecane is put into a metastable state due to the sudden depressurization to a lower pressure. This triggers the onset of an evaporation wave in the tube. The front velocities are a bit too large compared to the experimental ones but do recover the increasing front velocities for increasing temperatures. The results shown here were computed using the HLLP Riemann solver combined with the kinetic relation as evaporation model (see chapter 6.4.6.3) at the phase interface. In the numerical approach the front velocity is estimated as wave speed of the phase transition wave

$$s_{\mathrm{PB}} = \frac{[\![\rho v_n]\!]}{[\![\rho]\!]} \tag{6.42}$$

The differences between the experimental and numerical front velocities are related to the fact that the monotone kinetic relation used in the exact micro-solver always connects two stable states at the phase boundary. Thus, in the wave fan a two-shock approximation can be seen, that does not coincide with the experimental configuration (in which the initial metastable state is directly connected to the gaseous state). In the experiments and the other numerical investigation the metastable states are connected directly without considering a stable state in-between. The results could be improved by using an improved approximation of the entropy dissipation rate c_s at the interface as well as a more complete approximation of the experimental conditions.
For low liquid superheat the velocity of the evaporation wave is slightly overestimated. In the experiments for these conditions no stable evaporation front is

observed. This is related to the choice of the kinetic relation that is not suitable for the estimation of static phase boundaries under non-saturated conditions (see [1]). For larger temperatures no stable solutions could be found that is agreement with the experimental investigation.

6.4.8 Conclusion

Different methods to resolve phase transfer as well as surface tension effects were described in the last section. In the scope of the sharp interface approximation, the resolution based on a generalized Riemann problem at the phase interface was the appropriate approach without using an implicit thermodynamic equilibrium model for the phase transfer approximation. For the resolution of the generalized Riemann problem, an exact and two approximate Riemann solvers were developed. The chosen approach is suitable for general EOS, even non-analytical tabulated EOS. For the multi-dimensional validation of the HMM the focus is on the application of the HLLP solver as this solver provides an accurate solution at reasonable computational effort.
This sharp interface approximation is a suitable approach for sub-critical interface approximations. Under these conditions the interface width is rather small and the approximation within a sharp-interface approximation is applicable. This allows us to model the interface effects by appropriate jump conditions. For supercritical conditions, the interface starts to diffuse and a sharp approximation might be an unsuitable description. However, the description of the supercritical regime is much easier as a hyperbolic formulation is present in this region. By using different interface Riemann solvers depending on the initial interface bulk states the interface can be approximated either by the sharp interface approach for subcritical conditions or by a diffuse interface approach for supercritical conditions.
The comparison to literature data based on the resolution of the Baer-Nunziato mixture model, the obtained results show the similar trends. The overall solution strategy is heavily dependent on a suitable approximation of the mass transfer rate. Different models for the evaporation mass flux can be used within the framework of the generalized Riemann problem at the phase interface. The numerical solution structure of the Riemann problem is not changed by a different evaporation mass flux estimation. This is shown by the use of the mass flux estimation of Schrage within the HLLP Riemann solver. Different approx-

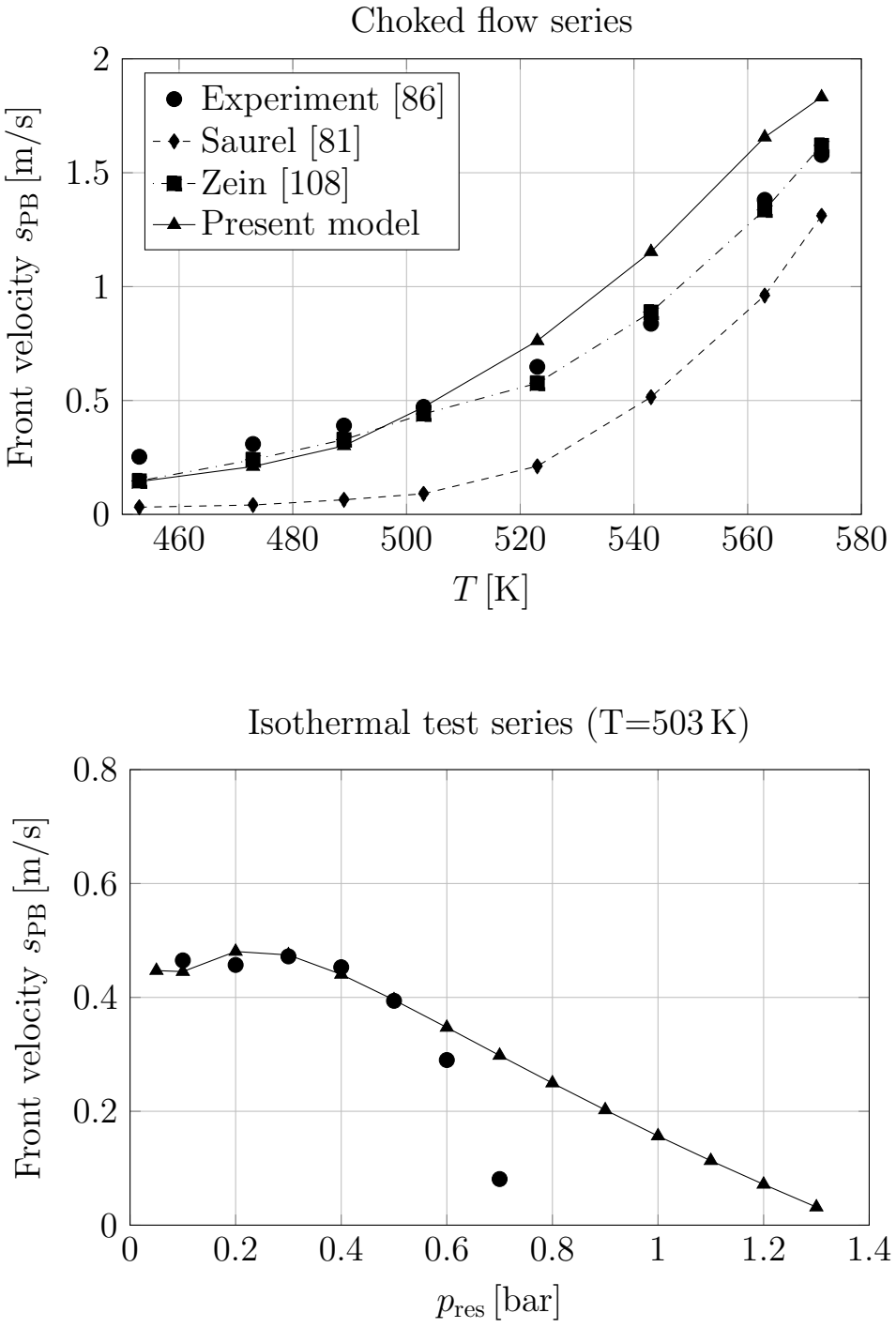


Figure 6.22: Comparison of evaporation front velocities to experimental data of Simoes-Moreira and Shepherd [86]. Shown is the comparison for the choked series (top) and the isothermal series (bottom) at T=503 K. Included is the comparison to numerical studies of Saurel et al. [81] and Zein et al. [108] based on the Baer-Nunziato model.

imation methods typically imply different mass transfer rates. The validation using the shock-tube experiments of Simoes-Moreira and the generic shock-tube test case of Saurel showed the same behavior for the inclusion of phase transfer effects.
However, one main drawback of the method is that the creation of new phases have to be modeled until the small scales can be resolved on the grid. This modeling may be based on thermodynamic conditions that promote the development of new phases. In the diffuse Baer-Nunziato approximation the creation of new phases can be modeled numerically by means of an initial low distribution of the volume fraction. Thus, it is possible to obtain numerically new phases.

7 Numerical results

In this section the application of the numerical method methods for the simulation of compressible two-phase flows to droplet test cases is presented. This shows the applicability of the building blocks described in the last sections and the close interaction between the different parts. In the first section results for two-phase flows without phase transfer effects are presented. These test cases serve as validation for the compressible two-phase flow approach, due to the simple EOS description in such cases and the relatively low computational effort. In this case simple linear Riemann solvers can be used for the solution of the interface Riemann problem at the phase boundary. These test cases are are calculated using the linearized Riemann solver (see description in section 6.3.1). The second section extends the test cases towards the description of the interface with phase transfer effects. Special methods are needed to solve the local Riemann solution with four waves. The investigated methods are described in chapter 6. The investigation for the test cases with phase transfer effects is limited to the approximate HLLP Riemann solver (see description in section 6.4.6.3) with phase transfer effects. This Riemann solver was chosen as it is by far the fastest interface Riemann solver investigated that resolves phase transfer effects reliably.

7.1 Generic EOS without phase transition

In this section test cases without phase transition are considered that are used as validation for the sharp interface approach with generalized Riemann problem at the phase boundary. In this section a simple combination of the ideal gas approximation for the gaseous phase and a Tait equation of state for the liquid phase is used. This EOS choice does not reflect a realistic fluid behavior but it is computationally cheap and accurate if the fluid is away from the critical point. The Tait EOS is chosen to reflect the stiff behavior of the liquid phase. In the following three validation test cases are chosen that validate the inclusion of the surface tension as forces in the local Riemann solution at the phase

interface as well as the ability of the resolution of a compressible flow field. Furthermore, the ability of a time-resolved curvature estimation is presented. These results have been published in Fechter and Munz [27].

7.1.1 Spherical droplet at rest

The first test is a simulation of a droplet at rest including the described surface tension approach. As described in section 6.3.1, the linearized Riemann problem solver is applied at the phase boundary taking into account surface tension effects as surface force. Therefore, the mean interface curvature at the interface position is needed. The droplet is initialized at a slightly higher pressure than the surrounding gas, which is balanced by the surface tension force. The parameters, given in detail in figure 7.1, are chosen such that the problem is steady. If the calculation of the curvature and the resulting local solution at the interface are accurate enough, the approximation should preserve the steadiness. Hence, this test case allows for an assessment of the induced parasitic currents due to the surface tension modeling.
The simulation has been conducted over a dimensionless time of $t = 0.5$ representing about 1040 iterations. The Cartesian grid consists of $40 \times 40 \times 40$ equidistant cells and the polynomial degree of $N = 3$ has been chosen to provide an accurate estimate for the interface curvature. The computational domain is a box measuring $[0.8, 0.8]^3$ in all axis directions. This resolution resolves one droplet diameter numerically with 80 points.
Figure 7.2 shows pressure information extracted on the mid-plane of the droplet at $t = 0.5$. The pressure jump associated with the surface tension is visible at the cell boundaries near the level-set zero. The constant states on the gaseous side have been perfectly conserved while minor fluctuations have been developed on the liquid side. This is related to the relatively stiff EOS for the liquid phase. Minor density fluctuations emerge as visual pressure disturbances. These fluctuations originate from the non-exact numerical curvature calculation that leads to some numerical noise. The corresponding result is shown for the case without and with the use of sub-cell refinement at the phase boundary. The resolution of the physical interface is much better using the sub-cell refinement especially for the high order DG scheme.
The amount of induced parasitic currents is measured by the maximum $\mathbb{L}_2$-norm for the momentum terms. Due to numerical and modeling inaccuracies

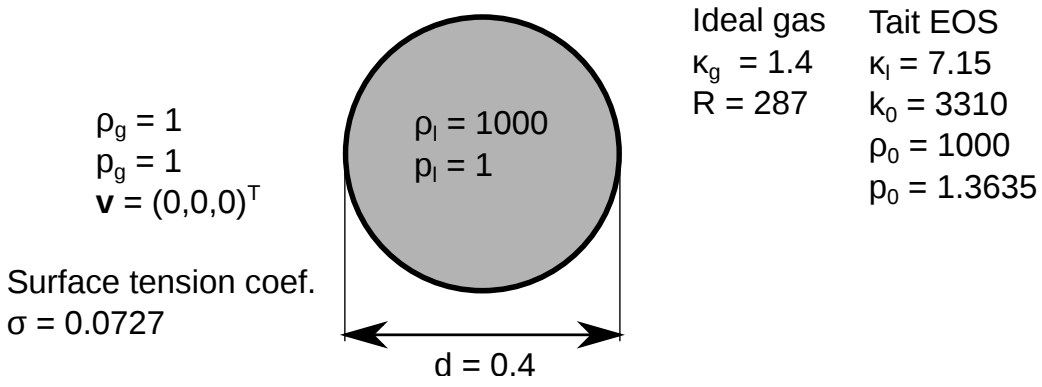


Figure 7.1: Initial conditions of the steady droplet with surface tension. Values chosen represent a water droplet in air.

DG cells	DOF per diameter	max($\mathbb{L}_2$) DG	max($\mathbb{L}_2$) DG sub-cells
10	20	7.485 E-04	6.838 E-05
20	40	1.921 E-04	2.683 E-05
40	80	7.549 E-05	1.642 E-05

Table 7.1: $\mathbb{L}_2$ error norms of the momentum as a measure of the induced parasitic currents at $t = 0.5$ with and without refinement at the interface. Here the $\mathrm{P}_N\mathrm{P}_M$ method is applied to improve the curvature calculation.

some spurious velocities or currents are visible, but they are very small and do not falsify the steady state, even over long times (see table 7.1).

