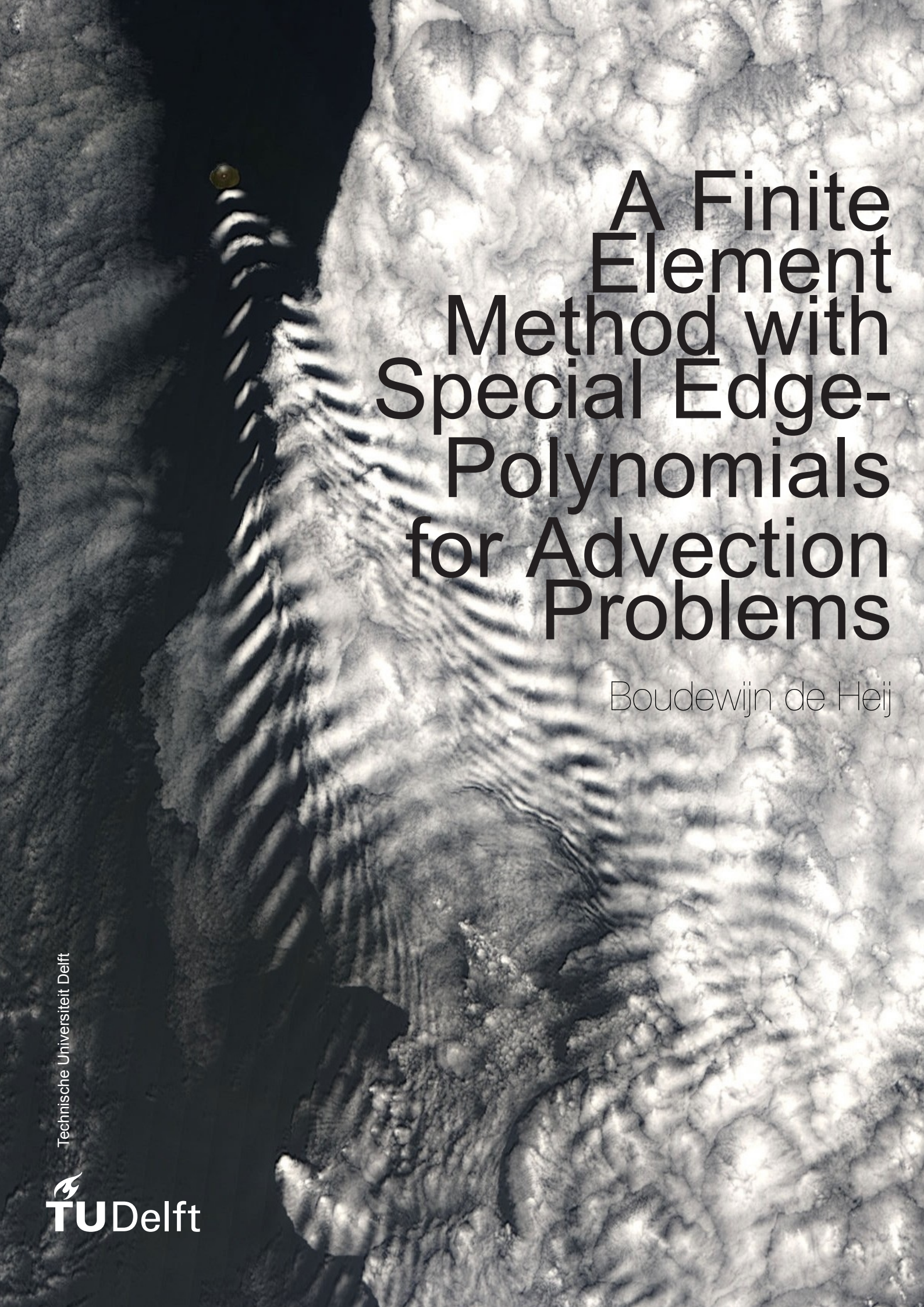


A Finite Element Method with Special Edge-Polynomials for Advection Problems

Boudewijn de Heij

Technische Universiteit Delft

TU Delft

A Finite Element Method with Special Edge-Polynomials for Advection Problems

by

Boudewijn de Heij

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday December 16, 2016 at 14:00 AM.

