On Building Blocks of Finite Volume Methods
– Limiter Functions and Riemann Solvers –

Der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University vorgelegte Dissertation zur Erlangung des akademischen Grades einer Doktorin der Naturwissenschaften

von

Dipl.-math. techn. Birte Schmidtmann

aus Bremerhaven
This work is dedicated to the memory of my grandfather

Dr. Eckart Volz

who passed away while I finished this work.

Science is a way of life.
Science is a perspective.
Science is the process that takes us from confusion to understanding in a manner that’s precise, predictive and reliable – a transformation, for those lucky enough to experience it, that is empowering and emotional.

Brian Greene
Acknowledgements

First of all, I would like to thank my supervisor Manuel Torrilhon for everything I have learned from him. He was always available for my countless questions, whether they were of scientific, future-oriented or bureaucratic nature. When I was unsatisfied with my work, he kept on encouraging me and gave helpful advice while he asked the right critical questions when I was too confident on the results.

I am very grateful to Rémi Abgrall for the many valuable discussions we had during my stays in Zürich, for sharing his insights in the academic world and his continuous support. Rémi knows me for the biggest part of my academic career and always encouraged me to look ahead and believe in myself. Our many conversations and Skype sessions were of great value to me and are always a pleasure.

Also, I wish to thank Andrew Winters, who taught me a lot about DG, MHD, and entropy. Working with Andrew has always been fun and productive at the same time.

I want to express my gratitude to the many people who made me feel at home during my stay in Valencia, especially Antonio Marquina, Pep Mulet, Miguel Ángel Aloy, Isabel Cordero, Antonio Baeza, Toni Veintimilla, Sergio López, David Zorío, and last but not least Erik Sarrión Pedralva.

I would like to thank Benjamin Seibold for his hospitality and for all the discussions on limiter functions during my stay in Philadelphia.

Furthermore, I want to thank everyone at MathCCES who has helped me along and kept encouraging me. Especially, I would like to thank
Thomas Camminady for being my best friend, for all our fun activities, our conversations on math, politics, social problems, ... basically for everything one might or might not encounter in life.

Philipp Otte for being a great office mate. All the discussions on the white board about each other’s research which we never fully understood but still successfully commented on. Our office was the perfect support group for the time of our theses.

Kerstin Küpper for being such a good and reliable friend throughout my time in Aachen and Valencia.

Pascal Richter for our discussions on renewable energy, good advice, and for cheering me up with his seemingly permanent good mood.

Furthermore, my time at RWTH Aachen would have been less fun without Jonas Bünger, Torsten Trimborn, Julian Köllermeier, Armin Westerkamp, Roman Schärer, and many more.

It would have been impossible to travel to conferences, workshops and invited talks without the help of Robert Pattai and Babette Biehne who always helped me taking care of Nora and Cola. Thank you for loving them as if they were your own!

My work-life balance was ensured – amongst others – by Roni Gärtner, Thomas Braun, and Lea Reinhardt. What a pleasure to talk about other things than math ;)

Most of all, I am extremely grateful to my parents and my brother who continuously supported me, even with crazy ideas. This work would not have been possible without them.

Finally, I am deeply thankful to my grandfather, Dr. Eckart Volz. He always told me that the PhD is the best time of your life and motivated me to conduct a thesis in the first place.
Contents

Abstract 1

Zusammenfassung 3

Résumé 7

1 Introduction 11

2 Finite Volume Methods 15

2.1 General Formulation for Conservation Laws 15

2.2 Building Blocks of Finite Volume Methods 19

2.3 Why Third Order? 21

2.4 Why Generic Riemann Solvers? 22

3 Part I – Higher Order Reconstruction 25

3.1 Limiter and the MUSCL Framework 27

3.1.1 Classical Limiter Functions and Useful Properties 30

3.2 Third-Order Limiter 33

3.2.1 Missing Symmetry of $\phi_{CT}$ and Definition of $\phi_{3L}$ 35

3.2.2 Asymptotic Region 39

3.2.3 Interpretation in 2D Slope Domain 39

3.2.4 A New Smoothness Indicator 43

3.3 WENO 50

3.3.1 Interpretation of WENO3 in 2D Slope Domain 51

3.3.2 WENO Smoothness Indicators 52

3.4 Unifying View 54
## Part I – Flux Limiters

3.4.1 Comparison of Limiter Functions ........................................ 55
3.5 Numerical Results comparing $H_{3L}^{(c)}$ and WENO3 ............... 57
  3.5.1 Preliminary Test Case .............................................. 58
  3.5.2 Advection Equation with Smooth Initial Condition ............. 59
  3.5.3 Advection Equation with Discontinuous Initial Condition .... 61
  3.5.4 Initial Condition Containing Smooth and Non-Smooth Features .................................................. 63
  3.5.5 Sod Shock Tube Problem ........................................... 66
  3.5.6 Shu-Osher Problem .................................................. 68
  3.5.7 Ideal Magnetohydrodynamics ...................................... 69
3.6 Generalization to Non-Equidistant Grids ............................... 72
  3.6.1 Numerical Results for Non-Equidistant Grids .................... 74
3.7 Third-Order Limiter in Two Space Dimensions ........................ 76
  3.7.1 Formulation for 2D Cartesian Grids ............................... 77
  3.7.2 Adaptive Mesh Refinement (AMR) ................................ 79
  3.7.3 Non-Uniform Rectangular 2D Grids ............................... 81
  3.7.4 Numerical Results in Two Dimensions ............................. 82

## Part II – Riemann Solvers

4.1 Classical Riemann Solvers .................................................. 93
  4.1.1 Upwind Godunov solver .......................................... 94
  4.1.2 Lax-Friedrichs and Rusanov solver .............................. 96
  4.1.3 Lax-Wendroff solver ............................................. 96
  4.1.4 HLL solver ......................................................... 97
  4.1.5 Comparison of Classical Riemann Solvers ....................... 98
4.2 Hybrid Riemann Solvers .................................................... 99
  4.2.1 FORCE ............................................................ 99
  4.2.2 MUSTA ............................................................ 100
  4.2.3 HLLX .............................................................. 100
4.3 Jacobian-Free Implementation ............................................ 102
4.4 HLLXω - a Family of Hybrid Riemann Solvers ........................ 104
  4.4.1 Beyond Monotonicity ............................................. 105
  4.4.2 Modified Equation .............................................. 106
In this thesis we are interested in numerically solving conservation laws with high-order finite volume methods. Hyperbolic systems of partial differential equations are especially challenging since even smooth initial flows may develop discontinuities in finite time. Naively discretizing such flows with high-order schemes may lead to undesired oscillations at discontinuities. First-order methods do not encounter this problem since shocks are smeared out. Nevertheless, high-order schemes are in demand because they have the advantage of reaching a fixed error bound on coarser grids than low-order methods. This reduces the overall computational time and thus the total cost.

Combining the advantages of several methods, limiter functions change from high-to low-order whenever necessary. This avoids oscillations at discontinuities while maintaining high-order accuracy at smooth parts of the solution. Thus, the resulting schemes are applicable to physically relevant problems which often contain smooth parts as well as large gradients, discontinuities, or shocks.

The aim of this work is the development of third-order finite volume methods by improving building blocks of the method. This means, we identify main routines in the finite volume framework and present new concepts for improving their performance.

We focus on two building blocks. First, the high-order reconstruction of interface values using limiter functions. Second, the numerical flux function, also referred to as Riemann solver. The former is crucial for the order of accuracy of the solution while the latter determines the amount of dissipation added to the scheme.

We develop a new third-order accurate reconstruction function for the spatial approximation of hyperbolic conservation laws. This reconstruction switches between first- and third-order, resulting in a scheme which is high-order accurate in smooth parts of the solution, does not create oscillations at discontinuities, and avoids extrema clipping as encountered by total variation diminishing (TVD) methods.

The novel limiter only needs information from the cell of interest and its nearest neighbors, thus keeping the stencil as compact as possible for obtaining third-order accuracy. Furthermore, the reconstruction remains in the traditional second-order framework, easing the implementation of the limiter in existing codes. Finally, a decision criterion without artificial parameters is incorporated in the limiter. This
decision criterion distinguishes shocks and large gradients from extrema, thus ensuring accurate shock capturing.

The obtained reconstructions at each side of the cell boundaries are then inserted into the numerical flux function which solves local Riemann problems. Many numerical flux functions, also referred to as Riemann solvers, have been developed over the last decades. However, most classical solvers add too much dissipation to the scheme such that discontinuities are smeared out. On the other side of the spectrum, Riemann solvers that do not add too much dissipation need information on the eigenstructure which is costly to compute for large systems.

There is the need for new Riemann solvers that avoid solving for the eigensystem and still reproduce all waves of the system with less dissipation than classical methods such as Rusanov and Harten-Lax-van Leer (HLL). We present a hybrid family of Riemann solvers, requiring only an estimate of the globally fastest wave speeds in both directions. Thus, the new solvers are particularly efficient for large systems of conservation laws when no explicit expression for the eigensystem is available or expensive to compute.

For the validation of the developed schemes we conduct a series of numerical experiments. First, we demonstrate that the novel high-order limiter function obtains the desired third-order accuracy for smooth solutions. Test cases include smooth and discontinuous linear transport, Euler equations, and ideal magnetohydrodynamics (MHD). Problems are presented in one and two space dimensions, on uniform as well as non-uniform grids and with adaptive mesh refinement.

In a second step, the hybrid family of Riemann solvers is tested in a first-order framework. Here, we show that the newly developed solvers induce less dissipation than schemes with comparable input data. This leads to sharper gradients and less smearing at discontinuities. Numerical examples contain linear transport, Euler equations, ideal MHD, as well as the regularized 13-moment equations (R13).

Finally, both parts of this work are combined to obtain third-order accurate results. We reconstruct using the novel third-order limiter and insert the reconstructed interface values into the hybrid family of Riemann solvers.

For all numerical examples, we also implement comparable methods to ascertain the quality of our schemes. The solutions obtained with the newly developed methods indeed indicate better or equally-good results and an excellent performance.

The results presented in this thesis have largely been published and can be found in a more concise form in [42] – [48].
Zusammenfassung


Limiter Funktionen kombinieren die Vorteile mehrerer Methoden und ändern das Schema von hoher zu niedrigerer Ordnung wenn nötig. Dies vermeidet Oszillationen bei Diskontinuitäten bei gleichzeitiger Erhaltung der hohen-Ordnung Genauigkeit in glatten Teilen der Lösung. Dadurch sind die resultierenden Schemata anwendbar auf physikalisch relevante Probleme, deren Lösungen oft glatte Teile als auch Diskontinuitäten, große Steigungen oder Schocks enthalten.


Wir konzentrieren uns auf zwei Bausteine. Erstens, die hohe-Ordnung Rekonstruktion von Zwischenzellwerten mit Limiter Funktionen und zweitens die numerische Flussfunktion, auch Riemann Löser genannt. Ersteres ist entscheidend für die Konvergenzordnung der Lösung während letzteres die Menge der Dissipation bestimmt, die dem Schema hinzugefügt wird.

Wir entwickeln eine neue dritte-Ordnung Rekonstruktion für die räumliche Approximation von hyperbolischen Erhaltungsgleichungen. Diese Rekonstruktion schaltet zwischen erster und dritter Ordnung, was zu einem Schema führt, das in glatten Teilen der Lösung hohe Ordnung erreicht und bei Diskontinuitäten keine Oszillationen erzeugt. Außerdem wird das Abschneiden von glatten Extremstellen vermieden, ein Nachteil bei total-variationsvermindernden (TVD) Methoden.
Der neue Limiter ist kompakt, da nur Zellmittelwerte der zentralen Zelle sowie der direkten Nachbarzellen benötigt werden. Darüber hinaus bleibt die Rekonstruktion in der Struktur von traditionellen zweite-Ordnung Verfahren, was die Implementierung des Limiters in bestehenden Codes erleichtert. Schließlich umfasst der Limiter ein Entscheidungskriterium ohne künstliche Parameter, welches Schocks und große Gradienten von Extremstellen unterscheidet.


Schließlich werden beide Teile dieser Arbeit kombiniert, um Ergebnisse dritter Ordnung zu erhalten. Wir rekonstruieren mit dem neuen dritte-Ordnung Limiter und fügen die rekonstruierten Zwischenzellenwerte in die hybride Familie von Riemann Lösern ein.

Für alle numerischen Beispiele testen wir auch vergleichbare Methoden, um die Qualität der entwickelten Schemata zu überprüfen. Die Lösungen, die mit den neu entwickelten Methoden erhalten wurden, zeigen bessere oder vergleichbar gute Ergebnisse und eine sehr gute Leistung.

Résumé

Dans cette thèse, nous nous intéressons à la résolution numérique des lois de conservation au moyen de méthodes de volumes finis. Les systèmes hyperboliques d’équations aux dérivées partielles sont particulièrement difficiles à résoudre car, même des conditions initiales très régulières, des discontinuités peuvent développer en temps fini. Une discrétisation naïve de ces équations au moyen de schémas d’ordre élevé conduit généralement à des solutions numériques possédant des oscillations numériques indésirables au voisinage des discontinuités. Les méthodes de premier ordre n’ont pas ce problème car les chocs sont étalés. Néanmoins, les méthodes d’ordre élevés ont l’avantage d’atteindre une erreur comparable sur des maillages plus grossiers que les méthodes d’ordre faible. Cela permet a priori de réduire le temps de calcul et donc le coût total des simulations.

En combinant les avantages de plusieurs méthodes, les limiteurs permettent de changer l’ordre du schéma, en passant de l’ordre élevé à l’ordre un, chaque fois que cela est nécessaire. Cela évite les oscillations aux discontinuités tout en conservant l’ordre élevé dans les parties lisses de la solution. Ces schémas s’appliquent à des problèmes physique pertinents où la solution est souvent très régulière sur une grande partie du domaine de calcul mais où des zones à fort gradients ou encore des chocs existent dans d’autres parties du domaine.

Le but de ce travail est le développement de méthodes de volumes finis du troisième ordre. Pour cela, nous discutons les principales parties des algorithmes de stabilisation et nous présentons de nouveaux concepts destinés à améliorer la performance des schémas.

Nous nous concentrons principalement sur deux parties des algorithmes. D’abord, la reconstruction d’ordre élevé des valeurs d’interface à l’aide des limiteurs. Deuxièmement, le flux numérique. La première est cruciale pour l’ordre de précision de la solution alors que la seconde détermine la quantité de dissipation ajoutée au schéma.

Nous développons une nouvelle fonction de reconstruction du troisième ordre pour l’approximation spatiale des lois de conservation hyperbolique. L’ordre de cette reconstruction évolue entre le premier et le troisième ordre. Cela permet d’obtenir
un schéma d’ordre élevé dans les parties lisses de la solution qui ne produit pas d’oscillations aux discontinuités. De plus, les atténuations d’extrema rencontrées par les méthodes TVD* sont évitées.

Ce nouveau limiteur ne nécessite que des informations sur la cellule centrale et ses voisins les plus proches: le schéma compact est compact. La reconstruction reste dans le cadre des schémas traditionnels de deuxième ordre, facilitant la mise en œuvre du limiteur dans des codes déjà existants. Enfin, un critère de décision sans paramètre artificial est incorporé dans le limiteur. Ce critère de décision permet de distinguer entre les chocs et les grands gradients d’une part et les extrêmes lisses d’autre part, donc assurant une capture de choc précise.

Les reconstructions ainsi obtenues de chaque côté des arêtes de la cellule sont ensuite insérées dans le flux numérique qui permet de résoudre localement les problèmes de Riemann. De nombreux flux numériques, également appelées solveurs de Riemann, ont été développés au cours des dernières décennies. Cependant, la plupart des solveurs classiques ajoutent trop de dissipation au schéma si bien que les discontinuités sont trop étalées. D’un autre coté, les solveurs de Riemann qui n’ajoutent pas trop de dissipation ont besoin d’informations sur la structure spectrale de la matrice jacobienne du flux. Pour les grands systèmes, ceci est coûteux.

Il existe donc un besoin de nouveaux solveurs de Riemann qui n’ont pas besoin d’informations trop précises sur la structure spectrale de la matrice jacobienne mais permettent tout de même de reproduire toutes les ondes du système avec moins de dissipation que les méthodes classiques telles que Rusanov et Harten-Lax-van Leer (HLL). Nous présentons une famille hybride de solveurs de Riemann ne nécessitant qu’une estimation de la vitesse d’onde globale la plus rapide. Ainsi, les nouveaux solveurs sont particulièrement efficaces pour les grands systèmes de lois de conservation quand aucune expression explicite de la structure propre n’est disponible ou quand celle ci est coûteuse à calculer.

Pour valider ces nouveaux schémas, nous effectuons une série d’expériences numériques. Tout d’abord, nous démontrons que le nouveau limiteur d’ordre élevé permet bien d’obtenir la précision souhaitée du troisième ordre pour des solutions lisses. Les cas d’épreuve comprennent le transport linéaire lisse et discontinu, les équations d’Euler et la magnétohydrodynamique (MHD) idéale. Des cas test sont présentés en une ou deux dimensions d’espace, sur des maillages uniformes et non-uniformes, avec ou sans raffinements adaptatifs.

*acronyme de l’anglais Total Variation Diminishing
Dans une deuxième partie, une famille hybride de solveurs de Riemann est testée. Nous montrons que ces nouveaux solveurs sont moins dissipatifs d’autres solveurs nécessitant des données d’entrée comparables. Cela permet d’obtenir des gradients plus prononcés et moins d’étalement aux discontinuités. Des expériences numériques sont faites sur des problèmes de transport linéaire, des équations d’Euler, MHD idéale, ainsi que les équations aux 13-moments régularisées (R13).

Pour finir, les deux parties de ce travail sont combinées pour obtenir des résultats de troisième ordre. Les valeurs d’interface sont reconstruits avec le nouveau limiteur et sont utilisées comme valeur d’entrée dans la famille hybride de solveurs de Riemann. Pour tous les exemples numériques, nous implémentons également des méthodes comparables pour vérifier la qualité de nos schémas. Les solutions obtenues avec les nouveaux méthodes développées montrent des résultats meilleurs ou égaux et une performance excellente.

Chapter 1

Introduction

In many applications there is the need to deal with two opposite constraints: having a high-order accurate method and at the same time dealing with discontinuities or singularities. Typical examples include – but are not limited to

- smooth flows and shock waves for compressible flows,
- high and low frequencies for wave problems,
- equilibrium and perturbations *e.g.* in shallow water equations or cosmological applications,
- constrained problems in magnetohydrodynamics (MHD).

These topics are at the core of many research groups worldwide and important progresses have been made over the last decade. Hereby, three main areas are currently being explored by the hyperbolic community: developing high-order accurate schemes, generating hybrid unstructured meshes using high-order curved elements, and high performance computing.

In this work, we will focus on the first area, *i.e.* advances in the development of high-order numerical schemes. In particular, we are interested in hyperbolic conservation laws, since many physical phenomena are modeled by this type of equations. Examples of hyperbolic partial differential equations (PDEs) in conservation form are wave propagation, the Euler equations, or the ideal magnetohydrodynamics (MHD) equations.
There are many different options to solve and analyze PDEs numerically, \textit{e.g.} using finite difference, finite element or finite volume methods. Among these methods, finite volumes are a popular choice and a powerful class of methods for solving hyperbolic conservation laws. They can be used for approximating linear as well as non-linear problems and they are naturally conservative \cite{34}.

This whole work is dedicated to the improvement of finite volume methods for hyperbolic conservation laws. To cope with the overall aim – the improvement – the task is split into sub-tasks, identifying the building blocks of finite volume methods, pointing out possibilities for improvement, developing new methods, proving theoretical results and implementing the new components.

We first introduce and summarize some of the basic notation and theory of finite volume methods in Chapter \ref{chap2}. This chapter also highlights the tasks on hand for reaching the overall objective of this thesis, namely developing high-order accurate finite volume methods. Furthermore, in Chapter \ref{chap2} the red line connecting the (at first) seemingly unrelated research topics that are discussed in this thesis becomes apparent.

In Chapter \ref{chap3} high-resolution methods are presented and the development of a new limiter function is explained. For the sake of simplicity, the theory is detailed for one-dimensional structured meshes. Numerical experiments are conducted to prove that desired properties are achieved and that the novel limiter compares favorable to other methods. The limiter is then extended to one-dimensional non-uniform grids and two-dimensional meshes including a routine for adaptive mesh refinement (AMR). Furthermore, the limiter function is extended to be applicable to non-uniform two-dimensional grids. We conclude this chapter by presenting more numerical results that show the improvement of solutions obtained with the new limiter compared to other third-order approaches.

In Chapter \ref{chap4} we present different choices for the numerical flux function, also referred to as Riemann solver. Firstly, classical choices are presented, then also combined Riemann solvers are treated. Finally, a new family of hybrid Riemann solvers is developed, especially applicable to large systems of conservation laws. In order to verify the performance of the developed schemes, we present numerical comparisons of the family of solvers and classical Riemann solvers.
The last theoretical part, Chapter 5, introduces the concept of entropy. Here we prove entropy stability of the diffusive terms of many well-known Riemann solvers and show that all solvers presented in Chapter 4 fulfill the entropy stable property. Thus, we generate novel entropy stable numerical flux functions. Numerical results combining the theory of high-order reconstruction and the novel family of Riemann solvers can be found in Chapter 6. Finally, in Chapter 7 we draw conclusions on the presented methods and discuss some open questions and suggestions for future work.

Part of the results presented in this thesis are published in


Chapter 2

Finite Volume Methods

2.1 General Formulation for Conservation Laws

We are interested in the numerical approximation of hyperbolic systems of conservation laws of the form

\[ \partial_t u(x, t) + \nabla \cdot f(u(x, t)) = 0. \]  
(2.1)

Here, \( u : \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}^m \) is a vector of \( m \) conserved quantities or state variables \( u = (u_1, \ldots, u_m)^T \). More specifically, \( u_j \) represents the density function of the \( j \)-th variable of interest and

\[ \int_{x_0}^{x_1} u_j(x, t)dx \]  
(2.2)

can be interpreted as the total quantity of this variable in the interval \([x_0, x_1]\) at time \( t \), see [33]. The change of \( \int_{x_0}^{x_1} u_j(x, t)dx \) over time can be due to two effects. Either, the flux \( f_j : \mathbb{R}^m \rightarrow \mathbb{R}^m \) through domain boundaries or a sink or source, generally referred to as source, even though physically the term might represent a sink. The density function of such a source is denoted by \( s : \mathbb{R}^m \times \mathbb{R}^d \times \mathbb{R}^+ \rightarrow \mathbb{R}^m \).

In case a source term exists, system (2.1) reads

\[ \partial_t u(x, t) + \nabla \cdot f(u(x, t)) = s(u(x, t), x, t). \]  
(2.3)

This system is often called balance law rather than conservation law.
When Eq. (2.1) is equipped with the initial condition $u_0(x)$, this initial value problem is called Cauchy problem [33]. To avoid boundary effects, we assume the initial conditions to be either periodic or with compact support. The initial conditions are supposed to be either smooth or piece wise smooth functions. The flux Jacobian matrices are given by $A_j(u) = \frac{\partial f_j}{\partial u}$ for $j = 1, \ldots, d$. Since we are interested in hyperbolic problems, any linear combination of the flux Jacobians \[ \sum_{j=1}^{m} \alpha_j A_j(u), \alpha_j \in \mathbb{R} \] should be diagonalizable with $m$ real eigenvalues $\lambda_j(u)$ for each value $u$. If additionally the eigenvalues are distinct for all $u$, the system is called strictly hyperbolic [33]. If the matrix $A_j$ does not depend on the values $u(x, t)$, system (2.1) is called linear, if a dependency on $u$ exists, it is called nonlinear. Whenever it is possible to find a matrix $A_j$ such that \[ \frac{\partial}{\partial x_j} (f_j(u(x, t))) = A_j(u) \frac{\partial}{\partial x_j} u, \] the system can be written in quasi-linear form

\[
\partial_t u + \sum_{j=1}^{d} A_j(u) \frac{\partial}{\partial x_j} u = 0. \tag{2.4}
\]

Smooth initial flows governed by (2.1) may develop discontinuities (e.g. shocks) in finite time. Thus, solutions are sought in the weak sense [34]. The idea is that we need a formulation that does not require differentiability of the solution $u(x, t)$. This can be obtained, by multiplying Eq. (2.1) with a smooth test function $\phi(x, t) \in C_0\infty(\mathbb{R} \times \mathbb{R}_+) \text{ and integrating over space and time } [30]

\[
\int_0^\infty \int_{-\infty}^{\infty} [\phi(x, t) \partial_t u(x, t) + \phi(x, t) \partial_x (f(u(x, t)))] \, dx \, dt = 0. \tag{2.5}
\]

With integration by parts, the smooth test functions are differentiated and the solution $u$ does not contain derivatives anymore\[
\int_0^\infty \int_{-\infty}^{\infty} [\partial_t \phi(x, t) u(x, t) + \partial_x \phi(x, t) f(u(x, t))] \, dx \, dt = - \int_{-\infty}^{\infty} \phi(x, 0) u(x, 0). \tag{2.6}
\]

Due to the compact support of $\phi$, the boundary terms $\phi(x, t) f(u(x, t))|_{x=\pm\infty}$ vanish and $\phi(x, t) u(x, t)|_{t=0} = - \int_{-\infty}^{\infty} \phi(x, 0) u(x, 0)$ yields the right hand side. A function $u(x, t)$ is called a weak solution of Eq. (2.1) with appropriate initial conditions $u_0(x)$, if Eq. (2.6) holds for all test functions $\phi(x, t) \in C_0\infty(\mathbb{R} \times \mathbb{R}_+)$. 

16
Unfortunately, weak solutions are not unique, even though from physics we expect one solution per specific initial data. Therefore, in order to find the right solution, additional admissibility criteria are needed [34]. One concept is to add artificial viscosity – which leads to a parabolic equation with a unique solution – and consider the vanishing viscosity solution. However, it is difficult to directly use this method. Thus, other concepts have been introduced to verify the physical admissibility of weak solutions of hyperbolic conservation laws. These conditions are so-called entropy conditions [34] and will be further dwelled upon in Chapter 5.

From now on we will use the term solution even though we are talking about weak solutions, i.e. the terms will be used interchangeably. Also, for the time being we will assume that the entropy conditions are fulfilled by our solutions.

For the sake of simplicity, we first restrict our discussion and analysis to the scalar, one-dimensional case \( m = d = 1 \). However, the ideas developed in Chapter 3 are also applicable to systems \((m > 1)\), as shown in Sec. 3.5. The generalization to multiple dimensions \((d > 1)\) is treated in Sec. 3.7. Furthermore, we start by considering a regular grid in space on a finite domain \( \Omega \) as depicted in Fig. 2.1. This means \( \Omega \) is divided into \( K \) non-overlapping grid cells

\[ \Omega = \bigcup_{i=1}^{K} C_i \quad \text{with} \quad C_i = [x_{i-1/2}, x_{i+1/2}], \quad i = 1, \ldots, K \]

and cell centers denoted by \( x_i \). The space intervals are of size \( \Delta x_i = x_{i+1/2} - x_{i-1/2} \), where \( x_{i\pm j} = x_i \pm j\Delta x \). In the equidistant case all intervals are of the
same size, \( i.e. \Delta x_i \equiv \Delta x \), see Fig. 2.1. The extension to non-equidistant grids is discussed in detail in Sec. 3.6.

As mentioned above, the quantity of interest at time \( t \) is given by (2.2), thus the definition of the cell average of the true solution \( u(\cdot, \cdot) \) follows naturally by

\[
\bar{U}(x, t) = \frac{1}{\Delta x} \int_{x-\Delta x/2}^{x+\Delta x/2} u(\xi, t) \, d\xi.
\]  
(2.7)

With this definition, \( \bar{U}^n_i = \bar{U}(x_i, t^n) \) is called the cell average of the true solution \( u \) in cell \( C_i \) at time \( t^n \). The goal is to find an update rule to advance approximate cell averages (which are defined below) from a given time \( t^n \) to a new time \( t^{n+1} = t^n + \Delta t \), such that the true cell averages are approximated with high order of accuracy. In addition, the approximate solution should not develop any (relevant) spurious oscillations.

Integrating Eq. (2.1) over cell \( C_i \) and dividing by \( \Delta x \) yields

\[
\frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \partial_t u(\xi, t) \, d\xi = -\frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \partial_x f(u(\xi, t)) \, d\xi
\]
\[
\iff \frac{d}{dt} \bar{U}(x_i, t) = -\frac{1}{\Delta x} \left( f(u(x_{i+1/2}, t)) - f(u(x_{i-1/2}, t)) \right)
\]  
(2.8)

which is an exact formulation. So far, no approximation has been applied. However, to obtain the necessary information \( f(u(x_{i\pm1/2}, t)) \) requires a large amount of work. We therefore want to approximate the exact averaged quantity of interest \( \bar{U}^n_i \) by \( \bar{u}^n_i \), such that

\[
\bar{u}^n_i = \bar{U}^n_i + O(\Delta x^p)
\]  
(2.9)

holds true. This means that we approximate the cell average of the true solution with \( p \)-th order accuracy, \( p \geq 1 \). The quality of the approximation \( \bar{u}_i \) depends on the accurate approximation of the flux function at the cell boundaries \( f(u(x_{i\pm1/2}, t)) \). This is achieved by constructing numerical flux functions \( \hat{f}_{i\pm1/2} \) of the form \( \hat{f}_{i-1/2} = \hat{f}(\bar{u}_{i-r}, \ldots, \bar{u}_{i+r-1}) \) and \( \hat{f}_{i+1/2} = \hat{f}(\bar{u}_{i-r+1}, \ldots, \bar{u}_{i+r}) \) with \( r \in \mathbb{N} \). In general, the numerical flux function depends on \( 2r \) arguments. However in this work, we are only interested in fluxes with two input values, \( i.e. r = 1 \) and thus

\[
\hat{f}_{i-1/2} = \hat{f}(\bar{u}_{i-1}, \bar{u}_i),
\]
\[
\hat{f}_{i+1/2} = \hat{f}(\bar{u}_i, \bar{u}_{i+1}).
\]  
(2.10)
When we are not talking about a specific cell interface but more general, we denote the numerical flux function by \( \hat{f}(U_L, U_R) \), where \( U_L \) and \( U_R \) are placeholders for the values to the left and right of a cell interface.

For the resulting scheme to be conservative, the numerical flux function has to be constructed such that it is Lipschitz continuous and consistent with the true flux function, i.e. \( \hat{f}(u, u) = f(u) \). In summary, the evolution of the approximate cell averages \( \bar{u}_i \) in semi-discrete form is given by

\[
\frac{d\bar{u}_i}{dt} = -\frac{1}{\Delta x} \left( \hat{f}_{i+1/2} - \hat{f}_{i-1/2} \right).
\] (2.11)

This formulation is continuous in time and has the advantage that we can first concentrate on the discretization in space, obtaining a system of ordinary differential equations (ODEs) of the form \( \frac{d\bar{u}}{dt} = L(\bar{u}) \). Note that this system is coupled, since the \( \hat{f}_{i\pm1/2} \) depend not only on \( \bar{u}_i \) but also the neighboring values.

### 2.2 Building Blocks of Finite Volume Methods

In the process of solving Eq. (2.1) numerically, certain routines need to be applied in every cell \( C_i \) or for all time steps \( t^n \). This means, these methods have to be applied many times for solving Eq. (2.1) and therefore significantly determine the outcome. These methods or routines are called building blocks.

From the discussion on benefits of the semi-discrete formulation (2.11) we can already see that the time integration plays an important role in the solution process of finite volume methods. Thus, time discretization schemes can be considered a building block of finite volume methods. As mentioned above, we can use any standard method for solving the resulting ODEs (2.11), such as Euler’s method, implicit Euler or Runge-Kutta methods. In this work we focus on third-order accuracy and therefore, we use the strong stability preserving (SSP) third-order Runge-Kutta time discretization introduced by Gottlieb et al. [21]. The next choice we can make for the discretization of Eq. (2.11) is in the construction of the numerical flux function \( \hat{f} \). As mentioned above, \( \hat{f} \) needs to be
Lipschitz continuous and consistent with the flux function \( f \). There are many functions fulfilling these criteria. Let e.g. \( u^*(\bar{u}_i, \bar{u}_{i+1}) \) be the solution of the Riemann problem with input values \( \bar{u}_i \) for \( x < x_{i+1/2} \) and \( \bar{u}_{i+1} \) for \( x > x_{i+1/2} \). Then, if the numerical flux function is given by \( \hat{f}_{i+1/2} = f(u^*(\bar{u}_i, \bar{u}_{i+1})) \), Eq. (2.11) yields Godunov’s method.

The numerical flux at interface \( i + 1/2 \) takes as input the values to the left and right of the interface in order to approximate the flux at the interface, \( \hat{f}(u_i, u_{i+1}) \approx f(u(x_{i+1/2}, t)) \) at a fixed point in time. Therefore, it is also referred to as Riemann solver. Since local Riemann problems occur at each interface in each time step, they have to be solved many times for finding the numerical solution. Thus, the Riemann solver can be considered a building block of the finite volume method. Different choices for \( \hat{f} \) as well as the development of new Riemann solvers are discussed in detail in Chapter 4.

Common numerical flux functions yield solutions which are first-order accurate

\[
\hat{f}(\bar{u}_i, \bar{u}_{i+1}) = f(u(x_{i+1/2}, t)) + O(\Delta x).
\] (2.12)

To get higher-order methods, we can precise the input values of the numerical flux function. The function \( \hat{f}_{i+1/2} \) now takes as input the left and right limiting values at boundary \( i + 1/2 \), i.e.

\[
\begin{align*}
\lim_{j \to i+1/2^-} u(x_j) &= u_i^{(-)} \\
\lim_{j \to i+1/2^+} u(x_j) &= u_i^{(+)}. 
\end{align*}
\] (2.13)

When only considering cell mean values, these limits yield exactly

\[
\begin{align*}
u_i^{(-)} &= \bar{u}_i \\
u_i^{(+) &= \bar{u}_{i+1}.
\end{align*}
\] (2.14)

as before, which yields a first-order method [33]. If instead we take into account neighboring cell information to reconstruct more exact limiting values \( u_i^{(\pm)} \), we can obtain higher-order methods. The way these reconstructions are performed significantly controls the scheme’s behavior. Therefore, high-order reconstruction is one of our main concerns in this work.
2.3 Why Third Order?

The reason we are interested in third order reconstruction is manifold. Let us first observe that monotone methods (for scalar conservation laws) are total variation diminishing (TVD), which guarantees convergence to the unique entropy solution in a non-oscillatory manner \cite{33}. However, monotone methods are at most first-order accurate \cite{22}. This leads to shocks being smeared out and smooth parts of the solution being resolved with poor accuracy. These effects are caused by the numerical dissipation incorporated in monotone methods. A certain amount of dissipation is needed to avoid oscillations at shocks and for convergence to the correct solution, however, the amount included in monotone methods seems to be too large.