7.1.2 Wobbling Droplet

The second test case is an example of a surface tension driven flow. Here, an initially ellipsoidal shaped droplet is considered that is not in the equilibrium state. Due to the surface tension force, the droplet starts to oscillate until it reaches its final spherical shape. The final state is a spherical droplet with a pressure drop fulfilling the Young-Laplace equation (2.14).
The simulation parameters are chosen similar to the first test case, only the interface geometry is changed to an ellipsoid. The initial conditions are summarized in figure 7.3. An initial ellipsoidal shaped interface geometry is assumed with the semi-axis 0.72 in x-direction and 0.42 in both other directions. A box measuring $[-1.4 \times 1.4]^3$ is used as computational domain; at the boundaries

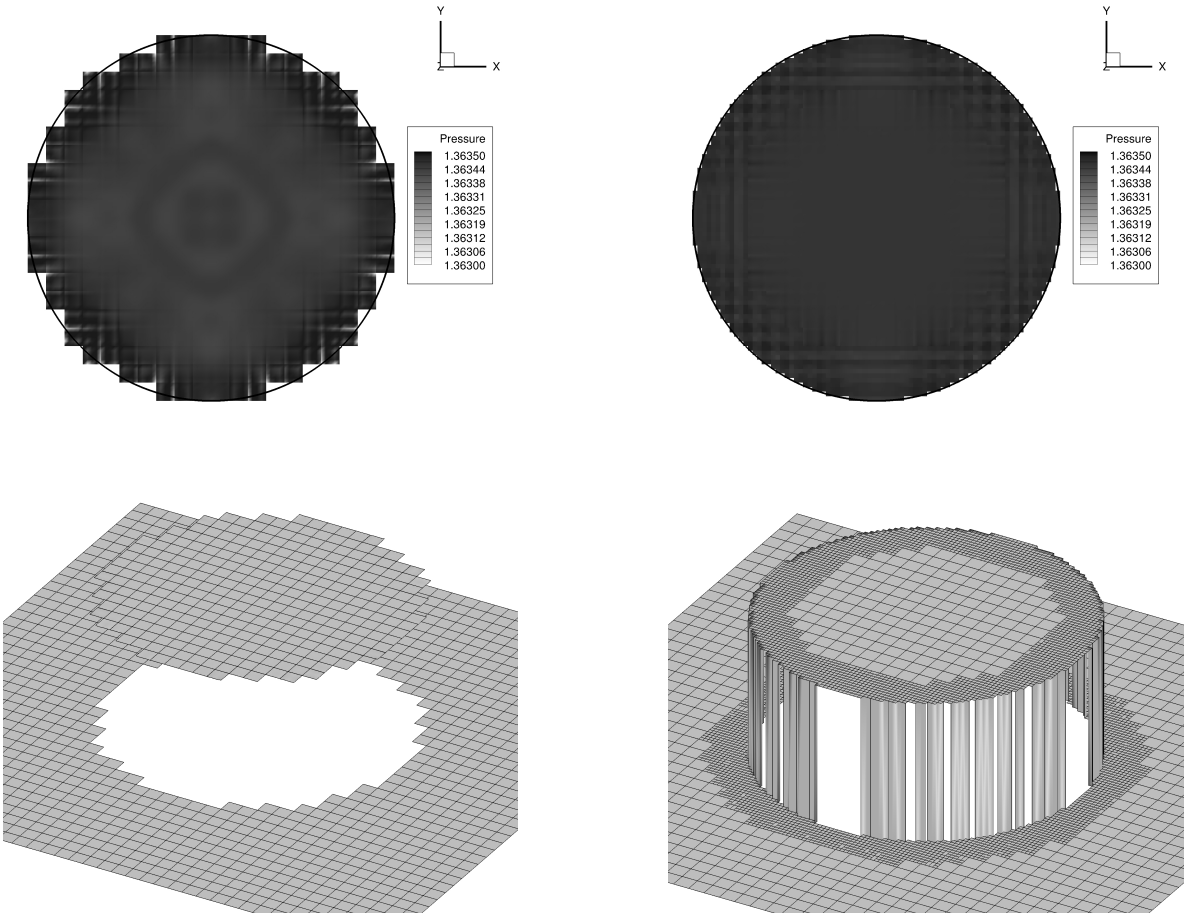


Figure 7.2: Result of the droplet at rest test case with surface tension at $t = 0.5$. The black line visualizes the physical phase boundary (zero level-set iso-contour). **Left:** Pressure contours on the droplet median plane obtained for the case without sub-cell refinement. **Right:** Pressure contours on the droplet median plane with sub-cell refinement at the phase boundary. Note that the scale in the top row does not cover the total pressure jump. The range is chosen to highlight the minimal oscillations in the liquid phase.

of the computational domain, wall boundary conditions are assumed. A mesh with $40 \times 40 \times 40$ equidistant grid cells and a polynominal degree of $N = 3$ is used.
The shape of the droplet during one oscillation period is visualized in figure 7.5. Due to numerical dissipation (physical dissipation is not present as the Euler equations are considered) the oscillation amplitude diminishes with time and finally reaches a spherical state. The main share of the numerical dissipation is introduced by the HJ-WENO scheme for the creation of the level-set advection velocity field $\mathbf{s}_{\mathrm{PB}}$. Additionally, dissipation is introduced by the reinitialization method as well as the through the curvature evaluation method.
To be able to assess the quality of the oscillatory motion of the water droplet, the oscillation frequency is compared to the analytical formula derived by Lamb [51]. He found for the resonance mode frequency f_l of the l^{th} oscillation mode

$$f_l^{\mathrm{ana}} = \sqrt{\frac{\sigma l(l-1)(l+1)}{3\pi\rho_0 V}}\,, \tag{7.1}$$

assuming a droplet with small oscillation amplitudes in vacuum or air. ρ_0 denotes the density of the droplet medium and V the initial volume of the droplet. The amplified motion of the droplet corresponds to the mode $l = 2$ (Rayleigh mode) and has an oscillation frequency of $f_2^{\mathrm{ana}} = 29\,\mathrm{s}$. The numerically obtained oscillation amplitude is plotted in figure 7.4 in terms of the maximum amplitude in x- and y-direction (the z-direction is identical to the y-direction). It can be seen that sinusoidal oscillation is reached quickly after the initial deflection. The present high-order oscillation modes (higher harmonics) diminish quickly and the Rayleigh frequency dominates the oscillation. The numerical oscillation frequency $f_2^{\mathrm{num}} \approx 29\,\mathrm{s}$ is in good agreement to the analytical estimation of Lamb.

7.1.3 Shock-droplet interaction

The third test considers a planar shock wave impinging on a spherical droplet. First, a 2D test case is considered as the published reference results consider a shock impinging onto a water column. In a second step the test case is generalized to a fully 3D shock-droplet interaction.
The initial conditions of the pre- and post-shock states as well as the initial state of the droplet are given in figure 7.6. At the domain boundaries in y- and z-direction, a wall boundary condition is assumed. The droplets initial position

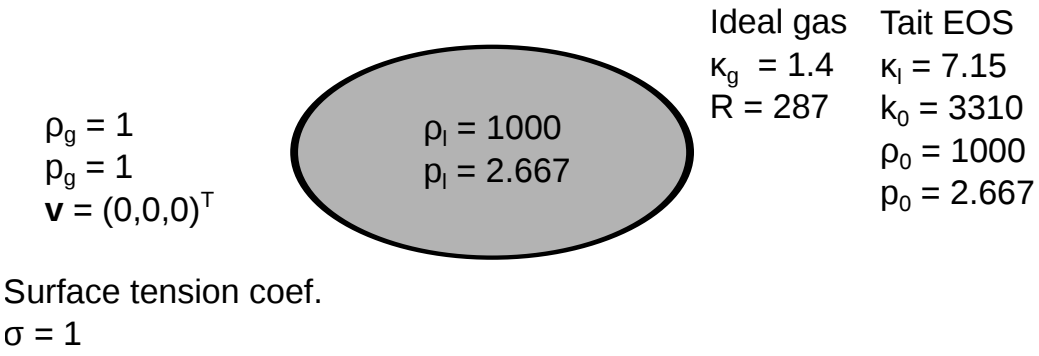


Figure 7.3: Initial conditions of the steady droplet with surface tension. The ellipsoidal semi-axes are chosen to 1.2 in x-direction and 0.7 in y- and z-direction.

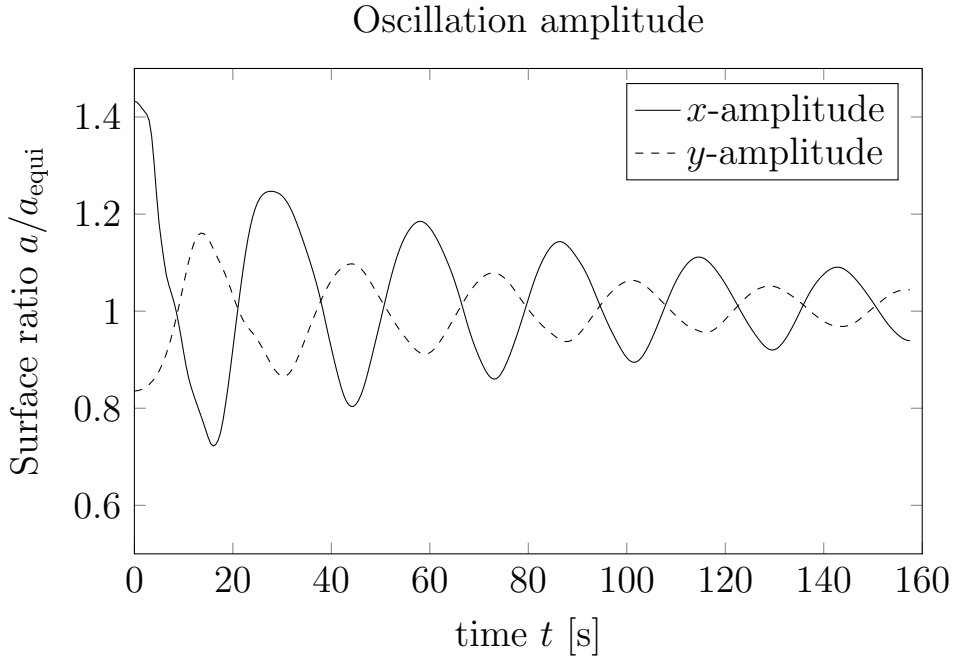


Figure 7.4: Plot of the maximum oscillation amplitude over time for an initially ellipsoidal droplet including surface tension effects. Note that the y- and z-amplitudes are identical.

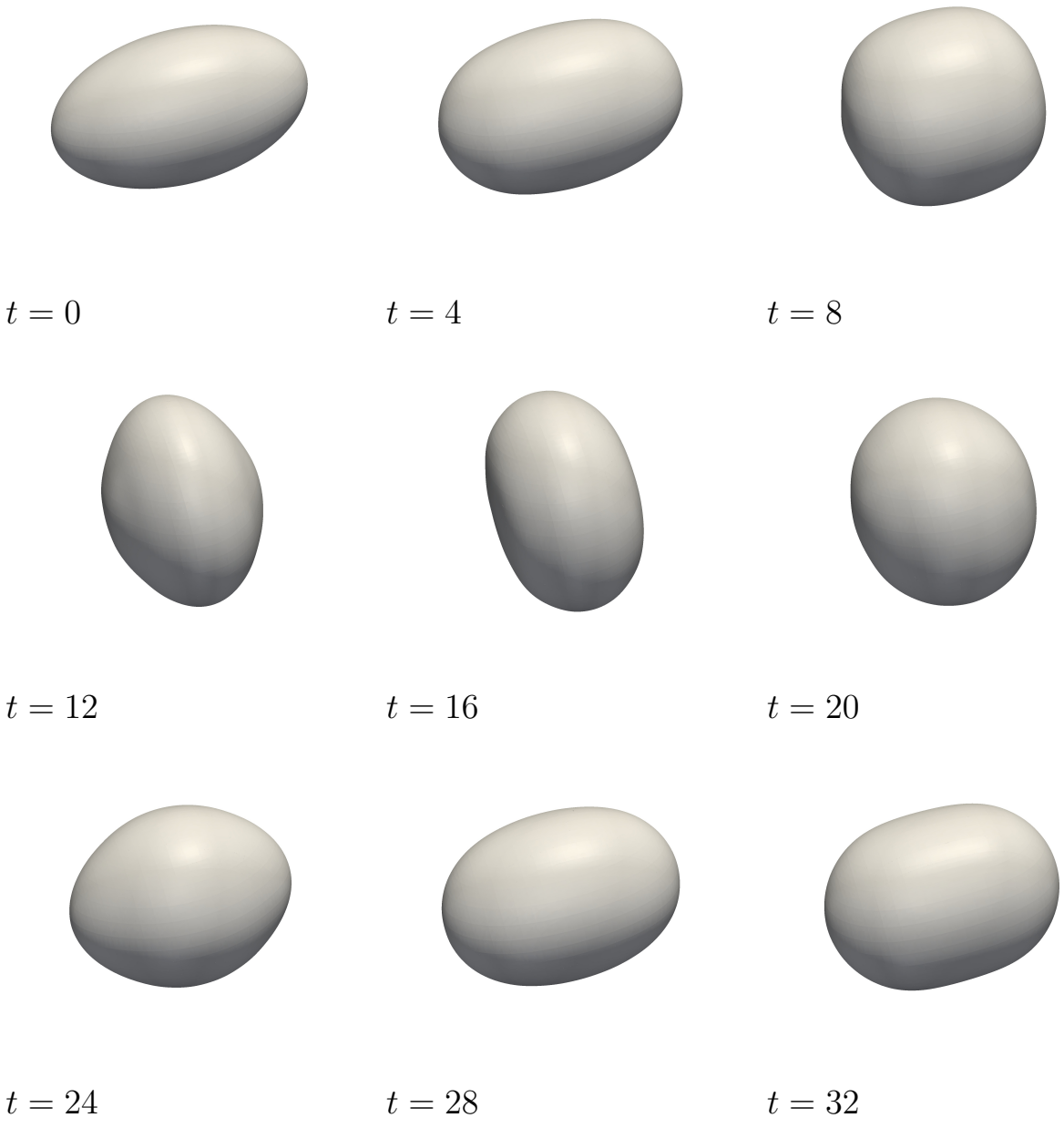


Figure 7.5: 3D surface contour (physical interface) for the oscillating droplet test case during one oscillation period. The numerically obtained oscillation frequency is $f_2^{\text{num}} \approx 29\,\text{s}$.

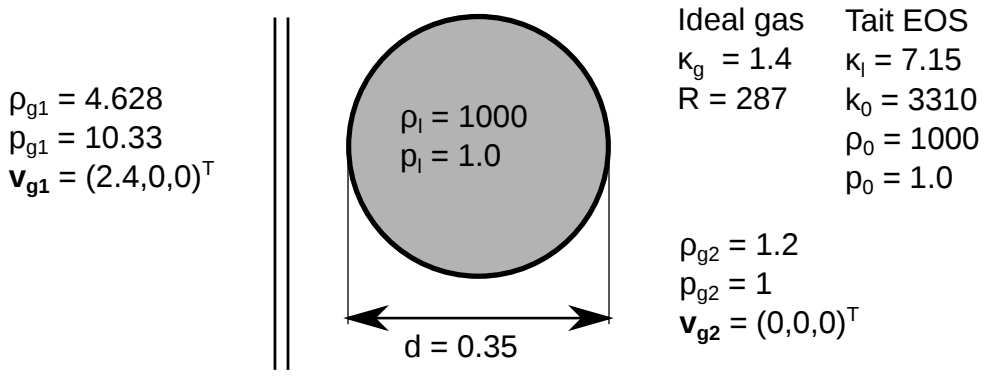


Figure 7.6: Initial conditions of the Shock Droplet interaction test case.

is $(0.55, 0, 0)$, the initial position of the planar $M = 3$ shock is set to $x = 0.35$. The non-dimensional parameters, as described by Hu et al. [44], are used in this test case.
A Persson indicator [74] based on the density is used for shock capturing purposes which reliably detects the shock position. Depending on the indicator value, the time update is calculated using the more accurate DG scheme in smooth region or the TVD stable finite-volume scheme. The latter has the ability to cope with strong discontinuities and shocks and is therefore used in non-smooth regions as found at the interface and at the shock front. A high order DG scheme would not be able to resolve the strong shock involved in this case.

