Implicit Finite Element Schemes for Compressible Gas and Particle-Laden Gas Flows

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

Der Fakultät für Mathematik
der Technischen Universität Dortmund

vorgelegt von

Diplom-Mathematiker Marcel Gurris

Implicit Finite Element Schemes for Compressible Gas and Particle-Laden Gas Flows

Marcel Gurris

Gutachter:

Prof. Dr. D. Kuzmin (1. Gutachter, Betreuer)

Prof. Dr. S. Turek (2. Gutachter)

Dissertation eingereicht am: 15.12.2009

To my parents
Marion and Franz-Josef
and my love
Kim Verena

Acknowledgments

Many people contributed directly or indirectly to the preparation of this thesis and it is impossible to thank everyone individually. First of all, I have to express my most sincere gratitude to my first supervisor Professor Dmitri Kuzmin for his invaluable support and guidance of my graduate and undergraduate study during the recent years. Without his commitment, I would never have finished my thesis yet.

At the same time I am deeply indebted to my second supervisor Professor Stefan Turek for offering me the opportunity to work in the research project SFB 708, his academic advice, and sharing his experience in numerical mathematics with me.

I am grateful to all my colleagues at 'LS III'. You were a great help in all topics corresponding to our finite element toolbox FEATFLOW. I would like to thank Dr. Matthias Möller for fruitful discussions on topics related to the Euler equations. My sincerest appreciation goes to Dr. Shu-Ren Hysing for proofreading my manuscripts and giving valuable feedback. The assistance with mesh generation of Jens F. Acker was a great help for the simulation of flows in complex geometries with curved boundaries.

Moreover, I would like to thank the DFG (German Research Association, SFB 708, TP B7) for the financial support.

My deepest gratitude goes to the members of my family, in particular my parents Marion and Franz-Josef. Their unimaginable support and backing during my whole life and their encouragement and patience during the time of my education were important prerequisites along my way to advanced topics in computational fluid dynamics. At the same time my gratefulness goes to my partner in life Kim Verena for her commitment, patience, and unconditional kindness. I am very grateful for the love you offer to me.

Dortmund, December 2009

Marcel Gurris

Contents

0 Introduction

The dynamics of gases and multicomponent mixtures plays an important role in human life. Numerous applications in natural sciences and engineering deal with gas flows and mixtures of gases with solid particles or liquid droplets. Such flows are subjects of active research since the knowledge of the flow properties is beneficial for the progress in natural science and engineering. Traditionally, physical phenomena like flow dynamics are investigated by means of experiments and measurements. This approach is time consuming, particularly if a certain optimization process requires multiple and expensive experiments. During the last decades there has been an increasing interest in computational fluid dynamics (CFD), which is nowadays a competitive and quite often a preferable approach to investigating fluid flows and related transport phenomena. Simulation of physical processes usually involves three basic steps:

1. Mathematical modeling of the phenomenon.
2. Numerical solution of the governing equations.
3. Visualization and analysis of the computed results.

The quality of numerical simulations strongly depends on the computational resources, the theoretical understanding of physical processes, and the design of suitable mathematical models governing the flow.

In the present thesis compressible flows are of interest. Compressible gas flows occur for example in aerospace engineering applications, like flows around planes, space shuttles, or rockets. Furthermore, explosions generate waves at large pressure and density gradients and compressible flow is also present in power plants. The examples above are merely a small number of applications involving compressible flow. Quite often the flow is polluted by solid or liquid material, which may be the result of chemical reactions. Some examples are volcanic eruptions, dust explosions (in coal mines), the outlet of solid rocket motors, or condensation in a (nuclear) power plant. Other important industrial applications of compressible particle-laden gas flows are the variety of spraying technologies which have been developed in recent years to create coatings. In particular, the possibility to model and predict arc spraying and high velocity flame spraying (HVOF) processes are challenging long term goals of the research presented in this study.

The compressible Navier-Stokes equations and the Euler equations as a simplified counterpart are widely accepted in the CFD community. They have proven to be rather accurate models of compressible flows. In contrast, the modeling of gas-solid or gas-liquid mixtures is much more complicated and the proper treatment has sometimes been controversially discussed in the literature. In principle, there are two approaches to modeling the dynamics of the particulate phase in a compressible gas-solid mixture.

These approaches differ in the description of the disperse phase. In a so-called Euler-Lagrange model, the particles are treated as moving objects immersed in a continuous gas phase. The trajectory of each particle is computed separately. If the number of particles is too large, the simulation process becomes very expensive in terms of memory and CPU time requirements. Moreover, it is difficult to incorporate collisions of particles and turbulence effects in such a model. Hence, the Lagrangian tracking of particles is only feasible for very dilute flows. In a so-called Euler-Euler model, both phases are treated as interpenetrating continua sharing each control volume. The hydrodynamic behavior of the particulate phase is governed by macroscopic conservation laws for the mass, momentum, and energy. This formulation is preferable for a large number of particles. Moreover, a nonzero pressure term can be incorporated to model collisions of particles in the case of a dense suspension. The Euler-Euler approach is adopted in the present thesis.

0.1 Research Goals

The primary goal of this thesis is to develop new implicit and Newton-like finite element solvers for inviscid pure gas and particle-laden gas flows. We focus our attention on stationary solutions. Both topics are related in some sense since a single-phase gas flow arises as a special case for a vanishing particulate phase. Although this study mainly focuses on numerical methods, the specification of an appropriate mathematical model for particle-laden gas flows is also an important aim. Macroscopic multiphase flow models can be postulated or derived from the single-phase equations by a suitable averaging procedure. The result is a system of balance equations for both phases, which is called two-fluid model. Special emphasis is laid on the analysis and critical evaluation of the involved interface exchange terms and their adjustment to particle-laden gas flows.

Macroscopic two-fluid models consist of an inhomogeneous hyperbolic set of equations. Typical standard discretizations, including those based on finite element methods, tend to produce numerical oscillations if they are applied to hyperbolic equations (or systems). This is unacceptable if the compressible Euler equations or the two-fluid model governing particle-laden gas flows are considered. From the physical point of view quantities like pressure and density have to be positive. This property can however be violated by the presence of undershoots and overshoots. To prevent the birth and growth of wiggles and preserve the physical properties of the solution, a suitable stabilization term should be added to the discretized equations.

Modern high-resolution schemes combine the advantages of low-order and high-order approximations. Roughly speaking, a diffusive low-order method is applied at steep fronts and local extrema, where linear high-order schemes produce spurious wiggles. On the other hand, a high-order method is employed in smooth regions, where a low-order approximation would be less accurate. The result is a nonlinear convex combination of the two methods. The share of each scheme is fitted to the local smoothness of the solution using limiters. This approach was introduced by Boris and Book in their pioneering work [9] on the flux-corrected-transport algorithm (FCT). Originally, this algorithm was also based on finite difference and finite volume discretizations. It was generalized to the finite element framework by the group of Löhner [53, 54, 55].

The FCT algorithm is known to be very accurate for highly time-dependent transport equations [38, 46]. On the other hand, it is not to be recommended for steady-state computations because the amount of numerical diffusion is proportional to the time step. Moreover, severe convergence problems are observed in the steady-state limit. Therefore, flux correction schemes of FCT type are of little value for the numerical study to be performed in this thesis. Nonlinear high-resolution schemes based on the total variation diminishing (TVD) criterion are also designed by blending high- and low-order approximations. Such schemes are independent of the time step and ideally suited for solving stationary problems. However, standard TVD limiters are not applicable to finite element discretizations and they are practically restricted to 1D. Using a slightly weaker constraint, known as the local extremum diminishing (LED) property, Kuzmin et al. extended the essentially 1D TVD schemes to multidimensional finite element methods in the framework of so-called algebraic flux correction methods [46, 47, 43, 44]. Algebraic flux correction schemes of TVD type are parameter-free and non-oscillatory by construction, at least for scalar transport equations. Based on the standard Galerkin finite element discretization a low-order method is derived by adding a suitably defined diffusion operator, which reduces the order of approximation to one. In a third step the order of approximation is increased by insertion of a limited amount of antidiffusion, which is controlled by TVD-like flux limiters. The result is a nonlinear high-resolution scheme. At its birth the methodology was designed for scalar hyperbolic equations and later extended to the Euler equations [47]. The limiters developed by Kuzmin et al. belong to the most accurate stabilization techniques for continuous finite elements [38]. Hence, they are adopted in this thesis. In the present work we focus on the algebraic flux correction technique and its extension to the two-fluid model.

Most numerical studies of multicomponent/multiphase flows are concerned with incompressible gas-liquid mixtures. A very popular class of this family are incompressible bubbly flows [42, 89]. In contrast to the significant advance in numerical methods for dispersed gas-liquid flows, publications on macroscopic multiphase flow FEM models for compressible gas-solid mixtures are relatively scarce. The governing equations of compressible multiphase flows can for example be solved by a pressure correction scheme like SIMPLE (semi-implicit method for pressure-linked equations) with strong coupling by the interphase slip/partial elimination algorithm (IPSA/PEA). Pressure correction schemes are more related to incompressible or weakly compressible transient flows, while their applicability to highly compressible problems is questionable. Most publications addressing compressible multiphase flows deal with one-dimensional (explicit) finite volume approximations, although there are a few publications containing 2D results [78, 79, 33, 10, 39, 67, 34, 70, 66]. The author is aware of just one numerical study involving finite elements [88]. In this publication, a FEM-based flux FCT algorithm was applied to a dusty shock tube problem. The present study focuses on the development of fully coupled finite element schemes, which do not rely on decoupling strategies like SIMPLE. There is strong numerical evidence that the schemes presented here are unconditionally stable. This is an important advantage in stationary computations on unstructured meshes due to the varying mesh sizes throughout the domain. Due to that fact, there is a high demand for implicit compressible flow solvers.

At the current stage, they are restricted to (semi-) implicit discretizations [15, 60, 18] or Newton-like methods [60, 93, 62, 7] for single-phase gas flows. A particularly challenging task is to overcome severe convergence problems associated with the use of non-differentiable limiters in steady-state computations. It is not unusual that the nonlinear residual decreases merely by a few orders of magnitude, after which convergence stalls [95]. In light of the above, discontinuous Galerkin (DG) methods or finite volume schemes with semi-differentiable limiters are commonly employed.

It is a major goal of the present study to design a semi-implicit or Newton-like finite element solver for the Euler equations and the two-fluid model. Special emphasis is laid on the computation of stationary solutions in a wide range of Mach numbers since the low Mach number regime is another cause for the lack of convergence in compressible flow simulations. The proposed algorithm converges even in the case of non-smooth limiters and oscillatory correction factors, which are usually blamed for convergence problems. We primarily focus our attention on the nonlinear solver and employ globally refined grids instead of adaptive ones for that reason.

The choice of boundary conditions is a very important topic in compressible CFD. Since waves may move in different directions, the same boundary part may represent an inlet for some waves and an outlet for other waves. An inappropriate choice of numerical boundary conditions may result in artificial reflections of waves at the boundaries and hamper convergence. To maintain unconditional stability and convergence of implicit solvers, boundary conditions must be integrated into the preconditioner. An explicit treatment of boundary conditions would degrade the overall performance of the algorithm [15] and render the implicit approach useless. In spite of their utmost importance, issues related to the numerical implementation of boundary conditions for hyperbolic systems are rarely discussed in the literature. Typically, they rely on a diagonalization of the Jacobian, which is impossible for the two-fluid equations. Implicit boundary conditions in the finite volume framework were implemented either in the strong sense [93] in terms of additional algebraic equations or alternatively the boundary fluxes were directly overwritten by the imposed boundary conditions [62]. In continuous finite element schemes [87] or DG methods [18, 15] Neumann-type flux boundary conditions were implemented by directly changing the boundary integrals without use of boundary Riemann solvers. An important goal of this thesis is to develop Neumann-type boundary conditions in an implicit way incorporating the solution of the boundary Riemann problem to increase robustness. The proposed algorithm is applicable to large and infinite CFL numbers and does not inhibit convergence to steady-state.

Another objective of this thesis is to generalize the developed implicit techniques to compressible particle-laden gas flows and to be able to achieve stationary solutions. These tasks are very challenging since additional nonlinearities due to large and stiff algebraic coupling terms arise and must be discretized in a proper way. Algebraic source terms are typically incorporated into the computational model making use of operator splitting [78, 79, 70, 82]. Numerical results show that operator splitting is inappropriate for steady-state simulations and subject to restrictive time step constraints or even does not allow the solution to approach steady-state. In the presented research thesis operator splitting is avoided by using a fully coupled Newton-like approach. The algorithm developed in this thesis features most properties of the single-phase gas solver, although it does not converge at the same rate, due to the additional and non-differentiable interfacial coupling. To the author's best knowledge, there is no publication on implicit schemes or boundary conditions for particle-laden gas flows in multi-dimensions. These concepts are developed in this thesis and shown to be crucial for a satisfactory convergence to steady-state.

0.2 Outline of the Thesis

The present study is mainly concerned with the design of implicit methods for compressible gas as well as particle-laden gas flows. Practical algorithms and remarks on implementation of the proposed methods are given. The thesis is subdivided into three parts.

First of all, the computational models to be examined in this thesis are presented. The compressible Navier-Stokes equations are formulated and simplified to the compressible Euler equations under the assumption of inviscid flows. The Euler equations serve as the computational model of pure gas flows. In the second chapter of the first part, the single-phase equations are averaged over a spatial control volume to derive the equations of motion governing multiphase flows. Moreover, constitutive laws, some simplifications valid for particle-laden gas flows, and the analysis of relevant interfacial forces are presented.

The second part of this study is the most important one since it contains the mathematical details and computational algorithms. In chapter three, a self-contained introduction to the algebraic flux correction procedure and TVD-like high-resolution finite element schemes is given. The presentation of this material is largely based on [45, 46]. Furthermore, a fully multidimensional node-based flux limiting strategy [44] is presented in the context of a scalar hyperbolic equation.

In chapter four, the concepts reported in chapter three are generalized to hyperbolic systems of equations and fitted to the set of conservation laws governing the particulate phase. These equations exhibit the same structure as the Euler system for the gas phase but there is no pressure, and the interphase transfer terms have opposite signs.

The presence of a nonzero pressure gradient in the gas phase equations is responsible for the hyperbolicity and occurance of distinct wave speeds. The mathematical properties of the gas phase equations are fundamentally different from those of the pressureless transport model for the particles. Therefore, the two PDE systems require completely different stabilization mechanisms, as illustrated in the fifth chapter.

Chapter six deals with the discretization of source terms and the interfacial coupling induced by the source terms of drag force and heat exchange. A robust and simple finite element discretization of source terms is proposed and operator splitting techniques to deal with the two-way coupling via the drag force and heat exchange are reported.

In chapter seven, a Neumann-type weak form of flux boundary conditions is proposed to maintain unconditional stability of the numerical algorithm. Special attention is paid to the numerical treatment of solid wall boundaries.

The last chapter of the second part is devoted to the solution of the arising nonlinear systems. Pseudo time stepping and Newton-like techniques based on the backward Euler scheme are presented and combined with a linearization of the nonlinear fluxes and source terms. A semi-implicit linearized pseudo time stepping scheme is employed. Our numerical results indicate that the scheme is unconditionally stable and convergent for arbitrary CFL numbers. The semi-implicit pseudo time stepping scheme transforms to a Newton-like method for $CFL = \infty$ and incorporates a low-order preconditioner, which also is proposed in this section.

Numerical results are presented in the third part. In chapter nine the convergence and performance of the developed implicit algorithm and boundary conditions are established for the Euler equations for several test cases. In chapter ten, benchmark computations with the inviscid two-fluid model are presented to give evidence for the performance of the nonlinear solver and boundary conditions at large or even infinite CFL numbers. Moreover, the results are analyzed for validation purposes.

Finally, conclusions are drawn and an outlook is given in chapter 11.

Part I

Modeling of Compressible Gas and Particle-Laden Gas Flows

1 The Euler Equations: Modeling of a Pure Compressible Gas

Compressible gas flows are usually modeled by the physical principles of conservation of mass, momentum, and energy, where the last two conserved quantities satisfy Newton's second law and the first law of thermodynamics, respectively. Based on these assumptions, one can derive the compressible Navier-Stokes equations

$$\begin{array}{rcl} \partial_t \rho + \nabla \cdot (\rho \mathbf{v}) & = & 0 \\ \partial_t (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + \mathbf{T}) & = & \mathbf{F}_b \\ \partial_t (\rho E) + \nabla \cdot (\rho \mathbf{v} E + \mathbf{v} \cdot \mathbf{T} + \mathbf{q}) & = & \mathbf{v} \cdot \mathbf{F}_b + Q, \end{array} \tag{1.1}$$

where ρ, $\mathbf{v} = (u, v)^T$, E denote the density, velocity, and total energy of the gas. For Newtonian fluids, the stress tensor $\mathbf{T}$ is given by

$$\mathbf{T} = \mathbf{I}P - \mu(\nabla \mathbf{v} + (\nabla \mathbf{v})^T) + \frac{2}{3}\mu \mathbf{I} \nabla \cdot \mathbf{v} \tag{1.2}$$

where P is the thermodynamic pressure and μ is the dynamic viscosity. The heat flux $\mathbf{q}$ is related to the gradient of the temperature T by Fourier's law of heat conduction

$$\mathbf{q} = -\kappa \nabla T. \tag{1.3}$$

Furthermore, Q is an external heat source, and the body forces are represented by $\mathbf{F}_b$.

Neglecting all viscous terms, heat conduction, heat sources, and body forces the Navier-Stokes equations simplify to the Euler equations

$$\begin{array}{rcl} \partial_t \rho + \nabla \cdot (\rho \mathbf{v}) & = & 0 \\ \partial_t (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I}P) & = & 0 \\ \partial_t (\rho E) + \nabla \cdot (\mathbf{v}(\rho E + P)) & = & 0. \end{array} \tag{1.4}$$

At high speeds the Euler equations are a suitable approximation of the Navier-Stokes equations. Equations (1.4) are the governing equations which will be used in this thesis to investigate single-phase gas flows.

1.1 Mathematical Properties

The Euler equations (1.4) can be rewritten in vectorial form as

$$\partial_t \underbrace{\begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{bmatrix}}_{=U} + \partial_x \underbrace{\begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho u v \\ u(\rho E + P) \end{bmatrix}}_{=F^{(x)}} + \partial_y \underbrace{\begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + P \\ v(\rho E + P) \end{bmatrix}}_{=F^{(y)}} = 0. \tag{1.5}$$

The vector of conservative variables is abbreviated by U, and the flux vectors $F^{(x)}$ and $F^{(y)}$ are associated with the coordinate directions. These equations may also be written in a more compact form

$$\partial_t \underbrace{\begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{bmatrix}}_{=U} + \nabla \cdot \underbrace{\begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + \mathbf{I}P \\ \mathbf{v}(\rho E + P) \end{bmatrix}}_{=\mathbf{F}} = 0, \tag{1.6}$$

where

$$\mathbf{F} = \left(F^{(x)}, F^{(y)} \right) \tag{1.7}$$

is the flux tensor used to simplify notation. Moreover, the velocity vector is given by $\mathbf{v} = (u, v)^T$.

1.1.1 Definitions and Equations of State

To close the system of governing equations an equation of state is required. In this study we assume the ideal gas law

$$P = \rho R T \tag{1.8}$$

with the specific gas constant R. This yields the constitutive pressure law [32]

$$P = (\gamma - 1)\rho \left(E - \frac{|\mathbf{v}|^2}{2} \right) \tag{1.9}$$

with a constant ratio of specific heats γ. For all computations reported later $\gamma = 1.4$ is prescribed. The Mach number

$$M = \frac{|\mathbf{v}|}{c} \tag{1.10}$$

is the ratio of the norm of the gas velocity and the speed of sound

$$c = \sqrt{\frac{\gamma P}{\rho}} \tag{1.11}$$

in the gas. In accordance with the definition of the Mach number, the flow regime can be characterized as subsonic $(M < 1)$, transonic $(M \approx 1)$, supersonic $(M > 1)$, and hypersonic $(M \gg 1)$. The total energy E is linked to the internal energy e by

$$E = e + \frac{|\mathbf{v}|^2}{2}. \tag{1.12}$$

Moreover, the specific enthalpy h and the total enthalpy H are defined by

$$h = e + \frac{P}{\rho}, \qquad H = E + \frac{P}{\rho}. \tag{1.13}$$

1.1.2 Hyperbolicity

Equations (1.6) are given in conservative form. For our purposes, the quasi-linear form is also of interest. One can apply the chain rule to (1.6) and derive

$$\partial_t U + \frac{\partial F^{(x)}}{\partial U}\partial_x U + \frac{\partial F^{(y)}}{\partial U}\partial_y U = 0 \quad \Leftrightarrow \quad \partial_t U + \left(\frac{\partial F^{(x)}}{\partial U}, \frac{\partial F^{(y)}}{\partial U}\right) \cdot \nabla U = 0. \tag{1.14}$$

To simplify notation the Jacobian tensor

$$\mathbf{A} = \left(\frac{\partial F^{(x)}}{\partial U}, \frac{\partial F^{(y)}}{\partial U}\right) \tag{1.15}$$

is defined, which transforms the quasi-linear form of the Euler equations to

$$\partial_t U + \mathbf{A} \cdot \nabla U = 0. \tag{1.16}$$

The governing equations are hyperbolic if the Jacobians $\frac{\partial F^{(x)}}{\partial U}$ and $\frac{\partial F^{(y)}}{\partial U}$ for both coordinate directions x and y are diagonalizable with real eigenvalues. The Jacobians are given by

$$\frac{\partial F^{(x)}}{\partial U} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ b_2 u^2 + b_1 v^2 & (3-\gamma)u & (1-\gamma)v & \gamma - 1 \\ -uv & v & u & 0 \\ b_1(u^3 + v^2 u) - Hu & H - (\gamma-1)u^2 & (1-\gamma)uv & \gamma u \end{bmatrix} \tag{1.17}$$

and

$$\frac{\partial F^{(y)}}{\partial U} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ b_2 v^2 + b_1 u^2 & (1-\gamma)u & (3-\gamma)v & \gamma - 1 \\ b_1(u^2 v + v^3) - Hv & (1-\gamma)uv & H - (\gamma-1)v^2 & \gamma v \end{bmatrix} \tag{1.18}$$

with

$$b_1 = \frac{\gamma - 1}{2}, \qquad b_2 = \frac{\gamma - 3}{2}. \tag{1.19}$$

A spectral analysis shows that the eigenvalues of both matrices are

$$\lambda^{(x)} = \{u - c, u, u, u + c\} \quad \text{and} \quad \lambda^{(y)} = \{v - c, v, v, v + c\}, \tag{1.20}$$

respectively. A complete set of eigenvectors exists. Eigenvectors, eigenvalues and Jacobians are reported in [76]. Therefore, the Euler equations are a hyperbolic set of coupled conservation laws, which enables the application of hyperbolic solvers based on characteristic variables.

1.1.3 Homogeneity Property

It is easy to verify [22] that the homogeneity relation

$$F^{(d)}(\theta U) = \theta F^{(d)}(U), \quad d = x, y \tag{1.21}$$

holds for the Euler equations with any constant θ. By the chain rule, it implies

$$F^{(d)} = \frac{\partial F^{(d)}}{\partial U} U. \tag{1.22}$$

This latter form of the homogeneity property is a useful tool for discretization purposes.

1.2 Nondimensionalization

In a numerical simulation it may be more convenient to rewrite the equations in dimensionless form to reduce round-off errors, equalize the scales and compute a solution independent of a particular system of units. A compressible flow satisfying the Euler equations can be characterized by the Mach number M and the ratio of specific heats γ up to some scaling factors. To obtain a dimensionless form we define the following dimensionless quantities

$$\begin{array}{lll} \rho^* = \dfrac{\rho}{\rho_\infty} & \mathbf{v}^* = \dfrac{\mathbf{v}}{c_\infty} & P^* = \dfrac{P}{\rho_\infty c_\infty^2} \\ E^* = \dfrac{E}{c_\infty^2} & x^* = \dfrac{x}{L} & t^* = \dfrac{t c_\infty}{L}. \end{array}$$

In the expressions above, ρ_∞, c_∞, c_∞^2, and $\rho_\infty c_\infty^2$ are scaling factors, while L, $\frac{L}{c_\infty}$ are the length and time scale of the underlying flow problem.

Substitution into (1.6) yields the desired dimensionless form of the Euler equations

$$\begin{aligned} \partial_{t^*}\rho^* + \nabla^* \cdot (\rho^* \mathbf{v}^*) &= 0 \\ \partial_{t^*}(\rho^* \mathbf{v}^*) + \nabla^* \cdot (\rho^* \mathbf{v}^* \otimes \mathbf{v}^* + \mathbf{I}P^*) &= 0 \\ \partial_{t^*}(\rho^* E^*) + \nabla^* \cdot (\mathbf{v}^*(\rho^* E^* + P^*)) &= 0. \end{aligned} \tag{1.23}$$

In the remainder of this thesis, we will drop the asterisks. For the pure gas simulations we assign the free stream values given in table 1.1 unless mentioned otherwise.

Variable	Free Stream Value
ρ^*	1
u^*	M_∞
v^*	0
P^*	$\frac{1}{\gamma}$
E^*	$\frac{M_\infty^2}{2} + \frac{1}{\gamma(\gamma-1)}$

Table 1.1: Free stream values for single-phase gas flow

2 The Two-Fluid Model: Modeling of a Particle-Laden Gas

In this section, we briefly examine the modeling of a compressible gas flow containing very small (solid) particles. We focus our attention on the development of a macroscopic model (Euler-Euler approach). It is generally accepted that single-phase gas flows can be modeled by macroscopic equations of mass, momentum and energy conservation. The particulate (or dispersed) phase is also supposed to admit a continuous description. As a natural assumption, the single-phase equations are also valid for multi-phase flows except at the interfaces separating the different components. A mixture of two or more different materials can be interpreted as a flow, which is subdivided into single-phase regions by infinitesimal thin interfaces. Each of these subdomains is governed by the single-phase conservation laws. The interfaces are moving boundaries in mathematical sense. Several physical and mathematical difficulties arise due to the presence of interfaces. In particular, interface exchange between different materials is up to now not completely understood. The existing models are based on empirical correlations that are not universally applicable. Due to limited computing resources it is impossible to locate the interfaces at the microscopic scale if the dispersed phase is distributed over the whole domain. It is necessary to transform the microscopic equations into their macroscopic counterparts. In this case, macroscopic equations are derived by using suitable averaging procedures so that they are related to the microscopic ones in a mathematical sense. They can therefore be interpreted as some kind of generalization.

Various techniques are reported in the literature and typical candidates are volume, time, statistical, and ensemble averagings, or combinations of the former families. A survey can be found in the textbooks of Drew and Passmann [16] and Ishii and Hibiki [35], where the latter text mainly focuses on time averaging. The different techniques typically yield similar results. The averaged equations are widely accepted and applied in a large number of publications. Saurel and Abgrall [82] derive a quite general hyperbolic non-conservative model, which is applicable to dense and dilute flows alike. Städke [91] used averaged hyperbolic equations for the numerical simulation of the interaction of water and steam. Such macroscopic so-called two-fluid models were successfully applied to incompressible bubbly flows [42, 89]. Computational models related to particle-laden gas flows can be found in [70, 66, 88, 79, 34, 78, 67, 39, 10, 33]. The models employed for the different flow regimes widely agree in modeling aspects, except in the treatment of interface exchange. Although the interaction between the involved materials at the interfaces has a strong influence on the flow behavior and the simulation results, it remains controversial in the literature (e. g. lift forces) and depends on the materials under consideration. Consequently, the interface exchange should be modeled carefully. Modeling of compressible particle-laden gas flows requires several assumptions and simplifications. In this work we use the following assumptions:

- No chemical reactions and no change of aggregate states appear.
- Both particles and gas are distributed over the whole domain.
- The inviscid equations of mass, momentum, and energy conservation are valid in the interior of each phase.
- The gas pressure satisfies the ideal gas equation of state (1.9).
- There is no considerable amount of particle collisions and the particles do not interact with each other.
- The material density of the particles is constant. That means, the particulate phase is incompressible.
- Dilute flow conditions, i. e. the volume occupied by the particles is small.
- The material density ratio $\frac{\rho_g}{\rho_p} \ll 1$ is small.
- The particles are solid, spherical, of uniform size and their diameter is small compared to the length scale.
- The influences of curvature is negligible and surface tension does not play a role for solid particles.
- There are interfacial momentum and heat transfer, but no external momentum and energy sources.

The present section summarizes the application of the volume averaging process to indicate the origin of the different terms. Additionally, the modeling of interfacial properties of particle-laden gas flows is discussed under the assumptions given above.

In the following the index k refers to either the gas phase g or the particulate phase p. The interface quantities are denoted by the index *int*. Hence, each phase satisfies a system of microscopic conservation laws that can be written as

$$\begin{aligned} \partial_t \rho_k + \nabla \cdot (\rho_k \mathbf{v}_k) &= 0 \\ \partial_t (\rho_k \mathbf{v}_k) + \nabla \cdot (\rho_k \mathbf{v}_k \otimes \mathbf{v}_k + \mathbf{T}_k) &= 0 \\ \partial_t (\rho_k E_k) + \nabla \cdot (\mathbf{v}_k \rho_k E_k + \mathbf{v}_k \cdot \mathbf{T}_k + \mathbf{q}_k) &= 0, \end{aligned} \tag{2.1}$$

where external body forces (like gravity) and heat sources are neglected. The stress tensor is denoted by $\mathbf{T}_k$. Equations (2.1) are valid for both phases exclusively in their interior. At the interfaces density, velocity, and energy are discontinuous and the jumps are constrained by [35, 16]

$$\sum_{k=g,p} \begin{bmatrix} \rho_k (\mathbf{v}_k - \mathbf{v}_{int}) \\ \rho_k \mathbf{v}_k \otimes (\mathbf{v}_k - \mathbf{v}_{int}) + \mathbf{T}_k \\ \rho_k E_k (\mathbf{v}_k - \mathbf{v}_{int}) + \mathbf{v}_k \cdot \mathbf{T}_k + \mathbf{q}_k \end{bmatrix} \cdot \mathbf{n}_k = \begin{bmatrix} 0 \\ 0 \\ -\frac{d_i e_{surf}}{dt} - e_{surf} \nabla \cdot \mathbf{v}_{int} \end{bmatrix} \tag{2.2}$$

since surface tension and the influence of curvature are neglected. Surface tension is only important if the particles are deformable. This is not the case for solid particles considered in this thesis. The vector $\mathbf{n}_k$ denotes the unit normal to the interface directed to the interior of phase k, $\mathbf{v}_{int}$ is the velocity of the interface, and e_{surf} is the surface internal energy density. The material derivative associated with the interface is referred

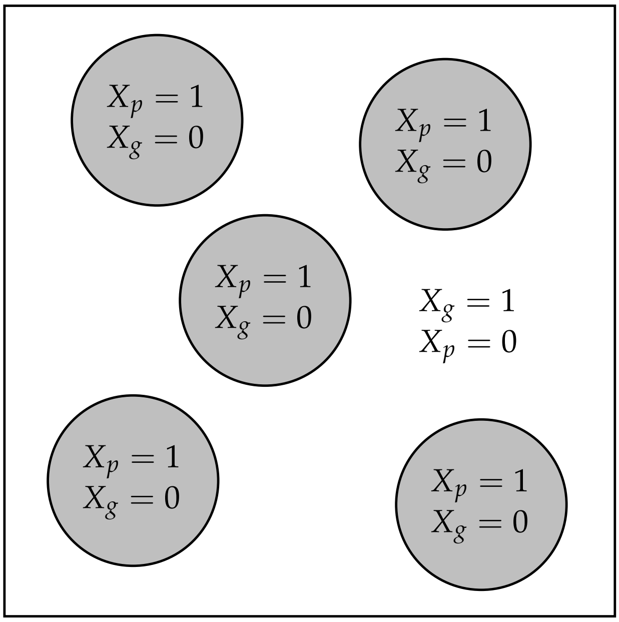


Figure 2.1: Distribution of the indicator functions X_g and X_p in a typical control volume V

to as $\frac{d_i}{dt}$. It is easy to verify that

$$\mathbf{n}_g = -\mathbf{n}_p \tag{2.3}$$

holds. To derive macroscopic equations, which are valid in the whole domain, we will multiply equations (2.1) by the phase indicator function

$$X_k(\mathbf{x}, t) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is in phase } k \text{ at time } t \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

and average the microscopic equations. The distribution of the indicator functions is sketched in figure 2.1.

The volume average of the function f over a small control volume V is defined by

$$\langle f \rangle = \frac{1}{V} \int_V f \, dV. \tag{2.5}$$

It follows that

$$\alpha_k = \langle X_k \rangle = \frac{V_k}{V} \tag{2.6}$$

is the fraction of the control volume V occupied by the phase k. Furthermore, we define the phasic average of a quantity ϕ by [16]

$$\tilde{\phi}_k = \frac{\langle X_k \phi \rangle}{\alpha_k}. \tag{2.7}$$

In the following sections the generalization of the microscopic conservation laws (2.1) by volume averaging is discussed.

2.1 Averaging of the Continuity Equation

A multiplication of the first equation in (2.1) by the indicator function and integration over V yields

$$\begin{aligned} 0 &= \frac{1}{V}\int_V X_k(\partial_t \rho_k + \nabla\cdot(\rho_k \mathbf{v}_k))\,dV \\ &= \frac{1}{V}\int_V \partial_t(X_k\rho_k) + \nabla\cdot(X_k\rho_k\mathbf{v}_k) - \rho_k(\partial_t X_k + \mathbf{v}_k\cdot\nabla X_k)\,dV. \end{aligned} \tag{2.8}$$

In [16] Drew and Passman have proved

$$\partial_t X_k + \mathbf{v}_{int}\cdot\nabla X_k = 0. \tag{2.9}$$

This enables us to rearrange equation (2.8) to

$$\begin{aligned} 0 &= \frac{1}{V}\int_V \partial_t(X_k\rho_k) + \nabla\cdot(X_k\rho_k\mathbf{v}_k) - \rho_k(\partial_t X_k + \mathbf{v}_k\nabla\cdot X_K) \\ &\quad + \rho_k(\partial_t X_k + \mathbf{v}_{int}\cdot\nabla X_k)\,dV \\ &= \frac{1}{V}\int_V \partial_t(X_k\rho_k) + \nabla\cdot(X_k\rho_k\mathbf{v}_k) - \rho_k(\mathbf{v}_k - \mathbf{v}_{int})\cdot\nabla X_k\,dV \end{aligned} \tag{2.10}$$

or equivalently

$$\frac{1}{V}\int_V \partial_t(X_k\rho_k) + \nabla\cdot(X_k\rho_k\mathbf{v}_k)\,dV = \frac{1}{V}\int_V \rho_k(\mathbf{v}_k - \mathbf{v}_{int})\cdot\nabla X_k\,dV. \tag{2.11}$$

The gradient ∇X_k (defined in a weak sense) is equal to zero everywhere except at the interface, where ∇X_k is parallel to the inner normal pointing into the interior of phase k

$$\mathbf{n}_k = \frac{\nabla X_k}{|\nabla X_k|}. \tag{2.12}$$

Hence,

$$\Gamma_k = \frac{1}{V}\int_V \rho_k(\mathbf{v}_k - \mathbf{v}_{int})\cdot\nabla X_k\,dV \tag{2.13}$$

is the rate of interphase mass transfer. Since chemical reactions are beyond the scope of this work, we assume $\Gamma_k = 0$. The control volume V is fixed in space and time and we can therefore move the derivative operators in front of the integrals and substitute (2.7) into (2.11) to obtain [16]

$$\partial_t(\alpha_k\tilde{\rho}_k) + \nabla\cdot(\alpha_k\tilde{\rho}_k\tilde{\mathbf{v}}_k) = \Gamma_k = 0. \tag{2.14}$$

In the equation above $\tilde{\mathbf{v}}_k$ is defined by

$$\tilde{\mathbf{v}}_k = \frac{\widetilde{\rho_k\mathbf{v}_k}}{\tilde{\rho}_k}. \tag{2.15}$$

The averaged momentum and energy equations are derived in a very similar way.

2.2 Averaging of the Momentum Equations

We multiply the single-phase momentum equations (second equation in (2.1)) by X_k and average over a small control volume V. A transformation similar to (2.8)-(2.11) yields

$$\begin{aligned}\frac{1}{V}\int_V \partial_t(X_k\rho_k\mathbf{v}_k) + \nabla\cdot(X_k(\rho_k\mathbf{v}_k\otimes\mathbf{v}_k + \mathbf{T}_k))\,dV &= \frac{1}{V}\int_V \mathbf{T}_k\cdot\nabla X_k\,dV \\ &+ \frac{1}{V}\int_V \rho_k\mathbf{v}_k\otimes(\mathbf{v}_k - \mathbf{v}_{int})\cdot\nabla X_k\,dV.\end{aligned} \tag{2.16}$$

The second term on the right hand side of equation (2.16) represents the momentum flux due to convection of mass across the interface with an exchange velocity $\mathbf{v}_k^{ex}$. The rate of mass transfer Γ_k is given by (2.13). Since $\Gamma_k = 0$, we deduce that

$$\Gamma_k\mathbf{v}_k^{ex} = \frac{1}{V}\int_V \rho_k\mathbf{v}_k\otimes(\mathbf{v}_k - \mathbf{v}_{int})\cdot\nabla X_k\,dV = 0. \tag{2.17}$$

The contribution of the first term to the right hand side of (2.16) is modeled by ([16], p. 128)

$$\begin{aligned}\frac{1}{V}\int_V \mathbf{T}_k\cdot\nabla X_k\,dV &= \mathbf{T}_k^{int}\cdot\frac{1}{V}\int_V \nabla X_k\,dV + F_k^{int} \\ &= \mathbf{T}_k^{int}\cdot\nabla\alpha_k + \mathbf{F}_k^{int},\end{aligned} \tag{2.18}$$

where $\mathbf{F}_k^{int}$ and $\mathbf{T}_k^{int}$ are the interfacial forces and stresses, respectively. The main mechanisms of interphase momentum transfer are viscous drag, lift, and virtual mass effects (see below). Taking equations (2.7) and (2.18) into consideration, we can transform (2.16) into

$$\partial_t(\alpha_k\tilde{\rho}_k\tilde{\mathbf{v}}_k) + \nabla\cdot(\alpha_k\tilde{\rho}_k\tilde{\mathbf{v}}_k\otimes\tilde{\mathbf{v}}_k + \alpha_k\tilde{\mathbf{T}}_k) = \mathbf{T}_k^{int}\cdot\nabla\alpha_k + \mathbf{F}_k^{int}. \tag{2.19}$$

2.3 Averaging of the Energy Equation

Again, the derivation process starts with the application of the above averaging procedure to the single-phase energy equation (third equation in (2.1)) to obtain

$$\begin{aligned}\frac{1}{V}\int_V \partial_t(X_k\rho_k E_k) + \nabla\cdot(X_k(\rho_k E_k\mathbf{v}_k + \mathbf{v}_k\cdot\mathbf{T}_k + \mathbf{q}_k))\,dV &= \frac{1}{V}\int_V (\mathbf{v}_k\cdot\mathbf{T}_k)\cdot\nabla X_k\,dV \\ &+ \frac{1}{V}\int_V \rho_k E_k(\mathbf{v}_k - \mathbf{v}_{int})\cdot\nabla X_k\,dV + \frac{1}{V}\int_V \mathbf{q}_k\cdot\nabla X_k\,dV.\end{aligned} \tag{2.20}$$

The first term on the right hand side of equation (2.20) corresponds to the work of interfacial stress. This integral is commonly expressed in terms of $\mathbf{v}_{int}$ and $\mathbf{T}_k^{int}$ as follows (compare to (2.18)):

$$\frac{1}{V}\int_V (\mathbf{v}_k\cdot\mathbf{T}_k)\cdot\nabla X_k\,dV = (\mathbf{v}_{int}\cdot\mathbf{T}_k^{int})\cdot\nabla\alpha_k + \mathbf{v}_{int}\cdot\mathbf{F}_k^{int}. \tag{2.21}$$

The second term represents the energy flux associated with the convective mass transfer across the interface. Since $\Gamma_k = 0$, this term vanishes

$$\Gamma_k E_k^{ex} = \frac{1}{V} \int_V \rho_k E_k (\mathbf{v}_k - \mathbf{v}_{int}) \cdot \nabla X_k \, dV = 0, \tag{2.22}$$

where E_k^{ex} represents the exchange energy. The third term on the right hand side of (2.20) describes the interphase heat transfer due to other mechanisms. It is modeled by

$$\alpha_k q_k^{int} = \frac{1}{V} \int_V \mathbf{q}_k \cdot \nabla X_k, \tag{2.23}$$

where q_k^{int} is the rate of interphase heat transfer. Hence, the volume-averaged energy equation reduces to

$$\partial_t(\alpha_k \tilde{\rho}_k \tilde{E}_k) + \nabla \cdot (\alpha_k (\tilde{\rho}_k \tilde{\mathbf{v}}_k \tilde{E}_k + \tilde{\mathbf{v}}_k \cdot \tilde{\mathbf{T}}_k)) = (\mathbf{v}_{int} \cdot \mathbf{T}_k^{int}) \cdot \nabla \alpha_k + \mathbf{v}_{int} \cdot \mathbf{F}_k^{int} + \alpha_k q_k^{int}. \tag{2.24}$$

In the remainder of this thesis, the tilde, which indicates the averaging process, is dropped to simplify notation. Taking the transformations described above in this chapter into consideration the averaged equations can be summarized by

$$\begin{aligned} \partial_t(\alpha_k \rho_k) + \nabla \cdot (\alpha_k \rho_k \mathbf{v}_k) &= 0 \\ \partial_t(\alpha_k \rho_k \mathbf{v}_k) + \nabla \cdot (\alpha_k \rho_k \mathbf{v}_k \otimes \mathbf{v}_k + \alpha_k \mathbf{T}_k) &= \mathbf{T}_k^{int} \cdot \nabla \alpha_k + \mathbf{F}_k^{int} \\ \partial_t(\alpha_k \rho_k E_k) + \nabla \cdot (\alpha_k (\rho_k \mathbf{v}_k E_k + \mathbf{v}_k \cdot \mathbf{T}_k)) &= (\mathbf{v}_{int} \cdot \mathbf{T}_k^{int}) \cdot \nabla \alpha_k + \mathbf{v}_{int} \cdot \mathbf{F}_k^{int} + \alpha_k q_k^{int}. \end{aligned} \tag{2.25}$$

2.4 The Computational Model

Equations (2.25) constitute a coupled set of eight nonlinear conservation laws in 2D. The left hand side of this system contains convective terms for both phases, while the interface exchange terms are located on the right hand side. Both phases are coupled at the interface by the interfacial stress and velocity, heat exchange, interfacial forces, and the volume fractions. Equations (2.25) are widely accepted as the general form of governing equations of two-phase flows if external heat sources, viscosity, and body forces are neglected. Primarily the interface terms $\mathbf{v}_{int}$, $\mathbf{T}_k^{int}$, q_k^{int}, and $\mathbf{F}_k^{int}$ require further modeling. Furthermore, there is a considerable debate on the modeling of the pressure related to the particulate phase. In this thesis only terms, which are clearly understood from the modeling point of view and which significantly influence simulation results, are incorporated into the computational model. The following sections are concerned with the modeling of important material interactions.

2.4.1 Remarks on Conservation

In a single-phase flow, no mass, momentum, or energy is either generated or destroyed in the interior of the domain. This physical fact carries over to mixtures of two or more different materials and should be accounted for by the computational model. We add the continuity equations of both phases to derive a corresponding conservation equation for the mixture density

$$\partial_t \sum_k (\alpha_k \rho_k) + \nabla \cdot \sum_k (\alpha_k \rho_k \mathbf{v}_k) = 0. \tag{2.26}$$

Since the left hand sides of all equations are in conservative form and there are no external sources of momentum and energy, all non-conservative effects are due to the interfacial momentum and energy exchange corresponding to the right hand sides or the boundary fluxes. Assuming that the interfacial forces, heat exchanges, and stresses satisfy

$$\sum_k \mathbf{F}_k^{int} = 0, \quad \sum_k q_k^{int} = 0, \quad \sum_k \mathbf{T}_k^{int} \cdot \nabla \alpha_k = 0, \quad \sum_k (\mathbf{v}_{int} \cdot \mathbf{T}_k^{int}) \cdot \nabla \alpha_k = 0, \tag{2.27}$$

an addition of the momentum and energy equations yields conservation laws for the mixture momentum

$$\partial_t \sum_k (\alpha_k \rho_k \mathbf{v}_k) + \nabla \cdot \sum_k (\alpha_k \rho_k \mathbf{v}_k \otimes \mathbf{v}_k + \alpha_k \mathbf{T}_k) = 0 \tag{2.28}$$

and energy

$$\partial_t \sum_k (\alpha_k \rho_k E_k) + \nabla \cdot \sum_k (\alpha_k (\rho_k \mathbf{v}_k E_k + \mathbf{v}_k \cdot \mathbf{T}_k)) = 0. \tag{2.29}$$

Due to this fact, the mixture momentum and energy (and also mass in the same manner) are conserved, although each phase may lose or gain momentum and energy by the interfacial exchange. Moreover, the saturation condition

$$\sum_k \alpha_k = \sum_k \frac{V_k}{V} = 1 \tag{2.30}$$

holds.

To close the computational model the terms $\mathbf{T}_k$,$\mathbf{T}_k^{int}$, $\mathbf{v}_{int}$, q_k^{int}, and $\mathbf{F}_k^{int}$ must be specified.

2.4.2 Stress Tensor

We suppose that the single-phase stress tensor consists of a pressure part and a viscous part (compare to 1.2). Similar to the modeling of a pure gas flow, we neglect the visous contribution and define the pressure tensor as

$$\mathbf{T}_k = \mathbf{I} P_k, \tag{2.31}$$

where P_k denotes the pressure of phase k.

In the present work the gas pressure P_g is given by the ideal gas equation of state (1.9), while the pressure of the particulate phase requires further modeling. It is much more difficult to interpret the pressure of a solid. In a gas-solid mixture, the pressure (or stress) describes the forces acting between particles due to collisions, or shear stress ([16], p.223). We assume spherical particles and a dilute flow satisfying

$$\alpha_p < \alpha_{crit} < 1, \tag{2.32}$$

or rather

$$\alpha_p \sim \frac{\rho_g}{\rho_p}. \tag{2.33}$$

Yang [96] proposes a simple model for the pressure of the particulate phase

$$P_p = \begin{cases} 0 & \text{if} \quad \alpha_p < \alpha_{crit} \\ P_{solid}(\alpha_p) & \text{if} \quad \alpha_p \geq \alpha_{crit} \end{cases}. \tag{2.34}$$

If relation (2.32) is valid, particle collisions can be neglected. Only if the volumetric concentration of the particles is close to the packing limit (the particles form a solid) a considerable amount of collisions appears and a pressure P_{solid} must be defined. The positive pressure bounds the effective particle density $\alpha_p \rho_p$ from above and prevents delta shocks. Usually P_{solid} is a function that grows to infinity if the effective density of the particles approaches the packing limit [68]. Since in the present work (2.32) is valid, the results are independent of P_{solid} and

$$P_p \equiv 0 \tag{2.35}$$

holds. This zero-pressure model is generally applied to dilute gas-particle flows [70, 88, 10, 79, 34, 67, 39, 66, 33]. It is better suited than the assumption of

$$P_p = P_g, \tag{2.36}$$

which is also a valid choice for the two-fluid model since surface tension does not paly a role [16].

Due to the lack of pressure, delta shocks may in principle appear in the particulate phase. They are excluded by the assumption of dilute conditions. In addition, the interface exchange terms also play an important role in preventing such unphysical phenomena. The gas pressure is linked to the velocity in such a way that the gas density remains bounded. Since the velocity of particles is also related to the gas velocity by the magnitude of interfacial drag, the gas pressure influences the velocity of the particles in some sense. This may be a reason why delta shocks are not observed in the particulate phase.

2.4.3 Interfacial Stress

The interfacial stress tensors are defined as the sums of viscous and pressure contributions (compare 2.4.2), while the viscosity is neglected. Hence, the interfacial stress tensors reduce to interfacial pressure contributions

$$\mathbf{T}_k^{int} = \mathbf{I} P_k^{int}. \tag{2.37}$$

Since conservation is a basic requirement, the third equation of (2.27) must be satisfied. This is equivalent to

$$\left(\mathbf{T}_g^{int} - \mathbf{T}_p^{int}\right) \cdot \nabla \alpha_k = 0 \tag{2.38}$$

and obviously satisfied by

$$\mathbf{T}_g^{int} = \mathbf{T}_{int} = \mathbf{T}_p^{int}. \tag{2.39}$$

Hence, we define

$$P_g^{int} = P_{int} = P_p^{int}. \tag{2.40}$$

This choice is also used by Drew and Passmann [16]. The interfacial pressure is modeled by several authors in different ways. A natural definition of the interfacial pressure contribution is proposed by Saurel and Abgrall [82]. They postulate the interfacial pressure as the mixture pressure

$$P_{int} = \sum_k \alpha_k P_k. \tag{2.41}$$

Most authors who deal with dilute flows, treat the dispersed phase as incompressible and assume pressure equilibrium. Consequently, they associate all pressures with the gas pressure

$$P_p = P_{int} = P_g. \tag{2.42}$$

This choice is sensible if the curvature can be neglected [16]. On the other hand, it may give rise to mathematical difficulties related to the lack of hyperbolicity of the resulting model. Other authors, like Powers [72], state that

$$P_{int} = 0 \tag{2.43}$$

or add a correction term to the gas pressure and define the interfacial pressure, e.g., by [78]

$$P_{int} = P_g + \delta(\alpha_g). \tag{2.44}$$

This correction is usually chosen to render the model hyperbolic, although there is still some physical background. The correction term models the influence of the particles on the gas pressure. In contrast, there are other definitions reported in the literature [64], which are primarily designed to enforce hyperbolicity. Such unjustifiable choices are beyond the scope of this work and will not be considered further.

Obviously, there is a large discrepancy in the treatment of the interfacial stresses. In the present work we examine compressible flows, which may contain shocks and give rise to weak solutions. Consequently, the non-conservative terms $\mathbf{T}_k^{int} \cdot \nabla \alpha_k$ and $(\mathbf{v}_{int} \cdot \mathbf{T}_k^{int}) \cdot \nabla \alpha_k$ on the right hand sides of the second and third equation of (2.25) must be dropped since the non-conservative gradient is not defined in the weak sense. This can also be justified by physical arguments. If one phase is finely dispersed throughout the other, the pressure of the dispersed component does not differ much from its value at the interface [16]. Hence, one can state

$$P_p^{int} = P_p = 0 \tag{2.45}$$

due to the assumption of a dilute flow and we define

$$P_{int} = 0. \tag{2.46}$$

It has become common practice in the CFD community to neglect the interfacial pressure in simulations of dilute particle-laden gas flows. Pressureless equations for the particulate phase [70, 88, 79, 34, 67, 39, 10, 66, 33] are usually formulated instead.

2.4.4 Interfacial Velocity

The determination of the averaged velocity $\mathbf{v}_{int}$ of the interface requires additional modeling. In [82] Saurel and Abgrall define the interfacial velocity using the center of mass velocity as

$$\mathbf{v}_{int} = \frac{\sum_k \alpha_k \rho_k \mathbf{v}_k}{\sum_k \alpha_k \rho_k}, \tag{2.47}$$

while Delhaye and Bouré [12] postulate the center of volume velocity

$$\mathbf{v}_{int} = \sum_k \alpha_k \mathbf{v}_k. \tag{2.48}$$

Although there is no rigorous theoretical justification they are both natural assumptions for flows with deformable interfaces. In the special case of particle-laden gas flows the modeling of the interfacial velocity is less complicated. It can simply be interpreted as the velocity of the particulate phase

$$\mathbf{v}_{int} = \mathbf{v}_p. \tag{2.49}$$

In most references regarding particle laden flows, the velocity of the particulate phase is recommended as the interfacial velocity [70, 88, 78, 81, 79, 66, 67]. This choice is justified by physical arguments for the particle-laden gas flows under consideration. Let us switch to the microscopic point of view and consider a spherical particle in a surrounding gas flow. We associate the interface with the surface of the particle, which does not change its shape. Hence, the gas velocity does not directly influence the shape of the particle and the position of its surface. Furthermore, the velocity of the surface is equal to the velocity of the particle and the interface mimics the same behavior. Consequently, (2.49) is a sensible assumption based on the physics of both materials.

We derive a different justification for (2.49) by the first component of the jump condition (2.2). The particle does not change its shape and the gas cannot penetrate through the surface of the particle. Hence, a no-penetration condition

$$\mathbf{v}_g \cdot \mathbf{n}_p = 0 \tag{2.50}$$

holds. This reduces the above mentioned jump condition to

$$\left(\mathbf{v}_p - (1 - \frac{\rho_g}{\rho_p})\mathbf{v}_{int}\right) \cdot \mathbf{n}_p = 0. \tag{2.51}$$

Since $\frac{\rho_g}{\rho_p} \ll 1$, we approximate

$$\left(\mathbf{v}_p - \mathbf{v}_{int}\right) \cdot \mathbf{n}_p = 0. \tag{2.52}$$

This equation is satisfied by (2.49) or if $\mathbf{v}_p - \mathbf{v}_{int}$ is tangential to the surface. The latter case is less sensible and therefore not considered further. It turns out that equation (2.49) is a meaningful definition satisfying the jump condition (2.2) under the current assumptions.

The most important interface coupling terms in compressible particle-laden gas flows are the algebraic source terms due to the interfacial forces and heat exchange. Constitutive laws for these terms are given in the next sections.

2.4.5 Interfacial Forces

Using Newton's law, the momentum equation for a single particle p with mass m_p reads

$$m_p \frac{d\mathbf{v}_p}{dt} = \mathbf{F}_D + \mathbf{F}_{VM} + \mathbf{F}_{BAS} + \mathbf{F}_L + \mathbf{F}_{OTHER}. \tag{2.53}$$

In other words, the change of momentum equals the sum of all forces acting on the particle. These are in particular the interfacial forces of drag $\mathbf{F}_D$, virtual mass $\mathbf{F}_{VM}$, Basset $\mathbf{F}_{BAS}$, and lift $\mathbf{F}_L$. The remaining forces are represented by $\mathbf{F}_{OTHER}$ (e. g. gravity) but are not considered further. It is impossible to account for all forces in the computational model and they are of different importance in a particular flow regime. Consequently, physically important ones should be included into the computational model, while the forces with small or negligible magnitude should be excluded. In the next sections the modeling of forces experienced by the particles is addressed.