Considering second-order accuracy, e.g. with a simple linear reconstruction, we can deduce from Godunov’s theorem that the resulting method cannot be monotonicity preserving.

**Theorem 2.1. (Godunov \cite{20})** A linear, monotonicity preserving method is at most first order accurate.

Thus, we need to introduce some non-linearity in the scheme, leading to high-resolution methods. This means, we want to develop methods which are second- (or higher- ) order accurate for smooth solutions and do not cause oscillations at discontinuities \cite{33}. There are several reasons we are not interested in second-order methods. The first argument is simply that many researchers nowadays do not consider second order as high order anymore. For the second reason, the state of the art needs to be highlighted a little bit: For decades, researchers working on high-order schemes were working in the MUSCL (Monotonic Upstream-centered Scheme for Conservation Laws) framework introduced by van Leer \cite{69}. This setting introduces the non-linearity into the scheme by including a limiter function $\phi$ which depends on the smoothness of a small neighborhood. The reconstruction is based on a compact stencil, remaining in the classical setting of only taking into account the three values $\bar{u}_{i-1}, \bar{u}_i$, and $\bar{u}_{i+1}$. Examples of limiter functions in this framework are e.g. Minmod, Superbee, van Leer’s limiter and the Koren
limiter. These functions need three input values, however, the resulting schemes are only second order accurate \[33\]. Nonetheless, with three cell mean values as input, it is possible to construct a quadratic polynomial, leading to a third-order accurate scheme. This yields two advantages: Without enlarging the stencil of the numerical scheme we get one order of accuracy "for free". Furthermore, remaining in the traditional setup the newly developed limiter functions can easily be implemented in existing (commercial) codes by changing only few lines. There is yet another reason why third-order methods are advantageous. Most physically relevant problems include discontinuities. The model problem of a discontinuous function is the step function. Examining analytic solutions of numerical methods with different orders of accuracy with the help of Fourier transform leads to a surprising observation. The amplitude of the over- or undershoot caused by a third-order method is smaller than the amplitude of a second-, fourth- or fifth-order method. This observation is summarized in Fig. 2.2.

### 2.4 Why Generic Riemann Solvers?

Riemann solvers determine the amount of dissipation added to the numerical scheme. Fast and robust methods such as Lax-Friedrichs or local Lax-Friedrichs tend to smear out shocks because they are too diffusive. These solvers only need

---

**Figure 2.2** – Comparing analytical solutions obtained by schemes with different order of accuracy.
the knowledge of the globally fastest signal velocity which is needed for the CFL condition anyway [33].

As detailed in Chapter 4, many solvers exist that bypass this drawback, at least partly. For example the HLL (Harten-Lax-van Leer) scheme [23]. The advantage of HLL and HLL-type methods is that no exact formulation of the eigenvalues and eigenvectors of the flux Jacobian is needed. Only (an estimate of) the globally fastest and slowest signal velocities is required [61]. This offers great flexibility, especially for large systems of hyperbolic conservation laws where the eigenstructure might be computationally expensive to obtain or no analytical expression exists. Although HLL-type schemes add more dissipation than e.g. upwind schemes, they compare favorably because they need less information on the eigendecomposition of the flux Jacobian.

An improvement of the HLL scheme - at least for the Euler equations - is the HLLC (Harten-Lax-van Leer-Contact) method which resolves the contact discontinuity of the equations more accurately. However, HLLC needs additional information in order to restore the contact discontinuity, thus more information on the eigensystem. The main drawback is the limited applicability to only the Euler equations it was developed for.

In this work we develop Riemann solvers which add less dissipation to the scheme than Lax-Friedrichs, Rusanov, and HLL but do not need the eigendecomposition of the flux Jacobian. This means, we promote solvers which are independent of the equations and applicable to large systems of conservation laws.
Chapter 3

Part I – Higher Order Reconstruction

In this part of the thesis we are interested in the input values of the numerical flux function \( \hat{f}(U_L, U_R) \). As indicated in Chapter 2, the choice of \( U_L \) and \( U_R \) determines the order of accuracy of the scheme. Therefore, the focus of this chapter is on the reconstruction of these values, giving a short overview on the state of the art and aiming at third-order accurate numerical schemes.

As introduced in Sec. 2.2, the numerical flux functions at the left and right boundaries of an arbitrary cell \( C_i \) read

\[
\hat{f}_{i-1/2} = \hat{f}(\hat{u}_{i-1/2}^(-), \hat{u}_{i-1/2}^ (+)), \quad (3.1a)
\]

\[
\hat{f}_{i+1/2} = \hat{f}(\hat{u}_{i+1/2}^(-), \hat{u}_{i+1/2}^ (+)). \quad (3.1b)
\]

The flux over the left boundary, \( \hat{f}_{i-1/2} \) takes as input the left and right limit values at cell boundary \( i - \frac{1}{2} \), denoted by \( \hat{u}_{i-1/2}^(-), \hat{u}_{i-1/2}^ (+) \), see Fig. 3.2. Analogously, the input values \( \hat{u}_{i+1/2}^(-) \) and \( \hat{u}_{i+1/2}^ (+) \) are the left and right limiting values at the right cell interface \( i + \frac{1}{2} \). The focus of this chapter is the reconstruction of these interface values. For the sake of simplicity, we shall drop the \( \hat{\cdot} \) for the rest of this work.

The key ingredient for reaching the aim of this chapter – namely reconstructions for third-order accurate numerical schemes – is the way of reconstructing these limit values at the cell boundaries.
In order to reach this goal, Chapter 3 is structured as follows. First, the state of the art is presented in Sec. 3.1, then third-order reconstruction techniques and the development of a new limiter function in one dimension are detailed in Sec. 3.2. The well-known WENO reconstruction is introduced in Sec. 3.3 and a comparison of the two techniques is conducted in Sec. 3.4. One-dimensional numerical results in Sec. 3.5 show the excellent performance of the new limiter function compared to other third-order reconstructions. The second half of this chapter treats the extension to non-equidistant grids 3.6 and to two space dimensions 3.7. Here, we first show how to apply the limiter on two-dimensional Cartesian grids, Sec. 3.7.1 and then also on meshes with adaptive mesh refinement (AMR), Sec. 3.7.2 and non-equidistant two-dimensional meshes 3.7.3. Finally, numerical test cases for these extensions are presented in Sec. 3.7.4.

Parts of this chapter are based on work published with coauthors. The content of Sec. 3.2 is based on [46], which is published with Rémi Abgrall and Manuel Torrilhon in Bulletin of the Brazilian Mathematical Society. The idea of developing a new smoothness indicator based on the one described by Čada and Torrilhon in [8] was proposed by Manuel Torrilhon. I was working out the detailed formulation of the new smoothness indicator, in parts during discussions with Rémi Abgrall. Moreover, I was able to formulate Lemma 3.8, the corresponding proof, and show that the formulation of the new limiter function (3.41) is meaningful.

The work has been extended in [45], published with Benjamin Seibold and Manuel Torrilhon in Springer Journal of Scientific Computing. These results are the main
contribution to Sec. 3.3 and 3.4. While the idea of treating limiter functions in the two-parameter framework and a short write up were given by my coauthors, I sharpened the results and proved many properties of the newly formulated limiter function. Moreover, I implemented the limiter functions and designed numerical test cases that confirmed the predicted results. The generalization to non-equidistant grids as well as two space dimensions has been developed together with Pawel Buchmüller and Manuel Torrilhon. The manuscript has been submitted and is currently under review [48].

3.1 Limiter and the MUSCL Framework

When the numerical flux function was first presented in Chapter 2, it was given with simple cell averages as input, see (2.10). This situation is depicted in Fig. 3.2 and yields a first-order accurate numerical scheme. To obtain higher-order accuracy, more elaborate formulations for the left and right limit at each cell boundary have been introduced, see Eq. (2.13). The left and right limiting values at interface $i + \frac{1}{2}$ for example are denoted by $u_{i+1/2}^(-)$, $u_{i+1/2}^+$. In case of first-order schemes, the interface values reduce to the cell mean values, cf. Eq. (2.14).

Gathering information on the neighboring cell mean values $\bar{u}_{i-1}$ and $\bar{u}_{i+1}$ to include e.g. the slope between these values, yields linear reconstruction functions, denoted by the red lines in Fig. 3.2. Evaluating these functions at the cell interfaces (indicated by the red dots) yields a second-order accurate scheme. The reconstruction in cell $C_i$ can be expressed as

$$u_{i+1/2}^(-) = \bar{u}_i + \frac{\Delta x}{2} \frac{\bar{u}_{i+1} - \bar{u}_{i-1}}{2\Delta x},$$

(3.2a)

$$u_{i-1/2}^+ = \bar{u}_i - \frac{\Delta x}{2} \frac{\bar{u}_{i+1} - \bar{u}_{i-1}}{2\Delta x}.$$  

(3.2b)

This reconstruction resulting from the slope between neighboring cells is the full (unlimited) second-order reconstruction, meaning that we obtain second-order accuracy for smooth solutions. However, this is a linear method and therefore Godunov’s theorem (2.1) states that when discontinuities are present, the scheme causes oscillations.
Denoting the slope in cell \( C_i \) by \( \sigma_i \), Eq. (3.2) can be rewritten as

\[
\begin{align*}
u_i^{(-)} &= \bar{u}_i + \frac{\Delta x}{2} \sigma_i, \\
u_i^{(+)} &= \bar{u}_i - \frac{\Delta x}{2} \sigma_i
\end{align*}
\]  

(3.3a) 

(3.3b)

with the slope \( \sigma_i = (\bar{u}_{i+1} - \bar{u}_{i-1}) / 2\Delta x \).

There is a variety of schemes which reconstruct based on three cell mean values and obtain 2\textsuperscript{nd}-order accuracy. These are presented in Sec. 3.1.1. The common factor of these limiter functions is that they can all be written in the form (3.3), \textit{i.e.} adding and subtracting the same slope from the cell mean value \( \bar{u}_i \) in order to get the left and right cell interface value. This does not hold true for all reconstructions. An example is depicted in Fig. 3.2. Here it is clear that the reconstructed values, obtained by evaluating the green function (resulting from a quadratic interpolation) at the cell interfaces, cannot be written in form of Eq. (3.3). These values cannot be derived by adding and subtracting the same value from \( \bar{u}_i \) as done before. Therefore, we need a more general formulation for the reconstruction.
Since we always consider three cells for the reconstruction, we define the left and right interface values to be determined by functions \( L \) and \( R \)

\[
\begin{align*}
  u_{i+\frac{1}{2}}^- &= L(\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}) \\
  u_{i-\frac{1}{2}}^+ &= R(\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}).
\end{align*}
\]  

(3.4a)  

(3.4b)

However, we will see later on in this chapter that the functions \( L \) and \( R \) do not need to be different. In [8, 17], the authors proposed the reconstruction to only depend on a function \( \phi \)

\[
\begin{align*}
  u_{i+1/2}^- &= \bar{u}_i + \frac{1}{2} \phi(\theta_i) \delta_{i+\frac{1}{2}}, \\
  u_{i-1/2}^+ &= \bar{u}_i - \frac{1}{2} \phi(\theta_i^{-1}) \delta_{i-\frac{1}{2}}.
\end{align*}
\]  

(3.5a)  

(3.5b)

Here, \( \phi \) is a limiter function depending on the local smoothness measure \( \theta_i \), given by

\[
\theta_i = \frac{\bar{u}_i - \bar{u}_{i-1}}{\bar{u}_{i+1} - \bar{u}_i}, \quad \equiv \frac{\delta_{i-\frac{1}{2}}}{\delta_{i+\frac{1}{2}}}
\]  

(3.6)

with the differences between neighboring cell averages

\[
\begin{align*}
  \delta_{i+\frac{1}{2}} &= \bar{u}_{i+1} - \bar{u}_i, \\
  \delta_{i-\frac{1}{2}} &= \bar{u}_i - \bar{u}_{i-1}.
\end{align*}
\]  

(3.7a)  

(3.7b)

These are also called undivided differences, an example is given in Fig. 3.3. The function \( \phi \) now fully characterizes the reconstruction.

Comparing the formulations (3.3) and (3.5) in the case of the full unlimited second-order reconstruction (3.2), we see that for \( \Delta x \sigma_i = (\bar{u}_{i+1} - \bar{u}_{i-1})/2 \) the constraints \( \Delta x \sigma_i = \frac{1}{2} \phi(\theta_i) \delta_{i+\frac{1}{2}} \) and \( \Delta x \sigma_i = \frac{1}{2} \phi(\theta_i^{-1}) \delta_{i-\frac{1}{2}} \) need to hold true. Indeed, the full second-order reconstruction can be rewritten in this form with the limiter function \( \phi(\theta_i) = \frac{1 + \theta_i}{2} \). This is true because for this reconstruction, the property \( \phi(\theta^{-1}) = \theta^{-1} \phi(\theta) \) is satisfied and therefore the two formulations (3.3) and (3.5) are equivalent. However, this does not hold true for all reconstructions. Therefore, the more general formulation Eq. (3.5) with \( \phi(\theta) \) is useful.
3.1.1 Classical Limiter Functions and Useful Properties

So far, the only reconstruction that has explicitly been states is the unlimited second order reconstruction (3.2). It can be seen in Fig. 3.4 that this reconstruction lies outside the total variation diminishing (TVD) region, visualized by the yellow shaded region. Thus, as expected, oscillations may occur when applying this reconstruction [33]. Godunov’s theorem (2.1) indicates that some non-linearity must be involved in the reconstruction process. This is where limiter functions come into play.

Limiter functions switch the reconstruction to high-order accuracy in smooth parts of the solution and to lower-order reconstructions near discontinuities in order to ensure that the total variation does not increase over time [33].

This type of function was first introduced by van Leer [69, 70, 71, 72, 73] and called MUSCL (Monotonic Upstream-centered Scheme for Conservation Laws) scheme. Other reconstructions within this framework have since been developed. Examples include the Minmod, Roe’s Superbee limiter [39], and van Leer limiter function, see [33] and references therein.
The limiter functions shown in Fig. 3.4 are given by

\begin{align*}
\text{Minmod:} & \quad \phi(\theta) = \max(0, \min(1, \theta)) \tag{3.8a} \\
\text{Superbee:} & \quad \phi(\theta) = \max(0, \min(2\theta, 1), \min(\theta, 2)) \tag{3.8b} \\
\text{van Leer:} & \quad \phi(\theta) = \frac{|\theta| + \theta}{|\theta| + 1} \tag{3.8c} \\
\text{Koren:} & \quad \phi(\theta) = \max \left( 0, \min \left( \frac{2\theta}{3}, 2 \right) \right). \tag{3.8d}
\end{align*}

Except for the full (unlimited) second-order reconstruction, the limiters depicted in Fig. 3.4 fulfill certain properties that are of interest for high-resolution numerical schemes.

**Property 3.1. General Properties of Limiter Functions**

\( (i) \) If \( \phi \) satisfies

\[
0 \leq \frac{\phi(\theta)}{\theta} \leq 2 \quad \text{and} \quad 0 \leq \phi(\theta) \leq 2 \quad \forall \theta \tag{3.9}
\]

the numerical scheme is TVD, and thus does not create spurious oscillations. This region is shown in Fig. 3.5a.
(ii) If $\phi$ passes continuously through $\theta = 1$ with $\phi(1) = 1$, then the resulting scheme is at least second order accurate in sufficiently smooth, monotonous regions of the solution.

(iii) If $\phi$ lies within the TVD region and additionally satisfies the conditions

\[
\begin{align*}
\theta & \leq \phi(\theta) \leq 1 & 0 & \leq \theta \leq 1 \\
1 & \leq \phi(\theta) \leq \theta & 1 & \leq \theta \leq 2 \\
1 & \leq \phi(\theta) \leq 2 & \theta & \geq 2 
\end{align*}
\]

the numerical scheme is second-order TVD \cite{55}. The region given by this restriction is also called Sweby region \cite{33}. Sweby realized that it is best that the limiters lie in the region spanned by the convex combination of Lax-Wendroff ($\phi(\theta) = 1$) and Beam-Warming ($\phi(\theta) = \theta$). This region is shown in Fig. 3.5b.

Proof. See e.g. \cite{34}.

A large number of limiter functions that have been designed lie within the Sweby region, see e.g. \cite{74}. In particular, the limiter functions given by Eq. \cite{3.8} all full the properties \cite{3.1}.
We have seen in this section that reconstructions within the MUSCL framework use only information of the cell $C_i$ and its immediate neighbors $C_{i-1}$ and $C_{i+1}$ for the reconstruction and obtain second-order accuracy. The restriction to immediate neighbor cells provides local update rules, we therefore want to remain in this setting. However, as shown in the following sections, three cell mean values are sufficient for reconstructions that yield third-order accurate schemes.

### 3.2 Third-Order Limiter

The aim of this work is to discuss schemes which use the three-point stencil to achieve 3rd order accurate reconstructions of the cell-interface values. One possibility is to construct a quadratic polynomial $p_i(x)$ in each cell $C_i$. Applying the computed polynomial to $x_i \pm 1/2$, we obtain the interface values

$$u^{(-)}_{i+1/2} = p_i(x_{i+1/2}) = \frac{1}{3}u_{i+1} + \frac{5}{6}u_i - \frac{1}{6}u_{i-1} \quad (3.13a)$$

$$u^{(+)}_{i-1/2} = p_i(x_{i-1/2}) = \frac{1}{3}u_{i-1} + \frac{5}{6}u_i - \frac{1}{6}u_{i+1}. \quad (3.13b)$$

It turns out that these expressions can be written in the form (3.5) to obtain the non-limited third-order reconstruction

$$\phi_3(\theta_i) := \frac{2 + \theta_i}{3} \quad (3.14)$$

with $\theta_i$ given by Eq. (3.6). This formulation results in a full-third-order-accurate scheme for smooth solutions, however, causes oscillations near discontinuities. This can be seen either by noticing that $\phi_3(\theta_i)$ does not lie in the TVD region, see Fig. 3.6, or as a direct consequence of Godunov’s theorem (2.1), since a linear scheme of more than first order cannot be monotone. Since oscillations should be avoided, limiter functions have been introduced, which apply the full third-order reconstruction (3.14) at smooth parts of the solution and switch to a lower-order reconstruction close to large gradients, shocks, and discontinuities. In [3], Artebrant and Schroll present a limiter function, which can be formulated as $\phi_{AS}(\theta_i, q)$, see [5], based on a local-double-logarithmic reconstruction. This function does not solely depend on $\theta_i$ but also contains a parameter $q$ which significantly changes the reconstruction function. The authors recommend $q = 33$. 

33
Figure 3.6 – Logarithmic limiter function $\phi_{AS}$ with $q = 1.4$ (black solid line), its approximation $\phi_{CT}$ (blue solid line), the TVD-version of $\phi_{CT}$ which does not have the non-zero part for $\theta < 0$, denoted by $\phi_{CT, TVD}$ (red solid line with red circles) and the full-third-order reconstruction $\phi_3$ (black dashed line).

1.4 and demonstrate that for $q \to 0$, the logarithmic limiter function reduces to $\phi_3(\theta_i)$. Their limiter function reads

$$
\phi_{AS}(\theta_i, q) = \frac{2p(\theta_i - 2p\theta_i + 1) \log(p) - (1 - \theta_i)(p^2 - 1)}{(p^2 - 1)(p - 1)^2},
$$

$$
p = p(\theta_i, q) = \frac{|\theta_i|^q}{1 + |\theta_i|^{2q}}.
$$

The downside of $\phi_{AS}(\theta_i, q)$ is its complexity, which renders the evaluation in each cell expensive. Čada and Torrilhon \[8\] derive, in an ad-hoc fashion, a limiter function $\phi_{CT}(\theta_i)$ which is based on $\phi_{AS}$. This function overcomes the drawbacks by approximating the properties of $\phi_{AS}$ and reducing the computational cost. It reads

$$
\phi_{CT}(\theta_i) = \max \left( 0, \min \left( \phi_3(\theta_i), \max \left( -\frac{1}{2} \theta_i, \min \left( 2 \theta_i, \phi_3(\theta_i), 1.6 \right) \right) \right) \right) \quad (3.15)
$$

and is shown in Fig. 3.6 together with $\phi_{AS}(\theta_i, 1.4)$ and $\phi_3(\theta_i)$. Note that $\phi_{CT}$ does not lie within the strict TVD bounds, i.e. it breaks with \[3.1\] (ii). The TVD property can be achieved by considering only those parts of $\phi_{CT}$ where $\theta \geq 0$, and setting it to 0 elsewhere, leading to the limiter function

$$
\phi_{CT, TVD}(\theta_i) = \max \left( 0, \min \left( 2\theta_i, \phi_3(\theta_i), 1.6 \right) \right). \quad (3.16)
$$
This situation is treated as a possible discontinuity: \( \phi_{CT} \neq \phi_3 \).

This situation is classified as smooth: \( \phi_{CT} = \phi_3 \).

Figure 3.7 – Two situations which are reflections of each other, treated differently by \( \phi_{CT} \).

However, the motivation for keeping the non-zero part in the construction of \( \phi_{CT}(\theta) \) for \( \theta \in [-2, 0] \) is to avoid the so-called extrema clipping. This is the effect occurring close to minima and maxima, where the normalized slopes \( \delta_{i \pm 1/2} \) are of the same order of magnitude but have opposite signs, i.e. \( \theta_i \approx -1 \). In this case, classical limiter functions that fully lie in the strict TVD bounds yield zero and thus generate a 1st-order accurate scheme. This undesirable reduction in accuracy is avoided when the non-zero part of \( \phi \) is included. Therefore, \( \phi_{CT} \) possesses better smoothness properties near \( \theta = -1 \) than \( \phi_{CT, TVD} \) since here, \( \phi_{CT}(\theta) = \phi_3(\theta) \) but \( \phi_{CT, TVD}(\theta) = 0 \). For more details, see [8], where the non-zero part was first introduced. In summary, \( \phi_{CT} \) breaks with the TVD property to avoid extrema clipping, thus improving classical TVD limiters by keeping high-order accuracy at extrema.

3.2.1 Missing Symmetry of \( \phi_{CT} \) and Definition of \( \phi_{3L} \)

Nevertheless, the limiter function \( \phi_{CT} \) contains one drawback, which is the missing symmetry in certain situations.

In order to fully understand and fix this issue, let us dwell upon an arbitrary but fixed situation \( \theta_s = \delta_L/\delta_R \), such as depicted in Fig. 3.7 with constants \( \delta_L, \delta_R \). In
Figure 3.8 – Comparison between $\phi_{CT}$, $\phi_3$, and the new limiter $\phi_{3L}$.

Some cases it might occur that

$$\phi_{CT} \left( \frac{\delta_L}{\delta_R} \right) = \phi_{CT} (\theta_s) = \phi_3 \left( \frac{\delta_L}{\delta_R} \right)$$

but

$$\phi_{CT} \left( -\frac{\delta_R}{-\delta_L} \right) = \phi_{CT} (\theta_s^{-1}) \neq \phi_3 \left( -\frac{\delta_R}{-\delta_L} \right).$$

This asymmetry is not natural, since these two situations are equally smooth (or non-smooth) and thus, should be considered as such.

We therefore present a new, symmetric limiter function which corrects this feature. This third-order limiter function, called $\phi_{3L}(\theta_i)$, is defined by

$$\phi_{3L}(\theta_i) = \max (0, \min (\phi_3(\theta_i), \max (-\theta_i, \min (2\theta_i, \phi_3(\theta_i), 1.5)))).$$  

(3.17)

The formulation seems very similar to $\phi_{CT}$, however, the differences are clearly visible in Fig. 3.8 which shows the new limiter in comparison to $\phi_{CT}$ and $\phi_3$.

**Definition 3.2. Symmetric Limiter**

A limiter function $\phi$ is called **symmetric** if it treats symmetric situations in a symmetric manner, i.e. if the limiter fulfills

$$\phi(\delta_L/\delta_R) = \phi_3(\delta_L/\delta_R) \iff \phi(-\delta_R/-\delta_L) = \phi_3(-\delta_R/-\delta_L)$$

or more general

$$\phi(\theta) = \phi_3(\theta) \iff \phi(\theta^{-1}) = \phi_3(\theta^{-1}).$$  

(3.18)
\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
$\delta_{i-\frac{1}{2}}$ & $\delta_{i+\frac{1}{2}}$ & $\theta_i$ & $\phi_{\text{CT,TVD}}(\theta_i)$ & $\phi_{\text{CT}}(\theta_i)$ & $\phi_{3L}(\theta_i)$ \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & $\leq 0$ & 0 & 0 & 0 & 0 \\
< 0 & 0 & $-\infty$ & 0 & 0 & 0 \\
> 0 & 0 & $+\infty$ & 1.6 & 1.6 & 1.5 \\
$\approx \delta_{i+\frac{1}{2}}$ & $\approx \delta_{i-\frac{1}{2}}$ & $\approx 1$ & $\phi_3(\theta_i)$ & $\phi_3(\theta_i)$ & $\phi_3(\theta_i)$ \\
$\approx -\delta_{i+\frac{1}{2}}$ & $\approx -\delta_{i-\frac{1}{2}}$ & $\approx -1$ & 0 & $\phi_3(\theta_i)$ & $\phi_3(\theta_i)$ \\
$\delta_1$ & $\delta_2$ & $\delta_1/\delta_2$ & $\phi_3(\delta_1, \delta_2)$ & $\phi_3(\delta_1, \delta_2)$ & $\phi_3(\delta_1, \delta_2)$ \\
$-\delta_2$ & $-\delta_1$ & $\delta_2/\delta_1$ & $\neq \phi_3(-\delta_2, -\delta_1)$ & $\neq \phi_3(-\delta_2, -\delta_1)$ & $\phi_3(-\delta_2, -\delta_1)$ \\
$\delta_1$ & $\delta_3$ & $\delta_1/\delta_3$ & 0 & $\phi_3(\delta_1, \delta_3)$ & $\phi_3(\delta_1, \delta_3)$ \\
$-\delta_3$ & $-\delta_1$ & $\delta_3/\delta_1$ & 0 & $\neq \phi_3(-\delta_3, -\delta_1)$ & $\phi_3(-\delta_3, -\delta_1)$ \\
\hline
\end{tabular}
\caption{Summary of situations, evaluated with $\phi_{\text{CT,TVD}}$, $\phi_{\text{CT}}$, and $\phi_{3L}$; $\delta_1, \delta_2 >$ and $\delta_3 < 0$ are constants s.t. the described situations occur.}
\end{table}

It is easy to verify that the new limiter $\phi_{3L}$ is symmetric.

A comparison of the different behaviors of $\phi_{\text{CT,TVD}}$, $\phi_{\text{CT}}$, and $\phi_{3L}$ in exemplary situations is summarized in Table 3.1.

For the better understanding of this important property, let us focus on the construction of a symmetric limiter. We split this task into two parts, $\theta_i < 0$ and $\theta_i \geq 0$, and propose a more general form of the limiter function, including three degrees of freedom $\alpha, \beta$, and $\gamma$

$$
\phi(\theta_i, \alpha, \beta, \gamma) = \max \left( 0, \min \left( \frac{2 + \theta_i}{3}, \max \left( -\alpha \theta_i, \min \left( \gamma \theta_i, \frac{2 + \theta_i}{3}, \beta \right) \right) \right) \right). 
$$

Ensuring symmetry for $\theta_i < 0$ boils down to first finding the ordinate $\theta_1$ where the full third-order reconstruction $\phi_3$ and the general limiter $\phi$ intersect, see Fig. 3.9, and then determining the value of $\alpha$. The intersection point $\theta_1$ can be found by
solving

$$\phi(\theta_1, \alpha, \beta, \gamma) = \frac{2 + \theta_1}{3}, \quad \theta_1 < 0$$

$$\Leftrightarrow -\alpha \theta_1 = \frac{2 + \theta_1}{3}$$

$$\Leftrightarrow \theta_1 = \frac{2}{-3\alpha - 1},$$

so that \(\phi(\theta, \alpha, \beta, \gamma) = \phi_3(\theta)\) needs to hold for \(\theta \in [-2, \theta_1] = [-2, \frac{2}{-3\alpha - 1}]\). This leads to

\[
\phi(-2, \alpha, \beta, \gamma) = \phi_3(-2) \quad \text{(minimum value for which this holds true)}
\]

\[
\Leftrightarrow \phi \left( -\frac{1}{2}, \alpha, \beta, \gamma \right) = \phi_3 \left( -\frac{1}{2} \right) = \frac{2 + \left( -\frac{1}{2} \right)}{3} = \frac{1}{2}
\]

\[
\Rightarrow -\alpha \cdot \left( -\frac{1}{2} \right) = \frac{1}{2}
\]

which leads to \(\alpha = 1\). This means, for the left part of the limiter there is no choice if we want it to fulfill Property (3.2).

For \(\theta_i \geq 0\), let us first find out the bounds \(\theta_2, \theta_3\) for which \(\phi = \phi_3\) needs to hold true.

$$\gamma \cdot \theta_2 = \frac{2 + \theta_2}{3} \quad \Leftrightarrow \quad \theta_2 = \frac{2}{3\gamma - 1} \quad \Rightarrow \quad \theta_3 = \frac{3\gamma - 1}{2}.$$  

The upper bound \(\theta_3\) finally leads to

$$\phi \left( \frac{3\gamma - 1}{2} \right) = \phi_3 \left( \frac{3\gamma - 1}{2} \right) = \frac{\gamma + 1}{2} = \beta.$$  

38
Therefore, we remain with one parameter, \(\gamma\), which yields a symmetric family of limiter functions

\[
\phi_{\gamma}(\theta_i) = \max \left(0, \min \left(\frac{2 + \theta_i}{3}, \max \left(-\theta_i, \min \left(\gamma \theta_i, \frac{2 + \theta_i}{3}, \frac{\gamma + 1}{2}\right)\right)\right)\right).
\] (3.20)

Choosing \(\gamma = 2\) leads to the limiter \(\phi_{3L}\), introduced and discussed above.

### 3.2.2 Asymptotic Region

Now we have fixed the symmetry and the so-called extrema clipping. Nonetheless, it might occur that the discretization of an extremum is such that one of the consecutive slopes is approximately zero. In this case, \(\theta \to 0\) or \(\theta \to \pm \infty\) and the interface values \(u_{i \pm 1/2}\) are again approximated by the cell mean values \(\bar{u}_i\), which yields a first-order scheme. That is, a zero-slope is interpreted as the onset of a discontinuity, even though it might in fact be the magnified view of a smooth extremum. This undesired case demonstrates that a criterion is needed which can differentiate a smooth extremum from a discontinuity or steep gradient. In the framework considered in this thesis, the criterion should only depend on the available information of the compact three-point stencil. Furthermore, it has to detect cases when switching to the third order reconstruction is safe, even though one of the normalized slopes is zero. We assume that using the third-order reconstruction is safe if the non-zero slope is 'small'. In turn, if the non-zero slope is not 'small', we assume to be near a discontinuity or large gradient, and the order should be reduced. The main focus of Section 3.2.3 is to determine what 'small' means and to define a suitable smoothness indicator \(\eta\).

### 3.2.3 Interpretation in 2D Slope Domain

From the discussion above, it is clear that such a switch function \(\eta\) has to explicitly depend on both normalized slopes \(\delta_{i \pm 1/2}\), Eq. (3.7a, 3.7b). The classical approach of only considering the ratio \(\theta_i\), Eq. (3.6), of neighboring slopes is overly restrictive because part of the information (the actual magnitude of the two slopes) is discarded. This is why we reformulate all limiter functions \(\phi\) in a two-parameter
framework and obtain the new formulation for the reconstructed interface values (see Eq. (3.5)) as

\begin{align}
u_{i+1/2}^{(-)} = \bar{u}_i + \frac{1}{2} H(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}), \\
u_{i+1/2}^{(+)} = \bar{u}_i - \frac{1}{2} H(\delta_{i+\frac{1}{2}}, \delta_{i-\frac{1}{2}}),
\end{align}

(3.21a, 3.21b)

with the limiter function \(H\) explicitly depending on both normalized slopes. The old limiter function \(\phi(\theta_i)\) can of course be rewritten in the new form of the two-parameter function \(H\) by setting

\[H(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}) := \phi(\theta_i)\delta_{i+\frac{1}{2}} = \phi\left(\frac{\delta_{i-\frac{1}{2}}}{\delta_{i+\frac{1}{2}}}\right)\delta_{i+\frac{1}{2}}.\]

(3.22)

In this setting, the full-third-order reconstruction \(\phi_3(\theta_i) = (2 + \theta_i)/3\), given by Eq. (3.14), now reads

\[H_3(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}) = \frac{2\delta_{i+\frac{1}{2}} + \delta_{i-\frac{1}{2}}}{3}.\]

(3.23)

This formulation has the advantage that there is no division by the normalized slope \(\delta_{i\pm1/2}\). Thus, a possible division by a number close to zero is avoided.

The limiter developed in [8], Eq. (3.15) can also be written in the two-parameter framework, now reading

\[H_{CT}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}) = \text{sgn}(\delta_{i+\frac{1}{2}}) \max(0, \min(\text{sgn}(\delta_{i+\frac{1}{2}}) H_3, \max(-\frac{1}{2} \text{sgn}(\delta_{i+\frac{1}{2}})\delta_{i-\frac{1}{2}}, \min(2 \text{sgn}(\delta_{i+\frac{1}{2}}\delta_{i-\frac{1}{2}}, \text{sgn}(\delta_{i+\frac{1}{2}}) H_3, 1.6|\delta_{i+\frac{1}{2}}|))) \quad (3.24)\]

Fig. 3.10a shows the limiter function \(H_{CT,TVD}\) which satisfies the strict TVD bounds. This can clearly be seen by the zero parts for \(\text{sgn}(\delta_{i-\frac{1}{2}}) \neq \text{sgn}(\delta_{i+\frac{1}{2}})\).

Fig. 3.10b shows the extended version \(H_{CT}\) in the two-parameter setting, Eq. (3.24). On the coordinate axis where \(\delta_{i-\frac{1}{2}} = 0\), i.e. \(\theta_i = 0\), the limiter function \(H_{CT}\) returns zero, meaning that it yields a 1st-order method. The same holds for the coordinate axis where \(\delta_{i+\frac{1}{2}} = 0\). For two consecutive slopes of approximately the same order of magnitude, i.e. around the diagonals, \(H_{CT}\) recovers the third-order reconstruction \(H_3\). This is the case for \(\delta_{i+\frac{1}{2}} \approx -\delta_{i-\frac{1}{2}}\) as well as for \(\delta_{i-\frac{1}{2}} \approx \delta_{i+\frac{1}{2}}\), contrary to \(H_{CT,TVD}\) which returns \(H_3\) only in the latter case and
0 for $\delta_{i-\frac{1}{2}} \approx -\delta_{i+\frac{1}{2}}$. These observations are also summarized in Table 3.1. The symmetry discussion from Sec. 3.2.1 translates into this setting by noting that $H_{CT}$ is not symmetric with respect to the diagonals. This asymmetry is clearly visible in Fig. 3.10b.