7.1.3.1 2D simulation

The Cartesian mesh in the 2D case consists of 90×100 cells and the computational domain extends to $[0, 1.2]$ in x-direction and $[-0.7, 0, 7]$ in y-direction. The polynomial degree of the DG approximation is chosen to $N = 3$ and surface tension effects are not considered. The pressure jump of the impinging shock wave dominates the pressure distribution and surface tension effects have only a minor impact. The result is visualized in figure 7.7 at various time instances, showing pressure contours and a visualization of the logarithmic density gradient (Schlieren-type visualization).
The impinging planar shock wave is partly reflected at the surface of the water column and partly enters the liquid phase. Inside the liquid the shock travels at much higher speed due to the larger liquid sound velocity. At the curved surface, the planar shock wave gets reflected and distorted. With advancing simulation time, the interface geometry gets deformed due to the impacting

$M = 3$ flow. Within the liquid phase negative pressure values are encountered due to the simple EOS considered here that does not account for physical effects like cavitation that would take place at low pressures. In reality, the water would vaporize and a gas region would be created.
The main features of the simulations appear physical and resemble the higher resolved results shown by Hu et al. [44] and Chang and Liou [12]. The deformation of the water column is in accordance to an experimental shadow-graph picture, shown by Theofanous et al. [95] at the final simulation time of $24\,\mu s$ as well as a reference simulation of Hu et al. [44]. The comparison to the experiments of Theofanous is shown in figure 7.8.
Due to the staircase-like approximation of the phase boundary some disturbances are visible directly next to the interface position that diminish quickly away from the interface. The shock front recovers quickly after passing the interface. The high pressure region next to the stagnation point is clearly visible and is not distorted by the staircase pattern approximation of the computational interface.

7.1.3.2 3D simulation

The Cartesian mesh is slightly coarser compared to the 2D case and consists of $80 \times 90 \times 90$ cells. The computational domain extends to $[0, 1.2]$ in x-direction and $[-0.7, 0, 7]$ in y- and z-direction. Similarly to the boundary condition in y-direction, a wall boundary condition is assumed in z-direction.
The results, plotted on the droplet median plane, are shown in figure 7.9 using a pressure and density gradient visualization, similar to the two-dimensional simulation. The 3D results resemble the findings of the previous 2D investigation. Due to the spherical droplet geometry, the transmitted shock wave is slightly weaker behind the droplet. The impinging shock wave is reflected at the spherical interface resulting in a three-dimensional shock structure in front of and in the wake of the droplet.
Due to the reduced numerical resolution, the approximation of the droplet interface is a bit coarser and, hence, the induced numerical disturbances at the interface are larger. The overall flow features in both calculations look alike due to the symmetric flow field (the computational domain represents a shock tube as used in the experiments by Theofanous et al. [95]). This also influences the pressure distribution inside the droplet as more disturbances close to the interface are visible.

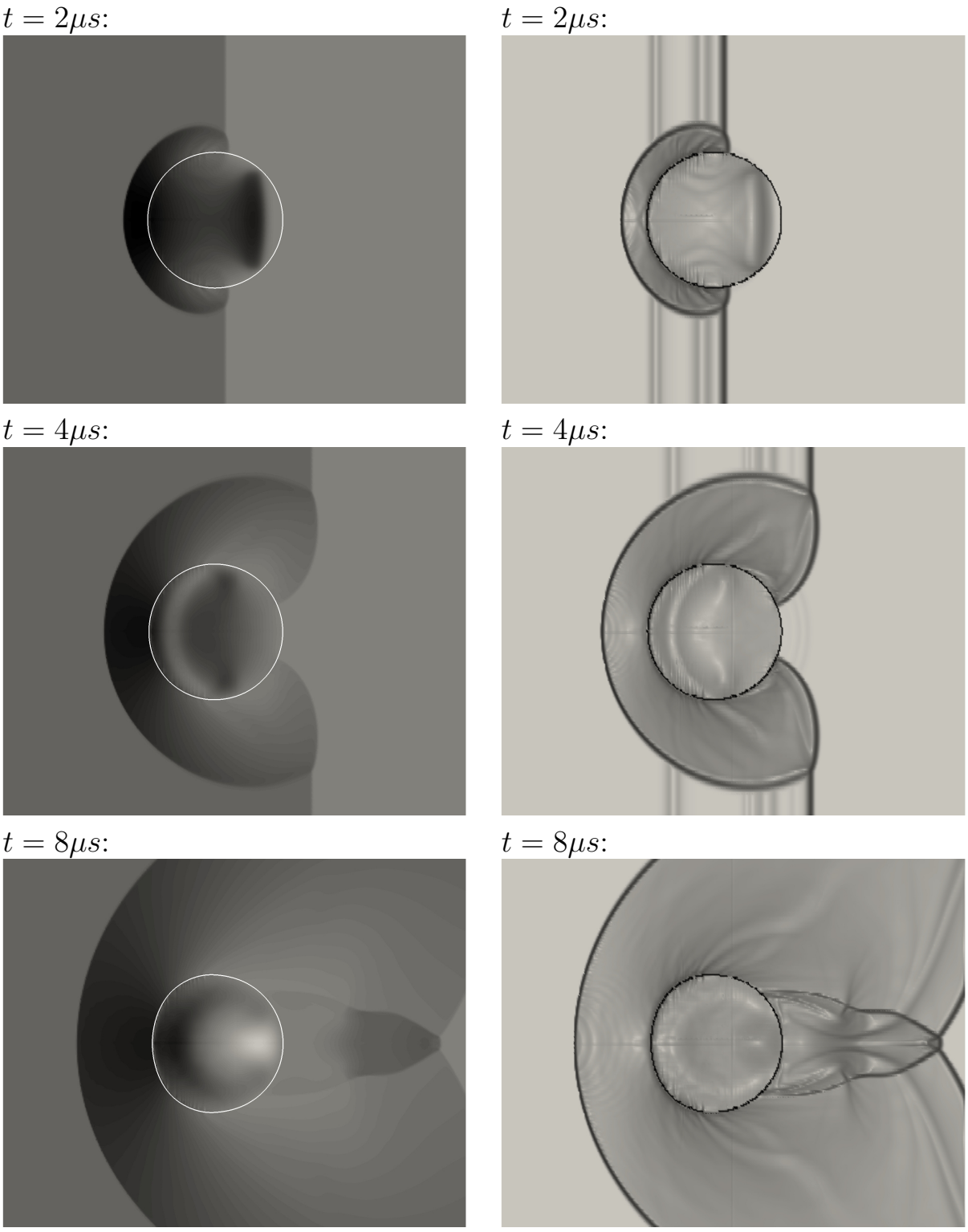


Figure 7.7: Result of a 2D water column interacting with a planar shock at various time instances. **Left**: Pressure contours in the range of -20 to 40 atm. The solid white line indicates the interface position. **Right:** Schlieren type image of the logarithmic density gradient $\log(\nabla\rho + 1)$.

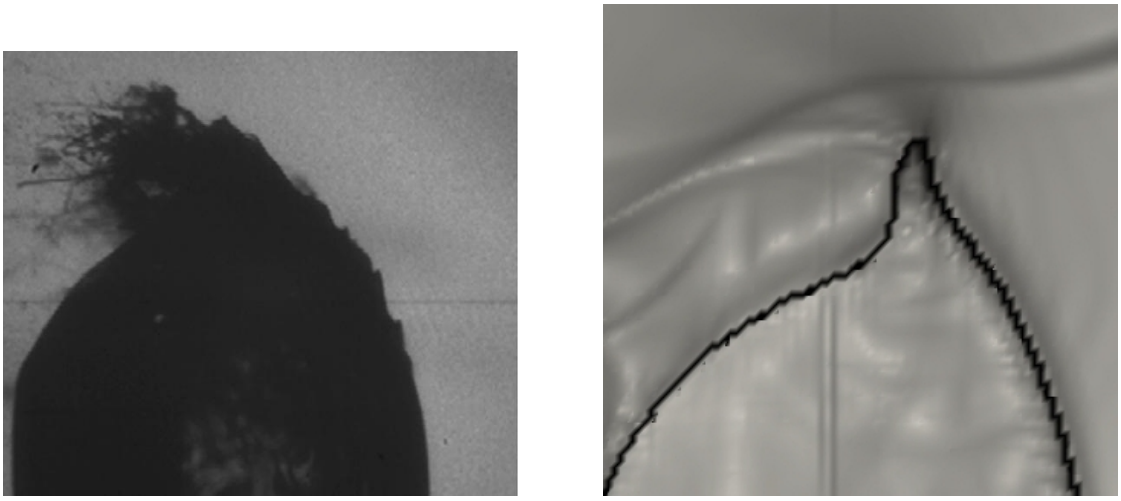


Figure 7.8: Schlieren pictures for shock-droplet interaction problem. **Left:** Tributyl Phosphate droplet subjected to $M = 3.0$ flow, $We = 2500$ (Theofanous et al. [95]). **Right:** numerical result by water droplet with $M = 3.0$ and $We = \infty$.

7.2 Realistic EOS with phase transition

In this section, realistic EOS for the description of the thermodynamic behavior are used. This usage enables the numerical representation of realistic fluid behavior at extreme ambient conditions. Due to the long evaluation times of the EOS, the EOS tabulation approach is used (for more details see section 5.3) that minimizes the EOS evaluation time.
For the three-dimensional investigation, the approximate interface Riemann solver HLLP (see section 6.4.6.3) is used. This solver is chosen due to its numerical stability and its (relatively) low computation time for the solution as the Riemann solution at the phase interface may be one bottleneck in the simulation.

7.2.1 Evaporating dodecane droplet

Based on the one-dimensional evaporation shock tube problem in section 6.4.7.1, the methodology is now applied to a three-dimensional evaporating droplet. The initial conditions are chosen equal to the shock tube problem (see equation (6.40)), now using a large droplet with an initial diameter of $d = 2\,\mathrm{mm}$. The initial conditions are summarized in figure 7.10. Due to the spherical shape the evaporation waves inside the droplet interact with each other after a short time,

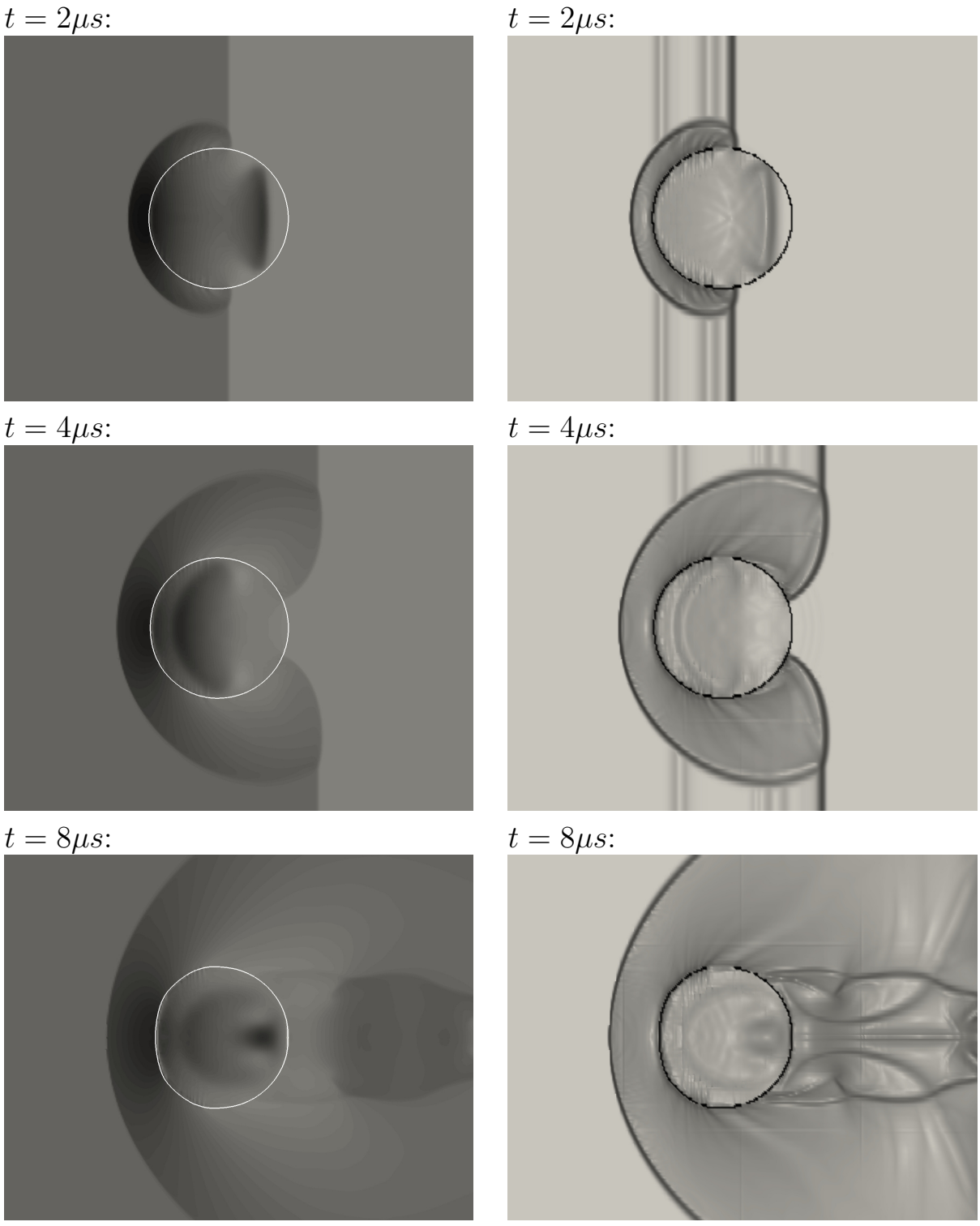


Figure 7.9: Result of a 3D water droplet interacting with a planar shock at various time instances. **Left**: Pressure contours in the range of -20 to 40 atm. The solid white line indicates the interface position. **Right:** Schlieren type image of the logarithmic density gradient $\log(\nabla\rho + 1)$.