Student number:	1507095	
Project duration:	October, 2015 – December 2016	
Thesis committee:	Dr. ir. M.I. Gerritsma,	TU Delft, aerodynamics, supervisor
	Prof. Dr. S. Hickel,	TU Delft, aerodynamics
	Dr. ir. A. Palha da Silva Clérigo	TU Eindhoven
	Dr. M. Möller	TU Delft, Applied Mathematics

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Cover image by NASA [35]

TU Delft

Executive Summary

The field of Computational Fluid Dynamics (CFD) is constantly finding new ways to improve simulation results. One of the large challenges in CFD, is advection. Advection dominated flows are governed by hyperbolic equations, which pose severe stability issues. Stabilized methods usually add significant numerical diffusion, which alters the solution. Another issue are discontinuities in solutions, such as found in shock waves and acoustics. Besides solving these issues, conservative properties have to be satisfied at all costs. In order to further improve results, new methods are still developed today.
The mathematical tools to do such developments are also further expanded. For instance, in the last decades the branch of differential geometry has steadily grown. This branch of mathematics can be extremely useful, since a clear distinction is made between geometric depend and geometric independent operators. These operators become more in use for numerical methods. Of all the available operators, the Lie derivative is important for advection. In junction with newly developed edge polynomials, a scheme for advection is created. The new edge polynomials are special, as these partly scale with a variable integrated over the cell and partly scale with the values on the cell boundaries. In order to do this, one polynomials integrates to 1 over the cell, while the other polynomials do not contribute, hence integrate to 0 over a cell. This gives several benefits. Conservation is clearly defined in the cells and not in points, hence conservation is locally defined. Also an extinction is made between conservative values and fluxes. One can also use the property of not contributing to conservation by correcting point values for stabilization or improvement of the solution. These polynomials are used within a Galerkin Finite Elements (FE) framework. Besides correction methods, upwind methods can be used as well.
The approach is tested on several equation sets. On a simple linear advection problem, excellent results are found. Four test cases are tried, from a sine wave, two discontinuous functions and a hat function. With the proposed polynomials, the shapes are conserved extremely well, also in comparison to widely used schemes.
On the inviscid Burgers' equations, the stabilization methods become of importance. In the inviscid Burgers' equation, initially smooth problems can become discontinuous. When a discontinuity forms, several issues can occur. For one, spurious oscillations can become apparent, or the velocity of the discontinuity can have errors. With several approaches, competitive results can be obtained. The discontinuity velocity is computed well, but spurious oscillation usually do occur. With several methods, the oscillations are suppressed. Not all approaches succeed in doing so.
The third and final equation set to be tested, are the Euler equations. Although solutions can be found, all solutions contain errors. Several error sources are indicated, which should be addressed in further research. In comparison to a Monotonic Upstream-Centered Scheme for Conservation Laws (MUSCL) scheme, results need to be further improved to be on par.
In conclusion, the suggested polynomials show potential. Several methods to improve solutions showed their usefulness. In linear advection, the new polynomials performed better compared to existing widely used methods. In non-linear advection, the results are on par with reference methods. For the Euler equations, further improvements are necessary. With further research, the benefits of such polynomials for numerical methods would become even more clear.

Preface

This thesis project started out with a simple idea: Let the values on cell faces in a Finite Volume method evolve in time by them self. With carefully derived polynomials this should be possible. But just like many ideas, the execution is the hard part. And so my research started and for several months many possible solutions for this idea were tested, refined, tested again. It took a while to find a good solution, but in the end several results were found which made it possible to get to this fantastic moment

Boudewijn de Heij
Delft, December 9, 2016

Acknowledgements

First of all I would like to thanks Marc Gerritsma. Under his supervision this thesis has formed to what is presented here. Without the brainstorm sessions, the feedback and after meeting small talk the thesis period would be not as great as it was. Furthermore many thanks to Dr. Ir. A. Palha da Silva Clérgi and Prof. Dr. S. Hickel for the help and feedback on other questions regarding numerical methods for solving the compressible Euler equations.
I would also like to thanks my family and girlfriend for all encouragements and support the past year. Finally I would like to thank all (study-)friends for the fantastic years I had in Delft.