The drawback has been cured by presenting the new symmetric third-order limiter, in this setting called $H_{3L}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}})$. It reads

$$H_{3L}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}) = \text{sgn}(\delta_{i+\frac{1}{2}}) \max(0, \min(\text{sgn}(\delta_{i+\frac{1}{2}}) H_3, \max(-\text{sgn}(\delta_{i+\frac{1}{2}}) \delta_{i-\frac{1}{2}}, \min(2 \text{sgn}(\delta_{i+\frac{1}{2}}) \delta_{i-\frac{1}{2}}, \text{sgn}(\delta_{i+\frac{1}{2}}) H_3, 1.5|\delta_{i+\frac{1}{2}}|))).$$

(3.25)

As discussed above and summarized in Table 3.1 $H_{3L}$ treats reflected situations in a symmetric way, i.e.

$$H_{3L}(\delta_L, \delta_R) = H_3(\delta_L, \delta_R) \quad \Leftrightarrow \quad H_{3L}(-\delta_R, -\delta_L) = H_3(-\delta_R, -\delta_L).$$

(3.26)

3.2.3.1 Properties of the Limiter Functions

Expressing the interface values in the most general form

$$u_{i+\frac{1}{2}}^{-} = L(\bar{u}_{i-1}, \bar{u}_{i}, \bar{u}_{i+1})$$

$$u_{i-\frac{1}{2}}^{+} = R(\bar{u}_{i-1}, \bar{u}_{i}, \bar{u}_{i+1}).$$
we can derive some properties for $L(\cdot, \cdot, \cdot)$ and $R(\cdot, \cdot, \cdot)$, following the concepts of [17]. These properties are valid for all limiter functions presented so far, i.e. for $H_3$, $H_{CT}$, $H_{3L}$.

**Property 3.3. Homogeneity**

Multiplying the arguments of $L$ and $R$ in Eq. (3.4) by the same real number $\lambda$ multiplies the interface values $u_{i+\frac{1}{2}}^{(-)}$ and $u_{i+\frac{1}{2}}^{(+)}$, respectively, by the same constant $\lambda$. This is, $L$ and $R$ are called homogeneous, i.e. linear along each line through the origin in the $(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}})$ plane.

$$J(\lambda u, \lambda v, \lambda w) = \lambda J(u, v, w), \quad J \in \{L, R\}, \lambda \in \mathbb{R} \quad (3.27)$$

**Property 3.4. Translational invariance**

Adding a constant $\lambda$ to the arguments of (3.4) adds the same constant $\lambda$ to the interface values. This is, $L$ and $R$ are called translationally invariant.

$$J(u + \lambda, v + \lambda, w + \lambda) = J(u, v, w) + \lambda, \quad J \in \{L, R\}, \lambda \in \mathbb{R} \quad (3.28)$$

**Property 3.5. Left-Right symmetry**

Exchanging the first and third argument of (3.4) interchanges the left and right interface values, (cf. [17] for more details and figures)

$$R(w, v, u) = L(u, v, w). \quad (3.29)$$

**Lemma 3.6.**

If properties 3.3 to 3.5 are satisfied, there exists an appropriate limiter function $\psi: \mathbb{R} \to \mathbb{R}$ such that

$$L(u, v, w) = v + \frac{1}{2} \psi \left( \frac{u - v}{w - v} \right) (w - v) \quad (3.30)$$

$$R(u, v, w) = v - \frac{1}{2} \psi \left( \frac{w - v}{u - v} \right) (v - u). \quad (3.31)$$

**Lemma 3.7.** With properties 3.3 to 3.5, it is easy to verify that

(i) computing the interface values $u_{i+1/2}^{(+)}$ with $L$ and $R$, given by Eq. (3.4), or in the form of Eq. (3.5), are equivalent.
(ii) For $\delta_{i+\frac{1}{2}} \neq 0$ the formulations of the cell interfaces in the one-parameter framework, Eq. (3.5), and in the two-parameter framework, Eq. (3.21), are equivalent.

**Proof.** We will only show $(i)$ for $u_{i+1/2}^(-)$; the other cases are similar.

(i) Setting $(u, v, w) = (\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1})$ yields

\[
\begin{align*}
u_{i+1/2}^{(-)} & \quad L(\bar{u}_{i-1}, \bar{u}_i, \bar{u}_{i+1}) \quad u_{i+1} + \frac{1}{2} \psi\left(\frac{\bar{u}_{i-1} - \bar{u}_i}{\bar{u}_{i+1} - \bar{u}_i}\right) (\bar{u}_{i+1} - \bar{u}_i) \\
& = \bar{u}_i + \frac{1}{2} \phi(\theta_i) \delta_{i+\frac{1}{2}}
\end{align*}
\]

where $\delta_{i+\frac{1}{2}} = \bar{u}_{i+1} - \bar{u}_i$ and $\psi(\bar{u}_{i-1} - \bar{u}_i/\bar{u}_{i+1} - \bar{u}_i) = \psi(-\theta_i) =: \phi(\theta_i)$ applying Eq. (3.6).

(ii) Due to the homogeneity of $L$ and $R$, we can easily see with Eq. (3.21) that

\[
H\left(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}\right) = H\left(\frac{\delta_{i-\frac{1}{2}}}{\delta_{i+\frac{1}{2}}}, 1\right) \quad \delta_{i+\frac{1}{2}} = H(\theta_i, 1) \quad \delta_{i+\frac{1}{2}} = \phi(\theta_i) \delta_{i+\frac{1}{2}}.
\]

\[\square\]

### 3.2.4 A New Smoothness Indicator

Given three cell averages, it is in general not possible to determine whether these points represent the onset of a discontinuity or the magnified view of an extremum. An example is shown in Fig. 3.11 where one set of cell mean values could
be obtained by the smooth function (green, square markers) or the discontinuous function (red, diamond markers). As stated in Sec. 3.2.3, the two-parameter setting is the necessary prerequisite for the definition of such a criterion. Čada and Torrilhon [8] proposed the function

\[
\eta_{CT}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}) = \frac{\delta_{i-\frac{1}{2}}^2 + \delta_{i+\frac{1}{2}}^2}{(r\Delta x)^2}. \tag{3.32}
\]

This switch function defines an asymptotic region of radius \(r\) around the origin in the \((\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}})\)-plane. Within this region the limiter switches to the third-order reconstruction.

The authors of [8] then modified the structure of the limiter function \(\phi_{CT}\) to include the asymptotic region around the origin. The combination, denoted by the superscript \((c)\), is defined as

\[
\phi^{(c)}_{CT}(\theta_i) := \begin{cases} 
\phi_3(\theta_i) & \text{if } \eta_{CT} < 1 \\
\phi_{CT}(\theta_i) & \text{if } \eta_{CT} \geq 1.
\end{cases} \tag{3.33}
\]

It is possible to make this function Lipschitz continuous, by introducing a small transition region and a linear function, cf. [8] for more details. This limiter function, combining \(\phi_{CT}\) and the switch function \(\eta_{CT}\), has been successfully employed in e.g. [27, 28, 36]. In [8], the authors did not provide a general formulation of the parameter \(r\) which determines the size of the asymptotic region. Instead it was chosen ad hoc, in a problem-specific way. To obtain some generic idea about suitable choices for \(r\), we conduct a numerical test, applying \(\phi^{(c)}_{CT}\) to the advection equation \(u_t + u_x = 0\) with smooth initial condition \(u_0(x) = \sin(\pi x)\) for different values of \(r\). Fig. 3.12 shows the double logarithmic plot of the \(L_1\)- and \(L_{\infty}\)-errors versus the number of grid cells of the solution advected until \(t_{end} = 1\) with CFL number \(\nu = 0.9\). For this smooth test case, we see that larger values of \(r\), corresponding to larger asymptotic regions are favorable. For smooth solution this makes sense because increasing values of \(r\) corresponds to increasing the region of directly applying the full-third-order reconstruction \(\phi_3\). From Fig. 3.12 we can deduce that for smaller \(r\), a finer space discretization is needed to obtain third-order accuracy.

Motivated by the limiter function of [8], we want to define a new smoothness
Figure 3.12 – Double logarithmic plot of the $L_1$- and $L_\infty$-errors versus the number of grid cells of the solution obtained with $\phi^{(c)}_{\text{CT}}$ advected until $t_{\text{end}} = 1$ with CFL number $\nu = 0.9$. The numerical solutions are shown for different values of $r$, see Eq. (3.32).

indicator without artificial parameters. A promising potential to distinguish discontinuities from smooth extrema is by measuring the magnitude of the vector $(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}})$. If this vector is sufficiently small in some appropriate norm, the reconstruction is switched to the full-third-order reconstruction, even if one of the lateral derivatives may be vanishing.

Lemma 3.8. In the vicinity of an extremum $\xi_0$, for $|x_i - \xi_0| \leq \Delta x$, the following relations hold for each time $t^n$:

\[
\left\| \left( \delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}} \right) \right\|_2 \leq \sqrt{c} \max_{x \in \Omega \setminus \Omega_d} |u''(x, t^n)| \Delta x^2 \quad \text{with} \quad c = \frac{5}{2} + O(\Delta x), \quad (3.34a) \\
\left\| \left( \delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}} \right) \right\|_1 \leq c \max_{x \in \Omega \setminus \Omega_d} |u''(x, t^n)| \Delta x^2 \quad \text{with} \quad c = 2 + O(\Delta x). \quad (3.34b)
\]

Here, $\Omega$ is the computational domain, and $\Omega_d$ is a set of points where the solution is discontinuous.

Proof. Let us recall the definition of $\bar{U}(x, t)$, Eq. (2.7),

\[ \bar{U}(x, t) = \frac{1}{\Delta x} \int_{x-\Delta x/2}^{x+\Delta x/2} u(s, t)\mathrm{d}s, \]
where $u$ is the exact solution. The following properties hold for $\bar{U}(x, t)$

$$
\bar{U}'(x, t) = \frac{u(x + \frac{\Delta x}{2}, t) - u(x - \frac{\Delta x}{2}, t)}{\Delta x},
$$

$$
\bar{U}''(x, t) = \frac{u'(x + \frac{\Delta x}{2}, t) - u'(x - \frac{\Delta x}{2}, t)}{\Delta x},
$$

(3.35)

where $\bar{U}(x, t^n)$ is the cell average of the exact solution at time $t^n$ in cell $C_i$ with cell center $x_i$.

Eq. (3.34a) can be proven regarding the following formulation of definitions (3.7a), (3.7b), with a constant $\alpha$ which is going to be specified. For the sake of simplicity we shall neglect $t^n$ for the rest of the proof.

$$
\frac{(\bar{U}(x_i + \Delta x) - \bar{U}(x_i))^2 + (\bar{U}(x_i) - \bar{U}(x_i - \Delta x))^2}{(\alpha \Delta x^2)^2} +
$$

$$
= \frac{1}{\alpha^2} \left( \bar{U}(x_i + \Delta x) - 2\bar{U}(x_i) + \bar{U}(x_i - \Delta x) \right)^2 +
$$

$$
+ \frac{2}{\alpha^2 \Delta x^2} \left( \bar{U}(x_i + \Delta x) - \bar{U}(x_i) \right) \left( \bar{U}(x_i) - \bar{U}(x_i - \Delta x) \right)
$$

(3.36)

a Taylor expansion of the functions $\bar{U}(x_i \pm \Delta x)$ around $x_i$ yields

$$
= \frac{1}{2} \left( \frac{\bar{U}''(x_i)}{\alpha} \right)^2 + \frac{2}{\Delta x^2} \left( \frac{\bar{U}'(x_i)}{\alpha} \right)^2 + \frac{2}{3} \frac{\bar{U}'(x_i)\bar{U}'''(x_i)}{\alpha^2} + O(\Delta x^2).
$$

In the vicinity of an extremum $\xi_0$, for $|x_i - \xi_0| \leq \Delta x$, the derivative satisfies

$$
\bar{U}'(x_i) = \bar{U}'(\xi_0) + \bar{U}''(\xi_0)(x_i - \xi_0) + O((x_i - \xi_0)^2)
$$

$$
= \bar{U}''(\xi_0)(x_i - \xi_0) + O(\Delta x^2)
$$

$$
\Rightarrow |\bar{U}'(x_i)| \leq |\bar{U}''(\xi_0)|\Delta x + O(\Delta x^2).
$$

Since we are interested in $(\bar{U}'(x_i))^2$, we find that

$$
(\bar{U}'(x_i))^2 \leq (\bar{U}''(\xi_0))^2 \Delta x^2 + O(\Delta x^3).
$$

Therefore, Eq. (3.36) reduces to

$$
\frac{(\bar{U}(x_i + \Delta x) - \bar{U}(x_i))^2 + (\bar{U}(x_i) - \bar{U}(x_i - \Delta x))^2}{(\alpha \Delta x^2)^2} \leq \frac{1}{2} \left( \frac{\bar{U}''(x_i)}{\alpha} \right)^2 + 2 \left( \frac{\bar{U}''(\xi_0)}{\alpha} \right)^2 + O(\Delta x).
$$

(3.37)
Consider the computational domain $\Omega$, and the set of points $\Omega_d$, where the solution is discontinuous. Setting $\alpha \equiv \max_{x_i \in \Omega \setminus \Omega_d} |\bar{U}''(x_i, t^n)|$ with the cell centers $x_i$, leads to

$$\frac{(\bar{U}(x_i + \Delta x) - \bar{U}(x_i))^2 + (\bar{U}(x_i) - \bar{U}(x_i - \Delta x))^2}{(\max_{x_i} |\bar{U}''(x_i, t^n)|\Delta x^2)^2} \leq \frac{5}{2} + O(\Delta x). \quad (3.38)$$

Since the numerical solution $\bar{u}_i^n$ is a third-order-accurate approximation of the true solution, i.e. $\bar{u}_i^n = \bar{U}(x_i, t^n) + O(\Delta x^3)$ and

$$\alpha = \max_{x_i} [\bar{U}''(x_i, t^n)] \leq \max_{x_i} \max_{\xi \in (x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2})} |u''(\xi, t)|$$

holds true, this shows Eq. (3.34a). In a similar manner, Eq. (3.34b) can be proven.

**Remark 3.9.** Often, the exact value of $\max_x |u''(x, t^n)|$ is not known or it is too expensive to compute. In these cases, a different estimator needs to be found.

(a) In many applications, one has some estimate of the largest second derivative of the solution, even if one does not know the solution itself.

(b) If a good estimate of the solution at time $t^n$ is unavailable, one can use the initial conditions $u_0(x)$ as an approximation of $\alpha$. Note that for a conservation law of the form $u_t + f(u)_x = 0$, certain information about $u_{xx}$ can in fact be inferred from the initial data (where the solution is smooth). For instance, second derivatives are actually preserved at extremal points. To see this, consider the equations that $u_x$ and $u_{xx}$ satisfy, namely $(u_x)_t + f'(u)(u_x)_x = -f''(u)(u_x)^2$ and $(u_{xx})_t + f'(u)(u_{xx})_x = -f''''(u)u_x^2 - 3f''(u)u_xu_{xx}$. In both equations, the left hand side represents the convective derivative along characteristics. Therefore, extremal points ($u_x = 0$) are just moved with the characteristics, and $u_{xx}$ remains unchanged from its value at initial time.

From now on, we will use $\alpha \equiv \max_{x \in \Omega \setminus \Omega_d} |u_0''(x)|$.

Lemma 3.8 makes a statement on the magnitude of the normalized slopes. Note that the upper bound only depends on the grid size $\Delta x$ and the initial condition $u_0$. From this lemma we can now deduce the definition of the smoothness indicator.
Definition 3.10. The switch function $\eta$ which marks the limit between smooth extrema and discontinuities is defined by

$$
\eta = \eta(\delta_{i-1/2}, \delta_{i+1/2}) = \sqrt{\frac{\delta_{i-1/2}^2 + \delta_{i+1/2}^2}{\sqrt{\frac{5}{2}} \alpha \Delta x^2}}
$$

with

$$
\alpha \equiv \max_{x \in \Omega \setminus \Omega_d} |u_0''(x)|
$$

$\Omega$ and $\Omega_d$ defined as in Lemma 3.8.

Note that $\alpha$ is proportional to $1/\Delta x^2$ and therefore, $\eta$ is a non-dimensional quantity, so that the overall scheme is not affected by changes of units.

The idea of using the largest second derivative of the initial conditions to relax limiting near extrema has already been proposed by Cockburn and Shu [12] in the context of discontinuous Galerkin methods. They use the constant $M_2 = \max_x |u_0''(x)|$ to overcome the degeneration to first order at critical points: $M_2$ is used to estimate the magnitude of the reconstructions $u_{i\pm1/2}^\mp, u_{i\pm1/2}^\pm$. If the reconstruction is smaller than a certain bound, the high order reconstruction is used, otherwise, a limiter function is applied. Note that in [12] the switch function is based on the reconstructed values $u_{i\pm1/2}^\mp, u_{i\pm1/2}^\pm$, whereas in this work, we consider the normalized slopes $\delta_{i\pm1/2}$.

With Def. 3.10, Lemma 3.8 states that in the vicinity of smooth extrema, $\eta \leq 1$ holds. Combining this information with the new limiter function $H_{3L}$, we use this result to define the combined limiter, denoted by the superscript $(c)$,

$$
H_{3L}^{(c)}(\delta_{i-1/2}, \delta_{i+1/2}) := \begin{cases} 
H_3(\delta_{i-1/2}, \delta_{i+1/2}) & \text{if } \eta < 1 \\
H_{3L}(\delta_{i-1/2}, \delta_{i+1/2}) & \text{if } \eta \geq 1.
\end{cases}
$$

The resulting two-dimensional limiter function for this approach is shown in Fig. 3.13b. Note that in the same manner as for $H_{CT}^{(c)}$, Lipschitz-continuity of the combined limiter $H_{3L}^{(c)}(\delta_{i-1/2}, \delta_{i+1/2})$, Eq. (3.41), can be achieved by using a continuous switch function for the transition between the limited and non-limited reconstruction.
Remark 3.11. The proposed limiter $H_{3L}^{(c)}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}})$ still satisfies properties 3.3 to 3.5, i.e. it is homogeneous, translationally invariant and fulfills the left-right symmetry. For the homogeneity, this can be seen by the fact that the stretched vector $(k u_{i-1}, k u_i, k u_{i+1})$ leads to the undivided differences $(k \delta_{i-\frac{1}{2}}, k \delta_{i+\frac{1}{2}})$, which are stretched by the same factor. Also, observe how $(k u_{i-1}, k u_i, k u_{i+1})$ affects $\alpha$, Eq. (3.40),

$$\alpha(k u_0) = \max_{x \in \Omega \setminus \Omega_d} |k u_0'(x)| = |k| \max_{x \in \Omega \setminus \Omega_d} |u_0''(x)| = |k| \alpha(u_0). \quad (3.42a)$$

With definition (3.39), we obtain that the decision criterion is homogeneous, i.e.,

$$\eta(k \delta_{i-\frac{1}{2}}, k \delta_{i+\frac{1}{2}}) = \eta(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}). \quad (3.42b)$$

Using the fact that $H_3(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}})$ and $H_{3L}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}})$ are homogeneous functions, finally leads to

$$H_{3L}^{(c)}(k \delta_{i-\frac{1}{2}}, k \delta_{i+\frac{1}{2}}) = k H_3^{(c)}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}). \quad (3.42c)$$

For the translational invariance, note that the shifted vector $(k + u_{i-1}, k + u_i, k + u_{i+1})$ leads to the original formulation $(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}})$ and does not affect

---

**Figure 3.13** – Limiter functions combined with the full-third-order reconstruction in a region around the origin.
\( \alpha \) either. Contrary to \( H_{3L}^{(c)} \), the limiter function \( H_{CT}^{(c)}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}) \) is not homogeneous, because \( \eta_{CT}(k\delta_{i-\frac{1}{2}}, k\delta_{i+\frac{1}{2}}) = k^2 \eta_{CT}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}) \neq \eta_{CT}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}) \), compare to Eq. (3.42b).

### 3.3 WENO

In this section we briefly review the third-order weighted essentially non-oscillatory (WENO) scheme for one-dimensional scalar conservation laws. We consider WENO schemes because they represent a very popular method for approximating the solution of hyperbolic conservation laws. Since we want to compare WENO with the limiter functions introduced in Section 3.2, we focus on WENO3. This is the family of schemes which only use three points for the reconstruction of the left and right cell-interfaces \( u^{(+)}_{i-1/2} \) and \( u^{(-)}_{i+1/2} \). WENO3 reconstructs an estimate of the solution \( u(x_{i+\frac{1}{2}}) \) from the cell averages \( \bar{u}_{i-1}, \bar{u}_i \), and \( \bar{u}_{i+1} \). The procedure, introduced by Jiang and Shu [26], is to start just as for the \( r \)-th order ENO scheme [24], where \( r = 2 \). This is, we consider the two-point stencils \( S_0 = C_{i-1} \cup C_i \) and \( S_1 = C_i \cup C_{i+1} \) and define on each stencil second-order accurate approximations:

\[
\begin{align*}
    p_0(x_{i+\frac{1}{2}}) &= -\frac{1}{2}u_{i-1} + \frac{3}{2}u_i, \quad \text{where } p_0 \text{ is the linear interpolation of } (x_{i-1}, \bar{u}_{i-1}) \quad \text{and } (x_i, \bar{u}_i) \quad \text{and} \\
    p_1(x_{i+\frac{1}{2}}) &= \frac{1}{2}u_i + \frac{1}{2}u_{i+1}, \quad \text{where } p_1 \text{ is the linear interpolation of } (x_i, \bar{u}_i) \quad \text{and} \quad (x_{i+1}, \bar{u}_{i+1}).
\end{align*}
\]

For the sake of simplicity, we only consider the procedure for the right cell interface \( x_{i+\frac{1}{2}} \). The left interface follows in a similar manner.

Based on an \( r \)-th order ENO scheme, the best one can get is a \( (2r-1) \)-th order WENO scheme, \textit{i.e.} in our case third order [26]. To obtain this, the WENO3 estimate of \( u(x_{i+\frac{1}{2}}) \), called \( \hat{u}_{i+\frac{1}{2}} \), is defined as a convex combination of the two second-order estimates \( p_0(x_{i-\frac{1}{2}}) \) and \( p_1(x_{i+\frac{1}{2}}) \):

\[
\hat{u}_{i+\frac{1}{2}} = w_{i-\frac{1}{2}}p_0(x_{i+\frac{1}{2}}) + w_{i+\frac{1}{2}}p_1(x_{i+\frac{1}{2}}).
\]

The weights satisfy \( w_{i+\frac{1}{2}} \geq 0 \) and \( w_{i-\frac{1}{2}} + w_{i+\frac{1}{2}} = 1 \). There is a particular choice of weights that generates a third-order-accurate approximation to \( u(x_{i+\frac{1}{2}}) \), namely
\[ w_{i-\frac{1}{2}} = \gamma_{i-\frac{1}{2}} = \frac{1}{3}, \text{ and } w_{i+\frac{1}{2}} = \gamma_{i+\frac{1}{2}} = \frac{2}{3}. \] This approximation is obtained when interpolating \((x_{i-1}, \bar{u}_{i-1}), (x_i, \bar{u}_i), \) and \((x_{i+1}, \bar{u}_{i+1})\) by a quadratic polynomial, and evaluating it at \(x_{i+\frac{1}{2}}\), see Eq. (3.13a).

The philosophy of WENO is to design formulas for the weights \(w_{i\pm\frac{1}{2}}\), such that in smooth regions, one has \(w_{i\pm\frac{1}{2}} \approx \gamma_{i\pm\frac{1}{2}}\), while in regions of large gradients, more weight is given to the approximation that generates fewer spurious oscillations. This is achieved using smoothness indicators \(\beta_{i\pm\frac{1}{2}}\). They are defined using the normalized slopes between neighboring cells \(\delta_{i\pm\frac{1}{2}}\):

\[ \beta_{i\pm\frac{1}{2}} = \left(\delta_{i\pm\frac{1}{2}}\right)^2. \]

The final weights \(w_{i\pm\frac{1}{2}}\) are defined by

\[ w_{i\pm\frac{1}{2}} = \frac{\alpha_{i\pm\frac{1}{2}}}{\alpha_{i-\frac{1}{2}} + \alpha_{i+\frac{1}{2}}}, \tag{3.44} \]

where

\[ \alpha_{i\pm\frac{1}{2}} = \frac{\gamma_{i\pm\frac{1}{2}}}{(\varepsilon + \beta_{i\pm\frac{1}{2}})^p}. \tag{3.45} \]

In equation (3.45), there are two parameters which need to be further detailed. The integer \(p \in \mathbb{N}\), which Liu et al. [35] suggest to set \(p = r\), the order of the base ENO scheme. Jiang and Shu [26] claim that for \(r = 2, 3\), setting \(p\) to 2 is adequate. In this work, we will set \(p = 2\).

The other parameter in Eq. (3.45) is \(\varepsilon\), a small positive number, originally introduced to avoid the division by zero [26]. We suggest that rather than fixing \(\varepsilon\) to some constant, it should depend on the spacial discretization \(\Delta x\). This will be discussed in more detail in Sec. 3.3.2.

### 3.3.1 Interpretation of WENO3 in 2D Slope Domain

A brief calculation yields for the WENO3 reconstruction

\[ \hat{u}_{i+\frac{1}{2}} = w_{i-\frac{1}{2}} p_0(x_{i+\frac{1}{2}}) + w_{i+\frac{1}{2}} p_1(x_{i+\frac{1}{2}}) = w_{i-\frac{1}{2}} \left( -\frac{1}{2} \bar{u}_{i-1} + \frac{3}{2} \bar{u}_i \right) + w_{i+\frac{1}{2}} \left( \frac{1}{2} \bar{u}_i + \frac{1}{2} \bar{u}_{i+1} \right) = \bar{u}_i + \frac{1}{2} \left( w_{i-\frac{1}{2}} (\bar{u}_{i-1} - \bar{u}_i) + w_{i+\frac{1}{2}} (\bar{u}_{i+1} - \bar{u}_i) \right) = \bar{u}_i + \frac{1}{2} \left( w_{i-\frac{1}{2}} \delta_{i-\frac{1}{2}} + w_{i+\frac{1}{2}} \delta_{i+\frac{1}{2}} \right). \]
Since the weights $w_{i\mp 1/2}$ themselves only depend on the slopes $\delta_{i\mp 1/2}$, the WENO3 estimate can be written in the form (3.21) with the limiter function $H_{\text{WENO3}}$, whose particular form depends on the parameters $p$ and $\varepsilon$. Explicitly written, it is

$$H_{\text{WENO3}}(\delta_{i-1/2}, \delta_{i+1/2}) = \frac{1}{3}(\delta_{i-1/2} + \delta_{i+1/2} - \delta_{i-1/2})^{1-2p} + \frac{2}{3}(\delta_{i+1/2} - \delta_{i-1/2})^{1-2p} \cdot (3.46)$$

In the vicinity of $(0, 0)$, i.e. for $|\delta_{i\mp 1/2}| \ll \varepsilon$, one has in leading order that

$$H \ll (\delta_{i-1/2}, \delta_{i+1/2}) = \frac{1}{3}\delta_{i-1/2} + \frac{2}{3}\delta_{i+1/2}. \quad (3.47)$$

As mentioned before, this linear function is homogeneous and results in a third-order-accurate approximation. On the other side of the spectrum, if $|\delta_{i\mp 1/2}| \gg \varepsilon$, one has in leading order that

$$H \gg (\delta_{i-1/2}, \delta_{i+1/2}) = \frac{1}{3}(\delta_{i-1/2})^{1-2p} + \frac{2}{3}(\delta_{i+1/2})^{1-2p} \quad (3.48)$$

This function is also homogeneous, i.e. $H_{\gg}(k\delta_{i-1/2}, k\delta_{i+1/2}) = k H_{\gg}(\delta_{i-1/2}, \delta_{i+1/2})$. That means, along each line through the origin in the $(\delta_{i-1/2}, \delta_{i+1/2})$ plane, it is linear, thus resembling the behavior of traditional finite volume limiters.

### 3.3.2 WENO Smoothness Indicators

The choice of the weights $\omega_{i\mp 1/2}$, which depend on $\alpha_{i\mp 1/2}$, is crucial for the order of accuracy of the resulting scheme. Its precise value influences the behavior of the limiting when $\beta_{i\mp 1/2}$ is close to zero. Clearly, this is of particular importance near extremal points of the solution. To point out the importance of $\varepsilon$, we repeated the numerical experiment conducted in Sec. 3.2.4 with the third-order WENO scheme. This is, we apply $H_{\text{WENO3}}$ to the advection equation $u_t + u_x = 0$ with smooth initial condition $u_0(x) = \sin(\pi x)$ for different values of $\varepsilon$. The result for $t_{\text{end}} = 1$ and CFL number $\nu = 0.9$ is depicted in Fig. 3.14. There is a strong similarity between this test case and the one shown in Fig. 3.12. This resemblance and also comparing the form of $\alpha_{i\mp 1/2}$, Eq. (3.45), with the new finite volume smoothness indicator $\eta$, Eq. (3.39), strongly suggests to take a closer look at $\varepsilon$. 

52
In recent years, this parameter has attracted a lot of attention, see e.g. [78], [1], [2], [29] and references therein. Originally, \(\varepsilon\) was introduced by Jiang and Shu to avoid the denominator to become zero [26]. The authors called it a small positive number and set to \(\varepsilon = 10^{-6}\) for their test-case studies. Therefore, WENO3 with the fixed value \(\varepsilon = 10^{-6}\) will be called WENO-JS henceforth. There are several drawbacks by fixing \(\varepsilon\) to some value \(\varepsilon_0\). One of them is the following scenario which might occur since \(\delta_{i\pm\frac{1}{2}} = \mathcal{O}(\Delta x)\) in smooth parts of the solution:

1. For large grid sizes, \(|\delta_{i\pm\frac{1}{2}}| \gg \varepsilon_0\) holds, leading to the homogeneous function \(H_{\text{WENO3}} \rightarrow H_{\gg}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}})\), Eq. (3.48), and thus to low order.

2. Refining the grid leads to \(|\delta_{i\pm\frac{1}{2}}| \ll \varepsilon_0\), and yields \(H_{\text{WENO3}} \rightarrow H_{\ll}(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}})\), Eq. (3.47), which is the full-third-order reconstruction.

This phenomenon is also demonstrated in Fig. 3.14. Wanting \(\varepsilon\) to be adaptive for all grid discretizations means that it has to depend on the grid size, i.e. \(\varepsilon = \varepsilon(\Delta x)\). This is also what Yamaleev and Carpenter claim in [77, 78]. However, the authors do not simply replace \(\varepsilon\) by a function depending on \(\Delta x\), they rather define new

---

**Figure 3.14** – Double logarithmic plot of the \(L_1\)- and \(L_\infty\)-errors versus number of grid cells of the solution obtained with \(H_{\text{WENO3}}\) advected until \(t_{\text{end}} = 1\) with CFL number \(\nu = 0.9\). Numerical solution for different values of \(\varepsilon\), see Eq. (3.46) and (3.45).
weight functions

\[ w_k = \frac{\alpha_k}{\sum_{i=0}^{r-1} \alpha_i} , \quad \alpha_k = \gamma_k \left( 1 + \frac{\tau}{\varepsilon + \beta_k} \right) , \quad k = 0, \ldots, r - 1, \]  

(3.49a)

\[ \varepsilon = \max_{x \in \Omega_d} (\|u_0^2\|_1, \|(u_0')^2\|_1, \ldots, \|(u_0^{(r-1)})^2\|_1) \Delta x^2 , \]  

(3.49b)

where \( \Omega_d \) is a set of points where the initial condition is discontinuous. In this thesis we consider \( r = 2 \). Thus, \( \gamma_0 = \gamma_{i-1/2}, \gamma_1 = \gamma_{i+1/2} \), etc. are defined as before and \( \tau \) is the square of the undivided difference on the entire stencil. In case of the compact three-point stencil this is \( \tau = (\bar{u}_{i+1} - 2\bar{u}_i + \bar{u}_{i-1})^2 = (\delta_{i+1/2} - \delta_{i-1/2})^2 \). Note that the definition of \( \varepsilon \) is not invariant under translation of the initial condition \( u_0 \). This leads to a different limiting behavior even though \( u_0 \) may simply be shifted by a constant, cf. Sec. 3.5.3.

Arándiga et al. [2] show that using the weights proposed by Yamaleev and Carpenter, the resulting scheme reaches the maximal order \((2r - 1)\) for sufficiently smooth solutions. However, near discontinuities, the scheme achieves order of accuracy \( O(\Delta x^2) \) which is worse or equal to \( O(\Delta x^r) \), \( r \geq 2 \), the order of accuracy of the underlying ENO scheme. Arándiga et al. report to have fixed this issue by slightly changing the weight functions using

\[ \alpha_k = \gamma_k \left( 1 + \left( \frac{\tau}{\varepsilon + \beta_k} \right)^\mu \right) , \quad k = 0, \ldots, r - 1, \quad \mu = \left\lceil \frac{r}{2} \right\rceil \]  

(3.50a)

\[ \varepsilon = K \Delta x^q , \text{ with } K > 0, \quad q \in \mathbb{N}, \quad q \leq 4r - 4 - r/\mu. \]  

(3.50b)

In the case of the three-point stencil, \( \mu = 1 \) and therefore the weight formulation remains the same. In the numerical test cases carried out in [2], \( K \) is set to 1, which makes \( \varepsilon \) a dimensional quantity that might be affected by changes of units. We will not consider these weight functions in our numerical experiments in Sec. 3.5 since the definition of \( \varepsilon \) in Eq. (3.49b) is clearly more elaborate. Solely in Sec. 3.5.4 we compare the obtained results with the scheme setting \( K = 1 \), i.e. \( \varepsilon = \Delta x^2 \) to show the importance of the definition of \( \varepsilon \).

### 3.4 Unifying View

In this section we want to place the different methods in a unifying view to point out their differences and especially their similarities.
3.4.1 Comparison of Limiter Functions

For the sake of simplicity, we do not discuss the three-dimensional plots of the limiter functions but rather sectional views at fixed values of $\delta_{i+\frac{1}{2}}$. This means, the limiter functions are depicted with a one-dimensional dependence on $\delta_{i-\frac{1}{2}}$. Lemma 3.7 states that for $\delta_{i+\frac{1}{2}} = 1$ this is equivalent to the standard $(\theta, \phi(\theta))$-plots such as Fig. 3.6, and found in textbooks.

To give an overview, Fig. 3.15 shows the limiter functions treated in the previous sections for $\delta_{i+\frac{1}{2}} \in \{2, 1, 0.5, 0.1\}$. In Fig. 3.15 we can nicely see that all limiter functions satisfy Property 3.1 (i). This is, for $\theta = 1$, i.e. $\delta_{i-\frac{1}{2}} = \delta_{i+\frac{1}{2}}$, the limiter functions continuously go through the point $(\delta_{i+\frac{1}{2}}, \delta_{i+\frac{1}{2}})$, which corresponds to

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.15}
\caption{Sectional view of different limiter functions for the fixed values $\delta_{i+\frac{1}{2}} \in \{2, 1, 0.5, 0.1\}$.}
\end{figure}
\( \phi(\theta) = 1 \). This situation represents smooth parts of the solution. It can be seen that WENO-JS has slope \( \frac{1}{3} \) only close to the point \( \delta_{i-\frac{1}{2}} = \delta_{i+\frac{1}{2}} \) meaning that as soon as the slopes (of same sign) differ from each other, the limiting takes effect, even though the function is still smooth. The smaller the values of \( \delta_{i\pm\frac{1}{2}} \), the stronger is this effect, see the trend from Fig. 3.15a to 3.15d.