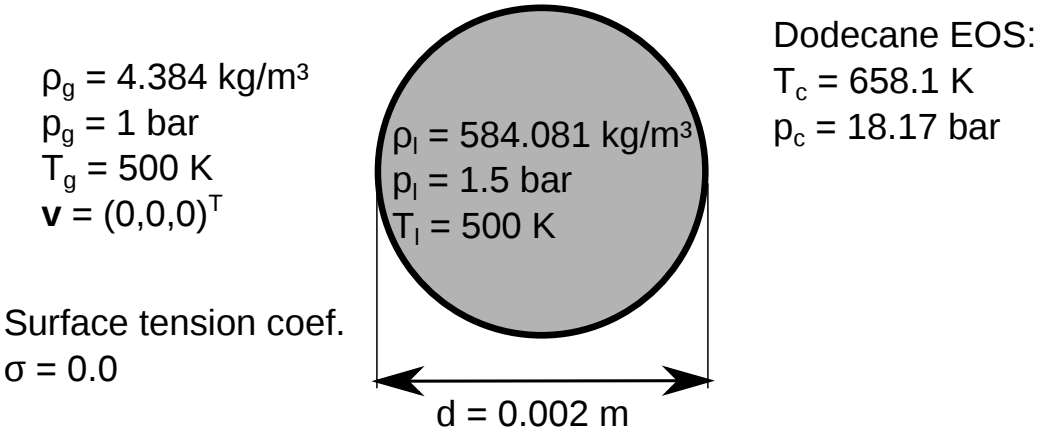


Figure 7.10: Initial conditions for the evaporation droplet test case using the fluid dodecane.

leading to a reduced pressure level inside the droplet. Similarly, a dodecane EOS is applied for the description of the fluid behavior in the gaseous and liquid phase. Surface tension effects are not considered in this test case. The inclusion would reduce the overall evaporation rates of the droplet and extend the droplet lifetime.

The droplet is put in the center of a computational domain of size $[-2.5, 2.5]^3$ mm. The used mesh resolution is 48^3 grid cells with a polynomial degree of $N = 3$ resulting in a total of 7 million DOF for the approximation of the evaporating droplet. Open boundary conditions are applied at the domain boundaries so that the evaporated mass flux may leave the domain. A constant pressure of 1 bar was set at the boundaries. Due to the large problem size and the expensive interface Riemann solver, the simulation was conducted at the Hornet system at HLRS. For this investigation the HLLP solver (as described in section 6.4.6.3) is used as a compromise between speed and accuracy.

The results for different time instances are shown in figure 7.11 visualizing the decrease of the droplet diameter due to evaporation effects. The initial droplet diameter is included as shaded geometry. Without considering evaporation effects, the droplet diameter would not decrease. At the final time of 0.9 ms more than one half of the droplet mass is evaporated.

In figure 7.12 the pressure distribution around and inside the droplet is visualized. Similar to the one-dimensional investigations presented in section 6.4.7.1, a pressure jump can be investigated at the droplet surface. This effect is related to the resolution of phase transfer effects. Similarly, a velocity jump at the phase interface can be investigated in figure 7.14. Due to evaporation, liquid

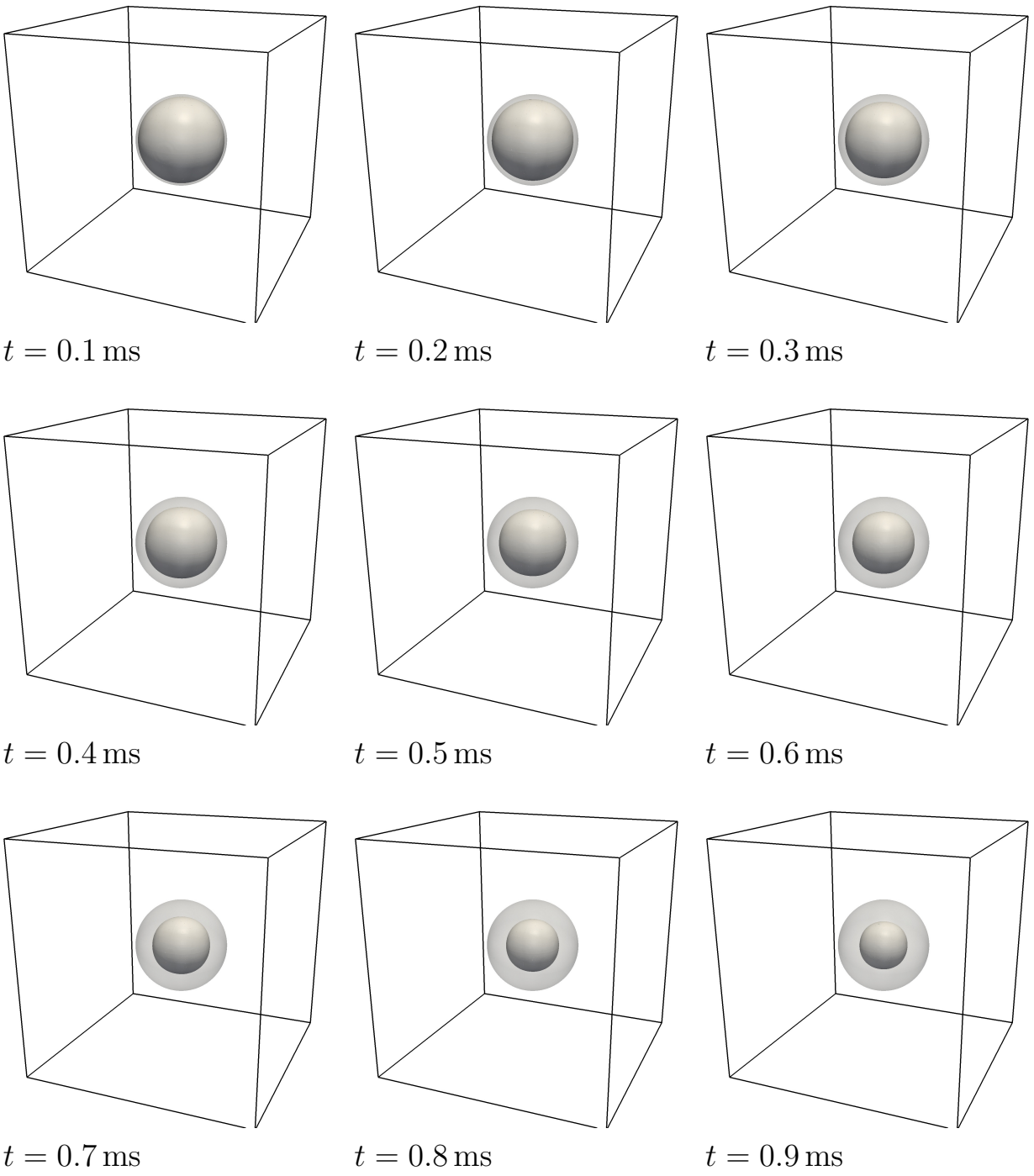


Figure 7.11: Time series of the evaporating dodecane droplet. The shadow structure visualizes the initial droplet size. For the final time, more than one half of the initial droplet mass is evaporated.

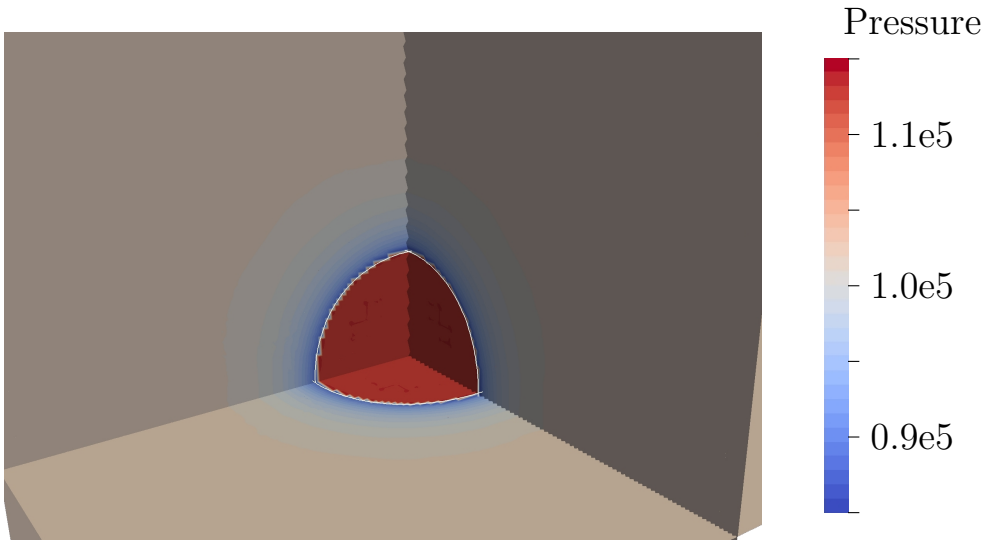


Figure 7.12: Instantaneous pressure distribution for $t = 0.6\,\mathrm{ms}$ for the evaporating droplet test case. One quadrant of the three-dimensional solution is shown. The position of the phase interface is indicated by a white line.

dodecane is vaporized and creates a high velocity next to the phase interface. The vaporization induces a gas flow outwards, away from the droplet. The induced velocity field diminishes with increasing distance from the droplet as the additional mass is distributed in the three-dimensional domain.

The resolution of vaporization effects has also an impact on the gas and liquid temperature in the computational domain. The vaporization process is endothermic, thus, energy is needed. This is modeled in the interface Riemann solver by the latent heat of vaporization. As a result the evaporated vapor has a lower temperature than the surrounding vapor and a thermal boundary layer around the droplet is developed. In the liquid phase the temperature is (nearly) unchanged due to the high heat capacity. Inside the temperature gradients are low due to Marangoni flows.

The Reynolds number of this evaporating droplet can be estimated using the initial droplet diameter, the bulk velocity at the interface ($v \approx 0.1\,{}^{m}/_{s}$; in the example no bulk flow is prescribed) and the gas viscosity. This results in a Reynolds number of

$$Re_d = \frac{\rho_{\mathrm{vap}} v D}{\mu_{\mathrm{vap}}} = 115.45\,. \tag{7.2}$$

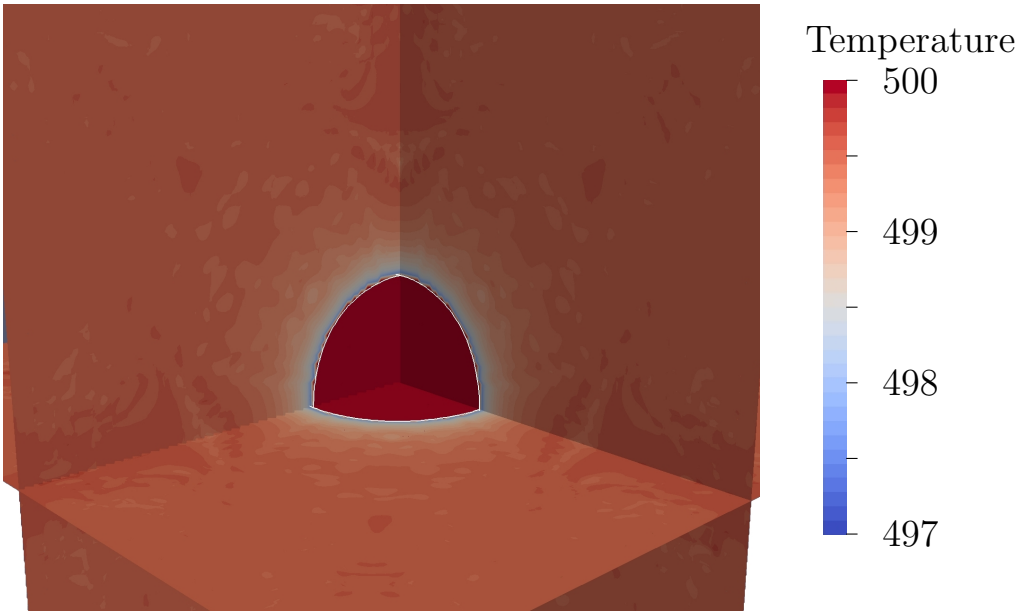


Figure 7.13: Instantaneous temperature distribution for $t = 0.6\,\mathrm{ms}$ for the evaporating droplet test case. One quadrant of the three-dimensional solution is shown. The position of the phase interface is indicated by a white line.

Similarly, the Schmidt number can be estimated to

$$Sc = \frac{\mu_{\mathrm{vap}}}{\rho_{\mathrm{vap}} D_{AB}} = 0.238\,. \tag{7.3}$$

using the diffusion coefficient of dodecane estimated in section A.2. With this non-dimensional numbers the Sherwood number of this test case can be estimated using the Ranz-Marshall correlation

$$Sh = 2 + 0.6 Re_d^{0.5} Sc^{1/3} = 9.067\,. \tag{7.4}$$

7.3 Conclusion

In this section the multi-dimensional validation of the HMM is shown. The inclusion of surface tension as forces within the interface Riemann solver description is validated using several test cases. The resolution within the local solution at the interface as well as an accurate curvature calculation limits the occurrence of spurious currents. The applicability of the method is shown using a compressible shock-droplet interaction.