2.4.5.1 Drag Force

The drag force [16, 68]

$$\mathbf{F}_D = \frac{3}{4}\frac{\rho_g}{d\rho_p} m_p C_D |\mathbf{v}_g - \mathbf{v}_p|(\mathbf{v}_g - \mathbf{v}_p) \tag{2.54}$$

determines the drag on the particle due to the slip velocity based on the drag coefficient C_D. A particle of diameter d moving with a velocity higher than that of the surrounding fluid will be decelerated by the slower fluid and vice versa. This force, acting on the particle is modeled by the drag force and it is important in any dispersed flow with significant slip velocities. Hence, it is included into our model and turns out to be the most important interfacial force. The amount of drag depends on the drag coefficient C_D. This dimensionless quantity is defined in this thesis by the (widely accepted) standard equation

$$C_D = \begin{cases} \frac{24}{Re}(1 + 0.15Re^{0.687}) & \text{if} \quad Re < 1000 \\ 0.44 & \text{if} \quad Re \geq 1000 \end{cases}. \tag{2.55}$$

It is valid for spherical particles and given as a function of the particle Reynolds number *Re* [8]

$$Re = \frac{\rho_g d |\mathbf{v}_g - \mathbf{v}_p|}{\mu}. \tag{2.56}$$

Here μ denotes the microscopic dynamic viscosity of the gas and d is the particle diameter. Both μ and d are assumed to be constant. Sommerfeld [90] argues that the standard drag coefficient is a valid choice only for steady flow problems and proposes a different definition

$$C_D = 112Re^{-0.98} \tag{2.57}$$

for time-dependent situations. Note that both configurations are based on empirical correlations and alternative choices exist. The interested reader is referred to the habilitation thesis of Sokolichin [89], where the author examines different configurations for incompressible bubbly flows. Since the present study is primarily concerned with steady-state computations, we adopt formula (2.55).

2.4.5.2 Virtual Mass Force

The correlations of the drag force are based on a constant slip velocity. If the particle accelerates or decelerates relatively to the gas motion, then a wake of the surrounding gas is also accelerated or decelerated by an additional force. This force is called virtual mass force and is modeled by [68]

$$\mathbf{F}_{VM} = C_{VM}\frac{\rho_g}{\rho_p} m_p \left((\partial_t \mathbf{v}_g + \mathbf{v}_g \cdot \nabla \mathbf{v}_g) - (\partial_t \mathbf{v}_p + \mathbf{v}_p \cdot \nabla \mathbf{v}_p) \right). \tag{2.58}$$

The influence on the numerical results is sometimes controversial at least for bubbly flows, although the existence of a virtual mass force is physically justified. It is in particular difficult to determine the dimensionless coefficient C_{VM}. For bubbly flows, most authors assume

$$C_{VM} = 0.5, \tag{2.59}$$

although there are many other choices reported in the literature, cf. [89] and the references therein. Crowe et al. [11] propose to define C_{VM} in terms of an acceleration parameter, which approaches $C_{VM} = 1$ in the limit of constant relative velocity. The choice of the dimensionless coefficient can be postulated, derived from analytical solutions to simplified configurations, or deduced from experimental data. Since the measurement of virtual mass effects is up to now a difficult task or even impossible at high speeds, the configurations originating from measurements and their application to compressible flows are doubtful due to the difficult choice of C_{VM}. Nevertheless, the virtual mass force makes the numerical solution of the governing equations much more challenging. In our case the virtual mass force can be neglected due to the small material density ratio $\frac{\rho_g}{\rho_p}$. On the other hand, some authors employ the virtual mass force to render their two-fluid models hyperbolic for numerical reasons. We will see below that this is not necessary.

2.4.5.3 Basset Force

The Basset force is given by [11]

$$\mathbf{F}_{BAS} = \frac{3}{2} d^2 \sqrt{\pi \rho_g \mu} \int_0^t \frac{\frac{d}{d\tau}(\mathbf{v}_g - \mathbf{v}_p)}{\sqrt{t-\tau}}\, d\tau. \tag{2.60}$$

It takes into account the acceleration history up to the current time level t. Obviously, this force incorporates unsteady effects and it is therefore restricted to transient flows. The Basset force is excluded from our computational model for that reason. In a time-dependent simulation of particle-laden gas flows, the magnitude of the Basset force is also insignificant due to the small particle diameter and gas viscosity. In this case, the convective terms are much larger than the Basset term, which is scaled by $d^2\sqrt{\pi \rho_g \mu}$.

2.4.5.4 Lift Force

The lift force accounts for effects transversal to the flow direction and arises from the rotation of a spherical particle. It can be subdivided into two parts. At first there is a force originating by particle rotation (Saffmann force) due to a surrounding fluid in shearing motion. At the same time the effects arising by a non-uniform pressure distribution on the surface of the rotating particle are a consequence of the Magnus force. The sum of Saffmann and Magnus force is modeled by the lift force [16]

$$\mathbf{F}_L = C_L m_p \frac{\rho_g}{\rho_p} (\mathbf{v}_p - \mathbf{v}_g) \times (\nabla \times \mathbf{v}_g). \tag{2.61}$$

Up to now the modeling of the lift force is not completely understood. Although the influence on the numerical results may be large [89] (depending on the flow regime and material properties) the existing models are highly controversial. Moreover, the lift force is sometimes included into computational models to fit the numerical solutions to experimental data. Hence, simulations computed without taking the lift force into account are typically more reliable than their counterparts incorporating transversal effects.

2.4.5.5 Pertinence of Interfacial Forces

The relevance of a force under consideration is determined by three basic requirements:

- The existence is physically and experimentally justified.
- The modeling is clearly understood and there are universal correlations for the involved coefficients.

- The force influences the computational results significantly.

Otherwise, the force should be excluded from the computational model. To assess the significance of the forces above we consider equation (2.53) and divide it by m_p

$$\begin{aligned}\frac{d\mathbf{v}_p}{dt} &= \frac{3}{4}\frac{\rho_g}{d\rho_p}C_D|\mathbf{v}_g - \mathbf{v}_p|(\mathbf{v}_g - \mathbf{v}_p) + C_L\frac{\rho_g}{\rho_p}(\mathbf{v}_p - \mathbf{v}_g) \times (\nabla \times \mathbf{v}_g) \\ &+ C_{VM}\frac{\rho_g}{\rho_p}\left((\partial_t\mathbf{v}_g + \mathbf{v}_g \cdot \nabla\mathbf{v}_g) - (\partial_t\mathbf{v}_p + \mathbf{v}_p \cdot \nabla\mathbf{v}_p)\right).\end{aligned} \tag{2.62}$$

In the equation above, F_{BAS} and F_{OTHER} are neglected. It becomes obvious that the contributions originating by lift force and virtual mass force are scaled by $\frac{\rho_g}{\rho_p}$. In a gas flow, laden with solid particles (e. g. iron or glass), the mass densities usually differ by several orders of magnitude. Due to the very small material density ratio $\frac{\rho_g}{\rho_p}$ neither lift nor virtual mass force significantly contribute to the velocity of the particle. And since they violate the basic requirements,we remove them from the computational model.

The insignificance of lift effects can also be justified by a more physical argumentation. The Magnus force is related to rotation, which is primarily caused by particle collisions or collisions with a wall. Since particle collisions are neglected and frictionless conditions are assumed, there is no significant rotation. Furthermore, effects of shearing motion at the surface of a particle are primarily important for relatively large particles. Hence, the Saffmann force is unimportant in the case of very small particles. Consequently, the lift force is also of small magnitude.

In contrast, the contribution of the drag force depends on $\frac{\rho_g}{d\rho_p} \gg 1$. This makes the contribution of viscous drag important for particle-laden gas flows. Considering the arguments above, the drag force is the most important force that belongs to the computational model, while the others are negligible in our situation. Consequently, we only incorporate the drag force into our computational model. Note that the three basic requirements are satisfied.

The insignificance of the virtual mass force compared to viscous drag can be further evidenced by a model problem with parameters closely related to the computations to be reported in part III. Consider a single particle, which is injected into a gas flow of constant velocity $\mathbf{v}_g = (100, 0, 0)^T \frac{m}{s}$. The mass densities are $\rho_p = 4000 \frac{kg}{m^3}$ and $\rho_g = 1 \frac{kg}{m^3}$ and the dynamic viscosity of the gas is set to $\mu = 10^{-5} \frac{kg}{m \cdot s}$. At time $T = 0$ the velocity of the particle is $v_p(0) = 0 \frac{m}{s}$ and it starts its motion due to the interaction with the gas. We assume that the particle does not influence the gas velocity and suppose a one-way coupling. This problem can be approximated by a 1D configuration. If we neglect lift effects, the momentum equation (2.62) transforms into

$$\frac{dv_p}{dt} = \frac{3}{4\left(1 + C_{VM}\frac{\rho_g}{\rho_p}\right)}\frac{\rho_g}{d\rho_p}C_D\left|v_g - v_p\right|\left(v_g - v_p\right). \tag{2.63}$$

The first term on the right hand side indicates that the virtual mass force will only slightly reduce the acceleration of the particle by the gas due to the drag force. Numerical solutions to this ordinary differential equation for two particle diameters $d = 0.1mm, 1mm$ and virtual mass coefficients $C_{VM} = 0, 1, 10$ were computed. Since the results for the different C_{VM} are almost equal, figure 2.2 (a) shows the particle velocities for $C_{VM} = 0$. The particle is quite large and the magnitude of the drag force is underestimated for that reason. On the other hand, the virtual mass coefficient $C_{VM} = 10$ is relatively large compared with data reported in the literature, which overestimates the influence of virtual mass effects. In spite of this discrepancy we can deduce from figure 2.2(b) and (c) that the relative velocity differences of the solutions with and without virtual mass effects are below 0.3% even in the case of $C_{VM} = 10$. In other words, the velocity is hardly affected by the virtual mass force. On the other hand, we observe in figure 2.2 (a) that the velocity is significantly influenced by the particle diameter, which proves the importance of viscous drag.

In macroscopic conservation laws, the interfacial forces acting on a single particle are replaced by their averaged counterparts defined as body forces in a control volume sense. We will use the same notation for the volume-averaged counterpart of each force. Since $\mathbf{F}_D$ is the force of primary importance, the rate of interfacial momentum transfer is given by

$$\mathbf{F}_p^{int} = \mathbf{F}_D = \frac{3}{4}\alpha_p \frac{\rho_g}{d} C_D |\mathbf{v}_g - \mathbf{v}_p|(\mathbf{v}_g - \mathbf{v}_p), \qquad \mathbf{F}_g^{int} = -\mathbf{F}_p^{int}. \tag{2.64}$$

Last but not least the closure of the governing equations requires a constitutive equation for the interfacial heat transfer.

2.4.6 Interfacial Heat Transfer

The rate of interfacial heat transfer is proportional to the temperature difference [40]

$$q_p^{int} = Q_T = \frac{Nu 6\kappa}{d^2}\alpha_p (T_g - T_p), \qquad q_g^{int} = -q_p^{int} \tag{2.65}$$

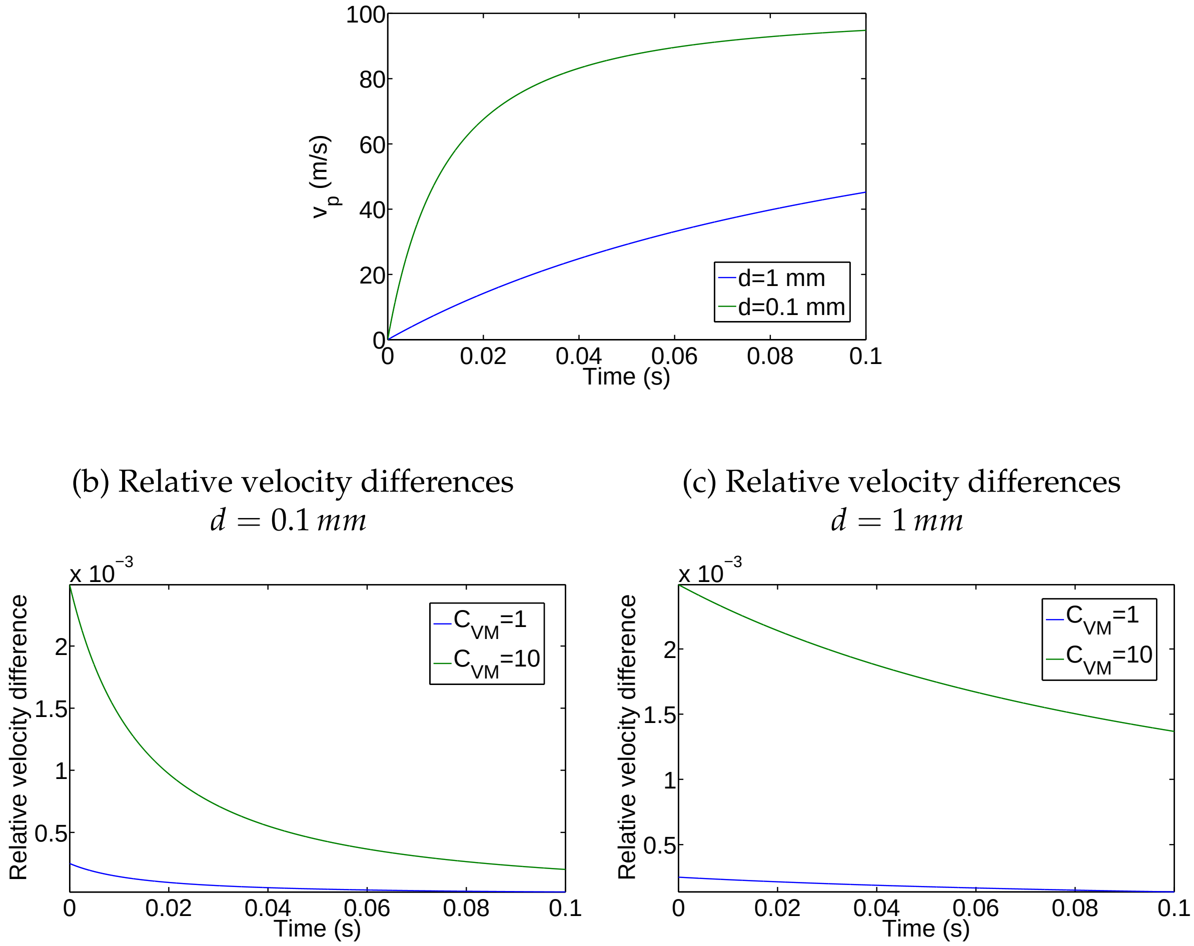
where the Nusselt number Nu is a function of the Prandtl number Pr

$$Nu = 2 + 0.65 Re^{\frac{1}{2}} Pr^{\frac{1}{3}}, \qquad Pr = \frac{c_{pg}\mu}{\kappa}. \tag{2.66}$$

The thermal conductivity κ, heat capacity at constant pressure c_{pg}, and (microscopic) dynamic viscosity μ of the gas are assumed to be constant. Furthermore, the temperature

$$T_k = \frac{1}{c_{vk}} \left(E_k - \frac{1}{2}|\mathbf{v}_k|^2 \right) \tag{2.67}$$

of the phase k is defined as a function of the total energy and the velocity, while the heat capacity at constant volume c_{vk} is supposed to be a constant.



(a) Particle velocity $d = 0.1\,mm, 1\,mm$

(b) Relative velocity differences $d = 0.1\,mm$

(c) Relative velocity differences $d = 1\,mm$

Figure 2.2: Sensitivity analysis of drag and virtual mass

2.4.7 Summary of the Equations

In the present chapter, we have derived an inviscid two-fluid model that describes the macroscopic behavior of compressible particle-laden gas flows under certain assumptions. In summary, the system of equations to be dealt with in this thesis reads

$$\begin{array}{rcl} \partial_t(\alpha_g\rho_g) + \nabla\cdot(\alpha_g\rho_g\mathbf{v}_g) &=& 0 \\ \partial_t(\alpha_g\rho_g\mathbf{v}_g) + \nabla\cdot(\alpha_g\rho_g\mathbf{v}_g\otimes\mathbf{v}_g + \alpha_g\mathbf{I}P) &=& -\mathbf{F}_D \\ \partial_t(\alpha_g\rho_g E_g) + \nabla\cdot\left(\alpha_g\mathbf{v}_g\left(\rho_g E_g + P\right)\right) &=& -\mathbf{v}_p\cdot\mathbf{F}_D - Q_T \\ \partial_t(\alpha_p\rho_p) + \nabla\cdot(\alpha_p\rho_p\mathbf{v}_p) &=& 0 \\ \partial_t(\alpha_p\rho_p\mathbf{v}_p) + \nabla\cdot(\alpha_p\rho_p\mathbf{v}_p\otimes\mathbf{v}_p) &=& \mathbf{F}_D \\ \partial_t(\alpha_p\rho_p E_p) + \nabla\cdot(\alpha_p\rho_p\mathbf{v}_p E_p) &=& \mathbf{v}_p\cdot\mathbf{F}_D + Q_T, \end{array} \tag{2.68}$$

where

$$\mathbf{F}_D = \frac{3}{4}\alpha_p\frac{\rho_g}{d}C_D|\mathbf{v}_g - \mathbf{v}_p|(\mathbf{v}_g - \mathbf{v}_p) \tag{2.69}$$

is the drag force and

$$Q_T = \frac{Nu6\kappa}{d^2}\alpha_p(T_g - T_p) \tag{2.70}$$

is the rate of interfacial heat transfer. Moreover, the pressure is modeled by the gas pressure in terms of the ideal gas equation of state

$$P = (\gamma-1)\rho_g\left(E_g - \frac{|\mathbf{v}_g|^2}{2}\right). \tag{2.71}$$

The pressure of the particulate phase is neglected due to the arguments above. Constitutive equations

$$T_k = \frac{1}{c_{vk}}\left(E_k - \frac{1}{2}|\mathbf{v}_k|^2\right) \tag{2.72}$$

link the temperature of both phases to the velocity and total energy. Note that the effective density $\alpha_p\rho_p$ is variable, although the particulate phase is incompressible with a constant material density ρ_p.

Part II

Numerical Methods for Compressible Gas and Particle-Laden Gas Flows

3 Scalar Conservation Laws

This chapter is devoted to the numerical solution of scalar hyperbolic equations of the form

$$\partial_t u + \nabla \cdot \mathbf{f}(u) = 0, \tag{3.1}$$

where $u : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ is a conserved quantity and $\mathbf{f} : \mathbb{R}^2 \to \mathbb{R}^2$ denotes a nonlinear flux function. The above equation must be supplied with suitable initial and boundary conditions. Numerical methods are firstly designed for the 1D counterpart of (3.1)

$$\partial_t u + \partial_x f(u) = 0 \tag{3.2}$$

and then generalized to systems of hyperbolic equations and multidimensions. For the sake of simplicity, suppose that equation (3.2) holds in an unbounded space-time domain, where the initial data has compact support. Due to the finite wave speed of hyperbolic equations the solution keeps the compact support at every finite time level.

The solution u to equation (3.1) is supposed to be smooth (differentiable). It is possible to multiply this hyperbolic equation by a smooth test function ω with compact support and integrate it in space and time. This results in a weak formulation

$$\int_0^\infty \int_{\mathbb{R}^2} u \partial_t \omega + \mathbf{f}(u) \cdot \nabla \omega \, d\mathbf{x} \, dt = 0, \tag{3.3}$$

where the derivative operators are shifted to the (smooth) test function. In contrast to (3.1), the weak formulation does not require the functions u and $\mathbf{f}$ to be smooth.

A measurable function u is called a weak solution of equation (3.1) if it satisfies (3.3). A smooth solution that satisfies (3.1) in the strong sense is also a solution of (3.3) and, therefore, a weak solution. Interestingly enough, a smooth initial solution may lose this property during the time evolution and become a weak solution. For example, the solution to the inviscid Burgers equation exhibits this behavior [51].

Nonlinear hyperbolic equations may admit an infinite number of weak solutions of which the majority is non-physical. A physically correct solution can be characterized by an entropy condition [51] or defined as the solution of the parabolic perturbed problem

$$\partial_t u_\varepsilon + \nabla \cdot \mathbf{f}(u_\varepsilon) = \varepsilon \triangle u_\varepsilon \tag{3.4}$$

in the limit of 'vanishing viscosity' $\varepsilon \to 0$.

It is a well known fact that discretizations of hyperbolic equations tend to produce spurious undershoots and overshoots if insufficient care is taken. The following section provides guidelines for the construction of non-oscillatory numerical schemes that preserve the important physical properties of the solution.

3.1 Physical and Numerical Criteria

Numerical methods for hyperbolic equations usually consist of a combination of a time stepping and a space discretization scheme. The spatial discretization yields a system of ordinary differential equations, which is solved by the time stepping scheme.

The analysis of numerical methods is usually based on the concepts of consistency, stability and convergence. The famous equivalence theorem of Lax [52] states that

$$\text{consistency and stability} \qquad \Leftrightarrow \qquad \text{convergence}$$

for a linear scheme. This simplifies the convergence analysis to the question of consistency and stability of which the proof of consistency quite often is easier than the proof of stability. In particular, the proof of stability for nonlinear approximations is a very challenging and perhaps also impossible task.

Although this thesis primarily is devoted to stationary solutions, time marching methods are also of interest since the time step Δt (or pseudo time step) serves as an under-relaxation parameter and may help a solution to converge to steady-state. Therefore, consistency and stability on the fully discrete level (for a given combination of time marching and space discretization schemes) are the main requirements.

Assume that the governing equation is discretized in space and time (fully discrete level). A numerical scheme is consistent with the governing equation if the local discretization error tends to zero

$$\tau_{h,\Delta t} = \|\mathcal{L}_{h,\Delta t}(u) - \mathcal{L}(u)\| \rightarrow 0, \tag{3.5}$$

for time step $\Delta t \rightarrow 0$ and mesh refinement $h \rightarrow 0$. In the equation above $\mathcal{L}$ denotes the continuous derivative operator and $\mathcal{L}_{h,\Delta t}$ is its numerical approximation. Due to this definition the local discretization error is the norm of the difference between the governing equation and its discretized counterpart evaluated with the exact data.

LeVeque [52] showed that stability follows directly if the numerical scheme is a contraction and consistent. In the stationary case, the numerical scheme reduces to a fixed point iteration. The Banach theorem states that it will converge to a unique solution if it is a contraction. Hence, the order of a possibly employed pseudo time stepping method is not essential in this case, while consistency with the governing equation is required for the numerical solution computed in this way to approximate the physical problem.

Space discretizations of hyperbolic equations are prone to spurious undershoots and overshoots if they are more than first order accurate. The wiggles generated by such schemes indicate that some small-scale features cannot be resolved properly on a given mesh. Let us consider equation (3.2) with the flux function $f = u$, which corresponds to convection in 1D with velocity one

$$\partial_t u + \partial_x u = 0. \tag{3.6}$$

We examine the finite difference discretization

$$\frac{du_i}{dt} = -\frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{h} \tag{3.7}$$

for the equidistant mesh size h, which in 1D is similar to the Galerkin FEM. In the latter equation, $f_{i\pm\frac{1}{2}}$ denotes the numerical flux evaluated in the center of the left (-) or right (+) neighboring cell. Two approximations of the numerical flux are investigated. We employ either

$$f^{CDS}_{i\pm\frac{1}{2}} = \frac{u_{i\pm 1} + u_i}{2} \tag{3.8}$$

or

$$f^{UDS}_{i\pm\frac{1}{2}} = u_I, \tag{3.9}$$

where I is located upstream to $i \pm \frac{1}{2}$. The former flux approximation results in the central difference scheme

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_{i+1}}{2h}, \tag{3.10}$$

which is second order accurate (high-order scheme). Formula (3.9) yields the upwind difference scheme of first order (low-order scheme)

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_i}{h}. \tag{3.11}$$

First, we apply both schemes to a continuous initial profile given by

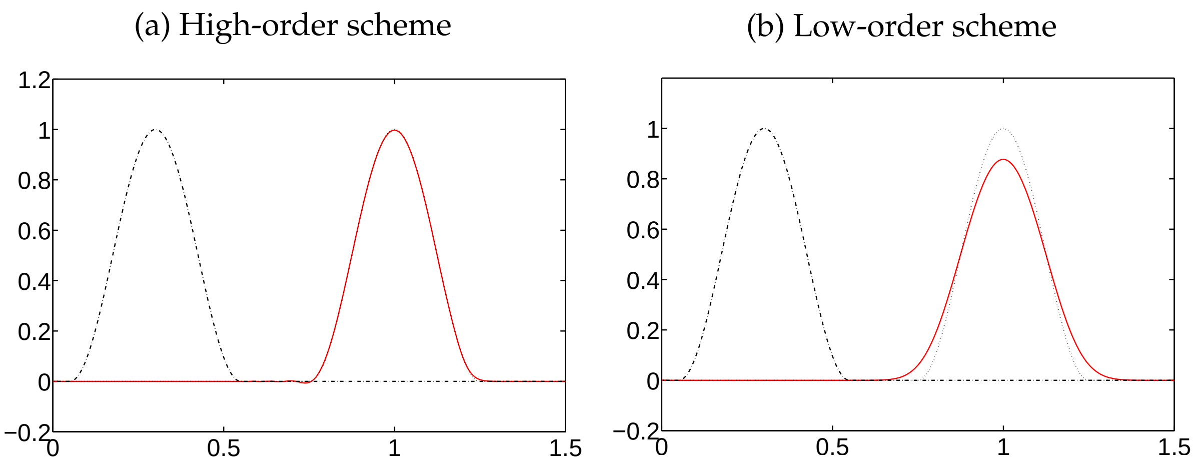
$$u(x,0) = \begin{cases} 0.5\cos\left(4\pi(x+0.2)\right) + 0.5 & \text{if} \quad 0.05 < x < 0.55 \\ 0 & \quad\text{otherwise} \end{cases}. \tag{3.12}$$

Figure 3.1 displays numerical solutions at the time instant $T = 0.7$. They are computed with $h = 0.005$, $\Delta t = 0.0001$ by the above discretization schemes combined with backward Euler time stepping. The black lines indicate the initial values (dashed-dotted) and the analytical solutions (dotted), while the numerical ones are marked by red lines. Obviously, both methods yield sensible results, although the solution computed by the low-order scheme is smeared. The high-order solution is virtually indistinguishable from the analytical one and preserves the shape of the initial data, as expected.

In a second test case we analyze both schemes for the discontinuous initial values

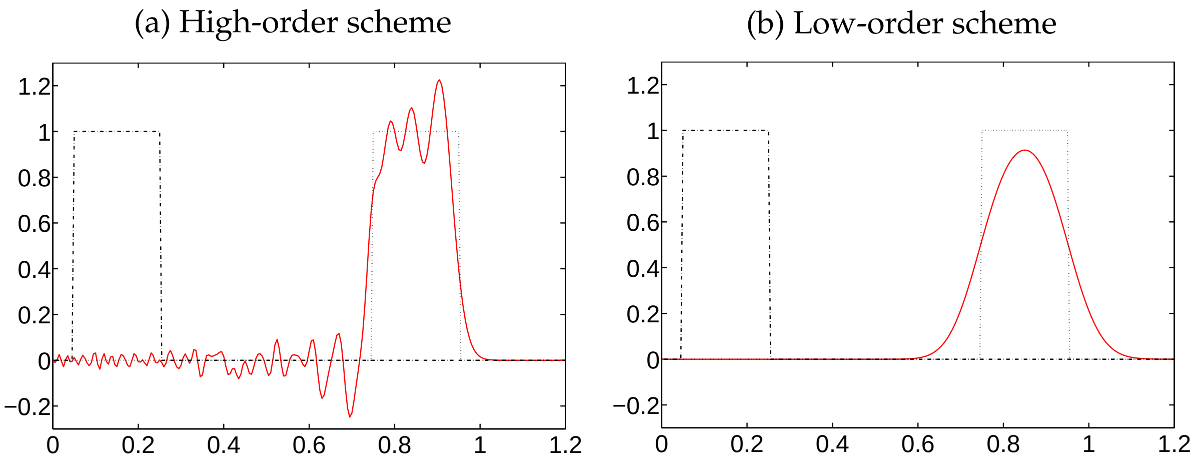
$$u(x,0) = \begin{cases} 1 & \text{if} \quad 0.05 < x < 0.25 \\ 0 & \quad\text{otherwise} \end{cases} \tag{3.13}$$

and the same mesh size and time step length. The numerical solutions to this configuration are depicted in figure 3.2. In comparison to the continuous profile the solutions differ significantly. The high-order solution is polluted by numerical oscillations and the positivity of the initial values is not preserved. The oscillatory curve has little in common with the analytical solution. In contrast, the low-order solution is non-negative and free of non-physical oscillations. It is strongly smeared but physically sensible. Indeed, the undershoots and overshoots of the high-order method in the second test case


(a) High-order scheme

(b) Low-order scheme

Figure 3.1: Convection of a continuous profile


(a) High-order scheme

(b) Low-order scheme

Figure 3.2: Convection of a discontinuous profile

are caused by the discontinuity of the solution. They are due to the inability of the numerical discretization scheme to resolve steep gradients, while the physical solution is free of oscillations. Further important physical properties of the solution like positivity or monotonicity preservation are violated due to the wiggles. This rules out the application of the high-order scheme to non-smooth data. It is demonstrated by the above examples that mathematical criteria are needed to keep the solution free of non-physical effects.

In the following sections mathematical and physical constraints are presented, which guarantee the preservation of the physical properties of the solution and prevent the birth and growth of non-physical oscillations.

3.1.1 Monotonicity

A useful physical property of two solutions v, w of the scalar equation (3.2) is

$$v_0(x) \geq w_0(x) \quad \Rightarrow \quad v(x,t) \geq w(x,t) \quad \text{for all } x,t, \tag{3.14}$$

where v_0, w_0 are the initial data. If the numerical scheme satisfies such a constraint everywhere for each time level n, i. e.

$$v_i^n \geq w_i^n \quad \Rightarrow \quad v_i^{n+1} \geq w_i^{n+1}, \tag{3.15}$$

it is called monotone. This implies that minima cannot decrease and maxima cannot increase [92], or in other words oscillations are prevented. Moreover, the monotonicity property implies that a positive and bounded initial solution remains positive and bounded [92]. Quite often that is an additional important physical feature, e. g. if the conserved quantity is the mass density of a gas. Although one can show that a consistent and monotone scheme converges to a physical solution [26, 27] and cannot generate wiggles, the monotonicity constraint is too restrictive for practical applications and difficult to verify for nonlinear schemes. A weaker condition to prevent spurious oscillations is the TVD (total variation diminishing) property.

3.1.2 Total Variation Diminishing Schemes

Total variation diminishing schemes were introduced by Harten in his pioneering work [26]. The total variation of a function u is defined by

$$TV(u) = \int_{-\infty}^{+\infty} |\partial_x u| \, dx. \tag{3.16}$$

Lax [50] showed for equation (3.2) in 1D

$$TV(u(\cdot, t_2)) \leq TV(u(\cdot, t_1)), \quad \forall\, t_2 \geq t_1, \tag{3.17}$$

i. e. the total variation of the function does not increase or in other words, the function is total variation diminishing (TVD). A discrete counterpart of (3.16) can be defined by

$$TV(u_h) = \sum_i |u_{i+1} - u_i|. \tag{3.18}$$

A numerical scheme is said to be TVD if the discrete solution satisfies

$$TV(u^{n+1}) \leq TV(u^n) \tag{3.19}$$

for all time levels. One can show [52] that a consistent numerical scheme is convergent if it is Lipschitz-continuous and the total variation is uniformly bounded. For TVD schemes, a uniform upper bound for the total variation is given in terms of the initial data, which are assumed to have a bounded total variation.

Furthermore, one can generalize the total variation diminishing property to monotonicity preservation of the approximate solution. This criterion guarantees that

$$\begin{gathered} u_i^n \geq u_{i+1}^n \quad \Rightarrow \quad u_i^{n+1} \geq u_{i+1}^{n+1}, \quad \forall i \\ \text{or} \\ u_i^n \leq u_{i+1}^n \quad \Rightarrow \quad u_i^{n+1} \leq u_{i+1}^{n+1}, \quad \forall i. \end{gathered}$$

In other words, an initially monotonic profile remains monotonic. LeVeque showed [52] that every TVD scheme is monotonicity preserving and therefore non-oscillatory. In contrast to the monotonicity constraint of a numerical scheme reported in section 3.1.1 the TVD property can be checked in practice using Harten's theorem [26].

Note that for linear schemes the concepts of total variation diminishing, monotonicity, and monotonicity preservation are equivalent [26, 23, 92]. Due to the famous Godunov theorem [23], a linear monotone scheme is at most first order accurate. This rules out linear schemes in practical applications since they are insufficient to compute accurate results at reasonable costs. Therefore, nonlinear high-resolution schemes are to be recommended for the numerical solution of hyperbolic equations, even if they require additional nonlinear iterations. Moreover, Goodman and LeVeque [24] proved that in two space dimensions the order of approximation of an arbitrary total variation diminishing scheme is bounded from above by one. In light of this result, nonlinear TVD schemes are suitable for 1D computations in contrast to multidimensions, where it is impossible to design a TVD scheme of higher order. One therefore has to find a weaker constraint to construct a non-oscillatory discretization scheme of higher than first order. In addition, such a scheme must be nonlinear to circumvent the Godunov theorem.

3.1.3 Local Extremum Diminishing Schemes

A slightly weaker constraint to be imposed on the numerical discretization is the local extremum diminishing (LED) property. A numerical method is local extremum diminishing if local minima do not decrease and local maxima cannot increase. This definition obviously rules out wiggles. The LED criterion below is defined on the semi-discrete level and all the necessary information is provided by the coefficients of the spatial discretization operator.

Theorem 3.1.1 (LED Criterion [36]) *A semi-discrete scheme of the form*

$$\frac{du_i}{dt} = \sum_{j \neq i} \sigma_{ij}(u_j - u_i) \tag{3.20}$$

is local extremum diminishing if $\sigma_{ij} \geq 0$ holds for all $j \neq i$.

To prove this theorem, suppose that a local maximum is attained at node i. This implies that the time derivative of the nodal value u_i is less than or equal to zero. Hence, a local maximum cannot increase. Similarly, a local minimum cannot decrease.

Theorem 3.1.1 forms the basis for the construction of the numerical schemes used in this thesis. The LED criterion can easily be verified in practice by examining the sign of the corresponding matrix entries. This criterion is also applicable to multidimensional discretizations and unstructured meshes. Moreover, total variation and local extremum diminishing schemes are closely related. As shown by Jameson [36], the total variation for a three-point finite difference can be written as

$$TV_h(u_h) = \sum_{i=-\infty}^{+\infty} |u_{i+1} - u_i| = 2\left(\sum \max u_h - \min u_h\right) \tag{3.21}$$

and, hence, the LED criterion implies the TVD property in the one-dimensional case. Note that the LED criterion is satisfied in the case of (3.11), while it is violated by the central difference approximation (3.10). This explains the oscillations in figure 3.2 (a) and the non-oscillatory solution in (b).

3.1.4 Positivity Preservation

Sometimes the conserved quantity, such as mass density, is known to be positive by physical reasons. This property should be respected by the discretization method. Positivity of the numerical solution on the fully discrete level also relies on the time integration method. The backward Euler scheme combined with an LED space discretization scheme turns out to be unconditionally positivity preserving since

$$u_i^{n+1} = u_i^n + \underbrace{\Delta t \sum_{j \neq i} \sigma_{ij}(u_j^{n+1} - u_i^{n+1})}_{\geq 0} \geq u_i^n \tag{3.22}$$

if u_i^{n+1} is the global minimum at the time level $n+1$. For other time stepping schemes, like the Crank-Nicolson method, the admissible time step may be bounded from above to guarantee positivity [48]. Note that an initially non-positive solution is not forced to be positive by the numerical scheme, while an initially positive one remains positive for all times. Furthermore, the combination of an LED scheme and backward Euler time stepping is stable with respect to the L_∞-norm. Suppose that u_i^{n+1} is the global maximum at the time level $n+1$ (with $u_k \geq 0\ \forall k$ for simplicity) and the solution at the time level n is bounded by a constant C, that is, $\max_j u_j^n \leq C$. Then it follows that

$$u_i^{n+1} = u_i^n + \underbrace{\Delta t \sum_{j \neq i} \sigma_{ij}(u_j^{n+1} - u_i^{n+1})}_{\leq 0} \leq u_i^n \leq C, \tag{3.23}$$

and by an inductive argument, the numerical solution is uniformly bounded from above. Hence, the numerical solution and the numerical errors alike are bounded from above and from below. The arguments above are restricted to the linear case.

3.1.5 Conservation Property

The governing equation (3.1) is written in a conservative divergence form. It can be transformed to an integral form

$$\partial_t \int_\Omega u\, d\mathbf{x} + \int_{\partial\Omega} \mathbf{f} \cdot \mathbf{n}\, ds = 0. \tag{3.24}$$

Therefore, the total amount of u may only change due to fluxes through the boundary, while there is no loss or gain of mass in the interior of the domain. This physical fact must be transferred to the numerical discretization. In the absence of inlets and outlets, the time derivative of the total mass should vanish on the discrete level. Moreover, a conservative discretization is also required from the mathematical point of view since non-conservative methods may yield waves moving at wrong speeds. An example for the inviscid Burgers equation is presented by LeVeque in [52]. The conservation property of the numerical schemes reported in the present thesis, will be investigated below.

3.2 High-Resolution Schemes Based on Algebraic Flux Correction

In the sections above we specified constraints to be imposed on discretization schemes for hyperbolic equations. These criteria are needed to design non-oscillatory methods, which preserve the physical properties of the solution. In practice, the LED criterion provides a handy tool for the design of non-oscillatory methods. To motivate the idea of high-resolution schemes and algebraic flux correction let us again focus on one space dimension and the discretizations (3.11) and (3.10), respectively. First of all, it is important to understand the origin of wiggles. Both schemes are related in some sense and they differ by a discrete diffusion term. The upwind difference scheme can be written as the sum of the central difference scheme and a first order diffusion term

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_i}{h} = \frac{u_{i-1} - u_{i+1}}{2h} + \frac{h}{2}\,\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}. \tag{3.25}$$

The second term on the right hand side is diffusive since it is the discrete form of $\frac{h}{2}\partial_{xx}u(x_i)$. Obviously, the upwind difference can be interpreted as the central difference plus a diffusive term. On the other hand, the central difference can be decomposed into the sum of the upwind difference and an antidiffusive term

$$\frac{u_{i-1} - u_{i+1}}{2h} = \frac{u_{i-1} - u_i}{h} - \frac{h}{2}\,\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}. \tag{3.26}$$

The diffusive term in (3.25) is proportional to the mesh size and reduces the order of approximation to one. It can be decomposed into a sum of two fluxes into node i satisfying the LED criterion

$$\frac{h}{2}\,\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = \frac{1}{2h}(u_{i-1} - u_i) + \frac{1}{2h}(u_{i+1} - u_i). \tag{3.27}$$

This is the reason why fluxes of LED type are called diffusive. If their coefficient is negative, they are of antidiffusive nature. The central difference scheme can be decomposed into a sum of increments

$$\frac{du_i}{dt} = \frac{u_{i-1} - u_{i+1}}{2h} = \frac{1}{2h}(u_{i-1} - u_i) - \frac{1}{2h}(u_{i+1} - u_i). \tag{3.28}$$

The first one (first term on the right hand side) satisfies the LED criterion since its coefficient $\frac{1}{2h} > 0$ is positive, while the LED criterion is violated by the second one. In fact, this contribution causes the numerical oscillations and corresponds to antidiffusion due to the negative sign of its coefficient $-\frac{1}{2h} < 0$. It is offset in (3.25) by the diffusive term. Hence, the numerical troubles in the central difference scheme are caused by increments with negative coefficients corresponding to antidiffusion. In contrast, the upwind difference is obviously of LED type, which explains the non-oscillatory solution in figure 3.2 (b).

As a conclusion from (3.25) the central difference scheme (high-order scheme) can be stabilized by addition of a discrete diffusion term, which offsets the antidiffusive fluxes to satisfy the LED criterion (upwind scheme, low-order scheme). On the other hand some antidiffusion is admissible at least in smooth regions, where it improves the resolution. This can be deduced from figure 3.1. Hence, a limited fraction of the antidiffusive fluxes can be reinserted into the low-order scheme to achieve high resolution at least in smooth regions. The amount of admissible antidiffusion is controlled by a nonlinear TVD limiter function Φ.

A high-resolution scheme can be constructed in terms of a nonlinear convex combination of the antidiffusive high-order and overdiffusive low-order scheme

$$f_{i\pm\frac{1}{2}}^{HIGHRES} = \Phi f_{i\pm\frac{1}{2}}^{HIGH} + (1-\Phi) f_{i\pm\frac{1}{2}}^{LOW} \tag{3.29}$$

to take advantage of both approximations and control the amount of antidiffusion. The limiter function is defined so as to place a greater weight on the high-order scheme in smooth parts of the solution, where it provides superior accuracy. In the vicinity of steep gradients the limiter function switches to the low-order scheme to prevent numerical oscillations. In other words, the limiter function balances the magnitude of diffusive and antidiffusive fluxes. This is the basic idea of high-resolution schemes. A solution to the second test case given in section 3.1 is depicted in figure 3.3. Obviously, it is free of non-physical effects and the discontinuities are much better resolved than in figure 3.2 (b). It is the main goal of this section to generalize the above formula to multidimensions and finite elements.

In recent years, Kuzmin et al. [43, 44, 45, 46, 47, 60, 49] developed a general approach to the construction of high-resolution LED schemes based on the principle of algebraic flux correction. In particular, this approach leads to a handy generalization of TVD limiters to implicit finite element discretizations of convection-dominated transport equations and hyperbolic systems. Algebraic flux correction constrains the coefficients of a numerical scheme so as to enforce the LED property and/or positivity conditions in a conservative way. This can be done in a black-box fashion since all the necessary information is inferred from the discrete transport operators. Moreover, the underlying limiting strategy is fully multidimensional and suitable for arbitrary meshes.

In the present thesis, algebraic flux correction based on a multidimensional nodal limiter of TVD type is employed [44]. This limiter requires row-sum mass lumping, which results in a loss of accuracy in time-dependent computations. Compared with the FCT algorithm that can accommodate the consistent mass matrix, this is a drawback. In the case of stationary computations, the time derivative vanishes in the steady-state limit and hence mass lumping does not induce an error. Moreover, it has a stabilizing effect and corresponds to an underrelaxation of the (implicit) solver. Therefore, mass lumping is undesirable for time-dependent problems but advantageous in steady-state computations because the matrix properties are improved without affecting the accuracy of

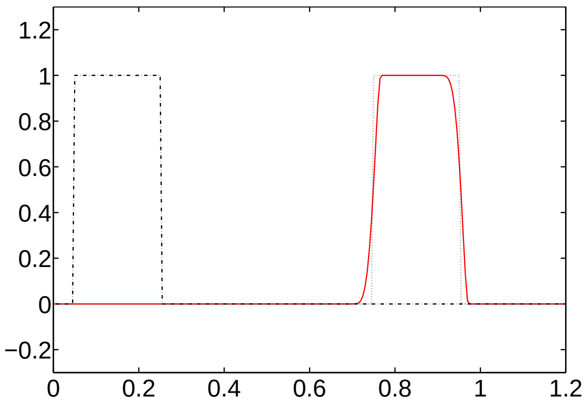


Figure 3.3: Convection of a discontinuous profile: High-resolution scheme

the converged solution. Algebraic flux correction of TVD type yields a nonlinear LED discretization which is guaranteed to preserve the physical properties of the solution.

The process of stabilization of convective terms by algebraic flux correction consists of three basic steps. First, the underlying high-order transport operator is assembled. In this thesis, the Galerkin finite element discretization serves as the high-order scheme. In a second step, the antidiffusive part of the discrete transport operator is removed by adding an artificial diffusion operator. The latter is designed on the algebraic level so as to enforce the LED constraint for the off-diagonal matrix entries. Finally, a limited amount of antidiffusion is reinserted into the scheme, to increase the order of approximation in regions of smoothness. This correction preserves the LED property and yields a nonlinear high-resolution scheme with solution-dependent coefficients. A roadmap of algebraic manipulations to be performed is as follows:

1. High-order scheme (Galerkin FEM) (section 3.2.1):

$$M_C \frac{du}{dt} = Ku$$

2. Low-order scheme (section 3.2.3):

$$M_L \frac{du}{dt} = Lu, \quad L = K + D$$

   The discrete diffusion operator D is presented in section 3.2.2.

3. High-resolution scheme (section 3.2.4):

$$M_L \frac{du}{dt} = Lu + F^* u = K^* u$$

   The operator F^* contains a limited amount of admissible antidiffusion, which is reinserted into the low-order scheme.

Each stage of the above roadmap will be described in detail in the following sections.

3.2.1 High-Order Scheme

The Galerkin finite element method is a powerful and modern tool for the spatial discretization of partial differential equations and it is in principle applicable to arbitrary unstructured meshes. It is based on a weak form of the governing equation and delivers the best approximation of the analytical solution in a finite-dimensional space spanned by piecewise-polynomial basis functions with compact support.

In this thesis the finite element method based on linear or bilinear elements is employed. For the sake of simplicity consider the continuity equation

$$\partial_t u + \nabla \cdot (u\mathbf{v}) = 0 \tag{3.30}$$

where u is a scalar quantity and $\mathbf{v}$ is a prescribed velocity field. This hyperbolic PDE is endowed with suitable initial and boundary conditions. We multiply equation (3.30) by a test function ω and integrate over the domain Ω

$$\int_\Omega \omega \partial_t u + \int_\Omega \omega \nabla \cdot (u\mathbf{v})\, d\mathbf{x} = 0. \tag{3.31}$$

Let $\{\varphi_i, i = 1, ..., NVT\}$ be the basis functions spanning the finite-dimensional subspace. Each basis function is associated with a vertex of the mesh and NVT is the total number of vertices. The approximate solution is defined as

$$u_h(\mathbf{x}, t) = \sum_j \varphi_j(\mathbf{x}) u_j(t). \tag{3.32}$$

It is common practice in finite element codes for hyperbolic conservation laws to interpolate the vector of inviscid fluxes in the same way as u_h

$$(u\mathbf{v})_h(\mathbf{x}, t) = \sum_j \varphi_j(\mathbf{x}) u_j(t) \mathbf{v}_j(t). \tag{3.33}$$

This kind of approximation was introduced by Fletcher [20] and examined in detail for the inviscid Burgers equation. It is called the group finite element formulation. A substitution of (3.32) and (3.33) into (3.31) and replacing the test function ω with the basis functions yields a semi-discretized system of ordinary differential equations

$$\sum_j \left[\int_\Omega \varphi_i \varphi_j \, d\mathbf{x} \right] \frac{du_j}{dt} + \sum_j \left[\int_\Omega \varphi_i \nabla \varphi_j \, d\mathbf{x} \right] \cdot \mathbf{v}_j u_j = 0 \quad \forall i. \tag{3.34}$$

It can be written in a more compact form

$$M_C \frac{du}{dt} = Ku \tag{3.35}$$

with the consistent mass matrix $M_C = \{m_{ij}\}$, where

$$m_{ij} = \int_\Omega \varphi_i \varphi_j \, d\mathbf{x}. \tag{3.36}$$

The matrix $K = \{k_{ij}\}$ represents a discrete transport operator. Its entries are given by

$$k_{ij} = -\mathbf{v}_j \cdot \mathbf{c}_{ij} \tag{3.37}$$

with the Galerkin coefficients

$$\mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, d\mathbf{x}. \tag{3.38}$$

The coefficients $\mathbf{c}_{ij}$ can be interpreted as the discrete form of the space derivative operator. They have zero row sums due to the fact that the basis functions sum to unity everywhere. A very efficient computation of the transport operator K is enabled by the group finite element formulation since the Galerkin coefficients can be computed in a preprocessing step and stored for later reference. Thus the numerical integration has to be performed only once.

The model problem (3.30) represents a linear equation. This is sufficient for the purpose of the present chapter and there is no loss of generality. Indeed, the semi-discrete group finite element formulation of equation (3.1) with a nonlinear flux function $\mathbf{f}$ reads

$$\sum_j \left[\int_\Omega \varphi_i \varphi_j \, d\mathbf{x} \right] \frac{du_j}{dt} + \sum_j \left[\int_\Omega \varphi_i \nabla \varphi_j \, d\mathbf{x} \right] \cdot (\mathbf{f}_j - \mathbf{f}_i) = 0 \quad \forall i \tag{3.39}$$

and can also be written in the form (3.35). In this case, the entries of the discrete transport operator are defined in terms of the (approximate) characteristic speed

$$\mathbf{v}_{ij} = \begin{cases} \frac{\mathbf{f}(u_j) - \mathbf{f}(u_i)}{u_j - u_i} & \text{if} \quad u_i \neq u_j \\ \frac{\partial \mathbf{f}(u)}{\partial u} & \text{if} \quad u_i = u_j = u \end{cases} \quad \Rightarrow \quad k_{ij} = -\mathbf{v}_{ij} \cdot \mathbf{c}_{ij}. \tag{3.40}$$

The Galerkin finite element discretization described so far is known to be second order accurate for linear elements and it is therefore referred to as the high-order scheme. On the one hand, the finite element method is sufficiently accurate, globally conservative, and linear. On the other hand, it turns out to be unstable and oscillatory due to the large amount of built-in antidiffusion. Therefore, a suitable stabilization in terms of numerical diffusion is required.

3.2.1.1 Discrete Conservation

The Galerkin finite element discretization described in section 3.2.1 is globally conservative. It can be written as a discrete conservation law since the finite element basis functions satisfy

$$\sum_i \varphi_i \equiv 1. \tag{3.41}$$

Summing up all equations of (3.34) or (3.39) and substituting (3.41) one obtains

$$\sum_j \int_\Omega \varphi_j \frac{du_j}{dt} \, d\mathbf{x} + \sum_j \int_\Omega \mathbf{f}_j \cdot \nabla \varphi_j \, d\mathbf{x} = 0. \tag{3.42}$$

This can be rewritten as

$$\sum_j \int_\Omega \varphi_j \frac{du_j}{dt}\, d\mathbf{x} + \sum_j \int_\Omega \nabla \cdot (\varphi_j \mathbf{f}_j)\, d\mathbf{x} = 0 \tag{3.43}$$

or equivalently

$$\partial_t \int_\Omega u_h \, d\mathbf{x} + \int_\Omega \nabla \cdot \mathbf{f}_h \, d\mathbf{x} = 0, \tag{3.44}$$

since the nodal values are independent of $\mathbf{x}$. The latter equation represents a discrete conservation law, similar to the integral form of (3.1). A change of mass is only possible due to fluxes through the boundary. Indeed, from the divergence theorem we have that

$$\partial_t \int_\Omega u_h \, d\mathbf{x} + \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{f}_h \, ds = 0 \tag{3.45}$$

holds true, which is the discrete analog to (3.24). In other words, when the boundary integral vanishes, so does the time derivative. Therefore, there is no loss or gain of mass in a closed system. Note that the conservation property is restricted globally. This is not a drawback compared with the local, elementwise conservation property of finite volume schemes. Finite element matrices are sparse due to the small support of finite element basis functions and therefore, mass exchange occurs only between neighboring nodes. The global conservation property of the finite element discretization (3.35) can therefore be interpreted as a somewhat weaker form of the local conservation principle.

3.2.1.2 Conservative Flux Decomposition

In general, the i-th equation in system (3.35) does not possess the LED representation (3.20). The contribution of the discrete transport operator can be written as a sum of compressible and incompressible contributions

$$\begin{aligned} (Ku)_i &= \sum_j k_{ij} u_j = \sum_{j \neq i} k_{ij} u_j + u_i k_{ii} + u_i \sum_{j \neq i} k_{ij} - u_i \sum_{j \neq i} k_{ij} && (3.46) \\ &= \underbrace{\sum_{j \neq i} k_{ij} (u_j - u_i)}_{\text{incompressible part}} + \underbrace{u_i \sum_j k_{ij}}_{\text{compressible part}} && (3.47) \end{aligned}$$

The compressible part acts as a source term and may generate extrema by physical reasons. This term should not be affected by the stabilization. On the other hand, the incompressible part is responsible for the numerical oscillations. The sum of fluxes into each node can be subdivided into fluxes, which violate the LED criterion ($k_{ij} < 0$) and have to be limited or canceled and others, which are harmless . The harmless fluxes are of a diffusive nature and the corresponding off-diagonal coefficients are positive ($k_{ij} > 0$). The troublesome antidiffusive terms are associated with negative off-diagonal matrix entries ($k_{ij} < 0$) that must be canceled or limited.

3.2.2 Discrete Diffusion Operators

As mentioned above, the LED criterion provides a convenient tool to construct non-oscillatory methods. To suppress numerical oscillations, stabilize the scheme and render the numerical solution local extremum diminishing, a numerical diffusion operator is added. It is designed on the algebraic level and is free of parameters since all information is derived from the discrete transport operator. Furthermore, the artificial diffusion operator is constructed based on the semi-discrete equation (3.35).

A quadratic matrix $D = \{d_{ij}\}$ is called a discrete or numerical diffusion operator if its entries satisfy

1. $d_{ij} = d_{ji}$
2. $\sum_i d_{ij} = \sum_j d_{ij} = 0$.

In other words, a discrete diffusion operator is a quadratic and symmetric matrix with vanishing row and column sums. Its entries corresponding to an edge I connecting nodes i and j can be written as

$$D_{|_I} = \begin{bmatrix} -d_{ij} & d_{ij} \\ d_{ij} & -d_{ij} \end{bmatrix}. \tag{3.48}$$

The diffusive/antidiffusive flux from node j into node i is given by ($d_{ij} > 0/d_{ij} < 0$)

$$f_{ij}^{diff} = d_{ij}(u_j - u_i). \tag{3.49}$$

Due to the constraints above the internodal fluxes are antisymmetric

$$f_{ij}^{diff} = d_{ij}(u_j - u_i) = -d_{ji}(u_i - u_j) = -f_{ji}^{diff} \tag{3.50}$$

and cancel out upon summation, as required by conservation. The contribution of the discrete diffusion operator D to the i-th equation admits the following representation:

$$\begin{aligned} (Du)_i &= \sum_j d_{ij}u_j = \sum_{j\neq i} d_{ij}u_j + u_i d_{ii} + u_i \sum_{j\neq i} d_{ij} - u_i \sum_{j\neq i} d_{ij} && (3.51) \\ &= \sum_{j\neq i} d_{ij}(u_j - u_i) + u_i \sum_j d_{ij} && (3.52) \\ &= \sum_{j\neq i} d_{ij}(u_j - u_i) = \sum_{j\neq i} f_{ij}^{diff}. && (3.53) \end{aligned}$$

Hence, the sum of diffusive fluxes is of LED type (3.20) provided that $d_{ij} \geq 0$ for $j \neq i$. In this case, the application of the discrete diffusion operator causes a flattening or smearing of solution profiles, while negative off-diagonal entries have a steepening effect and correspond to antidiffusion. Discrete diffusion operators with non-negative off-diagonal entries are diagonally dominant. Therefore, they improve the matrix properties and speed up the convergence of iterative solvers for the fully discrete problem.