Contents

List of Figures

List of Tables

List of Acronyms and Symbols

Acronyms

AF	Active Flux
AUSM	Advection Upstream Splitting Method
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy number
CG	Continuous Galerkin
DG	Discontinuous Galerkin
ENO	Essentially Non-Oscillatory
FCT	Flux-Corrected Transport
FD	Finite Differences
FE	Finite Elements
FV	Finite Volumes
HLL	Harten Lax van Leer
LSFEM	Least-Squares Finite Element Method
MUSCL	Monotonic Upstream-Centered Scheme for Conservation Laws
PDE	Partial Differential Equation
RK	Runge-Kutta
SEM	Spectral Element Method
SUPG	Stream-Line Upwind Galerkin
TVD	Total-Variation-Diminishing
WENO	Weighted Essentially Non-Oscillatory

Symbols

B	Bubble function perturbation
L_1	L_1 error norm
L_2	L_2 error norm
L	Domain length
R	Specific gas constant
T	Temperature
T	SUPG perturbation
Λ	space
Ω	Domain
β	Upwind parameter
γ	Specific heat ratio
$\mathbb{F}$	Convective flux matrix
$\mathbb{G}$	Rest terms matrix
$\mathbb{M}$	Mass matrix
$\mathbb{P}$	Pressure matrix
$\mathcal{L}$	Lie derivative
$\mathcal{O}$	Order
ρE	Energy density
ρ	Density
$\star$	Hodge
θ	Ratio between two timesteps in θ scheme
$\wedge$	Wedge
ξ	Scaled x-location
a	speed of sound

a	Advection velocity
c_s	shock velocity
d	Exterior derivative
i	Interior product
m^x	Momentum in x direction
p	Pressure
r	Ratio of slopes
t	time
u	Solution space
u	velocity u
v	Test space
N	Number of cells

1 Introduction

In the last decades, the engineering uses of computers have increased dramatically. Today, in every step of the design, the computer is involved. Partly, this is due to the computational power increases every year. Another important reason, is that software and numerical methods are further developed. These developments allow to accurately simulate and answer engineering question. In fluid dynamics, this allows to simulate a wide ranges of flows, such as pipeflow, the flow field around aircraft or other vehicles, but also interaction of fluid and other fluids or structures can be simulated. However, the results of such simulations fully rely on the performance of the underlying numerical method.
These numerical methods have advanced significantly over the years. In time, schemes have been developed with increased order of accuracy, schemes which are non-oscillatory and/or more stable. However, none of the methods are perfect. Roughly speaking, numerical methods can be divided in three branches: Finite Differences (FD), Finite Volumes (FV) and Finite Elements (FE). Each of these branches have their own advantages and disadvantages, however non has been able to fully replace real world testing [28]. Over time, the disadvantages have been covered up as much as possible, with all kind of tricks. However, these tricks affect the solution one-way or another. For example, methods to improve the solution in advection dominated flows, tend to add extra diffusion [43]. Other ways researches have tried to improve methods, is to combine ideas from multiple branches. For instance, this led to the development of Discontinuous Galerkin (DG) methods [11, 38]. However, these are much more complicated and have the advantages and disadvantages of both FV and FE.
Advection, is one of the big challenges for numerical methods in fluid dynamics. Another big issue can be found in hypersonic fluids and acoustics. In these problems, shocks occur. For many methods, discontinuities mean troubles. Often oscillations occur, regularly not damped out and resulting in a solution that diverges from the real solution. In order to damp such spurious oscillations, additional damping is often required, either found in limiters, upwind or other methods. These in turn add artificial diffusion to the governing equations.
In short, Even though numerical methods have improved immensely, the schemes still are far from perfect. In this thesis, a new advection operator is developed. This operator will be implemented into a FE framework. In order to do so, mathematical tools from differential geometry will be used. This is a branch of mathematics for which many of the operators have a discrete counterpart. This discrete counterpart is called algebraic topology. Within these frameworks, an operator for advection exist, namely the Lie derivative.
Further more, a special set of edge functions are derived from Hermite polynomials. With these polynomials and the Lie derivate, As a results, polynomials are found which partly scale with integrated values, like FV, and partly with point on the cell faces, which is usually done in FE methods. Another reason such polynomials can be a good idea can be found in the governing equations. For fluids, these equations are derived as a conservative quantity in a volume, which change in time due to flows through surfaces and forces acting upon the volume. With the proposed polynomials, a clear distinction can be made between different forms, being either volumes, surfaces, lines or points.