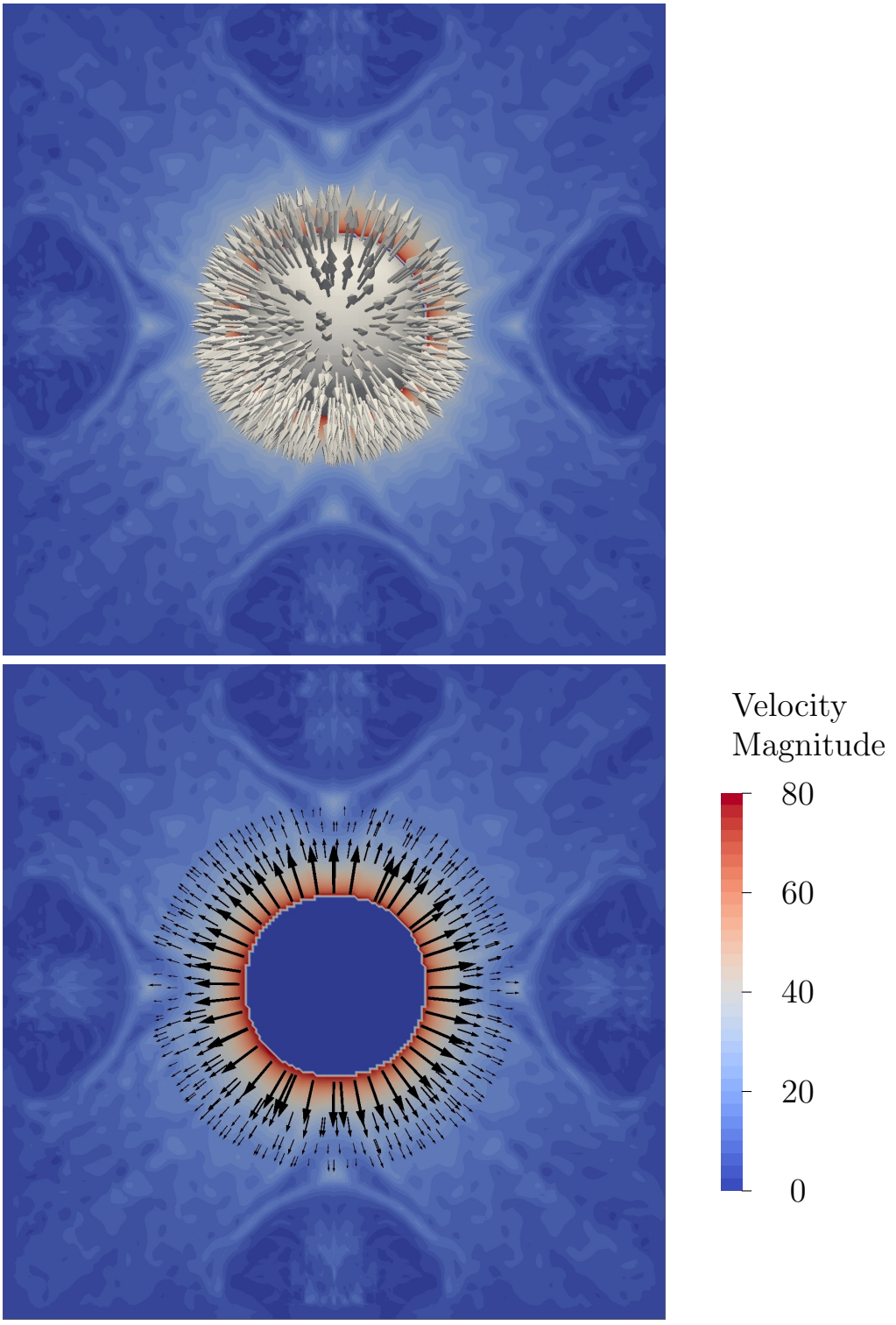


Figure 7.14: Instantaneous velocity distribution for $t = 0.6\,\mathrm{ms}$ for the evaporating droplet test case in the x-y-plane. **Top:** Velocity vectors on the phase interface. **Bottom:** Velocity vectors around the droplet in the x-y-plane. The velocity vectors point outwards due to the considered evaporation effects.

In a second step the description is extend to the resolution of phase transition effects. The application of the HLLP interface Riemann solver is validated using a single evaporating droplet. Due to the usage of the finite-volume subcells in the vicinity of the phase interface together with the resolution of the jump conditions by the HLLP solver no stability issues have been investigated. This is on the one hand related to the TVD stability of the FV subcells and on the other hand to the interface resolution. Thus, an accurate and robust numerical method has been obtained.

8 Conclusion and Prospects

8.1 Conclusion

In this thesis a numerical method for the simulation of compressible multi-phase flows based on a sharp interface approximation was presented. A compressible description of the multi-phase method was chosen due to the high pressure and high temperature ambient conditions investigated that are present in recent diesel injection systems and rocket combustion chambers. At such extreme ambient conditions the compressibility of both phases can not be neglected. Of major interest are the effects occurring during the transition from a subcritical state to a supercritical state. One example for such an application is the injection and evaporation of subcritical fuel droplets in combustion chambers. Due to the surrounding conditions the droplets evaporate quickly and reach a supercritical state. In droplet experiments characteristic dense gas regions behind the droplet can be investigated that can not be computed using the incompressible solution strategy.
The inclusion of compressibility effects in both phases introduces additional difficulties in the computation of the multi-phase flows: Due to compressible treatment the conservation equations of mass, momentum and energy are coupled closely via the equation of state and the commonly used decoupling of hydrodynamics and thermodynamics can not be used any more. This coupling implies, furthermore, that the inclusion of interface effects like surface tension and phase transition have to be done in a thermodynamic consistent way. In this thesis only the multi-phase system vapor-liquid is investigated for pure fluids (only one species).
In the second chapter the heterogeneous multi-scale method is described that is used for the resolution of the different scales within the flow field. Here, a sharp interface method based on the ideas of the ghost fluid method is applied to resolve the interface. A multi-scale method is chosen to be able to resolve the small scales at the phase interface. Local interface related effects like surface tension and phase transition are resolved by generalized Riemann based solver

at the phase boundary. This local solution at the phase boundary takes all interface related physics into account by means of appropriate fluxes that reproduce the proper waves in the bulk phases. The macro-scales are resolved by a discontinuous Galerkin spectral element method for the bulk flow as well as the level-set based interface tracking. All interface related effects are resolved by a special interface Riemann solver that resolves the interface jump conditions and provides suitable fluxes for the bulk phases. Through this modeling standard single-phase flow solvers can be used in the bulk phases.

In the third chapter the discontinuous Galerkin spectral element method is described. This method is used as macro-scale model for the resolution of the bulk phases within the flow field as well as the interface tracking. The numerical approximation of the phase interface is increased by a local finite-volume sub-cell refinement at the phase boundary. In the vicinity of the interface the continuous in-cell resolution of the DG method is replaced by a sub-cell finite-volume description. As the finite-volume allows a simple approximation of discontinuities this method is advantageous for the interface resolution. This sub-cell finite-volume method enters the DG description by means of a modified volume integral.

The sharp interface method used here, necessitates the use of an accurate method to track the interface. Here, a level-set method for compressible flow is used for interface tracking that is discretized with the high-order DG method. The level-set description is chosen as it is easier discretizable using the DG method. The high order is advantageous due to the low dissipation and dispersion errors leading to a low mass loss associated with the interface movement. The second advantage is the straightforward estimation of secondary interface quantities (interface curvature and interface normals) due to the high-order ansatz functions. The accuracy in the estimation of the curvature can be further increased with the use of a posteriori $\mathrm{P}_N\mathrm{P}_M$-reconstruction. The level-set method has been successfully validated using different validation examples available in literature, including Zalesak's disk and a vortex stretching.

An overview of different equations of state used in the scope of multi-phase flows is given in chapter 5. Different types of EOS are available from a simple approximation based on the stiffened gas EOS with a linear temperature dependence on the pressure up to highly accurate polynomials having up to 53 terms. All these approaches try to approximate the dissimilar behavior of the liquid and vapor phase: the liquid phase is relatively stiff and hardly compressible while the vapor phase is much easier compressible. The main disadvantage

of high-accurate EOS are the long computation times associated with the EOS evaluation. This can be avoided with the usage of a two-dimensional lookup table that is computed prior to the simulation and is interpolated during the runtime. This approach allows for speedups up to a factor of 1000 compared to the direct EOS evaluation and is only 50% more expensive compared to the simple ideal gas approximation.
The local resolution of interface effects by means of special Riemann-based solvers is described in chapter 6. In the simple case without phase transition the traditional Riemann solution with three distinct waves is recovered. This allows for an easy approximation based on linearized Riemann solvers in which the surface tension effects are included in the Riemann solution. In case phase transition effects are considered, the Riemann wave fan consists of four distinct waves. To be able to ensure uniqueness of the solution a kinetic relation is needed that determines the amount of mass transfer. This approximation has to be in accordance with the second law of thermodynamics. Several models for the estimation of the evaporation rate are present in literature. The generalized Riemann problem is solved such that a specific entropy production at the interface is modeled. With this generalized Riemann solver it is possible to resolve thermodynamic non-equilibrium effects. Their influence on the solution is assessed with a comparison to a simple thermodynamic equilibrium model. It is shownthat the approach is in good agreement to literature results obtained with the Baer-Nunziato mixture model and an explicit phase transition modeling.
These building blocks of the heterogeneous multi-scale method have been validated at several two- and three-dimensional test cases. The test cases include the compressible shock-droplet interaction as well as a test for the inclusion of surface tension effects by a measure of the induced currents at the interface due to surface tension modeling a wobbling droplet. In a second step this method is generalized to test cases with phase transfer effects and their influence on the solution. In this scope an evaporating droplet is shown.
This method is able to resolve the different scales in multi-phase flows as well as it is able to include local interface effects into the algorithm. Improved models for the interface resolution can be implemented by means of an adapted model in the generalized Riemann problem at the phase interface.

8.2 Prospects

In the long term, the aim is to apply the new numerical method to sprays and droplets at extreme ambient conditions at and near critical conditions. These conditions can be found in aeronautical injection systems, e. g. in diesel injection systems, rocket engines or flash boiling effects. For elevated pressures this compressible multi-phase method is able to resolve all waves present in the problem. Hence, an improved understanding of the associated physical processes should be possible as the numerical approach provides more information about the flow than an experimental investigation does.
In the last three years the heterogeneous multi-scale method has been developed for compressible multi-phase flows. Still, some important parts are missing to be able to provide a reliable approximation of multi-phase in industrial applications and in experiments. This is in line with a further validation of the numerical method to experiments in the high-pressure and high-temperature region. The overall aim is the applicability of the compressible multi-phase method to experimental results that requires the solution of large-scale problems, e. g. spray simulations.
One major simplification up to now is that a pure fluid is assumed in the liquid and vapor. However, in experiments typically a liquid droplet is placed into a nitrogen environment (or a different non-reacting gas). This surrounding foreign gas has huge impacts onto the evaporation rates and the droplet behavior inside the experimental chamber. Due to mixing effects onto the underlying EOS, the critical point of the mixture changes rapidly away from the critical point of a single fluid. To be able to assess the effects, a mixture model for the both phase is needed that complicates the underlying EOS description additionally. This gets even more difficult in case mixtures in the droplet are present, as e. g. in bio-fuels. Depending on the droplet composition and the surrounding conditions, first one liquid component evaporates totally until the other starts to evaporate. The estimate which component starts to evaporate first is sometimes thermodynamically not trivial. Mixture descriptions are possible based on several EOS, e. g. the Peng-Robinson EOS. Work was done within the SFB-TRR 75 to extend the description of the PCP-SAFT EOS for multi-component gases and liquids. The main advantage is that this approach is also valid in the vicinity of the critical point, a region that is not available in many simple approaches.

The resolution of compressible multi-phase flows is rather costly due to the restrictive explicit time step restriction of the numerical scheme based on the sound speed. Such a restriction is not present in an incompressible scheme. This implies large computation times for the resolution of practical examples that necessitate an efficient parallelization of the underlying numerical method. The DG method is proven to provide excellent scaling results on todays supercomputers. This approach has to be tested for such large scale problems and the load imbalance due to the interface resolution has to be taken into account. The computation time for an interface Riemann problem is about 10–100 times longer compared to a single-phase Riemann problem.
Another drawback is the large computational effort for the resolution of the generalized Riemann problem at the phase interface in case phase transfer effects are present. The present exact Riemann solver is very costly but needed for the validation of the numerical method. A further reduction in computation time could be possible with the use of approximate Riemann solvers. A first approach based on the HLL-approximation was introduced here. This investigation has to be improved e. g. with a better estimate of the wave speeds based on a Roe linearization at the interface or an improved model for the kinetic relation that estimates the mass transfer rate.
Of major interest within the collaborative research council SFB-TRR75 is the investigation of droplets at or in the vicinity of the critical point. This necessitates the numerical and experimental study of droplets in this region where compressibility effects can not be neglected. For industrial applications, the effects of trans-critical phenomena, e. g. a droplet that is injected in subcritical conditions and reaches a supercritical state within the experiment, is of major interest. This region is not representable using the incompressible approach and the research is limited to experimental investigations. Some effects occurring in this region are not totally understood and the developed numerical method for compressible multi-phase flows may contribute to the understanding of these effects.

A Appendix

A.1 Characteristics of the investigated fluids

In this section the fluid properties of the investigated fluids acetone, n-dodecane and water are listed for reference. The values are taken from CoolProp [8].

A.2 Estimation of the diffusion coefficient for dodecane

The estimation of the diffusion coefficient for dodecane is based on the correlations published in Poling et al. [75] and a correlation for the self-diffusion of alkanes published by Yu and Gao in [106]. Both methods are based on the estimation of the Lennard-Jones potential.
For dodecane the characteristic length σ_{AB} and the characteristic Lennard-Jones energy ϵ_{AB} are

$$\sigma_{AB} = 5.474\,10^{-10}\,m \tag{A.1}$$

$$\frac{\epsilon_{AB}}{k} = 228.54\,{}^{1}/_{K}\,, \tag{A.2}$$

that are taken from [106]. This estimation is based on the theory of Chapman and Enskog. Now the Lennard-Jones potential Λ_p can be estimated

$$\Omega_p = \frac{A}{(T^*)^B} + \frac{C}{\exp(DT^*)} + \frac{E}{\exp(FT^*)} + \frac{G}{\exp(HT^*)} \tag{A.3}$$

with

$$\begin{array}{lll} T^* = kT/\epsilon_{AB} & A = 1.06036 & B = 0.15610 \\ C = 0.19300 & D = 0.47635 & E = 1.03587 \\ F = 1.52996 & G = 1.76474 & H = 3.89411 \end{array} \tag{A.4}$$

Parameter		Value
Triple point		
Triple point temperature	K	263.6
Triple point pressure	Pa	0.626
Critical point		
Critical point temperature	K	658.1
Critical point pressure	Pa	1817000
Critical point density	kg/m^3	226.545
Other values		
Mole mass	kg/mol	170.335
Accentric factor	–	0.575
CAS number		112-40-3

Table A.1: Fluid properties for n-dodecane.

Parameter		Value
Triple point		
Triple point temperature	K	178.5
Triple point pressure	Pa	2.326
Critical point		
Critical point temperature	K	508.1
Critical point pressure	Pa	4700000
Critical point density	kg/m^3	272.972
Other values		
Mole mass	kg/mol	58.079
Accentric factor	–	0.309
CAS number		67-64-1

Table A.2: Fluid properties for acetone.