3.2.3 Low-Order Scheme

Since the optimal amount of artificial diffusion depends on the properties of the discrete transport operator K, the coefficients of the numerical diffusion operator D must be fitted to the underlying high-order discretization. The high-order scheme corresponds to a linear algebraic system of the form (3.35), where the consistent mass matrix M_C contains positive off-diagonal entries. These entries are due to mass antidiffusion and violate the LED principle. Therefore, the consistent mass matrix may also generate spurious undershoots and overshoots. To suppress this wiggles, one adds mass diffusion and replaces the consistent mass matrix by its lumped counterpart

$$M_L = diag\{m_i\}, \qquad m_i = \sum_j m_{ij} \tag{3.54}$$

to obtain the lumped mass Galerkin scheme

$$M_L \frac{du}{dt} = Ku. \tag{3.55}$$

The applied mass diffusion operator features the discrete diffusion operator property and satisfies the requirements of section 3.2.2. Therefore, the lumped mass Galerkin scheme remains conservative since the Galerkin discretization is globally conservative as stated in section (3.2.1.1).

A numerical diffusion operator $D = \{d_{ij}\}$ is also used to eliminate all negative off-diagonal entries of K and to satisfy the LED criterion. This is equivalent to extinguishing all the antidiffusive fluxes in (3.47). By default, the artificial diffusion coefficients are defined as [46]

$$d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} = d_{ji} \quad \forall j \neq i \tag{3.56}$$

and

$$d_{ii} = -\sum_{j \neq i} d_{ij}. \tag{3.57}$$

Obviously, the resulting operator $L = K + D$ is free of negative off-diagonal entries since $l_{ij} = k_{ij} + d_{ij} \geq 0 \quad \forall j \neq i$. Moreover, the modified Galerkin scheme remains conservative since the artificial diffusion term can be decomposed into numerical fluxes. The contribution of the modified transport operator L to the $i-$th equation reads

$$(Lu)_i = \sum_{j \neq i} (k_{ij} + d_{ij})(u_j - u_i) + u_i \sum_{j \neq i} k_{ij}. \tag{3.58}$$

In summary, a local extremum diminishing low-order scheme of the form

$$M_L \frac{du}{dt} = Lu, \qquad L = K + D, \qquad l_{ij} = k_{ij} + d_{ij} \tag{3.59}$$

has been constructed from a standard Galerkin discretization by

- replacing the consistent mass matrix M_C by its lumped counterpart M_L,
- enforcing the LED criterion by means of a discrete diffusion operator D.

It can be shown that the amount of artificial diffusion is proportional to the mesh size and therefore vanishes as the mesh is refined. This is also indicated by equation (3.25). On the one hand, this property implies that the low-order scheme inherits the consistency of the high-order one. On the other hand, the accuracy of approximation degrades to first order. Moreover, the low-order scheme is linear, unless the governing equation is nonlinear. Due to the Godunov theorem [23], one must switch to the class of nonlinear schemes, to increase the order of approximation and simultaneously preserve the monotonicity of numerical solutions.

3.2.4 Upwind-Biased Flux Limiting of TVD Type

The amount of artificial diffusion built into the low-order scheme depends on the mesh and on the velocity field but not on the local smoothness of the solution. Stabilization by numerical diffusion is crucial in the vicinity of steep gradients and local extrema, while it can be removed elsewhere. This can be accomplished by adding a limited amount of antidiffusion in regions where the solution varies smoothly.

Recall that the incompressible part of the discrete transport operator consists of diffusive and antidiffusive fluxes. The antidiffusive ones are responsible for the oscillations and must be limited. Due to the construction of the discrete diffusion operator, the low-order scheme is free of antidiffusive fluxes. To insert a limited amount of antidiffusion, the antidiffusive fluxes into each node i are multiplied by correction factors $\alpha_{ij} \in [0,1]$ before they are added to the low-order scheme:

$$f_{ij} = d_{ij}(u_i - u_j) \qquad \longrightarrow \qquad f_{ij}^* = \alpha_{ij} d_{ij}(u_i - u_j). \tag{3.60}$$

The correction factors should satisfy $\alpha_{ij} = \alpha_{ji}$ to keep the limited antidiffusive fluxes antisymmetric and hence conservative. Obviously, the high-order scheme is recovered by $\alpha_{ij} = 1$ and the low-order one by $\alpha_{ij} = 0$, respectively. We define the nonlinear antidiffusion operator F^*u by its edge-wise contributions

$$(F^*u)_i = \sum_{j \neq i} \alpha_{ij} d_{ij}(u_i - u_j) = \sum_{j \neq i} f_{ij}^*. \tag{3.61}$$

To achieve high resolution, as far as possible, the local extremum diminishing antidiffusive term F^*u is added to the right-hand side of the low-order scheme

$$M_L \frac{du}{dt} = Lu + F^*u = K^*u \tag{3.62}$$

and results in the high-resolution operator K^*u. It is worth mentioning that the quality of the resulting high-resolution scheme strongly depends on the features of the low-order method. The low-order scheme must naturally be free of wiggles, otherwise there will be no chance to construct a non-oscillatory high-resolution scheme. The low-order scheme described so far is non-oscillatory and local extremum diminishing by construction since all negative off-diagonal entries of the discrete transport operator have been eliminated in a conservative fashion.

The specification of the correction factors α_{ij} is still an open question. To evaluate the limiter (step three of the algorithm below), the edge $\{ij\}$ between two nodes i and j must be oriented in the flow direction. Node i is located upwind and the limiter function is evaluated at this node if

$$l_{ji} \geq l_{ij}, \tag{3.63}$$

otherwise j is the upwind node and the limiter is evaluated at node j. The correction factors and the amount of admissible antidiffusion can be computed by the following algorithm [43, 45]:

1. Prelimit the unlimited antidiffusive fluxes

$$f_{ij} = d_{ij}(u_i - u_j) \qquad \longrightarrow \qquad f_{ij} = \min\{d_{ij}, l_{ji}\}(u_i - u_j). \tag{3.64}$$

2. Compute the sums of positive and negative prelimited antidiffusive fluxes

$$P_i^+ = \sum_{k_{ij} \leq k_{ji}} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{k_{ij} \leq k_{ji}} \min\{0, f_{ij}\}. \tag{3.65}$$

3. Compute the local extremum diminishing upper and lower bounds

$$Q_i^+ = \sum_{j \neq i} d_{ij} \max\{0, u_j - u_i\}, \qquad Q_i^- = \sum_{j \neq i} d_{ij} \min\{0, u_j - u_i\}. \tag{3.66}$$

4. Evaluate the flux limiter function for each node i

$$R_i^\pm = \min\{1, \frac{Q_i^\pm}{P_i^\pm}\}. \tag{3.67}$$

5. Compute the correction factors for every edge $\{ij\}$ with upwind node i

$$\alpha_{ij} = \begin{cases} R_i^+ & \text{if } f_{ij} > 0 \\ R_i^- & \text{if } f_{ij} \leq 0 \end{cases}, \qquad \alpha_{ji} = \alpha_{ij}. \tag{3.68}$$

6. Compute the sums of limited antidiffusive fluxes for each node i

$$(F^* u)_i = \sum_{j \neq i} \alpha_{ij} f_{ij}. \tag{3.69}$$

Algebraic flux correction can be implemented in terms of a routine, which is called repeatedly in every iteration step. Since the finite element basis functions have a small support, the Galerkin coefficients $\mathbf{c}_{ij}$ are equal to zero, unless i and j are adjacent nodes or equivalently, i and j share an edge of the sparsity graph. Edge-based data structures and matrix/vector assembly algorithms are therefore recommended.

It follows by the definition of the nodal correction factors that [43]

$$|R_i^\pm P_i^\pm| \leq |Q_i^\pm|. \tag{3.70}$$

Therefore, the sum of limited antidiffusive fluxes received by an upwind node i from its downwind neighbors $j \neq i$ is constrained by the local extremum diminishing upper and lower bounds

$$Q_i^- \leq \sum_{j\neq i} \underbrace{R_i^{-*} d_{ij}}_{=q_{ij}^-} \min\{0, u_j - u_i\} \leq \sum_{j\neq i} \alpha_{ij} f_{ij} \leq \sum_{j\neq i} \underbrace{R_i^{+*} d_{ij}}_{=q_{ij}^+} \max\{0, u_j - u_i\} \leq Q_i^+ \quad (3.71)$$

with nonlinear diffusion coefficients $q_{ij}^{\pm} \geq 0$. Remarkably, due to the last estimate the antidiffusive fluxes can be interpreted as diffusive ones satisfying the LED criterion, although the coefficient

$$k_{ij}^* = k_{ij} + (1 - \alpha_{ij}) d_{ij} \quad (3.72)$$

corresponding to the upwind node i may be negative. The prelimiting step [43]

$$f_{ij} = \min\{d_{ij}, l_{ji}\}(u_i - u_j) \quad (3.73)$$

ensures that the contribution of the edge $\{ij\}$ to the downwind node j is local extremum diminishing. The off-diagonal coefficient k_{ji}^* is actually non-negative

$$k_{ji}^* = \underbrace{l_{ji}}_{\geq 0} - \underbrace{\min\{d_{ij}, l_{ji}\}}_{\leq l_{ji}} \underbrace{\alpha_{ij}}_{\leq 1} \geq 0. \quad (3.74)$$

To summarize, there exists a nonlinear equivalent operator

$$L^* u = K^* u, \qquad l_{ij}^* \geq 0, \quad (3.75)$$

which satisfies the LED criterion for a given solution u [49] and this property carries over to the operator K^*, although it may contain negative off-diagonal entries. Note that the equivalent operator L^* is needed for theoretical reasons, while the numerical scheme is implemented based on K^*.

4 Coupled Transport Equations

To obtain a numerical solution to the computational two-fluid model (2.68) is a challenging task since many effects must be taken into consideration. In this section the computational model will be simplified by neglecting the algebraic source terms which couple both phases. This will allow us to develop solvers for the convective terms separately. The final model consists of two sets of coupled mass, momentum, and energy conservation laws for the gas and particles. The pressureless model of the particulate phase will be considered in this section.

4.1 Mathematical Properties

To avoid confusion and simplify notation, the index p, denoting the particulate phase, will be dropped in the following. Neglecting the right hand sides of equations four to six in (2.68), the 2D system of conservation laws that govern the motion of particles can be written as

$$\partial_t \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} \\ \rho E \mathbf{v} \end{bmatrix} = 0 \tag{4.1}$$

or simply

$$\partial_t U + \nabla \cdot \mathbf{F} = \partial_t U + \partial_x F^{(x)} + \partial_y F^{(y)} = 0. \tag{4.2}$$

In the former equation $U = (\rho, \rho u, \rho v, \rho E)^T = (U^{(1)}, U^{(2)}, U^{(3)}, U^{(4)})$ is the vector of conservative variables. The effective density is given by $\rho = \alpha_p \rho_p$, and $\mathbf{v} = (u, v)^T$ denotes the velocity vector. The inviscid flux vectors

$$F^{(x)} = \begin{bmatrix} \rho u \\ \rho u^2 \\ \rho uv \\ \rho E u \end{bmatrix} = \begin{bmatrix} U^{(2)} \\ \frac{U^{(2)^2}}{U^{(1)}} \\ \frac{U^{(2)}U^{(3)}}{U^{(1)}} \\ \frac{U^{(2)}U^{(4)}}{U^{(1)}} \end{bmatrix} \quad \text{and} \quad F^{(y)} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 \\ \rho E v \end{bmatrix} = \begin{bmatrix} U^{(3)} \\ \frac{U^{(2)}U^{(3)}}{U^{(1)}} \\ \frac{U^{(3)^2}}{U^{(1)}} \\ \frac{U^{(3)}U^{(4)}}{U^{(1)}} \end{bmatrix} \tag{4.3}$$

define the rate of convective transport along the corresponding axes of the two-dimensional Cartesian reference frame. The tensor of all convective fluxes is denoted by

$$\mathbf{F} = \left(F^{(x)}, F^{(y)} \right). \tag{4.4}$$

The flux Jacobians for both directions are given by

$$\frac{\partial F^{(x)}}{\partial U} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{{U^{(2)}}^2}{{U^{(1)}}^2} & 2\frac{U^{(2)}}{U^{(1)}} & 0 & 0 \\ -\frac{U^{(2)}U^{(3)}}{{U^{(1)}}^2} & \frac{U^{(3)}}{U^{(1)}} & \frac{U^{(2)}}{U^{(1)}} & 0 \\ -\frac{U^{(2)}U^{(4)}}{{U^{(1)}}^2} & \frac{U^{(4)}}{U^{(1)}} & 0 & \frac{U^{(2)}}{U^{(1)}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -u^2 & 2u & 0 & 0 \\ -uv & v & u & 0 \\ -uE & E & 0 & u \end{bmatrix} \tag{4.5}$$

and

$$\frac{\partial F^{(y)}}{\partial U} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -\frac{U^{(2)}U^{(3)}}{{U^{(1)}}^2} & \frac{U^{(3)}}{U^{(1)}} & \frac{U^{(2)}}{U^{(1)}} & 0 \\ -\frac{{U^{(3)}}^2}{{U^{(1)}}^2} & 0 & 2\frac{U^{(3)}}{U^{(1)}} & 0 \\ -\frac{U^{(3)}U^{(4)}}{{U^{(1)}}^2} & 0 & \frac{U^{(4)}}{U^{(1)}} & \frac{U^{(3)}}{U^{(1)}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ -v^2 & 0 & 2v & 0 \\ -vE & 0 & E & v \end{bmatrix} \tag{4.6}$$

which together form the Jacobian tensor

$$\mathbf{A} = \left(\frac{\partial F^{(x)}}{\partial U}, \frac{\partial F^{(y)}}{\partial U} \right). \tag{4.7}$$

The coupled transport equations (4.1) describe the transport of the conserved quantities U with the velocity $\mathbf{v}$. This system can be written as

$$\partial_t \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho \mathbf{v} \\ \rho u \mathbf{v} \\ \rho v \mathbf{v} \\ \rho E \mathbf{v} \end{bmatrix} = 0. \tag{4.8}$$

A spectral analysis of both Jacobians shows that the eigenvalues are given by

$$\lambda^{(x)} = \{u, u, u, u\} \quad \text{and} \quad \lambda^{(y)} = \{v, v, v, v\} \tag{4.9}$$

for the coordinate directions x and y. Moreover, both spaces of eigenvectors are three dimensional. Hence, the Jacobians are not diagonalizable and the governing equations lack hyperbolicity. Since the eigenvalues $\lambda^{(x)}$ and $\lambda^{(y)}$ represent the characteristic speeds of wave propagation, information travels as a single wave with velocity $\mathbf{v}$.

Furthermore, the homogeneity property

$$F^{(d)}(U) = \frac{\partial F^{(d)}(U)}{\partial U} U, \quad d = x, y, \tag{4.10}$$

which also is a feature of the Euler equations (compare to (1.22)), makes it possible to rewrite the discretization equivalently as a matrix-vector product, and enables the derivation of semi-implicit time stepping schemes without loss of conservation.

4.2 Discretization and Stabilization

The discretization and stabilization of the coupled transport equations are essentially the same as for scalar equations. In particular, high-resolution finite element schemes based on algebraic flux correction of TVD type are readily applicable. The reader is referred to section 3.2 for an informal introduction to such schemes. The differences compared to the treatment of scalar transport equations are described below.

4.2.1 High-Order Scheme

Given a suitable set of basis functions $\{\varphi_i\}$, let the numerical solution and the convective fluxes be interpolated using the group finite element formulation [20]

$$U_h(\mathbf{x},t) = \sum_i \varphi_i(\mathbf{x}) U_i(t), \qquad \mathbf{F}_h(\mathbf{x},t) = \sum_i \varphi_i(\mathbf{x}) \mathbf{F}_i(t), \tag{4.11}$$

where the nodal flux tensor $\mathbf{F}_j$ is defined by (4.7) and evaluated using the solution values at node i. Multiplying the governing equations (4.1) by a test function and integrating over the domain, one obtains the Galerkin finite element discretization

$$\sum_j \left[\int_\Omega \varphi_i \varphi_j \, d\mathbf{x}\right] \frac{dU_j}{dt} + \sum_j \left[\int_\Omega \varphi_i \nabla \varphi_j \, d\mathbf{x}\right] \cdot \mathbf{F}_j = 0 \quad \forall i. \tag{4.12}$$

The latter equation can be written in a shorter form

$$\sum_j M_{ij} \frac{dU_j}{dt} = -\sum_{j \neq i} \mathbf{c}_{ij} \cdot \mathbf{F}_j \tag{4.13}$$

or

$$M_C \frac{dU}{dt} = KU \tag{4.14}$$

due to the homogeneity property. In the formula above, the $4\,NVT \times 4\,NVT$-matrix M_C (NVT = number of vertices) denotes the block consistent mass matrix

$$M_C = \{M_{ij}\} = \{m_{ij} I\}, \tag{4.15}$$

where I is the 4×4 identity matrix and m_{ij} are given by (3.36). Equation (4.13) corresponds to the standard Galerkin discretization. For the purpose of implementing numerical boundary conditions, it is more convenient to consider the weak formulation

$$\sum_j M_{ij} \frac{dU_j}{dt} = \sum_j \mathbf{c}_{ji} \cdot \mathbf{F}_j - \sum_j \left[\int_{\partial\Omega} \varphi_i \varphi_j \mathbf{n} \, ds\right] \cdot \mathbf{F}_j, \tag{4.16}$$

which offers the opportunity to prescribe boundary conditions in a weak sense (cf. section 7). In our algorithm, the last weak form of the Galerkin discretization (4.16) serves as the high-order scheme, while the stabilization is still based on (4.13).

4.2.2 Low-Order Scheme and Algebraic Flux Correction

As in section 3.2.3, the process of algebraic flux correction begins with row-sum mass lumping on the left hand side of the semi-discrete Galerkin scheme

$$M_L \frac{dU}{dt} = KU. \tag{4.17}$$

The lumped counterpart of the consistent mass matrix M_C is given by

$$M_L = diag\{M_i\}, \quad M_i = m_i I, \quad \text{and} \quad m_i = \sum_j m_{ij}. \tag{4.18}$$

We replace the consistent mass matrix by its lumped counterpart to eliminate antidiffusive mass fluxes. Since the row and column sums remain unchanged, this manipulation is conservative.

The lack of hyperbolicity rules out the usual approach in construction of approximate Riemann solvers, which is based on edge-by-edge transformations to local characteristic variables (see below). In the case of the pressureless particle equations, such transformations are neither possible nor necessary. Since there exists just one wave moving with the flow velocity $\mathbf{v}$, stabilization by scalar dissipation defined in terms of the conservative variables is feasible. Since the same diffusion coefficient applies to each equation, the lack of hyperbolicity turns out to be an advantage rather than a drawback as far as stabilization is concerned. The discrete transport coefficients

$$k_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{v}_j \tag{4.19}$$

are defined as in the case of scalar transport equations, and the artificial diffusion coefficients d_{ij} have the same value for all conservative variables. The default setting

$$d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} = d_{ji} \tag{4.20}$$

as stated in equation (3.56), is sufficient to satisfy the scalar LED criterion 3.1.1. However, the theoretical framework developed so far is invalid for nonlinear coupled systems. Therefore, a low-order scheme based on (4.20) may produce undershoots and overshoots that carry over to the flux-limited solution. It turns out that a slightly increased amount of diffusion is sufficient to get rid of non-physical oscillations and compute a physically correct Riemann solution. An algebraic analysis of the Rusanov-type scheme presented by Banks and Shadid [3] leads to the following revised definition

$$d_{ij} = \max\{|k_{ij}|, |k_{ji}|\} = d_{ji}. \tag{4.21}$$

The resulting low-order scheme is more diffusive than that based on (4.20). Indeed,

$$\max\{|k_{ij}|, |k_{ji}|\} \geq \max\{-k_{ij}, 0, -k_{ji}\}. \tag{4.22}$$

As before, the contribution of the artificial diffusion operator to the modified Galerkin scheme can be decomposed into a sum of numerical fluxes. In vectorial notation this looks like

$$F_{ij}^{diff} = |\Lambda_{ij}|(U_j - U_i), \qquad F_{ji}^{diff} = -F_{ij}^{diff}, \tag{4.23}$$

where

$$|\Lambda_{ij}| = diag\{d_{ij}, d_{ij}, d_{ij}, d_{ij}\} = d_{ij}I \tag{4.24}$$

is a diagonal matrix of diffusion coefficients.

The construction of the low-order scheme is followed by a nonlinear antidiffusive correction. As in the case of scalar equations, the amount of antidiffusion must be limited, so as to keep the scheme non-oscillatory. From the algorithm described in section 3.2.4 the correction factors can be computed for each equation separately. The only difference is the new formula (4.21) for the artificial diffusion coefficient. The pair of limited antidiffusive fluxes associated with nodes i and j is of the form

$$F_{ij}^{*} = |\Lambda_{ij}^{*}|(U_i - U_j), \qquad F_{ji}^{*} = -F_{ij}^{*}, \tag{4.25}$$

where

$$|\Lambda_{ij}^{*}| = diag\{\alpha_{ij}^{(1)} d_{ij}, \alpha_{ij}^{(2)} d_{ij}, \alpha_{ij}^{(3)} d_{ij}, \alpha_{ij}^{(4)} d_{ij}\} \tag{4.26}$$

is a limited counterpart of $|\Lambda_{ij}|$. Since the coupling is rather weak, the solution-dependent correction factors α_{ij} are chosen individually for each equation. The orientation of edges for all equations is the same as in the scalar case (3.63). This is due to the fact that there is just one wave and, consequently, just one characteristic direction in which the wave can travel.

Although algebraic flux correction for the equations of the particulate phase is performed in a segregated fashion, the resulting algebraic system should be solved in a fully coupled way. Decoupled solution strategies may require less memory but give rise to additional time step restrictions. Moreover, intermediate solutions are more likely to exhibit numerical oscillations that inhibit convergence to steady-state. Those strategies typically involve successive solution of a sequence of subproblems and demand the use of out-dated quantities. The result is an inaccurate stabilization as long as the solution has not converged. In the steady-state limit this problem vanishes since the solution will no longer change significantly. Nevertheless, convergence can only be reached for small time steps since inaccurate stabilization due to large time steps will prevent convergence. Therefore, the development of strongly coupled iterative solvers for flux-limited Galerkin discretizations constitutes a major highlight of the present thesis. The implementation of two-way coupling mechanisms, implicit time integration schemes, and Newton-like methods for nonlinear algebraic systems is described in the following chapters.

5 Euler Equations

Mathematical features of the Euler equations, which are important from the numerical point of view, were presented in section 1.1. The positive pressure term or more precisely the pressure gradient makes the governing equations hyperbolic and bounds the solution from above. This also prevents the appearance of delta shocks.

If we neglect the algebraic source terms, the equations modeling the gas phase degenerate to the Euler equations (1.6) for the effective density $\rho = \alpha_g \rho_g$. The ideal gas equation of state (1.9) is added to close the system. This system of partial differential equations is similar to (4.1), which is recovered for $P \equiv 0$. Although both systems of governing equations are closely related at first glance, the pressure term causes completely different mathematical properties and results in a much more complicated numerical solution and algebra.

On the other hand, the existence of a complete set of eigenvectors which is due to the nonzero pressure term, offers the opportunity to employ strongly coupled hyperbolic solvers, which are built on the diagonalization of the Jacobian matrices. Such solvers can be classified as exact or approximate Riemann solvers. Exact Riemann solvers like Godunov's method [23] determine the exact solution of a local Riemann problem between the states of neighboring cells (nodes). The exact solution of a Riemann problem is quite expensive and far from being trivial to compute. On the other hand approximate Riemann solvers, which are based on the diagonalization of the matrix are cheaper and easier to implement. The approximate Riemann solver of Roe, which belongs to the class of so-called flux difference splitting schemes, produces accurate results and it is very popular and implemented in many finite volume and finite element solvers [57, 52, 62, 93, 60, 47, 22, 86]. It is therefore employed in this thesis. Roe's Riemann solver takes into account the characteristic directions of the flow and chooses the upwind direction for each characteristic field separately. The concept of local extremum diminishing schemes introduced in section 3 is generalized to hyperbolic systems and applied to the Euler equations in terms of tensorial diffusion and nonlinear antidiffusion operators, which stabilize the scheme and enforce a vectorial LED criterion.

Flux limiters are designed for scalar equations and they are not directly applicable to hyperbolic systems of partial differential equations. A transformation to characteristic variables enables the application of the TVD type flux limiters reported in section 3.2.4.

In this chapter a high-resolution scheme for the discretization of the Euler equations is presented. It is constructed by a generalization of the framework of algebraic flux correction and particularly the LED criterion to the vectorial case. This kind of stabilization traces its origin to the work of Kuzmin et al. [47, 44].

5.1 Vectorial High-Resolution Scheme Based on Algebraic Flux Correction

The construction of the high-resolution TVD type scheme for the Euler equations is closely related to the scalar case (compare section 3.2) and follows the same road map as illustrated in 3.2.4. The Galerkin discretization also serves as the high-order scheme in the vectorial case, which is stabilized by a numerical diffusion operator D. In an additional step a limited amount of antidiffusion is added to increase the order of approximation in smooth regions.

5.1.1 High-Order Scheme

One can derive the Galerkin discretization in the same way as described in section 4.2.1. The result is therefore again (compare to (4.12))

$$\sum_j \left[\int_\Omega \varphi_i \varphi_j \, d\mathbf{x}\right] \frac{dU_j}{dt} + \sum_j \left[\int_\Omega \varphi_i \nabla \varphi_j \, d\mathbf{x}\right] \cdot \mathbf{F}_j = 0 \quad \forall i \tag{5.1}$$

with the flux tensor of the Euler equations introduced by (1.7). Since the basis functions sum to unity at every point one easily verifies

$$\sum_j \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \underbrace{\left(\sum_j \varphi_j\right)}_{\equiv 1} d\mathbf{x} = 0. \tag{5.2}$$

The semi-discretized Euler equations (5.1) equivalently transform to [47]

$$\sum_j M_{ij} \frac{dU_j}{dt} = -\sum_{j \neq i} \mathbf{c}_{ij} \cdot (\mathbf{F}_j - \mathbf{F}_i) \tag{5.3}$$

with the Galerkin coefficients (3.38) and the blocks of the consistent mass matrix (4.15). Equations (5.1) or equivalently (5.3) are in semi-discrete form. Due to the homogeneity property of the Euler equations, the discrete transport operator associated with the numerical Galerkin fluxes can be rewritten as a matrix vector product

$$M_C \frac{dU}{dt} = KU, \tag{5.4}$$

where

$$K = \{K_{ij}\}, \qquad K_{ij} = -\mathbf{c}_{ij} \cdot \mathbf{A}_j = K_{ij}^{(x)} + K_{ij}^{(y)} \tag{5.5}$$

with the Jacobian tensor (1.15).

For the imposition of boundary conditions it is preferable to integrate the Galerkin discretization (5.1) by parts, which yields the weak formulation

$$\sum_j M_{ij} \frac{dU_j}{dt} = \sum_j \mathbf{c}_{ji} \cdot \mathbf{F}_j - \sum_j \left[\int_{\partial\Omega} \varphi_i \varphi_j \mathbf{n} \, ds \right] \cdot \mathbf{F}_j. \tag{5.6}$$

The boundary integrals allow the application of weak Neumann-type flux boundary conditions, which are investigated in this thesis. Without taking the boundary conditions into account equation (5.6) is equivalent to (5.3). On the other hand, this formulation is improper for computing the stabilization since it does not allow the decomposition into suitable flux differences. Consequently, it is preferable to derive the stabilization operator for (5.3) and apply it to the high-order discretization (5.6).

5.1.2 Characteristic Variables

A vectorial analog to (3.40) is required for the stabilization. One can determine the eigenvalue with the largest magnitude and add scalar dissipation similar to section 4.2.2. In the case of the Euler equations this results in an overdiffusive scheme due to the different characteristic speeds associated with the eigenvalues of the Jacobians. The accuracy can be significantly improved by a stabilization which is fitted to the distinct wave speeds. This involves a diagonalization of the Jacobian matrices and a transformation to a suitable set of so-called characteristic variables. The concept of such variables is at the current state restricted to one space dimension. A dimensional splitting is therefore performed to compute the diffusion and antidiffusion tensors separately for each space dimension. Consider a linear hyperbolic system of the quasi-linear form

$$\partial_t U + A \partial_x U = 0 \tag{5.7}$$

with the constant $n \times n$-matrix A. Since the equations are assumed to be hyperbolic, the matrix A is diagonalizable with real eigenvalues

$$A = R \Lambda R^{-1}, \tag{5.8}$$

where R denotes the matrix of right eigenvectors and $\Lambda = diag\{\lambda_1, \dots, \lambda_n\}$ is a diagonal matrix of the eigenvalues. One defines characteristic variables in terms of

$$W = R^{-1} U. \tag{5.9}$$

A diagonalization of A transforms system (5.7) into a set of decoupled transport equations

$$\begin{aligned} 0 &= \partial_t U + R \Lambda R^{-1} \partial_x U & (5.10) \\ &= \partial_t (R^{-1} U) + \Lambda \partial_x (R^{-1} U) & (5.11) \\ &= \partial_t W + \Lambda \partial_x W, & (5.12) \end{aligned}$$

or equivalently

$$\partial_t W_k + \lambda_k \partial_x W_k = 0 \quad \text{for} \quad k = 1, \dots, n. \tag{5.13}$$

A solution of this set of transport equations is given by

$$W_k(x,t) = W_k(x - \lambda_k t, 0) \quad \text{for} \quad k = 1, \ldots, n \tag{5.14}$$

and can be transformed back to a solution of (5.7) by $U = RW$.

The transformation described above is restricted to the linear case and is not directly applicable to the nonlinear Euler equations since they are not in quasi-linear form. Moreover, the Jacobians depend on the conservative variables. On the other hand, local characteristic variables can be defined for a locally linearized formulation of (5.3). Roe [75] introduced mean values, known as Roe averages, which facilitate a natural generalization of characteristic variables to nonlinear systems and transform the Euler equations equivalently into a quasi-linear formulation similar to (5.7). Note that in the multidimensional case, characteristic variables are not unique since there is an infinite number of directions. The same holds true for one-dimensional nonlinear systems since the definition of characteristic variables depends on the type of linearization. Nevertheless, a transformation to characteristic variables is the only thing needed to evaluate the flux limiter function for the scheme presented here.

5.1.3 Roe Linearization

Roe's scheme belongs to the class of flux difference splitting schemes and represents an approximate Riemann solver. It was discovered by Roe [75] that the flux differences in formula (5.3) can be linearized exactly in terms of averaged Jacobians. This transforms the Euler equations into an equivalent quasi-linear formulation on the semi-discrete level, which enables the application of a vectorial LED criterion. This will be discussed below. Roe originally formulated his approach in 1D. The main idea of the Roe solver is to replace the original conservation laws by their quasi-linear form

$$\partial_t U + \tilde{A}(U)\partial_x U = 0. \tag{5.15}$$

Since the transformation to a set of characteristic variables is only possible for linear hyperbolic systems, a linearization is required to enable a local transformation. Roe formulated three conditions in 1D [75], which can be generalized to multidimensions:

1. $F_R - F_L = \tilde{A}(U_L, U_R)(U_R - U_L)$ for arbitrary left and right states U_L, U_R.
2. $\tilde{A}(U_L, U_R)$ is diagonalizable with real eigenvalues.
3. $\tilde{A}(U_L, U_R) \to \frac{\partial F(\tilde{U})}{\partial \tilde{U}}$ for $U_L, U_R \to \tilde{U}$.

In a finite volume scheme, Roe's first condition ensures that the scheme is conservative. Our high-order scheme is constructed by (5.6) and only the diffusion and antidiffusion tensors are based on the Roe linearization. Since they are assumed to be conservative by construction, this condition is not essential for conservation purposes. From the view point of conservation other averages, e. g. the arithmetic average (conditions two and three are satisfied), would also be admissible. On the other hand, such averages yield bad stabilizations since they correspond to wrong left and right states of the Riemann problem. Therefore, the first condition is important to compute a proper stabilization. The second condition is required to preserve the hyperbolicity and facilitate the transformation to characteristic variables (see above). Last but not least the third condition guarantees consistency for smooth solutions.

In his pioneering work [75] Roe shows that the three conditions above can be satisfied by evaluating the Jacobian matrix (or the uni-directional Jacobians in multidimensions) at a particular averaged state. The results can be summarized by the following theorem:

Theorem 5.1.1 (Roe Averages) *The uni-directional Roe matrices $\tilde{A}_{ij}^{(x)}$, $\tilde{A}_{ij}^{(y)}$ related to an edge $\{ij\}$, which satisfy the essential conditions (1-3), are defined by the uni-directional Jacobian matrices evaluated at an averaged state*

$$\tilde{A}_{ij}^{(x,y)} = A^{(x,y)}(\tilde{U}), \tag{5.16}$$

where

$$\tilde{U} = \begin{bmatrix} \tilde{\rho_{ij}} \\ \tilde{\mathbf{v}}_{ij} \\ \tilde{H}_{ij} \end{bmatrix}. \tag{5.17}$$

The averaged state is given by

$$\tilde{\rho}_{ij} = \sqrt{\rho_i \rho_j} \qquad \tilde{\mathbf{v}}_{ij} = \frac{\sqrt{\rho_i}\mathbf{v}_i + \sqrt{\rho_j}\mathbf{v}_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}, \qquad \tilde{H}_{ij} = \frac{\sqrt{\rho_i}H_i + \sqrt{\rho_j}H_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}. \tag{5.18}$$

Furthermore, the speed of sound

$$\tilde{c}_{ij} = \sqrt{(\gamma - 1)\left(\tilde{H}_{ij} - \frac{|\tilde{\mathbf{v}}_{ij}|^2}{2}\right)} \tag{5.19}$$

can be computed.

The proof can be extracted from [75] or verified by algebraic manipulations. Note that the Jacobians for both coordinate directions are independent of ρ.

5.1.4 Characteristic LED Criterion

The scalar LED criterion of Jameson given in theorem 3.1.1 can be generalized to hyperbolic systems of partial differential equations based on the characteristic decomposition of the Jacobians.

Theorem 5.1.2 (Characteristic LED criterion [47]) *The discretization of a linear hyperbolic system of conservation laws of the form*

$$M_i \frac{dU_i}{dt} = \sum_{j \neq i} L_{ij}(U_j - U_i) \tag{5.20}$$

is local extremum diminishing for a suitable set of characteristic variables, provided the off-diagonal blocks L_{ij} are positive semi-definite.

The LED criterion will therefore be satisfied if the off-diagonal blocks are free of negative eigenvalues. Kuzmin and Möller [47] applied theorem 5.1.2 to the Euler equations and enforced the LED criterion by eliminating negative eigenvalues on the algebraic level by an addition of a suitably defined diffusion tensor. Since this strategy, which is free of user-defined parameters, is applicable to arbitrary dimensions and unstructured meshes, it is adopted in the present work. The strategy can be interpreted as a generalization of the diffusion operator applied to scalar equations (compare to section 3.2.2).

5.1.5 Discrete Diffusion Tensors and Low-Order scheme

As mentioned above, the stabilization and hence the numerical diffusion and antidiffusion tensors are constructed on the algebraic level based on (5.3), while the high-order scheme (5.6) is obtained by integration by parts to provide a better treatment of boundary conditions. Equation (5.3) is clearly not of the form (5.20). The semi-discretized Euler equations therefore need to be cast in this form to impose the LED constraint.

First, one replaces the consistent mass matrix by the lumped one (4.18), which corresponds to an addition of mass diffusion as pointed out in sections 3.2.3 and 4.2.1. This yields the lumped mass Galerkin formulation of the Euler equations

$$M_i \frac{dU_i}{dt} = -\sum_{j \neq i} \mathbf{c}_{ij} \cdot (\mathbf{F}_j - \mathbf{F}_i). \tag{5.21}$$

The transformation to the desired LED type form implies a local linearization of the semi-discretized equations by employing the Roe linearization. The discrete transport operator of equation (5.3) applied to node i reads

$$\begin{aligned} (KU)_i &= -\sum_{j \neq i} \mathbf{c}_{ij} \cdot (\mathbf{F}_j - \mathbf{F}_i) && (5.22)\\ &= -\sum_{j \neq i} \left[c_{ij}^{(x)} (F_j^{(x)} - F_i^{(x)}) + c_{ij}^{(y)} (F_j^{(y)} - F_i^{(y)}) \right]. && (5.23) \end{aligned}$$

Due to theorem 5.1.1 one can rewrite the flux differences by the uni-directional Roe matrices $A_{ij}^{(x)}$ and $A_{ij}^{(y)}$ satisfying

$$F_j^{(x)} - F_i^{(x)} = A_{ij}^{(x)} (U_j - U_i) \quad \text{and} \quad F_j^{(y)} - F_i^{(y)} = A_{ij}^{(y)} (U_j - U_i). \tag{5.24}$$

For simplicity we drop the tilde. The matrix $c_{ij}^{(x)} A_{ij}^{(x)} + c_{ij}^{(y)} A_{ij}^{(y)}$ is known as the cumulative Roe matrix [47]. A substitution of (5.24) into (5.23) yields the desired locally linearized formulation

$$(KU)_i = -\sum_{j \neq i} \left[c_{ij}^{(x)} A_{ij}^{(x)} + c_{ij}^{(y)} A_{ij}^{(y)} \right] (U_j - U_i), \tag{5.25}$$

which corresponds to (5.20). This formulation is local extremum diminishing for a set of local characteristic variables if the matrix $-\left(c_{ij}^{(x)}A_{ij}^{(x)}+c_{ij}^{(y)}A_{ij}^{(y)}\right)$ does not contain negative eigenvalues. Since the definition of characteristic variables requires a dimensional splitting, which was introduced by Yee et al. [98], the LED criterion is imposed on the Roe matrices for both coordinate directions separately, i. e. $-c_{ij}^{(x)}A_{ij}^{(x)}$ and $-c_{ij}^{(y)}A_{ij}^{(y)}$. More precisely, this is essential for the evaluation of the TVD type flux limiter. Other techniques like slope limiters can be applied without a dimensional splitting. Note that the discrete transport operator of equation (5.3), which is equivalent to (5.25), is oscillatory and does not feature the LED property. Therefore, it usually contains off-diagonal blocks with negative eigenvalues, which can be removed by addition of suitably defined diffusion blocks.

The discrete diffusion tensor has to satisfy three requirements. At first it should be conservative to prevent the solution from generating shocks moving at wrong speeds [52]. At the same time the diffusion tensor has to provide a sufficient amount of diffusion to enforce the LED criterion. Last but not least, the resulting scheme should be independent of a particular configuration. Hence, the numerical diffusion tensor has to be free of user-defined parameters.

The discrete diffusion tensor is conservative if the diffusive internodal fluxes cancel each other out so that no mass is created or destroyed. In other words a sufficient condition for conservation is

$$F_{ij}^{diff}=D_{ij}(U_j-U_i)=-F_{ji}^{diff}. \tag{5.26}$$

Due to the hyperbolicity of the governing equations, the uni-directional Roe matrices of every edge $\{ij\}$ admit a diagonalization with real eigenvalues

$$A_{ij}^{(d)}=R_{ij}^{(d)}\Lambda_{ij}^{(d)}{R_{ij}^{(d)}}^{-1}, \qquad d=x,y, \tag{5.27}$$

where ${R_{ij}^{(d)}}^{-1}$, $R_{ij}^{(d)}$ are the matrices of left and right eigenvectors and

$$\Lambda_{ij}^{(d)}=diag\{\lambda_1^{(d)},\dots,\lambda_4^{(d)}\}_{ij} \tag{5.28}$$

is a diagonal matrix of the eigenvalues.

The Galerkin coefficients (3.38) satisfy

$$\mathbf{c}_{ij}=-\mathbf{c}_{ji}+\int_{\partial\Omega}\varphi_i\varphi_j\,\mathbf{n}\,ds \tag{5.29}$$

and can be decomposed into a skew-symmetric and a symmetric part

$$\mathbf{c}_{ij}=\mathbf{a}_{ij}+\mathbf{b}_{ij}, \quad \mathbf{a}_{ij}=\frac{\mathbf{c}_{ij}-\mathbf{c}_{ji}}{2}=-\mathbf{a}_{ji}, \quad \mathbf{b}_{ij}=\frac{\mathbf{c}_{ij}+\mathbf{c}_{ji}}{2}=\mathbf{b}_{ji}. \tag{5.30}$$

For the design of the diffusion tensor, the symmetric part corresponding to the boundary integral is neglected. One defines the 4 × 4 diffusion blocks D_{ij} in terms of their uni-directional counterparts $D_{ij}^{(d)}$

$$D_{ij} = D_{ij}^{(x)} + D_{ij}^{(y)}, \qquad D_{ij}^{(d)} = |a_{ij}^{(d)}||A_{ij}^{(d)}|, \qquad |A_{ij}^{(d)}| = R_{ij}^{(d)}|\Lambda_{ij}^{(d)}|{R_{ij}^{(d)}}^{-1} \tag{5.31}$$

for the space dimension d, to eliminate negative eigenvalues from the discrete transport operator. Due to the conservation property, the diffusion tensor has to feature block-symmetry, which is enforced by

$$D_{ji}^{(d)} = -D_{ij}^{(d)} \quad \forall j \neq i. \tag{5.32}$$

This corresponds to (5.26) and the diffusion tensor has vanishing row and column sums. The application of the discrete diffusion operator to node i results in the low-order operator

$$(LU)_i = \sum_{j\neq i} L_{ij}(U_j - U_i) = \sum_{j\neq i}(K_{ij} + D_{ij})(U_j - U_i), \tag{5.33}$$

which has positive semi-definite off-diagonal blocks L_{ij}. This is provided by the following theorem 5.1.3.

Theorem 5.1.3 *The off-diagonal blocks for each space dimension d*

$$L_{ij}^{(d)} = K_{ij}^{(d)} + D_{ij}^{(d)} \qquad \forall j \neq i \tag{5.34}$$

of the low-order operator are free of negative eigenvalues.

Proof: For simplicity we drop the index denoting the space dimension. Let λ_m for $m \in \{1, \dots, 4\}$ be an eigenvalue of K_{ij} and I be the 4×4 identity matrix. We will show that $\lambda_m + |\lambda_m| \geq 0$ is an eigenvalue of $K_{ij} + D_{ij}$. The eigenvalues are characterized by the roots x of the characteristic polynomial satisfying

$$\det(L_{ij} - xI) = 0. \tag{5.35}$$

Substituting $\lambda_m + |\lambda_m|$ into (5.35) and multiplying with

$$\det(R_{ij}^{-1})\det(R_{ij}) = 1 \tag{5.36}$$

we reveal that

$$\begin{aligned}\det(L_{ij}-(\lambda_m+|\lambda_m|)I) &= \det(R_{ij}^{-1})\det\left(K_{ij}+D_{ij}-(\lambda_m+|\lambda_m|)I\right)\det(R_{ij})\\ &= \det(R_{ij}^{-1}K_{ij}R_{ij}+R_{ij}^{-1}D_{ij}R_{ij}-(\lambda_m+|\lambda_m|)I)\\ &= \det\{\begin{bmatrix}\lambda_1+|\lambda_1| & & \\ & \ddots & \\ & & \lambda_4+|\lambda_4|\end{bmatrix}\\ &\quad -\begin{bmatrix}\lambda_m+|\lambda_m| & & \\ & \ddots & \\ & & \lambda_m+|\lambda_m|\end{bmatrix}\}\\ &= 0\end{aligned} \tag{5.37}$$

holds true since the matrix has at least one zero row/column. Hence, the four roots of (5.35) are given by

$$\lambda_k+|\lambda_k|\geq 0 \quad k=1,\dots,4. \tag{5.38}$$

These are the eigenvalues of L_{ij}, which completes the proof. □

In summary, the low-order scheme reads

$$M_L\frac{dU}{dt}=LU, \qquad L=K+D. \tag{5.39}$$

In a practical application the interior part of the low-order operator corresponding to the weak Galerkin discretization (5.6) can be assembled from the following algorithm:

- In a loop over all nodes assemble the diagonal part of the discrete transport operator

$$(LU)_i=(LU)_i+\mathbf{c}_{ii}\cdot\mathbf{F}_i. \tag{5.40}$$

- In a loop over all edges construct the low-order operator and initialize:

  1. Add Galerkin fluxes

$$(LU)_i=(LU)_i+\mathbf{c}_{ji}\cdot\mathbf{F}_j, \quad (LU)_j=(LU)_j+\mathbf{c}_{ij}\cdot\mathbf{F}_i. \tag{5.41}$$

  2. Determine Roe averages and add diffusive fluxes for $d=x,y$

$$(LU)_i=(LU)_i+D_{ij}^{(d)}(U_j-U_i), \quad (LU)_j=(LU)_j+D_{ji}^{(d)}(U_i-U_j). \tag{5.42}$$

Note that the assembly of the boundary terms of (5.6) is not included in the algorithm above. It is illustrated in chapter 7 and updates the Galerkin discretization.

5.1.6 Characteristic TVD Type Flux Limiting for the Euler Equations

After application of the discrete diffusion tensor defined in section 5.1.5 it is worthwhile to improve the accuracy by addition of a limited amount of antidiffusion. Characteristic flux limiters were first applied to the Euler equations in the 1980's [97, 98]. Unfortunately, flux limiters as described in section 3.2.4 are based on scalar equations and are not directly applicable to hyperbolic systems due to different characteristic speeds. In contrast to the coupled transport equations in section 4, the nonzero pressure term of the Euler equations is responsible for the hyperbolicity and different waves, which should be limited separately. In one space dimension this problem can easily be circumvented via a transformation to characteristic variables, which enables the application of the limiter function since there is only one unique characteristic direction. On the other hand, in multiple dimensions, waves move in arbitrary directions, which makes the definition of characteristic variables more challenging. The correction factors and the numerical solution will therefore depend on the underlying set of characteristic variables. During the last decades several multidimensional generalizations of the one-dimensional techniques have been applied in finite element codes [85, 58].

In the present thesis equations (5.3) with equivalent linearization (5.25) are transformed to a set of characteristic variables (5.43) on every edge and for every space dimension separately as suggested in [47]. In the case of the multidimensional nonlinear Euler equations, characteristic variables are defined in terms of characteristic differences. They are determined on the individual edges and for every space dimension separately based on the locally linearized equations (5.25) [47]

$$\Delta W_{ij}^{(d)} = {R_{ij}^{(d)}}^{-1} A_{ij}^{(d)} (U_i - U_j). \tag{5.43}$$

The uni-directional Roe matrix is denoted by $A_{ij}^{(d)}$ corresponding to (5.24) and ${R_{ij}^{(d)}}^{-1}$ is the matrix of left eigenvectors due to (5.27). Note that the characteristic variables may be discontinuous in every node i and, although theorem 5.1.3 holds, small numerical extrema may occur. The goal of this section is to define an artificial antidiffusion operator F^* on the algebraic level by a nodal TVD type flux limiter. The application of the operator F^* results in the high-resolution scheme

$$M_L \frac{dU}{dt} = K^* U = LU + F^* U. \tag{5.44}$$

The algorithmic steps are similar to the scalar case 3.2.4:

1. Remove all antidiffusive fluxes from the high-order Galerkin discretization by applying the numerical diffusion tensor.
2. Compute the sums of positive/negative antidiffusive/diffusive fluxes to individual nodes in terms of characteristic variables.
3. Compute correction factors in characteristic variables.
4. Add the limited antidiffusion in conservative variables.

Since the construction of the low-order operator was described in section 5.1.5, it remains to derive the computation of the nodal correction factors, which control the limited antidiffusive fluxes applied to the low-order scheme. We may initialize the high-resolution operator by the low-order one. The transformation to characteristic variables is primarily performed to compute the sums of upwind and downwind contributions, while the artificial limited antidiffusion is inserted into the residual in conservative variables.The proposed algorithm for the computation of the antidiffusive correction is based on [47, 44] and has to be applied for each space dimension d:

- In a loop over edges $\{ij\}$:
    1. Compute the Roe averaged eigenvalues and matrices of left and right eigenvectors of the uni-directional Roe matrices.
    2. Transform the solution differences on the edge to the characteristic variables
    $$\Delta W_{ij}^{(d)} = R_{ij}^{(d)^{-1}}(U_i - U_j). \tag{5.45}$$
    3. Determine the upwind node I and the downwind node J for each characteristic field k:
        - If $a_{ij}^{(d)}\lambda_k^{(d)} \geq 0$:
        $$I = i, \qquad J = j, \qquad \Delta W_{IJ}^{(k)} = \Delta W_{ij}^{(k)}. \tag{5.46}$$
        - If $a_{ij}^{(d)}\lambda_k^{(d)} < 0$:
        $$I = j, \qquad J = i, \qquad \Delta W_{IJ}^{(k)} = -\Delta W_{ij}^{(k)}. \tag{5.47}$$
    4. Compute the raw diffusive/antidiffusive fluxes on the current edge
    $$F_{IJ}^{(k)} = |a_{ij}^{(d)}\lambda_k^{(d)}|\Delta W_{IJ}^{(k)}. \tag{5.48}$$
    5. For each characteristic field k update the sums of upwind and downwind edge contributions
    $$P_{I,k}^{+} = P_{I,k}^{+} + \max\{0, F_{IJ}^{(k)}\} \qquad P_{I,k}^{-} = P_{I,k}^{-} + \min\{0, F_{IJ}^{(k)}\}.$$
    Compute upper and lower bounds
    $$\begin{aligned} Q_{I,k}^{+} &= Q_{I,k}^{+} + \max\{0, -F_{IJ}^{(k)}\} & Q_{J,k}^{+} &= Q_{J,k}^{+} + \max\{0, F_{IJ}^{(k)}\} \\ Q_{I,k}^{-} &= Q_{I,k}^{-} + \min\{0, -F_{IJ}^{(k)}\} & Q_{J,k}^{-} &= Q_{J,k}^{-} + \min\{0, F_{IJ}^{(k)}\}. \end{aligned}$$
- In a loop over nodes:
    1. Evaluate the nodal correction factors
    $$R_{i,k}^{\pm} = \min\{1, \frac{Q_{i,k}^{\pm}}{P_{i,k}^{\pm}}\}. \tag{5.49}$$
- In a loop over edges $\{ij\}$:
    1. Repeat the first three steps of the first loop over edges.
    2. Compute the limited amount of antidiffusion for each characteristic direction k:
        - Compute the raw antidiffusive/diffusive fluxes
        $$F_{IJ}^{(k)} = |a_{ij}^{(k)}\lambda_k^{(d)}|\Delta W_{IJ}^{(k)}. \tag{5.50}$$

Determine the correction factors

$$\alpha_{ij}^{(k)} = \begin{cases} R_{I,k}^{+} & \text{if} \quad F_{IJ}^{(k)} \geq 0 \\ R_{I,k}^{-} & \text{if} \quad F_{IJ}^{(k)} < 0 \end{cases}. \tag{5.51}$$

- Apply the correction factors

$$F_{IJ}^{(k)} = \alpha_{ij}^{(k)} F_{IJ}^{(k)}. \tag{5.52}$$

3. Transform back to conservative variables

$$F_{IJ} = R_{ij} F_{IJ}. \tag{5.53}$$

4. Add the antidiffusive fluxes into the upwind and downwind nodes

$$(K^*U)_I = (K^*U)_I + F_{IJ}, \qquad (K^*U)_J = (K^*U)_J - F_{IJ}. \tag{5.54}$$

Obviously, the whole stabilization is free of parameters and can be generalized to arbitrary hyperbolic systems of conservation laws which allow for the determination of Roe averages. If this is not the case, the Roe matrices can be replaced by other types of averages, e. g. arithmetic averages. One also obtains a conservative scheme due to the construction of the diffusion and antidiffusion tensors. The high-resolution scheme investigated in the present thesis incorporates the flux function of Roe for the evaluation of the boundary integrals and the construction of the numerical diffusion and antidiffusion tensors, while the high-order space discretization in the interior of the domain remains independent of the Roe linearization.

In the following section we investigate the discretization of the interfacial source terms.

6 Treatment of Source Terms

While the discretization of hyperbolic terms is described in the chapters above, the discretization of source terms is still an open question. The source terms introduce a two-way coupling and give rise to an additional nonlinearity in the model. The presence of small particles causes the source terms relating to the drag force and heat exchange to dominate. This in turn leads to slow convergence when using implicit solvers and aggravates the already very restrictive stability constraints in explicit computations. To circumvent this problem, source terms are usually included into the two-fluid model by way of operator splitting, see among others [70, 82, 79]. The operator splitting approach makes it possible to develop independent solvers for both the hyperbolic terms and the source terms. In such an approach, the hyperbolic terms are usually discretized using explicit methods developed in the framework of single-phase equations. On the other hand the source terms call for an implicit solver due to their stiffness. At the same time, they can be integrated by a semi-analytical way [82, 51].

In the references cited above the Yanenko splitting of first order accuracy is employed since it removes the source terms and therefore the associated stability constraints completely from the equations accounting for the hyperbolic terms. The computation of the transport terms is the most time and memory consuming part of the simulation of these types of multiphase flows. Operator splitting techniques seem to be promising methods since they even for implicit approaches make it possible to solve for the hyperbolic terms of both phases separately. In transient computations this completely holds true, while operator splitting techniques are less efficient or even less accurate in steady-state computations. Although there are splitting techniques, which are unconditionally stable, they do not allow the solution to approach steady-state for moderate and large CFL numbers.

For the space discretization of hyperbolic equations with source terms an upwind approximation of source terms is sometimes proposed, similar to the upwind approximation of hyperbolic terms [6]. A much simpler approach has been successfully used in the references cited above, where the source terms are discretized in a pointwise way. The accuracy of a pointwise approximation is emphasized in [78]. A similar finite-element-like approximation is proposed in this thesis.

6.1 Operator Splitting

For the time integration of (6.15) one may apply operator splitting to circumvent the convergence and stability problems associated with the nonlinearity induced by the source terms. As a starting point, we recall the operator splitting that was applied to the two-fluid model in [25]. In transient computations a time integration scheme of second or higher order is usually applied. To preserve the overall accuracy in time-dependent computations, the use of second- or higher-order operator splitting schemes is required. Since this study focuses on stationary solutions, first order splitting techniques combined with backward Euler time stepping suffice for this purpose. The time integration of the hyperbolic part is described in detail in chapter 8.