For \( \delta_{i-\frac{1}{2}} \to \pm\infty \), i.e. \( \theta \to \pm\infty \), all considered functions, except \( H_3 \), tend to a constant. For the WENO schemes, this constant is non-zero, contrary to the negative part of the finite volume limiter. However, for the reconstruction of the cell interface values, \( H(\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}) \equiv \text{const} \) leads to first-order accuracy, independent of the value of the constant. In Fig. 3.15c and 3.15d we can see the switch of the finite volume limiter to the full-third-order reconstruction, highlighted by the dashed black line. Clearly, the construction of the transition zone is rather ad-hoc for the finite volume limiter, while it comes more naturally in WENO3. However, the finite volume setup allows a much more systematic control about the shape of the function \( H \) far away from the origin, particularly in the regions where \( \delta_{i-\frac{1}{2}} \) and \( \delta_{i+\frac{1}{2}} \) have opposite signs. It is here that the finite volume and WENO limiters show a very different behavior. In the point \( \delta_{i-\frac{1}{2}} = -\delta_{i+\frac{1}{2}} \), i.e. at extrema, all proposed limiters equal \( H_3 \). However, while \( H^{(c)}_{3L} \) has the same slope as the full-third-order reconstruction, the WENO limiters have negative slopes. This indicates that the finite volume limiter reconstructs extrema with higher accuracy, since in general extremal points might not lie exactly at \( \delta_{i-\frac{1}{2}} = -\delta_{i+\frac{1}{2}} \) but rather \( \delta_{i-\frac{1}{2}} \approx -\delta_{i+\frac{1}{2}} \). In these cases, the new limiter function approximates the full-third-order reconstruction, leading to higher order solutions.

Another interesting observation is the nature of \( H^{(c)}_{3L} \) and WENO-YC at \( \delta_{i-\frac{1}{2}} = 0 \). When considering the univariate limiter functions \( \phi(\theta) \) with \( \theta = \delta_{i-\frac{1}{2}}/\delta_{i+\frac{1}{2}} \), the point \( \delta_{i-\frac{1}{2}} = 0 \) is equivalent to \( \theta = 0 \). In this case, the conventional finite volume limiter functions in the MUSCL framework are set to 0, yielding a first-order reconstruction. This is also the case for WENO-JS, whereas the limiters proposed in this work and by Yamaleev and Carpenter yield non-zero contributions for small \( \delta_{i+\frac{1}{2}} \), see Fig. 3.15b, 3.15c, and 3.15d. This is exactly the desired behavior close to extrema where one slope may be zero and the other slope small but non-zero. Further away from the origin in the \( (\delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}) \)-plane, it is clear that the
limiter functions should yield 0 for \( \delta_{i+1/2} = 0 \) because this situation may correspond to a discontinuity. This feature can be observed in Fig. 3.15a.

3.5 Numerical Results comparing \( H^{(c)}_{3L} \) and WENO3

In this section, we apply and compare the different limiter functions discussed in the previous sections. In all test problems proposed in this work, we compare

(1) the original WENO3 scheme as introduced in \[26\], i.e. fixing \( \varepsilon = 10^{-6} \), called WENO-JS;

(2) WENO3 with the weight functions proposed by Yamaleev and Carpenter \[78\], called WENO-YC;

(3) the full third-order reconstruction \( H_3 \), Eq. \( (3.23) \); and

(4) the finite volume limiter function \( H^{(c)}_{3L} \), as introduced in Sec. 3.2.

The numerical flux function is HLL and the time derivative is approximated using the third-order strong stability preserving Runge-Kutta (SSP-RK3) method developed by Gottlieb et al. \[21\].
For all test cases, rather than presenting tables with errors and convergence rates, we plot the $L_1$- and $L_\infty$-errors. In these plots, we include reference slopes of the design order of accuracy, i.e. the order of accuracy the schemes obtain in theory.

### 3.5.1 Preliminary Test Case

In this test, we want to point out the importance of the choice of $\varepsilon$ for WENO-YC as we have done for $H^{(c)}_{\text{CT}}$ and WENO-JS in Sec. 3.2.4 and 3.3.2, respectively. We set $\varepsilon = C \Delta x^2$. According to Eq. (3.49), $C = \max_{x \not\in \Omega} (\|u_0^2\|, \|(u_0')^2\|)$. However, in this test case we set $C \in \{10^3, 1.0, 10^{-1}, 10^{-2}, 10^{-3}\}$ to study the influence of the coefficient. The results for the repetition of the test

$$u_t + u_x = 0$$
$$u_0(x) = \sin(\pi x), \quad x \in [-1, 1]$$

with $n = 80$ cells and $t_{\text{end}} = 1.0$ is depicted in Fig. 3.16 for different values of $\varepsilon$. One can clearly see the loss in accuracy if the constant for $\varepsilon$ is chosen too small. Note that the correct value for $\varepsilon$, as proposed by Yamaleev and Carpenter [78], is $\varepsilon = \max(1, \pi^2) \Delta x^2 = \pi^2 \Delta x^2$. This indicates that for smooth functions a
coefficient which is too small decreases the quality of the approximation. This can also be observed in Fig. 3.17 which shows the \( L_1 \) - and \( L_\infty \)-errors of the solution. We can see that the solution with \( \varepsilon = 1000 \Delta x^2 \) is third-order-accurate in both norms starting at the lowest resolution. While the solution with \( \varepsilon = 1.0 \Delta x^2 \) is still directly third-order-accurate in the \( L_1 \)-norm starting from \( n = 20 \) grid cells, we observe that the smaller the coefficient of \( \varepsilon \), the larger \( n \) must be chosen so that third order is achieved.

Of course, the same conclusion holds true for the choice of \( \alpha \) in \( H^{(c)}_{3L} \). This test thus shows that we need to carefully evaluate Eq. (3.49) and (3.39). A misinterpretation of the coefficients may lead to results which are significantly worse than what the scheme is capable to achieve.

### 3.5.2 Advection Equation with Smooth Initial Condition

With this first test case we aim to verify that all considered schemes are third-order accurate for smooth solutions. We solve the linear advection equation,

\[
 u_t + u_x = 0
\]  

\[(3.52)\]
with the smooth initial condition
\[ u_0(x) = \begin{cases} 
(0.5 + 0.5 \cos(5\pi(x - 0.5)))^4 & \text{if } 0.3 \leq x \leq 0.7 \\
0 & \text{else}
\end{cases} \]  

and periodic boundary conditions. The computational domain is $[0, 1]$, the CFL number $\nu = 0.8$ and the solution is advected until $t_{\text{end}} = 10$. The spatial resolution is the sequence of refined uniform grids with $n = 20, 40, 50, 100, 120, 170, 200, 300, 500, 700, 1000, 1500, 3000$ cells. For WENO-YC, according to Eq. (3.49), we set $\varepsilon = 20.67 \Delta x^2$ with $\Delta x = 1/n$. For the finite volume limiter function Eq. (3.41) and (3.39) we fix $\alpha = \max_{x \in \Omega \setminus \partial \Omega} |u''_0(x)| = 493.48$.

Fig. 3.18a and 3.18b show zooms of the solution at $t_{\text{end}}$ with different numerical schemes on a 170-cell grid. It can be seen that $H_{3L}^{(c)}$ and WENO-YC, as well as the full-third-order scheme, perform much better than the conventional WENO-JS scheme in terms of accuracy. This can also be observed in Fig. 3.19 which shows the $L_1$- and $L_\infty$-errors obtained at $t_{\text{end}}$. The convergence rates of $H_{3L}^{(c)}$, WENO-YC, and the unlimited third-order scheme reach 3rd order starting at $n = 150$ grid cells whereas WENO-JS shows this order of convergence only for very large numbers of cells and with a larger error constant.
3.5.3 Advection Equation with Discontinuous Initial Condition

In this section we want to discuss the behavior of the numerical schemes for solutions containing discontinuities. Therefore, we consider the advection equation, Eq. (3.52) with \( a = 1 \) and square wave initial condition

\[
  u_0(x) = \begin{cases} 
    1 & \text{for } -0.5 < x < 0.5 \\
    0 & \text{else}.
  \end{cases} \tag{3.54}
\]

The computational domain is \([-1, 1]\), the CFL number \( \nu = 0.8 \) and the solution is advected until \( t_{\text{end}} = 10 \), which corresponds to 5 periods in time. Due to the large gradients contained in the initial condition, the limiter functions have to take effect in order to avoid spurious oscillations to appear. Solving this test case with the full-third-order reconstruction Eq. (3.14) generates oscillations. This is the reason the finite volume limiter functions presented in Sec. 3.2 restrict the reconstruction to first order in these situations and WENO3 reduces to Eq. (3.48).

The WENO-YC parameter \( \varepsilon \) is given by \( \Delta x^2 \) (see Eq. (3.49b)) and \( H_3^{(c)} \) reduces to \( H_{3L} \) because \( \alpha = \max_{x \in \Omega; \Omega_d} |u'_0(x)| = 0 \).

This test case nicely shows the already mentioned drawback of the definition of \( \varepsilon \) in WENO-YC, Eq. (3.49b), namely the coefficient of \( \varepsilon \) which is not translationally invariant if the initial condition is shifted. To point this out, a second test case
has been chosen, where the initial condition has simply been shifted by +100, i.e

\[ IC_{+100} : \quad u_{0,+100}(x) = \begin{cases} 
101 & \text{for } -0.5 < x < 0.5 \\
100 & \text{else.} 
\end{cases} \]  \tag{3.55}

When applying WENO-JS to this new initial condition, the solution is does not show more oscillations, because \( \varepsilon \) is fixed to \( 10^{-6} \). In fact, the solution is the same as for \( u_0(x) \), only shifted by +100. Also, in the proposed finite volume limiter, the constant \( \alpha = \max_{x \in \Omega} |u''_{0,+100}(x)| = 0 \) does not change. Thus, the scheme yields the exact same results, shifted by +100. However, for WENO-YC, \( \varepsilon \), as given by Eq. (3.49b), is no longer \( \Delta x^2 \) but \( 20201 \Delta x^2 \). The higher value of \( \varepsilon \) leads to augmented oscillations in the solution, as can be seen in Fig. 3.20.

Here, for better comparison, the solution \( u_{+100}(x, t_{\text{end}}) \) of the test with shifted initial condition, which lies in the range \([99.9, 101.1]\), has been shifted to the range \([-0.1, 1.1]\). Thus, both test cases, Eq. (3.54) and (3.55) lie between -0.1 and 1.1 in the plots and the magnitude of the oscillations can be compared.

As seen in Fig. 3.20, the original WENO3 scheme does not cause any oscillation but it is rather dissipative. Our proposed limiter does not produce overshoots either. Additionally, it approximates the sharp structure of the initial condition better than WENO-JS. WENO-YC with correctly chosen parameter approximates the discontinuity almost as good as \( H_{3L}^{(c)} \), however, it causes oscillations. These are even larger for the badly chosen \( \varepsilon \). This behavior can also be observed in Fig. 3.21a and 3.21b which show the \( L_1 \)-error and the total variation (TV), respectively. The best order of convergence we can expect from a third-order scheme and a solution containing discontinuities is \( 3/4 \). This can be shown by the Fourier analysis of the modified equation with self-similar initial condition.

Even though, both solutions computed with WENO-YC cause oscillations, they obtain the order of accuracy \( 3/4 \). The more dissipative WENO-JS scheme is also of order \( 3/4 \) but with a larger error constant. Among the tested schemes, the best error constant is obtained with the novel limiter function \( H_{3L}^{(c)} \). The total variation of all schemes represents their behavior as seen in the solution. WENO-JS attains the TV of the exact solution \( TV(u_{\text{ex}}) \) from below, meaning that is does not cause overshoots at all, whereas WENO-YC is larger than \( TV(u_{\text{ex}}) \) for all spatial discretizations. \( H_{3L}^{(c)} \) is closest to \( TV(u_{\text{ex}}) \) and never lies above \( TV(u_{\text{ex}}) \) for all time steps, i.e. it does not cause oscillations at any time.
Double-logarithmic plot of the $L_1$-error vs. number of grid cells.

Total Variation vs. number of grid cells of the different schemes.

Figure 3.21 – Results of advection equation Eq. (3.52) with discontinuous initial condition, $\nu = 0.8$ and $t_{\text{end}} = 10$, i.e. the solution has been advected 10 around the domain.

3.5.4 Initial Condition Containing Smooth and Non-Smooth Features

We consider the same setup as in Sec. 3.5.2 with CFL number $\nu = 0.8$, $t_{\text{end}} = 10.0$ and initial condition

$$u_0(x) = \max \left( \min \left( \frac{x}{0.1} - 2, \ -\left( \frac{x}{0.1} - 2 \right) + 1, \ 0 \right) + \exp \left( -\left( \frac{x - 0.7}{0.15} \right)^4 \right) \sin(30\pi x) \right).$$  (3.56)

Figure 3.22 – Initial condition (3.56) containing smooth and non-smooth features.
In this problem, we want to test how accurate the different schemes resolve small features in a larger setting of a more complex solution. The spatial discretizations are $n = 20 \cdot 2^i$, $i = 0, \ldots, 7$ grid cells. For WENO-YC, we set $\varepsilon = 1042.83 \Delta x^2$, $\Delta x = 1/n$, according to Eq. (3.49b). For the finite volume limiter function we fix $\alpha = \max_{x \in \Omega \setminus \partial \Omega} |u_0^{(x)}| = 8887.87$. We run a third case with WENO-YC but setting $\varepsilon = \Delta x^2$ to show the difference in performance as described in Sec. 3.3.2 and 3.5.1. This test case corresponds to the weight functions proposed in [2] [29], where $\varepsilon = K \Delta x^2$ was used with $K = 1$. Furthermore, a test with the full-third-order reconstruction $H_3$, Eq. (3.23), was performed and compared to the numerical schemes discussed in this work. Fig. 3.23 shows a close up view of two significant regions of the solution with $n = 640$ grid cells. We observe that $H_3^{(x)}$ and WENO-YC with correctly-chosen parameter $\varepsilon$ perform much better than...
WENO-YC with $\varepsilon = \Delta x^2$. The results of these two schemes are very close to the ones of the full-third-order reconstruction. Also, all four schemes outperform the conventional WENO-JS scheme. This can also be seen in Fig. 3.24 which shows the double-logarithmic plot of the $L_1$- and $L_\infty$-errors versus the number of grid cells in the smooth part of the solution, $x \in [0.4, 1]$. The solution cannot be accurately represented with few grid cells by any of the treated schemes. Even the full third-order reconstruction does not resolve the details and therefore has a large error constant. As soon as a reasonable space discretization is reached, the order of convergence reaches 3rd order in the range $x \in [0.4, 1]$, i.e. the smooth part of the solution. Solely WENO-JS does not reach this order even at the highest resolution.

Considering the error on the whole domain $[0, 1]$, the convergence rate of all schemes degenerates when increasing the resolution. This is due to the fact that at higher resolutions the kinks in $x \in [0.1, 0.3]$ become well-resolved and thus, are recognized as non-smooth. The initial condition is an interesting test case, combining smooth and non-smooth features and therefore testing the capabilities of limiter functions. As shown in Fig. 3.23, near the position of the kinks, all schemes - with the exception of WENO-JS and $H_{3L}$ - generate undershoots. For our proposed finite volume limiter $H_{3L}^{(c)}$ this can be explained with the large asymptotic region. Since the maximal second derivative of the initial condition is very large, $\alpha = \max_{x \in \Omega} |u_0''(x)| = 8887.87$, the region where $H_{3L}^{(c)}$ reconstructs
with full-third order is relatively large. At the same time, the discrete second derivative of the kinks of the triangle are small compared to the extreme regions of the smooth part, and at these points, the solution is reconstructed using $H_3$ with no limiting. As a result, $H_{3L}^{(c)}$ causes undershoots, just as $H_3$ does, because this is what is effectively used at the kinks. However, compared to other limiting methods, we can in principle control these undershoots using our finite volume limiter $H_{3L}^{(c)}$. By choosing a smaller value for $\alpha$, i.e. a smaller asymptotic region, these undershoots can be completely avoided, as can be seen in the test case with pure discontinuity, cf. Sec. 3.5.3. An adaptive value for $\alpha$ would therefore eliminate the undershoots in the non-smooth region $x \in [0,0.4]$ while still resolving the smooth smooth parts in $x \in [0.4,1]$ with high order accuracy. However, such a local adaptivity would necessarily require to use a wider stencil for the reconstruction, because as shown in Fig. 3.11, three points can not distinguish between the kink and a strongly curved extremum on a coarse grid.

3.5.5 Sod Shock Tube Problem

Let us consider Sod’s problem [53], which describes a shock tube containing two different ideal gases at the left and right side of a diaphragm, placed at $x = 0$. The density, velocity, and pressure of the gases in the left and right region are given by

\[
\begin{pmatrix}
\rho_L \\
v_L \\
\rho_L
\end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.0 \\ 1.0 \end{pmatrix}, \quad \begin{pmatrix}
\rho_R \\
v_R \\
\rho_R
\end{pmatrix} = \begin{pmatrix} 0.125 \\ 0.0 \\ 0.1 \end{pmatrix}.
\]

At time $t > 0$, the diaphragm is removed and the gases begin to mix. The time evolution is described by the one dimensional Euler equations,

\[
u_t + f(u)_x = 0
\]

with $u=(\rho, \rho u, E)$, the flux function

\[
f(u) = (\rho u, \rho u^2 + p, u(E + p))^T,
\]

and the equation of state for ideal gases

\[E = \frac{p}{\gamma - 1} + \frac{1}{2} \rho u^2,\]
with $\gamma = 1.4$. The computational domain is set to $[-2, 2]$ and the test is conducted with $N = 100$ grid cells until $T_{\text{end}} = 0.8$. We compare the new limiter function $H_{3L}^{(c)}$ with the full third-order reconstruction and WENO-JS. WENO-YC is not included in the plots because it produced negative values for pressure, when run with $\varepsilon = 2.25$.

Due to the purely discontinuous form of the initial condition, we obtain $\alpha = 0$, just as in Sec. 3.5.3. This means, that $H_{3L}^{(c)} = H_{3L}$, i.e. no asymptotic region exists in this test case. The reconstruction techniques have all been applied to the primitive variables $\rho$, $u$ and $p$.

Sod’s shock tube problem leads to three characteristic waves, which can be seen in the solution, Fig. 3.25. Both, the density and pressure profile show the rarefaction wave and the shock. The contact discontinuity can only be seen in the density profile, Fig. 3.25a, not in the pressure distribution, Fig. 3.25b. The solution shows that applying the full-third order reconstruction leads to over- and undershoots close to the discontinuities. WENO-JS yields good results concerning this feature, however, does not approximate the true solution as accurate as $H_{3L}^{(c)}$, see also discussion in Sec. 3.5.3.
Figure 3.26 – Solution of different reconstruction techniques for the Shu-Osher Problem, Eq. (3.59), (3.58) with $N = 640$ grid cells on the domain $[-4.5, 4.5]$, CFL $\nu = 0.95$ at time $T_{\text{end}} = 1.8$.

3.5.6 Shu-Osher Problem

In this problem, originally introduced by Shu and Osher [52], a Mach 3 shock is interacting with an acoustic wave in the density profile. The computational domain is fixed to $[-4.5, 4.5]$ and the shock at time $t = 0$ is situated at $x = -4$. The initial conditions of the primitive variables density, velocity and pressure, to the left and right of $x = -4$ are given by

$$
\begin{pmatrix}
\rho_L \\
v_L \\
p_L
\end{pmatrix} = \begin{pmatrix}
3.857143 \\
2.629369 \\
10.333333
\end{pmatrix}, \quad \begin{pmatrix}
\rho_R \\
v_R \\
p_R
\end{pmatrix} = \begin{pmatrix}
1 + 0.2 \sin(5x), \\
0.0 \\
1.0
\end{pmatrix}, \quad (3.59)
$$

and the time evolution is governed by the one-dimensional Euler equations, Eq. (3.58). Just as in [52], the solution is computed at $T_{\text{end}} = 1.8$. We compare the solutions of the full third-order reconstruction, WENO-JS, WENO-YC (where $\varepsilon = 21.932$), and the new limiter function $H^{(c)}_{\text{3L}}$ with $N = 640$ and $N = 1280$ cells to a reference solution, which is the numerical solution of WENO-JS with 10,000 grid cells. The CFL number is set to $\nu = 0.95$ in all tests. The combined limiter function $H^{(c)}_{\text{3L}}$ includes an asymptotic region, cf. (3.39), with $\alpha = 5.0$ the reconstruction techniques have all been applied to the primitive variables $\rho, u$ and $p$.  
Fig. 3.26 shows a comparison of the solutions obtained on 640 grid cells of the full third-order reconstruction, WENO-JS, WENO-YC, and the new limiter function \( H_{3L}^{(c)} \). A zoom of the areas of interest of the solutions computed on 1280 grid cells is shown in Fig. 3.27. Overall, it can be seen that applying the full-third-order reconstruction and also WENO-YC lead to under- and overshoots close to discontinuities. WENO-JS does not produce overshoots, however, it limits too much, so that the reference solution is not approximated as well as by the limiter function \( H_{3L}^{(c)} \). This is especially visible in regions with large gradients.

### 3.5.7 Ideal Magnetohydrodynamics

Ideal magnetohydrodynamics (MHD) describes the flow of plasma, assuming infinite electrical resistivity. The equations in one-dimensional processes read

\[
\partial_t \begin{pmatrix}
\rho \\
\rho v_x \\
\rho v_t \\
B_t \\
E
\end{pmatrix}
+ \partial_x \begin{pmatrix}
\rho v_x \\
\rho v_x^2 + p + \frac{1}{2}B_t^2 \\
\rho v_x v_t - B_x B_t \\
v_x B_t - B_x v_t \\
(E + p + \frac{1}{2}B_t^2)v_x - B_x B_t \cdot v_t
\end{pmatrix} = 0 \tag{3.60}
\]

with density \( \rho \), normal and transverse flow velocities \( v_x \), and \( v_t = (v_y, v_z) \), respectively. Due to divergence constraints, the normal component of the magnetic field \( B_x \) is constant in the one-dimensional case. The transverse magnetic field is \( B_t = (B_y, B_z) \). We consider ideal gas and the total energy \( E \) is given in terms of
the pressure $p$ by

$$E = \frac{1}{\gamma - 1} p + \frac{1}{2} \rho (v_x^2 + v_t^2) + \frac{1}{2} B_t^2,$$

(3.61)

see e.g. [64]. The adiabatic constant $\gamma$ is set to $5/3$. There is no equation for the normal component of the magnetic field in Eq. (3.60) since $B_x \equiv const.$ leads to the statement that $B_x$ is also constant in time [64]. Thus, $B_x$ is considered a parameter rather than an unknown variable.

System (3.60) is hyperbolic and contains seven equations for the seven unknowns, $U = (\rho, v_x, v_t, p, B_t)$, exhibiting seven characteristic velocities.

We consider the test case proposed by Torrilhon [64] with the initial conditions

$$(\rho^L, v_x^L, v_t^L, B_x^L, B_t^L, p^L) = (3, 0, (0, 0)^T, 1.5, (1, 0)^T, 3) \quad \text{if } x < 0,$$

$$(\rho^R, v_x^R, v_t^R, B_x^R, B_t^R, p^R) = (1, 0, (0, 0)^T, 1.5, (\cos(1.5), \sin(1.5))^T, 1) \quad \text{if } x \geq 0,$$

(3.62)

and $B_x = 1.5$. The computational domain is $[-4, 4]$, and the solutions depicted in Fig. 3.28 have been obtained with $N = 800$ grid cells, CFL condition $0.45$, and end time $T_{end} = 1.0$. The exact solution has been obtained by [62]. Fig. 3.28a shows the solution of the density profile computed with different third-order reconstructions. Furthermore, a zoom of the slow right moving shock at $x = 1.3$ has been included, where the resolution of all solvers can nicely be compared. Fig. 3.28b depicts solutions of the $y$-component of the magnetic field, computed with the same high-order reconstructions. Here too, a zoom of the slow right moving shock is shown to be able to compare the solutions.

It can be stated that the full third-order reconstruction $H_3$ yields oscillations at the discontinuities, as expected. WENO-YC also leads to over- and undershoots due to the way $K$ is set in $\varepsilon$, see Eq. (3.49b). As in the above test cases, WENO-JS yields conservative solutions without oscillations but smeared out shocks. The limiter function $H_{3L}$ does not show oscillations in the density profile and only slight overshoots in the magnetic field $B_y$ leading to steep gradients as desired.
Figure 3.28 – Solution of the ideal MHD equations using different reconstructions. The computational domain \([-4, 4]\) is discretized using \(N = 800\) grid cells. The simulation is conducted till \(T_{\text{end}} = 1.0\) with CFL condition \(\nu = 0.45\).
3.6 Generalization to Non-Equidistant Grids

For general grids, the size of cell $C_i$, denoted by $\Delta x_i$ is not uniform for all cells, i.e. $\Delta x_i \neq \Delta x \forall i$, see Fig. 3.29. In this case, the definition of the undivided differences $\delta_{i+\frac{1}{2}}$ Eq. (3.7b), (3.7a) is not meaningful anymore and new concepts need to be developed.

Starting again with the full third-order reconstruction, consider a quadratic polynomial $p_i(x)$ in cell $i$ that has to maintain the cell averages in the three cells $C_{i+\ell}, \ell \in \{-1, 0, 1\}$. This polynomial is then evaluated at the cell boundaries $x_{i\pm\frac{1}{2}}$ and yields the reconstructed cell interface values

$$u_{i+\frac{1}{2}}^{(-)} = p_i(x_{i+\frac{1}{2}}) = \bar{u}_i + \frac{1}{2} H_{3,\text{neq}}$$

$$u_{i-\frac{1}{2}}^{(+)} = p_i(x_{i-\frac{1}{2}}) = \bar{u}_i - \frac{1}{2} H_{3,\text{neq}}.$$  

Even though this procedure is similar to the full third-order reconstruction on uniform grids, Eq. (3.14), the reconstruction function $H_{3,\text{neq}}$ differs from $H_3$ since the different cell sizes need to be taken into account. The full (unlimited) third-order reconstruction function reads

$$H_{3,\text{neq}} \left( \delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}, \Delta x_i, \Delta x_{i-1}, \Delta x_{i+1} \right) = \frac{\Delta x_i}{\Delta_i} \frac{1}{3} \left( 2 \frac{\Delta x_{i+\frac{1}{2}}}{\Delta_{i+\frac{1}{2}}} \delta_{i+\frac{1}{2}} + \frac{\Delta x_{i+1}}{\Delta_{i+\frac{1}{2}}} \delta_{i-\frac{1}{2}} \right)$$  

$$\text{(3.64a)}$$
with the abbreviations
\[ \Delta_i = \frac{\Delta x_{i-1} + \Delta x_i + \Delta x_{i+1}}{3}, \quad \Delta_{i-\frac{1}{2}} = \frac{\Delta x_{i-1} + \Delta x_i}{2}, \quad \Delta_{i+\frac{1}{2}} = \frac{\Delta x_i + \Delta x_{i+1}}{2}. \] (3.64b)

As in the equidistant case, the reconstructed interface values, Eq. (3.63), can be compactly reformulated as
\[
u_{i\pm\frac{1}{2}}^{(+)} = \bar{u}_i \pm \frac{\Delta x_i}{2} \frac{\Delta x_{i-1} \delta_{i+\frac{1}{2}} + \Delta x_i \delta_{i+\frac{1}{2}} + \Delta x_{i+1} \delta_{i-\frac{1}{2}}}{\Delta x_{i-1} + \Delta x_i + \Delta x_{i+1}}
\] (3.65a)

with
\[
\delta_{i-\frac{1}{2}} = \delta_{i-\frac{1}{2}}, \quad \delta_{i+\frac{1}{2}} = \delta_{i+\frac{1}{2}}.
\] (3.65b)

It can easily be seen that for equidistant grids, \( i.e. \Delta x_{i-1} = \Delta x_i = \Delta x_{i+1} \equiv \Delta x \), the abbreviated terms reduce to \( \Delta_i = \Delta_{i-\frac{1}{2}} = \Delta_{i+\frac{1}{2}} = \Delta x \) and therefore, the formulas for \( H_{3,\text{neq}} \) and \( H_3 \) match, as expected.

Eq. (3.64) and (3.65) indicate that for non-equidistant meshes, the equivalent of the undivided differences \( \delta_{i\pm\frac{1}{2}} \) are the scaled slopes
\[
u_{i\pm\frac{1}{2}}^{(-)} : \quad \delta_{i-\frac{1}{2}} \rightarrow \Delta x_{i+1} \frac{\Delta x_{i+\frac{1}{2}}}{\Delta_{i+\frac{1}{2}}} = \Delta x_{i+1} \tilde{\delta}_{i-\frac{1}{2}} \]
\[
\delta_{i+\frac{1}{2}} \rightarrow \Delta_{i+\frac{1}{2}} \frac{\Delta x_{i+\frac{1}{2}}}{\Delta_{i+\frac{1}{2}}} = \Delta x_i \tilde{\delta}_{i+\frac{1}{2}} + \Delta x_{i-1} \tilde{\delta}_{i+\frac{1}{2}}
\] (3.66)

for the reconstruction of the right interface of cell \( C_i \) and
\[
u_{i\pm\frac{1}{2}}^{(+)} : \quad \delta_{i-\frac{1}{2}} \rightarrow \Delta_{i+\frac{1}{2}} \frac{\Delta x_{i-\frac{1}{2}}}{\Delta_{i+\frac{1}{2}}} = \Delta x_i \tilde{\delta}_{i-\frac{1}{2}} + \Delta x_{i+1} \tilde{\delta}_{i-\frac{1}{2}} \]
\[
\delta_{i+\frac{1}{2}} \rightarrow \Delta_{i+\frac{1}{2}} \frac{\Delta x_{i+\frac{1}{2}}}{\Delta_{i+\frac{1}{2}}} = \Delta x_{i-1} \tilde{\delta}_{i+\frac{1}{2}}
\] (3.67)

for the reconstruction of the left interface of cell \( C_i \). These expressions resemble the smoothness indicators introduced by Jiang and Shu [26], which are given by \( \Delta x_i \tilde{\delta}_{i-\frac{1}{2}} \) and \( \Delta x_i \tilde{\delta}_{i+\frac{1}{2}} \).
In order to generalize the third-order limiter function developed in [45], we replace
the undivided differences as mentioned above to obtain the reconstructions
\[ u_{i + \frac{1}{2}}^{(-)} = \bar{u}_i + \frac{1}{2} H_{3L}^{(c)} \left( \Delta x_{i+1} \tilde{\delta}_{i-\frac{1}{2}}, \Delta_{-\frac{1}{2}} \tilde{\delta}_{i+\frac{1}{2}} \right) \] (3.68a)
\[ u_{i - \frac{1}{2}}^{(+)} = \bar{u}_i - \frac{1}{2} H_{3L}^{(c)} \left( \Delta x_{i-1} \tilde{\delta}_{i+\frac{1}{2}}, \Delta_{+\frac{1}{2}} \tilde{\delta}_{i-\frac{1}{2}} \right). \] (3.68b)

with the limiter function \( H_{3L}^{(c)} \) (3.41) described in Sec. 3.2. The non-equidistant
version of the limiter function can be defined as a function \( H_{3L, \text{neq}}^{(c)} \) given by
\[ H_{3L, \text{neq}}^{(c)} \left( \delta_{i-\frac{1}{2}}, \delta_{i+\frac{1}{2}}, \Delta x_i, \Delta x_{i-1}, \Delta x_{i+1} \right) = H_{3L}^{(c)} \left( \Delta x_{i+1} \frac{\delta_{i-\frac{1}{2}}}{\Delta_{-\frac{1}{2}}}, \Delta_{+\frac{1}{2}} \frac{\delta_{i+\frac{1}{2}}}{\Delta} \right) \]
\[ H_{3L, \text{neq}}^{(c)} \left( \delta_{i+\frac{1}{2}}, \delta_{i-\frac{1}{2}}, \Delta x_i, \Delta x_{i+1}, \Delta x_{i-1} \right) = H_{3L}^{(c)} \left( \Delta x_{i-1} \frac{\delta_{i+\frac{1}{2}}}{\Delta_{+\frac{1}{2}}}, \Delta_{-\frac{1}{2}} \frac{\delta_{i-\frac{1}{2}}}{\Delta} \right). \] (3.69)

The decision criterion \( \eta \) (3.39) for non-uniform meshes reads
\[ \eta(\delta_1, \delta_2) = \sqrt{\frac{\delta_1^2 + \delta_2^2}{\frac{5}{2} \alpha dx^2}}, \] (3.70)
where \( \delta_1, \delta_2 \) are the same input arguments as for \( H_{3L, \text{neq}}^{(c)} \), see Eq. (3.41) and \( dx \)
is the average mesh size, \( dx = (\sum_i \Delta x_i) / \# \text{cells} \).

### 3.6.1 Numerical Results for Non-Equidistant Grids

In this section we want to verify that extending the limiter function \( H_{3L}^{(c)} \) to
non-equidistant grids in one space dimension still yields the desired third-order
accuracy. We therefore present different numerical examples validating the theory
of Sec. 3.6.

#### 3.6.1.1 Testing the Convergence Order on a Non-Uniform 1D Grid

In this section we want to verify that the extension of the third-order limiter
function \( H_{3L}^{(c)} \) from equidistant to non-equidistant grids still yields third order
accurate solutions. Thus, we consider the linear advection equation with smooth initial conditions
\[
\begin{align*}
  &u_t + u_x = 0 \\
  &u(x,0) = \sin(2\pi x)
\end{align*}
\]  
(3.71)
on the domain [0, 1] with periodic boundary conditions. In order to verify the order of convergence we carry out simulations with \(N = 25 \times 2^j, j = 0, \ldots, 6\) grid cells with end time \(t_{\text{end}} = 1.0\) and CFL number 0.95. Since we are interested in non-equidistant grids, the original grid is perturbed by adding \(c_1 \cdot \sin(c_2 2\pi x_{i+1/2})\) to each cell boundary \(x_{i+1/2}\) with some constants \(c_1, c_2 \in \mathbb{R}\). In this test case, \(c_1 = (10 \cdot c_2)^{-1}\) and \(c_2 = 5\) have been applied. Fig. 3.30 shows the exact solution as well as the solution obtained with \(H^{(c)}_{3L}\) on a grid with 25 cells. This solution is compared to the third order WENO method developed by Liu et. al. [35] with the smoothness measure by Jiang and Shu [26]. This scheme is denoted by WENO-JS. The choice of depicting a coarse grid emerges from the fact that the non-equidistant mesh structure is well-visible. Also, the improved solution quality of the limiter function can be best observed on coarse meshes, as for fine grids, all convergent methods look the same.