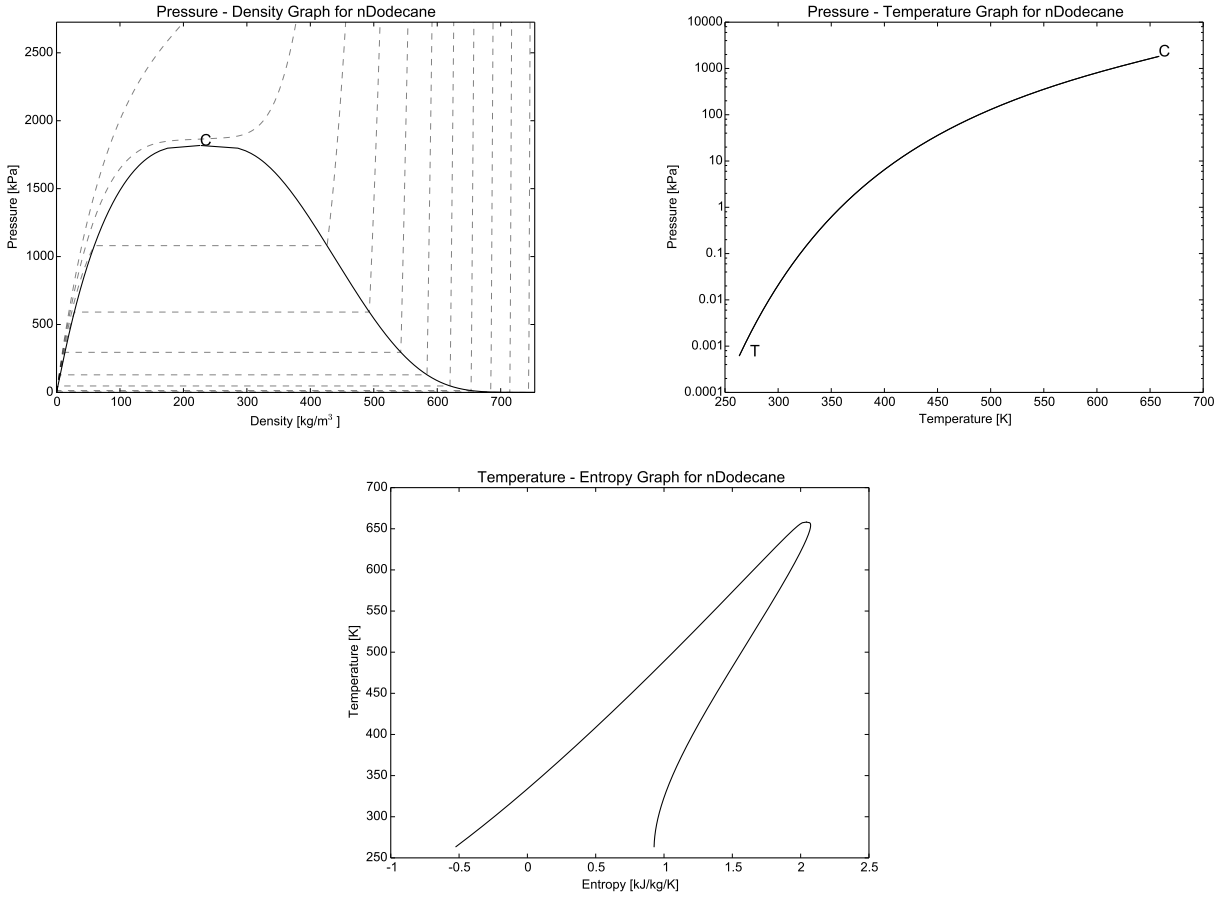


Figure A.1: Plot of the fluid properties for n-dodecane. The black line visualizes the saturation line, C marks the critical point and T the triple point. The dashed lines are the isotherms in the p-ρ-plots in the range between 300 and 700K. In the T-s-diagram the retrograde fluid behavior can be seen.

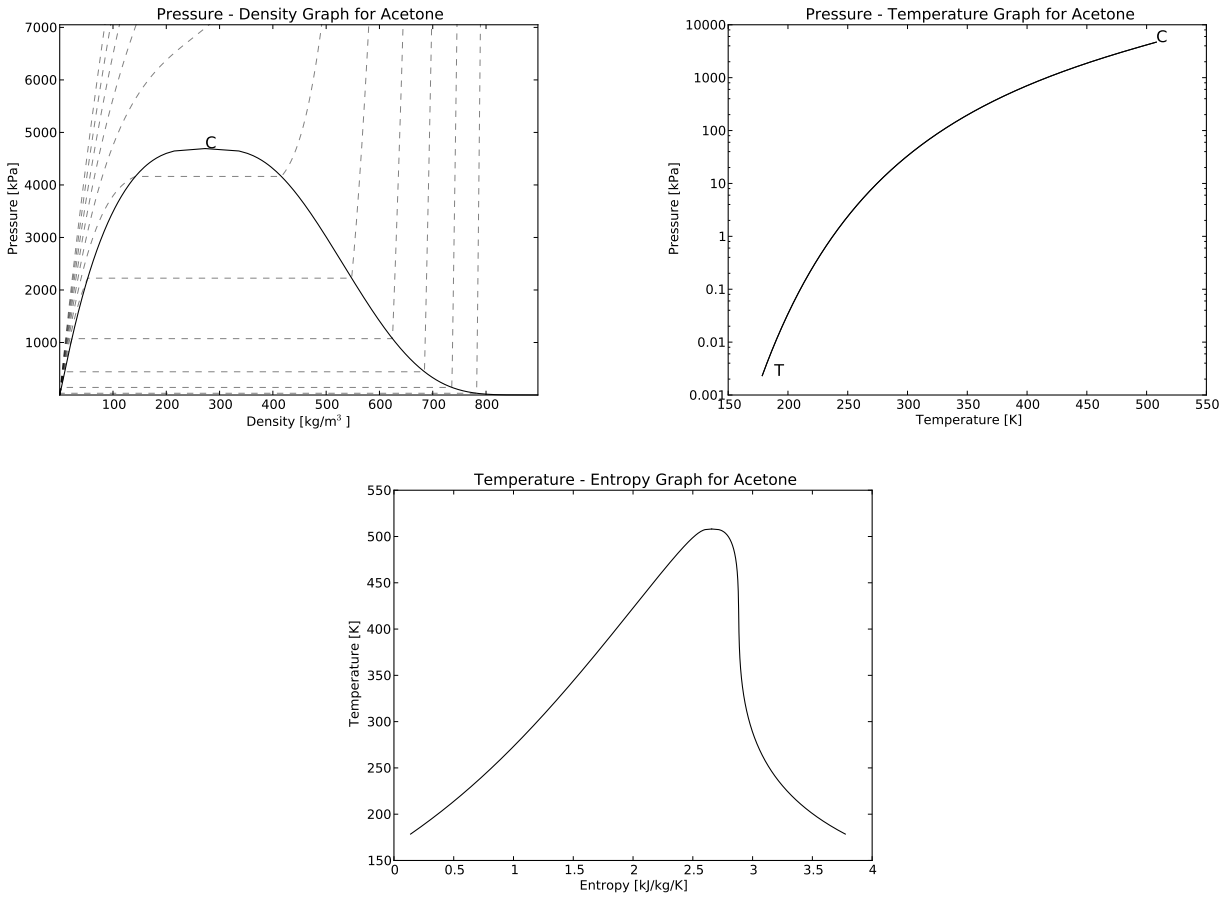


Figure A.2: Plot of the fluid properties for acetone. The black line visualizes the saturation line, C marks the critical point and T the triple point. The dashed lines are the isotherms in the p-ρ-plots in the range between 300 and 700K.

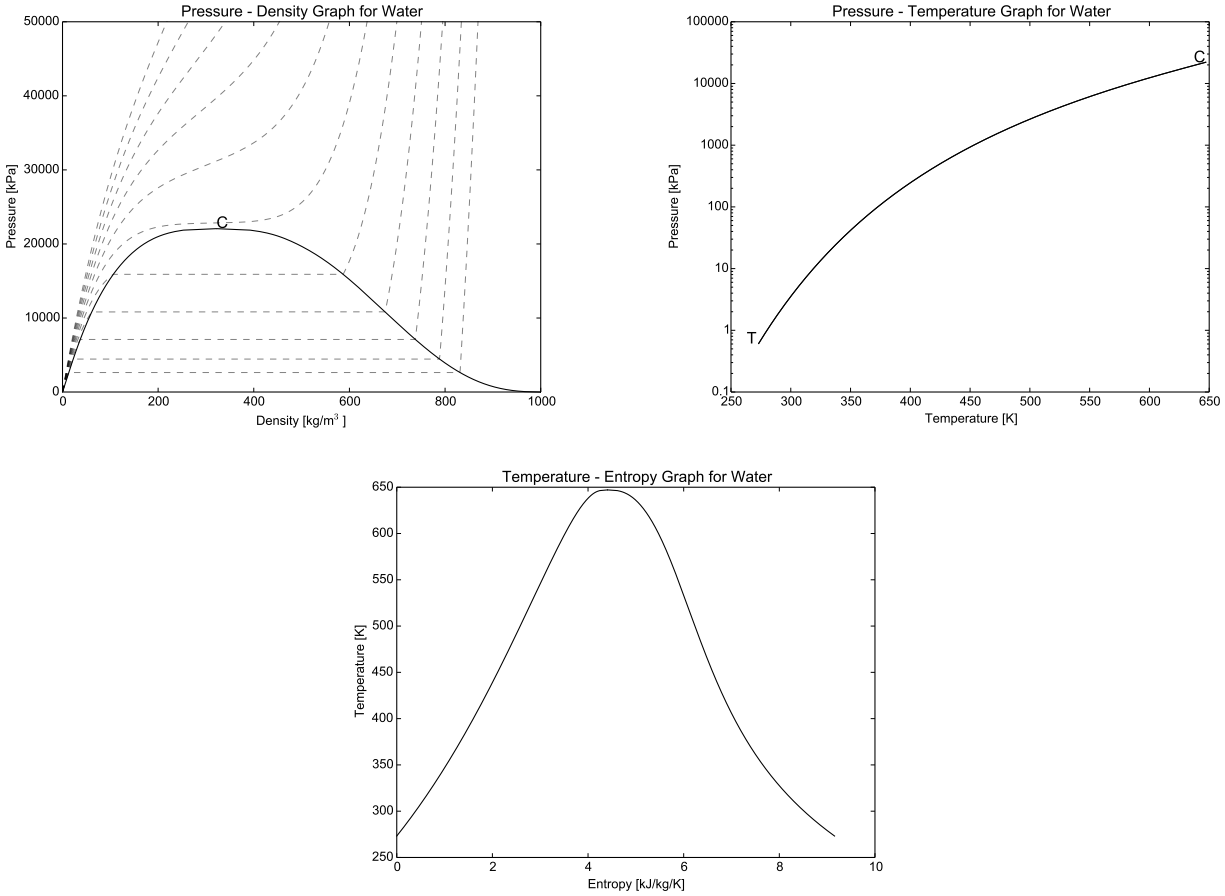


Figure A.3: Plot of the fluid properties for water. The black line visualizes the saturation line, C marks the critical point and T the triple point. The dashed lines are the isotherms in the p-ρ-plots in the range between 500 and 800K.

Parameter		Value
Triple point		
Triple point temperature	K	273.16
Triple point pressure	Pa	611.65
Critical point		
Critical point temperature	K	647.096
Critical point pressure	Pa	22064000.0
Critical point density	kg/m^3	322.0
Other values		
Mole mass	kg/mol	18.015
Accentric factor	–	0.344
CAS number		7732-18-5

Table A.3: Fluid properties for water.

Now, the diffusion coefficient can be estimated to

$$D_{AB} = \frac{3}{16} \frac{(4\pi kT/M_{AB})^{1/2}}{n\pi\sigma_{AB}\Omega_p} f_D \tag{A.5}$$

with

$$M_{AB} = 2\left[1/M_A + 1/M_B\right]^{-1}$$

with M_A, M_B being th molecular weights of A and B.

n = number density of molecules in the mixture

k = Boltzmann's constant

T = absolute temperature

f_D = correction term in the order of unity

For dodecane equation (A.5) predicts a self-diffusion coefficient at a temperature $T = 500\text{K}$ and a pressure of $p = 1\,\text{bar}$

$$D_{AB} = 7.289\,10^{-6}\,\frac{m^2}{s}\,. \tag{A.6}$$

A.3 Derivation of the HLLP star state

Left star state $\mathbf{U}_L^* = (\rho_L^*, v_L^*, e_L^*)^{\mathrm{T}}$:

$$\begin{aligned}
\rho_L^* =&(2S_L\rho_L^2 v_L + \rho_L v_L \rho_R v_R - S^\sharp \rho_L v_L + S^\sharp \rho_R v_R - S_L^2\rho_L^2 \\
&+ S_L\rho_L S_R\rho_R + S^\sharp S_L\rho_L - S^\sharp S_R\rho_R - \rho_L v_L S_R\rho_R - \rho_L^2 v_L^2 \\
&- S_L\rho_L\rho_R v_R)/(-\rho_L v_L^2 + \rho_R v_R^2 - \rho_L S_L^2 + S^\sharp S_L - S^\sharp S_R - S_R\rho_R v_R \\
&+ 2S_L\rho_L v_L + p_R - p_L - \Delta p - S_L\rho_R v_R + S_L S_R\rho_R) \qquad \text{(A.7)} \\
v_L^* =&(-S_R\rho_R v_R\rho_L v_L + S_R\rho_R v_R S_L\rho_L + p_L S_L\rho_L + \Delta p S_L\rho_L + 2\rho_L^2 v_L^2 S_L \\
&- \rho_L v_L S^\sharp S_R + S^\sharp S_L\rho_R v_R + S_L\rho_L S^\sharp S_R - p_L\rho_L v_L - S_L^2\rho_L^2 v_L \\
&+ \rho_L v_L\rho_R v_R^2 - \rho_L^2 v_L^3 - S_L\rho_L\rho_R v_R^2 - \rho_L v_L\Delta p + \rho_L v_L p_R - S_L\rho_L p_R \\
&- S_L S_R\rho_R S^\sharp)/(2S_L\rho_L^2 v_L + \rho_L v_L\rho_R v_R - S^\sharp\rho_L v_L + S^\sharp\rho_R v_R \\
&- S_L^2\rho_L^2 + S_L\rho_L S_R\rho_R + S^\sharp S_L\rho_L - S^\sharp S_R\rho_R - \rho_L v_L S_R\rho_R \\
&- \rho_L^2 v_L^2 - S_L\rho_L\rho_R v_R) \qquad \text{(A.8)} \\
p_L^* =&(2p_R S_L\rho_L^2 v_L + p_L S_L\rho_L S_R\rho_R - p_L S_L\rho_L\rho_R v_R + \rho_L S_L^2 S^\sharp\rho_R v_R \\
&+ p_L\rho_L v_L\rho_R v_R - p_L\rho_L v_L S_R\rho_R + 2S_L S_R\rho_R\rho_L^2 v_L^2 - 2S_R\rho_R v_R S_L\rho_L^2 v_L \\
&+ \rho_L^2 S_L^2 v_L\rho_R v_R + 2\rho_L^2 v_L^2 S_L S^\sharp - \rho_L S_L^2 S^\sharp S_R\rho_R + S_L^2\rho_L^2 S^\sharp S_R \\
&- 2\Delta p\rho_L^2 v_L S_L - S_L^2\rho_L^2 v_L S^\sharp + 2\rho_R v_R^2 S_L\rho_L^2 v_L - 2S^\sharp S_L\rho_L v_L\rho_R v_R \\
&+ 2S^\sharp S_L\rho_L v_L S_R\rho_R + \Delta p S_L^2\rho_L^2 + S_R\rho_R v_R S_L^2\rho_L^2 + S_R\rho_R v_R\rho_L^2 v_L^2 \\
&- 2S_L\rho_L^2 v_L S^\sharp S_R - p_R\rho_L^2 v_L^2 + \rho_L v_L^2 S^\sharp\rho_R v_R - \rho_L v_L^2 S^\sharp S_R\rho_R \\
&- 2\rho_L^2 v_L^2 S_L\rho_R v_R - \rho_L^2 S_L^2 v_L S_R\rho_R + p_L S^\sharp\rho_R v_R + \rho_L^2 v_L^2 S^\sharp S_R \\
&- p_L S^\sharp S_R\rho_R + \rho_L^2 v_L^3\rho_R v_R - \rho_R v_R^2 S_L^2\rho_L^2 - \rho_R v_R^2\rho_L^2 v_L^2 \\
&- p_L S^\sharp\rho_L v_L + S^\sharp S_L p_L\rho_L - \rho_L^2 v_L^3 S_R\rho_R - \rho_L^2 v_L^3 S^\sharp - p_R S_L^2\rho_L^2 \\
&+ \Delta p\rho_L^2 v_L^2)/(2S_L\rho_L^2 v_L + \rho_L v_L\rho_R v_R - S^\sharp\rho_L v_L + S^\sharp\rho_R v_R \\
&- S_L^2\rho_L^2 + S_L\rho_L S_R\rho_R + S^\sharp S_L\rho_L - S^\sharp S_R\rho_R - \rho_L v_L S_R\rho_R - \rho_L^2 v_L^2 - \\
&S_L\rho_L\rho_R v_R) \qquad \text{(A.9)} \\
e_L^* =&\frac{1}{\rho_L^*(v_L^* - S_L)}(-S_L\rho_L e_L + v_L(\rho_L e_L + p_L) - v_L^* p_L^*) \qquad \text{(A.10)}
\end{aligned}$$