6.1.1 Yanenko Splitting

A popular approach to solving (6.15) is the Yanenko splitting [82, 51, 79]

$$\frac{U^* - U^n}{\Delta t} + \nabla \cdot \mathbf{F}^* = 0 \tag{6.1}$$

$$\frac{U^{n+1} - U^*}{\Delta t} = S^{n+1}, \tag{6.2}$$

where the superscript n denotes the time level. In the first step, the numerical solution is advanced in time without taking the source terms into account. In the second step, the nodal values of the resulting solution U^* are corrected by adding the contribution of S^{n+1}.

For an explicit solver fitted to transient simulations, the Yanenko splitting is a good choice, although it is only first-order accurate. It turns out to be very robust and to be applicable to nearly arbitrary CFL numbers in practice. After the first step (6.1) the source term step (6.2) changes the intermediate solution U^* depending on the length of the time step, so that the final solution U^{n+1} does not satisfy the equations of the first subproblem. The solution will actually not approach steady-state since it depends on the length of the time step. The Yanenko splitting is therefore restricted to time-dependent flows and very small time steps.

6.1.2 Douglas-Rachford Splitting

To make sure that the splitting does not disturb solutions approaching a steady-state, we replace (6.1)–(6.2) by the Douglas-Rachford scheme [14]

$$\frac{U^* - U^n}{\Delta t} + \nabla \cdot \mathbf{F}^* = S^n \tag{6.3}$$

$$\frac{U^{n+1} - U^*}{\Delta t} = S^{n+1} - S^n, \tag{6.4}$$

which is known to be very robust, at least in the context of alternating direction implicit (ADI) iterative solvers for multidimensional problems.

The implicit correction in the second step does not change a converged stationary solution and allows the solution to approach steady-state. Moreover, the Douglas-Rachford splitting provides a closer link between the density and velocity of the particulate phase. This is another reason why it is preferred to the Yanenko splitting. Due to the above mentioned drawbacks of the Yanenko splitting, the Douglas-Rachford splitting is employed in this thesis.

6.1.2.1 Source Term Update

Let us focus our attention on the second step (6.4) of the Douglas-Rachford splitting since the first step (6.3) corresponds to the hyperbolic solver, which is described in chapter 8. It is followed by an implicit correction of the involved interface transfer terms. In this step, the drag force and heat exchange terms are discretized in a semi-implicit fashion. First, the velocities are updated by solving the linear system

$$\begin{aligned}(\alpha_p\rho_p)^* \frac{\mathbf{v}_p^{n+1} - \mathbf{v}_p^*}{\Delta t} &= \gamma_D^*(\mathbf{v}_g^{n+1} - \mathbf{v}_p^{n+1}) - \mathbf{F}_D^n && (6.5)\\ (\alpha_g\rho_g)^* \frac{\mathbf{v}_g^{n+1} - \mathbf{v}_g^*}{\Delta t} &= \gamma_D^*(\mathbf{v}_p^{n+1} - \mathbf{v}_g^{n+1}) + \mathbf{F}_D^n, && (6.6)\end{aligned}$$

where the superscript $*$ refers to the solution of system (6.3) and

$$\gamma_D^* = \frac{3}{4} C_D^* \frac{\rho_g^*}{d} \alpha_p^* |\mathbf{v}_g^* - \mathbf{v}_p^*|. \qquad (6.7)$$

Once the velocities have been updated, the changes in energy due to the interfacial drag and heat exchange are taken into account as follows:

$$\begin{aligned}(\alpha_p\rho_p)^* \frac{E_p^{n+1} - E_p^*}{\Delta t} &= \gamma_T^*(T_g^{n+1} - T_p^{n+1}) - \tilde{Q}_T^n && (6.8)\\ (\alpha_g\rho_g)^* \frac{E_g^{n+1} - E_g^*}{\Delta t} &= \gamma_T^*(T_p^{n+1} - T_g^{n+1}) + \tilde{Q}_T^n. && (6.9)\end{aligned}$$

The heat transfer coefficient γ_T^* and net source/sink $\tilde{Q}_T^n$ are given by

$$\gamma_T^* = \frac{Nu^* 6\kappa_g}{d^2}\alpha_p^*, \qquad \tilde{Q}_T^n = Q_T^n + \mathbf{v}_p^{n+1} \cdot \mathbf{F}_D^{n+1} - \mathbf{v}_p^n \cdot \mathbf{F}_D^n. \qquad (6.10)$$

Since mass transfer does not take place, there are no source terms in the continuity equations. Therefore, the effective densities $(\alpha\rho)_k^{n+1} := (\alpha\rho)_k^*$ remain unchanged.

Numerical experiments indicate that steady-state convergence of the Douglas-Rachford splitting can only be achieved for small CFL numbers. It is a well-known fact that decoupled solution strategies are unfavorable in steady-state computations due to their time step restrictions. In practice one has to make a decision in the trade-off between the low computational costs of segregated algorithms in comparison with strongly coupled methods and the much more restrictive time step constraints of the former family of methods. An implicit time stepping scheme without use of operator splitting, which offers the potential of unconditional stability and convergence, is preferable in steady-state simulations and is therefore developed in this thesis. Time integration as well as the fully coupled solution strategy are presented in chapter 8.

6.2 Finite Element Discretization

Let us consider system (2.68) and rewrite its weak formulation in the generic form

$$\int_\Omega \omega(\partial_t U + \nabla \cdot \mathbf{F})\, d\mathbf{x} = \underbrace{\int_\Omega \omega S\, d\mathbf{x}}_{=I_S} \tag{6.11}$$

for every admissible test function ω. In the expression above, the source terms are represented by I_S. The goal of this section is to evaluate I_S. A direct integration of that term involves numerical integration in each iteration step, which makes the integration very time consuming. To discretize the source terms in a finite-element-like way without numerical integration in each (pseudo) time step, we adopt the group finite element formulation, which was originally defined for hyperbolic terms [20]. The group finite element formulation is also employed for the discretization of the hyperbolic terms of the two-fluid model. Hence, we interpolate the source terms in the space of basis functions

$$S_h = \sum_i \varphi_i S_i, \tag{6.12}$$

where

$$S_i = \left(0,\quad -\mathbf{F}_{Di},\quad -\mathbf{v}_{p_i} \cdot \mathbf{F}_{Di} - Q_{Ti},\quad 0,\quad \mathbf{F}_{Di},\quad \mathbf{v}_{p_i} \cdot \mathbf{F}_{Di} + Q_{Ti}\right) \tag{6.13}$$

denotes the source terms evaluated at the corresponding node i. Substituting (6.12) into (6.11) yields semi-discretized equations of the form

$$M_C \frac{dU}{dt} = KU + M_C S, \tag{6.14}$$

where M_C denotes the consistent mass matrix and the matrix-vector product KU includes the discretized hyperbolic terms and boundary conditions as explained in sections 4, 5, and 7. At the same time, one replaces the consistent mass matrix by its lumped counterpart, which transforms (6.14) to

$$M_L \frac{dU}{dt} = KU + M_L S. \tag{6.15}$$

This type of source term discretization is closely related to the finite volume framework, and the source term integration by schemes which take advantage of operator splitting. The replacement of the consistent mass matrix with the lumped one enables a pointwise updating strategy in an operator splitting approach. Moreover, the lumped mass matrix prevents the birth of oscillations due to nonzero off-diagonal source term blocks. Such treatment of source terms is also feasible in a fully coupled approach, which is favored in this work since it does not violate the physical properties of the solution. It is a well known fact that the two-fluid equations do not conserve momentum and energy of each phase separately, in contrast to the mixture momentum and energy which are perfectly conserved (compare section 2.4). Equations (6.15) clearly feature this property. Furthermore, the positivity constraint is not affected by the source terms nor are numerical oscillations observed in the computational results.

The next section addresses the implementation of boundary conditions, which is one of the most challenging tasks in the development of any implicit hyperbolic solver and therefore also an important part of this thesis.

7 Boundary Conditions

A suitable implementation of boundary conditions is crucial for the convergence to steady-state in the presented framework of hyperbolic equations. The implementation of boundary conditions is a challenging task since information may travel in different directions depending on the sign of the corresponding eigenvalues. This may cause convergence problems if proper care is not taken. A very careful and accurate implementation is especially important to achieve convergence in steady-state computations. Waves may be artificially reflected at the boundary due to an improper specification of boundary values. Furthermore, the arising errors are transported into the interior of the domain by the incoming waves. This inhibits convergence to steady-state and may even cause divergence.

The implementation of boundary conditions can be generally divided into the strong enforcement and its weak counterpart. In a strong implementation, the boundary conditions are prescribed in a nodal sense on the algebraic level. Assume that the discretization yields a linear algebraic system of the form

$$Au = b \tag{7.1}$$

in each nonlinear iteration step. If a node i belongs to the boundary and a Dirichlet boundary condition $u_i = g_i$ has to be prescribed, the off-diagonal entries of the i-th row of the matrix A are deleted and the diagonal one is replaced by one. Finally, one overwrites the i-th entry of the right hand side b_i by g_i. Obviously, the solution of the linear system satisfies $u_i = g_i$. On the other hand, such ad-hoc changes of matrix entries influence the internodal mass exchange in the vicinity of the boundary and result in a deterioration of matrix properties and convergence rates or even divergence particularly in steady-state computations. Moreover, time step restrictions arise due to the explicit nature of that strategy. The explicitness mainly is caused by the deletion of off-diagonal matrix entries.

In the case of the Euler equations g_i typically depends on the solution, which requires a predictor-corrector algorithm in an implicit formulation [47, 60]. However, the predictor-corrector algorithm is semi-explicit as well. Trépanier et al. [93] prescribe fully implicit boundary conditions in the finite volume framework, where the boundary conditions are enforced by additional equations superimposed on the algebraic system, and avoid the explicitness arising by the deletion of matrix entries. The derivation and implementation of such boundary conditions is complicated and the authors observe a deterioration of convergence rates if the CFL number exceeds some upper bound. This upper bound may be problem-dependent and need adjusting, which is an unfavorable property and may be a consequence of the strong nature of boundary conditions.

The weak type of boundary conditions turns out to be much more stable and flexible. When boundary conditions are prescribed in a weak sense, only the boundary integral of the weak formulation is affected by the boundary conditions, while the volume integrals remain unchanged. This is similar to the boundary treatment, which is usually implemented in finite volume schemes. In the finite volume framework the boundary fluxes are directly overwritten by the imposed boundary conditions [84, 62, 86, 87]. The Neumann-type of boundary conditions, based on the weak formulation, can be incorporated into the matrix in a physical way. It improves the convergence rates and does not affect the matrix properties or give rise to stability restrictions in contrast to the strong type of boundary conditions. Unlike its strong counterpart, the weak formulation enforces boundary conditions in an integral sense. Consequently, the boundary conditions may not be satisfied exactly at each node. On the other hand, the fluxes over each boundary edge are influenced primarily by the neighboring edges. Hence, there is a local enforcement in some manner. Note that only the fluxes in the normal direction can be controlled with this kind of weak implementation. Otherwise, a penalty term should be added to the weak formulation.

Boundary conditions for hyperbolic systems are rarely discussed in the literature especially in the framework of finite elements. Selmin et al. overwrite the boundary integral by a lumped mass approximation [87], while Shapiro [84] and Lyra [57] directly change the boundary fluxes to incorporate the boundary conditions. A comparative study of solid wall boundary conditions for the Euler equations is given in [2] and a detailed description of boundary conditions for different hyperbolic systems is reported in [21]. Neumann-type boundary conditions were also implemented in the discontinuous Galerkin framework [18, 15]. To the author's best knowledge, there are no studies on boundary conditions for the coupled transport equations (4.1) or the two-fluid model (2.68). In [21] characteristic boundary conditions for a two-fluid model in 1D are briefly discussed, which take advantage of the hyperbolicity of the governing equations. This framework does not apply to the non-hyperbolic two-fluid model of particle-laden gas flows.

In contrast to finite volumes, finite element methods are node-based schemes and fluxes over the boundaries of the elements are not directly useful. Although it is possible to modify the boundary fluxes of the strong Galerkin formulation (5.1) by the imposed boundary conditions, it is more flexible and robust to simply overwrite the boundary integral of the weak Galerkin formulation (5.6) in the Neumann-sense. Therefore, the question of the imposition of boundary conditions reduces to the computation of the boundary flux in the boundary integrals (5.6) and (4.16).

To prescribe boundary conditions in the developed code, a Riemann problem is solved, either by the approximate Riemann solver of Roe or the exact Riemann solver of Toro [92], which is applicable to solid wall boundary conditions. For a strong enforcement of boundary conditions it was observed in [92, 60] that Toro's exact Riemann solver may offer the opportunity of convergence at very low CFL numbers and yields accurate results, even when other methods fail.

7.1 Ghost Nodes

Since the Euler equations may admit waves leaving and entering the domain in the normal direction simultaneously it is preferable to solve the Riemann problem on the boundary instead of simply overwriting the boundary fluxes. In order to define such a Riemann problem on the boundary a ghost node is added. Note that this ghost node is only used to solve the boundary Riemann problem, while no additional variables are introduced to the nonlinear system.

The context of ghost nodes can be interpreted as the finite element counterpart of ghost cells, which was quite successfully employed in the finite volume framework [52]. Typically in finite volume schemes a virtual cell is assumed on the outer side of the boundary and its state is defined so as to impose the boundary condition (figure 7.1 (a)). In this work it is proposed to introduce a virtual node or ghost node corresponding to each point located at the boundary, where a boundary condition has to be imposed. It is placed in the normal direction on the outer side of the boundary (figure 7.1 (b)). The corresponding ghost state is supposed to incorporate the boundary condition and it is defined in such a way that unphysical wave reflections at the boundary are avoided. Last but not least, the concept of ghost nodes is independent of the finite element space. For convenience the ghost node will be denoted by $\mathbf{x}_\infty$ and the point at the boundary, which is treated as an interior one, by $\mathbf{x}_i$. The corresponding states are referred to as U_∞ and U_i, respectively. Note that the framework of ghost nodes applies to arbitrary points at the boundary including quadrature points or boundary nodes. In this work a ghost node is typically associated with the quadrature points of the boundary integrals.

7.2 Weak Imposition of Boundary Conditions

The Galerkin discretizations (5.6) and (4.16) make it possible to incorporate the boundary conditions in a natural way by rewriting the boundary integrals. Substituting the group finite element approximation (4.11) into (5.6) and (4.16) one obtains

$$\sum_j M_{ij} \frac{dU_j}{dt} = \sum_j \left[\int_\Omega \varphi_j \nabla \varphi_i \, d\mathbf{x} \right] \cdot \mathbf{F}_j - \int_{\partial\Omega} \varphi_i \mathbf{n} \cdot \mathbf{F}_h \, ds \quad \forall i. \tag{7.2}$$

The imposition of weak boundary conditions for the Euler equations as well as the two-fluid model is based on the weak Galerkin discretization (7.2). We overwrite the fluxes $\mathbf{F}_h$ in the boundary integrals by their counterparts computed by the solution of a Riemann problem in the normal direction:

$$\int_{\partial\Omega} \varphi_i \mathbf{n} \cdot \mathbf{F}_h \, ds \qquad \longrightarrow \qquad \int_{\partial\Omega} \varphi_i \mathbf{n} \cdot \tilde{\mathbf{F}}_h \, ds. \tag{7.3}$$

The evaluation of the boundary integrals and the definition of the boundary fluxes $\tilde{\mathbf{F}}_h$ are delicate tasks. Moreover, a sufficiently accurate integration of the boundary terms is crucial for a satisfactory global accuracy since the errors due to improper boundary treatments are transported into the interior of the domain along the characteristics.

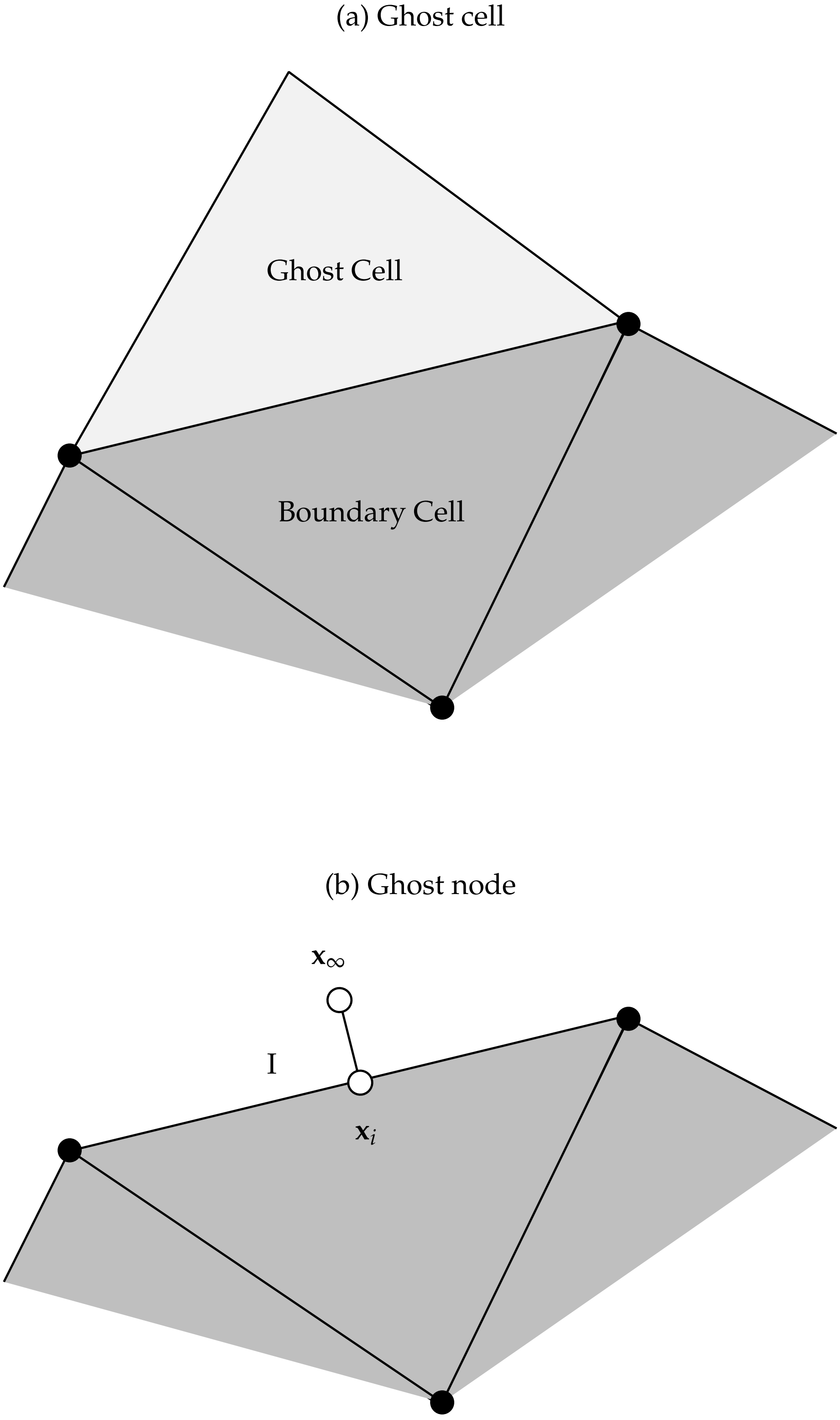


Figure 7.1: Ghost cell vs. ghost node

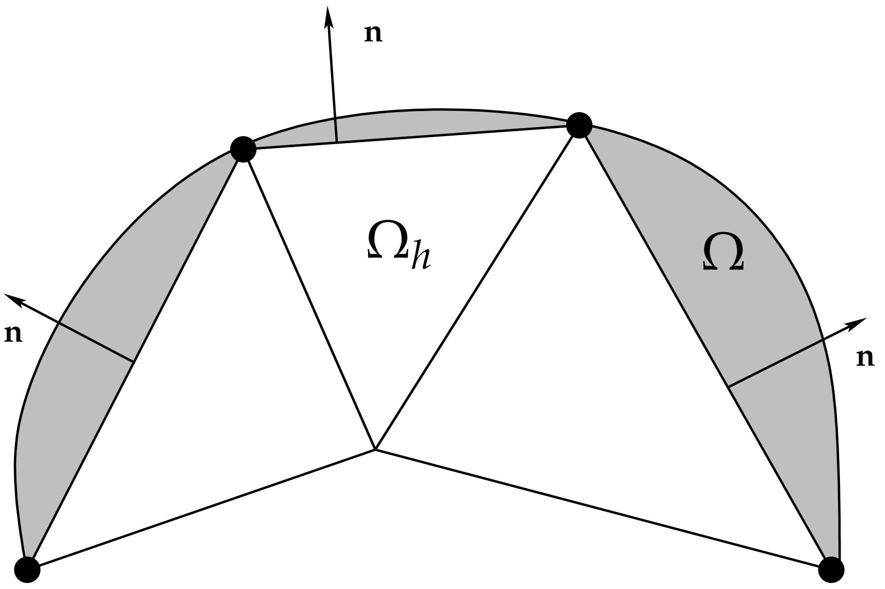


Figure 7.2: Physical vs. numerical domain

7.3 Definition of the Unit Outer Normal and Tangent

Several approaches are published in the literature to approximate the outward normal. Möller [60] proposes to determine the unit outer normal in each node by a weighted average of the normal vectors to the adjacent edges. Krivodonova and Berger [41] define the normal by the so-called curved boundary condition, which enables the approximation of an arbitrary curved boundary.

The problem of normal approximation is closely related to the definition of the boundary of the computational domain Ω_h, which may be different to the boundary of the physical domain Ω, see figure 7.2. In the present work the boundary is approximated by a linear polygon in terms of the edges adjacent to the boundary. This provides a unit outer normal, which is constant along every edge. The normal vector to the edge I with nodes $\mathbf{x}_i = [x_i, y_i]^T$ and $\mathbf{x}_j = \left[x_j, y_j\right]^T$ is determined by

$$\mathbf{n}_I = \frac{1}{\sqrt{(y_j - y_i)^2 + (x_i - x_j)^2}} \begin{bmatrix} y_j - y_i \\ x_i - x_j \end{bmatrix}. \tag{7.4}$$

For an interior/exterior domain j is the neighbor of i in counterclockwise/clockwise sense. Note that the order of nodes is crucial for the orientation of the normal vector. See also figure 7.3. The corresponding tangent is defined by

$$\boldsymbol{\tau}_I = \begin{bmatrix} -n_I^{(y)} \\ n_I^{(x)} \end{bmatrix}. \tag{7.5}$$

An advantage of the weak formulation is that the evaluation of the normal vector is not required in the boundary nodes, where it is usually undefined. Indeed, for the

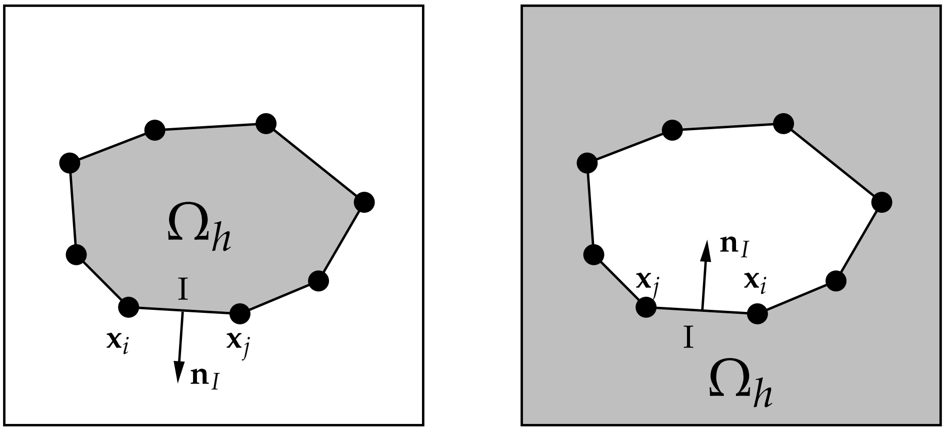


Figure 7.3: Ordering of boundary nodes for an interior/exterior domain

evaluation of the boundary integrals and boundary fluxes, it is sufficient to define an edge-wise normal. In this study, it is given by the normal to the current edge directed out of the domain. This can be easily extended to the more accurate approximation of Krivodonova and Berger [41].

7.4 Evaluation of the Boundary Integral

The boundary term

$$B_k = \int_{\partial\Omega} \varphi_k \mathbf{n} \cdot \tilde{\mathbf{F}}_h \, ds \tag{7.6}$$

related to boundary node k contains all the necessary information of the boundary condition in terms of the boundary flux $\tilde{\mathbf{F}}_h$. The term is equal to zero almost everywhere unless I is a boundary edge adjacent to node k. It therefore makes sense to compute the boundary contribution in a loop over all boundary edges by a sufficiently accurate quadrature rule. One computes the boundary integrals corresponding to nodes k,l of each boundary edge $I = \{kl\}$ in terms of

$$B_{k|_I} = \int_I \varphi_k \mathbf{n} \cdot \tilde{\mathbf{F}}_h \, ds, \qquad B_{l|_I} = \int_I \varphi_l \mathbf{n} \cdot \tilde{\mathbf{F}}_h \, ds. \tag{7.7}$$

In this study the two-point Gauss quadrature rule

$$\int_{-1}^{1} f(x)\, dx \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(-\frac{1}{\sqrt{3}}\right) \tag{7.8}$$

is transformed to the current edge and applied to the boundary integrals, although other choices are also feasible. Gauss quadrature rules are of open type. That is, their integration points are located in the interior of $[-1, 1]$. Hence, the normal vector is evaluated only in the interior of the boundary edges, where it is uniquely defined in a natural way. In contrast, node-based implementations or the strong type of boundary conditions require the unit outer normal at the boundary nodes where it has to be defined by artificial assumptions. On the other hand, the evaluation of the boundary flux

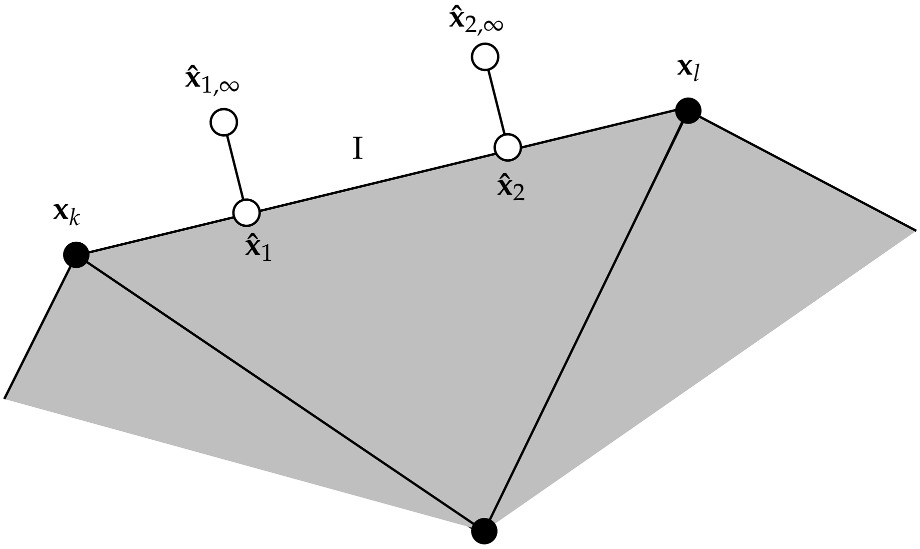


Figure 7.4: Quadrature points and ghost nodes

in the quadrature points is necessary. The fluxes depend on the given values of the conservative variables in a nonlinear way. However, the conservative variables are linear functions and they can easily be evaluated exactly in the quadrature points. For the two-point Gauss quadrature rule let the states at quadrature points $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ be given by $\hat{U}_1$ and $\hat{U}_2$, respectively. The application of (7.8) to the edge I yields

$$B_{k|_I} = \int_I \varphi_k \mathbf{n} \cdot \tilde{\mathbf{F}}_h \, ds \approx \frac{|I|}{2} \mathbf{n}_I \cdot \left(\varphi_k(\hat{\mathbf{x}}_1) \tilde{\mathbf{F}}_h(\hat{\mathbf{x}}_1) + \varphi_k(\hat{\mathbf{x}}_2) \tilde{\mathbf{F}}_h(\hat{\mathbf{x}}_2) \right) \tag{7.9}$$

and

$$B_{l|_I} = \int_I \varphi_l \mathbf{n} \cdot \tilde{\mathbf{F}}_h \, ds \approx \frac{|I|}{2} \mathbf{n}_I \cdot \left(\varphi_l(\hat{\mathbf{x}}_1) \tilde{\mathbf{F}}_h(\hat{\mathbf{x}}_1) + \varphi_l(\hat{\mathbf{x}}_2) \tilde{\mathbf{F}}_h(\hat{\mathbf{x}}_2) \right), \tag{7.10}$$

where $\tilde{\mathbf{F}}(\hat{\mathbf{x}}_1)$ and $\tilde{\mathbf{F}}(\hat{\mathbf{x}}_2)$ denote the boundary fluxes

$$\mathbf{n}_I \cdot \tilde{\mathbf{F}}(\hat{\mathbf{x}}_1) = F(\hat{U}_1, \hat{U}_{1,\infty}) \quad \text{and} \quad \mathbf{n}_I \cdot \tilde{\mathbf{F}}(\hat{\mathbf{x}}_2) = F(\hat{U}_2, \hat{U}_{2,\infty}) \tag{7.11}$$

corresponding to (7.49) and (7.54). This is illustrated in figure 7.4. The solution of the Riemann problem associated with the quadrature points as well as the flux Jacobians related to the boundary fluxes are provided below.

7.5 Euler Equations

The Euler equations admit waves moving in different directions if the signs of the eigenvalues differ. Therefore, some waves may enter the domain over the boundary, while simultaneously others leave the domain. Boundary conditions should only be imposed on the incoming waves, while the outgoing ones should remain unchanged to leave the boundary transparent to outgoing waves. Due to this fact, we prescribe characteristic boundary conditions and the ghost state in terms of the Riemann invariants. In the following sections we review the concept of Riemann invariants and describe the determination of boundary conditions in a general sense. Later on we turn to the solution of a boundary Riemann problem, which involves the formerly introduced concept of a ghost state. The solution of that problem is used to evaluate the boundary integral with flux (7.11).

7.5.1 Riemann Invariants

In order to derive Riemann invariants, one transforms the Euler equations to a local coordinate system consisting of the unit normal $\mathbf{n}$ and unit tangential $\boldsymbol{\tau}$ direction to the boundary. Note that without loss of generality the normal is defined by the unit normal vector directed out of the domain. A diagonalization of the resulting system of transformed Euler equations yields a system of four scalar transport equations for the Riemann invariants [84]

$$W = \{W_1, W_2, W_3, W_4\}. \tag{7.12}$$

It is governed by

$$\begin{aligned} \partial_t W_1 + (v_n - c)\partial_n W_1 &= 0 \\ \partial_t W_2 + v_n \partial_n W_2 &= 0 \\ \partial_t W_3 + v_n \partial_n W_3 &= 0 \\ \partial_t W_4 + (v_n + c)\partial_n W_4 &= 0 \end{aligned} \tag{7.13}$$

under the assumption a of local isentropic flow. The Riemann invariants are given by

$$W_1 = v_n - \frac{2c}{\gamma - 1}, \quad W_2 = c_v \log\left(\frac{P}{\rho^\gamma}\right), \quad W_3 = v_\tau, \qquad W_4 = v_n + \frac{2c}{\gamma - 1}, \tag{7.14}$$

for the corresponding eigenvalues

$$\lambda_1 = v_n - c, \qquad \lambda_2 = \lambda_3 = v_n, \qquad \lambda_4 = v_n + c \tag{7.15}$$

of the projected Jacobian tensor $\mathbf{n} \cdot \mathbf{A}$. The tangential and normal velocities are

$$v_n = \mathbf{n} \cdot \mathbf{v}, \qquad \text{and} \qquad v_\tau = \boldsymbol{\tau} \cdot \mathbf{v}, \tag{7.16}$$

respectively. For the sake of simplicity and without loss of generality W_2 is replaced by [84]

$$W_2 = \frac{P}{\rho^\gamma} \tag{7.17}$$

in the context of boundary conditions. A decision based on the sign of the corresponding eigenvalue of each characteristic variable is made to prescribe boundary conditions. If the eigenvalue is positive, then the wave leaves the domain and the Riemann invariant remains unchanged. Otherwise, in the case of a negative eigenvalue, a boundary condition has to be imposed on the wave.

7.5.2 Inflow and Outflow Boundary conditions

Inlet and outlet boundary conditions are usually prescribed based on the free stream conditions. It is also possible to fix other variables, like the pressure at a subsonic outlet or density, tangential velocity, and pressure at a subsonic inlet. The reader is referred to [32] for a detailed analysis.

7.5.3 Supersonic Inflow and Outflow Boundary Conditions

A supersonic inflow/outflow is characterized by $|v_n| > c$. If additionally $v_n > 0$, all eigenvalues are positive, or in other words all waves leave the domain. Therefore, no boundary condition is needed. This corresponds to a supersonic outlet and is the simplest case of boundary condition. Otherwise, in the case of $v_n < 0$, all eigenvalues have negative sign and all waves enter the domain, which characterizes a supersonic inlet. Hence, boundary conditions have to be imposed on all waves. Alternatively in this case one may prescribe boundary conditions in terms of conservative variables.

7.5.4 Subsonic Inflow and Outflow Boundary Conditions

If $|v_n| < c$, the flow is of subsonic nature. Consequently, either the first three eigenvalues are negative ($-c < v_n < 0$) and the fourth is positive, which corresponds to a subsonic inlet, or the first eigenvalue ($0 < v_n < c$) is the only one that has a negative sign. The latter case is related to a subsonic outlet and the former corresponds to a subsonic inlet. In the case of a subsonic inlet the first three Riemann invariants must be prescribed, while at a subsonic outlet only the first Riemann invariant has to be imposed due to the negative sign of the corresponding eigenvalue. Nevertheless, in both cases the flow behavior depends on both the incoming and outgoing characteristics and it is therefore impossible to prescribe boundary conditions in terms of conservative variables.

7.5.5 Free Stream Boundary Conditions

Subsonic boundary conditions are much more complicated then their supersonic counterparts since one has to compute the boundary flux that depends on four variables. However, there are less than four Riemann invariants defined by the boundary conditions. The remaining Riemann invariants correspond to outgoing waves and must be extracted from the solution.

Free stream boundary conditions can be implemented as follows [84]. Assume, that a set of free stream invariants

$$W_\infty = \{W_{\infty,1}, W_{\infty,2}, W_{\infty,3}, W_{\infty,4}\} \tag{7.18}$$

is specified. In the case of subsonic normal flow conditions, the boundary values depend on the free stream invariants as well as the outgoing Riemann invariants given by the current solution. Let the superscript * denote the state taken from the solver that is the current solution without taking into account the boundary conditions. The superscript ** refers to the final state. The algorithm below can be generally applied to determine free stream boundary values. It will be applied in section 7.5.9 to compute the ghost state required by (7.11). In this case, $U_\infty = U^{**}$ corresponds to the ghost state. Free stream boundary conditions can be computed from the following steps:

1. Compute the Riemann invariants of the current solution
$$W^* = \{W_1^*, W_2^*, W_3^*, W_4^*\}. \tag{7.19}$$
2. Define the final set of invariants W^{**} either by the predicted invariants or the free stream ones, depending on the signs of the corresponding eigenvalues λ_k
$$W_k^{**} = \begin{cases} W_k^* & \text{if} \quad \lambda_k \geq 0 \\ W_{\infty,k} & \text{if} \quad \lambda_k < 0 \end{cases} \tag{7.20}$$
for each characteristic field k.
3. Transform W^{**} back to conservative variables:
The speed of sound
$$c^{**} = \frac{\gamma - 1}{4}(W_4^{**} - W_1^{**}) \tag{7.21}$$
is given in terms of the difference of the first and the fourth Riemann invariant, and hence the primitive variables pressure, density, and velocity are determined by
$$\begin{aligned} \rho^{**} &= \left(\frac{c^{**2}}{\gamma W_2^{**}}\right)^{\frac{1}{\gamma-1}} && (7.22) \\ P^{**} &= \frac{c^{**2}\rho^{**}}{\gamma} && (7.23) \\ \mathbf{v}^{**} &= \frac{W_1^{**} + W_4^{**}}{2}\mathbf{n} + W_3^{**}\boldsymbol{\tau}. && (7.24) \end{aligned}$$
A few algebraic manipulations transform the boundary state to the final conservative variables
$$\begin{aligned} \rho^{**} &= \left(\frac{c^{**2}}{\gamma W_2^{**}}\right)^{\frac{1}{\gamma-1}} && (7.25) \\ (\rho\mathbf{v})^{**} &= \rho^{**}\left[\frac{W_1^{**} + W_4^{**}}{2}\mathbf{n} + W_3^{**}\boldsymbol{\tau}\right] && (7.26) \\ (\rho E)^{**} &= \frac{P^{**}}{\gamma - 1} + \rho\frac{|\mathbf{v}^{**}|^2}{2}. && (7.27) \end{aligned}$$
4. Prescribe boundary conditions.

In a practical implementation one may prescribe free stream invariants at each quadrature point of the inlets and outlets a priory in a preprocessing step. Then the algorithm computes the characteristic direction and determines the desired state automatically during the simulation for the subsonic and supersonic inlets/outlets. Such kind of boundary conditions is transparent to outgoing waves since the algorithm uses the solution-dependent invariants for outgoing waves instead of the a priory defined ones, which are taken to specify incoming waves. Free stream boundary conditions as defined in table 1.1 are employed in the computations reported in this thesis if not stated otherwise.

Note that it is also possible and sometimes more physical to prescribe primitive variables at subsonic boundaries [32] instead of free stream values.

7.5.6 Pressure Outlet

Due to physical reasons, it may be preferable to prescribe the pressure P_{out} at a subsonic outlet instead of the first Riemann invariant. In this case boundary conditions have to be imposed on the first Riemann invariant only. The exit pressure P_{out} can be fixed by replacing the first Riemann invariant by [84]

$$W_{\infty,1} = -\frac{4}{\gamma-1}\sqrt{\frac{\gamma P_{out}}{\rho^*}\left(\frac{P^*}{P_{out}}\right)^{\frac{1}{\gamma}}} + W_4^* \tag{7.28}$$

in step two of the algorithm in section 7.5.5.

7.5.7 Pressure-Density Inlet

In the case of a subsonic inlet, density, pressure and tangential velocity may be prescribed and other choices are also possible [32]. Let the boundary conditions be given in terms of P_{in}, ρ_{in}, and ${v_\tau}^{in}$. The first Riemann invariant can be determined by subtracting it from the fourth one

$$W_1^{**} = W_4^{**} - \frac{4c^{**}}{\gamma-1} = W_4^* - \frac{4c^{**}}{\gamma-1}, \tag{7.29}$$

where the speed of sound

$$c^{**} = \sqrt{\frac{\gamma P_{in}}{\rho_{in}}} \tag{7.30}$$

is given in terms of the imposed boundary condition. An addition of the first and the fourth Riemann invariant yields an equation for the normal velocity

$$v_n^{**} = \frac{W_1^{**} + W_4^{**}}{2} = \frac{W_1^{**} + W_4^*}{2}. \tag{7.31}$$

Therefore, the first three Riemann invariants are defined as

$$W_1^{**} = W_4^* - \frac{4c^{**}}{\gamma - 1} \tag{7.32}$$

$$W_2^{**} = \frac{P_{in}}{\rho_{in}^{\gamma}} \tag{7.33}$$

$$W_3^{**} = v_\tau{}^{in} \tag{7.34}$$

and the boundary state in conservative variables is consequently determined by

$$\rho^{**} = \rho_{in} \tag{7.35}$$

$$(\rho\mathbf{v})^{**} = \rho_{in}\left[\frac{W_1^{**} + W_4^*}{2}\,\mathbf{n} + v_\tau{}^{in}\,\boldsymbol{\tau}\right] \tag{7.36}$$

$$(\rho E)^{**} = \frac{P_{in}}{\gamma - 1} + \rho_{in}\frac{|\mathbf{v}^{**}|^2}{2}. \tag{7.37}$$

Hence, it suffices to overwrite the first three Riemann invariants in the second step of the algorithm in section 7.5.5 by

$$W_{\infty,1} = W_4^* - \frac{4c^{**}}{\gamma - 1} \tag{7.38}$$

$$W_{\infty,2} = \frac{P_{in}}{\rho_{in}^{\gamma}} \tag{7.39}$$

$$W_{\infty,3} = v_\tau{}^{in}. \tag{7.40}$$

7.5.8 Wall Boundary Conditions

In an inviscid flow mass and energy fluxes through the boundary are zero, while the fluxes parallel to the wall are not affected. In other words, the normal velocity vanishes

$$\mathbf{n} \cdot \mathbf{v} = 0. \tag{7.41}$$

Since the solid wall boundary condition prevents the flow from penetrating through the wall, it is called free-slip or no-penetration condition. This is different from viscous flow dynamics, where a no-slip condition with $\mathbf{v} = 0$ is satisfied due to the viscous stresses. The implementation of solid wall boundary conditions is a challenging task in computational fluid dynamics and carried out in a weak sense in this thesis. In [2] Balakrishnan and Fernandez compare different types of solid wall boundary implementations. It turns out that strong, weak and mixed implementations yield almost the same results.

The easiest way to constrain the normal velocity to zero, is to overwrite the flux in the boundary integral by

$$\mathbf{n} \cdot \tilde{\mathbf{F}}_h \Big|_{wall} = \begin{bmatrix} 0 \\ n^{(x)}P \\ n^{(y)}P \\ 0 \end{bmatrix}, \tag{7.42}$$

where the pressure serves as a kind of source term to enforce the free-slip condition. This treatment does not involve the solution of a Riemann problem and is refered to as zero-flux condition. Moreover, in some cases it provides increased robustness in comparison with the wall boundary type to be described below. Nevertheless, this boundary treatment was successfully applied e. g. in [87, 15, 2]. Note that other implementations of the wall boundary condition are possible , see e. g. [2, 47].

A more physical constraint is the so-called mirror or reflection condition since it mimics a continuous model of the ideal collision of a spherical particle with a solid wall. Moreover, this condition requires an additional ghost node (compare section 7.1) and enables the application of a Riemann solver at the boundary, which takes into account the characteristic direction of the flow. It yields very accurate results [92, 60] and is consistent with the interior discretization and the inlet/outlet boundary conditions. To implement the mirror condition the normal velocity is reflected at the boundary, while the tangential velocity, density and pressure or total energy remain unchanged. Adopting the notation introduced in section 7.1 the ghost state is defined by

$$\rho_\infty = \rho_i, \qquad v_{n\infty} = -v_{ni}, \qquad v_{\tau\infty} = v_{\tau i}, \qquad E_\infty = E_i. \tag{7.43}$$

This can be transformed to conservative variables

$$\begin{aligned} \rho_\infty &= \rho_i \\ (\rho\mathbf{v})_\infty &= \rho_\infty(-v_{ni}\mathbf{n} + v_{\tau i}\boldsymbol{\tau}) \\ (\rho E)_\infty &= (\rho E)_i. \end{aligned} \tag{7.44}$$

Note that the ghost state can also be defined in terms of the primitive varibles

$$\rho_\infty = \rho_i, \qquad v_{n\infty} = -v_{ni}, \qquad v_{\tau\infty} = v_{\tau i}, \qquad P_\infty = P_i, \tag{7.45}$$

which yields almost the same solution. The results computed by the mirror condition and (7.42) are similar. It is shown by the numerical results that (7.42) sometimes provides increased robustness, while the mirror condition represents a slightly more physical boundary condition. The mirror condition is applied in this thesis unless otherwise stated and both possibilities are compared in a numerical study in section 9.6.

7.5.9 Solution of the Boundary Riemann Problem

In order to solve the Riemann problem associated with a projection of the Euler equations onto the unit outer normal either an approximate or exact Riemann solver is necessary to compute the boundary state. To avoid artificial reflections of waves at the boundary, the interior state U_i in the sense of section 7.1 is taken from the hyperbolic solver without any changes, while the ghost state U_∞ is computed based on both the Riemann invariants of the interior one and the boundary condition. If an eigenvalue is positive, that is a wave leaves the domain, the related Riemann invariant is taken from the interior node. Otherwise, in the case of a negative eigenvalue, a boundary condition has to be imposed on the wave and the free stream Riemann invariant is taken. Once the ghost state is fixed, the Riemann problem between the inner node and the ghost node in normal direction can be solved. One may use the algorithm reported in section 7.5.5 to determine the ghost state as follows:

- Compute the invariants at the interior node i of the current solution
$$W^* = W^*(U_i). \tag{7.46}$$
- Determine the final Riemann invariants for each characteristic field
$$W_k^{**} = \begin{cases} W_k^* & \text{if} \quad \lambda_k^* \geq 0 \\ W_{\infty,k} & \text{if} \quad \lambda_k^* < 0 \end{cases} \tag{7.47}$$
- Transform back to conservative variables (see step 3 of the algorithm in section 7.5.5) and define the ghost state by
$$U_\infty = U^{**}. \tag{7.48}$$
- Solve the Riemann problem by Roe's flux formula (7.49) and compute the boundary flux $\mathbf{n} \cdot \tilde{\mathbf{F}}_h = F(U_i, U_\infty)$ at the intermediate state.
- Insert $\mathbf{n} \cdot \tilde{\mathbf{F}}_h$ into the boundary integral.

In this study the Riemann problem is solved approximately by means of Roe's Riemann solver although other choices are also possible. This treatment of the boundary flux is consistent with the inner discretization and makes it possible to compute a Jacobian matrix, which is important for the computation of stationary solutions by implicit schemes. This issue is addressed later. For the wall boundary conditions an exact Riemann solver presented by Toro [92] will be employed for comparison. The Riemann solver takes into account the direction of information propagation and avoids artificial non-physical effects like negative pressures or densities in the vicinity of the boundary particularly at high CFL numbers. This sometimes prevents the algorithm from divergence in comparison with an ad-hoc specification of boundary conditions, where the flux associated with the ghost node is directly substituted into the boundary term. In other words, the Riemann solver automatically chooses the upwind direction. To employ the approximate Riemann solver, the flux formula of Roe

$$\mathbf{n} \cdot \tilde{\mathbf{F}}_h = F(U_i, U_\infty) = \frac{1}{2}\mathbf{n} \cdot (\mathbf{F}(U_i) + \mathbf{F}(U_\infty)) - \frac{1}{2}|A_{i\infty}^n|(U_\infty - U_i) \tag{7.49}$$

is simply substituted into the boundary integral. In the equation above,

$$A_{i\infty}^n = \mathbf{n} \cdot \mathbf{A}_{i\infty} \tag{7.50}$$

is the Jacobian tensor evaluated in the normal direction and

$$|A_{i\infty}^n| = R_{i\infty}^n |\Lambda_{i\infty}^n| \, [R_{i\infty}^n]^{-1} \tag{7.51}$$

corresponds to (5.31). Obviously, (7.49) is Roe's 1D flux formula projected onto the outer normal.

7.6 Two-Fluid Model

The two-fluid model consists of the Euler equations written in terms of the effective density and the pressureless transport equations, which are linked by algebraic source terms. The flux Jacobians of the coupled system lack diagonalizability since an incomplete set of seven independent eigenvectors exists. Therefore, it is neither possible nor necessary to prescribe characteristic boundary conditions for the coupled two-fluid equations. As a matter of fact, it suffices to implement boundary conditions for the equations governing each phase separately.

Both phases involve similar boundary integrals. Therefore, the computation of the boundary integral is exactly the same for both phases, but the treatment of the boundary fluxes differs. The Euler equations admit three waves moving at different speeds and in different directions, which necessitates the specification of characteristic boundary values in terms of the Riemann invariants. In contrast, the equations modeling the particulate phase only admit one wave moving with the fluid velocity. Hence, the whole information propagates with the particle velocity. This enables the specification of boundary values in terms of conservative variables rather than Riemann invariants.

7.6.1 Inlet and Outlet Boundary Conditions

Without loss of generality we assume that

$$v_{p,n}\, v_{g,n} > 0, \tag{7.52}$$

where $v_{p,n} = \mathbf{n} \cdot \mathbf{v}_p$ and $v_{g,n} = \mathbf{n} \cdot \mathbf{v}_g$ are the normal velocities of the particles and gas, respectively. In other words, the case that one phase enters the domain while the other phase leaves the domain simultaneously is excluded from the scope of this thesis. Note that this simplifies notation, but does not limit the computational scheme.

A part of the boundary is called inlet if the normal velocities of both phases satisfy

$$v_{p,n} < 0 \quad \text{and} \quad v_{g,n} < 0, \tag{7.53}$$

otherwise it is referred to as an outlet. Inlet and outlet boundary conditions for the gas phase are the same as described in section 7.5 for the effective density $\rho = \alpha_g \rho_g$ and it suffices to discuss boundary conditions for the particulate phase in the present section. Consequently, the inflow and outflow can be subsonic or supersonic depending on the Mach number. In contrast, the wave of the particulate phase enters the domain at an inlet, which requires the specification of the complete vector of conservative variables. At an outlet, the wave leaves the domain and no information has to be prescribed. To simplify notation, the index denoting the particulate phase is neglected in the following formulas.

The restriction to one wave simplifies the solution of the boundary Riemann problem of the particulate phase. Similar to section 7.5.9 the flux formula of Roe is given by

$$\mathbf{n} \cdot \tilde{\mathbf{F}}_h = F(U_i, U_\infty) = \frac{1}{2}\mathbf{n} \cdot (\mathbf{F}(U_i) + \mathbf{F}(U_\infty)) - \frac{1}{2}|A^n_{i\infty}|(U_\infty - U_i) \tag{7.54}$$

with the flux tensor $\mathbf{F}$ defined by (4.4). Due to scalar dissipation the matrix $|A^n_{i\infty}|$ exhibits a diagonal structure

$$|A^n_{i\infty}| = diag\{d_{i\infty}, d_{i\infty}, d_{i\infty}, d_{i\infty}\}, \quad d_{i\infty} = \max\{|\mathbf{n} \cdot \mathbf{v}_i|, |\mathbf{n} \cdot \mathbf{v}_\infty|\} \tag{7.55}$$

in contrast to its counterpart corresponding to the Euler equations. This is similar to the scalar upwind-formulation of each equation. In order to clarify the last statement, let us consider a local linearization of the characteristic speed

$$\hat{v} = \begin{cases} \mathbf{v}_i \cdot \mathbf{n} & \text{if} \quad \mathbf{v}_i \cdot \mathbf{n} \geq 0 \\ \mathbf{v}_\infty \cdot \mathbf{n} & \text{if} \quad \mathbf{v}_i \cdot \mathbf{n} < 0 \end{cases} \tag{7.56}$$

of the particulate phase. Due to this assumption, the boundary flux (compare to (7.49)) is given by (the non-conservative formulation)

$$\begin{aligned} F_{lin}(U_i, U_\infty) &= \frac{\hat{v}}{2}(U_i + U_\infty) - \frac{\hat{v}}{2}(U_\infty - U_i) \\ &= \frac{\hat{v} + |\hat{v}|}{2}U_i + \frac{\hat{v} - |\hat{v}|}{2}U_\infty \\ &= \begin{cases} U_i\hat{v} & \text{if} \quad \hat{v} \geq 0 \\ U_\infty\hat{v} & \text{if} \quad \hat{v} < 0 \end{cases} \end{aligned} \tag{7.57}$$

and is clearly equivalent to the one-dimensional upwind approximation. The boundary flux (7.54) provides a quite similar treatment, where $d_{i\infty}$, given by equation (7.55), mimics an approximation of the characteristic speed, which is consistent with the inner discretization. Considering the last arguments it is sufficient to define the ghost state by the values of the interior state in the case of an outlet or by the imposed boundary condition at an inflow part of the boundary.

7.6.2 Wall Boundary Conditions

The wall boundary condition

$$v_{p,n} = 0 \tag{7.58}$$

should also be imposed on the equations governing the particulate phase. Since they do not involve a pressure gradient in contrast to the Euler equations, it is insufficient to apply the concepts reported in section 7.2. The wall boundary condition cannot be enforced just by canceling the boundary integral arising from the Galerkin discretization. Such an implementation does not inhibit fluxes penetrating through the walls. In the case of the Euler equations the nonzero pressure serves as a kind of source term, which prevents nonzero normal fluxes. Due to the lack of pressure the volume integrals may become large in comparison with the boundary integrals, which may also be a reason for the penetration of particles through the wall.

A simple way to circumvent this problem is to enforce flow tangency after each iteration by subtracting the normal components of the momentum equations

$$(K^*U)_i = (K^*U)_i - \left((K^*U)_i \cdot \begin{bmatrix} 0 \\ \mathbf{n}_i \\ 0 \end{bmatrix} \right) \begin{bmatrix} 0 \\ \mathbf{n}_i \\ 0 \end{bmatrix} \tag{7.59}$$

at any wall boundary node i in a strong sense. This conflicts with the boundary conditions imposed on the gas phase and gives rise to spurious boundary layers due to the inconsistent wall boundary treatment of both phases. On the other hand, the weak implementation of boundary conditions provides superior robustness, faster convergence, and accuracy in steady-state computations.

We therefore add an additional penalty term to the weak form of the momentum equations of both phases

$$\text{penalty} := -\sigma \int_{\Gamma_{wall}} \omega |v_n| \rho^2 v_n \mathbf{n} \, ds, \tag{7.60}$$

where $\sigma \gg 1$ is a large positive penalty parameter and ω represents the test function. The integration is carried out over the wall boundary and the penalty term, which appears on the right hand sides of the momentum equations, is set to zero elsewhere. Penalty terms were successfully applied to the incompressible Navier-Stokes equations [37] and reactive bubbly flows [42]. For the Euler equations a penalty term very similar to (7.60) was proposed in [13] to enforce the wall boundary condition. A more theoretical analysis of penalty techniques can be found in [94, 83]. Moreover, the performance of those terms in the framework of (scalar) transport equations as well as the incompressible Navier-Stokes equations was established in [5]. Although the free-slip condition can be prescribed much easier for the gaseous phase, the particle equations require such a penalty term. It is also added to the gas momentum equations to equalize the treatment of both phases and to avoid boundary layers, which may arise due to different boundary implementations. Nevertheless, the no-penetration condition

$$\mathbf{v} \cdot \mathbf{n} = 0 \tag{7.61}$$

can be substituted into the boundary integrals of each phase. Due to this fact, the boundary integrals of the particle equations related to solid walls vanish and the corresponding terms of the gas equations simplify to (7.42). Obviously, any weak solution of the governing equations also satisfies the modified equations with the additional penalty term. On the other hand, a weak solution of the modified equations satisfies the free-slip condition as well as the governing equations since convergence enforces the penalty term to vanish and the free-slip condition to be satisfied.