Finally, to verify that the limiter function is third-order accurate on non-equidistant grids, Table 3.2a displays the \(L_1\)- and \(L_{\infty}\)-errors of \(H^{(c)}_{3L}\). The corresponding empirical order of convergence (EOC) is obtained by
<table>
<thead>
<tr>
<th>Grid</th>
<th>$|u - u_{ex}|_1$</th>
<th>EOC</th>
<th>$|u - u_{ex}|_\infty$</th>
<th>EOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>8.322E-03</td>
<td></td>
<td>1.311E-02</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>1.347E-03</td>
<td><strong>2.63</strong></td>
<td>2.209E-03</td>
<td><strong>2.57</strong></td>
</tr>
<tr>
<td>100</td>
<td>1.817E-04</td>
<td><strong>2.89</strong></td>
<td>2.921E-04</td>
<td><strong>2.92</strong></td>
</tr>
<tr>
<td>200</td>
<td>2.323E-05</td>
<td><strong>2.97</strong></td>
<td>3.663E-05</td>
<td><strong>3.00</strong></td>
</tr>
<tr>
<td>400</td>
<td>2.920E-06</td>
<td><strong>2.99</strong></td>
<td>4.589E-06</td>
<td><strong>3.00</strong></td>
</tr>
<tr>
<td>800</td>
<td>3.656E-07</td>
<td><strong>3.00</strong></td>
<td>5.743E-07</td>
<td><strong>3.00</strong></td>
</tr>
</tbody>
</table>

(a) Perturbed grid by adding $\frac{1}{50} \sin(10\pi x_{i+1/2})$ to each cell boundary $x_{i+1/2}$.

<table>
<thead>
<tr>
<th>Grid</th>
<th>$|u - u_{ex}|_1$</th>
<th>EOC</th>
<th>$|u - u_{ex}|_\infty$</th>
<th>EOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>6.412E-03</td>
<td></td>
<td>1.004E-02</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>7.888E-04</td>
<td><strong>3.02</strong></td>
<td>1.240E-03</td>
<td><strong>3.02</strong></td>
</tr>
<tr>
<td>100</td>
<td>9.844E-05</td>
<td><strong>3.00</strong></td>
<td>1.547E-04</td>
<td><strong>3.00</strong></td>
</tr>
<tr>
<td>200</td>
<td>1.234E-05</td>
<td><strong>3.00</strong></td>
<td>1.940E-05</td>
<td><strong>3.00</strong></td>
</tr>
<tr>
<td>400</td>
<td>1.535E-06</td>
<td><strong>3.01</strong></td>
<td>2.411E-06</td>
<td><strong>3.01</strong></td>
</tr>
<tr>
<td>800</td>
<td>1.930E-07</td>
<td><strong>2.99</strong></td>
<td>3.033E-07</td>
<td><strong>2.99</strong></td>
</tr>
</tbody>
</table>

(b) Non-equidistant grid with random cell sizes.

Table 3.2 – Errors of $H^{(c)}_{3L}$ in $L_1$- and $L_\infty$-norm and corresponding empirical order of convergence (EOC).

\[
\log(\text{err}_{j+1}/\text{err}_j)/\log(N_j/N_{j+1}).
\]

It can be seen that the limiter function obtains the desired accuracy already on coarse meshes.

### 3.7 Third-Order Limiter in Two Space Dimensions

In this section we extend the third-order limiter function to two space dimensions covering three core areas. First, we discuss how to apply the scheme on uniform Cartesian grids. Then we extend the method to adaptively refined grids. The last part of this section explains how the method can be used on rectangular grids.
which are non-uniform in \( x \)- and \( y \)-direction.

### 3.7.1 Formulation for 2D Cartesian Grids

In two space dimensions, the domain of interest \( \Omega \) is divided into non-overlapping cells \( C_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] \) such that \( \Omega = \bigcup_{i,j} C_{i,j} \). Denote by \((x_i, y_j)\) the cell center of cell \( C_{i,j} \). The mesh width is given by \( \Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} \) and \( \Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}} \). Furthermore we denote by \( \tilde{u}_{i,j} \) the cell–averaged value over cell \( C_{i,j} \) and by \( \tilde{u}_{i,\frac{1}{2},j} \) and \( \bar{u}_{i,\frac{1}{2},j} \) the interface–averaged values over the corresponding interface. The tilde notation \( \tilde{\cdot} \) denotes the average in \( y \)-direction and bar \( \bar{\cdot} \) denotes the average in \( x \)-direction as in the one-dimensional case.

Integrating a hyperbolic conservation law of the form

\[
\partial_t u(x, y, t) + \partial_x f(u(x, y, t)) + \partial_y g(u(x, y, t)) = 0 \tag{3.72}
\]

over cell \( C_{i,j} \) and dividing by the cell area \( \Delta x_i \Delta y_j \) yields the two-dimensional semi-discrete flux-differencing finite volume scheme \[34\]

\[
\frac{d \tilde{u}_{i,j}}{d t} = -\frac{1}{\Delta x_i} \left( \tilde{f}_{i+\frac{1}{2},j} - \tilde{f}_{i-\frac{1}{2},j} \right) - \frac{1}{\Delta y_j} \left( \bar{g}_{i,j+\frac{1}{2}} - \bar{g}_{i,j-\frac{1}{2}} \right). \tag{3.73}
\]

Here, the numerical flux functions \( \tilde{f}_{i+\frac{1}{2},j} \) and \( \bar{g}_{i,j+\frac{1}{2}} \) are approximations to averages of the flux across the corresponding interface \[34\]

\[
\tilde{f}_{i+\frac{1}{2},j} \approx \frac{1}{\Delta y_j} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(u(x_{i+\frac{1}{2}}, y, t)) dy, \tag{3.74a}
\]

\[
\bar{g}_{i,j+\frac{1}{2}} \approx \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g(u(x, y_{j+\frac{1}{2}}, t)) dx. \tag{3.74b}
\]

Similar to other finite volume methods, applying the schemes described in Sec. 3.2 in a dimension–by–dimension fashion results in a second order scheme, see e.g. \[51\][\,79\][\,6\]. In order to remain third-order accurate, we apply the fourth order transformation proposed by Buchmüller and Helzel \[6\]. Incorporating this so-called order-fix, the scheme can be summarized as follows.

1. Compute the averaged values of the conserved quantities at the cell interfaces in the interior of cell \( C_{i,j} \) for all \( i, j \) using the one-dimensional limiter...
functions described in Sec. 3.2

\[
\begin{align*}
\bar{u}_{i+\frac{1}{2},j}^{(-)} &= \hat{u}_{i,j} + \frac{1}{2} \Delta \bar{H}(\delta_{i-\frac{1}{2},j}, \delta_{i+\frac{1}{2},j}), \\
\bar{u}_{i-\frac{1}{2},j}^{(+)} &= \hat{u}_{i,j} - \frac{1}{2} \Delta \bar{H}(\delta_{i+\frac{1}{2},j}, \delta_{i-\frac{1}{2},j}), \\
\bar{u}_{i,j+\frac{1}{2}}^{(-)} &= \hat{u}_{i,j} + \frac{1}{2} \Delta \bar{H}(\delta_{i,j-\frac{1}{2}}, \delta_{i,j+\frac{1}{2}}), \\
\bar{u}_{i,j-\frac{1}{2}}^{(+)} &= \hat{u}_{i,j} - \frac{1}{2} \Delta \bar{H}(\delta_{i,j+\frac{1}{2}}, \delta_{i,j-\frac{1}{2}}).
\end{align*}
\] (3.75a)

Here, the reconstruction function \( \bar{H} \) can be the unlimited third-order reconstruction \( H_3 \), the limiter function \( H_{3L}^{(c)} \) or any other third-order limiter fitting the setting. The undivided differences in two-dimensions, \( \delta_{i\pm\frac{1}{2},j}, \delta_{i,j\pm\frac{1}{2}} \), are defined similarly to their one-dimensional equivalents, Eq. (3.7a) and (3.7b),

\[
\begin{align*}
\delta_{i-\frac{1}{2},j} &= \bar{u}_{i,j} - \bar{u}_{i-1,j} \\
\delta_{i+\frac{1}{2},j} &= \bar{u}_{i+1,j} - \bar{u}_{i,j} \\
\delta_{i,j-\frac{1}{2}} &= \bar{u}_{i,j} - \bar{u}_{i,j-1} \\
\delta_{i,j+\frac{1}{2}} &= \bar{u}_{i,j+1} - \bar{u}_{i,j}.
\end{align*}
\] (3.76)

2. Compute point values of the conserved quantities at the center of each cell interface, \textit{i.e.} compute

\[
\begin{align*}
u_{i+\frac{1}{2},j}^{(\pm)} &= \hat{u}_{i+\frac{1}{2},j}^{(\pm)} - \frac{1}{24} \left( \bar{g}_{i+\frac{1}{2},j}^{(\pm)} - 2 \bar{u}_{i+\frac{1}{2},j}^{(\pm)} + \bar{u}_{i+\frac{1}{2},j+1}^{(\pm)} \right), \\
\nu_{i,j+\frac{1}{2}}^{(\pm)} &= \bar{u}_{i,j+\frac{1}{2}}^{(\pm)} - \frac{1}{24} \left( \bar{g}_{i,j+\frac{1}{2}}^{(\pm)} - 2 \bar{u}_{i,j+\frac{1}{2}}^{(\pm)} + \bar{u}_{i+1,j+\frac{1}{2}}^{(\pm)} \right).
\end{align*}
\] (3.77)

3. Compute fluxes at the center of the cell interfaces using the computed point values and a consistent numerical flux function, \textit{i.e.}

\[
\begin{align*}
\hat{f}_{i+\frac{1}{2},j} &= \hat{f}(\nu_{i+\frac{1}{2},j}^{(-)}, \nu_{i+\frac{1}{2},j}^{(+)}), \\
\hat{g}_{i,j+\frac{1}{2}} &= \hat{g}(\nu_{i,j+\frac{1}{2}}^{(-)}, \nu_{i,j+\frac{1}{2}}^{(+)}).
\end{align*}
\] (3.78)

4. Compute averaged values of the numerical flux function, \textit{i.e.} compute

\[
\begin{align*}
\bar{f}_{i+\frac{1}{2},j} &= \hat{f}_{i+\frac{1}{2},j} + \frac{1}{24} \left( \hat{f}_{i+\frac{1}{2},j-1} - 2 \hat{f}_{i+\frac{1}{2},j} + \hat{f}_{i+\frac{1}{2},j+1} \right), \\
\bar{g}_{i,j+\frac{1}{2}} &= \hat{g}_{i,j+\frac{1}{2}} + \frac{1}{24} \left( \hat{g}_{i-1,j+\frac{1}{2}} - 2 \hat{g}_{i,j+\frac{1}{2}} + \hat{g}_{i+1,j+\frac{1}{2}} \right).
\end{align*}
\] (3.79)
5. Use a high-order accurate Runge-Kutta method for the update in time. In this work, we use the strong stability preserving third-order Runge-Kutta method described by Gottlieb et. al. [21].

This procedure is quite robust even when discontinuities are present. Nevertheless in some situations an unphysical state may be created, therefore we apply a simple limiting as suggested by Buchmüller et. al. [7]. Details on this limiting procedure can be found in the original paper.

Step 1. comprises the reconstruction function $H(\cdot, \cdot)$ that has been described in Sec. 3.2. For purely smooth solutions, the full third-order reconstruction $H_3$ can be used in this step. However, when discontinuities are present, the limiter function $H_{3L}^{(c)}$, Eq. (3.41), is more advisable since oscillations are prevented. In principle, in the dimension-splitting approach described above, $H_{3L}^{(c)}$ can be applied in the same manner as in one dimension. The decision criterion $\eta$ (3.39) can also be used without any changes in the splitting approach. Only the definition of the radius of the asymptotic region, $\alpha$, Eq. (3.40), has to be adapted. In two space dimensions it is defined as

$$\alpha = \max_{(x,y) \in \Omega \setminus \Omega_d} |\Delta u_0(x,y)|.$$  

Equation (3.80)

Again, $\Omega$ is the domain of interest and $\Omega_d \subset \Omega$ the subset containing discontinuities.

### 3.7.2 Adaptive Mesh Refinement (AMR)

For computations in two dimensions, we use the parallel AMR framework Racoon developed by Dreher and Grauer [16]. Both, the grid adaptivity and the parallelization are based on a block-structure. In a 2-dimensional space, a grid of level $\ell$ consists of $(2^2)^\ell = 4^\ell$ blocks. Computations can be performed simultaneously on each block and due to the Cartesian grid structure within each block, we can simply apply the method described in Sec. 3.7.1. A typical block is illustrated in Fig. 3.31b. The cells in the gray region are ghost cells needed for the communication between the blocks. For refinement a block of level $\ell$ is replaced by $2^d$ blocks of level $\ell + 1$. These blocks may then be further refined until the maximum refinement level is reached. There are three reasons for refinement.
Some refinement criteria is met. Here we compute

\[
\delta = \frac{|q_{i-1,j} - 2q_{i,j} + q_{i+1,j}| + |q_{i,j-1} - 2q_{i,j} + q_{i,j+1}|}{|q_{i,j}| \Delta x \Delta y}
\]  \hspace{1cm} (3.81)

for each cell. If \( \delta \) is bigger than a predefined threshold \( \delta_0 \), the cell is marked for refinement and therefore the block will be refined.

Neighbouring blocks are may also refined, so that after refinement the region with the marked cell is surrounded by fine blocks. In 2D for example, if a cell in the upper left part of a block is marked for refinement, then the upper block, the block on the left-hand side, and the block in the upper left diagonal direction will be refined as well.

Finally the grid needs to be properly nested, that is the level of neighboring blocks is not allowed to differ by more than one. Which may lead to further refinement.

In Fig. 3.31\( ^a \) a typical block structure is illustrated. The blocks in this figure are of level \( 2 - 5 \), where light gray corresponds to 2 and increasing darkness corresponds to increasing refinement level.
As mentioned above, ghost cells are used for communication between blocks and need to be updated in every stage of the time stepping scheme. In most cases this means simply copying the cell-averaged data from the neighboring block. To transfer data from a fine block to a coarse block, the values of the corresponding cells are averaged. Values for the fine block are created by polynomial reconstruction using data of the coarse block. The same is procedure is applied when a block is refined or coarsen again, see [6] for more details.

### 3.7.3 Non-Uniform Rectangular 2D Grids

In this section we consider non-uniform two-dimensional meshes. The Cartesian grid cells are transformed into non-uniform cells in $x$- and $y$-direction by adding a perturbation to the cell centers $(x_i, y_j)$. In this work we used the transformation

\[
\begin{align*}
  x_i & \rightarrow x_i + \delta x \sin(c_x \pi x_i), \\
  y_j & \rightarrow y_j + \delta y \sin(c_y \pi y_j)
\end{align*}
\]

(3.82)

with the constants $\delta x, c_x, \delta y, c_y$, which determine the structure of the mesh. This procedure yields rectangles that are still aligned with the $x$- and $y$-axes but exhibit different cell sizes. Fig. 3.32 shows an example of such a mesh.

For the linear advection equation, in order to apply the numerical schemes presented in the first chapters of this thesis, we need to adapt the reconstruction routine. The values of the conserved quantities at the cell interfaces in the interior of cell $C_{i,j}$ for all $i, j$ are obtained using the one-dimensional limiter functions for non-uniform grids, described in Sec. 3.6

\[
\begin{align*}
  u^{(-)}_{i+\frac{1}{2},j} &= \tilde{u}_{i,j} + \frac{1}{2} H(\delta_{i-\frac{1}{2},j}, \delta_{i+\frac{1}{2},j}, \Delta x_i, \Delta x_{i-1}, \Delta x_{i+1}), \\
  u^{(+)}_{i-\frac{1}{2},j} &= \tilde{u}_{i,j} - \frac{1}{2} H(\delta_{i+\frac{1}{2},j}, \delta_{i-\frac{1}{2},j}, \Delta x_i, \Delta x_{i+1}, \Delta x_{i-1}), \\
  u^{(-)}_{i,j+\frac{1}{2}} &= \tilde{u}_{i,j} + \frac{1}{2} H(\delta_{i-j-\frac{1}{2}}, \delta_{i-j+\frac{1}{2}}, \Delta y_j, \Delta y_{j-1}, \Delta y_{j+1}), \\
  u^{(+)}_{i,j-\frac{1}{2}} &= \tilde{u}_{i,j} - \frac{1}{2} H(\delta_{i,j+\frac{1}{2}}, \delta_{i,j-\frac{1}{2}}, \Delta y_j, \Delta y_{j+1}, \Delta y_{j-1}).
\end{align*}
\]

(3.83a)

Here, $H = H_{3,\text{neq}}$ or $H = H_{3L,\text{neq}}$. The two-dimensional undivided differences $\delta_{i\pm\frac{1}{2},j}, \delta_{i,j\pm\frac{1}{2}}$ are defined above, Eq. (3.76) and are adapted to the non-uniform
setting as explained in Sec. 3.6. These values are then inserted into the numerical flux functions and finally, the time update of $\tilde{u}_{i,j}$ is obtained by applying a high-order Runge-Kutta method.

### 3.7.4 Numerical Results in Two Dimensions

Now we want to verify that the results obtained on one-dimensional equidistant and non-equidistant grids can also be obtained in two dimensions. We therefore present different numerical examples validating the concepts introduced in Sec. 3.7.

In Sec. 3.7.4.1 the vortex evolution performed on a two-dimensional Cartesian mesh shows that also in 2D, the limiter function $H_{3L}^{(c)}$ yields third-order accuracy. Then, Sec. 3.7.4.2 presents the two-dimensional advection equation on a non-uniform mesh aligned with the axes. Finally, the double Mach reflection, Sec. 3.7.4.3 and the two-dimensional Riemann problem with four shocks show the excellent performance of $H_{3L}^{(c)}$ using AMR.

All simulations were performed using the third-order accurate strong stability preserving Runge-Kutta (SSP-RK3) time integrator described in [21].

**Figure 3.32** – Example of a non-uniform 2D mesh.
3.7.4.1 2D Vortex Evolution on Cartesian Grids

This problem, originally proposed by Hu [25], describes a two-dimensional vortex evolution on the periodic domain $[-7, 7] \times [-7, 7]$, where the flow is described by the Euler equations. The initial data consists of a mean flow $\rho = u = v = p = 1$, perturbed by

$$
\begin{pmatrix}
\delta \rho \\
\delta u \\
\delta v \\
\delta p
\end{pmatrix} = \begin{pmatrix}
(1 + \delta T)^{1/(\gamma-1)} - 1 \\
-\frac{y}{\gamma} \frac{x^2}{2\pi} e^{0.5(1-r)} \\
\frac{x}{\gamma} \frac{y^2}{2\pi} e^{0.5(1-r)} \\
(1 + \delta T)^{\gamma/(\gamma-1)} - 1
\end{pmatrix}.
$$

(3.84)

The perturbations in density and pressure are expressed in terms of perturbation in temperature, $\delta T$, given by

$$
\delta T = -\frac{(\gamma - 1)\sigma^2}{8\gamma \pi^2} e^{1-r^2},
$$

(3.85)

with $r^2 = x^2 + y^2$, the adiabatic index $\gamma = 1.4$, and the vortex strength $\sigma = 5$. The initial data is also used as a reference solution at time $t = 14$, where it agrees with the exact solution. For the limiter function we need to compute the radius of the asymptotic region, $\alpha$, given by Eq. (3.80). In this case, $\alpha = 7.9$ is used for the simulation. By applying the method on a two-dimensional Cartesian grid, as suggested in Sec. 3.7.1, we obtain the full third order, as shown in Table 3.3. The results compare well with the third-order WENO-Z3 implementation developed by Borges et. al. [5] and further improved by Don and Borges [15]. Both schemes, the $H^{(c)}_{3L}$ limiter function and WENO-Z3, are implemented following the algorithm described in Sec. 3.7.1.

3.7.4.2 2D Advection Equation on Non-Uniform Rectangular Grids

This numerical problem verifies the accuracy of the two-dimensional numerical scheme on non-uniform grids, described in Sec. 3.7.3 and 3.7.1. We consider the two-dimensional linear advection equation with smooth initial conditions

$$
\begin{cases}
u_t + a \, u_x + b \, u_y = 0 \\
u(x, y, 0) = u_0(x, y) = \frac{1}{2} \sin(\pi x) \sin(\pi y).
\end{cases}
$$

(3.86)

The computational domain is set to $\Omega = [-1, 1] \times [-1, 1]$ and the non-uniformity is obtained by Eq. (3.82) with $\delta x = 0.1, c_x = 2, \delta y = 0.1, c_y = 1$. 

83
Table 3.3 – Results for the Vortex evolution problem on a uniform grid.

<table>
<thead>
<tr>
<th>Grid</th>
<th>(|\rho - \rho_{ex}|_1)</th>
<th>EOC</th>
<th>(|\rho - \rho_{ex}|_1)</th>
<th>EOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 \times 64</td>
<td>1.284E-03</td>
<td></td>
<td>1.186E-03</td>
<td></td>
</tr>
<tr>
<td>128 \times 128</td>
<td>2.302E-04</td>
<td>2.48</td>
<td>2.281E-04</td>
<td>2.38</td>
</tr>
<tr>
<td>256 \times 256</td>
<td>3.125E-05</td>
<td>2.88</td>
<td>3.120E-05</td>
<td>2.87</td>
</tr>
<tr>
<td>512 \times 512</td>
<td>3.955E-06</td>
<td>2.98</td>
<td>3.953E-06</td>
<td>2.98</td>
</tr>
<tr>
<td>1024 \times 1024</td>
<td>4.954E-07</td>
<td>3.00</td>
<td>4.954E-07</td>
<td>3.00</td>
</tr>
<tr>
<td>2048 \times 2048</td>
<td>6.194E-08</td>
<td>3.00</td>
<td>6.194E-08</td>
<td>3.00</td>
</tr>
</tbody>
</table>

Applying an advection speed of either \((a,b) = (1,0)\) or \((a,b) = (1,1)\) and the simulation time \(t_{\text{end}} = 2\), the initial condition can be used as exact solution. Thus, the \(L_1\)-error of the numerical solution \(u^{n}_{ij}\) can easily be computed as

\[
\|u^{n}_{ij} - u_0(x_i, y_j)\|_1 = |C_{i,j}| \sum_{i,j} |u^{n}_{ij} - u_0(x_i, y_j)|.
\]

For the simulation, the CFL condition 0.5 has been imposed and for the decision criterion \(\eta\) of the limiter function \(H^{(e)}_{3L,\text{neq}}\), the input value \(\alpha = \pi^2\) is obtained by Eq. (3.80). In order to verify the order of convergence, we carry out simulations with \(N = \{5 \times 5, 10 \times 10, 20 \times 20, 30 \times 30, 50 \times 50\}\) grid cells. The mesh with \(30 \times 30\) grid cells obtained by Eq. (3.82) with \(\delta x = \delta y = 0.1, c_x = 2, c_y = 1\), is depicted in Fig. 3.33a and the solution obtained using \(H^{(e)}_{3L,\text{neq}}\) is shown in Fig. 3.33b. The \(L_1\)-errors and the corresponding empirical orders of convergence are given in Table 3.4. The errors for advection speed \((1,0)\) are by a mean factor of 0.7 better than the errors of the solutions advected in diagonal direction \((1,1)\). Nevertheless, both simulations yield third-order accuracy; see Table 3.4.

### 3.7.4.3 Double Mach Reflection

In this section we apply the limiter function on a Cartesian grid with AMR, as described in Sec. 3.7.2. The test case consists of the double Mach reflection problem proposed by Woodward and Colella [76]. It describes a Mach 10 shock reflection off a 30-degree wedge. The computational domain is the rectangle
(a) Non-uniform mesh with
30 × 30 grid cells.

(b) Solution obtained with $H_{3L}$.

**Figure 3.33** – Computation of Eq. (3.86) on a non-uniform mesh with 30 × 30 grid cells.

| Grid  | $||u - u_{ex}||_1$  | EOC | $||u - u_{ex}||_1$  | EOC |
|-------|---------------------|-----|---------------------|-----|
| 5 × 5 | 3.886E-01           |     | 5.671E-01           |     |
| 10 × 10 | 1.536E-01        | **1.34** | 2.503E-01        | **1.18** |
| 20 × 20 | 2.621E-02        | **2.55** | 6.626E-02        | **1.92** |
| 30 × 30 | 8.304E-03        | **2.83** | 2.088E-02        | **2.85** |
| 50 × 50 | 1.856E-04        | **2.93** | 4.410E-03        | **3.04** |

**Table 3.4** – $L_1$-errors of $H_{3L}^{(c)}$ and corresponding empirical order of convergence (EOC).

85
Figure 3.34 – AMR computation of the Double Mach Reflection problem computed with $H^{(c)}_{3L}$. The grid resolution corresponds to a grid with $288 \times 96$ mesh cells on the coarsest level and up to $4608 \times 1536$ mesh cells on the finest level.

[0, 3] \times [0, 1]. To obtain the same resolution for both, the $x$– and $y$–direction, each block contains $36 \times 12$ mesh cells. We set level 3 as the coarsest level and allow up to 4 additional refinements, thus the finest level corresponds to a discretization with $4608 \times 1536$ mesh cells. The refinement threshold is set to $\delta_0 = 2000$. Due to the constant initial date, $\alpha$ turns out to be 0, cf. Eq. (3.80). Therefore, the combined limiter function $H^{(c)}_{3L}$, Eq. (3.41), reduces to $H_{3L}$, see Eq. (3.25).

Fig. 3.34 shows the result of the simulation using the third-order limiter function $H_{3L}$ at time $t_{\text{end}} = 0.2$, including the block structure. A close–up view of the Mach stem region is shown in Fig. 3.35. The computations were performed with four (Fig. 3.35a and 3.35b) and five (Fig. 3.35c and 3.35d) levels of refinement.

For comparison, we also show the results of a third-order WENO-Z reconstruction on the left hand side. In direct comparison the $H^{(c)}_{3L}$ scheme produces more roll ups in the inner region. This is a desired feature since the slip line is physically instable, indicating that the scheme with $H_{3L}$ introduces less numerical viscosity than WENO-Z3.

3.7.4.4 2D Riemann Problem

The next testcase we consider is a configuration of four interacting shocks in the domain $[0, 1]^2$. The initial values have the form
Figure 3.35 – AMR computation of the Double Mach Reflection problem.
(a) Computed with WENO-Z3 on a uniform grid.
1024 × 1024 mesh cells.

(b) Computed with $H_{3L}^{(c)}$ on a uniform grid.
1024 × 1024 mesh cells.

(c) Computed with WENO-Z3 on a uniform grid.
2048 × 2048 mesh cells.

(d) Computed with $H_{3L}^{(c)}$ on a uniform grid.
2048 × 2048 mesh cells.

Figure 3.36 – Results of the 2D Riemann problem at final time $t_{end} = 0.3$. 

88
\[(\rho, p, u, v)(x, y, 0) = \begin{cases} 
(1.5, 1.5, 0.0, 0.0) & x > 0.5, y > 0.5 \\
(0.5323, 0.3, 1.2060, 0.0) & x < 0.5, y > 0.5 \\
(0.1380, 0.029, 1.2060, 1.2060) & x < 0.5, y < 0.5 \\
(0.5323, 0.3, 0.0, 1.2060) & x > 0.5, y < 0.5 
\end{cases} \] (3.87)

This testcase was originally proposed by Schulz-Rinne [49, 50] along with several other configurations of 2D Riemann problems for the Euler equations of inviscid, compressible isentropic and polytropic flow. Due to the constant initial data we set here \(\alpha = 0\) for the limiter \(H^{(c)}_{3L}\), see (3.80).

Fig. 3.36 shows the results at final time \(t_{end} = 0.3\). The results obtained by applying the limiter function \(H^{(c)}_{3L}\) are compared to results computed with the third-order WENO-Z reconstruction. As for the double Mach reflection, Sec. 3.7.4.3, the scheme with \(H^{(c)}_{3L}\) introduces less numerical viscosity than WENO-Z3.
Chapter 4

Part II – Riemann Solvers

In finite volume methods, integrating conservation laws over a control volume leads to a formulation which requires the evaluation of local Riemann problems at each cell interface, see Chapter 2 for more details. The initial states for these problems are typically given by the left and right adjacent cell values. Since these local Riemann problems have to be solved many times in order to find the numerical solution, the Riemann solver is a building block of the finite volume method. Over the last decades, many different Riemann solvers were developed, see e.g. [61] for a broad overview. The main challenges are the need for computational efficiency and easy implementation, while at the same time, accurate results without artificial oscillations need to be obtained.

Riemann solvers can be classified into complete and incomplete schemes, depending on whether all present characteristic fields are considered in the model or not. According to this classification, upwind, Godunov’s method and Roe’s scheme, are complete [33]. These schemes yield monotone results, however, an evaluation of the eigensystem of the flux Jacobian is needed. Especially for large systems this characteristic decomposition is expensive to compute and in some cases an analytic expression is not available at all. Nevertheless, using Roe’s scheme, all waves are well-resolved and it typically yields the best resolution of the Riemann wave fan. In order to reduce computational cost while keeping high resolution, there have been many attempts to approximate the upwind scheme without solving the eigenvalue problem, see e.g. [10, 13, 67] and references therein.
In this work, we are interested in incomplete Riemann solvers. In comparison to complete solvers, they need less characteristic information and are easier to implement. However, they contain more dissipation and thus, yield lower resolution, especially of slow waves. Nevertheless, in many test cases, these Riemann solvers may be sufficient to obtain good results, especially if the system contains only fast waves.

The rest of Chapter 4 is structured as follows: Sec. 4.1 reviews some well-known Riemann solvers. In Sec. 4.2 we discuss some hybrid Riemann solvers, i.e. solvers which are constructed as weighted combinations of the ones presented in the previous section. In Sec. 4.3 we describe how to implement the described flux functions without knowing the flux Jacobian, i.e. a Jacobian-free implementation. Then, in Sec. 4.4 we present the new family of Riemann solvers and discusses construction and parameter choices. The numerical results of Sec. 4.5 underline the excellent performance of the new solvers.

To a large extend, Chapter 4 relies on articles published with coauthors. Firstly, [47] which is published with Mariia Astrakhantceva and Manuel Torrilhon in the proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS). Secondly, the full paper [43], published together with Manuel Torrilhon in SIAM Journal on Scientific Computing. While the idea of the family of Riemann solvers and a basic sketch were given by Manuel Torrilhon, Mariia Astrakhantceva, who was writing her master thesis at our institute, was doing the preliminary work and I was working out the detailed formulations given in Sec. 4.4. It was necessary to test the formulation of the new solvers with respect to possible weaknesses (e.g. for smallest eigenvalue = - largest eigenvalue) and rewrite certain formula. Furthermore, I implemented the numerical schemes and performed all numerical test cases presented in Sec. 4.5.
4.1 Classical Riemann Solvers

In this section we will recall some well-known Riemann solvers, which are necessary for the development of the new family of hybrid Riemann solvers.

Let us first point out that the numerical flux function \( \hat{f}(U_L, U_R) \), introduced in Chapter 2, can be written in the general form

\[
\hat{f}(U_L, U_R) = \frac{1}{2} \left( f(U_L) + f(U_R) \right) - \frac{1}{2} D(U_L, U_R) \left( U_R - U_L \right)
\]  

(4.1)

with the dissipation matrix \( D \). This matrix depends on the left and right adjoint states of the interface of interest i.e. \( D = D(U_L, U_R) \).

First, we note that the numerical flux function (4.1) and thus the resulting scheme is completely dictated by the dissipation matrix \( D \). Therefore, we break down the following discussion to the comparison of the dissipation matrices \( D \) of the schemes.

Since \( D = D(U_L, U_R) \), the matrix also depends on the characteristics of the flux Jacobian. Since we are interested in hyperbolic problems, the flux Jacobian can be diagonalized to read \( \tilde{\mathbf{A}}(U) = R(U) \Lambda(U) R(U)^{-1} \). Here, \( \Lambda(U) \) is the eigenvalue matrix \( \Lambda(U) = \text{diag}(\lambda_1(U), \ldots, \lambda_N(U)) \), \( \lambda_1 < \lambda_2 < \ldots < \lambda_N \) and \( R(U) \) the corresponding eigenvector matrix. With this knowledge in hand, it can be shown that

\[
\frac{\Delta t}{\Delta x} D(\tilde{\mathbf{A}}) = R \frac{\Delta t}{\Delta x} \text{diag}(\tilde{d}(\lambda_1), \ldots, \tilde{d}(\lambda_N)) R^{-1} = R \text{diag}(d(\nu_1), \ldots, d(\nu_N)) R^{-1},
\]  

(4.2)

holds true for all dissipation matrices discussed in this thesis. The eigenvector matrix \( R \) remains the same, independent of \( D \). It can easily be seen that Eq. (4.2) holds, since all dissipation matrices considered in this work are either of polynomial nature, given by the absolute value of the flux Jacobian, or linear combinations of these functions. More specifically, for the dissipation matrices discussed in this thesis it holds that

\[
D(\tilde{\mathbf{A}}) = \alpha_0 \tilde{\mathbf{A}}^0 + \alpha_1 \tilde{\mathbf{A}}^1 + \alpha_2 \tilde{\mathbf{A}}^2 + \alpha_3 |\tilde{\mathbf{A}}|
\]
with coefficients $\alpha_i \in \mathbb{R}$, $i = 1, \ldots, 4$, which might be zero. It then follows from Eq. (4.2) that

$$\frac{\Delta t}{\Delta x} D(\tilde{A}) = \frac{\Delta t}{\Delta x} R \left( \alpha_0 I + \alpha_1 \Lambda + \alpha_2 \Lambda^2 + \alpha_3 |\Lambda| \right) R^{-1}$$

$$= R \frac{\Delta t}{\Delta x} \text{diag}(\alpha_0 + \alpha_1 \mu_1 + \alpha_2 \mu_1^2 + \alpha_3 |\mu_1|, \ldots, \alpha_0 + \alpha_1 \mu_N + \alpha_2 \mu_N^2 + \alpha_3 |\mu_N|) R^{-1}$$

$$= R \text{diag}(d(\mu_1), \ldots, d(\mu_N)) R^{-1},$$

Here, $\mu_i = \lambda_i \Delta t / \Delta x$ is the dimensionless characteristic speed and $d(\mu)$ is the dimensionless scalar dissipation function.

Throughout the whole thesis, $d(\mu)$ denotes the scalar, non-dimensional dissipation function. It can be understood as the effect of the dissipation matrix $D$ on the eigenvalues of $\tilde{A}$ as shown in Eq. (4.2).

In order to obtain $L^2$ stability, $d(\mu) \geq \mu^2$ is required. If we want to restrict ourselves to monotone schemes, the dissipation function has to lie above the absolute value function $|\mu|$. Below this bound, the scheme is non-monotone, which means that undesired oscillations may occur at discontinuities. In general we want the dimensionless scalar dissipation function to fulfill $d(\mu) \geq |\mu|$.