Right star state $\mathbf{U}_R^* = (\rho_R^*, v_R^*, e_R^*)^{\mathrm{T}}$:

$$\begin{aligned}
\rho_R^* =&(\rho_L v_L S_R \rho_R - S^\sharp \rho_L v_L - \rho_L v_L \rho_R v_R - S_L \rho_L S_R \rho_R + S^\sharp S_L \rho_L \\
&+ S_L \rho_L \rho_R v_R - 2 S_R \rho_R^2 v_R + S_R^2 \rho_R^2 - S^\sharp S_R \rho_R + S^\sharp \rho_R v_R + \rho_R^2 v_R^2)/ \\
&(-\rho_L v_L^2 + S_L \rho_L v_L + \rho_L v_L S_R - \rho_L S_L S_R + S_R^2 \rho_R - 2 S_R \rho_R v_R - p_L + \rho_R v_R^2 \\
&- \Delta p + S^\sharp S_L - S^\sharp S_R + p_R) \qquad \text{(A.11)} \\
v_R^* =&(S_R^2 \rho_R^2 v_R - \rho_L v_L S^\sharp S_R + S^\sharp S_L \rho_R v_R + S_L \rho_L S^\sharp S_R - 2 \rho_R^2 v_R^2 S_R \\
&- p_R S_R \rho_R + p_R \rho_R v_R - S_L S_R \rho_R S^\sharp - \rho_L v_L^2 \rho_R v_R + \rho_R^2 v_R^3 + S_L \rho_L v_L \rho_R v_R \\
&- S_L \rho_L v_L S_R \rho_R + \Delta p S_R \rho_R + p_L S_R \rho_R - p_L \rho_R v_R - \Delta p \rho_R v_R + \rho_L v_L^2 S_R \rho_R)/ \\
&(\rho_L v_L S_R \rho_R - S^\sharp \rho_L v_L - \rho_L v_L \rho_R v_R - S_L \rho_L S_R \rho_R + S^\sharp S_L \rho_L \\
&+ S_L \rho_L \rho_R v_R - 2 S_R \rho_R^2 v_R + S_R^2 \rho_R^2 - S^\sharp S_R \rho_R + S^\sharp \rho_R v_R + \rho_R^2 v_R^2) \qquad \text{(A.12)} \\
p_R^* =&(p_R S^\sharp S_L \rho_L - p_R S^\sharp S_R \rho_R + p_R S^\sharp \rho_R v_R - p_R S^\sharp \rho_L v_L \\
&- 2 S_R \rho_R^2 v_R p_L + \rho_R v_R^2 S^\sharp S_L \rho_L + 2 \rho_R^2 v_R^2 \rho_L v_L S_R - 2 S_R \rho_R^2 v_R \rho_L v_L^2 \\
&+ 2 S_L \rho_R^2 v_R \rho_L v_L S_R - \rho_L v_L S_R^2 \rho_R S^\sharp - S_L S_R^2 \rho_R^2 \rho_L v_L + S_R^2 \rho_R^2 \rho_L v_L^2 \\
&- 2 S_R \rho_R v_R S^\sharp S_L \rho_L - 2 \rho_R^2 v_R^2 S_L \rho_L S_R + 2 S_L \rho_R^2 v_R S^\sharp S_R \\
&+ \rho_R^2 v_R^2 \rho_L v_L^2 - \rho_R v_R^2 S^\sharp \rho_L v_L + S_L \rho_L S_R^2 \rho_R S^\sharp + 2 S_R \rho_R v_R S^\sharp \rho_L v_L \\
&- 2 S_R \rho_R^2 v_R \Delta p - S_R^2 \rho_R^2 v_R \rho_L v_L + S_R^2 \rho_R^2 v_R S_L \rho_L - S_L S_R^2 \rho_R^2 S^\sharp \\
&- S_L \rho_R^2 v_R^2 \rho_L v_L + \rho_R^2 v_R^3 S_L \rho_L - 2 \rho_R^2 v_R^2 S^\sharp S_R - \rho_R^2 v_R^3 \rho_L v_L \\
&- S_L \rho_R^2 v_R^2 S^\sharp + S_R^2 \rho_R^2 v_R S^\sharp - p_R S_L \rho_L S_R \rho_R + S_R^2 \rho_R^2 \Delta p \\
&+ p_R S_L \rho_L \rho_R v_R - p_R \rho_L v_L \rho_R v_R + p_R \rho_L v_L S_R \rho_R + \rho_R^2 v_R^2 p_L + \rho_R^2 v_R^2 \Delta p \\
&+ S_R^2 \rho_R^2 p_L + \rho_R^2 v_R^3 S^\sharp)/(\rho_L v_L S_R \rho_R - S^\sharp \rho_L v_L - \rho_L v_L \rho_R v_R \\
&- S_L \rho_L S_R \rho_R + S^\sharp S_L \rho_L + S_L \rho_L \rho_R v_R - 2 S_R \rho_R^2 v_R + S_R^2 \rho_R^2 - S^\sharp S_R \rho_R \\
&+ S^\sharp \rho_R v_R + \rho_R^2 v_R^2) \qquad \text{(A.13)} \\
e_R^* =&\frac{1}{\rho_R^*(v_R^* - S^\sharp)}(\Delta p S^\sharp - S^\sharp \rho_L^* e_L^* - v_R^* p_R^* + v_L^*(\rho_L^* e_L^* + p_L^*) - j h_{\mathrm{lat}}) \qquad \text{(A.14)}
\end{aligned}$$

Bibliography

[1] R. Abeyaratne and J. Knowles. *Evolution of Phase Transitions: A Continuum Theory*. Cambridge University Press, 2006. ISBN: 9780521661478.

[2] D. Adalsteinsson and J. Sethian. "The Fast Construction of Extension Velocities in Level Set Methods". In: *Journal of Computational Physics* 148.1 (1999), pp. 2–22. ISSN: 0021-9991.

[3] C. Altmann, G. Gassner, F. Lörcher, and C.-D. Munz. "A Space-Time Expansion Discontinuous Galerkin Scheme With Local Time Stepping for the Ideal and Viscous MHD Equations". In: *Plasma Science, IEEE Transactions on* 37.4 (2009), pp. 513–519.

[4] D. M. Anderson, G. B. McFadden, and A. A. Wheeler. "Diffuse-interface methods in fluid mechanics". In: *Annual review of fluid mechanics* 30.1 (1998), pp. 139–165.

[5] M. Bardi and S. Osher. "The nonconvex multidimensional Riemann problem for Hamilton-Jacobi equations". In: *SIAM Journal on Mathematical Analysis* 22.2 (1991), pp. 344–351.

[6] F. Bassi and S. Rebay. "A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations". In: *J. Comput. Phys.* 131 (1997), pp. 267–279.

[7] A. D. Beck, T. Bolemann, D. Flad, H. Frank, G. J. Gassner, F. Hindenlang, and C.-D. Munz. "High-order discontinuous Galerkin spectral element methods for transitional and turbulent flow simulations". In: *International Journal for Numerical Methods in Fluids* 76.8 (2014), pp. 522–548.

[8] I. H. Bell, J. Wronski, S. Quoilin, and V. Lemort. "Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp". In: *Industrial & Engineering Chemistry Research* 53.6 (2014), pp. 2498–2508.

[9] T. Bolemann, S. Fechter, C.-D. Munz, and M. Dumbser. “Fast and accurate tabulation of reference-quality equations of state for 3D compressible multi-phase flow simulations”. In Preparation. 2015.

[10] 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.

[11] C. Chalons, F. Coquel, P. Engel, and C. Rohde. “Fast relaxation solvers for hyperbolic-elliptic phase transition problems”. In: *SIAM Journal on Scientific Computing* 34.3 (2012), A1753–A1776.

[12] C.-H. Chang and M.-S. Liou. “A robust and accurate approach to computing compressible multiphase flow: Stratified flow model and AUSM+-up scheme”. In: *Journal of Computational Physics* 225.1 (2007), pp. 840–873.

[13] D. L. Chapman. “VI. On the rate of explosion in gases”. In: *Philosophical Magazine Series 5* 47.284 (1899), pp. 90–104.

[14] S. Chen, B. Merriman, S. Osher, and P. Smereka. “A simple level set method for solving Stefan problems”. In: *Journal of Computational Physics* 135.1 (1997), pp. 8–29.

[15] B. Cockburn and C.-W. Shu. “The local discontinuous Galerkin method for time-dependent convection-diffusion systems”. In: *SIAM Journal on Numerical Analysis* 35.6 (1998), pp. 2440–2463.

[16] M. G. Crandall and P.-L. Lions. “Two approximations of solutions of Hamilton-Jacobi equations”. In: *Mathematics of Computation* 43.167 (1984), pp. 1–19.

[17] P. G. Debenedetti. *Metastable liquids: concepts and principles*. Princeton University Press, 1996.

[18] A. Dressel and C. Rohde. “A finite-volume approach to liquid-vapour fluids with phase transition”. In: *Finite volumes for complex applications V*. ISTE, London, 2008, pp. 53–68.

[19] W. Dreyer. *On jump conditions at phase boundaries for ordered and disordered phases*. WIAS, 2003.

[20] M. Dumbser. “Arbitrary high order $P_N P_M$ schemes on unstructured meshes for the compressible Navier–Stokes equations”. In: *Computers & Fluids* 39.1 (2010), pp. 60–76.

[21] M. Dumbser. *Arbitrary High Order Schemes for the Solution of Hyperbolic Conservation Laws in Complex Domains*. Aachen: Shaker Verlag, 2005.

[22] M. Dumbser, U. Iben, and C.-D. Munz. "Efficient implementation of high order unstructured WENO schemes for cavitating flows". In: *Computers & Fluids* 86 (2013), pp. 141–168.

[23] M. Emmett. *PyWENO*. URL: `http://pyweno.readthedocs.org/en/latest/index.html`.

[24] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. "A hybrid particle level set method for improved interface capturing". In: *Journal of Computational Physics* 183.1 (2002), pp. 83–116.

[25] S. Fechter, F. Hindenlang, H. Frank, C.-D. Munz, and G. Gassner. "Discontinuous Galerkin Schemes for the Direct Numerical Simulation of Fluid Flow and Acoustics". In: Aeroacoustics Conferences. American Institute of Aeronautics and Astronautics, June 2012.

[26] S. Fechter, F. Jaegle, and V. Schleper. "Exact and approximate Riemann solvers at phase boundaries". In: *Computers & Fluids* 75 (2013), pp. 112–126.

[27] S. Fechter and C.-D. Munz. "A discontinuous Galerkin-based sharp-interface method to simulate three-dimensional compressible two-phase flow". In: *International Journal for Numerical Methods in Fluids* 78.7 (2015), pp. 413–435.

[28] S. Fechter, C.-D. Munz, C. Rohde, and C. Zeiler. "A sharp interface method for compressible liquid-vapor flow with phase transition and surface tension". Submitted to the Journal of Computational Physics. 2015.

[29] S. Fechter, C.-D. Munz, C. Rohde, and C. Zeiler. "Approximate Riemann solver for compressible liquid vapor flow with phase transition and surface tension". In Preparation. 2015.

[30] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. "A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)". In: *J. Comput. Phys.* 152.2 (1999), pp. 457–492.

[31] D. Flad, A. D. Beck, G. Gassner, and C.-d. Munz. "A Discontinuous Galerkin Spectral Element Method for the direct numerical simulation of aeroacoustics". In: AIAA Aviation. American Institute of Aeronautics and Astronautics, June 2014.

[32] G. Gassner. "Discontinuous Galerkin methods for the unsteady compressible Navier-Stokes equations". PhD thesis. Universität Stuttgart, 2009. ISBN: 978-3-86853-011-7.

[33] P. Glaister. "An approximate linearised Riemann solver for the Euler equations for real gases". In: *Journal of Computational Physics* 74.2 (1988), pp. 382–408.

[34] J. Grooss and J. S. Hesthaven. "A level set discontinuous Galerkin method for free surface flows". In: *Computer Methods in Applied Mechanics and Engineering* 195.25–28 (2006), pp. 3406–3429.

[35] J. Gross. "A density functional theory for vapor-liquid interfaces using the PCP-SAFT equation of state". In: *The Journal of chemical physics* 131.20 (2009), p. 204705.

[36] S. Hardt and F. Wondra. "Evaporation model for interfacial flows based on a continuum-field representation of the source terms". In: *Journal of Computational Physics* 227.11 (2008), pp. 5871–5895.

[37] A. Harten, P. D. Lax, and B. v. Leer. "On upstream differencing and Godunov-type schemes for hyperbolic conservation laws". In: *SIAM review* 25.1 (1983), pp. 35–61.