The penalty term turns out to be very large and stiff due to the large penalty parameter σ. This gives rise to rather large errors in the early stages of a simulation since the free-slip condition is usually not yet satisfied initially. Care must therefore be taken in the design of the preconditioner to achieve convergence. Additionally, the abrupt change in the normal velocity due to an initial solution violating the free-slip condition may cause divergence. This is not a drawback of the penalty term, the problem is caused by physical reasons since such an impulsive start is physically impossible [57].

The initially large errors decrease significantly by several orders of magnitude during a few iterations since the normal velocity vanishes after a few iteration steps. However, a converged solution does not depend on the way it was computed. Therefore, one may start the simulation with a small penalty parameter and increase it during the first iterations. Such a procedure is closely related to the underrelaxation of the wall boundary condition proposed by Lyra [57]. Note that the penalty term is also applicable to transient computations since the initial solution satisfies the wall boundary condition by the physical reasons pointed out above.

8 Fully Coupled Implicit Time Integration

It is one of the main goals of this study to design an implicit algorithm for the computation of stationary solutions to the Euler equations as well as the two-fluid model of particle-laden gas flows. Implicit schemes were rarely used in compressible CFD algorithms in the past. During recent years, there has been an increasing effort to design implicit schemes for the compressible Euler equations [93, 62, 60, 15, 18, 7]. To the author's best knowledge, there is no publication of an implicit discretization of the two-fluid equations. In spite of their infrequent use implicit methods are favorable compared to explicit ones because of the CFL restriction of the latter family of methods. Particularly in real life applications with complicated (or highly anisotropic) unstructured meshes created by automatic mesh generators and/or adaptivity [60], the spatial length scales h and CFL numbers

$$CFL = \lambda_{max} \frac{\Delta t}{h}, \tag{8.1}$$

where λ_{max} corresponds to the maximum wave speed, may differ by several orders of magnitude. Therefore, the stability constraint may be dictated by a few very small cells, while the remaining part is less restricted. This hampers the performance of explicit schemes and calls for implicit algorithms, which are more stable or independent of stability constraints. In practical applications a very restrictive CFL bound is caused by the usually large stiffness of the source terms of the two-fluid model.

Due to their algorithmic simplicity and low memory requirements, explicit schemes are often employed in the compressible flow community and there are some approaches to deal with the stiffness arising at the sonic point and in the low Mach number regime. The first one is local preconditioning [56]. In this case a preconditioner matrix corresponding to modified degrees of freedom is applied to the governing equations, unfortunately time accuracy is completely lost. Only a fully converged steady-state solution satisfies the governing equations. There are several types of preconditioners for the different flow regimes. This makes the approach less flexible in practical computations where both low Mach numbers and sonic points occur (compare the computational results reported in part III). Moreover, local preconditioning does not circumvent the stability restriction of explicit schemes. Another approach is local time stepping, where the same CFL number is applied to each node or perhaps even to each characteristic field. This can easily be implemented and it is also a handy tool, particularly in the start-up process of an implicit algorithm, where the CFL number may be increased step by step. Last but not least there are acceleration techniques like multigrid [30, 56] and residual smoothing [86].

Implicit schemes have the potential of being unconditionally stable without CFL restrictions if they are designed in a proper way. In particular, a suitable implicit treatment of boundary conditions is required. Since the convergence to steady-state depends on the propagation speed of the error waves, large CFL numbers accelerate the convergence to steady-state. This is primarily important at the sonic point and in low Mach number regions, where wave speeds are close to zero. Moreover, the implicit Euler approach corresponds to upwinding in time and therefore enjoys very useful numerical properties. This makes the backward Euler scheme a favorable choice in steady-state computations and it is therefore applied in this study. On the other hand, nonlinear systems must be solved and the computation of the nonlinear preconditioner is a challenging task. Therefore, the implementation of implicit schemes is far from being trivial. In this work a semi-implicit approach is employed to circumvent the computationally expensive nonlinear iterations. Implicit and semi-implicit schemes require a very powerful linear solver since the Jacobians usually lack diagonal dominance at least at high CFL numbers. This has an adverse effect on the convergence of many iterative solvers.

It is common practice to solve the arising nonlinear systems by a (pseudo-) time stepping scheme and march the solution to steady-state. In this case, the time step can be interpreted as an underrelaxation, where an infinite time step corresponds to a direct solution of the stationary problem without underrelaxation.

Boundary conditions are an important part of the implicit solver. The deletion of off-diagonal matrix blocks cancels mass fluxes between the boundary and adjacent nodes. This makes the commonly used strong enforcement of boundary conditions explicit and introduces a CFL restriction, which turns out to be very restrictive. Dolejší and Feistauer [15] observe a stability restriction of $CFL \lesssim 6$ for an explicit implementation of weak wall boundary conditions, while the stability is significantly improved for a semi-implicit version up to $CFL \approx 100$. Hence, an implicit treatment of boundary conditions, which is built into the preconditioner, is crucial for the stability of the implicit algorithm. This requirement is matched by the weak implementation of boundary conditions examined in this thesis.

8.1 Linearized Backward Euler Scheme

The spatially discretized Euler equations and the two-fluid model can be expressed in the condensed form

$$M_L \frac{dU}{dt} = F + S, \tag{8.2}$$

where U is the vector of conservative variables, F is the flux vector, and S is the source term of the two-fluid model. The case of the Euler equations is included by $S = 0$. This system of ordinary differential equations can be integrated in time by the backward Euler scheme

$$M_L \frac{U^{n+1} - U^n}{\Delta t} = F^{n+1} + S^{n+1} \tag{8.3}$$

with a time step of length Δt. The superscript n refers to time level n and the result is a nonlinear system of algebraic equations, which calls for nonlinear iterations in each (pseudo-) time step. Nonlinear iterations are computationally very expensive. To circumvent this problem, one assumes sufficient smoothness and linearizes the equations around the current solution U^n by a Taylor series expansion of the fluxes

$$F^{n+1} = F^n + \left(\frac{\partial F}{\partial U}\right)^n (U^{n+1} - U^n) + \mathcal{O}(\|U^{n+1} - U^n\|^2) \tag{8.4}$$

and the source terms

$$S^{n+1} = S^n + \left(\frac{\partial S}{\partial U}\right)^n (U^{n+1} - U^n) + \mathcal{O}(\|U^{n+1} - U^n\|^2). \tag{8.5}$$

A substitution of equations (8.4) and (8.5) into the nonlinear equations (8.3) leads to a linear algebraic system

$$\left[\frac{M_L}{\Delta t} - \left(\frac{\partial F}{\partial U} + \frac{\partial S}{\partial U}\right)^n\right] (U^{n+1} - U^n) = F^n + S^n. \tag{8.6}$$

Due to the linearizations (8.4) and (8.5) the latter scheme is time accurate of first order and applicable to stationary as well as transient flows, although a time integration scheme for transient flows should be at least second order accurate. Furthermore, nonlinear iterations are avoided and merely a linear system of algebraic equations has to be solved at each time level. Note that due to the homogeneity property the semi-implicit time marching scheme remains conservative, even if it is applied to transient problems.

At first glance, this scheme seems to be conditionally stable since it is semi-implicit. On the other hand, there is strong numerical evidence that the semi-implicit scheme based on the backward Euler method is unconditionally stable if the initial values are sufficiently accurate.

8.2 Newton's Method

Suppose, that both the fluxes and the source terms are differentiable with respect to the conservative variables. Setting the time step (or the CFL number) of the semi-implicit time stepping scheme to infinity, one recovers Newton's method

$$-\left(\frac{\partial F}{\partial U} + \frac{\partial S}{\partial U}\right)^n (U^{n+1} - U^n) = F^n + S^n, \tag{8.7}$$

which is known to have second order of convergence (quadratic convergence) under these conditions. Corresponding to the notation used in this thesis, Newton's scheme is also denoted by $CFL = \infty$, which is related to a time step of infinite length in (8.6).

It is a well known fact that the convergence of Newton's method depends on a sufficiently accurate initial state. The Taylor linearization is only valid in a vicinity of the current solution U^n. This is the reason, why a suitable initial guess is essential. Note that this requirement is not a stability restriction and the time step can actually be arbitrarily large. The only condition is that the solution does not change rapidly at each time level. To satisfy this constraint and make the linearization valid, either small time steps or a suitable initial guess are crucial. After the algorithm starts converging one can increase the CFL number to an arbitrarily large or infinite value. Since the computation of the original Jacobian matrix is a complicated task and may worsen the matrix properties, a low-order approximation of the Jacobian will serve as a preconditioner. Its computation is described in section 8.4. To indicate this fact, the scheme will be called Newton-like. The approximation does not provide second order convergence, but yields a robust and parameter-free scheme. There is no significant loss of efficiency since second order convergence is related to differentiability of the residual, which is not available for the flux function of Roe and the limiter function applied in this study.

8.3 Iterative Solution of Stationary Equations

The spatial discretization of the stationary Euler or two-fluid equations results in a nonlinear algebraic system of equations

$$R(U) = 0, \tag{8.8}$$

due to the nonlinearities induced by the physical nature of the equations as well as the spatial discretization. Therefore, a solution of large nonlinear systems is required. It is common practice to solve nonlinear equations by fixed point iterations of the form

$$U^{n+1} = U^n + [A^n]^{-1} R^n, \qquad n = 0, 1, 2, \dots \tag{8.9}$$

where A is suitably defined preconditioner and R is the nonlinear residual of the steady-state problem

$$R(U) = F(U) + S(U). \tag{8.10}$$

It results from the spatial discretization of the flux vector $F(U)$ and the source terms $S(U)$ ($S(U) = 0$ in the case of the single-phase Euler equations). The superscript n denotes either the index of the nonlinear iteration or, if a (pseudo-) time stepping method is applied, the time level. Feasible choices for the preconditioner are

$$A^n = -\left(\frac{\partial R}{\partial U}\right)^n = -\left(\frac{\partial F}{\partial U} + \frac{\partial S}{\partial U}\right)^n \tag{8.11}$$

or

$$A^n = \left[\frac{M_L}{\Delta t} - \left(\frac{\partial R}{\partial U}\right)^n\right] = \left[\frac{M_L}{\Delta t} - \left(\frac{\partial F}{\partial U} + \frac{\partial S}{\partial U}\right)^n\right]. \tag{8.12}$$

This results in Newton's method or the semi-implicit time stepping scheme (8.6), which offers the potential of up to second order convergence in the Newton case. Note that other choices are possible and may be preferable since the Jacobian computation is far from being trivial or even impossible due to the lack of smoothness of the flux limiter function investigated in this thesis. A generalization of the operator L^* (3.75) as introduced for scalar equations may be useful as a linearization in steady-state computations if a low-order preconditioner inhibits convergence or introduces time step restrictions. In fact, Yee et al. [98] used this sort of preconditioning in their linearized non-conservative implicit (LNI) scheme for stationary computations. In this case, the linearized operator is an M-matrix and preserves the positivity of each iterate but only the fully converged solution is guaranteed to be conservative. The computation of the preconditioner is described in detail in section 8.4.

In a practical implementation the inversion of the preconditioner A is avoided by solving the linear system

$$A^n \Delta U^{n+1} = R^n \tag{8.13}$$

and updating the solution

$$U^{n+1} = U^n + \Delta U^{n+1}. \tag{8.14}$$

8.3.1 Underrelaxation

At an early stage of a simulation, where the linearization is rather inaccurate, it is preferable to update the solution by adding a rather small fraction of the computed increment instead of the whole. Let the linear system (8.13) represent (8.6) or (8.7) at the time level n. Its solution is given by ΔU^{n+1}, which updates the solution of the discretized governing equations

$$U^{n+1} = U^n + \Delta U^{n+1}. \tag{8.15}$$

In an explicit underrelaxation procedure this equation is replaced by

$$U^{n+1} = U^n + \pi \Delta U^{n+1}, \tag{8.16}$$

where a parameter $0 < \pi < 1$ controls the solution increment [69, 19].

Underrelaxation may also be applied in a more implicit way. In this approach, the diagonal entries of the matrix are modified by addition of a positive number $\delta_i > 0$ [69, 19]

$$A^n_{ii} = A^n_{ii} + \delta_i. \tag{8.17}$$

This improves the matrix properties by enhancing diagonal dominance. Large diagonal entries inhibit a rapid mass exchange between adjacent nodes. This is closely related to the explicit way of underrelaxation.

Obviously, the semi-implicit scheme (8.6) can be interpreted as an implicit underrelaxation of the Newton (or Newton-like) scheme with

$$\delta_i = \frac{M_{Li}}{\Delta t} \quad \text{and} \quad A^n = \left[\frac{M_L}{\Delta t} - \left(\frac{\partial F}{\partial U} + \frac{\partial S}{\partial U} \right)^n \right]. \tag{8.18}$$

Hence, a small time step or CFL number corresponds to large underrelaxation and vice versa. In implicit codes the implicit underrelaxation technique is advantageous since explicit underrelaxation does not improve the matrix properties. This is sometimes important to achieve convergence of linear solvers or prevent divergence of the nonlinear iteration at an early stage of a simulation. Therefore, implicit underrelaxation in terms of local (pseudo-) time stepping is applied to compute an initial guess and to evaluate the performance of the developed methods at large CFL numbers.

8.3.2 The Linear Solver

The solution of the linear systems arising from the discretization of the governing equations of compressible flows is a challenging task since the Jacobian matrices are not diagonally dominant. This deteriorates the linear convergence rates and calls for powerful iterative or direct solvers. On the other hand, the involved matrices are sparse due to the small support of finite element basis functions, which makes the application of iterative solvers competitive. The numerical linear algebra in the field of sparse iterative solves has significantly advanced during the recent years. In particular, Krylov subspace techniques were found to be reliable and efficient enough for many practical purposes. They take advantage of a projection of large sparse systems into smaller subspaces [77]. The most famous algorithm of this family is the generalized minimal residual method (GMRES). The GMRES algorithm combined with an incomplete lower upper factorization (ILU) preconditioner was quite successfully applied to compressible flows (see, e. g., [73, 62, 7]) and integrated into many software packages.

As a matter of fact, the performance of iterative solvers depends heavily on the availability of a suitable preconditioner. This study focuses primarily on the development of the nonlinear solver and details of the linear solver are beyond the scope. Therefore, either a direct LU decomposition by UMFPACK or the BiCGSTAB algorithm with ILU preconditioning are applied as black box solvers. A detailed explanation of the BiCGSTAB solver and its application to compressible flows is reported in [71]. Note that by a clever renumbering of nodes, for example by the Cuthill-McKee algorithm [77], the band width of the sparse matrix can be reduced, which improves the performance of the ILU decomposition.

8.3.3 Computation of the Initial Guess

In steady-state computations the solution is usually initialized by the inlet boundary condition, which is extended into the whole domain. This is an efficient initialization technique since it operates automatically without knowledge of flow details. On the other hand it yields a very inaccurate estimate of the solution and consequently the linearization may fail. Therefore, as a kind of start-up phase, the semi-implicit scheme (8.6) with small CFL numbers can be employed and the CFL number may be increased step by step. This strategy combines implicit underrelaxation (compare section 8.3.1) and local time stepping, where the same CFL number is assigned to each node in terms of

$$\Delta t = \frac{CFL}{\lambda_{max}} h \qquad \Longrightarrow \qquad \delta_i = \frac{M_{Li}}{h} \frac{\lambda_{max}}{CFL}. \tag{8.19}$$

After the nonlinear residual falls below some error bound, one may set the CFL number to infinity and switch to the Newton-like scheme (8.7).

A second option, which is employed for all computations investigated in this thesis, is to prescribe the initial value in terms of the stationary low-order solution. This solution can be computed in a few iteration steps, possibly starting with low CFL numbers, and exhibits the main flow features. It is usually sufficient to satisfy the requirements of the Taylor linearization. On very fine meshes or extremely difficult problems, one may additionally start with a moderate CFL number, which is increased during the simulation. Underrelaxation plays an important role in the early stage of the computation of the low-order solution to the two-fluid equations. It may also be applied in the first steps of the main iteration phase on very fine meshes or for difficult problems. Last but not least, a coarse grid solution may serve as an initial guess for the simulation on finer grids. This can be implemented within the framework of nested iteration [56].

8.4 The Approximate Jacobian

The solution of the nonlinear algebraic system resulting from schemes (8.6) or (8.7) involves the computation of the Jacobian of the nonlinear numerical fluxes $F(U)$ and the source terms $S(U)$ with respect to the conservative variables U. Both functions lack differentiability, which prevents Newton's scheme from achieving second order convergence. There are some approaches to compute an approximation of the numerical flux Jacobian reported in the literature. The easiest option to compute such an approximation is finite differencing of first or second order [62, 73]

$$\frac{\partial R}{\partial U} \approx \frac{R(U+\varepsilon) - R(U)}{|\varepsilon|} \qquad \text{or} \qquad \frac{\partial R}{\partial U} \approx \frac{R(U+\varepsilon) - R(U-\varepsilon)}{2|\varepsilon|}. \tag{8.20}$$

The approximation can be further improved by extrapolation [60]. Finite differencing is a convenient way to differentiate the fluxes and source terms and can also be applied directly to the nonlinear residual. On the other hand, the convergence behavior depends heavily on the choice of the vector ε. There is an optimal value of ε, which differs for each problem and needs tuning to be determined by trial and error. This rules out the use of finite differencing in a general purpose code, which should be applicable more or less as a black box tool to a wide range of real life problems.

A more appropriate approach is to replace the non-differentiable functions of maximum, minimum and the absolute value by smooth approximations [59]. Note that approximations should be used for the Jacobian computation, while the residual should be computed by the original functions for accuracy reasons. This leads to another promising way to compute an approximation of the Jacobian matrix. If the non-differentiable functions in the residual are replaced by smooth approximations, the analytic derivative can be determined. This way of computing the Jacobian is very accurate and does not require the specification of additional parameters, except for the definition of the smooth approximations. The convergence behavior will be stable with respect to perturbations of these parameters. On the other hand, the analytical determination of the Jacobian of Roe's flux formula is subject to a rather complicated algebra and programming. It can be found in [4].

Another approach to Jacobian approximation, which has been applied quite successfully, is automatic differentiation. It is available as commercial software and it is able to compute the derivative of arithmetic operations using the chain rule. However, it relies on a differentiable residual function. Therefore, the non-differentiable maximum and minimum as well as the absolute value have to be replaced by smooth approximations.

The use of a limiter increases the stencil of the scheme and results in a considerable amount of additional fill-in compared to the Galerkin flux Jacobian. Since the limiter function is evaluated at the upwind node of an edge, the correction factors, may depend on the direct neighbors of the considered node as well as their neighbors, which are responsible for the additional matrix entries [60]. Furthermore, the computation of approximate Jacobians of the correction factors results in a deterioration of matrix properties and decreased robustness. In particular for the discretization of the two-fluid model, robustness is more important than efficiency up to a certain level.

8.4.1 Edge-Based Approximate Interior Flux Jacobian

In this thesis a different approach compared to the ones reported above is used to approximate the Jacobian. The original Jacobian in (8.7) and (8.6) is replaced by an edge-based approximation of the low-order Jacobian, where the correction factors are neglected. The approximate Jacobian (or preconditioner) constructed in this way is free of additional problem-dependent parameters and enjoys several advantages in comparison with the previously discussed approximations. No additional fill-in is created since the low-order fluxes only depend on the direct neighbors of a current node. For scheme (8.6) the resultant matrix is of M-matrix type, which is related to positivity preservation of each iterate subject to a suitable time step (at least on the characteristic level). This is an important fact since it rules out unphysical effects, which may cause divergence (e. g. zero density). The low-order approximation therefore provides increased robustness and is subject to lower memory requirements. Moreover, its computation is much less expensive than that of the approximation of the full Jacobian.

The kind of Jacobian approximation proposed here, can be determined analytically by a derivation of the low-order fluxes under certain simplifying assumptions. The sum of fluxes to a node i, which is related to the i-th row of the residual vector, is given by

$$F_i^{low} = \sum_j \mathbf{c}_{ji} \cdot \mathbf{F}_j - \underbrace{\int_{\partial\Omega} \varphi_i \mathbf{n} \cdot \mathbf{F}_h \, ds}_{=B_i} + \sum_{j \neq i} D_{ij}(U_j - U_i). \tag{8.21}$$

On the right hand side the first two terms represent the Galerkin discretization (7.2) corresponding to (5.6) or (4.16). For the Euler equations or the gaseous phase of the two-fluid model D_{ij} are the diffusion blocks defined by equation (5.31). In the case of the particulate phase D_{ij} can be written as

$$D_{ij} = diag\{d_{ij}, d_{ij}, d_{ij}, d_{ij}\}, \tag{8.22}$$

where d_{ij} are given by (4.21). For the sake of simplicity one assumes that the diffusion blocks are constant with respect to the conservative variables. The first term of equation (8.21) involves the Galerkin coefficients $\mathbf{c}_{ji}$ defined by (3.38). They are independent of the conservative variables. Under the above simplification, the derivatives of the low-order nodal flux F_i^{low} with respect to the conservative variables at nodes i and j are given by

$$\frac{\partial F_i^{low}}{\partial U_j} = \mathbf{c}_{ji} \cdot \mathbf{A}_j - \frac{\partial B_i}{\partial U_j} + D_{ij}, \qquad \frac{\partial F_i^{low}}{\partial U_i} = \mathbf{c}_{ii} \cdot \mathbf{A}_i - \frac{\partial B_i}{\partial U_i} - D_{ij} \tag{8.23}$$

where $\mathbf{A}_j$ and $\mathbf{A}_i$ are the Jacobian tensors (1.15) or (4.7) evaluated at nodes j and i, respectively. The derivation of the boundary part B_i, which takes into account the boundary condition, is addressed in section (8.4.2). The approximate Jacobian proposed so far is exact with respect to the Galerkin discretization if the derivative of the boundary part can be determined exactly.

8.4.2 Approximate Boundary Flux Jacobian

This section concerns the derivation of an approximate Jacobian of the boundary integrals in the weak form of the Galerkin discretization (8.21). Discrete counterparts of $\frac{\partial B_i}{\partial U_i}$ and $\frac{\partial B_i}{\partial U_i}$ in (8.23) need to be specified. Note that the discretization of the Euler equations is included. The derivation is complicated since the boundary flux usually consists of the solution of the boundary Riemann problem (7.49) or (7.54). The boundary state may depend on both the prescribed boundary condition and information extracted from the interior of the domain. Recall that the boundary integral associated with node k is overwritten by the imposed boundary condition in terms of

$$\int_{\partial\Omega} \varphi_k \mathbf{n} \cdot \mathbf{F}_h \, ds \qquad \longrightarrow \qquad \int_{\partial\Omega} \varphi_k \mathbf{n} \cdot \tilde{\mathbf{F}}_h \, ds, \tag{8.24}$$

where $\tilde{\mathbf{F}}_h$ contains the information due to the boundary condition. It is necessary to derive the Jacobian of $\mathbf{n} \cdot \tilde{\mathbf{F}}_h$ since it is the only function in the boundary integral depending on the conservative variables.

It is worth mentioning that the final Jacobian of the boundary term depends on the boundary flux Jacobian as well as the quadrature rule, see (7.9) and (7.10). Only the former topic is dealt with in the present thesis. It is convenient to assemble the Jacobian in a loop over edges. In the case of the Euler equations, the derivation of the Jacobian can be significantly simplified by multiple applications of the chain rule. We refer to Appendix A for a detailed description of the procedure that was used to determine the boundary flux Jacobian.

8.4.3 Approximate Source Term Jacobian

Last but not least, the source term Jacobian must be defined. Like the fluxes the source terms of the two-fluid model lack differentiability. A suitable approximation of the corresponding Jacobian is presented in Appendix B. It is employed as a preconditioner in stationary computations.

8.4.4 Flux Jacobian Structure and Assembly

The approximate flux Jacobians of the Euler equations as well as the equations of the particulate phase can be assembled in a loop over edges in terms of edge-wise contributions. Due to the small support of the finite element basis functions, the approximate flux Jacobians are very sparse and exhibit a 4 × 4-block structure:

$$DF = \begin{bmatrix} \boxed{DF_{11}} & & \boxed{DF_{1i}} & & & \boxed{DF_{1n}} \\ & \boxed{DF_{22}} & & & \boxed{DF_{2k}} & \\ \boxed{DF_{i1}} & & \ddots & & & \\ & & & \boxed{DF_{jj}} & & \boxed{DF_{jn}} \\ & \boxed{DF_{k2}} & & & \ddots & \\ \boxed{DF_{n1}} & & & \boxed{DF_{nj}} & & \boxed{DF_{nn}} \end{bmatrix}, \qquad n = NVT. \tag{8.25}$$

In (8.25) the blocks are defined by equation (8.23)

$$\boxed{DF_{ij}} = \frac{\partial F_i^{low}}{\partial U_j}. \tag{8.26}$$

In a practical implementation the flux Jacobian DF can be assembled as follows:

1. Initialize $DF = 0$.
2. In a loop over edges $I = \{ij\}, i \neq j$:
   - Add the off-diagonal blocks of the interior part of the discrete transport operator:
     $$DF_{ij} = DF_{ij} + \mathbf{c}_{ji} \cdot \mathbf{A}_j \qquad DF_{ji} = DF_{ji} + \mathbf{c}_{ij} \cdot \mathbf{A}_i.$$
   - Insert the stabilization:
     $$\begin{aligned} DF_{ii} &= DF_{ii} - D_{ij} & DF_{ij} &= DF_{ij} + D_{ij} \\ DF_{ji} &= DF_{ji} + D_{ij} & DF_{jj} &= DF_{jj} - D_{ij} \end{aligned}$$
3. In a loop over nodes assemble the diagonal blocks:
   $$DF_{ii} = DF_{ii} + \mathbf{c}_{ii} \cdot \mathbf{A}_i.$$
4. In a loop over all boundary edges insert the boundary flux Jacobians $DB_{kl} = \frac{\partial B_k}{\partial U_l}$:
   $$\begin{aligned} DF_{ii} &= DF_{ii} - DB_{ii} & DF_{ij} &= DF_{ij} - DB_{ij} \\ DF_{ji} &= DF_{ji} - DB_{ji} & DF_{jj} &= DF_{jj} - DB_{jj} \end{aligned}$$

8.4.5 The Two-Fluid Model Approximate Jacobian

The two-fluid model (2.68) consists of eight equations, where the first four govern the gas phase and the last four equations are associated with the particulate phase. The fluxes in conservative variables are independent of each other, while the source terms depend on all conservative variables of the gas as well as the particulate phase. The flux Jacobian of the i-th row of the flux vector with respect to the j-th state is therefore an 8 × 8-block-diagonal matrix of the form

$$DF_{ij} = \begin{bmatrix} DF_{g_{ij}} & 0 \\ 0 & DF_{p_{ij}} \end{bmatrix}. \tag{8.27}$$

In the equation above $DF_{g_{ij}}$ approximates the flux Jacobian of the gas phase corresponding to (8.23) for the effective density. Moreover, $DF_{p_{ij}}$ refers to the flux Jacobian of the particulate phase. The interfacial coupling is due to the source terms. Therefore, the source term Jacobian does not exhibit the block-diagonal structure and introduces the interfacial coupling into the matrix. Let the discretized form of the source term contain the nodal values

$$S_{p_i} = M_{Li} \begin{bmatrix} 0 \\ \mathbf{F}_{Di} \\ \mathbf{v}_{p_i} \cdot \mathbf{F}_{Di} + Q_{Ti} \end{bmatrix}, \qquad S_{g_i} = -S_{p_i}. \tag{8.28}$$

The source term Jacobian of the i-th row with respect to the i-th state admits the following representation (compare (2.68)):

$$DS_{ii} = \begin{bmatrix} \frac{\partial S_{g_i}}{\partial U_{g_i}} & \frac{\partial S_{g_i}}{\partial U_{p_i}} \\ \frac{\partial S_{p_i}}{\partial U_{g_i}} & \frac{\partial S_{p_i}}{\partial U_{p_i}} \end{bmatrix}. \tag{8.29}$$

Due to mass lumping, the derivatives of the i-th row of the source term vector with respect to the state j are zero unless $i = j$. Hence, the source term Jacobians of each node are 8 × 8-blocks, which are located on the diagonal of the two-fluid Jacobian.

For the sake of simplicity and to avoid confusion with the notation, the Jacobian as well as the state vector may be reordered. Let U_g denote the vector of nodal values of the gas phase, while its analog of the particulate phase is represented by U_p. Both vectors can be fitted together and form the reordered vector of nodal values of the two-fluid model

$$U = \begin{bmatrix} U_g \\ U_p \end{bmatrix}. \tag{8.30}$$

The source term vector as well as the flux vector are reordered in the same way

$$F = \begin{bmatrix} F_g \\ F_p \end{bmatrix}, \qquad S = \begin{bmatrix} S_g \\ S_p \end{bmatrix}, \qquad S_g = -S_p. \tag{8.31}$$

In terms of the latter definitions, one may rewrite the discretized two-fluid model by

$$\partial_t U = F + S. \tag{8.32}$$

Consequently, the flux Jacobian exhibits the following structure:

$$DF = \begin{bmatrix} DF_g & 0 \\ 0 & DF_p \end{bmatrix}. \tag{8.33}$$

Both DF_g and DF_p are $4\,NVT \times 4\,NVT$-blocks of the form (8.25). The source term Jacobian transforms to

$$DS = \begin{bmatrix} \frac{\partial S_g}{\partial U_g} & \frac{\partial S_g}{\partial U_p} \\ \frac{\partial S_p}{\partial U_g} & \frac{\partial S_p}{\partial U_p} \end{bmatrix}. \tag{8.34}$$

The off-diagonal blocks are obviously nonzero due to the interfacial coupling. On the other hand, the i-th row of the source term vector depends only on the conservative variables corresponding to the state i. Therefore, the $4\,NVT \times 4\,NVT$-blocks of the source term Jacobian (8.34) are block-diagonal matrices consisting of 4 × 4-blocks. They are related to the source term derivatives of each node

$$\frac{\partial S_k}{\partial U_l} = M_L\, diag\{\frac{\partial S_{k1}}{\partial U_{l1}}, \dots, \frac{\partial S_{kNVT}}{\partial U_{lNVT}}\}, \tag{8.35}$$

where k, l denote either index g or p.
In summary, the Jacobian of the two-fluid model reads

$$\frac{\partial F}{\partial U} + \frac{\partial S}{\partial U} = DF + DS. \tag{8.36}$$

It is constituted by the flux Jacobians and the source term Jacobians of both phases and has nonzero off-diagonal blocks due to the interfacial source term coupling. Therefore, the equations of both phases must be solved simultaneously. This requires the solution of a linear system of size $8\,NVT \times 8\,NVT$. To decompose it into two systems of size $4\,NVT \times 4\,NVT$ one may delete the off-diagonal blocks of the source term Jacobian, which contain just a few nonzero entries. On the one hand, this reduces the memory requirements and the computational costs significantly. On the other hand, the resulting preconditioner is a bad approximation of the original Jacobian and introduces a restrictive step size constraint. This behavior corresponds to the physical properties of the equations. The source terms are usually large and dominating in comparison with the fluxes and the deletion of off-diagonal matrix blocks introduces some explicitness.

Finally, the Jacobian of the two-fluid model consists of the flux Jacobians of both phases and the source term Jacobian. Consequently, the Jacobian matrix is given by equation (8.36).

Part III

Numerical Results

9 Euler Equations

The last part of the thesis is concerned with numerical studies of the presented methods. The accuracy is examined by grid convergence studies for several test cases and the numerical performance at large and infinite CFL numbers is analyzed.

In this chapter, the compressible Euler equations are investigated for some well known test cases, as well as much more difficult computations. The numerical study illustrates the convergence of the proposed Newton-like method at practically arbitrary Mach numbers and in domains with curved boundaries. Numerical evidence of unconditional stability is presented for both the high-resolution and the low-order scheme. The numerical results computed by the low-order scheme show that the weak imposition and implicit treatment of boundary conditions bring about a marked improvement of the nonlinear convergence rates. A similar behavior can be observed for the high-resolution scheme. The results indicate that the nonlinear convergence rates improve with increasing CFL number and stagnate for large CFL numbers above $CFL = 100$. Since this is not the case in the low-order computations, the phenomenon is caused by the limiter and the performance of the boundary conditions should be judged for the low-order computations. Nevertheless, convergence is always reached without freezing the correction factors, although limiters of the employed type are often assumed to inhibit convergence to steady-state. The fact that the presented scheme converges, despite oscillatory correction factors, may also be associated with improved boundary conditions.

Moreover, the grid convergence of both the low-order and the high-resolution scheme based on algebraic flux correction are examined for several test cases. Remarkably, it is shown by numerical results that the proposed scheme does not exhibit the carbuncle phenomenon, which is known to be a weakness of approximate Riemann solvers. The goal is to show the performance of the methods on unstructured as well as structured meshes. Linear or bilinear elements are employed and the involved grids consist of either triangles or quadrilaterals.

For the purpose of accuracy assessment we measure the relative L_2-error. The analytical solution to the test cases to be presented in this thesis is typically unknown. Consequently, the error analysis is based on a sufficiently accurate numerical reference solution. For the results below, the solutions computed on the finest grid serve as the reference solutions. This enables us to determine the empirical order of convergence [89]. Let the numerical solutions of either the low-order or the high-resolution scheme, associated with the mesh sizes h and $\frac{h}{2}$, be given by U_h respectively $U_{\frac{h}{2}}$. Assume that the error satisfies approximately

$$\|U_h - U_{ref}\|_2 \approx Ch^p, \qquad \|U_{\frac{h}{2}} - U_{ref}\|_2 \approx C\left(\frac{h}{2}\right)^p, \tag{9.1}$$

where U_{ref} is the reference solution and p indicates the order of convergence. Under these conditions the order of approximation can be determined by

$$p \approx \frac{\log\left(\frac{\|U_h - U_{ref}\|_2}{\|U_{\frac{h}{2}} - U_{ref}\|_2}\right)}{\log(2)}. \tag{9.2}$$

In what follows the relative L_2-error is calculated by

$$E_2 = \frac{\|U_h - U_{ref}\|_2}{\|U_{ref}\|_2}. \tag{9.3}$$

Let us at first focus our attention on the GAMM channel and NACA airfoil benchmarks, which are well documented in the literature and usually employed for the validation of compressible flow solvers. The above problems incorporate the most important features of compressible flow, like shocks and reflections of waves as well as subsonic and wall boundaries at various Mach numbers.

9.1 10% GAMM Channel

In this section the well known GAMM channel is investigated. Transonic flow at free stream Mach number $M_\infty = 0.67$ impinges on a 10% circular bump in a channel. The flow is reflected on the bump, accelerated, and a shock arises on the bump. In this test, the Mach number varies between approximately 0.22 and 1.41. Subsonic, transonic, and supersonic regions as well as the sonic point at Mach number $M = 1$ occur. Characteristic stiffness associated with the sonic point, the curved boundary of the bump, and the relatively wide range of Mach numbers make this test case a challenging task for the numerical scheme. Additionally, the subsonic nature of the inlet and outlet requires a suitable boundary treatment to achieve convergence to steady-state.

The unstructured triangular coarse mesh is illustrated in figure 9.1 (b). To derive higher mesh levels, the grid is globally refined by subdividing each edge into two equal parts. Mesh data for the different refinement levels is listed in table 9.1, where NVT denotes the number of vertices and NEL refers to the number of elements involved. A numerical solution on mesh level six computed by the high-resolution scheme is presented in figure 9.1 (a), where the Mach number is illustrated. This serves as the reference solution for the grid convergence analysis reported below. It can be observed that the solution computed by the high-resolution scheme is well resolved and free of unphysical effects like numerical oscillations. All simulations by the high-resolution scheme are initialized by the solutions of the low-order problem, which provide sufficiently accurate linearizations to employ infinite CFL numbers. The free stream values serve as

Level	NVT	NEL	E_2^{Low}	p^{Low}	E_2^{Lim}	p^{Lim}
1	176	292	$5.47 \cdot 10^{-2}$	0.59	$3.05 \cdot 10^{-2}$	0.56
2	643	1168	$3.64 \cdot 10^{-2}$	0.64	$2.07 \cdot 10^{-2}$	1.04
3	2453	4672	$2.34 \cdot 10^{-2}$	0.59	$1.01 \cdot 10^{-2}$	0.99
4	9577	18688	$1.55 \cdot 10^{-2}$	0.61	$5.07 \cdot 10^{-3}$	1.45
5	37841	18688	$1.01 \cdot 10^{-2}$		$1.85 \cdot 10^{-3}$	
6	150433	299008				

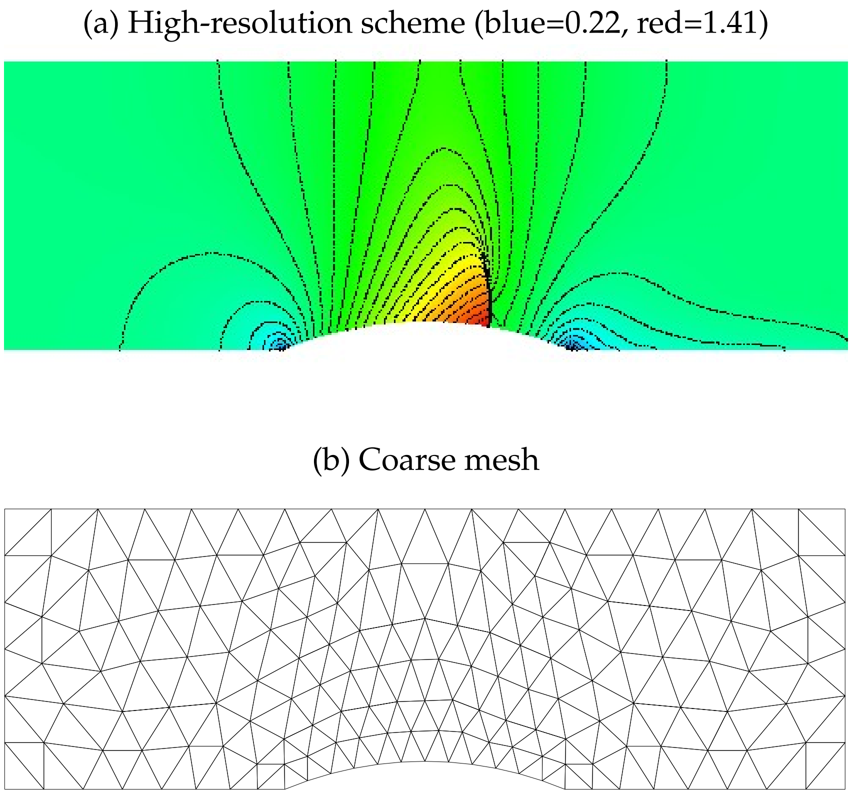
Table 9.1: GAMM channel: Mesh properties and error analysis

initial values for the low-order computations. During the low-order startup phase one employs a moderate CFL number of 100. After the relative residual falls to 10^{-2} the linearization is sufficiently accurate and the CFL number is increased to the desired level. Note that even $CFL = \infty$ is possible and divergence was never observed. In figure 9.2 nonlinear convergence histories for both the high-resolution scheme and the low-order scheme are depicted in logarithmic scale. The relative error is measured in the L_2-norm.

It follows from the numerical results that the nonlinear convergence rates of the semi-implicit pseudo time stepping scheme improve as the CFL number increases. In particular for the low-order scheme, an infinite CFL number results in the best convergence rates. This illustrates the performance of the proposed weak Neumann-type boundary conditions. After ten iterations the residual falls below 10^{-12}, while the solution approaches steady-state very slowly in the case of small CFL numbers. Small CFL numbers correspond to large underrelaxation, which is unnecessary for the linear low-order scheme. Moreover, an almost monotone decreasing error can be observed for sufficiently large CFL numbers.

A similar behavior of the high-resolution scheme is indicated by the convergence history, although more nonlinear iterations are required for convergence. When the CFL number is increased, the convergence rates improve until the threshold $CFL = 100$ is reached. At higher values of the CFL number, the convergence rates remain approximately the same. In contrast to [93] the rate of convergence does not deteriorate if the CFL number is too large. The high-resolution scheme is highly nonlinear and the correction factors lack differentiability. In fact, the correction factors oscillate. The lack of smothness of correction factors is a reason for the slower convergence as reported in [59]. Moreover, the employed preconditioner does not incorporate the correction factors. Convergence is nevertheless reached. Therefore, the performance of the developed boundary techniques can be better illustrated by the low-order computations. On the other hand, the additional computational effort is the price to pay for a superior solution. A preconditioner equipped with a differentiable limiter is likely to deliver better convergence rates [62, 59].

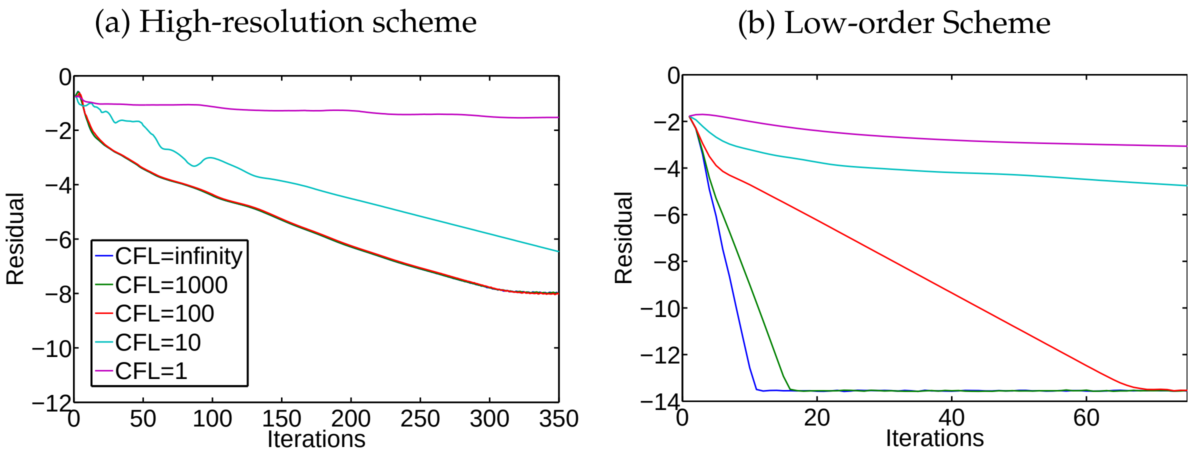
Last but not least let us focus our attention on the grid convergence analysis. Since, no analytical solution is known, a numerical counterpart is computed on mesh level six and serves as the reference solution. The relative L_2-error is shown in table 9.1. The empirical order of convergence can be determined, if one assumes that the reference solution is exact. For the low-order scheme the empirical order of convergence



(a) High-resolution scheme (blue=0.22, red=1.41)

(b) Coarse mesh

Figure 9.1: GAMM channel: Mach number (30 contours) and coarse grid



(a) High-resolution scheme

(b) Low-order Scheme

Figure 9.2: GAMM channel: Nonlinear convergence history in logarithmic scale for different CFL numbers on mesh level four

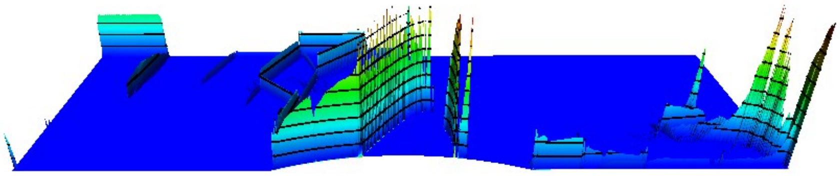


Figure 9.3: GAMM channel: Correction factors for negative fluxes in terms of $1 - R^-$ (blue= 0, red= 1)

is approximately 0.6. As expected, the high-resolution scheme converges faster. In this example, the order of approximation is about one. Note that the correction factors depend on the solution. Hence, the order of convergence of the high-resolution scheme is problem-dependent. For this particular problem a higher order of convergence is impossible due to the lack of smoothness. The discontinuous solution at the shock is approximated by the continuous numerical one. Therefore, the largest errors are to be expected at the shock. This is the main reason for an order of convergence much less than two as expected for a smooth solution.

At the same time the correction factors are not equal to one in smooth parts of the flow, which is a second reason for less than second order accuracy. This phenomenon is illustrated in figure 9.3, where the correction factors for negative antidiffusive fluxes R^- of the first characteristic field associated with the x-direction are presented. For visualization purpose $1 - R^-$ is plotted. Sokolichin compared several limiters for the linear 1D continuity equation discretized using finite volumes on a uniform mesh. He found out [89] that the order of convergence is less than or equal to one when the solution is discontinuous. In fact, in our results the numerical error of the reference solution may also play a role and the rate of convergence to the exact solution may be slightly worse. Nevertheless, grid convergence takes place, as shown by the results above.

9.2 NACA 0012 Airfoil

The second test case is the flow around an airfoil at free stream Mach number $M_\infty = 0.8$. The symmetric profile of the airfoil surface is given by the function [47]

$$f^{\pm}(x) = \pm 0.6 \left(0.2969\sqrt{x} - 0.126x - 0.3516x^2 - 0.1015x^4\right) \tag{9.4}$$

with $x \in [0, 1.00893]$. The computational domain consists of a circle with radius 10, which is centered at the tip of the airfoil and covered by an unstructured triangular grid. The coarse grid is illustrated in figure 9.4 (b). For the numerical solution an inclination angle of $\alpha = 1.25°$ is prescribed and a hierarchy of four meshes is investigated. The flow takes place at Mach numbers ranging between about $M = 0.02$ at the tip and $M = 1.36$ on the top of the airfoil. This indicates a characteristic stiffness related to the low Mach number regime as well as the sonic point. Moreover, the solution contains a shock and there are curved boundaries. The mesh data is summarized in table 9.2. Steady-state low-order solutions act as initial values for the high-order computations.

Level	NVT	NEL	E_2^{Low}	p^{Low}	E_2^{Lim}	p^{Lim}
1	2577	4963	$4.08 \cdot 10^{-2}$	0.51	$1.68 \cdot 10^{-3}$	1.02
2	10117	19852	$2.86 \cdot 10^{-2}$	0.46	$8.27 \cdot 10^{-4}$	1.02
3	40086	79408	$2.08 \cdot 10^{-2}$		$2.68 \cdot 10^{-4}$	
4	159580	317632				

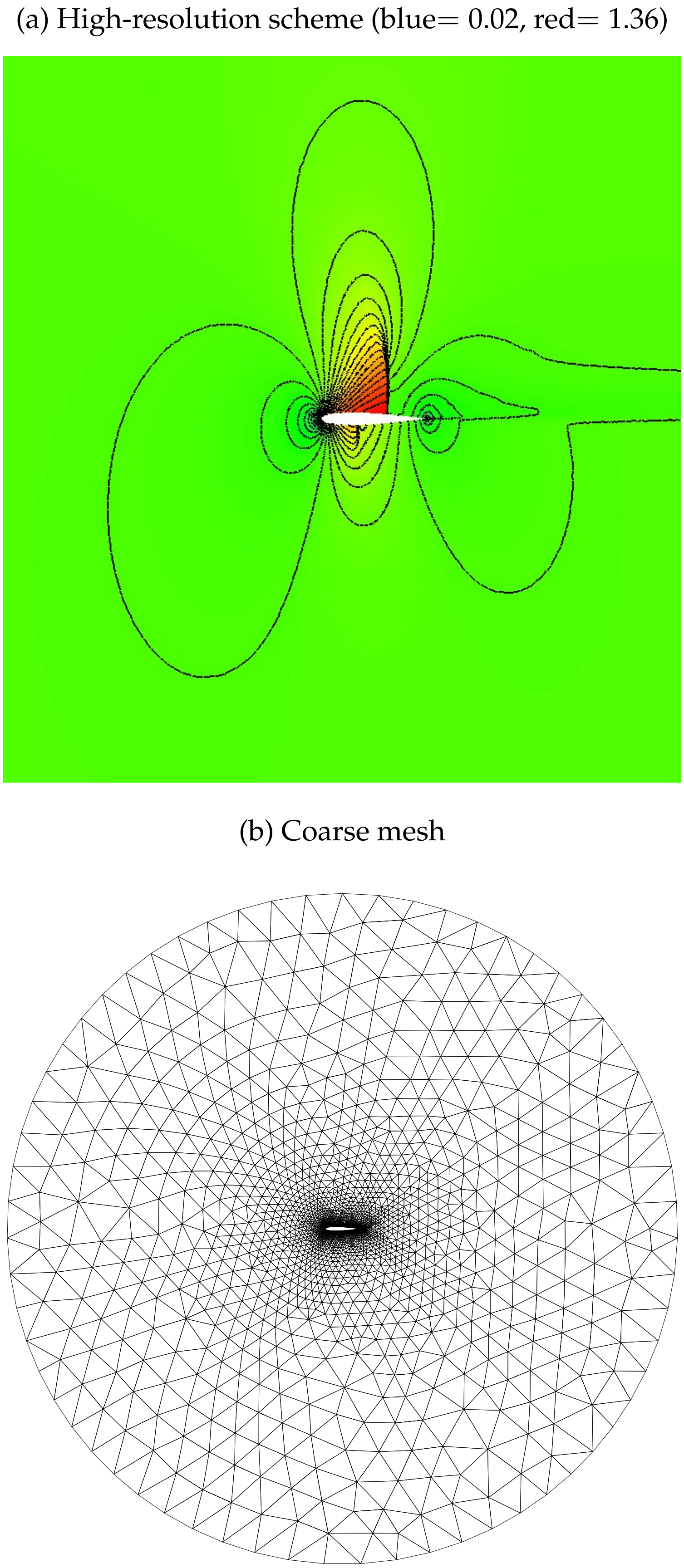
Table 9.2: NACA airfoil: Mesh properties and error analysis

We initialized the low-order solutions by the free-stream values and a few pre-iterations with $CFL = 10$ are performed before the simulation is switched to a higher CFL number. Divergence was never observed.

Similar to the GAMM channel test case we examine the grid convergence and the related empirical order of convergence based on a reference solution, which is computed on mesh level four. A zoom of the reference solution computed by the high-resolution scheme is depicted in figure 9.4 (a). One observes that the Mach number contours compare well with the results reported in [17, 47, 62, 29]. Table 9.2 lists the grid convergence study for the current test case. We obtain comparable convergence rates as for the GAMM channel test case. Convergence is of order 0.5 in the case of the low-order scheme. At the same time the high-resolution scheme exhibits faster convergence of order one.

The convergence behavior of the nonlinear iteration is also comparable to the GAMM channel test case (compare figure 9.5). The error of the low-order scheme falls below 10^{-15} in 20 iterations for an infinite CFL number and the rate of convergence deteriorates with decreasing CFL numbers. On the other hand, the relative error hardly decreases for small CFL numbers. The high-resolution scheme exhibits a qualitatively similar convergence history, although the number of iterations required for convergence is much larger due to the reasons pointed out above. After about 200 iterations the relative error falls below 10^{-8} and the solution can be considered to be stationary. There is an upper bound for the convergence rate. In other words, an increase of CFL numbers in the range above $CFL = 100$ yields only slightly better convergence rates. The convergence histories corresponding to $CFL = 100$, $CFL = 1000$, and $CFL = \infty$ are almost equal. In fact the relative error does not decrease significantly for $CFL = 1$ and $CFL = 10$. The error curves of both simulations are similar.

However, the very fast convergence of the low-order scheme gives evidence for the performance and robustness of the proposed boundary treatment for the semi-implicit FEM discretization. In spite of the characteristic stiffness associated with the low Mach numbers at the tip of the airfoil and the sonic point, the low-order residual decreases quickly. It is shown by the numerical results that a satisfactory error decay by several orders of magnitude can be achieved despite oscillating correction factors, provided sufficient care is taken in the implementation of the boundary conditions and of the implicit time stepping scheme. Moreover, it turns out that large CFL numbers are required for a satisfactory convergence to steady-state.