### 4.1.1 Upwind Godunov solver

The upwind method is a complete Riemann solvers, respecting all $N$ waves resulting from System (2.1). For each local Riemann problem in the finite volume setting, the local wave propagation speeds, which for linear problem are given by the eigenvalues, determine which one-sided spatial derivative to choose. For a linear system, where we can write $\partial_t f(U) = A \partial_x U$ with a constant matrix $A \in \mathbb{R}^{N \times N}$, the upwind scheme reads

$$U^{n+1}_i = U^n_i - \frac{\Delta t}{\Delta x} \left( A^- \cdot (U_{i+1} - U_i) + A^+ \cdot (U_i - U_{i-1}) \right).$$

Here, $A^\pm$ are the matrices containing only the positive and negative eigenvalues, respectively, i.e.

$$A^\pm = R \Lambda^\pm R^{-1}$$
with $\lambda_1 < \lambda_2 < \ldots < \lambda_r < 0 < \lambda_{r+1} < \ldots < \lambda_N$

$$\Lambda^- = \text{diag} (\lambda_1, \ldots, \lambda_r, 0, \ldots, 0), \quad (4.4b)$$

$$\Lambda^+ = \text{diag} (0, \ldots, 0, \lambda_{r+1}, \ldots, \lambda_N). \quad (4.4c)$$

Since $A^+ A^- = A$ and $A^+ - A^- = |A|$, the numerical flux function of the upwind scheme can be written in form of Eq. (4.1)

$$\hat{f}_{\text{up}} = \frac{1}{2} A (U_i + U_{i+1}) + \frac{1}{2} (A^- - A^+) (U_{i+1} - U_i) \quad (4.5)$$

$$= \frac{1}{2} A (U_i + U_{i+1}) - \frac{1}{2} |A| (U_{i+1} - U_i). \quad (4.6)$$

The dissipation matrix can be easily identified as

$$D_{\text{up}} = |A|. \quad (4.7)$$

Following Eq. (4.2), the transformation from $D_{\text{up}}$ to the scalar, non-dimensional dissipation function yields $d_{\text{up}}(\nu) = |\nu|$. For non-linear systems with general flux function $f$, it is in general not possible to find a matrix $A$ such that $\partial_x f(U) = A \partial_x U$ holds true. In this case, splittings analogous to Eq. (4.3) can be found which respect the wave propagation to the left and to the right. A popular method is Godunov’s method [20]. For linear systems, this method reduces to Eq. (4.3). In the non-linear case, Godunov’s method requires the exact solution of local Riemann problems at each cell interface. The Godunov scheme can be adapted by using a so-called Roe Matrix $\tilde{A}(U_L, U_R)$, satisfying (amongst other properties) $\tilde{A}(U_L, U_R)(U_R - U_L) = F(U_R) - F(U_L)$, cf. [38]. The dissipation matrix of Roe’s scheme is then given by

$$D_{\text{Roe}} = |\tilde{A}| \iff d_{\text{Roe}}(\nu) = |\nu|. \quad (4.8)$$

In the following, we will use the notation $\tilde{A}$, even though in some cases, this matrix might simplify to $A$.

The upwind Godunov and Roe scheme require a characteristic decomposition of $\tilde{A}$, thus, they are complete Riemann solvers. Recalling that every dissipation function below the absolute value function is non-monotone [19], we can conclude that the upwind method has the minimal dissipation while still being monotone. Thus it is the optimal scheme in this sense. The drawback is that the characteristic decomposition might be expensive to compute, especially for large systems and in some cases an analytic expression is not available at all.
4.1.2 Lax-Friedrichs and Rusanov solver

The numerical flux function of the Lax-Friedrichs method is given by

\[
\hat{f}_{\text{LF}}(U_L, U_R) = \frac{1}{2} \left( f(U_L) + f(U_R) \right) - \frac{1}{2} \frac{\Delta x}{\Delta t} (U_R - U_L). \tag{4.9}
\]

The dissipation matrix can be identified to read

\[
D_{\text{LF}}(U_L, U_R) = \frac{\Delta x}{\Delta t} I \iff d_{\text{LF}}(\nu) = 1. \tag{4.10}
\]

It can be directly seen from Eq. (4.10) that this method does not need any information of the flux Jacobian or its eigenvalues apart from the CFL condition. Thus, the Lax-Friedrichs scheme is computationally very efficient. Another advantage of the Lax-Friedrichs method is that it does not cause oscillations at discontinuities. However, the scheme is known to smear out discontinuities.

The Rusanov scheme \[41\], also referred to as local Lax-Friedrichs scheme, explicitly takes into account the globally fastest eigenvalue of the system. The dissipation matrix reads

\[
D_{\text{LLF}}(U_L, U_R) = \max \{ \bar{\lambda}(U_L), \bar{\lambda}(U_R) \} I. \tag{4.11}
\]

In comparison to the Lax-Friedrichs scheme, the Rusanov scheme reduces the dissipation added to the solution. Therefore, discontinuities are better resolved.

4.1.3 Lax-Wendroff solver

The Lax-Wendroff scheme \[32\] for linear systems is characterized by the numerical flux function

\[
\hat{f}_{\text{LW}}(U_L, U_R) = \frac{1}{2} \left( f(U_L) + f(U_R) \right) + \frac{1}{2} \frac{\Delta t}{\Delta x} \tilde{A}^2 (U_L - U_R). \tag{4.12}
\]

This yields the dissipation matrix

\[
D_{\text{LW}} = \frac{\Delta t}{\Delta x} \tilde{A}^2 \iff d_{\text{LW}}(\nu) = \nu^2. \tag{4.13}
\]

For non-linear flux functions \( f \), Richtmyer’s two-step version of the Lax-Wendroff scheme \[37\] yields a Jacobien-free way of implementing the numerical flux function

\[
\hat{f}_{\text{LW}}(U_L, U_R) = f \left( \frac{1}{2} (u_L + u_R) - \frac{1}{2} \frac{\Delta t}{\Delta x} (f(u_R) - f(u_L)) \right). \tag{4.14}
\]
Due to the small amount of viscosity of this solver, discontinuities are approximated with steep gradients. However, the method is known to cause oscillations due to dispersion \[33\]. See also Fig. 4.1 which show that the Lax-Wendroff method is non-monotone.

### 4.1.4 HLL solver

The method introduced by Harten, Lax and van Leer \[23\] belongs to the category of incomplete solvers. The Riemann problem is solved considering only the fastest and slowest wave speeds of the system; \( \lambda_{\text{min}} \) and \( \lambda_{\text{max}} \), respectively. The numerical flux function is most often found in the literature in the form

\[
\hat{f}_{\text{HLL}}(U_L, U_R) = \frac{\lambda_R f(u_L)\lambda_L f(u_R) + \lambda_L \lambda_R (u_R u_L)}{\lambda_R \lambda_L}.
\]  

(4.15)

Here, \( \lambda_L = \lambda_{\text{min}}(U_L) \) is the minimal eigenvalue of the left state and \( \lambda_R = \lambda_{\text{max}}(U_R) \) the maximal eigenvalue of the right state of the system. A simple calculation leads to the formulation

\[
\hat{f}_{\text{HLL}}(U_L, U_R) = \frac{1}{2} (f(u_L) + f(u_R)) - \frac{1}{2} (a_0 \Delta x \frac{\Delta t}{\Delta t} I (u_R - u_L) + a_1 (f(u_R) - f(u_L))),
\]  

(4.16a)

with

\[
a_0 = -\frac{|\nu_L| \nu_R - |\nu_R| \nu_L}{\nu_L - \nu_R} = \frac{\Delta t}{\Delta t} \frac{|\lambda_L| \lambda_R - |\lambda_R| \lambda_L}{\lambda_R - \lambda_L},
\]  

(4.16b)

\[
a_1 = \frac{|\nu_L| - |\nu_R|}{\nu_L - \nu_R} = \frac{|\lambda_R| - |\lambda_L|}{\lambda_R - \lambda_L}.
\]  

(4.16c)

This yields the dissipation matrix and function

\[
D_{\text{HLL}}(U_L, U_R) = a_0 \frac{\Delta x}{\Delta t} I + a_1 \tilde{A} \quad \Leftrightarrow \quad d_{\text{HLL}}(\nu) = a_0 + a_1 \nu.
\]  

(4.17)

The dimensionless dissipation function is given by the dimensionless scalar dissipation function is given by

\[
d_{\text{HLL}}(\nu) = a_0 + a_1 \nu
\]  

(4.18a)

with

\[
a_0 = -\frac{|\nu_L| \nu_R - |\nu_R| \nu_L}{\nu_L - \nu_R},
\]  

(4.18b)

\[
a_1 = \frac{|\nu_L| - |\nu_R|}{\nu_L - \nu_R}.
\]  

(4.18c)
With the definition of $a_0$ and $a_1$, we can rewrite the dissipation matrix as

$$D_{\text{HLL}}(U_L, U_R) = a_0 \frac{\Delta x}{\Delta t} I + a_1 \tilde{A} - \frac{\lambda_L |\lambda_R| - |\lambda_R| \lambda_L}{\lambda_L - \lambda_R} I. \quad (4.19)$$

### 4.1.5 Comparison of Classical Riemann Solvers

Now that all classical solvers reviewed in Sec. 4.1 are written in scalar, non-dimensional form, they can easily be compared. Fig. 4.1 shows the scalar dissipation of the methods depending on $\nu \in \{-1, 1\}$. This restriction is due to the CFL condition which requires that $|\lambda_{\text{max}} \Delta t/\Delta x| \leq 1$ is fulfilled by any $\nu$ based on the eigenvalues of the flux Jacobian. The shaded region denotes the part which is non-monotone. The upwind scheme (and Godunov, and Roe scheme, see discussion above) marks the border between the monotone and non-monotone region. Thus, this scheme has the smallest amount of dissipation while it is still monotone. The other extreme among the monotone schemes is the Lax-Friedrichs method. This scheme induces more dissipation than all other methods considered in this work. This can also be seen in Fig. 4.1, where it becomes clear that the Lax-Friedrichs dissipation function $d_{\text{LF}}(\nu)$ has larger values than all other dissipation functions. The Rusanov (or local Lax-Friedrichs) scheme reduces the dissipation. However, in case fast and slow traveling waves exist in the system, especially the slower waves for which $\nu \approx 0$, suffer from too much dissipation. Consider e.g. the one-dimensional Euler equations with three emerging eigenvalues $v \pm c$ and $v$, where $v$ is the velocity and $c$ the speed of sound. In case of small velocities $v$, the amount
of dissipation added to the solution by the Rusanov dissipation function is only little more than by the upwind scheme for the fastest and slowest eigenvalues $v \pm c$. However, at the contact discontinuity, traveling with speed $v \approx 0$, it can be seen from Fig. 4.1 that the amount of dissipation added by the Rusanov scheme is a lot more than by the upwind scheme. This explains that solution obtained by the Lax-Friedrichs and Rusanov method have smeared out discontinuities. The Lax-Wendroff scheme on the other side of the spectrum has very little dissipation and causes oscillations because it is non-monotone. Thus, concerning the balance between little dissipation and no spurious oscillations, the upwind scheme seems to be the scheme of choice. However, this scheme necessitates the decomposition of the eigensystem. In some cases no analytic form of the eigensystem is available but an estimate of the fastest and slowest eigenvalues can be obtained. With these two values, the HLL solver can be computed. Compared to solvers which only use the globally fastest wave speed, HLL reduces the amount of dissipation. This is true especially for faster waves close to the globally fastest signal velocities, where HLL intersects with the upwind Godunov method, see Fig. 4.1. Note that the Lax-Wendroff scheme is not the only scheme lying outside the monotone region. For the Rusanov and HLL schemes, the wave speeds have to lie within the region defined by the slowest and fastest actual wave speeds, here denoted by $\nu_{\text{min}}$ and $\nu_{\text{max}}$. Therefore, outside this range, the dissipation functions are depicted using dashed lines, indicating that these parts are not of interest as long as the wave speeds are computed or approximated correctly.

4.2 Hybrid Riemann Solvers

In this section, we introduce some hybrid Riemann solvers that can be found in the literature. By hybrid we mean solvers which can be constructed using weighted combinations of the classical solvers introduced in Sec. 4.1.

4.2.1 FORCE

We first consider the First Order Centered (FORCE) scheme, introduced by Toro et al. [58, 59]. This scheme can be viewed as a monotone version of the
Lax-Wendroff method and the numerical flux function can be expressed as the average of the Lax-Friedrichs and the Lax-Wendroff method. Consequently, the dissipation matrix and function are given by

\[
D_{\text{FORCE}} = \frac{1}{2} \frac{\Delta x}{\Delta t} \left( \frac{\Delta t^2}{\Delta x^2} A^2 + I \right) \quad \Leftrightarrow \quad d_{\text{FORCE}}(\nu) = \frac{1}{2} \left( \nu^2 + 1 \right). \tag{4.20}
\]

Note that this solver needs as little characteristic information as the Lax-Friedrichs scheme. The only addition is the extra flux evaluation required by the Lax-Wendroff method.

### 4.2.2 MUSTA

In [57] and [60], a multi stage flux called MUSTA has been introduced. This flux function is based on the repetition of a simple flux function in order to resolve the Riemann fan and obtain high-resolution solutions of the Riemann problem. The general MUSTA\(_k\) flux consists of \(k\) repetitions. Here, we only consider the schemes for \(k \in \{0, 1\}\). It turns out that MUSTA\(_0\) is the FORCE scheme, Eq. (4.20), as suggested in [57, 60]. For \(k = 1\) the dissipation matrix of the scheme is given by

\[
D_{\text{MUSTA}_1} = \frac{1}{4} \frac{\Delta x}{\Delta t} I + \frac{\Delta t}{\Delta x} \tilde{A}^2 - \frac{1}{4} \left( \frac{\Delta t}{\Delta x} \right)^3 \tilde{A}^4 \quad \Leftrightarrow \quad d_{\text{MUSTA}_1} = \frac{1}{4} + \nu^2 - \frac{1}{4} \nu^4. \tag{4.21}
\]

It is interesting to realize that \(d_{\text{MUSTA}_1}\) slightly drops below the absolute value function for larger wave speeds, see Fig. 4.2. This feature will be discussed in more detail in Sec. 4.4.1.

### 4.2.3 HLLX

In this section we introduce a solver which also includes a quadratic term in the dissipation matrix. This means, it has the same number of flux evaluations as FORCE and MUSTA\(_1\).
4.2.3.1 The $P_2$-Dissipation Function

We recall three requirements which have been proposed by Degond et al. [13]. The resulting monotone Riemann solver, named $P_2$, is based on a quadratic dissipation function, fully determined by

$$d_{P_2}(\nu_{\text{min}}) = d_{up}(\nu_{\text{min}}) = |\nu_{\text{min}}|,$$

$$d_{P_2}(\nu_{\text{max}}) = d_{up}(\nu_{\text{max}}) = |\nu_{\text{max}}|,$$

and

$$d'_{P_2}(\hat{\nu}) = d'_{up}(\hat{\nu}) = \text{sign}(\hat{\nu}), \quad \hat{\nu} = \begin{cases} \nu_{\text{max}} & \text{if } |\nu_{\text{max}}| \geq |\nu_{\text{min}}| \\ \nu_{\text{min}} & \text{if } |\nu_{\text{min}}| > |\nu_{\text{max}}| \end{cases}.$$  

The first two requirements indicate that the dissipation matrix matches the absolute value function at the slowest and fastest wave speeds. This means that the numerical flux function at these points equals the upwind flux. The third condition is that the slope at the globally fastest wave speed has to match the one of the absolute value function. Note that $\hat{\nu}$ includes the sign of the fastest wave, compared to $\bar{\nu}$, for which $\bar{\nu} = |\hat{\nu}|$ holds true.

The dissipation function automatically fulfills $d_{P_2}(\nu) \geq |\nu|$ for $\nu \in [\nu_{\text{min}}, \nu_{\text{max}}]$, which means that it is monotone in this region. Even though the requirements (4.22) have been proposed in [13], neither the dissipation matrix nor the numerical flux function of $P_2$ have been explicitly stated.

4.2.3.2 The HLLX-Dissipation Function

We now present a simple way of implementing $P_2$, based on the flux functions of Lax-Friedrichs (LF), HLL, and Lax-Wendroff (LW), and call this Riemann solver...
HLLX. The dissipation function consists of a constant, an affine linear, and a quadratic part. Thus, it can be expressed as the weighted average

\[ d_{\text{HLLX}}(\nu) = \alpha_0 d_{\text{LF}}(\nu) + \alpha_1 d_{\text{HLL}}(\nu) + \alpha_2 d_{\text{LW}}(\nu), \quad (4.23a) \]

where the coefficients are given by

\[ \alpha_0 = \alpha |\nu_{\text{min}} \cdot \nu_{\text{max}}|, \quad (4.23b) \]

\[ \alpha_1 = 1 - \alpha (|\nu_{\text{max}}| + |\nu_{\text{min}}|), \quad (4.23c) \]

\[ \alpha_2 = \alpha, \quad (4.23d) \]

and

\[ \alpha = \frac{\nu_{\text{max}} - \nu_{\text{min}} - |\nu_{\text{max}}| - |\nu_{\text{min}}|}{(\nu_{\text{max}} - \nu_{\text{min}})^2}. \quad (4.23e) \]

An even simpler but not as demonstrative way of defining the dissipation function with only one coefficient is

\[ d_{\text{HLLX}}(\nu) = d_{\text{HLL}}(\nu) + \alpha (\nu - \nu_{\text{min}})(\nu - \nu_{\text{max}}), \quad (4.24) \]

with \( \alpha \), as defined in Eq. (4.23e).

Based on Eq. (4.24), it is possible to write the dissipation matrix as

\[ D_{\text{HLLX}}(U_L, U_R) = D_{\text{HLL}}(U_L, U_R) + \alpha \Delta t \Delta x (\tilde{A} - \lambda_{\text{min}} I)(\tilde{A} - \lambda_{\text{max}} I). \quad (4.25) \]

This directly leads to the corresponding numerical flux function

\[ \hat{f}_{\text{HLLX}}(U_L, U_R) = \hat{f}_{\text{HLL}} - \frac{\alpha}{2} \Delta t \Delta x (\tilde{A} - \lambda_{\text{min}} I)(\tilde{A} - \lambda_{\text{max}} I)(U_R - U_L). \quad (4.26) \]

### 4.3 Jacobian-Free Implementation

In the formulation above, it is necessary to know the flux Jacobian or Roe matrix \( \tilde{A} \), in order to calculate the dissipation matrix \( D(U_L, U_R) \). In case of the Euler equations, an explicit expression of the flux Jacobian is available, however, for larger systems of conservation laws this might be difficult to compute or implement. In these cases, a Jacobian-free implementation is desirable. We now describe the implementation of the cases treated in this thesis, which are scaled...
versions of $\tilde{A}$ and $\tilde{A}^2$. For a generalization of the Jacobian-free implementation, the interested reader is referred to [67, Sec. 3.2].

In the flux formula $\hat{f}(U_L, U_R) = \frac{1}{2} (f(U_L) - f(U_R)) + \frac{1}{2} D(U_L, U_R) (U_R - U_L)$, the dissipation matrix is always multiplied by the state difference $\Delta U = U_R - U_L$, so that the Jacobian matrix always appears as a matrix vector multiplication with $\Delta U$. We exploit this fact by using the finite difference formulation

$$
\hat{A} \Delta U = \lim_{\varepsilon \to 0} \frac{f(\overline{U} + \varepsilon \Delta U) - f(\overline{U})}{\varepsilon} \tag{4.27}
$$

with the average value $\overline{U} = 0.5 (U_L + U_R)$. In the following, assuming $\varepsilon \ll 1$, we use the implementation

$$
\hat{A} \Delta U = \frac{f(\overline{U} + \varepsilon \Delta U) - f(\overline{U})}{\varepsilon}, \quad \hat{A}^2 \Delta U = f\left(\overline{U} + \varepsilon \frac{f(\overline{U} + \varepsilon \Delta U) - f(\overline{U})}{\varepsilon}\right) - f(\overline{U}) \tag{4.28}
$$

Another option is to note that the way of writing the dissipation function (4.23) directly leads to a simple form for the numerical flux function. In order to implement $\hat{f}_{HLLX}$ we do not actually need to compute the dissipation matrix $D_{HLLX}$ based on Eq. (4.23a). We rather use the fact, that most users already implemented Lax-Friedrichs, HLL, and Lax-Wendroff. Defining $\tilde{f} = 0.5(f(U_L) + f(U_R))$, we can rewrite the numerical flux functions (4.1) of these three classical Riemann solvers to extract their dissipation matrices. Altogether this yields the numerical flux function

$$
\hat{f}_{HLLX}(U_L, U_R) = \tilde{f} + \alpha_0 (\hat{f}_{LF} - \tilde{f}) + \alpha_1 (\hat{f}_{HLL} - \tilde{f}) + \alpha_2 (\hat{f}_{LW} - \tilde{f}) \tag{4.29a}
$$

which can equivalently be written as

$$
= \hat{f}_{HLL} + \alpha \left\{ |\nu_{\min}| \cdot |\nu_{\max}| (\hat{f}_{LF} - \tilde{f}) - (|\nu_{\max}| + |\nu_{\min}|) (\hat{f}_{HLL} - \tilde{f}) + (\hat{f}_{LW} - \tilde{f}) \right\}, \tag{4.29b}
$$

with $\alpha, \alpha_i, i = 0, 1, 2$ as in Eq. (4.23). This formulation does not include the Jacobian matrix $\tilde{A}$ and can therefore be used for any (large) system of hyperbolic conservation laws.
4.4 HLLX$\omega$ - a Family of Hybrid Riemann Solvers

The aim of this section is to construct a new family of Riemann solvers, called HLLX$\omega$, which require as little information as HLL but are less dissipative. Additionally, the solvers should be computationally not more or only marginally more expensive than HLL. Thus, we allow for one more flux evaluation, which means that the new solvers are based on a quadratic dissipation function. Their dissipation functions shall be closer to the absolute value function, \textit{i.e.} the upwind scheme, for all waves $\lambda_i$ of the hyperbolic system, and thus for all $\nu_i$. This means, that the family of solvers yields even less dissipation than HLLX. The new schemes require as little information as HLL and HLLX, namely the globally slowest and fastest characteristic waves of the system, $\lambda_{\text{min}}$ and $\lambda_{\text{max}}$. Additionally, HLLX$\omega$ only needs the same number of flux evaluations as HLLX, namely two. Since we do not want to increase neither the number of input information, nor the number of flux evaluations, we lower the dissipation function by a certain amount. This amount is described by a parameter $\omega \in [0, 1]$, which determines the monotonicity behavior of the solver. For $\omega = 0$ we recover the monotone HLLX solver, and for $\omega = 1$, the non-monotone Lax-Wendroff solver.

All intermediate members of the HLLX$\omega$ family are slightly non-monotone for a certain range of waves. However, we show in this section that under some mild assumptions, the results do not show spurious oscillations.

There are many attempts to approximate the absolute value function by polynomial or rational functions \cite{9, 10, 13, 67}. In this work, we focus on simple yet efficient solvers in the style of HLL with maximal two flux evaluations. Thus, we do not have the ambition to compete with solvers that approximate the upwind scheme on a high level. On the other hand, such schemes are generally expensive in terms of computational costs, such that the use of a simpler scheme, such as HLLX$\omega$, might be more useful.
4.4.1 Beyond Monotonicity

Before we introduce this family of Riemann solvers, we state and validate some observations made by Torrilhon [67]. Firstly, it was perceived that the MUSTA fluxes introduced by Toro [57] slightly drop below the upwind flux, which means that they do not fully lie in the monotonicity preserving region, see also Fig. 4.2 and Sec. 4.2.2. Therefore, as expected, the numerical solutions obtained with MUSTA fluxes show some non-monotone behavior. However, this behavior is far from the oscillations created by the Lax-Wendroff scheme. Additionally, the observed oscillations of MUSTA solutions decay in time and disappear after a certain number of time steps, cf. [67, Fig. 5, p. A2084]. These results are essentially independent of the grid size and were observed for dissipation functions which only slightly drop below the absolute value function. Let us introduce the dissipation function

\[ d_\omega(\nu) = \omega d_{LW}(\nu) + (1 - \omega) d_{UP}(\nu) \]  

\[ \Leftrightarrow D_\omega(U_L, U_R) = \omega D_{LW}(U_L, U_R) + (1 - \omega) D_{UP}(U_L, U_R), \ \omega \in [0, 1]. \]  

For \( \omega = 0 \) we recover the monotone upwind scheme \( d_{\omega=0}(\nu) = d_{UP}(\nu) \) and for \( \omega = 1 \), \( d_{\omega=1}(\nu) = d_{LW}(\nu) \) holds true. This can be seen in Fig. 4.3 which shows \( d_\omega(\nu) \) for \( \omega \in \{0.0, 0.3, 0.7, 1.0\} \).

The aim of this section is to study the monotonicity behavior of \( d_\omega(\nu) \) and produce similar effects to those found in [67]. We therefore investigate solutions of the scheme based on Eq. (4.30) for different values of \( \omega \). We consider the scalar advection equation \( u_t + u_x = 0 \) and use as initial condition the sign function, \( \text{sgn}(x) \) on the interval \( x \in [-1, 1] \). The jump evolves with time on a grid with \( n = 200 \) grid cells until \( T_{end} = 0.25 \). The CFL number is set to \( \bar{\nu} = 0.5 \), which shows the maximal deviation of Lax-Wendroff from Upwind, cf. Fig 4.3. For different values of \( \omega \), we analyze the test case with the flux function resulting from Eq. (4.1) with (4.30).

The numerical results for all tested values of \( \omega \) are shown in Fig. 4.4b. It can be easily seen, that \( \omega = 0 \) and \( \omega = 1 \) correspond to the upwind and the Lax-Wendroff schemes. That is, for \( \omega = 1 \) we can observe the well-known oscillations.
As $\omega$ decreases, the oscillations also decrease and for $\omega \lesssim 0.5$ no oscillations seem to be apparent in Fig. 4.4b. For further analysis, the maximum value of the solutions $u$ as a function of the number of time steps is shown in Fig. 4.4a, where 50 time steps correspond to $T_{\text{end}} = 0.25$. Here, it can be seen that the oscillations of the Lax-Wendroff scheme do not decrease over time, whereas the Upwind scheme does not oscillate at any time. For the mixed schemes, a certain amount of weight needs to be given to the monotone Upwind scheme to make sure that the oscillations decrease in time. Now it can be seen what was not clearly visible in Fig. 4.4b, namely that oscillations only completely decrease within 50 time steps for $\omega \to 0$. Note that these results are essentially independent of the grid size.

For the rest of this thesis we will assume that $\omega \leq 0.4$ is sufficiently small to diminish oscillations to an amount which can be considered "vanished". This assumption is tested and verified in the numerical experiments in Sec. 4.5.

### 4.4.2 Modified Equation

The phenomenon observed in Sec. 4.4.1 can be explained by taking a look at the modified equation of the scheme. This equation is obtained, when the difference equation of a numerical scheme is modeled by a differential equation [33]. More specifically, the modified equation is the differential equation which is more accurately solved by the numerical scheme than the original equation (2.1). A scheme which solves Eq. (2.1) with order $p$, solves the modified equation

$$\partial_t U(x, t) + \partial_x f(U(x, t)) = D(\nu)\partial_x^{p+1} U(x, t)$$

(4.31)
Maximum value of the solution $u$ as a function of the number of time steps.

(b) Zoom of the solution for different $\omega$.

Figure 4.4 – Test case with the sign function as initial condition on $x \in [-1, 1]$ with $n = 200$ grid cells, $CFL = 0.5$, and end-time $T_{end} = 0.25$.

with order $p + 1$. Here, $D(\nu)$ is the dissipation matrix which can be obtained by computing the local truncation error of the method. Solving the linear advection equation $\partial_t U + a \partial_x U = 0$ with the Upwind Godunov and the Lax-Wendroff scheme, their modified equations read

\begin{align*}
\text{UP: } & \quad \partial_t U(x, t) + a \partial_x U(x, t) = \frac{1}{2} a \Delta x (1 - \nu) \partial_{x x} U(x, t) = D_{\text{UP}}(\nu) \partial_{x x} U(x, t), \\
\text{LW: } & \quad \partial_t U(x, t) + a \partial_x U(x, t) = \frac{1}{6} a \Delta x^2 (\nu^2 - 1) \partial_{x x x} U(x, t) = D_{\text{LW}}(\nu) \partial_{x x x} U(x, t). \tag{4.32a} \tag{4.32b}
\end{align*}

Now we define the shift $\xi := x - at$ and the shifted function $\tilde{u}(\xi, t) := u(x, t)$, as well as the Fourier transform

\begin{align*}
\hat{u}(k, t) & = \mathcal{F}[\tilde{u}(\xi, t)] := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \tilde{u}(\xi, t) \exp(i k \xi) \, d\xi, \tag{4.33a} \\
\hat{\tilde{u}}(k, t) & = \mathcal{F}^{-1}[\hat{u}(k, t)] := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{u}(k, t) \exp(-i k \xi) \, dk, \tag{4.33b}
\end{align*}

using the default definition of modern physics. Applying the Fourier transform to Eq. (4.31) leads to an ordinary differential equation which can be solved in terms of $\hat{u}$ with solution

\begin{equation}
\hat{u}(k, t) = \hat{u}_0(k) \exp(D(\nu)(-i k)^{p+1} t). \tag{4.34}
\end{equation}
With the step function as initial condition, i.e. \( \tilde{u}_0(\xi) = \text{sgn}(\xi) \), we obtain \( \tilde{u}_0(k) = \sqrt{2/\pi} \frac{i}{k} \), and thus

\[
\tilde{u}(k, t) = \sqrt{2/\pi} \frac{1}{k} i \exp(D(\nu) (-i k)^{p+1} t). \tag{4.35}
\]

The solution \( \tilde{u}(\xi, t) \) can now be obtained by inverse Fourier transform \((4.33b)\), when inserting the diffusion coefficients \( D_{\text{UP}}(\nu) \) and \( D_{\text{LW}}(\nu) \). With the values described in Fig. 4.4, Eq. (4.32) yields \( D_{\text{UP}} = 0.0025 \) and \( D_{\text{LW}} = -0.0000125 \).

In order to study the behavior of the scheme with diffusion matrix \( D_\omega \) \((4.30)\), we compute its modified equation

\[
\partial_t U(x, t) + a \partial_x U(x, t) =
\]

\[
= \frac{1}{2} a \Delta x (1 - \nu)(1 - \omega) \partial_{xx} U(x, t) + \frac{1}{6} a \Delta x^2 (\nu^2 - 1) \partial_{xxx} U(x, t) \tag{4.36}
\]

\[
= D_{\text{UP}}(\nu)(1 - \omega) \partial_{xx} U(x, t) + D_{\text{LW}}(\nu) \partial_{xxx} U(x, t).
\]

For \( \omega = 1 \), the term \( D_{\text{UP}}(\nu)(1 - \omega) \partial_{xx} U(x, t) \) disappears, leaving only the modified equation of Lax-Wendroff as in Eq. \((4.32b)\). For \( \omega = 0 \), we expect to recover the modified equation of the upwind scheme, Eq. \((4.32a)\). This is the case including the second and third terms of the Taylor expansion of the local truncation error.

Following the procedure described above, we obtain

\[
\hat{u}(k, t) = \sqrt{2/\pi} \frac{1}{k} i \exp \{ (1 - \omega) D_{\text{UP}}(-i k)^2 t + D_{\text{LW}}(-i k)^3 t \}, \quad \omega \in [0, 1]. \tag{4.37}
\]
Non-dimensionalizing Eq. (4.37) leads to a formulation with only one remaining parameter. We set $\hat{k} = k/k_0$ and $\hat{\xi} = \xi/\xi_0$, with the constants

$$k_0 = \sqrt[3]{\frac{1}{D_{LW} t}}, \quad \xi_0 = \frac{1}{k_0} = \sqrt[3]{D_{LW} t}, \quad \text{and} \quad \hat{D} = \sqrt{\frac{t}{D_{LW}^2}} D_{UP} (1 - \omega). \quad (4.38)$$

Applying the inverse Fourier transform (4.33b) yields the solution

$$\tilde{\hat{u}}(\hat{\xi}, \hat{D}) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{1}{\hat{k}} \exp(-\hat{D}\hat{k}^2) \left( \cos(\hat{k}^3\hat{\xi}) - \sin(-\hat{k}^3) \cos(\hat{k}^3) \right) d\hat{k}, \quad (4.39)$$

which only depends on the non-dimensional parameters $\hat{\xi}$ and $\hat{D}$. Plotting $\hat{\xi} - \tilde{\hat{u}}(\hat{\xi}, \hat{D})$ means that only the parameter $\hat{D}$ remains to yield different solutions, as depicted in Fig. 4.5. Increasing $\hat{D}$ can imply two things: 1) at a fixed value $\omega \in [0, 1]$, the time $t$ has been increased, corresponding to Fig. 4.4a. Or 2), at a fixed time $t$, the parameter $\omega$ has been decreased, which corresponds to Fig. 4.4b. Both scenarios result in smoother solutions and the disappearance of oscillations, as shown in Fig. 4.5

### 4.4.3 HLL$\omega$

Based on the findings of Sec. 4.4.1 and 4.4.2, we define a new Riemann solver, called HLL$\omega$. This solver is a modification of HLL with less dissipation. Instead of intersecting with the absolute value function at the points $\nu_{\min}$ and $\nu_{\max}$, HLL$\omega$ fulfills the following constraints

$$d_{\text{HLL} \omega}(\nu_{\min}) = d_{\omega}(\nu_{\min}), \quad (4.40a)$$
$$d_{\text{HLL} \omega}(\nu_{\max}) = d_{\omega}(\nu_{\max}). \quad (4.40b)$$

These conditions yield the scalar dimensionless dissipation function

$$d_{\text{HLL} \omega}(\nu) = b_0 + b_1 \nu. \quad (4.41)$$

The coefficients $b_0 = b_0(\omega)$ and $b_1 = b_1(\omega)$ are given by

$$b_0(\omega) = \frac{\nu_{\max}(\omega \nu_{\min}^2 + (1 - \omega) |\nu_{\min}|) - \nu_{\min}(\omega \nu_{\max}^2 + (1 - \omega) |\nu_{\max}|)}{\nu_{\max} - \nu_{\min}}, \quad (4.42a)$$
$$b_1(\omega) = \frac{(1 - \omega)(|\nu_{\max}| - |\nu_{\min}|) + \omega(\nu_{\max}^2 - \nu_{\min}^2)}{\nu_{\max} - \nu_{\min}}. \quad (4.42b)$$
Both coefficients are non-dimensional, which leads to the dissipation matrix $D_{\text{HLL}\omega}$ (having dimension $\Delta x \Delta t$)

$$D_{\text{HLL}\omega} = b_0(\omega) \frac{\Delta x}{\Delta t} I + b_1(\omega) \check{A}. \quad (4.43)$$

As a result, the numerical flux function for $\text{HLL}\omega$ can be written as

$$\hat{f}_{\text{HLL}\omega}(U_L, U_R) = \bar{f} - \frac{1}{2} \left( b_0(\omega) \frac{\Delta x}{\Delta t} (U_R - U_L) + b_1(\omega) (f(U_R) - f(U_L)) \right) \quad (4.44)$$

with the average flux $\bar{f} = 0.5(f(U_L) + f(U_R))$.