1.1. Thesis Goals

This leads us to the goal of this thesis. The main goal of this research can be stated as: *Construct a numerical scheme for advection problems, using polynomials which scale with both integrated values and point values by using the Lie derivative*.
This scheme is to be benchmarked, which is done through a series of linear, non-linear and multi-equation one-dimensional problems. A comparison is made with mainstream methods. These methods are widely used FV methods with limiters and FE methods with piecewise polynomials.

1.2. Thesis Outline

In order to reach the goals for this thesis, first an overview of the current state-of-art methods is given in Chapter 2. Some of the methods presented in this chapter are used as benchmark methods. Also treated in this chapter are the mathematical tools used to develop the scheme. In this thesis, the method proposes will be tested on the linear advection equation, the inviscid Burgers' equation and the one-dimensional Euler equations. These equations and how these are solved are explained in Chapter 3. In Chapter 4, the method is proposed. In order to do so, first a special set of edge polynomials is derived. With these polynomials, the FE scheme is made, followed by an example of the linear advection. The properties of the polynomials are further used to propose several stabilization and correction options. For this, the properties of the equations are used.
The following chapters are used to present the results. In Chapter 5, extensive results on linear advection are shown. This is a simple problem, but with several test cases both the strong and weak points of a scheme can be found. In Chapter 6, inviscid, non-linear advection is investigated. Being a non-linear problem, challenges increase for schemes. An initial smooth problem forms at a certain point a shock. In Chapter 7, the scheme is further expanded for the one-dimensional Euler equations. Several shock-tube problems are introduces and tested. One of the test cases in this chapter is Sod's problem [41], a well known benchmark problem.
The conclusions are drawn in Chapter 8. This is followed by the final chapter. Chapter 9 gives recommendations for further research, as well as changes which should have done differently.

2 Literature

Over the years many different types of numerical methods have been developed, both for time integration and spatial discretization. In the most crude classification, methods can be Finite Differences (FD), Finite Volumes (FV), or of the Finite Elements (FE) type. In chapter the methods used for advection dominated flows, in particular for the Euler equations, are elaborated upon. In advection dominated flows, the most widely used methods are of the FV type, so these methods will be discussed first. After that, the FE methods will be explored. FD will not be discussed in detail, as these methods became out of fashion after the advantages of FE and FV became clear in the late 80s and 90s.

2.1. Finite Volume Methods

FV methods are very well suited for advection flows. In general these methods are easy to program and conservation is assured. The general approach for FV methods is to integrate conservative properties over each element, resulting in integrated values in volumes, hence the name. The change in time of these integrated values, is defined by the in- and outflow fluxes. Also, the forces of shear and presure are defined on the boundaries, working on the volume. For simple advection problems, depending on the discretization, a central or upwind scheme can be constructed. Since many fluid problems, such as those governed by the Euler equations, are non-linear hyperbolic conservations laws, other techniques are necessary. In this section a part of the history and important methods regarding FV are discussed. Extensive overviews of can be found in the review paper by Van Leer [49] and the book by Toro [46].
FV methods for fluid problems started when Godunov developed the Godunov scheme in 1959 [20, 46]. This FV type scheme calculates fluxes by solving a local Riemann problem on the boundaries of the cells. The local Riemann problems are solved exactly. This yields accurate results, but at the cost of being computationally expensive and directional. It took a while before changes to Godunov's method were proposed. Russian research only slowly dripped to the rest of the world in that time. In the 80s several approximate Riemann solvers were published. One of the best known approximate solvers is the one from Roe [39], which locally linearises the flux equations. In the same year Osher [15] proposed his method and slightly later the HLL solver of Harten, Lax and Van Leer [22] was presented.
Another way to solve the fluxes is by flux vector splitting. Flux splitting schemes use the direction of information to create a local upwind. Known flux splitting schemes are the Beam-Warming scheme [4] and Advection Upstream Splitting Method (AUSM), developed by Liou [31]. On Riemann solvers many comparison research has been performed [27, 34].
Godunov is also known for his theorem, stating:

> if an advection scheme preserves the monotonicity of the solution it is at most first-order accurate [20, 49].