(a) High-resolution scheme (blue= 0.02, red= 1.36)

(b) Coarse mesh

Figure 9.4: NACA airfoil: Mach number zoom (40 contours) and coarse mesh

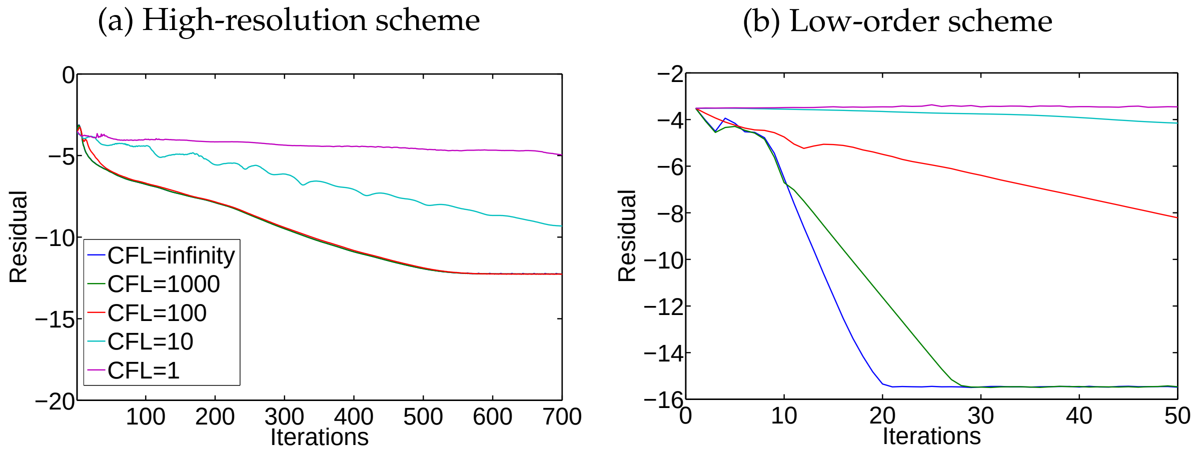


Figure 9.5: NACA airfoil: Nonlinear convergence history in logarithmic scale for different CFL numbers on mesh level two

	Upper wall						Upper Obstacle				
x_i	0.0	0.4	4.9	12.6	14.25	16.9	4.9	8.9	9.4	14.25	12.6
y_i	3.5	3.5	2.9	2.12	1.92	1.7	1.4	0.5	0.5	1.2	1.4

Table 9.3: Scramjet Geometry

9.3 Scramjet Inlet

In this section we study a flow that is supersonic almost everywhere and exhibits multiple shock reflections at the boundaries. The flow enters the domain at free stream Mach number $M_\infty = 3$ and impinges on two internal obstacles, where it is reflected in multiple directions. A small subsonic area arises at the throat, while the remaining part of the flow including the outlet is supersonic. The computational domain consists of three piecewise linear boundary components and it is symmetric with respect to the x-axis. The upper part of the geometry is spanned by the points depicted in table 9.3 [47]. An unstructured triangular coarse gird, illustrated in figure 9.6 (c), is utilized and refined up to three times for the numerical simulations. Its properties are listed in table 9.4. The wall boundary conditions require special treatment and the approximate/exact Riemann solver applied to the mirror condition fails to converge. This problem is closely related to the grid singularities at the obstacles and can be cured as suggested by Shapiro [84] who has developed a technique for handling grid singularities as interior nodes. In our case it suffices to treat just one edge at the tip of the

Level	NVT	NEL	E_2^{Low}	p^{Low}	E_2^{Lim}	p^{Lim}
1	2266	4132	$1.70 \cdot 10^{-1}$	0.45	$1.01 \cdot 10^{-1}$	0.74
2	8665	16528	$1.25 \cdot 10^{-1}$	0.44	$6.06 \cdot 10^{-2}$	1.11
3	33859	66112	$9.18 \cdot 10^{-2}$		$2.81 \cdot 10^{-2}$	
4	133831	264448				

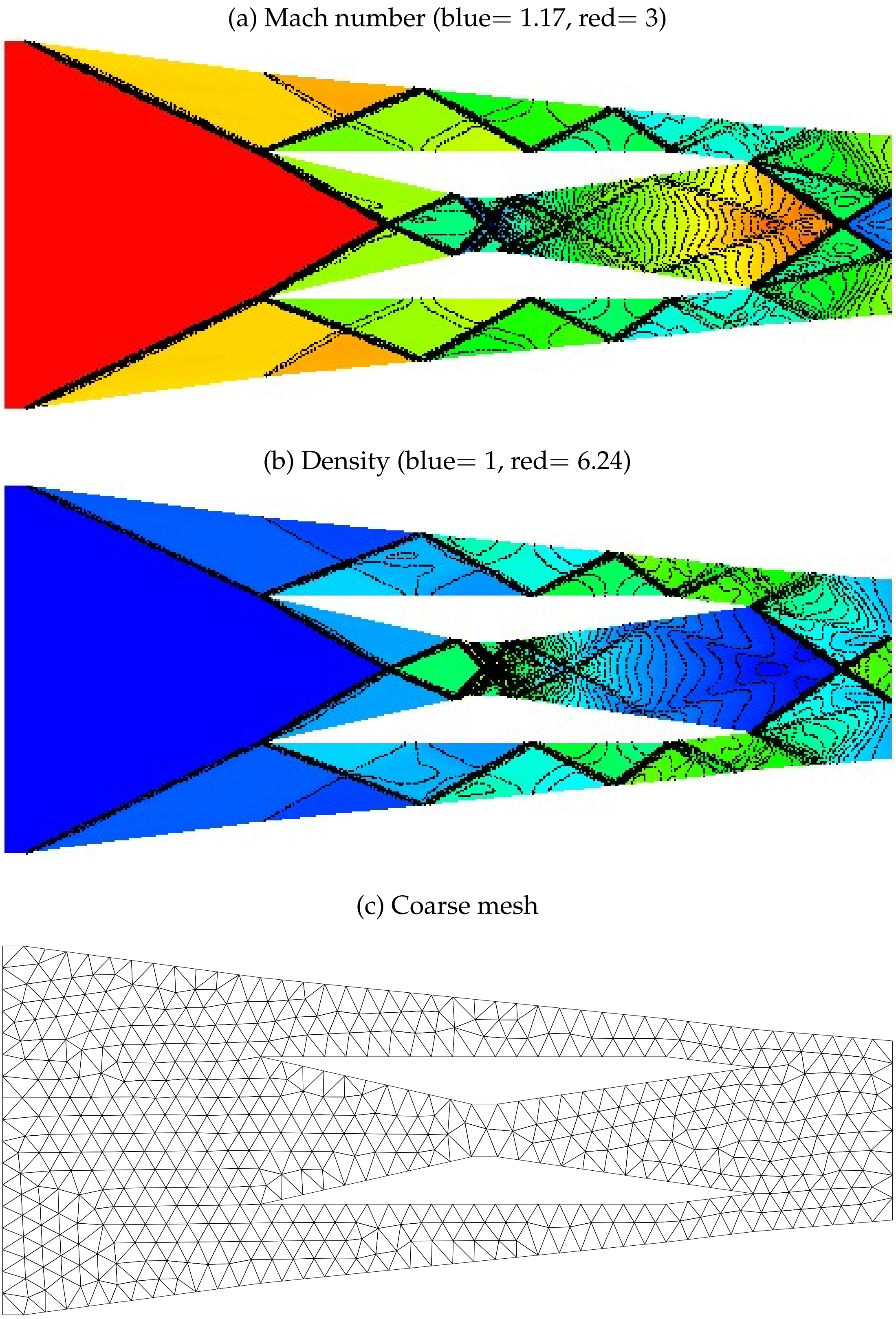
Table 9.4: Scramjet: Mesh properties and error analysis

obstacles as a supersonic outlet. Nevertheless, this artificial outlet may give rise to nonphysical solutions and restricts the generality of the algorithm. Hence, we propose a different treatment. The free-slip boundary conditions should be implemented in terms of the zero-flux condition (7.42), which does not require the solution of a Riemann problem and facilitates convergence, without artificial outlets.

We initialize the high-resolution scheme by the corresponding low-order solution for all CFL numbers. A few preiterations at lower CFL numbers are required for the startup phase of the low-order scheme. We choose $CFL = 10$ for the computations reported here. The reference solution, associated with mesh level four, is depicted in figures 9.6 (a) and (b). Strong as well as weak shocks are resolved rather well, although the small subsonic region at the throat is unresolved due to the globally refined grid. Mesh adaptivity should be employed to achieve a higher resolution of the weak shocks and of the subsonic region [60]. However, the contour lines compare well with the simulations of Möller [60] and Shapiro [84].

A grid convergence study is given in table 9.4. For the low-order scheme one observes an empirical rate of grid convergence of about 0.45, which is slightly less than the results for the NACA 0012 airfoil and the GAMM channel test case. The difference arises due to the multiple shock reflections in the current test case, while the solutions to the first two test problems are relatively smooth almost everywhere. Typically shocks degrade the rate of convergence since they necessitate a larger amount of numerical diffusion. For the high-resolution scheme the empirical rate of convergence varies between about 0.75 and 1.11. The reference solution also approximations shocks in a continuous way and contains a considerable amount of numerical diffusion. For that reason the rate of convergence may be slightly overestimated by the latter result.

Finally, the convergence history of the nonlinear iteration is examined for the scramjet inlet. Due to the multiple shocks, the correction factors vary over the whole domain and the induced nonlinearity is very strong. Hence one expects a very slow convergence to steady-state particularly of the high-resolution scheme. Nevertheless, convergence of the high-resolution scheme is reached in about 500 iterations since the relative error falls below 10^{-8} and the error of the low-order simulation approaches 10^{-14} in about 11 nonlinear iterations. The total number of low-order iterations is comparable to the previous test cases since the low-order scheme is linear and the preconditioner is close enough to the exact Jacobian. At the same time, the high-resolution scheme approaches steady-state slower compared to the NACA 0012 airfoil and GAMM channel simulations. The main reason is the stronger nonlinearity of the correction factors due to the multiple shock reflections. However, this test case indicates that the proposed treatment of boundary conditions is feasible also for highly non-smooth flows with shock reflections at the boundary. Moreover, a more physical treatment of mesh singularities is enabled by the weak form of the free-slip condition.



(a) Mach number (blue= 1.17, red= 3)

(b) Density (blue= 1, red= 6.24)

(c) Coarse mesh

Figure 9.6: Scramjet: Numerical solution (50 contours) and coarse mesh

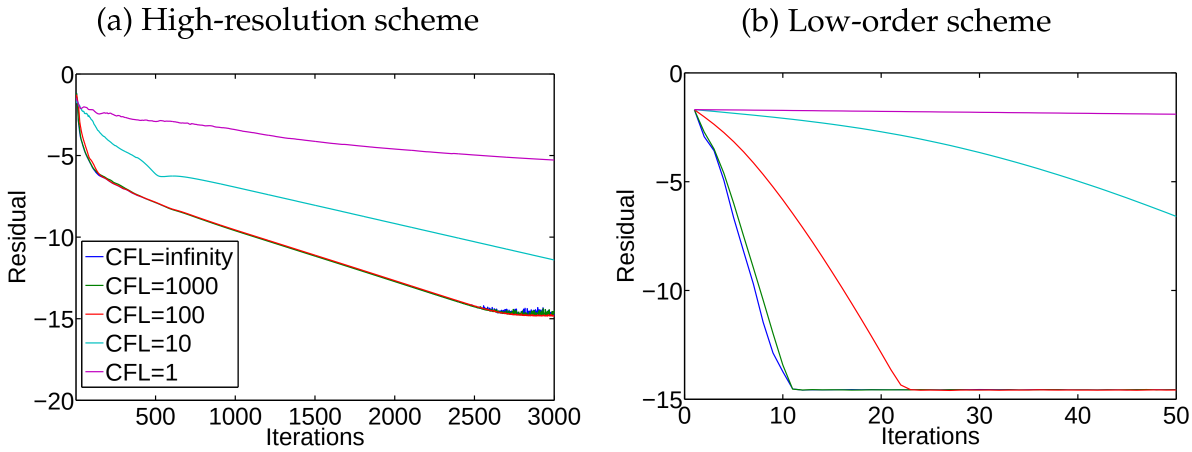


Figure 9.7: Scramjet: Nonlinear convergence history in logarithmic scale for different CFL numbers on mesh level three

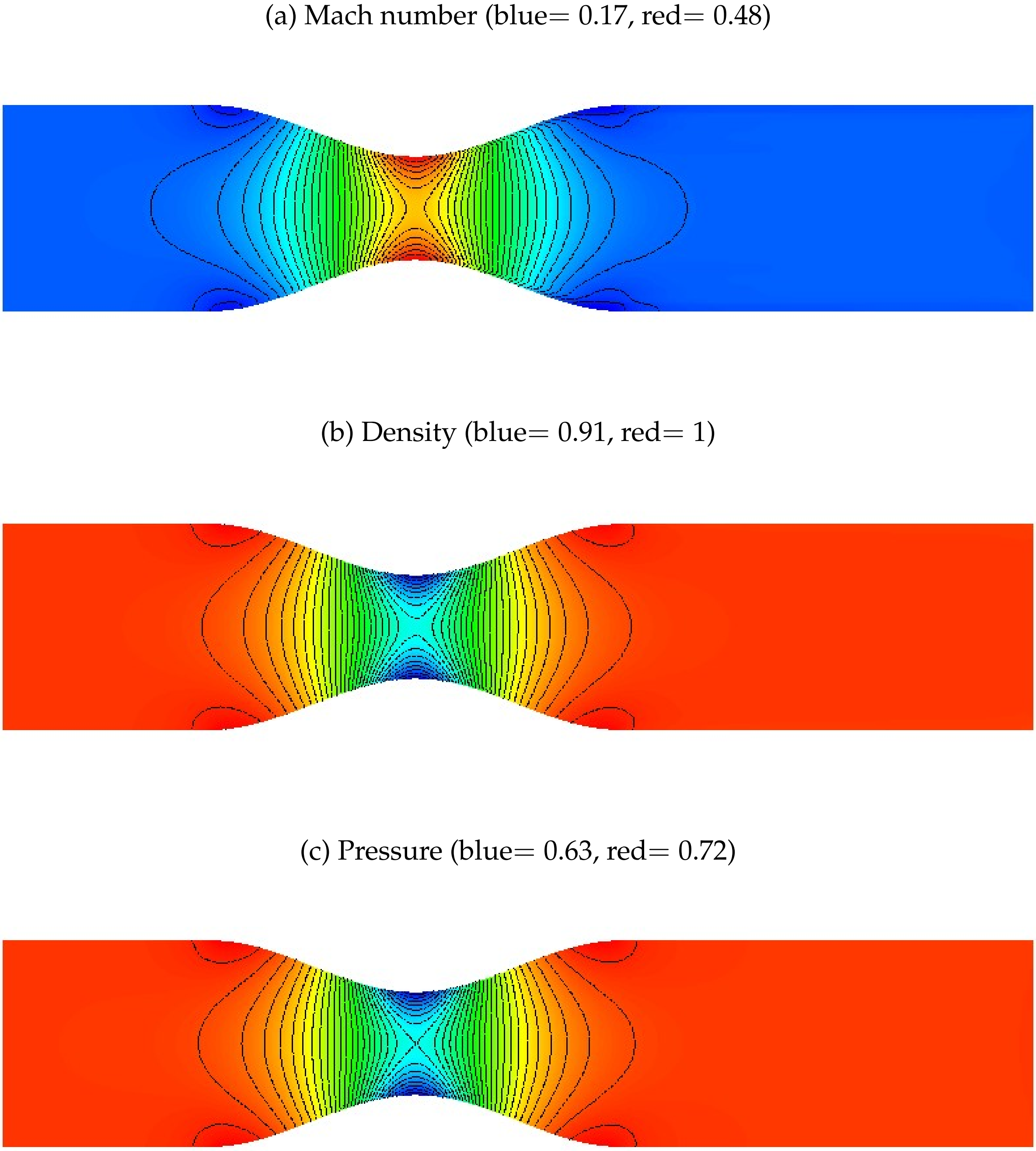
9.4 Converging-Diverging Nozzle Flow

Let us focus our attention on a converging-diverging nozzle. The upper and lower walls are defined by the function

$$g^{\pm}(x) = \begin{cases} \pm 1 & \text{if} \quad -2 \leq x \leq 0 \\ \pm\frac{\cos\left(\frac{\pi x}{2}\right)+3}{4} & \text{if} \quad 0 < x \leq 4 \\ \pm 1 & \text{if} \quad 4 < x \leq 8 \end{cases}, \tag{9.5}$$

which is adopted from [29]. Three different flow regimes are prescribed in terms of the inlet and outlet boundary conditions. The lower and upper boundaries are solid walls. First we consider a purely subsonic flow at free stream Mach number $M_\infty = 0.2$, which is prescribed as inlet and outlet boundary condition. The flow is accelerated up to Mach number $M = 0.48$ at the throat and decelerated in the diverging part of the nozzle by the outlet boundary condition. Due to the low Mach number, which varies between $M = 0.17$ and $M = 0.48$ the solution is smooth and the flow weakly compressible since the density variations are below 10%. Therefore, it is a challenging test case for the implicit flow solver since convergence problems of the nonlinear iteration are often associated with low Mach numbers, even though the solution is smooth. The properties of the employed structured quadrilateral mesh are given in table 9.5 and the numerical solution computed by the high-resolution scheme on mesh level seven is depicted in figure 9.8.

The convergence history reported in figure 9.9 reveals that the solution becomes stationary. The relative residual approaches nearly 10^{-10} for the high-resolution scheme and about 10^{-15} for the low-order scheme. The convergence history of the low-order scheme, illustrated in figure 9.9 in logarithmic scale, is qualitatively similar to the GAMM channel and NACA 0012 airfoil test cases. It turns out that the rate of convergence improves if the CFL number is increased. Moreover, the computations with small CFL numbers do not decrease the residual significantly for a considerable number of iterations. For an infinite CFL number, which corresponds to the Newton-like scheme, the



(a) Mach number (blue= 0.17, red= 0.48)

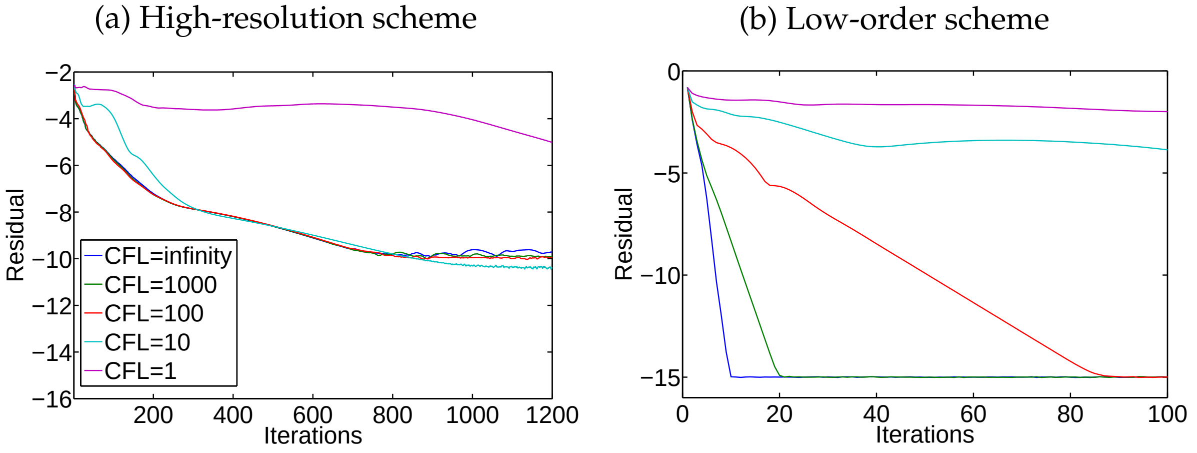
(b) Density (blue= 0.91, red= 1)

(c) Pressure (blue= 0.63, red= 0.72)

Figure 9.8: Subsonic nozzle flow (30 contours)

Level	NVT	NEL
1	33	20
2	105	80
3	369	320
4	1377	1280
5	5313	5120
6	20865	20480
7	82689	81920

Table 9.5: Nozzle: Mesh data


(a) High-resolution scheme

(b) Low-order scheme

Figure 9.9: Subsonic nozzle flow: Nonlinear convergence history for different CFL numbers on mesh level five in logarithmic scale

relative error falls below 10^{-15} after ten iterations. At the same time, 20 iterations are required to approach steady-state if $CFL = 1000$ is prescribed.

The high-resolution scheme exhibits a slightly different behavior. In the first 300 iterations the computations at $CFL \geq 100$ exhibit the best but almost invariable convergence rates. After the residual falls below 10^{-8} (350 iterations), that is the solution can be considered to be stationary, the error evolution is comparable to the simulation with $CFL = 10$. Hence, in this low Mach number test case also moderate CFL numbers yield a satisfactory convergence to steady-state. Note that in a practical application the convergence history is typically unknown. Therefore, the choice of large or even infinite CFL numbers is still preferable since it yields better results for most test cases and comparable results in this low Mach number computation.

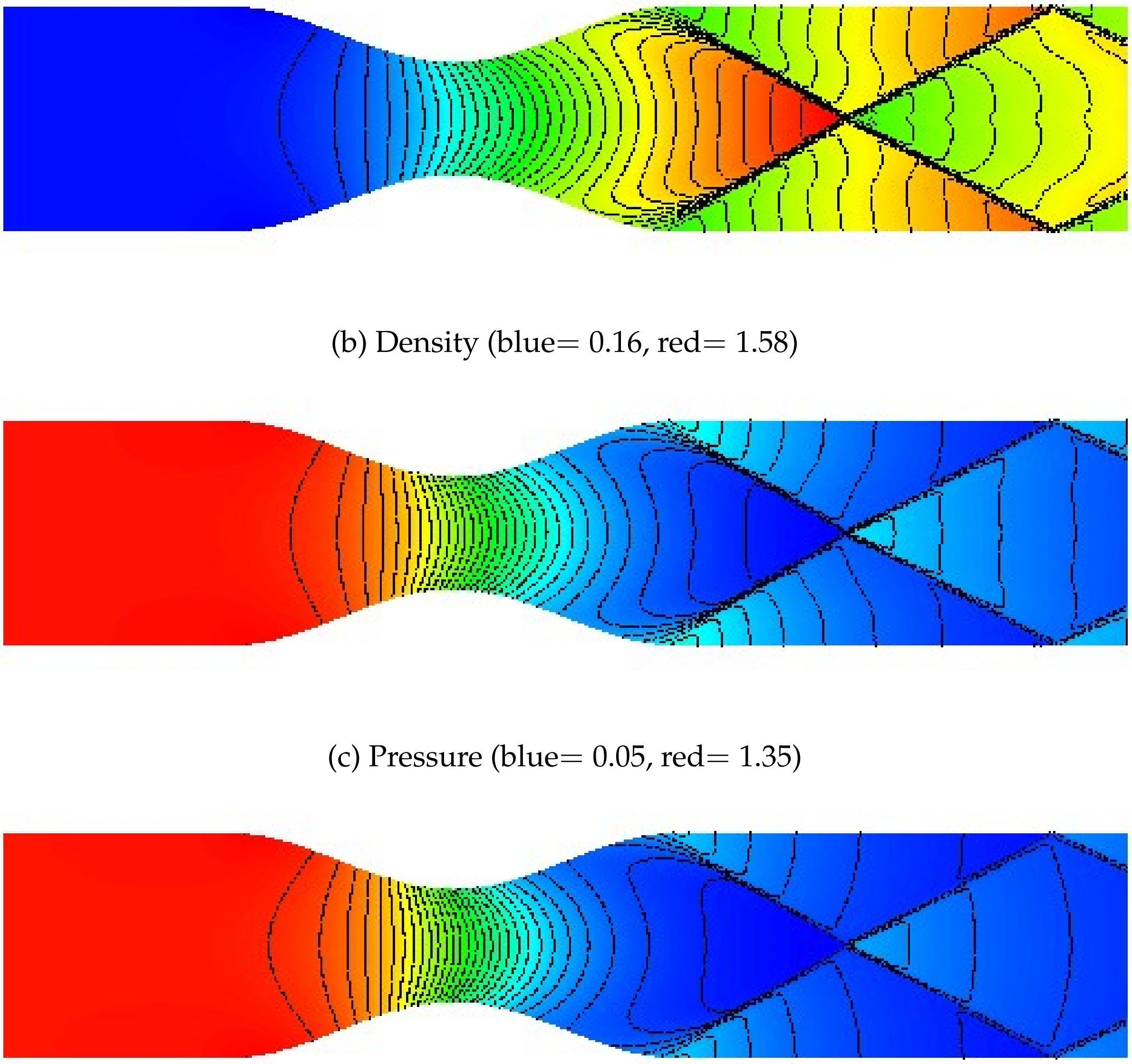
The second computation deals with a transonic inlet at free stream Mach number $M_\infty = 0.8$ and a supersonic outlet. The flow is accelerated in the diverging part of the nozzle up to Mach number $M = 2.8$ and shocks occur downstream the nozzle, which are reflected at the upper and lower walls. A numerical solution computed on mesh level seven can be found in figure 9.10. Results of both the first and the second computation compare well to the results of Hartmann [28].

Last but not least a subsonic/supersonic flow with a shock is investigated. At the inflow boundary subsonic flow at free stream Mach number $M_\infty = 0.3$ is prescribed. For the purpose of comparison we prescribe the same free stream conditions as Hartmann. That is, we impose a free stream pressure $P_\infty = 1$ instead of $P_\infty = \frac{1}{\gamma}$. At the outlet the free stream boundary conditions are replaced by a pressure outlet, which is presented in section 7.5.6. The prescribed pressure is $P_{out} = \frac{2}{3}$. This definition of boundary conditions yields subsonic flow of Mach number $M = 0.27$, which is accelerated in the diverging part of the nozzle up to Mach number $M = 2$. The flow suddenly decelerates at the end of the diverging part of the nozzle as it passes through a shock. The shock arises due to the pressure outlet boundary condition. Computational results obtained on mesh level seven are depicted in figure 9.11. They are almost indistinguishable from those obtained in [28]. A more detailed analysis of the boundary conditions concerning this test case can be found in section 9.6.

9.5 Circular Cylinder

The last test case for the Euler equations consists of a circular half cylinder, which is placed in a gas flow of free stream Mach number $M_\infty = 10$. This problem turns out to be a very challenging task for the nonlinear solver. The hypersonic flow, impinging on the surface of the cylinder, creates a bow shock. In front of the cylinder the Mach number is about zero. Hence, the flow regime switches from hypersonic to the low Mach number (stagnation) regime, which is responsible for nearly infinite stiffness. It makes the stationary problem very challenging and requires a powerful solver equipped with strong underrelaxation to prevent divergence of the nonlinear iteration.

Many approximate Riemann solvers, in particular the Roe solver, are known to exhibit

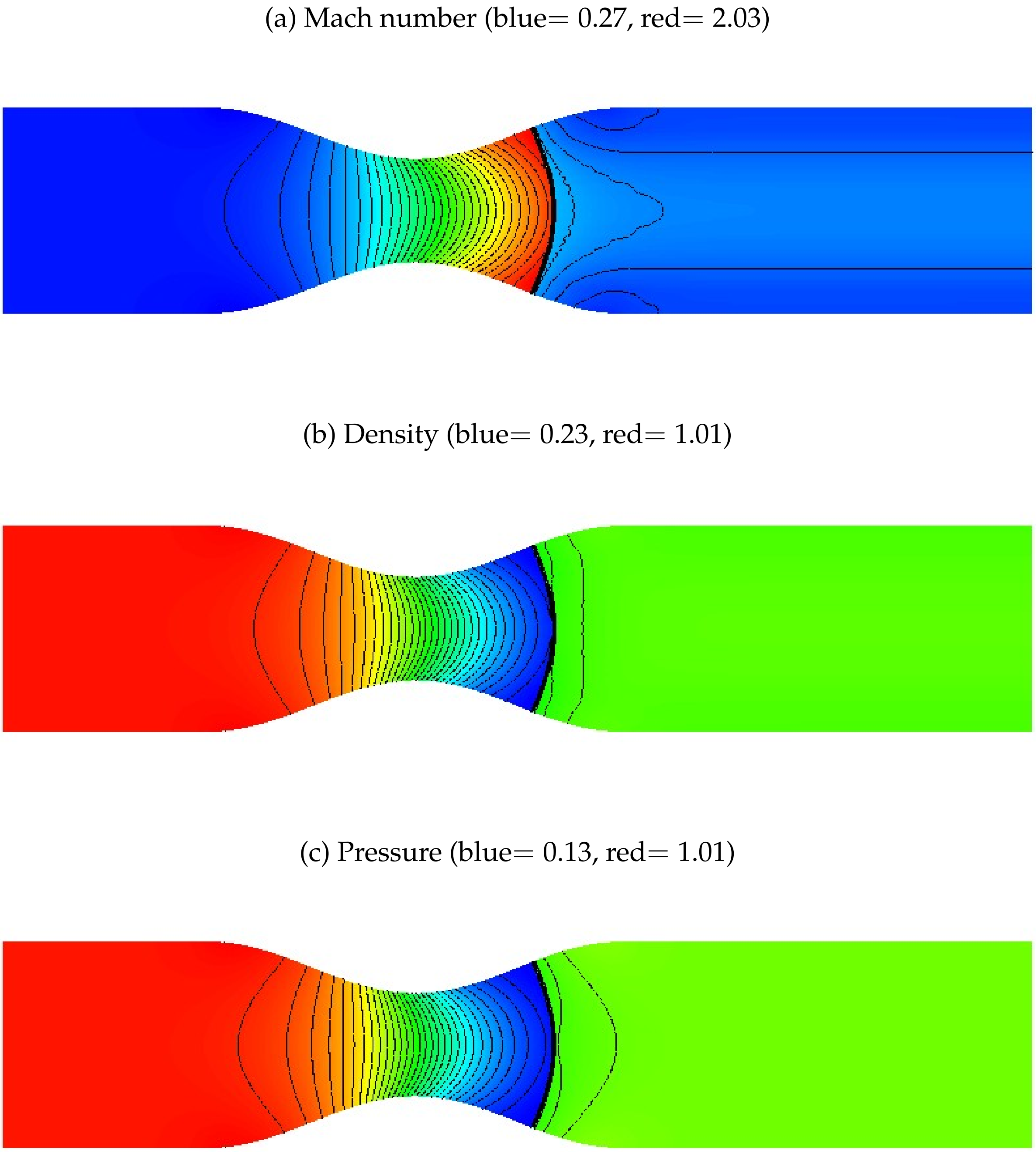


(a) Mach number (blue= 0.27, red= 2.79)

(b) Density (blue= 0.16, red= 1.58)

(c) Pressure (blue= 0.05, red= 1.35)

Figure 9.10: Transonic nozzle flow (30 contours)

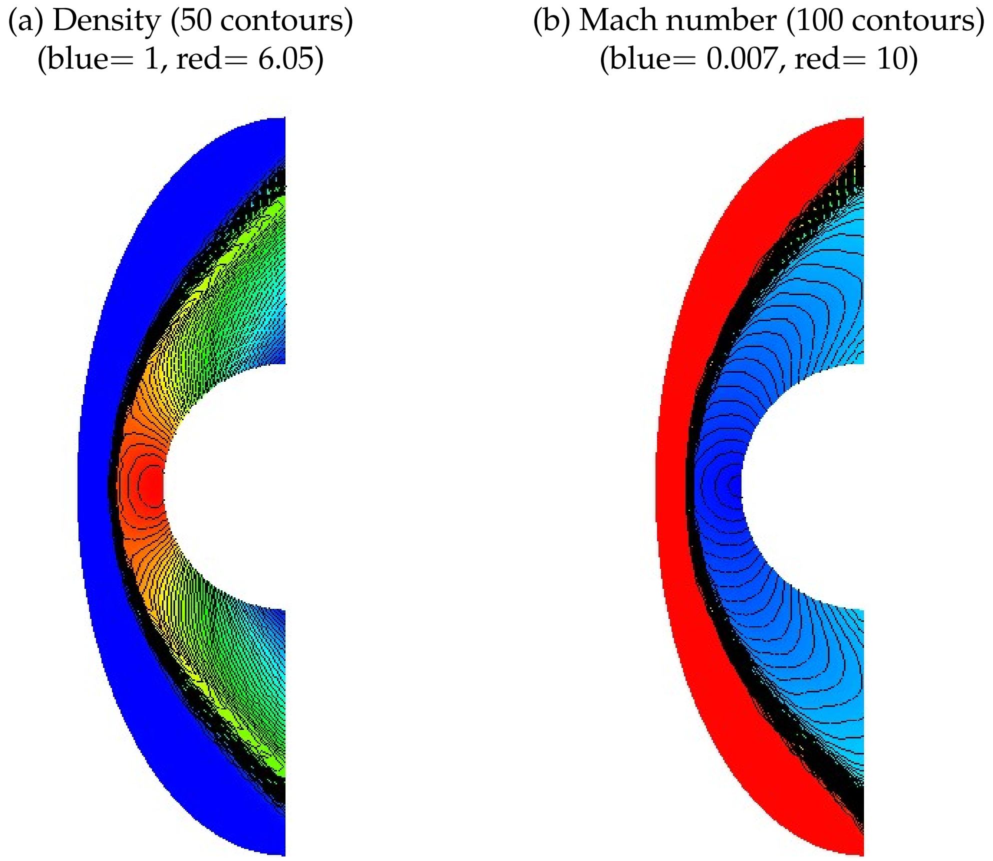


(a) Mach number (blue= 0.27, red= 2.03)

(b) Density (blue= 0.23, red= 1.01)

(c) Pressure (blue= 0.13, red= 1.01)

Figure 9.11: Nozzle flow with pressure outlet (30 contours)



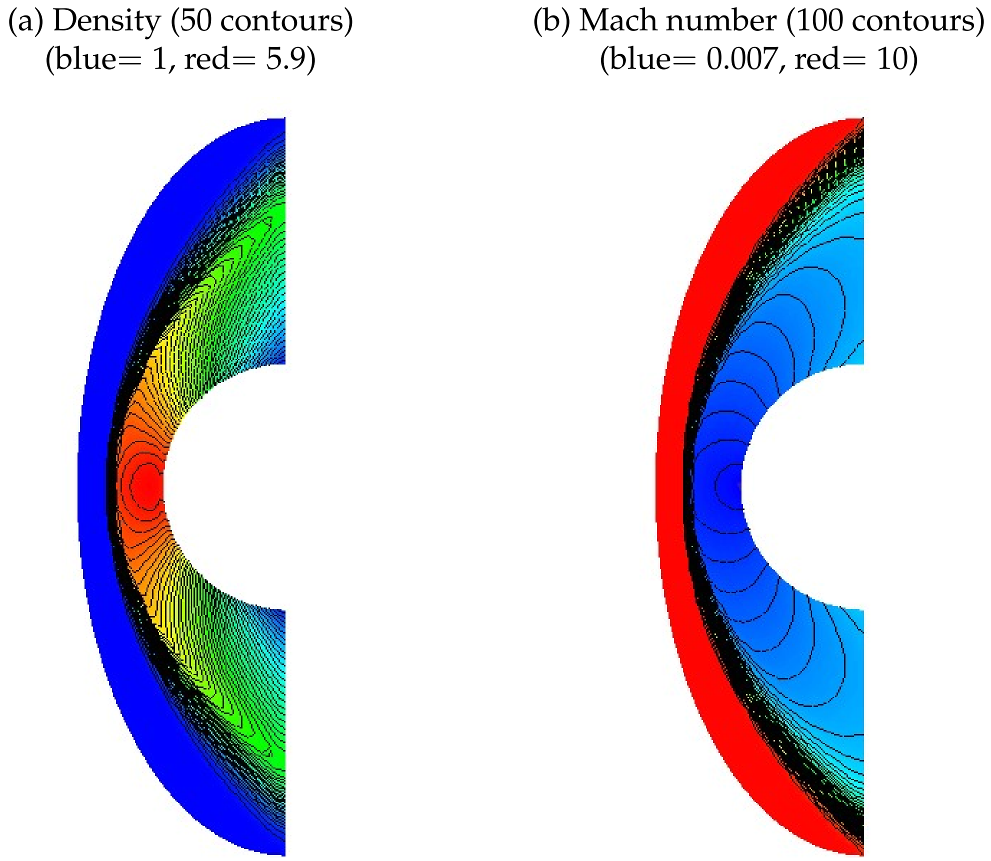
(a) Density (50 contours)
(blue= 1, red= 6.05)

(b) Mach number (100 contours)
(blue= 0.007, red= 10)

Figure 9.12: Circular cylinder: Flux-corrected solution

the so-called carbuncle phenomenon if they are applied to this test case. The carbuncle phenomenon manifests itself in spurious oscillations ahead of the circle. It is a well known fact that the oscillations become more pronounced when the grid is aligned with the bow shock. The simulations are performed on the coarse mesh illustrated in figure 9.14 (b), which is a copy of the mesh used in [74] and consists of 2,418 vertices and 2,300 quadrilateral elements. Ahead of the circle the mesh is almost perfectly aligned with the bow shock causing spurious oscillations which have been observed in [74] and the references therein.

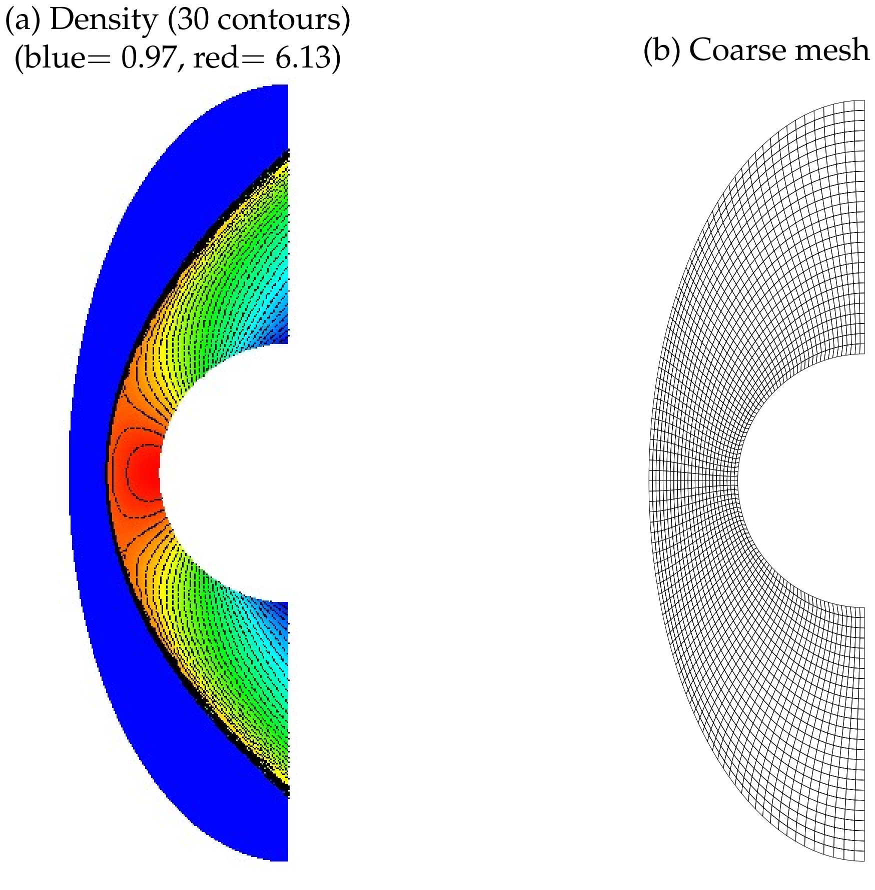
It follows from the numerical results that both the high-resolution scheme (compare figure 9.12) and the low-order scheme (compare figure 9.13) reported in this thesis exhibit convergence to steady-state despite the nearly infinite characteristic stiffness. Moreover, both schemes are immune to the carbuncle phenomenon without any ad-hoc tuning. To get a better resolved solution, the coarse mesh is refined two times. The solution computed by the high-resolution scheme on this mesh consisting of 37,269 vertices and 36,800 quadrilateral elements is plotted in figure 9.14 (a). This solution additionally indicates the performance of the scheme on fine meshes since the convergence rates are known to be mesh-dependent. This is a reason, why a multigrid procedure is employed by some authors. The interested reader is refered to [30] among others.



(a) Density (50 contours)
(blue= 1, red= 5.9)

(b) Mach number (100 contours)
(blue= 0.007, red= 10)

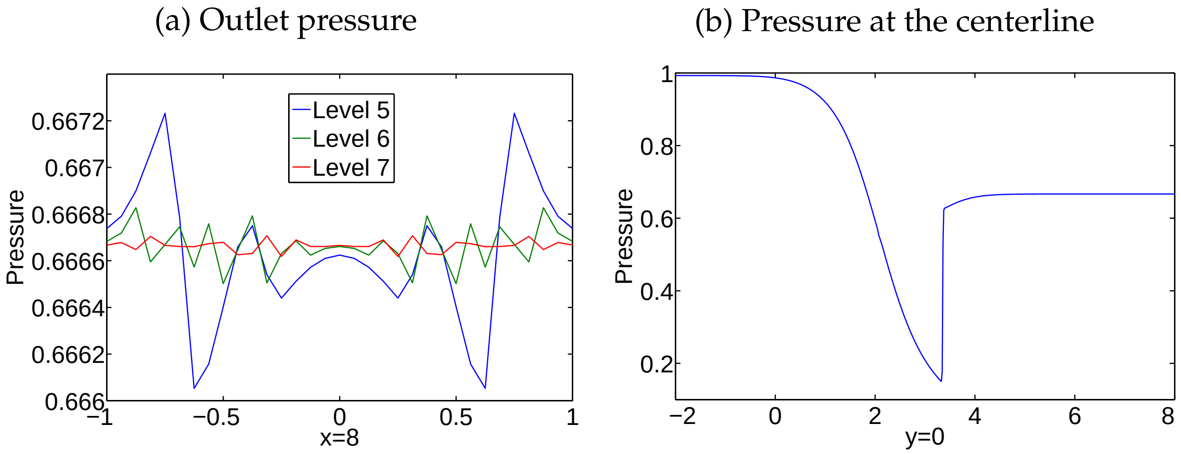
Figure 9.13: Circular cylinder: Low-order solution



(a) Density (30 contours)
(blue= 0.97, red= 6.13)

(b) Coarse mesh

Figure 9.14: Circular cylinder: Solution on grid level 3 and coarse mesh


(a) Outlet pressure

(b) Pressure at the centerline

Figure 9.15: Investigation of the pressure outlet boundary condition

9.6 Numerical Verification of Weak Boundary Conditions

A numerical treatment of boundary conditions that provides enhanced stability and robustness is one of the main goals of this thesis. Proper implementation of boundary conditions is crucial for the overall accuracy of the scheme since the errors arising from an inaccurate boundary discretization propagate into the interior of the domain. This section is therefore concerned with the numerical analysis of the proposed boundary techniques. The accuracy and competitiveness of the weak treatment of solid wall boundary conditions as compared to its strong counterpart was already analyzed in the finite volume framework [2]. Our numerical results reported in the sections above indicate that the proposed weak treatment of boundary conditions is unconditionally stable. A comparison of the weak wall boundary conditions, proposed in section 7.5.8, is drawn in the sections below. Weakly imposed boundary conditions are satisfied in an integral and possibly not pointwise sense. Recall that finite element basis functions have small support. Therefore, a weak treatment enforces boundary conditions locally. Last but not least, a comparison of weak and strong boundary conditions is performed below.

9.6.1 Inlet and Outlet Boundary Conditions

The accuracy of inlet and outlet boundary conditions is verified in this section. It is possible to examine the pressure outlet of the third simulation for the converging-diverging nozzle flow (compare section 9.4) as well as the scramjet inlet (compare section 9.3) as representative test cases of subsonic and supersonic boundary conditions. For the former test case, the prescribed outlet pressure is $P_{out} = \frac{2}{3}$. To verify the convergence of the outlet boundary condition, three mesh levels of bilinear elements are involved. Snapshots of the pressure at the outlet boundary for mesh levels five, six, and seven are reported in figure 9.15 (a). The relative error in the L_2-norm

$$E_2^{out} = \frac{||P - P_{out}||_{2,\Gamma_{out}}}{||P_{out}||_{2,\Gamma_{out}}} \tag{9.6}$$

Level	NVT	NEL	NVT$_{out}$	E_2^{out}	p^{out}
5	5313	5120	33	$2.50 \cdot 10^{-3}$	1.32
6	20865	20480	65	$1.00 \cdot 10^{-3}$	1.12
7	82689	81920	129	$4.62 \cdot 10^{-4}$	

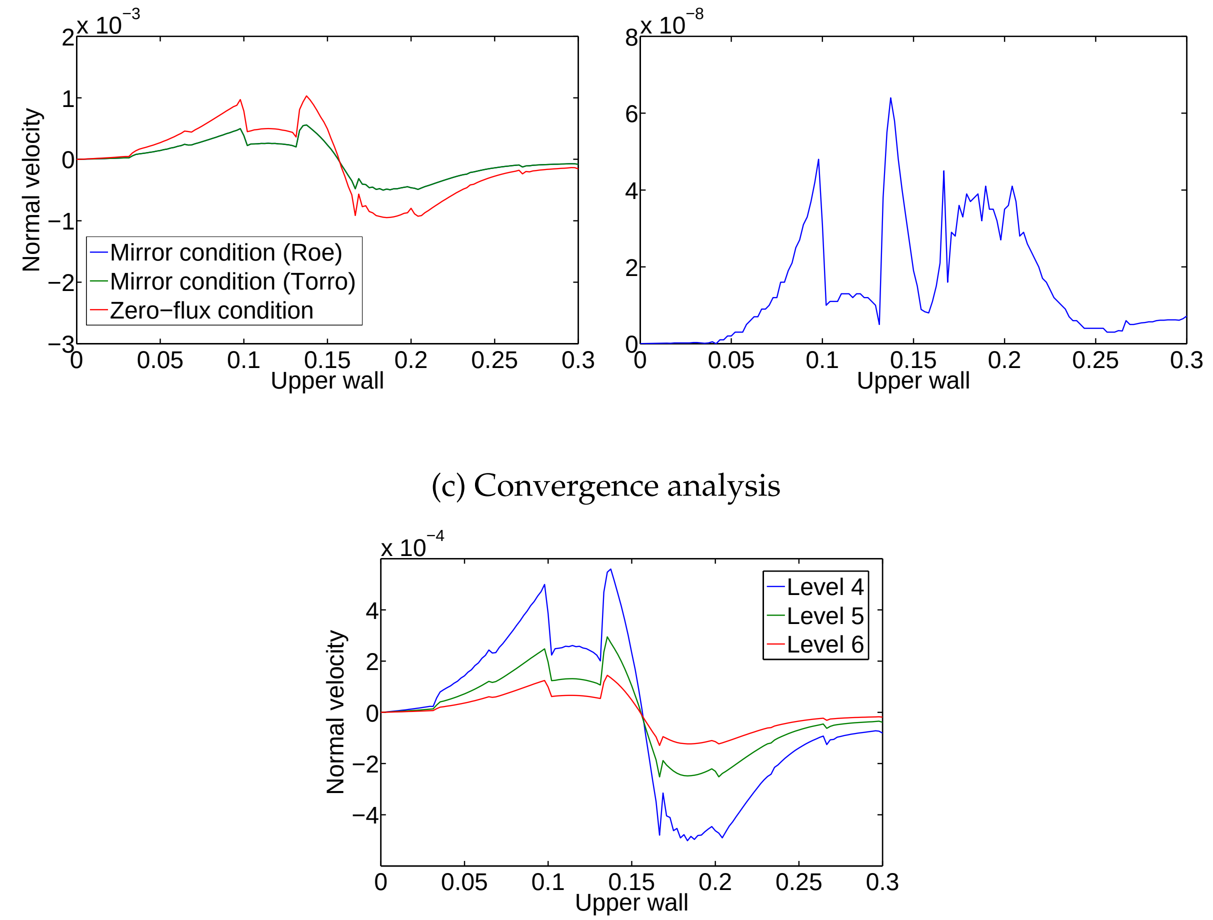
Table 9.6: Mesh properties and outlet boundary error analysis

as well as the empirical rate of convergence p^{out} are listed in table 9.6, where NVT$_{out}$ denotes the number of nodes at the outlet. Obviously, the boundary discretization error is very small, even on a relatively coarse grid. Similarly to the discretization error of the whole domain, the boundary discretization error vanishes with vanishing mesh size, which is important for grid convergence. To indicate the importance and influence of the outlet boundary condition on the whole flow, the pressure at the centerline of the nozzle is displayed in figure 9.15 (b).
For the scramjet inlet the supersonic inlet boundary condition is perfectly reproduced. The errors of the boundary Mach number, pressure, and first velocity components are zero, while the error in the second velocity component does not exceed $1.33 \cdot 10^{-11}$. Hence, no detailed analysis is required in this supersonic test case.

9.6.2 Wall Boundary Conditions

In section 7.5.8 two different weak treatments of solid walls are mentioned. On the one hand, one may enforce the free-slip condition by cancelling the normal velocity components (compare equation (7.42)), which turns out to be an easy but robust and accurate way. On the other hand, a more physical constraint in terms of the reflection or mirror condition may be imposed (compare equation (7.44)) and the related Riemann problem should be solved by either an approximate Riemann solver or its exact counterpart. These two basic approaches are compared for the GAMM channel test case already considered in section 9.1.

It is difficult to verify the free-slip condition on a curved boundary since the computational domain is an approximation of the physical one and the normal vector does not exist at the nodes. To circumvent this trouble it is possible to consider a wall boundary formed by a straight line. Therefore, the normal vector is defined at every point of the boundary. This simplifies the evaluation significantly. For the analysis we choose the upper wall boundary of the GAMM channel domain. Obviously, the normal velocity corresponding to the upper wall is simply the second velocity component. The characteristics of the involved mesh are reported in table 9.1. The mesh levels four, five, and six are used. First of all, we compare the zero-flux condition (7.42) with the mirror condition. Snapshots of the normal velocity computed by both schemes are depicted in figure 9.16 (a). The mirror condition is embedded either into the approximate Riemann solver of Roe or the exact one of Toro [92]. It turns out that the mirror as well as the zero-flux solid wall boundary condition yield comparable results with very small errors, while the mirror condition gives rise to slightly better approximations. Moreover, the solutions computed by the approximate Riemann solver and the exact solver



(a) Comparison of different wall boundary conditions (mesh level five)

(b) Difference of approximate/exact Riemann solver (mesh level five)

(c) Convergence analysis

Figure 9.16: Analysis of wall boundary conditions at the upper wall boundary

of Toro are almost the same. The difference between the exact and the approximate Riemann solvers are illustrated in figure 9.16 (b) in terms of $|v_n^{Toro} - v_n^{Roe}|$. In the following it remains to consider the mirror condition. A grid convergence analysis of the above configuration can be found in figure 9.16 (c) and the relative errors are listed in table 9.7. Obviously, the error in the normal velocity decreases with decreasing mesh size and the empirical order of convergence is about 1.0. This result is almost the same as the empirical rate of convergence associated with the whole domain, compare table 9.1.

Summarizing the results, we conclude that the accuracy of the wall boundary treatment is comparable to the interior discretization, which is essential for the overall accuracy. The wall boundary discretizations, proposed in this thesis, yield comparable results, while the mirror condition turns out to be slightly more accurate. At the same time it is much more complicated and sometimes slightly less robust than the zero-flux condition. Moreover, there are practically no differences between the approximate Riemann solver of Roe and the exact Riemann solver of Toro applied to the mirror condition. Of course one can identify and analyze the small errors and differences in the solutions computed by the different wall boundary conditions. In a practical application they are negligible.

Level	NVT$_{out}$	E_2^{Lim}	p^{Lim}
4	145	$3.95 \cdot 10^{-4}$	0.99
5	289	$1.99 \cdot 10^{-4}$	1.00
6	577	$9.94 \cdot 10^{-5}$	

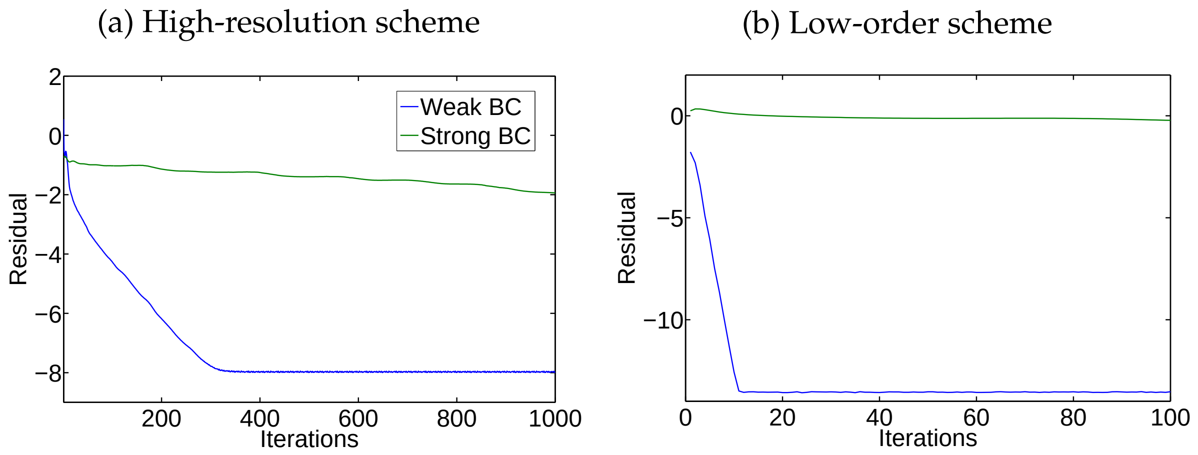
Table 9.7: Wall boundary error analysis

9.6.3 Strong vs. Weak Boundary Conditions

Alternatively to the proposed weak Neumann-type boundary treatment, boundary conditions may be implemented in a strong sense. For the comparison to be drawn in this section the predictor-corrector algorithm described in [47, 60], which has proven to be an accurate and reliable technique in implicit schemes, is implemented in the semi-implicit time stepping scheme developed so far. The solid wall boundary condition is replaced by the Riemann solver of Toro to equalize the wall boundary treatment. Both codes are applied to the GAMM channel test case reported in section 9.1 and the computations are performed on mesh level four.

In figure 9.17 the convergence histories of both the weak and the strong type of boundary conditions are reported. The CFL number of the simulation, where weak boundary conditions are used, is $CFL = \infty$ since it was already shown that there are no time step restrictions. In contrast, the strong treatment of boundary conditions hampers convergence to steady-state. In fact, the scheme diverges for $CFL \geq 1$. In the reported computation, the CFL number is set to $CFL = 0.8$. It follows from the convergence histories of both boundary treatments depicted in figure 9.17 that the strong version is less competitive since the CFL number is bounded from above by one. Therefore, several thousand iterations are required to achieve convergence if there is any. The comparison of both types of boundary conditions is best suited for the low-order schemes (compare figure 9.17 (b)) since the nonlinearity of the limiter, which degrades the rate of convergence, is not involved in the solutions.

The computational observations can also be justified in a more mathematical way. While the weak type of boundary conditions can be incorporated into the matrix in a natural way, the strong enforcement of boundary conditions requires that off-diagonal matrix entries be eliminated. This is similar (but not completely equivalent) to dropping the Jacobian in (8.6) at boundary nodes. It makes the scheme explicit at boundary nodes and therefore gives rise to the well known stability restrictions. In a fully implicit code this does not completely hold true and the stability bound may be slightly larger since the solution is updated in an outer fixed point iteration loop. Nevertheless, the predictor-corrector algorithm remains semi-explicit. On the other hand, off-diagonal matrix entries also have to be deleted. The elimination of off-diagonal matrix entries as well as solution estimates, involved in the predictor-corrector algorithm, introduce some explicitness. Moreover, many iterations are required in each pseudo time step to compute the solution update, which makes the fully implicit approach computationally expensive. In summary, it is shown in this section that the weak Neumann-type of boundary conditions yields accurate results and far superior stability and convergence compared to its strong counterpart.



(a) High-resolution scheme

(b) Low-order scheme

Figure 9.17: Nonlinear convergence history for weak and strong boundary conditions in logarithmic scale

10 Two-Fluid Model

Now that the performance of the developed numerical scheme has been verified for the Euler equations, we can focus on the analysis of the two-fluid model featuring a two-way coupling. The first goal of this chapter is to validate the code and compare with benchmark computations from the literature. At the same time it is shown that the discretization of the two-fluid model features most of the properties of the single-phase gas code. The nonlinear convergence analysis is therefore another important goal. It follows from the numerical results that the nonlinear convergence behavior is qualitatively comparable to the single-phase gas computations. The rate of nonlinear convergence improves with increasing CFL number. In spite of two conflicting non-differentiable nonlinearities, which act together in the case of the high-resolution scheme, the solution approaches steady-state in all computations. One can observe both a genuine unconditional stability and a high and stable convergence rate for very large or even infinite CFL numbers. The performance is highlighted for low Mach numbers and complex flow situations.

In the present chapter two test cases are studied. First the flow in a jet propulsion nozzle (JPL nozzle) is investigated. This test case is characterized by subsonic flow at low Mach numbers in the converging part of the nozzle, which is accelerated up to the supersonic regime. We analyze the nonlinear convergence for both the low-order and the high-resolution scheme. Special attention is also paid to the influence of different mass fractions on the physics of both phases. The second computation deals with the reflection of a stationary shock wave at a ramp under purely supersonic conditions. Based on these computations, we study the effect of different particle sizes and mass fractions.