In the limit $\omega \to 0$ we expect $\text{HLL}\omega$ to recover HLL. This can be seen by inserting $\omega = 0$ in Eq. (4.42), yielding $b_0(0) = a_0, b_1(0) = a_1$.

The dissipation function $d_{\text{HLL}\omega}(\nu)$ is shown in Fig. 4.6 where it is well-visible, that $\text{HLL}\omega$ adds less dissipation to the solution than HLL. This figure also reveals that $\text{HLL}\omega$ is not monotone for all - however for most - wave speeds in the range $[\nu_{\text{min}}, \nu_{\text{max}}]$.

### 4.4.4 HLLX$\omega$

Now we come back to the aim of this section, the construction of a new family of approximate Riemann solvers, called HLLX$\omega$. Here, $\omega \in [0, 1]$ is a parameter, which controls the amount of dissipation of the solvers. The improvement of
HLLXω over HLLX lies in the fact that its dissipation functions are closer to the absolute value function. Therefore, the resulting solvers have less dissipation than HLLX for ω > 0.0. The dissipation functions of HLLXω are designed in a similar fashion as dHLLX, see Sec. 4.2.3. By this we mean that dHLLXω(ν) is a quadratic function, fully determined by three properties

\[
d_{\text{HLLX}}(\nu_{\min}) = d_{\omega}(\nu_{\min}), \tag{4.45a}
\]
\[
d_{\text{HLLX}}(\nu_{\max}) = d_{\omega}(\nu_{\max}), \tag{4.45b}
\]
\[
d'_{\text{HLLX}}(\hat{\nu}) = d'_{\omega}(\hat{\nu}), \quad \text{where } \hat{\nu} = \begin{cases} \nu_{\max}, & \text{if } |\nu_{\max}| \geq |\nu_{\min}| \\ \nu_{\min}, & \text{if } |\nu_{\min}| > |\nu_{\max}|. \end{cases} \tag{4.45c}
\]

These conditions yield a dissipation function which can be written as a weighted linear combination of the Lax-Friedrichs (LF), HLLω, and Lax-Wendroff (LW) dissipation functions

\[
d_{\text{HLLX}}(\nu) = \beta_0 d_{\text{LF}}(\nu) + \beta_1 d_{\text{HLL}}(\nu) + \beta_2 d_{\text{LW}}(\nu), \tag{4.46a}
\]

where

\[
\beta_0(\omega) = \beta(\omega) \frac{(1 - \omega)|\nu_{\min}|\nu_{\max}}{(1 - \omega) + \omega(|\nu_{\min}| + |\nu_{\max}|)^{-1}}, \tag{4.46b}
\]
\[
\beta_1(\omega) = 1 - \beta(\omega) \left( \frac{1 - \omega}{|\nu_{\min}| + |\nu_{\max}|} + \omega \right)^{-1}, \tag{4.46c}
\]
\[
\beta_2(\omega) = \beta(\omega), \tag{4.46d}
\]
\[
\beta(\omega) = \omega + (1 - \omega) \frac{\nu_{\max} - \nu_{\min} - |\nu_{\max}| - |\nu_{\min}|}{(\nu_{\max} - \nu_{\min})^2}. \tag{4.46e}
\]

The coefficients depend on the parameter ω, i.e. β = β(ω), βᵢ = βᵢ(ω), i = 0, 1, 2. Therefore, the whole dissipation function of HLLXω changes its behavior with ω. Depending on the choice of this parameter, the dissipation function lies more or less inside the monotone region, as shown in Fig. 4.7. Note that β(ω) = ω + (1 - ω)α, with the HLLX coefficient α, Eq. (4.23e). Thus, it is easy to verify that for ω = 0, the coefficients of HLLX (4.23e) are recovered

\[
\beta_0(\omega = 0) = \alpha_0, \\
\beta_1(\omega = 0) = \alpha_1, \\
\beta_2(\omega = 0) = \alpha_2.
\]
For $\omega = 1$, HLLX\(\omega\) recovers the $L^2$ stable but non-monotone Lax-Wendroff method, see Fig. 4.7. The choice of $\omega$ remains problem-dependent. However, $\omega \leq 0.5$ turned out to be a good choice and will be used in the following. The choice of $\omega$ will be further investigated in future work.

It can be seen in Fig. 4.6 that HLLX\(\omega\) is less dissipative than HLL, HLL\(\omega\), and HLLX. However, it does not fully lie in the monotone region, thus, one would expect oscillations near discontinuities. However, for the linear advection equation we observe that oscillations appearing close to discontinuities disappear after a certain number of time steps. For non-linear systems of equations, no oscillations are observed during the whole time of simulation. Thus, in any case, the final result obtained with HLLX\(\omega\) is non-oscillatory, see discussion in Sec. 4.4.1 and 4.4.2.

As proposed for HLLX, there is an easy way of implementing the numerical flux function of HLLX\(\omega\). By defining $\tilde{f} = 0.5(f(U_L) + f(U_R))$, we can rewrite the numerical flux function of Lax-Friedrichs, HLL\(\omega\), and Lax-Wendroff to extract their dissipation matrices. These can then be used to formulate the Jacobian-free
version of the numerical flux function

\[
\hat{f}_{\text{HLLX}}(U_L, U_R) = \hat{f} + \beta_0 (\hat{f}_{\text{LF}} - \hat{f}) + \beta_1 (\hat{f}_{\text{HLL}} - \hat{f}) + \beta_2 (\hat{f}_{\text{LW}} - \hat{f}). \tag{4.47}
\]

The dissipation function can also be formulated with only one coefficient to read

\[
d_{\text{HLLX}}(\nu) = d_{\text{HLL}}(\nu) + \beta(\omega) \cdot (\nu - \nu_{\min})(\nu - \nu_{\max}), \tag{4.48}
\]

with \(\beta(\omega)\) defined by (4.46e). Based on this formulation, it is possible to write the dissipation matrix as

\[
D_{\text{HLLX}}(U_L, U_R) = D_{\text{HLL}}(U_L, U_R) + \beta \frac{\Delta t}{\Delta x} (\tilde{A} - \lambda_{\min}I)(\tilde{A} - \lambda_{\max}I), \tag{4.49}
\]

and thus, the numerical flux function can be written as

\[
\hat{f}_{\text{HLLX}}(U_L, U_R) = \hat{f}_{\text{HLL}} - \frac{\beta}{2} \frac{\Delta t}{\Delta x} (\tilde{A} - \lambda_{\min}I)(\tilde{A} - \lambda_{\max}I)(U_R - U_L). \tag{4.50}
\]

This form might be advantageous when the flux Jacobian \(\tilde{A}\) is known.

### 4.5 Numerical Results

In this section, we provide numerical experiments in order to demonstrate the performance of the new family of Riemann solvers described in Sec. 4.4. As already stated, we are especially interested in large systems of conservation laws. Nevertheless, we start with the one-dimensional Euler equations with three emerging wave speeds. Already in this example, the difference in dissipation for slow waves (in this case the contact discontinuity) demonstrates differences in the results, as expected, see Sec. 4.5.1. Other numerical examples are the ideal magnetohydrodynamics (MHD) equations, which exhibit seven characteristic velocities, and the 13-moment equations of Grad.

Since the focus of this part of the thesis is on the solvers themselves, all tests in this section will be conducted with first order accuracy, using the explicit Euler method for time evolution. The simulation described in Sec. 113. The combination of HLLX\(\omega\) with higher order numerical schemes can be found in Chapter 6.
4.5.1 Sod’s Shock Tube Problem

We consider Sod’s problem \[53\], describing a shock tube with two different ideal gases at the left and right side of a membrane, placed at \( x = 0 \). The density, velocity, and pressure of the gases in the left and right region are given by

\[
\begin{pmatrix}
\rho_L \\
v_L \\
p_L
\end{pmatrix} = \begin{pmatrix} 1.0 \\
0.0 \\
1.0
\end{pmatrix}, \quad \begin{pmatrix}
\rho_R \\
v_R \\
p_R
\end{pmatrix} = \begin{pmatrix} 0.125 \\
0.0 \\
0.1
\end{pmatrix}
\]

(4.51)

At time \( t = 0 \), the diaphragm is removed and the gases begin to mix. The time evolution is described by the one dimensional Euler equations,

\[
U_t + f(U)_x = 0
\]

(4.52a)

with the conserved variables \( U=(\rho, \rho v, E) \), the flux function

\[
f(U) = (\rho v, \rho v^2 + p, v(E + p))^T
\]

(4.52b)

and the equation of state for ideal gases

\[
E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2,
\]

(4.52c)

where the ratio of specific heats is set to \( \gamma = 1.4 \). The computational domain is \([-2, 2]\) and the test is conducted with \( N = 20 \) and \( N = 60 \) grid cells until \( t_{end} = 0.8 \), with CFL number \( \bar{\nu} = 0.95 \), and \( \gamma = \frac{5}{3} \).
Figure 4.9 – Solution of different Riemann solvers for Sod’s shock tube problem on the domain \( x \in [-2, 2] \), with \( N = 20 \) grid cells, \( t_{\text{end}} = 0.8 \), \( \tilde{\nu} = 0.95 \), and \( \gamma = \frac{5}{3} \).

It can be seen that HLLX with \( \omega = 0.5 \) best approximates the gradients of the discontinuities since the dissipation function has very little diffusion compared to the other solvers. However, the solution shows slight overshoots in the velocity profile 4.9b. As already mentioned in Sec. 4.5.1, HLLX with \( \omega = 0.3 \) approximates the contact discontinuity almost as well as with \( \omega = 0.5 \) and does not show spurious oscillations.

We compare the new family of solvers, HLLX, for the two choices \( \omega = 0.3 \) and \( \omega = 0.5 \) with the HLL and HLLX solvers. The results for density on \( N = 60 \) grid cells are plotted in Fig. 4.8 together with the exact solution. Plots of velocity and pressure for \( N = 20 \) grid cells are shown in 4.9. Sod’s shock tube problem leads to three characteristic waves, namely a left-traveling rarefaction wave, a right-traveling contact discontinuity which is only present in the density profile and a right-traveling shock. The pressure and velocity profile, Fig. 4.9a and 4.9b, illustrate the rarefaction and the shock wave. The contact discontinuity can only be seen in the density profile, Fig. 4.8a. A zoom of the density profile around the contact discontinuity is depicted in Fig. 4.8b. This is of special interest since it highlights the different performances of the tested Riemann solvers because the contact discontinuity relates to the slow-moving wave of the system. As expected,
HLLX$\omega$ with $\omega = 0.5$ best approximates the steep gradient of the contact discontinuity since its dissipation function shows little diffusion around $\nu \approx 0$ compared to the other solvers. However, the solution shows slight overshoots in the velocity profile 4.9b. HLLX$\omega$ with $\omega = 0.3$ approximates the contact discontinuity almost as well as with $\omega = 0.5$ and does not create spurious oscillations at $t_{\text{end}} = 0.8$. HLLX and HLL are more dissipative than the new family of solvers, as expected from the discussions in Sec. 4.1.5, 4.2.3, and 4.4. Therefore, too much dissipation is added to the solution and discontinuities are smeared out.

4.5.2 Ideal Magnetohydrodynamics

Ideal magnetohydrodynamics (MHD) describes the flow of plasma, assuming infinite electrical resistivity. The equations in one-dimensional processes are given in Sec. 3.5.7 Eq. (3.60).

The seven unknown quantities of interest are: density $\rho$, normal velocity $v_x$, transverse velocity $v_t = (v_y, v_z)$, the transverse magnetic field $B_t = (B_y, B_z)$, and the energy $E$, given in terms of the pressure by Eq. (3.61). The normal component of the magnetic field $B_x$ is constant in the one-dimensional case, thus it is not considered in the vector of unknown. The adiabatic constant $\gamma$ is set to $5/3$. With seven equations for the seven unknowns, the ideal MHD equations can be considered a large system of conservation laws.

4.5.2.1 Smooth Alfvén Wave

In this simulation we want to verify and compare the order of accuracy of HLLX$\omega$ for different values of $\omega$. The smooth Alfvén wave test by Tóth [68] which is used in this section is described in detail in [14]. For the one-dimensional case, the initial conditions, describing a circularly polarized Alfvén wave, are given by

$$
\begin{pmatrix}
\rho \\
v_x \\
v_y \\
v_z \\
p \\
B_y \\
B_z
\end{pmatrix} =
\begin{pmatrix}
1.0 \\
0.0 \\
0.1 \sin(2\pi x) \\
0.1 \cos(2\pi x) \\
0.1 \\
v_y \\
v_z
\end{pmatrix}
$$

(4.53)
and the normal magnetic field $B_x \equiv 1.0$. The Alfvén wave propagates across the periodic domain $[0, 1]$, with the adiabatic index $\gamma = \frac{5}{3}$ and a CFL number of $\bar{\nu} = 0.6$. The test is run until the final time $t_{\text{end}} = 1.0$ where the solution agrees with the initial condition. This allows for the error computation given below. Tables 4.1, 4.2, and 4.2 show the $L_1$-errors of the density for different resolutions with the solvers HLL and HLLX with $\omega = 0.3$. We compute the resulting empirical order of convergence (EOC) which shows that we obtain first-order schemes in all three cases. Incorporating a high-order reconstruction in the scheme would increase the order of accuracy. This is presented in Chapter 6.
\begin{table}
\centering
\begin{tabular}{cccccc}
\hline
Grid & $\|\rho - \rho_{ex}\|_1$ & EOC & $\|\rho - \rho_{ex}\|_\infty$ & EOC & run time (sec) \\
\hline
20 & 1.851E-02 & 2.918E-02 & 0.198 \\
40 & 9.752E-03 & 0.92 & 1.537E-02 & 0.93 & 0.059 \\
80 & 5.013E-03 & 0.96 & 7.894E-03 & 0.96 & 0.156 \\
160 & 2.547E-03 & 0.98 & 4.007E-03 & 0.98 & 0.433 \\
320 & 1.283E-03 & 0.99 & 2.018E-03 & 0.99 & 1.855 \\
640 & 6.444E-04 & 0.99 & 1.013E-03 & 0.99 & 6.217 \\
1280 & 3.229E-04 & 1.00 & 5.073E-04 & 1.00 & 25.611 \\
\hline
\end{tabular}
\caption{Computed errors and empirical order of convergence (EOC) for test (4.53) with HLLX$\omega$ with $\omega = 0.5$.}
\end{table}

4.5.2.2 Magnetic Tube

Let us consider the Riemann problem given by

\begin{align*}
(\rho^L, v_2^L, v_t^L, p^L, B_t^L) &= \begin{pmatrix} 1, 0, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, 1, \begin{pmatrix} 0.5 \\ 0.6 \end{pmatrix} \end{pmatrix} \text{ if } x < 0, \\
(\rho^R, v_2^R, v_t^R, p^R, B_t^R) &= \begin{pmatrix} 1, 0, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, 1, \begin{pmatrix} 1.6 \\ 0.2 \end{pmatrix} \end{pmatrix} \text{ if } x \geq 0,
\end{align*}

and the normal magnetic field $B_x \equiv 1.5$. This problem, first introduced in [66], represents a magnetic shock tube, since density, pressure, and velocity are constant in the whole domain and all fluid movements are generated only by the difference in the magnetic field. The solution of magnetic field and velocity in $y$-direction at time $t_{end} = 1.0$ are shown in Fig. 4.10 and 4.11. The test has been computed in the domain $[-4, 4]$ with $N = 200$ grid cells and CFL number $\bar{\nu} = 0.95$.

We compare the new family of Riemann solvers with the parameter choices $\omega \in \{0.3, 0.5, 0.7\}$ to the HLL solver and the exact solution, which has been obtained by [62, 63, 64, 66]. The magnetic field, as well as the velocity in $y$-direction exhibit all waves of the system, except the contact discontinuity. The six waves are: a fast shock or rarefaction, a rotational discontinuity, and a slow shock or rarefaction, all of them to the left and to the right. The rotational discontinuity is also called Alfvén wave. The shocks and discontinuities can be seen in Fig. 4.10 and 4.11.
Figure 4.10 – Magnetic field in $y$-direction, solution of the ideal MHD equations with initial conditions (4.54) at time $t_{end} = 1.0$. The test has been computed with $x \in [-4, 4]$, $N = 200$ grid cells, and CFL number $\bar{\nu} = 0.95$. Plot markers are only depicted to differentiate between solutions, not to represent plot points.

Figure 4.11 – Velocity in $y$-direction, solution of the ideal MHD equations with initial conditions (4.54) at time $t_{end} = 1.0$. The test has been computed with $x \in [-4, 4]$, $N = 200$ grid cells, and CFL number $\bar{\nu} = 0.95$. Plot markers are only depicted to differentiate between solutions, not to represent plot points.
Here, Fig. 4.10b shows a zoom of the right-moving slow shock and the right rotational discontinuity of the $y$-component of the magnetic field. This extraction shows that the exact solution is better approximated with increasing values of $\omega$, which corresponds to decreasing dissipation. However, the solution of HLLX with $\omega = 0.7$ yields some oscillations close to steep gradients at time $t_{\text{end}} = 1.0$, so that it might preferable to use HLLX with $\omega = 0.5$. In summary, all three simulations with HLLX$\omega$ yield closer approximations of the exact solution than HLL, which in comparison introduces more diffusion.

4.5.2.3 Non-Planar Riemann Problem

We consider the Riemann problem given by [64]

$$
\begin{align*}
(\rho^L, v^L_x, v^L_t, p^L, B^L_t) &= \begin{pmatrix} 3, 0, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, 3, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} \quad &\text{if } x < 0, \\
(\rho^R, v^R_x, v^R_t, p^R, B^R_t) &= \begin{pmatrix} 1, 0, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, 1, \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix} \end{pmatrix} \quad &\text{if } x \geq 0,
\end{align*}
$$

(4.55)

with constant normal magnetic field $B_x = 1.5$. The angle $\alpha$ is the twist angle of the planes of the magnetic fields. For $\alpha = 0$, the transverse magnetic fields $B^L_t$ and $B^R_t$ have the same orientation and the Riemann problem is called planar. For $\alpha = \pi$ the left field is $B^L_t = (1, 0)^T$ and the right one $B^L_t = (0, 1)^T$. This problem is named coplanar. We consider here the twist angle $\alpha = 1.5$, which is referred to as non-planar [64].

The computational domain is $[-4, 4]$, and the solution depicted in Fig. 4.12 has been obtained with $N = 300$ grid cells, CFL condition 0.9, time step $\Delta t = 0.01$, and end time $T_{\text{end}} = 1.0$.

Fig. 4.12 shows the solution obtained with HLL, HLLX and HLLX$\omega$ with $\omega = 0.3, 0.5$. The exact solution has been obtained by [62]. Fig. 4.12a shows the density and Fig. 4.12b a zoom of the slow right moving shock at $x = 1.3$, where the resolution of all solvers can nicely be compared.

It can be stated that HLLX increases the resolution compared to HLL, causing a steeper gradient. HLLX$\omega$ further increases the steepness of the gradient for increasing $\omega$, due to decreasing dissipation. This effect is present in an even stronger form at the contact discontinuity, which corresponds to a slower wave.
Figure 4.12 – Solution of the ideal MHD equations with initial conditions (4.55) at time $T_{\text{end}} = 1.0$. The test has been computed with $x \in [-4, 4]$, $N = 300$ grid cells, and CFL condition $\overline{\nu} = 0.9$. Plot markers are only depicted to differentiate between solutions, not to represent plot points.
At the fast shock, the differences between the solutions of the four solvers are less significant, because this discontinuity relates to a larger wave speed \( \lambda \). This observation corresponds well to Fig. 4.6 where we showed that differences in dissipation functions are larger for slower waves.

4.5.2.4 Efficiency Study

Finally, we illustrate that the new methods not only yield more accurate results but are also more efficient than classical methods. In Table 4.4 we compare the performance of different schemes applied to the magnetic tube problem (4.54). For the five simulations, the time was measured until an \( L_1 \)-error of 0.005 was reached. The field “efficiency” stands for simulation time of the methods compared to the time of the HLL-simulation, \( t_{\text{HLL}} / t_{\text{HLL}} \). These results are also depicted in Fig. 4.13a which shows the mesh refinement and the corresponding error. For each method, the CPU time has been measured for the simulation corresponding to the mesh needed in order to obtain an \( L_1 \)-error of 0.005.

Fig. 4.13b shows the CPU time of different methods against the \( L_1 \)-error corresponding to meshes with \( N = 20 \cdot 2^j \), \( j = 3, \ldots, 9 \) grid points.

4.5.3 The 13-Moment Equations of Grad

In order to show the performance of the new solver-family for a larger system of conservation laws, we consider the regularized 13-moment equations (R13). This
fluid model describes rarefied fluids and micro-flows with high accuracy because
it includes effects of higher moments. The R13 equations were derived from the
Boltzmann equation by Struchtrup and Torrilhon [54] and are treated in detail
in [65]. In this work, we consider the homogeneous hyperbolic system of the R13
equations, i.e. source terms and gradient terms in the flux are neglected.
The primitive variables of interest are mass density \( \rho \), velocities \( v_i \), pressure tensor
\( p_{ij} \), and heat flux \( q_i \) with \( i, j \in \{1, 2, 3\} \). The pressure tensor is symmetric, so
that the R13 equations yield a total of 13 unknowns.
Using the Einstein notation, for the sake of simplicity, we define the total energy
\( E \) and the total energy flux \( Q_i \)
\[
E = \frac{1}{2} (\rho v_i^2 + p_{kk}) \quad , \quad s
\]
\[
Q_i = q_i + E v_i + p_{ik} v_k
\]
with the following tensor notation \( q_3 v_j = \frac{1}{2} (q_i v_j + q_j v_i) \) and \( \delta_{ij} q_k = \)
\[
\frac{1}{3} (\delta_{ij} q_k + \delta_{jk} q_i + \delta_{ki} q_j).
\]
The one-dimensional, homogeneous R13 equations are given by
\[
\partial_t \rho + \partial_x (\rho v_1) = 0,
\]
\[
\partial_t (\rho v_i) + \partial_x (\rho v_1 v_i + p_{1i}) = 0, \quad i = 1, 2, 3,
\]
\[
\partial_t (p_{ij} + \rho v_i v_j) + \partial_x \left( \rho v_1 v_i v_j + 3p_{(ij)1} + \frac{6}{5} \delta_{(ij)1} \right) = 0, \quad i, j = 1, 2, 3,
\]
\[
\partial_t Q_i + \partial_x \left( E v_i v_1 + 2v_k p_{k(ij)}v_j + \frac{2}{5} q_k v_k \delta_{i1} + \frac{14}{5} q_{(i)1} + \frac{1}{2} \left( p_{11} v_1^2 + \frac{5p^2}{\rho} \delta_{11} \right) \right) = 0.
\]

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLLX_\omega, \omega = 0.7</td>
<td>22.396</td>
</tr>
<tr>
<td>HLLX_\omega, \omega = 0.5</td>
<td>8.057</td>
</tr>
<tr>
<td>HLLX_\omega, \omega = 0.3</td>
<td>4.163</td>
</tr>
<tr>
<td>HLLX</td>
<td>2.253</td>
</tr>
<tr>
<td>HLL</td>
<td>1</td>
</tr>
<tr>
<td>LF</td>
<td>0.510</td>
</tr>
</tbody>
</table>

Table 4.4 – Performance table with reference solver HLL.
Let us consider the Riemann problem

\[
\begin{align*}
(r^L_i, v_i^L, p_{ij}^L, q_i^L)_{i,j=1,2,3} &= \begin{pmatrix} 3, 0.1 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ if } x < 0, \quad (4.58a) \\
(r^R_i, v_i^R, p_{ij}^R, q_i^R)_{i,j=1,2,3} &= \begin{pmatrix} 1, 0.1 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ if } x \geq 0. \quad (4.58b)
\end{align*}
\]

The test has been computed in the domain \([-2, 2.4]\) with \(N = 200\) grid cells and CFL condition \(\bar{\nu} = 0.8\) until time \(T_{\text{end}} = 0.9\). The reference solution has been obtained with HLL on 3000 grid cells. Fig. 4.14 shows the density field and 4.15 the velocity field in \(y\)-direction. The new family of solvers with the parameter choices \(\omega \in \{0.3, 0.5, 0.7\}\) is compared to the HLL and HLLX solvers. A reference solution has been computed on 3000 cells with the HLL flux. Again, the reference solution is better approximated with HLLX\(\omega\) than with HLL or HLLX. Also, with increasing value of \(\omega\), which corresponds to decreasing dissipation, the gradients...
Figure 4.15 – Solution of the R13 equations - velocity field in $y$-direction on the domain $[-2, 2.4]$ with $N = 200$ grid cells and CFL condition $\tilde{\nu} = 0.8$ until time $T_{\text{end}} = 0.9$. Plot markers are only depicted to differentiate between solutions, not to represent plot points.

of the numerical solutions are steeper. However, the solution of HLLX$\omega, \omega = 0.7$ yields some oscillations, e.g. in density, Fig. 4.14, around $x \approx 1.1$, so that it might be preferable to use HLLX$\omega$ with $\omega = 0.5$. 
Chapter 5

Entropy Stable Riemann Solvers

From physics we expect one unique solution for given initial data. However, not all hyperbolic equations yield unique (weak) solutions. Therefore, additional admissibility criteria need to be imposed to be able to select the physically meaningful solution \[34\]. One concept is to add artificial viscosity – which leads to a parabolic equation with a unique solution – and consider the vanishing viscosity solution. However, it is difficult to directly use this method. Thus, other concepts have been introduced to verify the physical admissibility of weak solutions of hyperbolic conservation laws. These conditions are called admissibility conditions or entropy conditions \[34\].

The objective of Chapter 5 is to prove entropy stability for the HLL dissipation operator and show that the HLL-type schemes developed in Sec. 4.4 yield entropy stable numerical flux functions. Towards this aim we first introduce the concept of entropy in Sec. 5.1 then provide a brief background on entropy stable numerical flux functions in Sec. 5.2. Further, in Sec. 5.3 we show entropy stability for the Lax-Friedrichs (LF), HLL, and Lax-Wendroff (LW) dissipation terms. Entropy stability of the family of hybrid Riemann solvers introduced in Sec. 4.4 is shown in Sec. 5.4. We demonstrate in Sec. 5.5 that the new hybrid entropy stable numerical flux reduces the overall dissipation in a standard finite volume scheme.

Parts of this chapter were published in \[44\]:

5.1 The Concept of Entropy

As detailed in previous chapters, solutions of hyperbolic conservation laws might develop discontinuities, even if the initial conditions are smooth. Therefore, solutions are sought in the weak sense, cf. Sec. 2.1. However, this is not enough since it is well-known that weak solutions are not unique. To single out the physically correct solutions, the concept of entropy is employed [34]. Consider a scalar hyperbolic conservation law of the form

\[ \frac{\partial}{\partial t} u(x,t) + \frac{\partial}{\partial x} f(u(x,t)) = 0, \quad (x,t) \in \mathbb{R} \times \mathbb{R}^+ \] (5.1)

with appropriate initial condition, where \( u \) is the conserved quantity and \( f(u) \) is a smooth flux function. Assume there exist a strongly convex mathematical entropy function \( \eta(u) \) and a corresponding entropy flux \( h(u) \) such that \( h'(u) = \eta'(u)f'(u) \) holds true [56]. Now we multiply Eq. (5.1) with the entropy variable \( v(u) = \eta'(u) \). This yields yet another conservation equation

\[ \frac{\partial}{\partial t} \eta(u) + \frac{\partial}{\partial x} h(u) = 0. \] (5.2)

This relation only holds for smooth solutions. For discontinuous solutions entropy has to be dissipated at shocks, leading to [56]

\[ \frac{\partial}{\partial t} \eta(u) + \frac{\partial}{\partial x} h(u) \leq 0 \] (5.3)

in the sense of distributions. A weak solution of Eq. (5.1) that fulfills condition (5.3) is called an entropy solution. Krushkov [31] proved uniqueness and stability of entropy solutions for scalar conservation laws.

Following the work of e.g. [56] [75], we use the concept of entropy to construct discretizations that agree with the second law of thermodynamics. This means that the numerical flux function will possess entropy stability, cf. [56] and references therein.

5.2 Entropy Stable Numerical Flux Functions

A numerical method that recovers the local changes in entropy as predicted by the continuous entropy conservation law Eq. (5.2) is said to be entropy conserva-
Entropy conservation is only valid for smooth flow configurations. For discontinuous solutions, the entropy conservation law becomes the entropy inequality. A numerical scheme is said to be **entropy stable** as long as the numerical approximation always obeys the entropy inequality Eq. (5.3). It is known that without additional dissipation, entropy conservative numerical schemes produce high-frequency oscillations near shocks, see e.g. [18, 75]. Thus, for the approximation to remain valid for general flow configurations we must add a carefully designed dissipation term to ensure that (5.3) discretely holds.

To create an entropy stable (ES) numerical approximation we start with a baseline entropy conserving (EC) numerical flux and then add a dissipation term. The resulting numerical flux at an arbitrary cell interface $i + \frac{1}{2}$ takes the form

$$f^{*,ES} = f^{*,EC} - \frac{1}{2} D [u],$$  \hspace{1cm} (5.4)$$

where $u$ is the vector of conserved variables, $D = D(u_i, u_{i+1})$ is a suitable dissipation matrix evaluated at some mean state between the two cells, and $[\cdot] = (\cdot)_{i+1} - (\cdot)_i$ is the jump between the right and left cells. For simplicity of presentation we suppress the indices on the numerical flux, the dissipation matrix, and any jump terms. For an ES scheme, the baseline central flux from a classical Riemann solver is replaced by the baseline EC flux. To guarantee entropy stability, the dissipation term in (5.4) must be carefully designed to ensure that $f^{*,ES}$ discretely satisfies the entropy inequality (5.3). To do so, we rewrite the dissipation term [40]

$$\frac{1}{2} D [u] \simeq \frac{1}{2} DH [v],$$  \hspace{1cm} (5.5)$$

with the vector of entropy variables $v = \frac{\partial n}{\partial u}$ and the entropy Jacobian $H = \frac{\partial u}{\partial v}$ which relates the variables in conserved and entropy space. Substituting (5.5) into (5.4) the entropy stable numerical flux becomes

$$f^{*,ES} = f^{*,EC} - \frac{1}{2} DH [v].$$  \hspace{1cm} (5.6)$$

The reformulation of the dissipation term, incorporating the jump in entropy variables (rather than the jump in conservative variables) makes it possible to
show entropy stability [4]. From the structure of the entropy stable flux (5.6), we find a discrete version of the entropy inequality (5.3) in cell $i$ to be

$$\frac{\partial \eta_i}{\partial t} + \left( h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}} \right) \leq -\frac{1}{2} [v]^T DH [v] \leq 0. \quad (5.7)$$

Thus, to guarantee discrete entropy stability, it is sufficient to show that $DH$ is symmetric positive definite (s.p.d).

### 5.3 Entropy Stable Classical Riemann Solvers

In this section we demonstrate entropy stability for the numerical flux of the form (5.6) for the dissipation matrix $D$ of the LF, HLL, and LW scheme. To do so, we first assume that the flux Jacobian, $A$, or a suitable Roe matrix, exists with the properties

$$A = R \Lambda R^{-1}, \quad H = (RZ) (RZ)^T, \quad (5.8)$$

where $R$ is the eigenvector matrix, $\Lambda$ the diagonal corresponding eigenvalue matrix, and $Z$ is a positive diagonal scaling matrix which creates a set of entropy scaled eigenvectors $RZ$ [4]. We see that, by construction in (5.8), the matrix $H$ is s.p.d. In the later proofs we only use the existence of the matrices $R$, $\Lambda$, and $Z$, whereas, in practice, their explicit form does not need to be known. This is advantageous, because for large systems of conservation laws, the eigendecomposition is expensive to compute or is not available.

To write the entropy stable LF scheme we substitute the dissipation matrix $D_{LF} = \frac{\Delta x}{\Delta t} I$, into the form (5.6). Now the complete dissipation term for (5.6) only depends of the known s.p.d matrix $H$, so discrete entropy stability (5.7) for $D_{LF}$ follows immediately. We note that under the same assumption the local Lax-Friedrichs (LLF) and Roe-type dissipation terms satisfy the discrete entropy stability [75].

Next, we consider the HLL flux, described in more detail in Sec. 4.1.4. The numerical flux function of ES-HLL can be written in form (5.6), with the dissipation matrix given by

$$D_{HLL} = a_0 I + a_1 A, \quad a_0 = \frac{|\lambda_L| \lambda_R - |\lambda_R| \lambda_L}{\lambda_R - \lambda_L}, \quad a_1 = \frac{|\lambda_R| - |\lambda_L|}{\lambda_R - \lambda_L}.$$
Here, $\lambda_{L,R}$ are the fastest signal velocities with $\lambda_L < \lambda_R$. From assumption (5.8) it is straightforward to show that the discrete entropy stability condition (5.7) is equivalent to showing that $a_0 + a_1 \lambda_i \geq 0$ for all $\lambda_i \in [\lambda_L, \lambda_R]$. From the form of the coefficients $a_0$ and $a_1$, keeping in mind that $\lambda_L < \lambda_R$ and $\lambda_i \in [\lambda_L, \lambda_R]$, we find

$$|\lambda_L| \lambda_R - |\lambda_R| \lambda_L + (|\lambda_R| - |\lambda_L|) \lambda_i \geq 0 \iff |\lambda_L| (\lambda_R - \lambda_i) + |\lambda_R| (\lambda_i - \lambda_L) \geq 0.$$  
(5.10)

Finally, we consider the dissipation matrix for LW

$$D_{\text{LW}} = \frac{\Delta t}{\Delta x} A^2,$$  
(5.11)

in the ES numerical flux (5.6) and find that the discrete entropy inequality (5.7) holds, i.e.,

$$\frac{\partial \eta}{\partial t} + \left( h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}} \right) \leq -\frac{1}{2} \frac{\Delta t}{\Delta x} \left( ZR^T [v] \right)^T \Lambda^2 \left( ZR^T [v] \right) \leq 0.$$  
(5.12)

## 5.4 Hybrid Entropy Stable Riemann Solvers

In this section, we consider dissipation matrices for the hybrid Riemann solvers constructed in Chapter 4. As detailed above, the hybrid solvers were constructed using weighted combinations of the dissipation matrices described in Sec. 5.3. The weighting is chosen to reduce dissipation, especially for signal velocities close to zero. As such, the hybrid terms contain a parameter $\omega \in [0, 1]$ which allows further control over the amount of dissipation added to the scheme.