In short, this means that higher order accurate schemes will show oscillatory behaviour near strong gradients. These oscillations can be very detrimental for the solution. Luckily, in several ways Godunov's theorem can be circumvented. This could be done by using reduced order approximations near shocks, while smooth regions are solved with higher order schemes. Such schemes are usually

called high-resolution schemes. A very popular way to achieve this, is by means of limiters. Nowadays many different limiters are available, such as minmod, Koren, Sweby and superbee. Another method developed early to reduce spurious oscillations are the Flux-Corrected Transport (FCT) and Monotonic Upstream-Centered Scheme for Conservation Laws (MUSCL). Both these methods are predictor-corrector schemes. Comparison between MUSCL and FCT can be found in [50]. Later, after it was shown that the total variation of the solution could not increase, de concept of Total-Variation-Diminishing (TVD) schemes was proposed by Harten [23]. Later, Harten was also part of ground breaking research that lead to Essentially Non-Oscillatory (ENO) interpolation [23]. This interpolation type automatically selects the smoothest stencil to interpolate the solution. Weighted Essentially Non-Oscillatory (WENO), was a further improved version of ENO and is widely used. A huge benefits of the ENO-type interpolation methods, is that it can be truly multidimensional while being non-oscillatory. A completer review of the WENO was written by Shu [40].
Only recently, a new type of scheme for advection problems was proposed by Eymann and Roe, the Active Flux (AF) scheme [16]. Based on a scheme developed during the 1970's by Van Leer [48], the team of Roe and Eyeman [16, 17] developed a true multidimensional scheme. In this scheme, the vertex values evolve in time independently from the conserved value in the cells. The original Van Leer scheme V was deemed to be computationally too complex for the time. In [16], the idea of an AF scheme was brought back to life and the newly developed AF scheme was proposed. Comparison of the Van Leer scheme V [48], the new developed AF scheme and existing methods showed promising results, hence further research was conducted. In [17], the group achieved a third order, two-dimensional scheme for linear advection, acoustics and linearized Euler equations. For both the linear advection and acoustic wave equations, the results were excellent and demonstrated the potential of the method. The acoustic pressure waves showed truly multidimensional behaviour. The schemes are currently being extended to advection-diffusion and to deal with source terms [37]. Again results are good with a relative small error margin. As the reader will notice later on, the proposed method in this thesis shares several ideas with the AF scheme. In the proposed method, similar basis functions will be used, which result in independent updates of fluxes and integrated properties.
The recent development of the AF method shows that new ideas within FV methods still arise. FV methods are proven to be successful for fluid problems and in general are easy to implement. Conservation is implied and a wide range of methods, such as limiters and WENO constructions, are available for non-oscillatory behaviour. However, to go to higher order than second order, the computational costs significantly increases. Due to the volume form used, FV methods can be easily used on unstructured grids. As a result, second order FV methods are widely used in the industry at this moment.