Both test cases involve the flow of nitrogen laden with small ceramic Al_2O_3 particles. The constants associated with such flow conditions are listed in table 10.1. At the inlet a chamber with a homogeneous mixture of uniformly distributed gas and particles is

Constant	Value	Unit
γ	1.4	
Pr	0.75	
μ	$2.76 \cdot 10^{-5}$	$\frac{kg}{m \cdot s}$
ρ_p	4000	$\frac{kg}{m^3}$
c_{pg}	1040	$\frac{J}{kg \cdot K}$
c_{vg}	743	$\frac{J}{kg \cdot K}$
c_{vp}	1380	$\frac{J}{kg \cdot K}$

Table 10.1: Physical constants

Level	NVT	NEL
1	182	277
2	640	1108
3	2387	4432
4	9205	17728

Table 10.2: Mesh properties JPL nozzle

assumed. The temperatures and velocities are in equilibrium. Let us characterize the flow further by the mass fraction

$$\phi = \frac{\alpha_p \rho_p}{\alpha_p \rho_p + \alpha_g \rho_g} \tag{10.1}$$

to measure the amount of particles in the chamber. In all computations reported below, the penalty parameter is set to $\sigma = 10^8$. Moreover, the high-resolution scheme is always initialized by the low-order solution, which is a sufficiently accurate estimate to render the linearization adequate.

10.1 Jet Propulsion Nozzle Flow

The converging diverging nozzle is characterized by the presence of curved boundaries, a quite steep entrance of 45° degrees, and a relatively thin throat, which results in the large acceleration from Mach number $M = 0.1$ at the inlet up to Mach number $M = 2$ (depending on the mass fraction). Therefore, a large characteristic stiffness arises.

The geometry of the nozzle is adopted from [34], where the upper and lower boundaries serve as solid walls. The left and right boundaries correspond to the inlet and outlet of the domain, respectively. At the subsonic inlet, the free stream conditions from table 10.3 are prescribed in contrast to the supersonic outlet, where no boundary condition is needed. The domain is covered by an unstructured triangular coarse grid (compare figure 10.2), which is refined several times for the simulations. The mesh properties at different levels are depicted in table 10.2. Computations for various mass fractions are performed to assess the influence on the gas phase and to compare the results with data reported in the literature. First we compute five solutions with $\phi = 0.0$ (pure gas), 0.1, 0.5 and $d = 1\mu m$, $20\mu m$ on mesh level four. For comparison figure 10.1 displays the properties of a pure gas flow. Figure 10.3 illustrates the Mach number distribution for the different mass fractions and particle diameters.

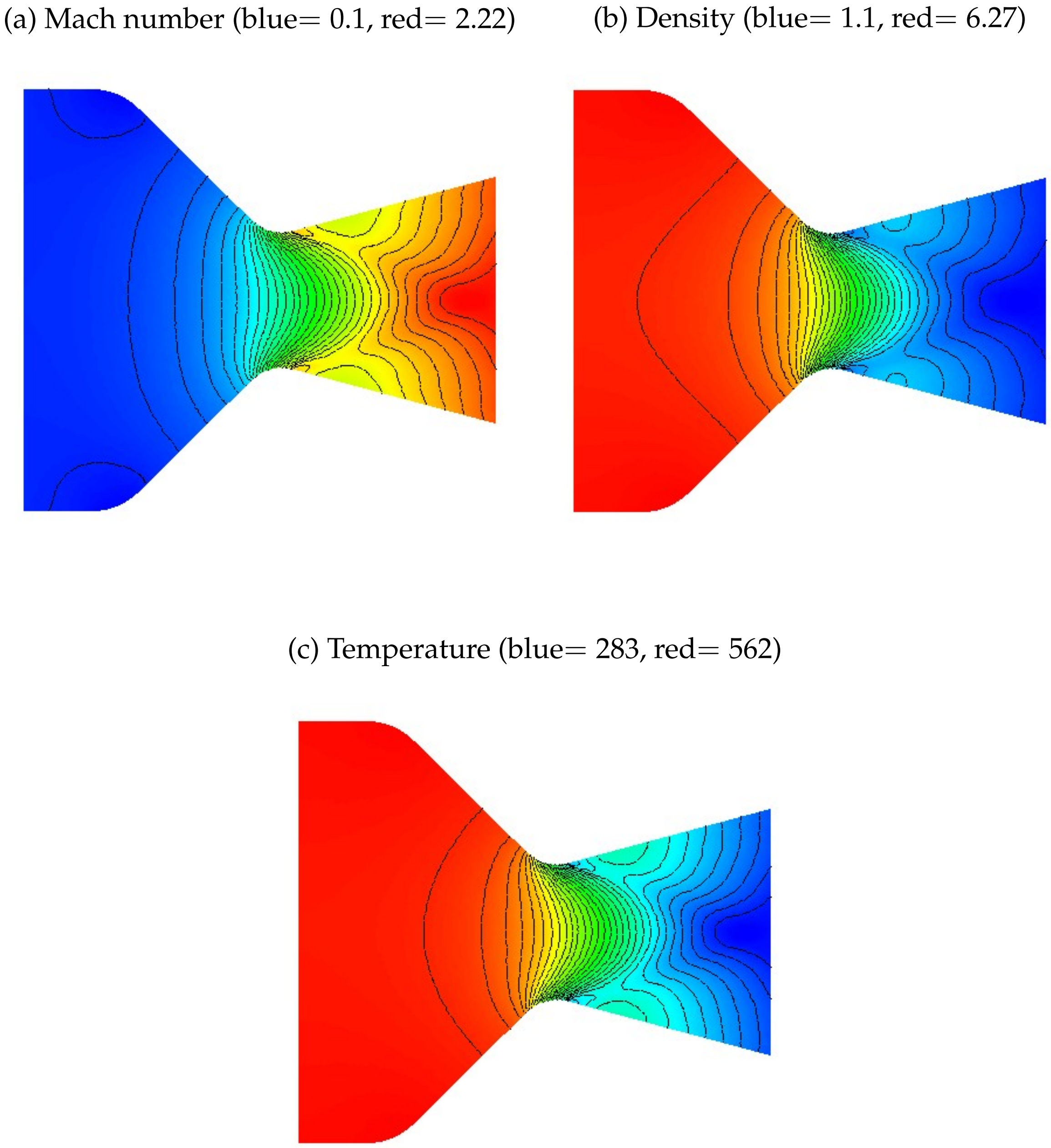
We observe that the influence of the particles on the gas Mach number increases with increasing mass fraction. The interfacial area and the amount of drag increase due to larger volume fractions at higher mass fractions, which decelerates the gas and decreases the Mach number. This physical argumentation is clearly confirmed by the numerical solutions, see figures 10.3 and 10.1 (a). Effective gas density, pressure, and temperature slightly increase with increasing mass fraction. Snapshots of the quantities

Quantity	Value	Unit
M_∞	0.2	
ρ_∞	6.0708	$\frac{kg}{m^3}$
P_∞	10^6	Pa
$\mathbf{v}_\infty$	$\sqrt{\frac{\gamma P_\infty}{\rho_\infty}}\begin{pmatrix} M_\infty \\ 0 \end{pmatrix}$	$\frac{m}{s}$
ϕ	0.1, 0.5	

Table 10.3: Free stream conditions of the JPL nozzle flow

Quantity	Value	Unit
M_∞	2	
ρ_∞	6.0708	$\frac{kg}{m^3}$
P_∞	10^6	Pa
$\mathbf{v}_\infty$	$\sqrt{\frac{\gamma P_\infty}{\rho_\infty}}\begin{pmatrix} M_\infty \\ 0 \end{pmatrix}$	$\frac{m}{s}$
ϕ	0.1, 0.3	

Table 10.4: Free stream conditions of the oblique shock wave


(a) Mach number (blue= 0.1, red= 2.22)

(b) Density (blue= 1.1, red= 6.27)

(c) Temperature (blue= 283, red= 562)

Figure 10.1: JPL nozzle: Pure gas flow physics (30 contours). Density in $\frac{kg}{m^3}$, temperature in K

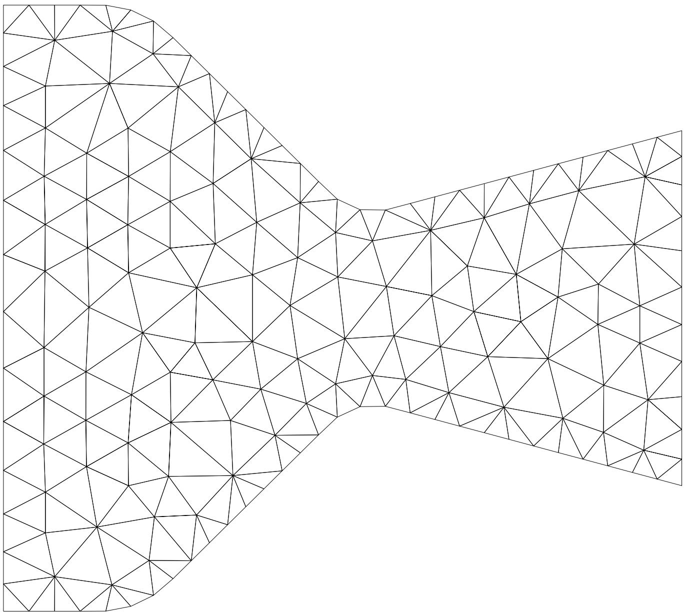
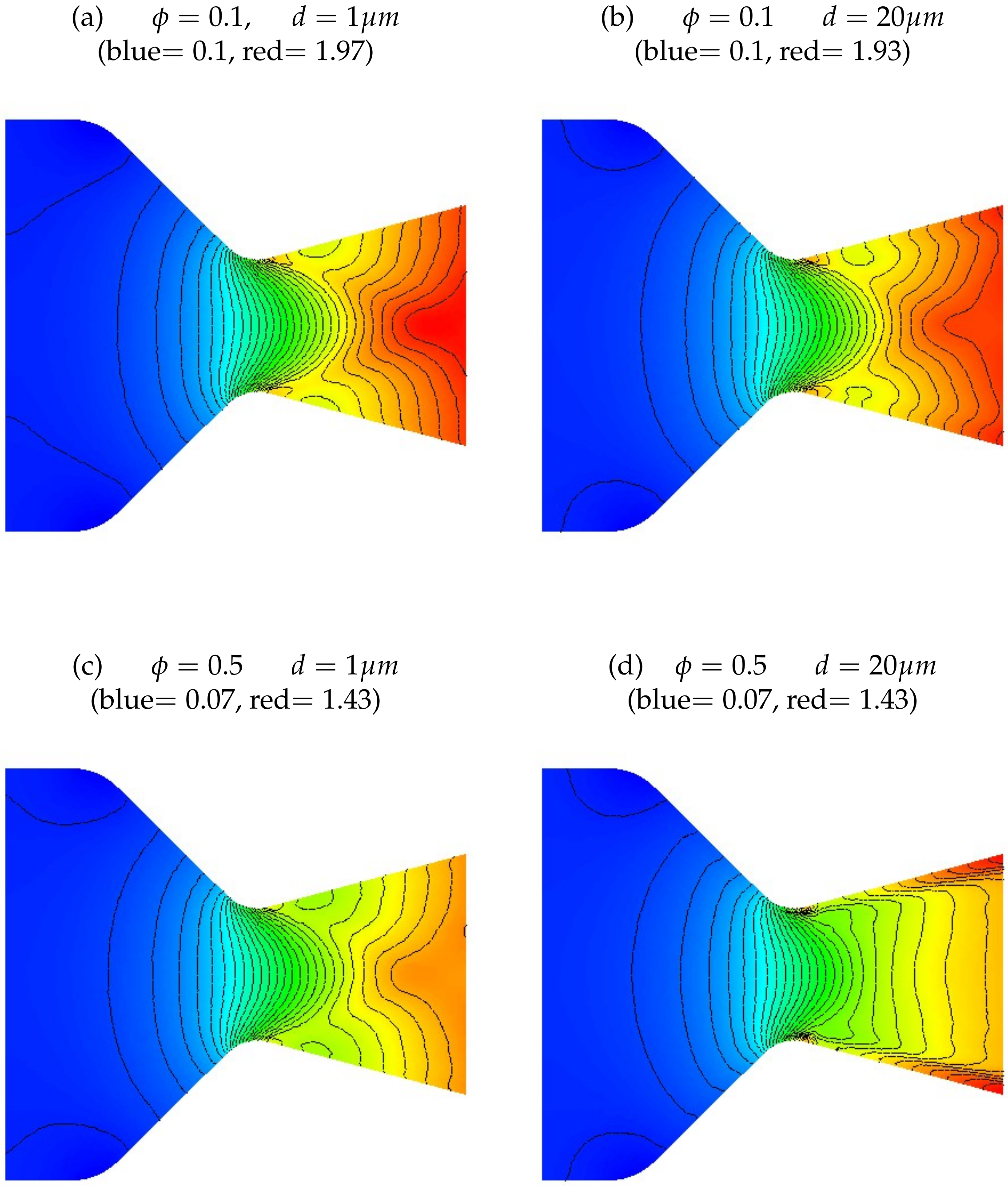


Figure 10.2: JPL nozzle: Coarse mesh

of interest at the centerline are depicted in figure 10.4. The results are in a good qualitative agreement with the observations of Nishida and Ishimaru [66] and Chang [10], although the configurations are not exactly equivalent.

Figures 10.3 (c) and (d) also illustrate the influence of the particle diameter on the gas Mach number contours. It is shown by the contour lines that small particles mimic the gas behavior. Despite a much higher volume fraction of the larger particles downstream and in the throat, compare figure 10.5 (b), the gas Mach number is approximately the same or even slightly higher (at the throat) than in the flow laden with smaller particles (see figure 10.5(a)). At first glance this looks surprising. The interfacial area of the flow with $d = 20\mu m$ is indeed smaller than the corresponding interfacial area of the flow laden with particles of diameter $d = 1\mu m$ (for the same volume fraction). Therefore, the magnitude of interfacial drag increases with decreasing particle diameter. Hence, smaller particles are more capable of decelerating the gas than their larger counterparts, which compensates the different magnitude of the volume fractions of both flows. A Mach number increase for larger particles was also observed in [10].

The particle distributions for the mass fraction under consideration, and particle diameters of $d = 1\mu m$, $20\mu m$ can be compared in figure 10.6. A comparison of the two different particle diameters indicates a particle clustering at the walls in the converging part of the nozzle for $d = 20\mu m$. In contrast to the gas phase, the particle velocity is not linked to a pressure term by a constitutive equation, which inhibits such a clustering. The particulate phase is only coupled to the gas pressure by the interfacial forces. A larger particle diameter results in less drag and temperature exchange (for the same volume fraction) due to the smaller interfacial area. Therefore, larger particles are less influenced by the gas than smaller ones, which more or less mimic the gas behavior. The larger particles are therefore less deflected to the centerline by the gas flow, or more precisely due to the pressure gradient, and a larger amount of particles hit the wall. At the



(a) $\phi = 0.1, \quad d = 1\mu m$
(blue= 0.1, red= 1.97)

(b) $\phi = 0.1 \quad d = 20\mu m$
(blue= 0.1, red= 1.93)

(c) $\phi = 0.5 \quad d = 1\mu m$
(blue= 0.07, red= 1.43)

(d) $\phi = 0.5 \quad d = 20\mu m$
(blue= 0.07, red= 1.43)

Figure 10.3: JPL nozzle: Mach numbers for $\phi = 0.1, 0.5$ and $d = 1\mu m, 20\mu m$ (30 contours)

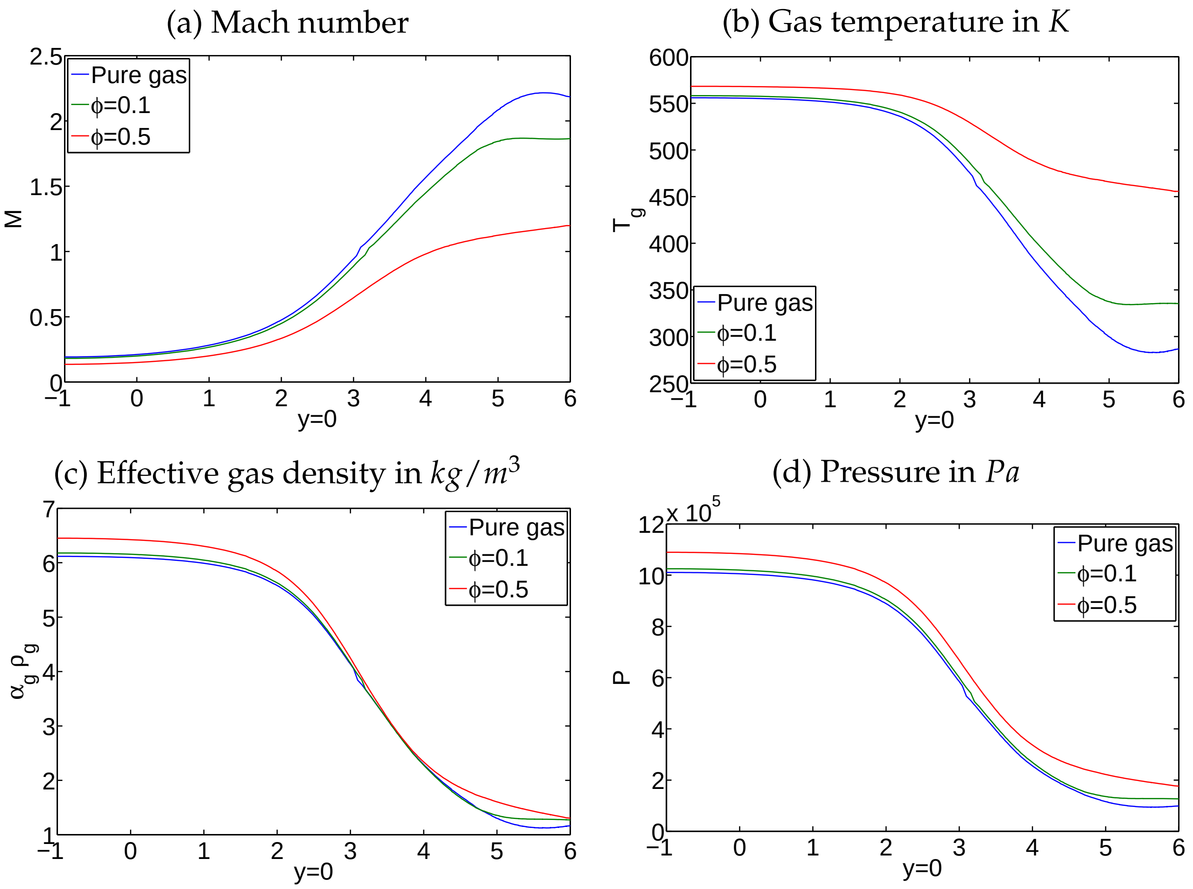


Figure 10.4: JPL nozzle: Gas properties at centerline for $d = 20$ and various mass fractions

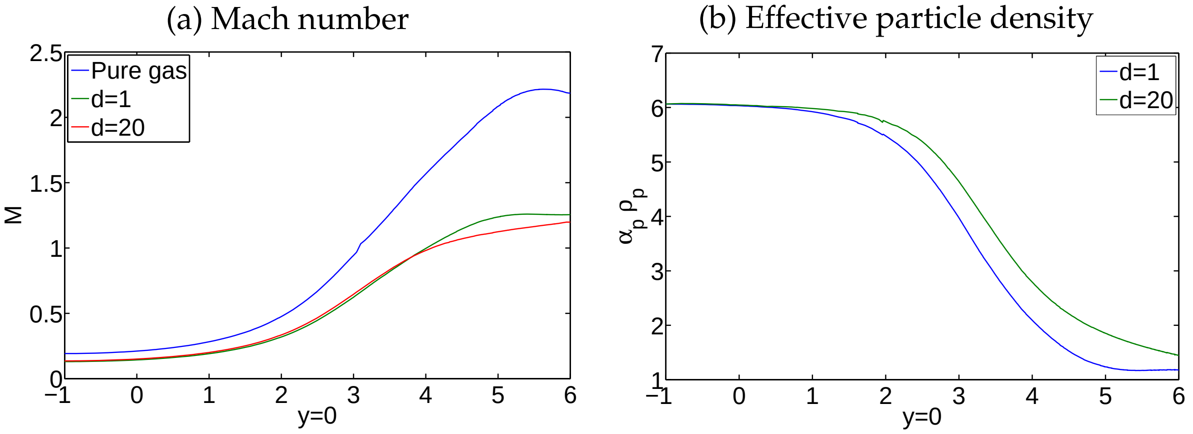
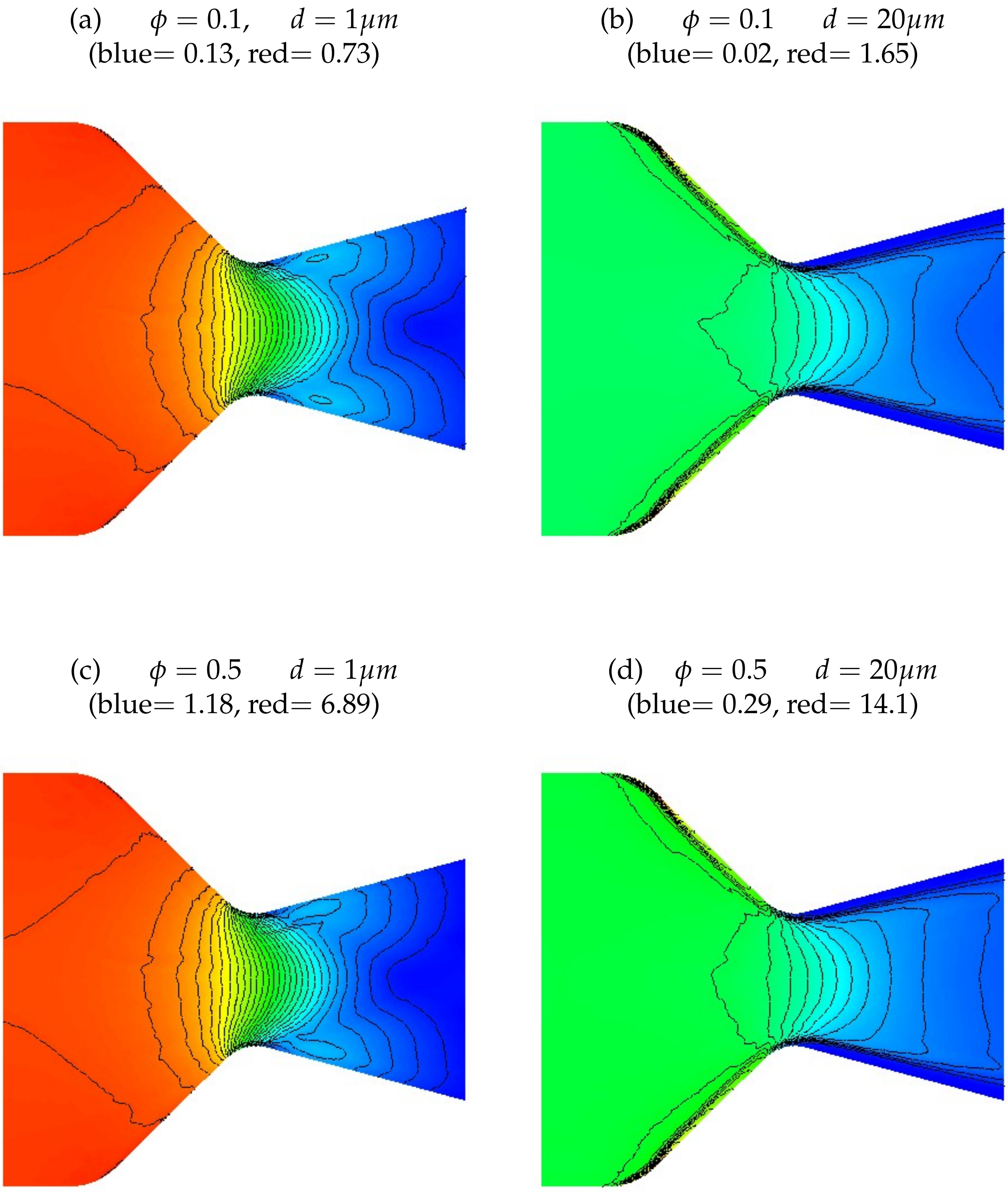


Figure 10.5: JPL nozzle: Effective particle density in $\frac{kg}{m^3}$ and Mach number at $\phi = 0.5$



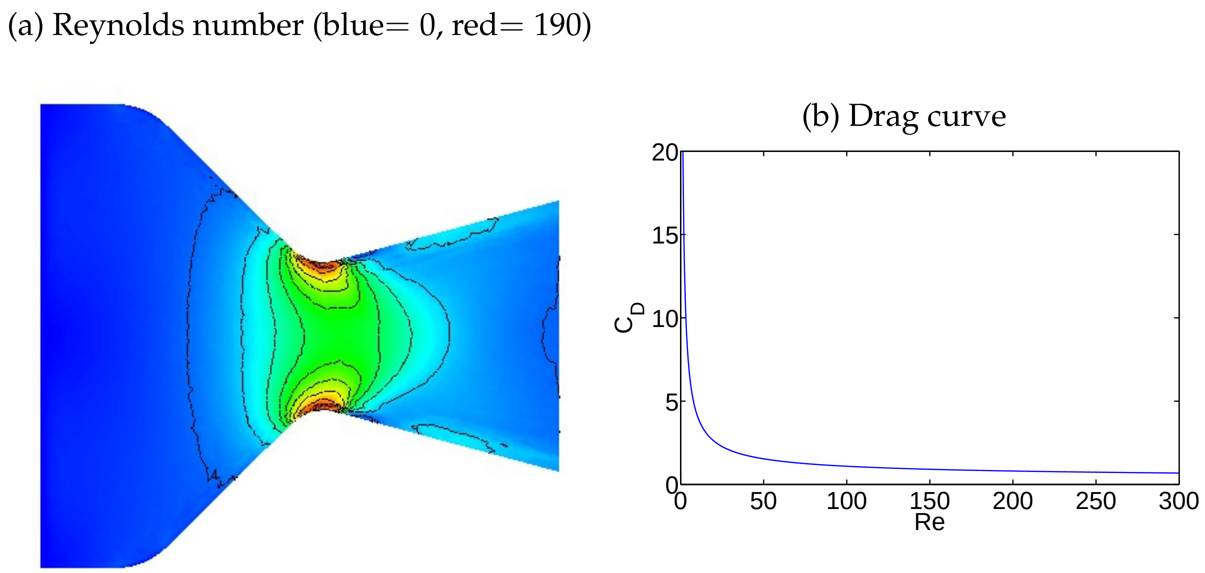
(a) $\phi = 0.1, \quad d = 1\mu m$
(blue= 0.13, red= 0.73)

(b) $\phi = 0.1 \quad d = 20\mu m$
(blue= 0.02, red= 1.65)

(c) $\phi = 0.5 \quad d = 1\mu m$
(blue= 1.18, red= 6.89)

(d) $\phi = 0.5 \quad d = 20\mu m$
(blue= 0.29, red= 14.1)

Figure 10.6: JPL nozzle: Effective particle densities in $\frac{kg}{m^3}$ for $\phi = 0.1, 0.5$ and $d = 1\mu m$, $20\mu m$ (30 contours)



(a) Reynolds number (blue= 0, red= 190)

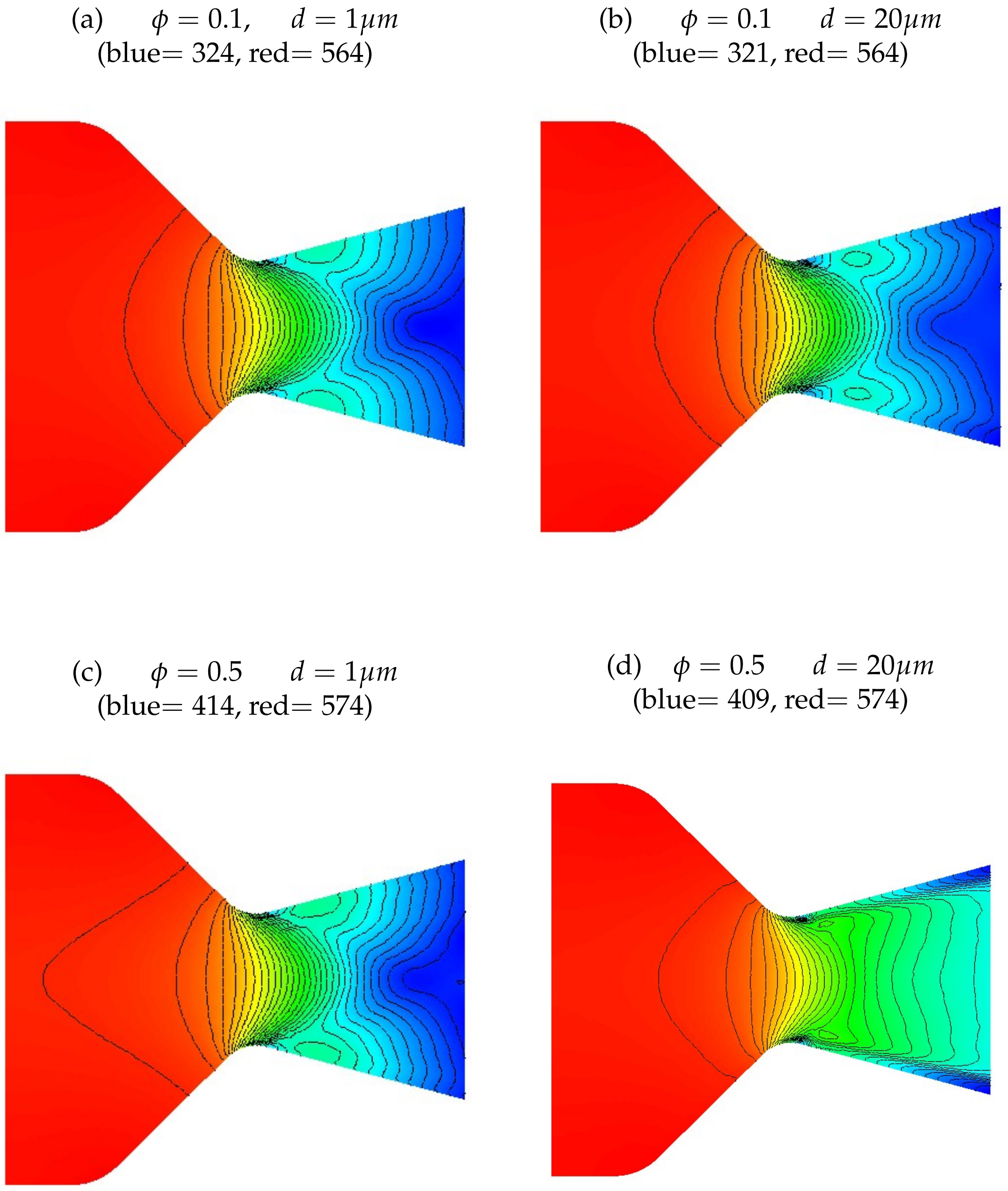
(b) Drag curve

Figure 10.7: Reynolds number for $\phi = 0.5$, $d = 20\mu m$ (15 contours) and drag curve

wall, the particles are deflected to the center of the nozzle by the boundary condition.

In contrast, the amount of particles in the vicinity of the walls in the diverging part of the nozzle decreases with increasing particle diameter, see figure 10.6. This can be explained from physical reasons in the same way as above. Moreover, the large Reynolds numbers at the throat (see figure 10.7 (a)) cause less drag due to the drag curve in figure 10.7 (b). Therefore, the larger particles are not able to follow the gas streamlines parallel to the walls. Nishida and Ishimaru [66], Chang [10], and Ishii and Umeda [34] claim that there are particle free layers in the vicinity of the walls, which increase with increasing particle size. In the present study, a small amount of particles is still present in the vicinity of the walls in the diverging part. Note that these particles vanish with decreasing mesh size, so that the added numerical diffusion can be seen as responsible for that phenomenon. The temperatures of both phases are almost equal, while the gas temperature is more affected by the particles, when the particle diameter or the mass fraction increases as indicated by figure 10.8. To summarize the observations reported above, we state that the illustrated effects are physically sensible and compare well with the results reported in the literature.

Last but not least, we examine the convergence behavior of the nonlinear iteration for $\phi = 0.1$ and $d = 1\mu m$ to rate on the implicit scheme and particularly the boundary conditions. Note that in this case the interface momentum and heat exchange are rather large due to the small particle diameter. The convergence history of the low-order scheme presented in figure 10.9 is qualitatively comparable to the single-phase computations already discussed in chapter 9. Once again, the Newton-like scheme with $CFL = \infty$ exhibits the best convergence rates, while the convergence rates deteriorate with decreasing CFL numbers. For $CFL = \infty$ the residual falls below 10^{-12} in eleven iterations, while the residual hardly decreases for $CFL = 1$. This demonstrates the superior performance of the proposed boundary treatment and of the whole semi-implicit



(a) $\phi = 0.1, \quad d = 1\mu m$
(blue= 324, red= 564)

(b) $\phi = 0.1 \quad d = 20\mu m$
(blue= 321, red= 564)

(c) $\phi = 0.5 \quad d = 1\mu m$
(blue= 414, red= 574)

(d) $\phi = 0.5 \quad d = 20\mu m$
(blue= 409, red= 574)

Figure 10.8: JPL nozzle: Gas temperature in K for $\phi = 0.1, 0.5$ and $d = 1\mu m, 20\mu m$ (30 contours)

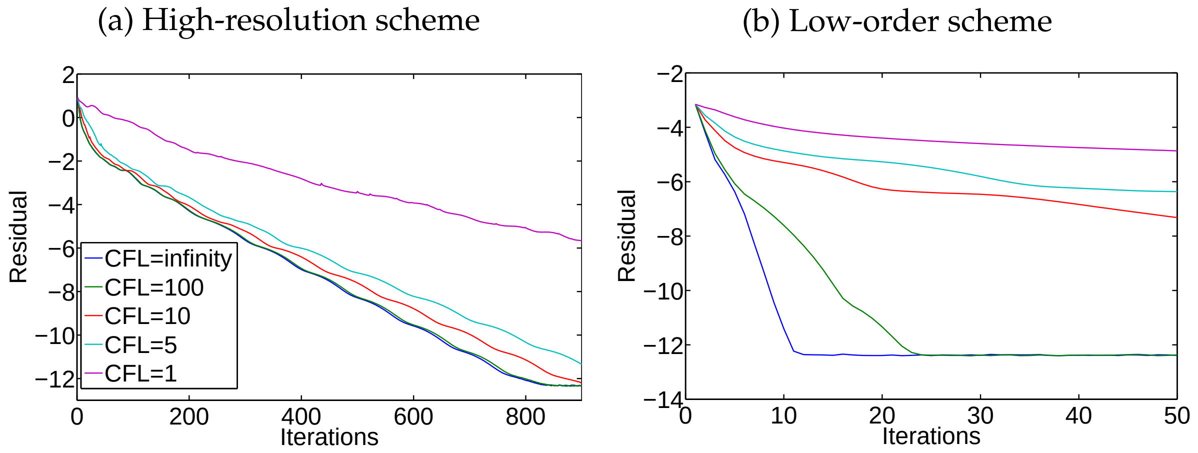


Figure 10.9: JPL nozzle: Nonlinear convergence history in logarithmic scale for different CFL numbers on mesh level three

scheme in spite of the strong nonlinearity of the interfacial transfer terms. The excellent convergence behavior clearly justifies the use of implicit schemes.

In the computations with the high-resolution scheme convergence is also reached and unconditional stability can be observed. The results of the computations with $CFL = \infty$ and $CFL = 100$ are almost the same, while the former case exhibits slightly faster convergence. Obviously, the convergence rates deteriorate with decreasing CFL numbers as in the single-phase case. The convergence histories are qualitatively comparable with the single-phase computations. In agreement with figure 9.9 moderate CFL numbers also yield satisfactory convergence rates and the performance improvement with increasing CFL numbers stagnates at $CFL = 100$. Moreover, it is slightly less pronounced than in the single-phase gas computations. Two conflicting nonlinearities due to the interfacial exchange terms and the correction factors are present, which decelerates convergence. On the other hand, the deterioration of convergence rates due to the interfacial coupling remains relatively small. Nevertheless, the rate of convergence and the performance of the scheme are not affected if the CFL number exceeds some upper bound, as observed in [93] for the Euler equations.

10.2 Oblique Shock Reflection

The second test case in this chapter is a purely supersonic wave reflection at a ramp of angle 10°. In contrast to the JPL nozzle flow, a shock arises in the solution and is reflected at the ramp. At the supersonic inlet boundary (left side) the free stream conditions from table 10.4 are prescribed, while no boundary condition is involved at the supersonic outlet boundary (right side). The upper and lower boundaries consist of solid walls. The goal of this section is to study the influence of the particle diameter as well as that of the mass fraction on the wave position. A comparison with a pure gas flow is performed for the purpose of code validation. Although the already analyzed JPL nozzle flow is a much more complicated test case, the current benchmark is also reported in the literature and allows an additional validation of the code in terms of the wave position. For a pure nitrogen flow, which is characterized by table 10.1, the analytical solution downstream the shock can be determined by shock wave theory [1]. It reads

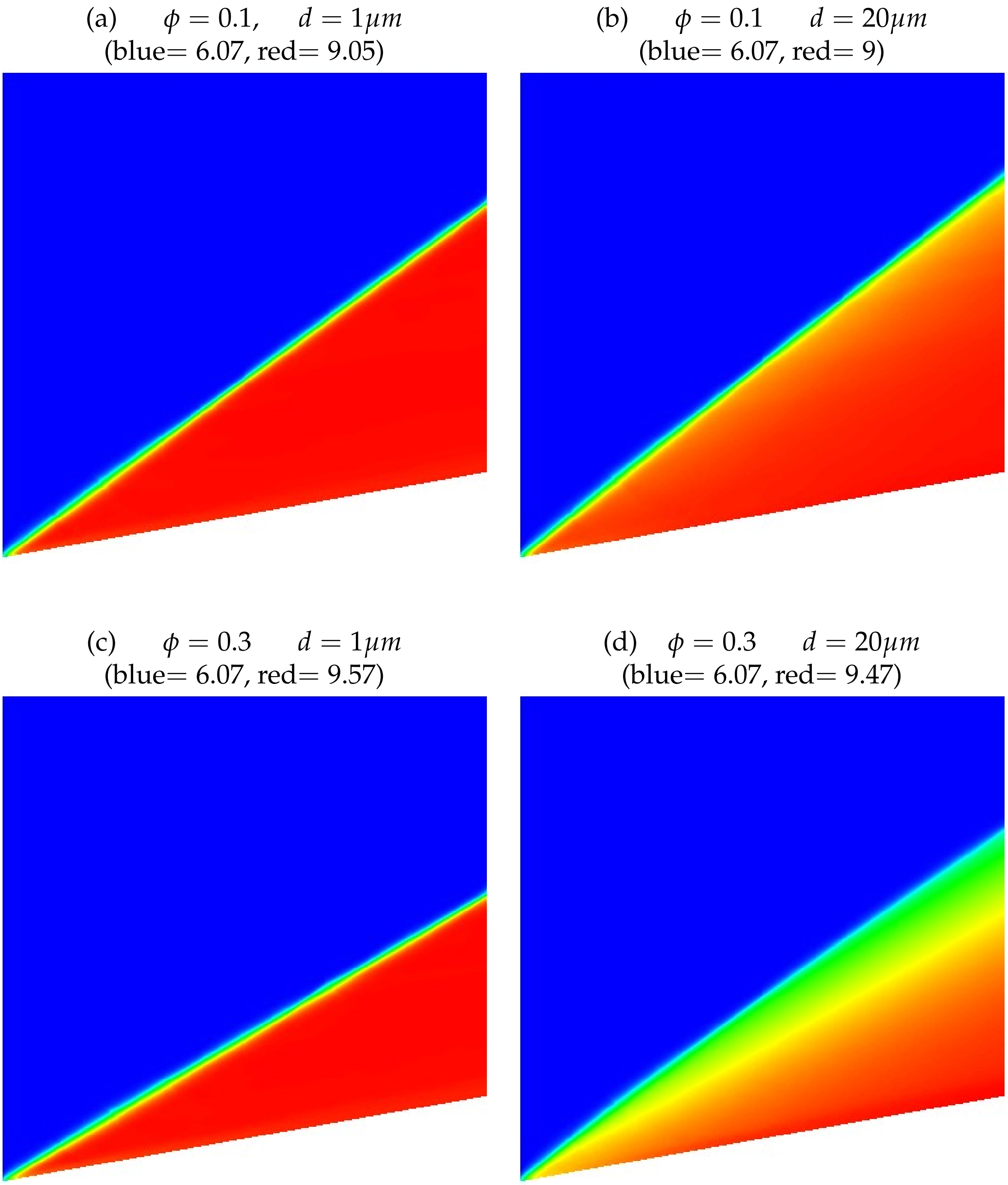
$$\rho_R = 8.8538 \, \frac{kg}{m^3} \qquad M_R = 1.6405 \qquad P_R = 1706578.6040 \, Pa$$

with wave angle $\beta = 39.3139°$. The particulate phase consists of solid Al_2O_3 ceramic particles, which possess the physical properties listed in table 10.1.

We compute numerical solutions of the configuration described so far for mass fractions $\phi = 0.1, 0.3$ and particle diameters $d = 1\mu m, 20\mu m$. Figure 10.10 displays the effective gas densities, which are computed on an unstructured mesh consisting of 14,105 nodes and 27,904 triangles. The pure gas flow density is depicted in figure 10.11 (a) for comparison. We observe that the wave angle decreases with increasing mass fraction, while it increases with increasing particle diameter. The same observations on the wave position are reported in [80] and the decrease of the wave angle can be clearly observed by a comparison of figures 10.11 (a), 10.10 (a),(c) and 10.10 (b),(d). At the same time the increase of the wave angle due to increasing particle diameter is less obvious in figure 10.10 but still present and also observed in [80]. An increasing particle diameter has an additional effect on the gas phase. Figure 10.10 illustrates a smearing of the shock in the gas phase for larger particles. At first glance this seems to be due to an additional amount of diffusion. This behavior can in fact be explained by the physical nature of the particles. The particulate phase, if it is considered isolated, does not exhibit a shock at that position due to the lack of pressure. Moreover, it was already pointed out in section 10.1 that large particles are less inclined to follow the gas streamlines. The Reynolds number distribution (see figure 10.11 (b)) exhibits large Reynolds numbers in the vicinity of the shock, which results in less drag (compare figure 10.7 (b)) and clearly justifies the above explanation. At the same time, interfacial drag is still present and its magnitude and influence on the gas phase increases with increasing particle mass fraction. Hence, the shock in the gas phase is smeared and smoothed out by the influence of the particulate phase rather than by additional numerical diffusion.

Finally, we examine the convergence of the nonlinear iteration by a comparison of different CFL numbers for computations on a mesh consisting of 3,565 nodes and 6,976 triangles. The developed scheme turns out to be unconditionally stable for a suitable initial guess, which is provided by the low-order solution. To initialize the low-order computations a few pre-iterations with $CFL = 10$ are performed. Figure 10.12 shows the convergence histories for both schemes and the convergence behavior is qualitatively comparable to the formerly reported test cases. In the case of the low-order scheme the rate of convergence improves with increasing CFL number, while $CFL = \infty$ produces the best results since the relative error falls to 10^{-11} in about 40 iterations. In a simulation with $CFL = 1$, it hardly decreases in the first 100 iterations. The high-resolution scheme exhibits a similar behavior. Also in this case, the choice of $CFL = \infty$ offers the fastest convergence. The relative error falls to 10^{-8} in about 600 iterations after which the solution can be considered stationary, while $CFL = 100$ results in nearly the same rate of convergence. However, the rate of convergence deteriorates with decreasing CFL numbers.

In summary, the choice of large CFL numbers is usually preferable since the scheme



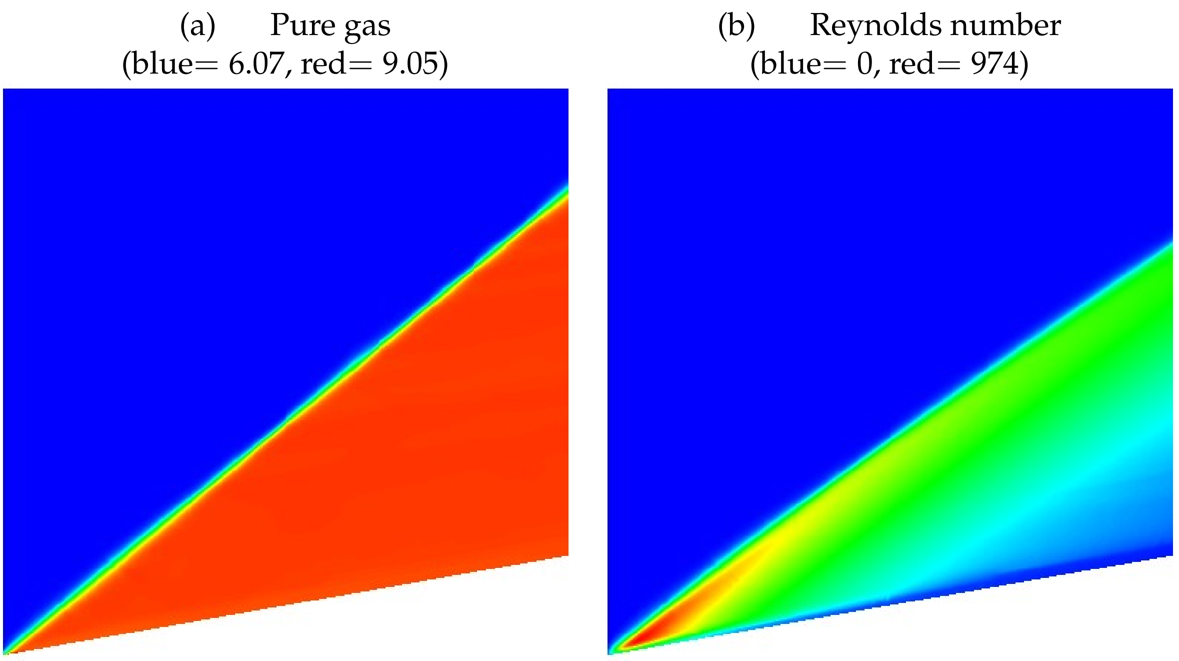
(a) $\phi = 0.1, \quad d = 1\mu m$
(blue= 6.07, red= 9.05)

(b) $\phi = 0.1 \quad d = 20\mu m$
(blue= 6.07, red= 9)

(c) $\phi = 0.3 \quad d = 1\mu m$
(blue= 6.07, red= 9.57)

(d) $\phi = 0.3 \quad d = 20\mu m$
(blue= 6.07, red= 9.47)

Figure 10.10: Oblique shock: Effective gas density in $\frac{kg}{m^3}$ for $\phi = 0.1, 0.3$ and $d = 1\mu m$, $20\mu m$

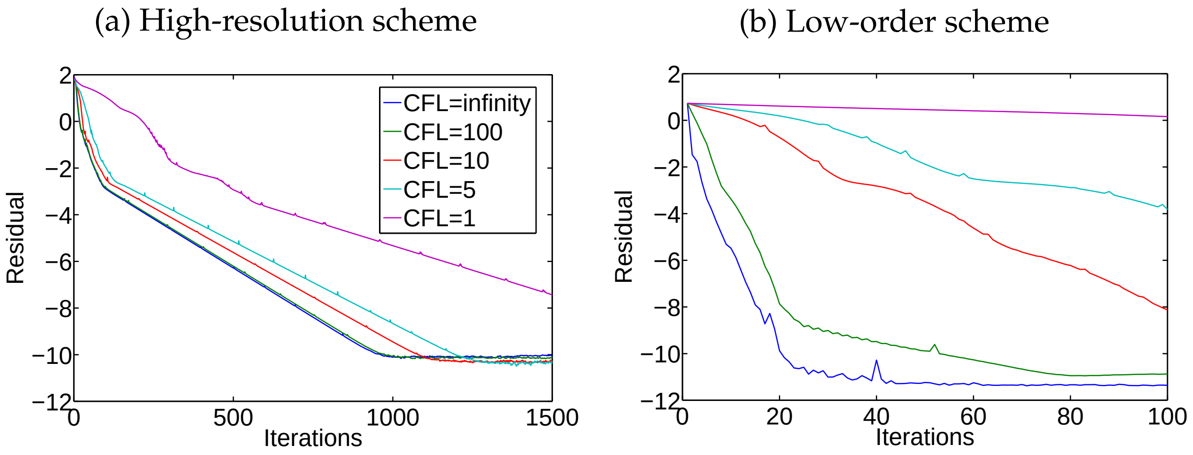


(a) Pure gas (blue= 6.07, red= 9.05)

(b) Reynolds number (blue= 0, red= 974)

Figure 10.11: Oblique shock: Pure gas flow density in $\frac{kg}{m^3}$ and Reynolds number with $\phi = 0.3$, $d = 20\mu m$

proves unconditionally stable in practical computations for a suitable initial condition, which can be easily obtained in terms of the low-order solution. Note that no parameter tuning is required. Hence, the developed scheme inherits the convergence behavior of the single-phase gas code in a qualitative sense.



(a) High-resolution scheme

(b) Low-order scheme

Figure 10.12: Oblique shock: Nonlinear convergence histories in logarithmic scale

10.3 Operator Splitting vs. Fully Coupled Solution Strategy

There are two basic approaches to an implicit numerical treatment of the interfacial transfer terms. Either operator splitting techniques as presented in chapter 6 and applied in [25] to stationary as well as non-stationary problems may be employed, or the equations may be integrated in time by the fully coupled implicit time integration as discussed in chapter 8. At first glance, the former approach significantly reduces the computational costs since the arising algebraic systems can be solved separately. In comparison with the fully coupled approach, where an $8\,NVT \times 8\,NVT$-system must be solved, operator splitting reduces the computational effort to the solution of two algebraic systems of size $4\,NVT \times 4\,NVT$ and a source term integration step. Operator splitting of Yanenko type is rather stable and robust but not suitable for the computation of stationary solutions. It does not allow the solution to approach steady-state and the final result depends on the (pseudo) time step. Douglas-Rachford operator splitting (compare section 6.1) is therefore investigated, while the Yanenko splitting may serve to compute an initial guess. The Douglas-Rachford splitting is known to be very robust at least in the framework of alternating direction implicit (ADI) iterative solvers.

It follows from the numerical results that the Douglas-Rachford splitting introduces an upper bound for the pseudo time step, which hampers the convergence to steady-state. This turns the promising reduction of computational costs into a drawback due to the increasing number of nonlinear iterations. Note that in a time-dependent application, where rather small time steps are essential for accuracy reasons, operator splitting is still competitive due to the low costs.

For the comparison of the fully coupled and the operator splitting approach both schemes were applied to the problem reported in section 10.1 on mesh level three. Mass fractions of $\phi = 0.5$ and particle diameters of $d = 20\mu m$ are prescribed for the numerical tests. The logarithmic plots of nonlinear convergence histories produced by both the fully coupled and the operator splitting approach are presented in figure 10.13. All computations are based on the low-order scheme, which is sufficient to examine the treatment of source terms. Obviously, the convergence of the fully coupled approach is far superior since it enables the use of large CFL numbers. On the contrary, the Douglas-Rachford splitting exhibits convergence only for small pseudo time steps. The scheme remains stable and does not converge for moderate CFL numbers, which can be explained by the large stiffness of the interfacial forces and the explicit treatment in the hyperbolic step. Since it is a well known fact that decoupled methods are subject to time step restrictions, the observations are as expected. The fully coupled implicit scheme converges in about 15 iterations for $CFL = \infty$ and exhibits convergence for all applied pseudo time steps. In contrast, Douglas-Rachford splitting does not converge for $\Delta t > 3 \cdot 10^{-4}$, which corresponds to a maximal CFL number of about $CFL = 9$. However, it converges for $\Delta t = 3 \cdot 10^{-4}$ in approximately 1500 iterations, which corresponds to 75 times more iterations.

Summarizing the results of the presented numerical study, we conclude that the unconditional stability of the fully coupled implicit time integration proposed in this thesis

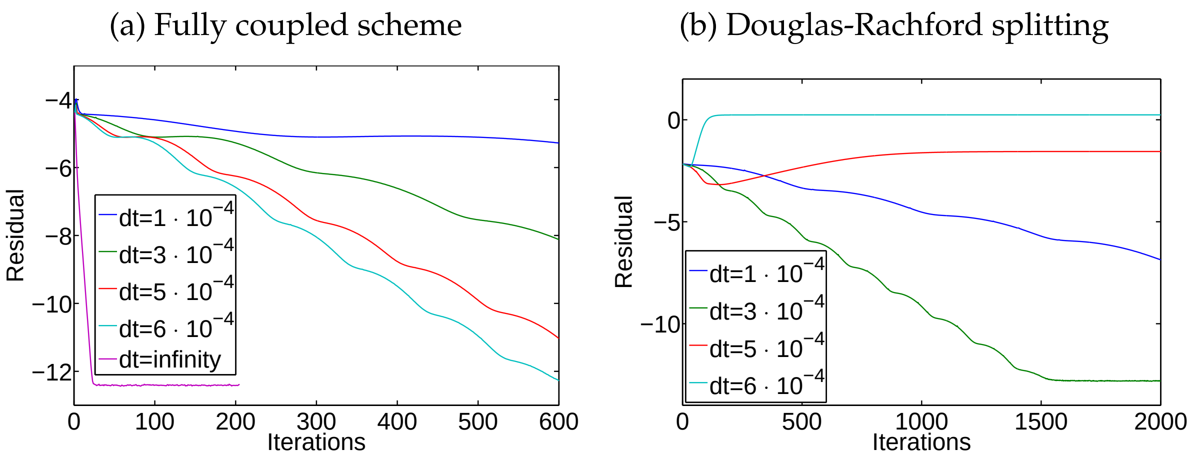


Figure 10.13: Nonlinear convergence history in logarithmic scale of the low-order schemes for different pseudo time steps on mesh level three

makes it a highly promising solution strategy for the two-fluid model.

11 Conclusions and Outlook

11.1 Conclusions

The topic of this thesis has been the development of (semi-) implicit and Newton-like finite element solvers for the compressible Euler equations and the two-fluid model governing compressible particle-laden gas flows. Particular emphasis has been laid on stationary solutions. The spatial discretization of hyperbolic systems is a challenging task since standard discretization techniques like the finite volume method or the Galerkin finite element method have very unfavorable properties. These methods violate important physical properties of the solution due to the large amount of built-in antidiffusion. A typical side effect is the birth and growth of numerical oscillations.