[38] D. Hartmann, M. Meinke, and W. Schröder. "Differential equation based constrained reinitialization for level set methods." In: *J. Comput. Physics* 227.14 (2008), pp. 6821–6845.

[39] D. Hartmann, M. Meinke, and W. Schröder. "The constrained reinitialization equation for level set methods." In: *J. Comput. Physics* 229.5 (2010), pp. 1514–1535.

[40] L. G. Hill and B. Sturtevant. *An experimental study of evaporation waves in a superheated liquid.* Springer, 1990.

[41] F. Hindenlang, G. J. Gassner, C. Altmann, A. Beck, M. Staudenmaier, and C.-D. Munz. "Explicit discontinuous Galerkin methods for unsteady problems". In: *Computers & Fluids* 61 (2012), pp. 86–93.

[42] C. W. Hirt and B. D. Nichols. "Volume of fluid (VOF) method for the dynamics of free boundaries". In: *J. Comput. Phys.* 39.1 (1981), pp. 201–225.

[43] Hokanomono. *A typical phase diagram.* URL: `http://commons.wikimedia.org/wiki/File:Phase-diag2.svg` (visited on 10/02/2014).

[44] X. Y. Hu, N. A. Adams, and G. Iaccarino. "On the HLLC Riemann solver for interface interaction in compressible multi-fluid flow". In: *J. Comput. Phys.* 228.17 (2009), pp. 6572–6589.

[45] G.-S. Jiang and D. Peng. "Weighted ENO schemes for Hamilton–Jacobi equations". In: *SIAM Journal on Scientific computing* 21.6 (2000), pp. 2126–2143.

[46] E. Johannessen, J. Gross, and D. Bedeaux. "Nonequilibrium thermodynamics of interfaces using classical density functional theory". In: *The Journal of chemical physics* 129.18 (2008), p. 184703.

[47] C. A. Kennedy, M. H. Carpenter, and R. M. Lewis. "Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations". In: *Applied numerical mathematics* 35.3 (2000), pp. 177–219.

[48] D. A. Kopriva. *Implementing Spectral Methods for Partial Differential Equations.* Springer, 2009.

[49] D. A. Kopriva. "Metric Identities and the Discontinuous Spectral Element Method on Curvilinear Meshes". In: *Journal of Scientific Computing* 26.3 (Mar. 2006), pp. 301–327.

[50] D. A. Kopriva and G. Gassner. "On the Quadrature and Weak Form Choices in Collocation Type Discontinuous Galerkin Spectral Element Methods". In: *J. Sci. Comput.* 44 (2010), pp. 136–155.

[51] Lamb. *Hydrodynamics.* Cambridge University Press, 1932.

[52] E. Lauer, X. Y. Hu, S. Hickel, and N. A. Adams. "Numerical investigation of collapsing cavity arrays". In: *Physics of Fluids* 24 (2012), p. 052104.

[53] P. G. LeFloch. "Kinetic relations for undercompressive shock waves. Physical, mathematical, and numerical issues". In: *Nonlinear partial differential equations and hyperbolic wave phenomena.* 2010, pp. 237–272.

[54] P. G. LeFloch and M. Shearer. "Non-classical Riemann solvers with nucleation". In: *Proceedings of the Royal Society of Edinburgh: Section A Mathematics* 134.05 (2004), pp. 961–984.

[55] O. LeMétayer, J. Massoni, and R. Saurel. "Elaborating equations of state of a liquid and its vapor for two-phase flow models; Elaboration des lois d'etat d'un liquide et de sa vapeur pour les modeles d'ecoulements diphasiques". In: *International Journal of Thermal Sciences* 43.3 (2004), pp. 265–276.

[56] E. W. Lemmon, M. McLinden, and D. Friend. "Thermophysical Properties of Fluid Systems". In: *NIST Chemistry WebBook*. Ed. by P. Linstrom and W. Mallard. Vol. 69. Gaithersburg MD: National Institute of Standards and Technology.

[57] E. W. Lemmon and R. Span. "Short fundamental equations of state for 20 industrial fluids". In: *Journal of Chemical & Engineering Data* 51.3 (2006), pp. 785–850.

[58] R. J. LeVeque. "High-resolution conservative algorithms for advection in incompressible flow". In: *SIAM Journal on Numerical Analysis* 33.2 (1996), pp. 627–665.

[59] R. Malladi, J. A. Sethian, and B. C. Vemuri. "Shape modeling with front propagation: A level set approach". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 17.2 (1995), pp. 158–175.

[60] E. Marchandise and J.-F. Remacle. "A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows". In: *Journal of Computational Physics* 219.2 (2006), pp. 780–800.

[61] R. Menikoff and B. J. Plohr. "The Riemann problem for fluid flow of real materials". In: *Reviews of modern physics* 61.1 (1989), pp. 75–130.

[62] C. Merkle and C. Rohde. "The sharp-interface approach for fluids with phase change: Riemann problems and ghost fluid techniques". In: *M2AN Math. Model. Numer. Anal.* 41.6 (2007), pp. 1089–1123.

[63] R. S. Miller, K. Harstad, and J. Bellan. "Evaluation of equilibrium and non-equilibrium evaporation models for many-droplet gas-liquid flow simulations". In: *International Journal of Multiphase Flow* 24.6 (1998), pp. 1025–1055.

[64] W. D. Monnery, W. Y. Svrcek, and M. A. Satyro. "Gaussian-like volume shifts for the Peng-Robinson equation of state". In: *Industrial & engineering chemistry research* 37.5 (1998), pp. 1663–1672.

[65] C.-D. Munz, M. Auweter-Kurtz, S. Fasoulas, A. Mirza, P. Ortwein, M. Pfeiffer, and T. Stindl. "Coupled Particle-In-Cell and Direct Simulation Monte Carlo method for simulating reactive plasma flows". In: *Comptes Rendus Mécanique* 342.10 (2014), pp. 662–670.

[66] J. Neudorfer, A. Stock, R. Schneider, S. Roller, and C.-D. Munz. "Efficient parallelization of a three-dimensional high-order particle-in-cell for the simulation of a 170 GHz gyrotron resonator". In: *IEEE Transactions on Plasma Science* 41 (2013), pp. 87–98.

[67] V.-T. Nguyen, J. Peraire, B. C. Khoo, and P.-O. Persson. "A discontinuous Galerkin front tracking method for two-phase flows with surface tension". In: *Computers & Fluids* 39.1 (2010), pp. 1–14.

[68] C. A. Niedermeier, H. Müller, M. Jarczyk, S. Hickel, N. A. Adams, and M. Pfitzner. "Large-eddy simulation of turbulent trans-and supercritical mixing". In: *AIAA paper* 2950 (2013), p. 2013.

[69] 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.

[70] S. Osher and C.-W. Shu. "High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations". In: *SIAM Journal on Numerical Analysis* 28.4 (1991), pp. 907–922.

[71] A. Péneloux, E. Rauzy, and R. Fréze. "A consistent correction for Redlich-Kwong-Soave volumes". In: *Fluid Phase Equilibria* 8.1 (1982), pp. 7–23.

[72] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. "A PDE-based fast local level set method". In: *Journal of Computational Physics* 155.2 (1999), pp. 410–438.

[73] D.-Y. Peng and D. B. Robinson. "A new two-constant equation of state". In: *Industrial & Engineering Chemistry Fundamentals* 15.1 (1976), pp. 59–64.

[74] P.-O. Persson and J. Peraire. "Sub-Cell Shock Capturing for Discontinuous Galerkin Methods". In: *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, 2006.

[75] B. E. Poling, J. M. Prausnitz, and J. P. O'Connel. *The properties of gases and liquids*. 5th edition. McGraw Hill Book Co., New York, NY, 2007.

[76] J. Pye et al. *FPROPS*. 2014.

[77] P. Reinke and G. Yadigaroglu. "Explosive vaporization of superheated liquids by boiling fronts". In: *International journal of multiphase flow* 27.9 (2001), pp. 1487–1516.

[78] P. L. Roe. "Approximate Riemann solvers, parameter vectors, and difference schemes". In: *Journal of computational physics* 43.2 (1981), pp. 357–372.

[79] C. Rohde and C. Zeiler. *A relaxation Riemann solver for compressible two-phase flow with phase transition and surface tension*. 2014.

[80] G. Russo and P. Smereka. "A remark on computing distance functions". In: *Journal of Computational Physics* 163.1 (2000), pp. 51–67.

[81] R. Saurel, F. Petitpas, and R. Abgrall. "Modelling phase transition in metastable liquids: application to cavitating and flashing flows". In: *Journal of Fluid Mechanics* 607.1 (2008), pp. 313–350.

[82] R. W. Schrage. *A theoretical study of interphase mass transfer*. Columbia University Press, 1953.

[83] J. Sethian and P. Smereka. "Level set methods for fluid interfaces". In: *Annual Review of Fluid Mechanics* 35.1 (2003), pp. 341–372.

[84] U. Setzmann and W. Wagner. "A new equation of state and tables of thermodynamic properties for methane covering the range from the melting line to 625 K at pressures up to 1000 MPa". In: *J. Phys. Chem. Ref. Data* 20.6 (1991), pp. 1061–1155.

[85] U. Setzmann and W. Wagner. "A new method for optimizing the structure of thermodynamic correlation equations". In: *International Journal of Thermophysics* 10.6 (1989), pp. 1103–1126.

[86] J. R. Simoes-Moreira and J. E. Shepherd. "Evaporation waves in superheated dodecane". In: *Journal of Fluid Mechanics* 382 (1999), pp. 63–86.

[87] A. Stock, J. Neudorfer, M. Riedlinger, G. Pirrung, G. Gassner, R. Schneider, S. Roller, and C.-D. Munz. "Three-dimensional numerical simulation of a 30-GHz gyrotron resonator with an explicit high-order discontinuous Galerkin based parallel particle-in-cell method". In: *IEEE Transactions on Plasma Science* 40 (2012), p. 1860.

[88] M. Sussman and E. Fatemi. "An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow". In: *SIAM Journal on Scientific Computing* 20.4 (1999), pp. 1165–1191.

[89] 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.

[90] I. Tanasawa. "Advances in condensation heat transfer". In: *Advances in heat transfer* 21 (1991), pp. 55–139.

[91] X. Tang and J. Gross. "Density functional theory for calculating surface tensions with a simple renormalization formalism for the critical point". In: *The Journal of Supercritical Fluids* 55.2 (2010), pp. 735–742.

[92] X. Tang and J. Gross. "Renormalization-group corrections to the perturbed-chain statistical associating fluid theory for binary mixtures". In: *Industrial & Engineering Chemistry Research* 49.19 (2010), pp. 9436–9444.

[93] S. Tanguy and A. Berlemont. "Application of a level set method for simulation of droplet collisions". In: *International journal of multiphase flow* 31.9 (2005), pp. 1015–1035.

[94] H. Terashima and G. Tryggvason. "A front-tracking/ghost-fluid method for fluid interfaces in compressible flows". In: *Journal of Computational Physics* 228.11 (2009), pp. 4012–4037.

[95] T. G. Theofanous, G. Li, T.-N. Dinh, and C. Chang. "Aerobreakup in disturbed subsonic and supersonic flow fields". In: *Journal of Fluid Mechanics* 593 (2007), pp. 131–170.

[96] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics.* Second. A practical introduction. Berlin: Springer-Verlag, 1999, pp. xx+624.

[97] E. F. Toro, M. Spruce, and W. Speares. "Restoration of the contact surface in the HLL-Riemann solver". In: *Shock waves* 4.1 (1994), pp. 25–34.

[98] S. O. Unverdi and G. Tryggvason. "A front-tracking method for viscous, incompressible, multi-fluid flows". In: *Journal of computational physics* 100.1 (1992), pp. 25–37.

[99] J. VonNeumann and R. D. Richtmyer. "A Method for the Numerical Calculation of Hydrodynamic Shocks". In: *Journal of Applied Physics* 21.3 (1950), pp. 232–237.

[100] J. D. van der Waals. "On the Continuity of the Gaseous and Liquid States". PhD thesis. Universiteit Leiden, 1873.

[101] W. Wagner, J. Cooper, A. Dittmann, J. Kijima, H.-J. Kretzschmar, A. Kruse, R. Mares, K. Oguchi, H. Sato, I. Stocker, et al. "The IAPWS industrial formulation 1997 for the thermodynamic properties of water and steam". In: *Journal of engineering for gas turbines and power* 122.1 (2000), pp. 150–184.

[102] C. Waibel. "Berechnung von Transportwiderstandskoeffizienten an Phasengrenzflächen mit der klassischen Dichtefunktionaltheorie". MA thesis. Universität Stuttgart, 2013.

[103] F. Weckenmann, B. Bork, E. Oldenhof, G. Lamanna, B. Weigand, B. Böhm, and A. Dreizler. "Single acetone droplets at supercritical pressure: droplet generation and characterization of PLIFP". In: *Zeitschrift für Physikalische Chemie International journal of research in physical chemistry and chemical physics* 225.11-12 (2011), pp. 1417–1431.

[104] A. Wood. "A Textbook of Sound. 1930". In: *G. Bell and Sons Ltd., London* (1930).

[105] J. Yan and S. Osher. "A local discontinuous Galerkin method for directly solving Hamilton–Jacobi equations". In: *Journal of Computational Physics* 230.1 (2011), pp. 232–244.

[106] Y.-X. Yu and G.-H. Gao. "Lennard–Jones chain model for self-diffusion of n-alkanes". In: *International journal of thermophysics* 21.1 (2000), pp. 57–70.

[107] S. T. Zalesak. "Fully multidimensional flux-corrected transport algorithms for fluids". In: *Journal of computational physics* 31.3 (1979), pp. 335–362.

[108] A. Zein, M. Hantke, and G. Warnecke. "Modeling phase transition for compressible two-phase flows applied to metastable liquids". In: *Journal of Computational Physics* 229.8 (2010), pp. 2964–2998.

List of Tables

List of Figures

Lebenslauf

1985	Geboren in Leonberg (Baden-Württemberg)
1992 – 1996	Grundschule, Renningen
1996 – 2005	Gymnasium Renningen
2005	Abitur
2005 – 2011	Diplomstudium Luft- und Raumfahrttechnik and der Universität Stuttgart, Vertiefungsrichtungen: Strömungslehre und Thermodynamik
2009 – 2010	Auslandsstudium an der KTH Stockholm, Schweden
2011	Diplomarbeit an der EPFL Lausanne, Schweiz
2011 – 2015	Wissenschaftlicher Mitarbeiter am Institut für Aerodynamik und Gasdynamik, Universität Stuttgart Thema der Promotion: Compressible multi-phase simulation at extreme conditions using a discontinuous Galerkin scheme”