### 5.4.1 HLL$\omega$

First we consider a generalization of the ES-HLL flux. The dissipation matrix of HLL$\omega$ is presented in Sec. 4.4.3 containing a parameter $\omega \in [0, 1]$.

We note that if $\omega = 0$ we recover the ES-HLL matrix (4.17). For $D_{\text{HLL}\omega}$ to fulfill the discrete entropy stability condition (5.7), it is sufficient to show that the coefficients Eq. (4.42) fulfill $b_0(\omega) + b_1(\omega) \lambda_i \geq 0$ for all $\lambda_i \in [\lambda_L, \lambda_R]$. Inserting
the definitions of $b_0(\omega)$, $b_1(\omega)$, this condition rearranges to
\begin{align*}
\omega \left[ \lambda_L^2 (\lambda_R - \lambda_i) + \lambda_R^2 (\lambda_i - \lambda_L) \right] + (1 - \omega) \left[ |\lambda_L| (\lambda_R - \lambda_i) + |\lambda_R| (\lambda_i - \lambda_L) \right] &\geq 0 \quad \forall \omega \in [0, 1] \\
\end{align*} 
which is obviously fulfilled since all terms are positive.

5.4.2 HLLX$\omega$

Next, we treat a hybrid dissipation matrix, HLLX$\omega$, that includes the parameter $\omega$ and the quadratic LW term. This term requires squaring the flux Jacobian (or applying the flux twice) but reduces the overall magnitude of the dissipation. The dissipation matrix of HLLX$\omega$ is a weighted combination of LF, HLL$\omega$, and LW, see Sec. 4.4.4 for more details and the formulation of the dissipation matrix. The resulting numerical flux is not strictly monotone. However, we can guarantee entropy stability. We have already shown discrete entropy stability for LF, HLL$\omega$, and LW. Thus, to demonstrate entropy stability for HLLX$\omega$, it suffices to show the positivity of the coefficients $\beta_i(\omega)$, $i = 0, 1, 2$, cf. Eq. (4.46). For $\omega = 1$ we obtain LW, which has already been shown to be discretely ES. Thus, let us consider $\omega \in [0, 1)$. The claim holds since
\begin{align*}
\beta_2(\omega) &= \beta(\omega) \geq 0 \\
\iff \omega + (1 - \omega)\alpha &\geq 0 \\
\iff \alpha &\geq 0 \\
\iff \lambda_R - \lambda_L - ||\lambda_R| - |\lambda_L|| &\geq 0.
\end{align*}
Hence, we directly see that $\beta_0 \geq 0$ because it is a combination of non-negative terms, see (4.46). In order to show that $\beta_1(\omega) \geq 0$ we consider the equivalent expression

$$\beta(\omega) \leq \left( \frac{1 - \omega}{|\lambda_L| + |\lambda_R|} + \omega \right)$$

$$\Leftrightarrow \omega + (1 - \omega)\alpha \leq \left( \frac{1 - \omega}{|\lambda_L| + |\lambda_R|} + \omega \right)$$

$$\Leftrightarrow \alpha(|\lambda_L| + |\lambda_R|) \leq 1.$$

We distinguish two cases. Case 1: $\lambda_L$ and $\lambda_R$ are both positive or both negative. Then $\lambda_R - \lambda_L = ||\lambda_R| - |\lambda_L||$ and therefore $\alpha = 0$. Case 2: If $\lambda_L$ and $\lambda_R$ are of opposite sign, then $\lambda_R - \lambda_L = |\lambda_R| + |\lambda_L|$, which yields

$$\alpha \left(|\lambda_R| + |\lambda_L|\right) = 1 - |\lambda_R| - |\lambda_L| \left(|\lambda_R| + |\lambda_L|\right) / (\lambda_R - \lambda_L)^2 \leq 1 \quad (5.14)$$

as desired.

### 5.5 Numerical Example - Application to Ideal MHD

We now apply the new hybrid ES-HLLX$\omega$ Riemann solver to the equations of ideal magnetohydrodynamics (MHD). The ideal MHD equations are a hyperbolic system that describes the flow of plasma assuming infinite electric resistivity, see e.g. [75]. As a proof of concept we implement the hybrid ES-HLLX$\omega$ solver into a first order finite volume framework. We use a 1D shock tube problem for the ideal MHD equations to demonstrate the reduced dissipation of the new hybrid numerical flux. We consider the magnetic shock tube of Torrilhon [66]

$$[\rho, \rho u, \rho v, p, B_x, B_y, B_z]^T = \begin{cases} [1, 0, 0, 0, 1, 1.5, 0.5, 0.6]^T, & \text{if } x \leq 0, \\ [1, 0, 0, 0, 1, 1.5, 1.6, 0.2]^T, & \text{if } x > 0, \end{cases} \quad (5.15)$$

on the domain $\Omega = [-4, 4]$ with Dirichlet boundary conditions and an adiabatic index $\gamma = \frac{5}{3}$. The baseline EC flux needed for the numerical flux ansatz (5.4) is chosen to be the entropy conserving and kinetic energy preserving flux found in
Figure 5.2 – Comparison of the computed solution of $B_y$ and $B_z$ using ES-Roe (dashed) and ES-HLLX$\omega$ (solid with knots) with $\omega = 0.25$ for the magnetic shock tube problem (5.15) at $T_{\text{end}} = 1.0$ on 300 regular grid cells.

Briefly, we note that the non-linear stability of the scheme depends on which underlying baseline EC flux is chosen to build the scheme. However, in our experience, the non-linear stability properties of a scheme created with the available EC baseline fluxes is nearly identical for low Mach number test cases (like the magnetic shock tube). We compare the ES-Roe flux of (75) and ES-HLLX$\omega$ with $\omega = 0.925$. Each dissipation term $D = D(u_i, u_{i+1})$ in the ES-HLLX$\omega$ dissipation matrix is created using a simple arithmetic mean state of the primitive variables. We note that the value of $\omega$ could be chosen adaptively using for example a pressure switch (111). Fig. 5.2 presents the computed solution of $B_y$ and $B_z$ on 300 regular grid cells against the exact solution of the Riemann problem at the final time $T_{\text{end}} = 1.0$. We see that the entropy stable hybrid numerical flux has less dissipation than the entropy stable Roe-type scheme.
Chapter 6

Numerical Results Combining Part I and Part II

In this chapter we want to combine the improved methods developed in Chapter 3 and Chapter 4. This means, we want to solve the hyperbolic conservation law $\partial_t u + \partial_x f(u) = 0$ with a numerical scheme of the form

$$\frac{d\hat{u}_i}{dt} = -\frac{1}{\Delta x} \left( \hat{f}(\hat{u}_{i-1/2}^{-}, \hat{u}_{i-1/2}^{+}) - \hat{f}(\hat{u}_{i+1/2}^{-}, \hat{u}_{i+1/2}^{+}) \right).$$

This semi-discrete form contains the three building blocks

1. high-order reconstruction,
2. Riemann solver, and
3. time discretization,

see Sec. 2.2 for definitions of these terms and more details.

6.1 Second-Order Methods with HLLXω

We verified that combining the new family of Riemann solvers with higher-order schemes yields high-order accurate results. The simulation has been performed using Heun’s method and linear, second order reconstruction. The results are computed with the ideal MHD equations and the smooth Alfvén wave test by Tóth [68], described in detail in Derigs et al. [14, p. 21] and in Sec. 4.5.2.1.
At the end time $t_{\text{end}} = 1.0$, the Alfvén wave has crossed the domain once and the initial conditions can be considered as exact solution. We compute the $L_1$-error of the $y$-component of the velocity and the resulting empirical order of convergence (EOC) which are given in Table 6.1. We observe that the expected second order is obtained for all tested choices of $\omega$. This test case validates that the novel Riemann solvers can be used with higher-order reconstructions. We do not further investigate second-order schemes, as the aim of this thesis is the development of third-order accurate methods.

Table 6.1 – $L_1$-errors of results obtained with different Riemann solvers and the full second-order reconstruction.
6.2 Third-Order Methods with HLLXω

In this section, we are combining different Riemann solvers with the higher-order reconstructions $H_3$ and $H_3^{(c)}$ developed in 3 in order to obtain third-order accurate numerical methods. The solvers we are interested in include standard choices such as local Lax-Friedrichs (LFF) and Harten-Lax-van Leer (HLL). These are compared to the newly developed solvers HLLX, and HLLXω, see Chapter 4, Sec. 4.2.3 and 4.4 for details.

All simulations have been performed using the third-order SSP Runge-Kutta method introduced by Gottlieb et al. [21]. In order to verify the performance of the new Riemann solvers, we consider the ideal MHD equations which can be considered a large system of conservation laws, cf. 4 and discussions therein.

As done in Chapter 3 where the limiter functions have been introduced, we first verify that the desired order of accuracy is obtained and compare the accuracy of different methods. Towards this end, we consider the smooth Alfvén wave test, described in detail in Sec. 4.5.2.1.

Since the superiority of limiter functions over unlimited reconstructions is only
Grid $\|v_y - v_{yex}\|_1$ EOC $\|v_y - v_{yex}\|_1$ EOC $\|v_y - v_{yex}\|_1$ EOC
25 5.457E-04 0 5.457E-04 0 5.425E-04 0
50 6.794E-05 3.01 6.794E-05 3.01 6.758E-05 3.01
100 8.484E-06 3.00 8.484E-06 3.00 8.435E-06 3.00
200 1.060E-06 3.00 1.060E-06 3.00 1.054E-06 3.00
400 1.323E-07 3.00 1.323E-07 3.00 1.316E-07 3.00
800 1.699E-08 2.96 1.699E-08 2.96 1.692E-08 2.96

<table>
<thead>
<tr>
<th>Grid</th>
<th>$|v_y - v_{yex}|_1$ EOC</th>
<th>$|v_y - v_{yex}|_1$ EOC</th>
<th>$|v_y - v_{yex}|_1$ EOC</th>
<th>$|v_y - v_{yex}|_1$ EOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>4.378E-04 0</td>
<td>3.695E-04 0</td>
<td>3017E-04 0</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>5.444E-05 3.01</td>
<td>4.578E-05 3.01</td>
<td>3.714E-05 3.02</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>6.797E-06 3.00</td>
<td>5.709E-06 3.00</td>
<td>4.620E-06 3.01</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>8.498E-07 3.00</td>
<td>7.137E-07 3.00</td>
<td>5.773E-07 3.00</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>1.063E-07 3.00</td>
<td>8.939E-08 3.00</td>
<td>7.228E-08 3.00</td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>1.333E-08 3.00</td>
<td>1.129E-08 2.98</td>
<td>9.178E-09 2.98</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2 – $L_1$-errors of results obtained with different Riemann solvers and the full third-order reconstruction $H_3$.

visible when discontinuities are present, we further treat the Magnetic shock tube problem which is detailed in Sec. 4.5.2.2.

6.2.1 Proof of Concept - $H_3$ with Different Riemann Solvers

As a first step, the full third-order reconstruction $H_3$ is tested with different Riemann solvers. We compute the solution with different Riemann solvers for different resolutions and compute the $L_1$-errors of the $y$-component of the velocity. The errors as well as the resulting EOCs are given in Table 6.2 and depicted in Fig. 6.1.

It can be seen that third-order accuracy is achieved already at very low
resolutions. Comparing the results to the second-order scheme presented in Sec. 6.1, we see e.g. that the same $L_1$-error is obtained at a resolution of $N = 50$ grid cells using the third-order scheme as opposed to $N = 160$ grid cells with the second-order scheme. The accuracy reached at $N = 800$ grid cells with $H_3$ is not even obtained at a resolution of $N = 2560$ grid cells with the lower-order scheme.

The $L_1$-errors of $v_y$ are also depicted in Fig. 6.1 which shows that we obtain indeed third-order accuracy for all resolutions and all Riemann solvers. It can be observed that the novel family of solvers has a slightly better error constant which decreases even further with increasing $\omega$.

6.2.2 $H_{3L}^{(c)}$ with Different Riemann Solvers

Now we include the limiter function $H_{3L}^{(c)}$ in the computation of the smooth Alfvén wave to validate the order of accuracy. Fig. 6.2 depicts the solution of the $y$-component of the velocity using the limiter function $H_{3L}^{(c)}$ with different Riemann solvers. The exact solution is obtained by the initial condition at the given resolution. We show the solution on such a coarse grid, since here the solutions are well-distinguishable.

The solutions for LLF and HLL are visually indistinguishable from HLLX. It can be seen how the line for HLLX lies closer to the exact solution than for HLLX.
Furthermore, with increasing values of $\omega$, the novel family of solvers HLLX$\omega$ further improves the solution. This can be explained by the decreasing amount of dissipation added to the scheme. This feature avoids the smearing out of the solution.

The $L_1$-errors of the $y$-component of the velocity as well as the resulting EOCs are given in Table 6.3 and also shown in Fig. 6.3. The desired third-order accuracy is obtained for coarse resolutions. For increasing grid cells, we loose the full third-order. In these cases, the reconstruction $H_{3L}^{(c)}$ limits too much, resulting in lower convergence orders. Nevertheless, the limiter function outperforms the unlimited reconstruction whenever discontinuities are present, as shown in Sec. 6.2.3.

### Table 6.3 – $L_1$-errors of results obtained with different Riemann solvers and the limiter function $H_{3L}^{(c)}$.

<table>
<thead>
<tr>
<th>Grid</th>
<th>$|v_y - v_{yex}|_1$</th>
<th>EOC</th>
<th>$|v_y - v_{yex}|_1$</th>
<th>EOC</th>
<th>$|v_y - v_{yex}|_1$</th>
<th>EOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>5.434E-04</td>
<td>0</td>
<td>5.434E-04</td>
<td>0</td>
<td>5.401E-04</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>6.797E-05</td>
<td>3.00</td>
<td>6.797E-05</td>
<td>3.00</td>
<td>6.758E-05</td>
<td>3.00</td>
</tr>
<tr>
<td>100</td>
<td>8.485E-06</td>
<td>3.00</td>
<td>8.485E-06</td>
<td>3.00</td>
<td>8.437E-06</td>
<td>3.00</td>
</tr>
<tr>
<td>200</td>
<td>1.131E-06</td>
<td>2.91</td>
<td>1.131E-06</td>
<td>2.91</td>
<td>1.126E-06</td>
<td>2.91</td>
</tr>
<tr>
<td>400</td>
<td>1.728E-07</td>
<td>2.71</td>
<td>1.728E-07</td>
<td>2.71</td>
<td>1.720E-07</td>
<td>2.71</td>
</tr>
<tr>
<td>800</td>
<td>2.729E-08</td>
<td>2.66</td>
<td>2.729E-08</td>
<td>2.66</td>
<td>2.716E-08</td>
<td>2.66</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grid</th>
<th>$|v_y - v_{yex}|_1$</th>
<th>EOC</th>
<th>$|v_y - v_{yex}|_1$</th>
<th>EOC</th>
<th>$|v_y - v_{yex}|_1$</th>
<th>EOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>4.359E-04</td>
<td>0</td>
<td>3666E-04</td>
<td>0</td>
<td>2987E-04</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>5.444E-05</td>
<td>3.00</td>
<td>4.577E-05</td>
<td>3.00</td>
<td>3.706E-05</td>
<td>3.00</td>
</tr>
<tr>
<td>100</td>
<td>6.800E-06</td>
<td>3.00</td>
<td>5.732E-06</td>
<td>3.00</td>
<td>4.741E-06</td>
<td>2.97</td>
</tr>
<tr>
<td>200</td>
<td>9.548E-07</td>
<td>2.83</td>
<td>8.358E-07</td>
<td>2.78</td>
<td>7.365E-07</td>
<td>2.69</td>
</tr>
<tr>
<td>400</td>
<td>1.499E-07</td>
<td>2.67</td>
<td>1.329E-07</td>
<td>2.65</td>
<td>1.188E-07</td>
<td>2.63</td>
</tr>
<tr>
<td>800</td>
<td>2.361E-08</td>
<td>2.67</td>
<td>2.151E-08</td>
<td>2.63</td>
<td>1.962E-08</td>
<td>2.60</td>
</tr>
</tbody>
</table>
6.2.3 Combined Schemes applied to the Magnetic Tube Test Case

We now consider the magnetic tube test case described in Sec. 4.5.2.2. This test case has constant initial conditions which means that $H^{(c)}_{3L}$ reduces to $H_{3L}$, see Chapter 3. The solution consists of shocks, a rotational discontinuity, and rarefaction waves, cf. Sec. 4.5.2.2. Here, the full third-order reconstruction $H_3$ leads to oscillations which is avoided by the new limiter $H^{(c)}_{3L}$.

The domain of interest is $[-4, 4]$ and we consider the coarse resolution with $N = 50$ grid cells to be able to distinguish the solutions. We test $H_3$ against $H_{3L}$, each with different Riemann solvers and CFL number $\bar{\nu} = 0.45$. The solvers we treat are LLF, HLL, HLLX and HLLX$\omega$ with $\omega = 0.3$ and $0.5$.

Fig. 6.4 shows a zoom of the solution of $\rho$ reconstructed with $H_3$ and $H_{3L}$. It is directly visible that the overshoots caused by the full third-order reconstruction are not present when the new limiter function is incorporated into the scheme. Furthermore, this figure shows that the over- and undershoots caused by the unlimited reconstruction do not disappear when refining the grid.

Figure 6.3 – $L_1$-errors measured with the smooth Alfvén wave test case using $H^{(c)}_{3L}$. The lines for the LLF and HLL schemes are visually indistinguishable from HLLX.
\begin{figure}[h]
\centering
\begin{subfigure}{0.45\textwidth}
\centering
\includegraphics[width=\textwidth]{fig6a}
\caption{$N=50$.}
\end{subfigure} \hfil
\begin{subfigure}{0.45\textwidth}
\centering
\includegraphics[width=\textwidth]{fig6b}
\caption{$N=100$.}
\end{subfigure}
\begin{subfigure}{0.45\textwidth}
\centering
\includegraphics[width=\textwidth]{fig6c}
\caption{$N=200$.}
\end{subfigure} \hfil
\begin{subfigure}{0.45\textwidth}
\centering
\includegraphics[width=\textwidth]{fig6d}
\caption{$N=400$.}
\end{subfigure}
\caption{Comparison of $H_3$ and $H_{3L}$ with different Riemann solvers. Zoom of the solution of $\rho$ obtained by the magnetic tube test case at $t_{\text{end}} = 1.0$ with $N = 50$ grid cells and CFL number 0.45 using different Riemann solvers.}
\end{figure}
Chapter 7

Conclusions

In this work we have studied finite volume methods for hyperbolic conservation laws with a main focus on Riemann solvers and high-order reconstructions with limiter functions. Hyperbolic systems of partial differential equations (PDEs) are especially challenging because shocks and discontinuities may develop out of initially smooth flows. Numerically solving such equations with high-order accuracy is difficult because undesired oscillations at discontinuities need to be avoided while high-order accuracy has to be maintained in the smooth parts.

Facing the challenge we obtained several results presented in detail in the previous chapters. In the following, we discuss some of the results and point out possible future tasks.

For reaching the aim of this thesis, the development of third-order accurate schemes, we have first studied reconstruction techniques with limiter functions. Here, we have developed a limiter function called $H_{3L}^{(c)}$ which yields third-order accurate solutions for smooth test cases. The novel limiter compares favorably with other third-order reconstructions and reaches lower error bounds. For problems containing shocks, such as Sod’s shock tube problem, the new limiter outperforms other methods, reaching given error bounds faster. This is especially due to the fact that $H_{3L}^{(c)}$ does not create over- and undershoots, as opposed to other third-order reconstructions. Furthermore, $H_{3L}^{(c)}$ also shows an excellent performance for the Shu-Osher problem, which contains oscillations that are part of the physical problem and therefore shall not be limited.
We have first developed this limiter function on one-dimensional equidistant grids and later extended it to non-equidistant grids as well as to two dimensions. Also on test cases with these meshed, the limiter shows good results compared to other third-order methods. In the future, it would be worthwhile to extend the use of $H^{(c)}_{\delta L}$ to unstructured meshes.

In the second part of this thesis we have focused on the numerical flux function of the scheme. This function is also called Riemann solver because it solves local Riemann problem at each cell boundary. The numerical flux function determines the amount of dissipation added to the scheme and therefore, how much smearing or oscillation we observe at discontinuities. The challenge here is that we want to decrease the amount of dissipation compared to classical solvers but without increasing the input information, nor the number of flux evaluations.

In this work we presented a new, hybrid family of Riemann solvers, called HLLX$\omega$, which fulfills these requirements. The family of solvers contains a parameter $\omega$ which regulates the amount of dissipation. The only input requirement is an estimate of the globally fastest wave speeds in both directions. Furthermore, only two flux evaluations are needed, which is the same as for the Lax-Wendroff scheme. These properties make the new solvers particularly efficient for large systems of conservation laws, when no explicit expression for the eigensystem of the flux Jacobian is available or it is expensive to compute.

As mentioned above, the solvers contain a parameter $\omega$ to regulate the amount of dissipation added to the scheme. So far, the choice of $\omega$ remains problem-dependent. However, $\omega \leq 0.5$ turned out to be a good choice. The aim of future work should be to find an expression for $\omega$, potentially relating the choice to the problem or describing it as a function which is able to change values during the simulation.

Testing HLLX$\omega$ against classical Riemann solvers shows the superiority of the new solvers in all numerical experiments we have conducted. The solvers cope well with large gradients and shocks, thus managing well the key issue. Additionally, our method is relatively cheap since it does not need to solve for the eigensystem of the flux Jacobian.
Altogether, the aim of this thesis – the development of third-order accurate schemes that neither smears out shocks, nor creates oscillations – has been successfully accomplished. Given the difficulty of hyperbolic partial differential equations, we have obtained very good results.
Appendix A

Time Integration

As mentioned in Chapter 2, one of the building blocks in finite volume methods is the time update. In this thesis, the evolution of the approximate cell averages $\bar{u}_i$ is considered in the so-called semi-discrete form

$$\frac{d}{dt} \bar{u}_i(t) = -\frac{1}{\Delta x} \left( \hat{f} \left( \bar{u}(t); i + \frac{1}{2} \right) - \hat{f} \left( \bar{u}(t); i - \frac{1}{2} \right) \right)$$

which in short reads

$$\frac{d}{dt} \bar{u}(t) = L(\bar{u}(t)). \quad (A.1)$$

This formulation is continuous in time and has the advantage that the discretization in space can be decoupled from the time discretization. The former has been extensively dwelt upon in Chapter 3 and 4. It remains to solve the system of ordinary differential equations (ODEs), Eq. (A.1). This can be done by applying standard methods for the solution of ODEs [33]. In this chapter, we want to present some of these methods. However, since this is not the main focus of this thesis, we only state the most important concepts and methods needed for obtaining high-order accurate solutions.

Another advantage of the semi-discrete form is the possibility to conduct stability analyses for the resulting ODEs, e.g. when applying the numerical scheme to the linear advection equation.
A.1 High-Order Time Integration Methods

The first intuition might be to discretize Eq. (A.1) in time using the explicit Euler method with a time step $\Delta t$

$$\bar{u}^{n+1} = \bar{u}^n + \Delta t \, L(\bar{u}^n)$$

$$\iff \quad \bar{u}_i^{n+1} = \bar{u}_i^n - \frac{\Delta t}{\Delta x} \left( \hat{f} \left( \bar{u}^n; i + \frac{1}{2} \right) - \hat{f} \left( \bar{u}^n; i - \frac{1}{2} \right) \right).$$  \hfill (A.2)

This form is called fully discrete and is the standard form of conservative methods [33]. However, even if the third-order reconstruction introduced in Chapter 3 is inserted into the numerical flux functions of Eq. (A.2), the resulting solution will only be first-order accurate. This can be explained by treating the error in time and space separately. The explicit Euler method is a first-order method and thus, the error due to time discretization is $O(\Delta t)$. The spatial discretization yields an error of $O(\Delta x^3)$ which leads to the global error $O(\Delta x^3 + \Delta t)$. Since the CFL condition links $\Delta t$ and $\Delta x$, the overall error is $O(\Delta x)$. This means that time-dependent problems not only need high order discretization in space but also a higher-order accurate time discretization.

A widely-used time discretization is Heun’s method, which is a two-stage Runge-Kutta method, consisting of two Euler-like time steps

$$\bar{u}^* = \bar{u}^n + \Delta t \, L(\bar{u}^n)$$

$$\bar{u}^{n+1} = \bar{u}^n + \frac{\Delta t}{2} \left( L(\bar{u}^n) + L(\bar{u}^*) \right).$$ \hfill (A.3)

This method is second-order accurate. For the desired third-order accuracy, we need yet another method. Within the framework of this thesis, we use the optimal third-order strong stability preserving (SSP) Runge-Kutta method developed by Gottlieb et al. [21]

$$\bar{u}^* = \bar{u}^n + \Delta t \, L(\bar{u}^n)$$

$$\bar{u}^{**} = \frac{3}{4} \bar{u}^n + \frac{1}{4} \bar{u}^* + \frac{1}{4} \Delta t \, L(\bar{u}^*)$$

$$\bar{u}^{n+1} = \frac{1}{3} \bar{u}^n + \frac{2}{3} \bar{u}^{**} + \frac{2}{3} \Delta t \, L(\bar{u}^{**}).$$ \hfill (A.4)
A.2 Stability Analysis

The three ODE solvers presented above, Eq. (A.2) - (A.4) can be written in the compact form

\[ \bar{u}^{n+1} = M [\Delta t L] (\bar{u}^n) \]

with the operator \( M \) applied to the time step \( \Delta t \) times the matrix \( L \), containing the (non-linear) spatial approximations, see A.1. For the investigation of linear stability of the algorithms, we assume \( L \) to be linear. For the explicit Euler scheme (A.2) the operator reads

\[ M [\Delta t L] = I + \Delta t L, \]

Heun’s method (A.3) and the third-order SSP Runge-Kutta method are given by

\[
\begin{align*}
\text{Heun:} \quad & M [\Delta t L] = I + \Delta t L + \frac{1}{2} (\Delta t L)^2, \\
\text{SSP RK31:} \quad & M [\Delta t L] = I + \Delta t L + \frac{1}{2} (\Delta t L)^2 + \frac{1}{6} (\Delta t L)^3.
\end{align*}
\]
A Fourier analysis shows that the schemes are stable for $|\mathcal{M}[z]| \leq 1$ where $z = x + iy \in \mathbb{C}$ is a complex number. The stability regions of the three methods are depicted in Fig. A.1. This figure shows that the stability region of the higher-order methods is larger, however, it does not scale the same in all directions.

A von Neumann stability analysis [19] of the finite volume schemes with the novel limiter functions lead to the stability regions depicted in Fig. A.2. The limited reconstructions have been combined with upwinding on the linear advection equation with speed $a = 1$. 

**Figure A.2** – Stability region of Runge-Kutta 3 and different reconstructions with different CFL conditions.
Bibliography


156


List of Figures

2.1 Definition of grid cells $C_i$, cell centers $x_i$, cell boundaries, $x_{i\pm 1/2}$, and grid size $\Delta x$. ........................................ 17
2.2 Comparing analytical solutions obtained by schemes with different order of accuracy. ........................................ 22
3.1 Notation of grid cells and location of interface reconstructions. ... 26
3.2 Linear and quadratic reconstruction. ................................. 28
3.3 Basic setting for reconstructions: Visualization of the undivided differences $\delta_{i\pm 1/2}$. ........................................ 30
3.4 Some classical limiter functions; the yellow shaded region is the TVD region. ........................................ 31
3.5 Regions where limiter functions $\phi(\theta)$ need to lie in, for obtaining TVD- or second-order TVD-methods, respectively. ... 32
3.6 Comparison of logarithmic limiter, $\phi_{CT}$, $\phi_{CT,TVD}$ and $\phi_3$. ................................. 34
3.7 Two situations which are reflections of each other, treated differently by $\phi_{CT}$. ................................. 35
3.8 Comparison between $\phi_{CT}$, $\phi_3$, and the new limiter $\phi_{3L}$. ................................. 36
3.9 The general limiter and its components. ................................. 38
3.10 $H_{CT,TVD}$ and $H_{CT}$ in the two-parameter framework. ................................. 41
3.11 Identical cell mean values, corresponding to a smooth solution, or to a discontinuous solution. ................................. 48
3.12 $L_1$- and $L_\infty$-errors of $\phi^{(e)}_{CT}$ with different values for $r$. ................................. 45
3.13 Limiter functions combined with the full-third-order reconstruction in a region around the origin. ................................. 49
3.14 $L_1$- and $L_\infty$-errors of $\phi^{(e)}_{CT}$ with different values for $\varepsilon$. ................................. 53
3.15 Sectional view of different limiter functions for the fixed values
\[ \delta_{i+\frac{1}{2}} \in \{2, 1, 0.5, 0.1\}. \]

3.16 Zoom of the solution of the preliminary test case, Eq. (3.51).

3.17 \( L_1 \) - and \( L_\infty \)-errors of \( \phi^{(c)}_{C_T} \) with different values for \( \varepsilon = C \Delta x^2 \).

3.18 Results of advection equation with smooth initial condition, \( \nu = 0.8, n = 170 \) grid cells and \( t_{\text{end}} = 10 \).

3.19 \( L_1 \) - and \( L_\infty \)-error for advection equation with smooth initial conditions.

3.20 Enlarged view of the discontinuities of the solution.

3.21 Results of advection equation Eq. (3.52) with discontinuous initial condition., \( \nu = 0.8 \) and \( t_{\text{end}} = 10 \).

3.22 Initial condition (3.56) containing smooth and non-smooth features.

3.23 Advection equation with smooth and non-smooth features.

3.24 \( L_1 \) - and \( L_\infty \)-error only in the smooth part of the solution.

3.25 Solution of Sod’s shock tube problem.

3.26 Solution of the Shu-Osher Problem with \( N = 640 \) grid cells, \( \nu = 0.95 \) at time \( T_{\text{end}} = 1.8 \).

3.27 Zoom at the two most challenging parts of the solution.

3.28 Solution of the ideal MHD equations using different reconstructions.

3.29 Stencil for reconstruction of interface values in the non-equidistant case.

3.30 Comparison of solutions obtained with WENO-JS and \( H^{(c)}_{3L} \).

3.31 (a) Hierarchy of grid blocks of level 2 – 5. (b) A single block consisting of \( 16 \times 16 \) grid cells, and a layer of two ghost cells.

3.32 Example of a non-uniform 2D mesh.

3.33 Computation of Eq. (3.86) on a non-uniform mesh with \( 30 \times 30 \) grid cells.

3.34 AMR computation of the Double Mach Reflection.

3.35 AMR computation of the Double Mach Reflection problem.

3.36 Results of the 2D Riemann problem at final time \( t_{\text{end}} = 0.3 \).

4.1 Comparing the scalar dissipation functions of different schemes.
4.2 Scalar dissipation functions of hybrid Riemann solvers FORCE, MUSTA1 and HLLX alias P2.

4.3 Scalar dissipation function \( d_\omega(\nu) \), Eq. (4.30) for \( \omega = 0.3 \) and \( \omega = 0.7 \).

4.4 Test case with the sign function as initial condition on \( x \in [-1, 1] \) with \( n = 200 \) grid cells, \( CFL = 0.5 \), and end-time \( T_{\text{end}} = 0.25 \).

4.5 Analytical solution \( \tilde{u}(\hat{\xi}, \hat{D}) \) (4.39), depending only on the parameter \( \hat{D} \).

4.6 The monotone Riemann solvers HLL and HLLX and their non-monotone, however, less dissipative versions HLL\( \omega \) and HLLX\( \omega \).

4.7 HLLX\( \omega \) dissipation function for different values of \( \omega \).

4.8 Solution of different Riemann solvers for Sod’s shock tube problem.

4.9 Solution of different Riemann solvers for Sod’s shock tube problem.

4.10 \( B_y \)-solution of the ideal MHD equations with different Riemann solvers.

4.11 \( v_y \)-solution of the ideal MHD equations with different Riemann solvers.

4.12 Solution of the ideal MHD equations with different Riemann solvers.

4.13 Efficiency study of classical Riemann solvers and HLLX\( \omega \).

4.14 Solution of the R13 equations - density field.

4.15 Solution of the R13 equations - \( v_y \)-field.

5.1 Visualization of the dissipation matrix \( D \) as a function of the eigenvalue \( \lambda \). For the hybrid dissipation matrices we select \( \omega = 0.4 \).

5.2 Comparison of the computed solution of \( B_y \) and \( B_z \).

6.1 \( L_1 \)-errors measured with the smooth Alfvén wave test case using \( H_3 \).

6.2 Solution of the smooth Alfvén wave test obtained by \( H_{3L}^{(c)} \) at \( t_{\text{end}} = 1.0 \) with \( N = 8 \) cells on the domain \([0, 1]\).

6.3 \( L_1 \)-errors measured with the smooth Alfvén wave test case using \( H_{3L}^{(c)} \).

6.4 Comparison of \( H_3 \) and \( H_{3L} \) with different Riemann solvers.

A.1 Stability regions of the explicit Euler method, Heun’s method and Runge-Kutta SSP3.
A.2 Stability region of Runge-Kutta 3 and different reconstructions with different CFL conditions.
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Summary of situations, evaluated with ( \phi_{CT,TVD}, \phi_{CT}, ) and ( \phi_{3L} ); ( \delta_1, \delta_2 &gt; ) and ( \delta_3 &lt; 0 ) are constants s.t. the described situations occur.</td>
</tr>
<tr>
<td>3.2</td>
<td>Errors of ( H_{3L}^{(c)} ) in ( L_1 )- and ( L_\infty )-norm and corresponding empirical order of convergence (EOC).</td>
</tr>
<tr>
<td>3.3</td>
<td>Results for the Vortex evolution problem an a uniform grid.</td>
</tr>
<tr>
<td>3.4</td>
<td>( L_1 )-errors of ( H_{3L}^{(c)} ) and corresponding empirical order of convergence (EOC).</td>
</tr>
<tr>
<td>4.1</td>
<td>Computed errors and empirical order of convergence (EOC) for test (4.53) with HLL.</td>
</tr>
<tr>
<td>4.2</td>
<td>Computed errors and empirical order of convergence (EOC) or test (4.53) with HLLX( \omega ) with ( \omega = 0.3 ).</td>
</tr>
<tr>
<td>4.3</td>
<td>Computed errors and empirical order of convergence (EOC) for test (4.53) with HLLX( \omega ) with ( \omega = 0.5 ).</td>
</tr>
<tr>
<td>4.4</td>
<td>Performance table with reference solver HLL.</td>
</tr>
<tr>
<td>6.1</td>
<td>( L_1 )-errors of results obtained with different Riemann solvers and the full second-order reconstruction.</td>
</tr>
<tr>
<td>6.2</td>
<td>( L_1 )-errors of results obtained with different Riemann solvers and the full third-order reconstruction ( H_3 ).</td>
</tr>
<tr>
<td>6.3</td>
<td>( L_1 )-errors of results obtained with different Riemann solvers and the limiter function ( H_{3L}^{(c)} ).</td>
</tr>
</tbody>
</table>