To suppress these oscillations, preserve the physical properties of the solution, and keep the scheme stable the amount of antidiffusion was constrained by a nodal TVD type flux limiter developed by Kuzmin et al. [44, 49]. The two-fluid model lacks hyperbolicity, which makes it impossible to control the amount of antidiffusion for a set of characteristic variables. This difficulty was circumvented by a separate stabilization of the gas and the particle equations. Due to their hyperbolicity, the equations governing the gas phase enable the limiter to be applied in a characteristic sense [47]. The particle equations only permit a single wave. Consequently, the use of scalar dissipation is in order. Since the scalar LED criterion is invalid for the nonlinear system, the minimal diffusion coefficient of scalar algebraic flux correction schemes [46] may produce undershoots and overshoots. It was replaced by a slightly larger one proposed by Banks and Shadid [3]. This revised definition was shown to be a suitable choice to compute a physically sensible Riemann solution and suppress numerical oscillations.

On the one hand, the employed limiter increases the order of approximation of the underlying low-order scheme in smooth parts of the flow and it is one of the most accurate limiter functions available at present. On the other hand, the limiter slows down convergence to steady-state due to oscillatory correction factors. In some publications [95] convergence is not achieved if a limiter is applied, unless the correction factors are frozen. In contrast to the present study, Trépanier et al. [93] observed that the rate of convergence at large CFL numbers deteriorates if the CFL number exceeds some upper bound. The deterioration of convergence rates in their study might be related to the strong nature of boundary conditions and the absence of a boundary Riemann solver. In the present study convergence and a decrease of the stationary residual by several orders of magnitude has always been achieved without freezing the oscillatory correction factors. Steady-state solutions were obtained by pseudo time marching combined with local time stepping or a Newton-like method. Remarkably, the rate of convergence did not deteriorate for arbitrarily large CFL numbers.

Since the stabilization term lacks smoothness, the original Jacobian was replaced by its low-order counterpart. This results in improved matrix properties and less demanding memory requirements. It was shown by the numerical results that convergence was achieved by the proposed scheme in a wide range of Mach numbers for both the Euler equations and the fully coupled two-fluid model. The author is not aware of any publication, where steady-state solutions to the two-fluid model are obtained in the sense that the nonlinear residual of the equations approaches zero.

Boundary conditions for the Euler equations and other hyperbolic systems are rarely discussed in the literature, particularly not in the context of implicit solvers and unstructured meshes. To the author's best knowledge there is no publication on unconditionally stable boundary conditions for the Euler equations discretized by finite elements. It was an important goal of the present thesis to develop a boundary treatment, which is unconditionally stable, allows fast convergence to steady-state at large CFL numbers, and is of finite element nature. Weak Neumann-type boundary conditions, which were proposed in this thesis, combined with a boundary Riemann solver were shown to be practically unconditionally stable and convergent for arbitrary CFL numbers. Moreover, they do not rely on finite volume features. As a matter of fact the boundary fluxes in the surface integral arising in the weak formulation are affected by the boundary conditions, rather than the volume integrals.

The potential of the developed boundary techniques is illustrated by the presented numerical study for the low-order scheme which is not influenced by the additional nonlinearity arising from the application of the flux limiter function. It was shown in each of the reported test cases that in the case of the low-order scheme the convergence rates improve with increasing CFL numbers. At the same time, the rate of convergence of the high-resolution scheme does not deteriorate appreciably if the CFL number exceeds some upper bound. This convergence behavior is in striking contrast to numerical studies in other publications [93]. In light of the above, the weak imposition of boundary conditions in the framework of a Newton-like scheme appears to be very promising. Numerical investigations have demonstrated that the new type of boundary treatment offers far superior robustness and convergence compared to its strong counterpart. Moreover, it was shown that the weak type of boundary conditions maintains accuracy and convergence of the spatial discretization, although it enforces boundary conditions in an integral rather than a pointwise sense.

Last but not least the developed techniques were tailored to the two-fluid equations. In spite of the interfacial two-way coupling, it was shown by the numerical results that unconditional stability and convergence can be achieved for arbitrary CFL numbers, provided that the source terms are incorporated in the solver by a fully coupled strategy and operator splitting is avoided. It can be inferred from a comparison with the single-phase computations that the two-fluid model solver features most properties of its single-phase counterpart. Moreover, the proposed fully coupled implicit time integration was compared to the usually employed operator splitting techniques. We have found that the new technique is far superior concerning robustness as well as efficiency since it is applicable to arbitrary CFL numbers. The lack of pressure in the particle equations requires a modification of the weak wall boundary conditions. A straightforward generalization of the concepts developed for the Euler equations turned out to be insufficient. A penalty term was introduced and added to the momentum equations of both phases to enforce the free-slip condition.

In summary, the essential novel contributions of this work are:

- Unconditionally stable finite element solver for the Euler equations and the two-fluid model.
- Unconditionally stable weak Neumann-type boundary conditions for both systems.
- The penalty term for reinforcing the weakly imposed no-penetration condition.
- The ghost state Riemann solver for the weak boundary treatment.
- Newton-like solver for the two-fluid equations.
- Inclusion of interfacial transfer terms without use of operator splitting.
- Success in computing steady-state solutions to the two-fluid model.

11.2 Outlook

Since we have developed an unconditionally stable nonlinear solver for compressible single-phase gas as well as particle-laden gas flows, a future goal of primary importance is to increase the efficiency and to accelerate convergence to steady-state. A better approximation of the original flux Jacobian may be derived to improve the preconditioner. The resulting Newton scheme combined with a matrix-free version of the Krylov subspace linear solvers could significantly reduce CPU time and memory requirements. On the other hand an approximation of the original Jacobian is difficult to derive since the stabilization term lacks smoothness. Moreover, the matrix properties of the original Jacobian are far less attractive than those of its low-order counterpart.

At the same time, the investigated fixed point iteration seems to be a suitable smoother in a multigrid procedure. It was pointed out by Hemker and Koren [31] that the dependence of convergence rates on the spatial length scales can almost be removed by nonlinear multigrid. Therefore, nonlinear multigrid, where the fixed point iteration scheme proposed in this thesis serves as smoother, is a highly promising way to accelerate convergence to steady-state. Moreover, a strongly coupled smoother of Vanka type can be employed as an alternative to the solution of very large linear algebraic systems arising in two-fluid models with source term coupling.

The developed methods are currently implemented in 2D, although they do not rely on 2D features. The whole algorithm is directly applicable to 3D. Real life applications, in particular in the field of thermal spraying processes, require 3D simulations. Hence, the 3D implementation will be another important topic in the future. Due to the significant increase of problem size in 3D simulations, a very efficient solution strategy, like the one based on nonlinear multigrid, is crucial for the extension to 3D.

Mesh adaptivity is another way to reduce the computational costs. In 3D applications it is particularly important to compute sufficiently accurate solutions at reasonable costs. The extension of the adaptivity techniques developed in [60] to the two-fluid model will therefore be addressed in future work.

This thesis was concerned with stationary solutions. Some physical phenomena like dust explosions or volcanic eruptions are time-dependent. For the simulation of such transient particle-laden gas flows, the FCT algorithm is better suited than TVD-like limiters. The development of implicit high-resolution FEM-FCT schemes for fully time-dependent two-phase flows is another challenging long term goal.

A Approximate Boundary Flux Jacobians

A.1 Euler Equations

Consider a boundary edge I with adjacent nodes k, l and state vectors U_k, U_l. The Jacobians of (7.9) and (7.10) are required. Let U_i be the interior state in the sense of section 7.1 at a quadrature point related to the edge (compare figure 7.4). In section 7.4 the quadrature points are $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$. The Jacobians with respect to U_k and U_l can be derived for each quadrature point separately, and the interior state corresponds to either $U_i = U(\hat{\mathbf{x}}_1)$ or $U_i = U(\hat{\mathbf{x}}_2)$. It is obtained by interpolation. The result depends on the location of the quadrature points as well as both states U_k and U_l. Note that linear interpolation on the boundary edges is exact for linear and bilinear finite elements. We focus our attention to the determination of the Jacobian with respect to U_k since the procedure is the same for U_l. For convenience one drops the index of the quadrature point and denotes the corresponding ghost state by U_∞. The most complicated part is the derivation of the flux $\mathbf{n} \cdot \mathbf{F}(U_\infty) = \mathbf{n} \cdot \tilde{\mathbf{F}}_{h\,\infty}$ related to the ghost state in formula (7.49). Recall that

$$U_\infty = U_\infty \left(W \left(U_i \left(U_k, U_l\right)\right)\right) \tag{A.1}$$

is defined by the Riemann invariants W, which are determined by the interior state U_i as well as the prescribed boundary condition. Hence, the chain rule yields

$$\begin{aligned} \mathbf{n} \cdot \frac{\partial \tilde{\mathbf{F}}_{h\,\infty}}{\partial U_k} &= \mathbf{n} \cdot \frac{\partial \tilde{\mathbf{F}}_{h\,\infty}}{\partial U_\infty} \frac{\partial U_\infty}{\partial U_k} \\ &= \mathbf{n} \cdot \mathbf{A}_\infty \frac{\partial U_\infty}{\partial U_k} \end{aligned} \tag{A.2}$$

in the normal direction, where $\mathbf{A}_\infty$ is the Jacobian tensor (1.15) evaluated at the ghost state. According to (A.1) the latter equation can be further decomposed into

$$\mathbf{n} \cdot \frac{\partial \tilde{\mathbf{F}}_{h\,\infty}}{\partial U_k} = \mathbf{n} \cdot \mathbf{A}_\infty \frac{\partial U_\infty}{\partial U_i} \frac{\partial U_i}{\partial U_k}. \tag{A.3}$$

The Jacobian of (7.49) with respect to the conservative variables is given by

$$\mathbf{n} \cdot \frac{\partial \tilde{\mathbf{F}}_h}{\partial U_k} = \mathbf{n} \cdot \frac{\partial \tilde{\mathbf{F}}_h}{\partial U_i} \frac{\partial U_i}{\partial U_k} \tag{A.4}$$

and

$$\mathbf{n} \cdot \frac{\partial \tilde{\mathbf{F}}_h}{\partial U_i} = \frac{1}{2} \left(\mathbf{n} \cdot \mathbf{A}_i + |A^n_{i\infty}|\right) + \frac{1}{2} \left(\mathbf{n} \cdot \mathbf{A}_\infty - |A^n_{i\infty}|\right) \frac{\partial U_\infty}{\partial U_i}, \tag{A.5}$$

where $|A^n_{i\infty}|$ is assumed to be independent of the conservative variables. This is consistent with the interior discretization. To evaluate formulas (A.4) and (A.5) it remains to specify $\frac{\partial U_\infty}{\partial U_i}$ and $\frac{\partial U_i}{\partial U_k}$. The latter matrix depends on the location of the quadrature points dictated by the quadrature rule. It can be derived easily. If both matrices are known, we take advantage of (A.4) to obtain the final Jacobian of the discretized boundary integrals (7.9) and (7.10) corresponding to the edge I. In the following sections $\frac{\partial U_\infty}{\partial U_i}$ will be determined.

A.1.1 Supersonic Inlet and Outlet Boundaries

The case of supersonic boundary conditions is the easiest one regarding Jacobian determination. At a supersonic inlet, all conservative variables are determined by the imposed boundary condition and they do not depend on the conservative variables. Hence,

$$\left.\frac{\partial U_\infty}{\partial U_i}\right|_{superin} = 0 \tag{A.6}$$

is a matrix of zero entries. Thus the boundary flux Jacobian (A.5) degenerates to

$$\mathbf{n} \cdot \left.\frac{\partial \tilde{\mathbf{F}}_h}{\partial U_i}\right|_{superin} = \frac{1}{2}\left(\mathbf{n} \cdot \mathbf{A}_i + |A^n_{i\infty}|\right). \tag{A.7}$$

Otherwise, if the boundary corresponds to a supersonic outlet, the ghost state is equal to the interior state and (A.5) simplifies to

$$\mathbf{n} \cdot \left.\frac{\partial \tilde{\mathbf{F}}_h}{\partial U_i}\right|_{superout} = \mathbf{n} \cdot \mathbf{A}_i. \tag{A.8}$$

A.1.2 Subsonic Inlet and Outlet Boundaries

In this section the derivation of U_∞ with respect to the interior state U_i is discussed. It can be substituted into (A.5) and completes the boundary flux Jacobian. We focus our attention on subsonic free stream boundary conditions since the derived Jacobians are also used to construct the preconditioner at other types of subsonic boundaries. A multiple application of the chain rule to equation (A.1) leads to the formula

$$\frac{\partial U_\infty}{\partial U_i} = \frac{\partial U_\infty}{\partial W}\frac{\partial W}{\partial U_i}\frac{\partial U_i}{\partial U_k}. \tag{A.9}$$

The following sections are concerned with the determination of the derivatives $\frac{\partial U_\infty}{\partial W}$ and $\frac{\partial W}{\partial U_i}$.

A.1.2.1 Free Stream Inlet Ghost State Jacobian

Subsonic inlets and outlets involve the most complicated types of boundary conditions since the ghost state is determined by the Riemann invariants, which also depend on the conservative variables. At a subsonic inlet the first three eigenvalues are negative, while the last is positive. To prescribe characteristic boundary conditions, the first three Riemann invariants are overwritten by their imposed free stream counterparts (compare section 7.5.5). The case of a pressure-density inlet as described in section 7.5.7 is quite similar, although it requires some additional effort. On the other hand, the boundary flux Jacobian is merely needed for the preconditioner of the nonlinear iteration and does not affect the residual. It is therefore acceptable to neglect the latter dependency and use the Jacobian described in the present section.

The ghost state is uniquely defined by

$$\begin{aligned} \rho_\infty &= \left(\frac{c_\infty^2}{\gamma W_{\infty,2}}\right)^{\frac{1}{\gamma-1}} \\ \mathbf{v}_\infty &= \frac{1}{2}\mathbf{n}(W_{\infty,1}+W_4^*)+\boldsymbol{\tau}\,W_{\infty,3} \\ P_\infty &= \frac{\rho_\infty c_\infty^2}{\gamma} \\ c_\infty &= \frac{\gamma-1}{4}\left(W_4^*-W_{\infty,1}\right), \end{aligned} \tag{A.10}$$

where $\mathbf{v}_\infty = (u_\infty, v_\infty)^T$ and the Riemann invariants are given by (7.14). It remains to determine the derivatives of the boundary state U_∞, which is subject to (A.10). One easily verifies

$$\left.\frac{\partial U_\infty^{(1)}}{\partial U_i^{(j)}}\right|_{subin} = \frac{\partial \rho_\infty}{\partial U_i^{(j)}} = \frac{\partial \rho_\infty}{\partial c_\infty}\frac{\partial c_\infty}{\partial W_4^*}\frac{\partial W_4^*}{\partial U_i^{(j)}} = \frac{\rho_\infty}{2c_\infty}\frac{\partial W_4^*}{\partial U_i^{(j)}} \tag{A.11}$$

by an application of the chain rule, where $j \in \{1,\ldots,4\}$ denotes the j-th component of U_i. By similar arguments the corresponding derivatives of the ghost state with respect to the conservative variables

$$\left.\frac{\partial U_\infty^{(2)}}{\partial U_i^{(j)}}\right|_{subin} = u_\infty\frac{\partial \rho_\infty}{\partial U_i^{(j)}} + \frac{1}{2}n^{(x)}\rho_\infty\frac{\partial W_4^*}{\partial U_i^{(j)}} \tag{A.12}$$

$$\left.\frac{\partial U_\infty^{(3)}}{\partial U_i^{(j)}}\right|_{subin} = v_\infty\frac{\partial \rho_\infty}{\partial U_i^{(j)}} + \frac{1}{2}n^{(y)}\rho_\infty\frac{\partial W_4^*}{\partial U_i^{(j)}} \tag{A.13}$$

$$\left.\frac{\partial U_\infty^{(4)}}{\partial U_i^{(j)}}\right|_{subin} = \frac{1}{\gamma-1}\frac{\partial P_\infty}{\partial U_i^{(j)}} + \frac{|\mathbf{v}_\infty|^2}{2}\frac{\partial \rho_\infty}{\partial U_i^{(j)}} + \frac{1}{4}\rho_\infty(W_{\infty,1}+W_4^*)\frac{\partial W_4^*}{\partial U_i^{(j)}} \tag{A.14}$$

can be determined, where

$$\frac{\partial P_\infty}{\partial U_i^{(j)}} = \frac{\gamma-1}{2}\rho_\infty c_\infty \frac{\partial W_4^*}{\partial U_i^{(j)}} + \frac{c_\infty^2}{\gamma}\frac{\partial \rho_\infty}{\partial U_i^{(j)}}. \tag{A.15}$$

Last but not least, the derivatives of the fourth Riemann invariant are required to compute the desired Jacobian. They are given by

$$\begin{aligned}
\frac{\partial W_4^*}{\partial U_i^{(1)}} &= -\frac{1}{\rho_i}\mathbf{v}_i\cdot\mathbf{n} + \frac{\gamma}{c_i}\left(-\frac{E_i}{\rho_i} + \frac{|\mathbf{v}_i|^2}{\rho_i}\right)\\
\frac{\partial W_4^*}{\partial U_i^{(2)}} &= \frac{n^{(x)}}{\rho_i} - \frac{\gamma}{c_i}\frac{u_i}{\rho_i}\\
\frac{\partial W_4^*}{\partial U_i^{(3)}} &= \frac{n^{(y)}}{\rho_i} - \frac{\gamma}{c_i}\frac{v_i}{\rho_i}\\
\frac{\partial W_4^*}{\partial U_i^{(4)}} &= \frac{\gamma}{c_i}\frac{1}{\rho_i}
\end{aligned} \tag{A.16}$$

with $\mathbf{v}_i = (u_i, v_i)^T$.

A.1.2.2 Free Stream Outlet Ghost State Jacobian

The subsonic outlet ghost state Jacobian can be derived in a very similar way. The differences are due to the different characteristic directions. In the case of a subsonic outlet the last three eigenvalues are positive and only the first has negative sign. Hence, the last three Riemann invariants of (7.14) are taken from the interior, while the first one is overwritten by the imposed boundary condition. In this section the first Riemann invariant is assumed to be independent of the conservative variables. This assumption is satisfied for free stream boundary conditions, while it does not hold in the case of a pressure outlet (compare to section 7.5.6) since the first Riemann invariant depends on its second and fourth counterparts. For the sake of simplicity, this dependency is neglected in this study.

With respect to the characteristic direction the ghost state is given in terms of

$$\begin{aligned}
\rho_\infty &= \left(\frac{c_\infty^2}{\gamma W_2^*}\right)^{\frac{1}{\gamma-1}}\\
\mathbf{v}_\infty &= \frac{1}{2}\mathbf{n}(W_{\infty,1} + W_4^*) + \boldsymbol{\tau}\, W_3^*\\
P_\infty &= \frac{\rho_\infty c_\infty^2}{\gamma}\\
c_\infty &= \frac{\gamma-1}{4}\left(W_4^* - W_{\infty,1}\right).
\end{aligned} \tag{A.17}$$

Corresponding to (A.17) one determines the derivatives of the ghost state with respect to the conservative variables by a multiple application of the chain rule similar to the section above. Let once again i denote the index of a boundary node and j its components. Due to these abbreviations the first row of the Jacobian reads

$$\left.\frac{\partial U_{\infty}^{(1)}}{\partial U_i^{(j)}}\right|_{subout} = \frac{\partial \rho_\infty}{\partial U_i^{(j)}} = \frac{\partial \rho_\infty}{\partial c_\infty}\frac{\partial c_\infty}{\partial W_4^*}\frac{\partial W_4^*}{\partial U_i^{(j)}} + \frac{\partial \rho_\infty}{\partial W_2^*}\frac{\partial W_2^*}{\partial U_i^{(j)}} \tag{A.18}$$

and can be simplified to

$$\left.\frac{\partial U_{\infty}^{(1)}}{\partial U_i^{(j)}}\right|_{subout} = \frac{\rho_\infty}{2c_\infty}\frac{\partial W_4^*}{\partial U_i^{(j)}} + \frac{\rho_\infty}{(1-\gamma)W_2^*}\frac{\partial W_2^*}{\partial U_i^{(j)}}. \tag{A.19}$$

The second and third rows involve derivatives of the velocity associated with the ghost state

$$\frac{\partial u_\infty}{\partial U_i^{(j)}} = \frac{1}{2}n^{(x)}\frac{\partial W_4^*}{\partial U_i^{(j)}} - n^{(y)}\frac{\partial W_3^*}{\partial U_i^{(j)}} \tag{A.20}$$

$$\frac{\partial v_\infty}{\partial U_i^{(j)}} = \frac{1}{2}n^{(y)}\frac{\partial W_4^*}{\partial U_i^{(j)}} + n^{(x)}\frac{\partial W_3^*}{\partial U_i^{(j)}}, \tag{A.21}$$

where the tangent is given by (7.5). The remaining three rows of the desired Jacobian are

$$\left.\frac{\partial U_{\infty}^{(2)}}{\partial U_i^{(j)}}\right|_{subout} = \rho_\infty\frac{\partial u_\infty}{\partial U_i^{(j)}} + u_\infty\frac{\partial \rho_\infty}{\partial U_i^{(j)}} \tag{A.22}$$

$$\left.\frac{\partial U_{\infty}^{(3)}}{\partial U_i^{(j)}}\right|_{subout} = \rho_\infty\frac{\partial v_\infty}{\partial U_i^{(j)}} + v_\infty\frac{\partial \rho_\infty}{\partial U_i^{(j)}} \tag{A.23}$$

$$\left.\frac{\partial U_{\infty}^{(4)}}{\partial U_i^{(j)}}\right|_{subout} = \frac{1}{\gamma-1}\frac{\partial P_\infty}{\partial U_i^{(j)}} + \frac{\rho_\infty}{2}\frac{\partial |\mathbf{v}_\infty|^2}{\partial U_i^{(j)}} + \frac{|\mathbf{v}_\infty|^2}{2}\frac{\partial \rho_\infty}{\partial U_i^{(j)}}, \tag{A.24}$$

where

$$\frac{\partial |\mathbf{v}_\infty|^2}{\partial U_i^{(j)}} = 2W_3^*\frac{\partial W_3^*}{\partial U_i^{(j)}} + \frac{1}{2}\left(W_{1,\infty} + W_4^*\right)\frac{\partial W_4^*}{\partial U_i^{(j)}} \tag{A.25}$$

and

$$\frac{\partial P_\infty}{\partial U_i^{(j)}} = \frac{\gamma-1}{2\gamma}\rho_\infty c_\infty\frac{\partial W_4^*}{\partial U_i^{(j)}} + \frac{1}{\gamma}c_\infty^2\frac{\partial \rho_\infty}{\partial U_i^{(j)}}. \tag{A.26}$$

Note that the derivatives of the fourth Riemann invariant are adopted from (A.16), while the corresponding derivatives related to the second and third Riemann invariant are given by

$$\begin{aligned}
\frac{\partial W_2^*}{\partial U_i^{(1)}} &= \frac{\gamma-1}{2\rho_i^\gamma}|\mathbf{v}_i|^2 - \frac{\gamma}{\rho_i}W_2^* & \frac{\partial W_3^*}{\partial U_i^{(1)}} &= \frac{1}{\rho_i}\left(u_i n^{(y)} - v_i n^{(x)}\right) \\
\frac{\partial W_2^*}{\partial U_i^{(2)}} &= \frac{1-\gamma}{\rho_i^\gamma}u_i & \frac{\partial W_3^*}{\partial U_i^{(2)}} &= -\frac{n^{(y)}}{\rho_i} \\
\frac{\partial W_2^*}{\partial U_i^{(3)}} &= \frac{1-\gamma}{\rho_i^\gamma}v_i & \frac{\partial W_3^*}{\partial U_i^{(3)}} &= \frac{n^{(x)}}{\rho_i} \\
\frac{\partial W_2^*}{\partial U_i^{(4)}} &= \frac{\gamma-1}{\rho_i^\gamma} & \frac{\partial W_3^*}{\partial U_i^{(4)}} &= 0.
\end{aligned} \tag{A.27}$$

A.1.3 Solid Walls

In the single-phase gas flow simulations the mirror condition or the zero-flux condition (7.42) is prescribed. To apply the mirror condition a boundary Riemann problem is solved either by the approximate Riemann solver of Roe or the exact one of Toro [92]. Since it is difficult to compute the original Jacobian of Toro's solver, it is replaced by the Jacobian of the flux formula of Roe. The ghost state of a node located on a solid wall is not determined by the Riemann invariants. We prescribe the mirror condition by reflecting the normal velocity on the tangent and the ghost state of a node located at a solid wall boundary is determined by the reflection rather than Riemann invariants. This significantly simplifies the process of Jacobian derivation. Hence, $\frac{\partial U_\infty}{\partial U_i}$ is given by

$$\left.\frac{\partial U_\infty}{\partial U_i}\right|_{wall} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & {n^{(y)}}^2 - {n^{(x)}}^2 & -2n^{(x)}n^{(y)} & 0 \\ 0 & -2n^{(x)}n^{(y)} & {n^{(x)}}^2 - {n^{(y)}}^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{A.28}$$

In the case of the zero-flux condition (7.42), the flux Jacobian simplifies to

$$\mathbf{n}\cdot\left.\frac{\partial \tilde{\mathbf{F}}_h}{\partial U_i}\right|_{wall} = (\gamma-1)\begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{|\mathbf{v}_i|^2}{2}n^{(x)} & -u_i n^{(x)} & -v_i n^{(x)} & n^{(x)} \\ \frac{|\mathbf{v}_i|^2}{2}n^{(y)} & -u_i n^{(y)} & -v_i n^{(y)} & n^{(y)} \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{A.29}$$

A.2 Two-Fluid Model

The two-fluid model of interest consists of conservation laws for the gas as well as the particulate phase and the fluxes of both phases in conservative variables are independent of each other. Hence, the boundary flux Jacobians of both phases can be derived individually. The flux Jacobians of the gas phase, except at solid walls, are the same as reported in section A.1 since the equations governing the gas phase of the two-fluid model are exactly the Euler equations for the effective density $\rho = \alpha_g \rho_g$. The solid wall boundary condition for the two-fluid model is implemented by the penalty term and elimination of fluxes in normal direction from the boundary integrals, see section 7.6. Hence, it remains to specify the boundary flux Jacobians associated with the particulate phase and the solid wall boundary flux Jacobians. The boundary flux Jacobians of the particulate phase take the form (A.5) as in the case of the gas phase and the final Jacobians of the boundary integral can be computed analogously.

A.2.1 Particulate Phase Inlet and Outlet Boundary Flux Jacobians

Let us at first consider inlet boundaries. At an inlet all conservative variables of the particulate phase have to be imposed corresponding to section 7.6.1. Hence, the ghost state depends on the imposed boundary condition rather than the conservative variables of the interior state. Let us again assume that $|A^n_{i\infty}|$ in (7.55) is independent of the conservative variables. Then the inlet flux Jacobian of the particulate phase with respect to the interior state corresponding to (A.5) simplifies to

$$\mathbf{n} \cdot \frac{\partial \tilde{\mathbf{F}}_{h,p}}{\partial U_{p,i}}\bigg|_{in} = \frac{1}{2}\left(\mathbf{n} \cdot \mathbf{A}_i + |A^n_{i\infty}|\right). \tag{A.30}$$

The Jacobian tensor, defined in equation (4.7) and evaluated at the interior state, is denoted by $\mathbf{A}_i$. On the other hand at an outlet, all conservative variables are taken from the interior state and consequently the outlet boundary flux Jacobian is

$$\mathbf{n} \cdot \frac{\partial \tilde{\mathbf{F}}_{h,p}}{\partial U_{p,i}}\bigg|_{out} = \mathbf{n} \cdot \mathbf{A}_i. \tag{A.31}$$

A.2.2 Wall Boundary Flux Jacobian

The solid wall boundary conditions are imposed by the addition of the penalty term as well as the elimination of normal fluxes from the boundary integrals. For the gas phase the boundary flux simplifies to

$$\mathbf{n} \cdot \tilde{\mathbf{F}}_{h,g}\bigg|_{wall} = \begin{bmatrix} 0 \\ n^{(x)}P \\ n^{(y)}P \\ 0 \end{bmatrix}, \tag{A.32}$$

while it vanishes for the particulate phase due to the lack of pressure

$$\mathbf{n} \cdot \tilde{\mathbf{F}}_{h,p}\bigg|_{wall} = 0. \tag{A.33}$$

Hence, the solid wall boundary flux Jacobian associated with the gas phase is (A.29) and its analogous counterpart of the particulate phase is just a block of zero

$$\mathbf{n} \cdot \frac{\partial \tilde{\mathbf{F}}_{h,p}}{\partial U_{p,i}}\bigg|_{wall} = 0. \tag{A.34}$$

Last but not least, the Jacobian of the penalty term (7.60) is needed. Due to the large penalty parameter σ the penalty term is rather large and stiff, which rules out ad-hoc approximations of the Jacobian. To circumvent the problem of non-differentiability one substitutes the absolute value by a smooth approximation [59]

$$softabs(x) = \sqrt{x^2 + \varepsilon} \tag{A.35}$$

with the very small constant $\varepsilon > 0$. Note that this parameter does not affect the convergence behavior significantly and does not call for any tuning in contrast to the parameters associated with finite difference approximations like (8.20). It is just a small perturbation of the absolute value. All the computations reported later were performed with the same value of $\varepsilon = 10^{-8}$. The smooth approximation of the absolute value enables us to derive the flux Jacobian of the penalty term

$$\mathbf{n} \cdot \frac{\partial \tilde{\mathbf{F}}_{h}}{\partial U}\bigg|_{penalty} = -\sigma \frac{2v_n^2 + \varepsilon}{\sqrt{v_n^2 + \varepsilon}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & {n^{(x)}}^2 & n^{(x)}n^{(y)} & 0 \\ 0 & n^{(x)}n^{(y)} & {n^{(y)}}^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{A.36}$$

The Jacobian is the same for both phases and the indices distinguishing both phases are dropped for that reason. Note that the sign of the penalty term (7.60) is chosen to restrict the contribution to the preconditioner positive semi-definite so as to increase its matrix properties [13]. Indeed the eigenvalues of (A.36) are

$$\lambda_{1,2,3} = 0, \qquad \text{and} \qquad \lambda_4 = -\sigma \frac{2v_n^2 + \varepsilon}{\sqrt{v_n^2 + \varepsilon}} < 0 \tag{A.37}$$

so that the contribution to the preconditioners (8.11) and (8.12) is positive semi-definite.

B Approximate Drag Force and Heat Exchange Jacobians

B.1 Drag Force Jacobian

The application of the drag force (2.64) can be written in vectorial form

$$\begin{aligned}
\tilde{\mathbf{F}}_D = \begin{bmatrix} 0 \\ \mathbf{F}_D \\ \mathbf{v}_p \cdot \mathbf{F}_D \end{bmatrix} &= Q_{drag} U_g^{(1)} U_p^{(1)} \begin{bmatrix} 0 \\ \frac{U_g^{(2)}}{U_g^{(1)}} - \frac{U_p^{(2)}}{U_p^{(1)}} \\ \frac{U_g^{(3)}}{U_g^{(1)}} - \frac{U_p^{(3)}}{U_p^{(1)}} \\ \frac{1}{U_p^{(1)}} \begin{bmatrix} U_p^{(2)} \\ U_p^{(3)} \end{bmatrix} \cdot \begin{bmatrix} \frac{U_g^{(2)}}{U_g^{(1)}} - \frac{U_p^{(2)}}{U_p^{(1)}} \\ \frac{U_g^{(3)}}{U_g^{(1)}} - \frac{U_p^{(3)}}{U_p^{(1)}} \end{bmatrix} \end{bmatrix} \\
&= Q_{drag} \begin{bmatrix} 0 \\ U_g^{(2)} U_p^{(1)} - U_p^{(2)} U_g^{(1)} \\ U_g^{(3)} U_p^{(1)} - U_p^{(3)} U_g^{(1)} \\ U_g^{(2)} U_p^{(2)} + U_g^{(3)} U_p^{(3)} - \frac{U_g^{(1)}}{U_p^{(1)}} \left({U_p^{(2)}}^2 + {U_p^{(3)}}^2 \right) \end{bmatrix},
\end{aligned} \tag{B.1}$$

where Q_{drag} is an abbreviation of

$$Q_{drag} = \frac{3}{4 \rho_p d \alpha_g} |\mathbf{v}_p - \mathbf{v}_g| C_D. \tag{B.2}$$

In the definition above, Q_{drag} depends on the drag coefficient C_D and consists of the non-smooth part of $\tilde{\mathbf{F}}_D$. Consequently, Q_{drag} is assumed to be constant with respect to the conservative variables, instead of replacing the non-differentiable functions by smooth approximations. The advantages of this approach are twofold. First, it yields a more robust scheme compared to the former mentioned opportunity. Second, it is possible to define a user-defined drag curve very easily.

Due to the assumptions above, the approximate Jacobians of the drag force with respect to the conservative variables of both the gas and the particulate phase are given by

$$\frac{\partial \tilde{\mathbf{F}}_D}{\partial U_g} = Q_{drag} \begin{bmatrix} 0 & 0 & 0 & 0 \\ -U_p^{(2)} & U_p^{(1)} & 0 & 0 \\ -U_p^{(3)} & 0 & U_p^{(1)} & 0 \\ -\frac{U_p^{(2)^2}+U_p^{(3)^2}}{U_p^{(1)}} & U_p^{(2)} & U_p^{(3)} & 0 \end{bmatrix} \tag{B.3}$$

and

$$\frac{\partial \tilde{\mathbf{F}}_D}{\partial U_p} = Q_{drag} \begin{bmatrix} 0 & 0 & 0 & 0 \\ U_g^{(2)} & -U_g^{(1)} & 0 & 0 \\ U_g^{(3)} & 0 & -U_g^{(1)} & 0 \\ \frac{U_g^{(1)}\left(U_p^{(2)^2}+U_p^{(3)^2}\right)}{U_p^{(1)^2}} & \frac{U_g^{(2)}U_p^{(1)}-2U_g^{(1)}U_p^{(2)}}{U_p^{(1)}} & \frac{U_g^{(3)}U_p^{(1)}-2U_g^{(1)}U_p^{(3)}}{U_p^{(1)}} & 0 \end{bmatrix}. \tag{B.4}$$

B.2 Temperature Exchange Jacobian

An approximate Jacobian of the interfacial temperature exchange can be derived in a similar way. The interfacial heat exchange is modeled by (2.65)

$$Q_T = \frac{Nu 6\kappa}{d^2}\alpha_p(T_g - T_p). \tag{B.5}$$

One can rewrite it in a more compact form

$$Q_T = Q_{tem} U_p^{(1)}(T_g - T_p), \tag{B.6}$$

where Q_{tem} is defined by

$$Q_{tem} = \frac{Nu 6\kappa}{\rho_p d^2}. \tag{B.7}$$

Like the approximate derivation of the drag force contributions, Q_{tem} is assumed to be independent of the conservative variables. This eliminates the need for differentiating the non-smooth Nusselt number Nu. In addition, it is possible to specify a user-defined Nusselt number in an external subroutine without changing the source code, which makes it more flexible. Taking the latter assumption into consideration the approximate Jacobians with respect to the conservative variables are

$$\frac{\partial Q_T}{\partial U_g} = Q_{tem}\frac{U_p^{(1)}}{c_{vg}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{U_g^{(4)}}{U_g^{(1)^2}} + \frac{U_g^{(2)^2}+U_g^{(3)^2}}{U_g^{(1)^3}} & -\frac{U_g^{(2)}}{U_g^{(1)^2}} & -\frac{U_g^{(3)}}{U_g^{(1)^2}} & \frac{1}{U_g^{(1)}} \end{bmatrix} \tag{B.8}$$

and

$$\frac{\partial Q_T}{\partial U_p} = Q_{tem} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ T_g - T_p + \frac{U_p^{(4)}}{U_p^{(1)}} - \frac{1}{c_{vp}} \left(\frac{{U_p^{(2)}}^2 + {U_p^{(3)}}^2}{{U_p^{(1)}}^2} \right) & \frac{U_p^{(2)}}{c_{vp} U_p^{(1)}} & \frac{U_p^{(3)}}{c_{vp} U_p^{(1)}} & -\frac{1}{c_{vp}} \end{bmatrix}. \tag{B.9}$$

Hence the approximate source term Jacobian (8.34) of the two-fluid model is defined in terms of

$$\frac{\partial S_p}{\partial U_g} = M_L \ diag\{\frac{\partial \tilde{\mathbf{F}}_{D\,1}}{\partial U_{g\,1}} + \frac{\partial Q_{T\,1}}{\partial U_{g\,1}}, \ldots, \frac{\partial \tilde{\mathbf{F}}_{D\,NVT}}{\partial U_{g\,NVT}} + \frac{\partial Q_{T\,NVT}}{\partial U_{g\,NVT}}\} = -\frac{\partial S_g}{\partial U_g} \tag{B.10}$$

and

$$\frac{\partial S_p}{\partial U_p} = M_L \ diag\{\frac{\partial \tilde{\mathbf{F}}_{D\,1}}{\partial U_{p\,1}} + \frac{\partial Q_{T\,1}}{\partial U_{p\,1}}, \ldots, \frac{\partial \tilde{\mathbf{F}}_{D\,NVT}}{\partial U_{p\,NVT}} + \frac{\partial Q_{T\,NVT}}{\partial U_{p\,NVT}}\} = -\frac{\partial S_g}{\partial U_p} \tag{B.11}$$

corresponding to (8.35).

Bibliography

[1] Jr. J. D. Anderson, J. C. Tannenhill, R. H. Plecher: *Modern Compressible Flow with Historical Perspective*, Series Aeronautical and Aerospace Engineering, McGraw-Hill Publishing Company, 1990.

[2] N. Balakrishnan, G. Fernandez: *Wall boundary conditions for inviscid compressible flows on unstructured meshes*, Int. J. Num. Meth. in Fluids, Vol. 28, 1998, pp. 1481-1501.

[3] J. W. Banks, W. D. Henshaw, J. N. Shadid: *An evaluation of the FCT method for high-speed flows on structured overlapping grids*, J. Comp. Phys., Vol. 228 (15), 2009, pp. 5349-5369.

[4] T. J. Barth: *Analysis of implicit local linearization techniques for upwind and TVD algorithms*, AIAA Paper, 1987, Article No. 87-0595.

[5] Y. Bazilevs, T. J. R. Hughes: *Weak imposition of Dirichlet boundary conditions in fluid mechanics*, Comp. and Fluids, Vol. 36, 2007, pp. 12-26.

[6] A. Bermúdez, M. E. Vázquez: *Upwind methods for hyperbolic conservation laws with source terms*, Comp. and Fluids, Vol. 23 (8), 1994, pp. 1049-1071.

[7] M. Blanco, D. W. Zingg: *A fast solver for the Euler equations on unstructured grids using a Newton-GMRES method*, AIAA Paper, 1997, Article No. 97-0331.

[8] O. Borchers, C. Busch, A. Sokolichin, G. Eigenberger: *Applicability of the standard k-ϵ turbulence model to the dynamic simulation of bubble columns. Part II: Comparison of detailed experiments and flow simulations*, Chem. Eng. Sci., Vol. 54, 1999, pp. 5927-5935.

[9] J. P. Boris, D. L. Book: *Flux-corrected transport. I. SHASTA, A fluid transport algorithm that works*, J. Comp. Phys., Vol. 11, 1973, pp. 38-69.

[10] I. Chang: *One- and two-phase nozzle flows*, AIAA J., Vol. 18 (12), 1980, pp. 1445-1461, Article No. 80-0272R.

[11] C. T. Crowe, M. Sommerfeld, Y. Tsuji: *Multiphase Flows with Droplets and Particles*, CRC Press LLC, 1998.

[12] J. M. Delhaye, J. A. Bouré: *General equations and two-phase flow modeling*, In *Handbook of Multiphase Systems*, G. Hestroni, editior, Vol. 1, 1982, pp. 36-95.

[13] L. Demkowicz, J. T. Oden, W. Rachowicz, O. Hardy: *An $h-p$ Taylor-Galerkin finite element method for compressible Euler equations*, Comp. Meth. Appl. Mech. Eng., Vol. 88, 1991, pp. 363-396.

[14] J. Douglas, H. H. Rachford: *On the numerical solution of heat conduction problems in two or three space variables*, Trans. Amer. Math. Soc., Vol. 82, 1956, pp. 421-439.

[15] V. Dolejší, M. Feistauer: *A semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow*, J. Comp. Phys., Vol. 198, 2004, pp. 727-746.

[16] D. A. Drew, S. L. Passmann: *Theory of Multicomponent Fluids*, Springer, 1999.

[17] M. Feistauer, J. Felcman, I. Straškraba: *Mathematical and computational methods for compressible flow*, Clarendon Press, Oxford, 2003.

[18] M. Feistauer, V. Kučera: *On a robust discontinuous Galerkin technique for the solution of compressible flow*, J. Comp. Phys., Vol. 224, 2007, pp. 208-231.

[19] J. H. Ferziger, M. Perić: *Computational Methods for Fluid Dynamics*, Springer, 1996.

[20] C. A. J. Fletcher: *The group finite element formulation*, Comp. Meth. Appl. Mech. Eng., Vol. 37, 1983, pp. 225-243.

[21] J.-M. Ghidaglia, F. Pascal: *On boundary conditions for multidimensional hyperbolic systems of conservation laws in the finite volume framework,* Technical Report, ENS de Cachan, 2002.

[22] E. Godlewski, P. A. Raviart: *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, Springer Verlag, 1995.

[23] S. K. Godunov: *A difference scheme for numerical computation of discontinuous solutions of hydrodynamic equations*, (Russian), *Math. Sbornik*, Vol. 47, 1959, pp. 271-306.

[24] J. B. Goodman, R. J. LeVeque: *On the accuracy of stable schemes for 3D scalar conservation laws*, Math. Comp., Vol. 45, 1985, pp. 15-21.

[25] M. Gurris, D. Kuzmin, S. Turek: *Finite element simulation of particle-laden gas flows*, J. Comp. Appl. Math., 2009, in press, doi:10.1016/j.cam.2009.07.041.

[26] A. Harten: *High resolution schemes for hyperbolic conservation laws*, J. Comp. Phys., Vol. 49, 1983, pp. 357-393.

[27] A. Harten: *On a class of high resolution total-variation-stable finite difference schemes*, SIAM J. Numer. Anal., Vol. 21 (1), 1984, pp. 1-23.

[28] R. Hartmann: *http://www.numerik.uni-hd.de/ hartmann/*, Homepage.

[29] R. Hartmann, P. Houston: *Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations*, J. Comp. Phys., Vol. 183 (2), 2002, pp. 508-532.

[30] P. W. Hemker, B. Koren, W. M. Lioen, M. Nool, H. T. M. van der Maarel: *Multigrid for steady gas dynamics problems*, In M. Hafez and K. Oshima, editors, Computational Fluid Dynamics Review, 1995, pp. 477-494, Chichester, Wiley.

[31] P. W. Hemker, B. Koren: *Defect correction and nonlinear multigrid for steady Euler equations*, In: W. G. Habashi, M. M. Hafez (eds), Computational Fluid Dynamics Techniques, London: Gordon and Breach Publishers, 1995, pp. 699-718.

[32] C. Hirsch: *Numerical Computation of Internal and External Flows: Computational Methods for Inviscid and Viscous Flows*, Vol. II, John Wiley and Sons, New York, 1990.

[33] O. Igra, G. Hu, J. Falcovitz, B. Y. Wang: *Shock wave reflection from a wedge in a dusty gas*, Int. J. Multiphase Flow, Vol. 30, 2004, pp. 1139-1169.

[34] R. Ishii, Y. Umeda: *Nozzle flows of gas-particle mixtures*, Phys. Fluids, Vol. 30 (3), 1987, pp. 752-760.

[35] M. Ishii, T. Hibiki: *Thermo-Fluid Dynamics of Two-Phase Flow*, Springer, 2006.

[36] A. Jameson: *Analysis and design of numerical schemes for gas dynamics I. Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence*, Int. J. of Comp. Fluid Dyn., Vol. 4, 1995, pp. 171-218.

[37] V. John, A. Liakos: *Time-dependent flow across a step: the slip with friction boundary condition*, Int. J. Num. Meth. Fluids, Vol. 50, 2006, pp. 713-731.

[38] V. John, E. Schmeyer: *Finite element methods for time-dependent convection-diffusion-reaction equations with small diffusion*, Comput. Meth. Appl. Mech. Eng., Vol 198, 2008, pp. 475-494.

[39] S.-W. Kim, K.-S. Chang: *Reflection of shock wave from a compression corner in a particle-laden gas*, Shock Waves, Vol. 1, 1991, pp. 65-73.

[40] J. G. Knudsen, D. L. Katz: *Fluid Mechanics and Heat Transfer*, McGraw-Hill, New York, 1958.

[41] L. Krivodonova, M. Berger: *High-order accurate implementation of solid wall boundary conditions in curved geometries*, J. Comp. Phys., Vol. 211, 2006, pp. 492-512.

[42] D. Kuzmin: *Numerical Simulation of Reactive Bubbly Flows*, PhD Thesis, University of Jyväskylä, 1999.

[43] D. Kuzmin: *On the design of general purpose flux limiters for finite element schemes. I. Scalar convection*, J. Comp. Phys., Vol. 219, 2006, pp. 513-531.

[44] D. Kuzmin: *Algebraic flux correction for finite element discretizations of coupled systems*, In E. Onate, M. Papadrakakis, B. Schrefler, editors, Comp. Meth. for Coupled Prob. in Sc. and. Eng. II, CIMNE, 2007, pp. 653-656.

[45] D. Kuzmin: *High-Resolution Schemes for Numerical Simulation of Transport Phenomena*, in preparation.

[46] D. Kuzmin, M. Möller: *Algebraic flux correction I. Scalar conservation laws*, In D. Kuzmin, R. Löhner, S. Turek, editors, Flux-Corrected Transport, Principles, Algorithms, and Applications, pp. 155-206, Springer, 2005.

[47] D. Kuzmin, M. Möller: *Algebraic flux correction II. Compressible Euler equations*, In D. Kuzmin, R. Löhner, S. Turek, editors, Flux-Corrected Transport, Principles, Algorithms, and Applications, pp. 207-250, Springer, 2005.

[48] D. Kuzmin, S. Turek: *Flux correction tools for finite elements*, J. Comp. Phys., Vol. 175 (2), 2002, pp. 525-558.

[49] D. Kuzmin, S. Turek: *High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter*, J. Comp. Phys., Vol. 198 (1), 2004, pp. 131-158.

[50] P. D. Lax: *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, SIAM Publications, Philadelphia, 1973.

[51] R. J. LeVeque: *Numerical Methods for Conservation Laws*, Lectures in mathematics, Birkhäuser, 1992.

[52] R. J. LeVeque: *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.

[53] R. Löhner: *Applied CFD Techniques: An Introduction Based on Finite Element Methods*, Wiley, 2001.

[54] R. Löhner, K. Morgan, J. Peraire, M. Vahdati: *Finite element flux corrected transport (FEM-FCT) for the compressible Euler and Navier-Stokes equations*, Int. J. Numer. Meth. Fluids, Vol. 7, 1987, pp. 1093-1109.

[55] R. Löhner, K. Morgan, M. Vahdati, J. P. Boris, D. L. Book: *FEM-FCT: combining unstructured grids with high resolution*, Commun. Appl. Numer. Meth., Vol. 4, 1988, pp. 717-729.

[56] J. F. Lynn: *Multigrid Solution of the Euler Equations with Local Preconditioning*, PhD Thesis, University of Michigan, 1995.

[57] P. R. M. Lyra: *Unstructured Grid Adaptive Algorithms for Fluid Dynamics*, PhD Thesis, Swansea, 1994.

[58] P. R. M. Lyra, K. Morgan: *TVD algorithms for the solution of the compressible Euler equations on unstructured meshes*, Int. J. Numer. Meth. Fluids, Vol. 19, 1994, pp. 827-847.

[59] K. Michalak, C. Ollivier Gooch: *Differentiability of Slope Limiters on Unstructured Grids*, In: Proceedings of Fourteenth Annual Conference of the Computational Fluid Dynamics Society of Canada, 2006.

[60] M. Möller: *Adaptive High-Resolution Finite Element Schemes*, PhD Thesis, TU Dortmund, 2008.

[61] K. Morgan, J. Peraire: *Unstructured grid finite element methods for fluid mechanics*, Reports on Progress in Physics, Vol. 61 (6), 1998, pp. 569-638.

[62] A. Nejat: *A Higher-Order Accurate Unstructured Finite Volume Newton-Krylov Algorithm for Inviscid Compressible Flows*, PhD Thesis, Vancouver, 2007.

[63] J. Nicols, D. W. Zingg: *A three-dimensional multi-block Newton-Krylov flow solver for the Euler equations*, AIAA Paper, 2005, Article No. 205-5230.

[64] M. Ndjinga: *Influence of interfacial pressure on the hyperbolicity of the two-fluid model*, C. R. Acad. Sci. Paris, Series I, Vol. 344, 2007, pp. 407-412.

[65] E. J. Nielson, W. K. Anderson, R. W. Walters, D. E. Keyes: *Application of Newton-Krylov methodology to a three-dimensional unstructured Euler code*, AIAA Paper, 1995, Article No. 95-1733.

[66] M. Nishida, S. Ishimaru: *Numerical analysis of gas-solid two-phase nonequilibrium nozzle flows*, JSME Int. J., Series II, Vol. 33 (3), 1990, pp. 494-500.

[67] J. S. Park, S. W. Beak: *Interaction of a moving shock wave with a two-phase reacting medium*, Int. J. of Heat and Mass Transf., Vol. 46, 2003, pp. 4117-4732.

[68] A. R. Paschedag: *CFD in der Verfahrenstechnik*, (German), Wiley, 2004.

[69] S. V. Patankar: *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.

[70] M. Pelanti, R. J. LeVeque: *High-resolution finite volume methods for dusty gas jets and plumes*, SIAM J. Sci. Comp., Vol. 28 (4), 2006, pp. 1335-1360.

[71] B. Pollul: *Iterative Solvers in Implicit Time Integration for Compressible Flows*, PhD Thesis, RWTH Aachen, 2008.

[72] J. M. Powers, D. S. Stewart, H. Krier: *Theory of two-phase detonation. Part I. Modeling*, Combust. and Flame, Vol. 80, 1990, pp. 264-279.

[73] A. Pueyo, D. W. Zingg: *An efficient Newton-GMRES solver for aerodynamic computations*, AIAA Paper, 1997, Article No. 97-1955.

[74] J. J. Quirk: *A contribution to the great Riemann solver debate*, Int. J. Num. Meth. Fluids, Vol. 18, 1994, pp. 555-574.

[75] P. L. Roe: *Approximate Riemann solvers, parameter vectors and difference schemes*, J. Comp. Phys., Vol. 43, 1981, pp. 357-372.

[76] A. Rohde: *Eigenvalues and eigenvectors of the Euler equations in general geometries*, AIAA Paper, 2001, Article No. 2001-2609.

[77] Y. Saad: *Iterative Methods for Sparse Linear Systems*, SIAM, second edition, 2003.

[78] L. Sainsaulieu: *Finite volume approximation fo two phase-fluid flows based on an approximate Roe-type Riemann solver*, J. Comp. Phys., Vol. 121, 1995, pp. 1-28.

[79] T. Saito: *Numerical analysis of dusty-gas flows*, J. Comp. Phys., Vol. 176, 2002, pp. 129-144.

[80] G. A. Saltanov: *Neravnovesniye i nestacionarniye processy v gazodinamike odnofaznih i dvuhfaznih sred*, (Russian), Springer, Moscow, 1979.

[81] R. Saurel, M. Larini, J. C. Loraud: *Ignition and growth of a detonation by a high energy plasma*, Shock Waves, Vol. 2 (1), 1992, pp. 91-102.

[82] R. Saurel, R. Abgrall: *A multiphase Godunov method for compressible multifluid and multiphase flows*, J. Comp. Phys., Vol. 150, 1999, pp. 425-467.

[83] F. Schieweck: *On the role of boundary conditions for CIP stabilization of higher order finite elements*, Preprint, 2007.

[84] R. A. Shapiro: *An Adaptive Finite Element Solution Algorithm for the Compressible Euler Equations*, PhD Thesis, Massachusetts, 1988.

[85] V. Selmin: *Finite element solution of hyperbolic equations. II. Two-dimensional case*, INRIA Research Report, Vol. 708, 1987.

[86] V. Selmin: *The node-centered finite volume approach: Bridge between finite differences and finite elements*, Comp. Meth. Appl. Mech. Eng., Vol. 102, 1993, pp. 107-138.

[87] V. Selmin, L. Formaggia: *Unified construction of finite element and finite volume discretizations for compressible flows*, Int. J. Num. Meth. Eng., Vol. 39, 1996, pp. 1-32.

[88] S. Sivier, E. Loth, R. Baum, R. Löhner: *Unstructured adaptive remeshing finite element method for dusty shock flow*, Shock Waves, Issue 3, 1999, pp. 15-23.

[89] A. Sokolichin: *Mathematische Modellbildung und Numerische Simulation von Gas-Flüssigkeits-Blasenströmungen*, (German), Habilitation Thesis, Stuttgart, 2002.

[90] M. Sommerfeld: *The unsteadyness of shock waves propagating through gas-particle mixtures*, Experiments in Fluids, Vol. 3 (4), 1985, pp. 197-206.

[91] H. Städke: *Gasdynamic Aspects of Two-Phase Flow*, Wiley, 2006.

[92] E. F. Toro: *Riemann Solvers and Numerical Methods for Fluid Dynamics*, A Practical Introduction, Springer, 1999.

[93] J.-Y. Trépanier, M. Reggio, D. Ait-Ali-Yahia: *An implicit flux-difference splitting method for solving the Euler equations on adaptive triangular grids*, Int. J. Num. Meth. Heat Fluid Flow, Vol. 3, 1993, pp. 63-77.

[94] M. Utku, G. F. Carey: *Boundary penalty techniques*, Comp. Meth. Appl. Mech. Engrg., Vol. 30, 1982, pp. 103-118.

[95] D. L. Whitaker, D. C. Slack, R. W. Walters: *Solution algorithms for the two-dimensional Euler equations on unstructured meshes*, AIAA Paper, 1990, Article No. 90-0697.

[96] H. Yang: *Wave Packets and their Bifurcations in Geophysical Fluid Dynamics*, Springer 1991.

[97] H. C. Yee: *Construction of explicit and implicit symmetric TVD schemes and their applications*, J. Comp. Phys., Vol. 43, 1987, pp. 151-179.

[98] H. C. Yee, R. F. Warming, A. Harten: *Implicit total variation diminishing (TVD) schemes for steady-state calculations*, J. Comp. Phys., Vol. 57, 1985, pp. 327-360.