2.2. Finite Element Methods

For a long time Finite Elements were not the common choice within fluid dynamics. Due to convection, the governing equations become hyperbolic, with the result that FD and FE become highly unstable. FD however is computationally less expensive and by using upwind differences the schemes can be made stable fairly simply. FV methods are conservative by nature and less expensive as well. Due these facts, research for Computational Fluid Dynamics (CFD) methods focussed mostly on FD and FV schemes. FE was at that time a method for structural engineers. However, FE has some very big benefits. For one, it can handle different grids and boundary conditions with relative ease. FE can be made conservative, while FD is not. However, to make FE locally conservative requires significantly more effort compared to FV. Also the mathematical foundations FE is based on are extremely solid and can be used to derive FD and FV schemes as well.
The fundamentals for Finite Elements were developed by Ritz as early as 1909. Finite Element Methods are residual minimizing methods, the residual of the weak formulation that is. The weak formulation is found by multiplying the governing equations and boundary conditions with test function. When this is integrated over the domain, a system of equations is found which can be solved [7]. The most well known FE approach in fluid dynamics, is the Galerkin method. In the Continuous Galerkin (CG) method, the test space functions are the same as those selected to represent the solution and at least C^0 continuous. Other basic ideas are the Discontinuous Galerkin (DG), which represent the solution with discontinuities, so the solution space and the test spaces are not smooth. For DG schemes, additional boundary conditions and variables are required. DG methods have shown great results, which made the method popular. The Least-Squares Finite Element Method (LSFEM) uses continuous elements,

just like continuous Galerkin, but the Partial Differential Equation (PDE) is imposed on the test space as well [26]. As a result a positive definite system of equations can always be found. On the other hand this requires a lot of effort and leads to non-sparse systems. Also popular in research, are Spectral Element Method (SEM). Spectral Elements are very similar to normal FE, but try to achieve accuracy by resorting to high degree piecewise polynomials.
Although many different FE approaches exist, it took a while before the methods became widely used in fluid dynamics. The advection terms in the PDE's resulted in unstable schemes. In the 80's, this problem was tackled, when the idea of upwinding could add stability. Upwind can be done by modifying the functions in the testspace. By modifying the testspace only in direction of the flow, the Stream-Line Upwind Galerkin (SUPG) method was born [24]. The first work of Hughes on SUPG on convection dominated flow [44] and compressible flow [8], was a breakthrough. Later it was shown that SUPG worked on entropy variables as well [25], which later allowed the development of shock capturing terms. Where SUPG adds a small artificial viscosity in direction of the flow for stabilization purposes, shock capturing terms further increase viscosity near large gradients.
Although already shortly treated, DG deserves additional attention since interest in FE for fluid dynamics increased due to it. Originally introduced by Reed and Hill [38], it can be seen as a combination of FV and Galerkin FE. The Galerkin method, the 'standard FE', weights the governing equations with test functions which are the same function used to represent the solution between nodes. DG does this as well, but the functions used in DG methods allow discontinuities between cells. As a result, between each cell boundary conditions determined, which can be a finite volume style flux. Because of this, some nodes have multiple values. The result of that is an increase in storage requirements. After the original paper of Reed and Hill, a set of papers by Cockburn and Shu [10–13] brought the development of this branch of FE in an acceleration. Nowadays DG is a widely used method and many different papers on the subject can be found.
Since the DG method has some sort of flux, limiters can be used just as for Godunov and other FV schemes. Just as with FV methods, a wide variety of papers can be found only on the design of limiters. Although limiters usually reduce a method to first order locally, limiters exist which achieve higher order approximation near shocks [29]. Limiters designed to work with higher order time integrators such as Runge-Kutta (RK) methods exist as well [9]. Even combinations of oscillation oppressing methods exist, which lead to limiters in combination with WENO reconstruction [51]. However, DG methods are complicated, difficult to program and require more memory. In that respect, DG methods are both the best and the worst of both FE and FV.
Nowadays, FE slowly start to take their place within CFD. The discussed methods allow Finite Elements to be used in strong advective flows, without becoming unstable. Also, when continuous elements are used, global conservation is implied, but local conservation is not, which results in difficulties when shocks occur. The mathematical basis is, however, very solid. With the stabilization techniques of today, FE becomes more popular.

2.3. Differential Geometry, Algebraic Topology and Mimetic Methods

A relatively new branch of mathematics emerged in the last couple of decades, although several ideas are already from the 1900-1930's. In this branch, continuous mathematical structures have a discrete counterpart with exactly the same properties. The continuous mathematical structure is called differential geometry, the discrete counterpart is algebraic topology and methods developed with these ideas are sometimes called mimetic methods. The schemes developed in this research rely heavily on these theories. In order to understand the concepts of mimetic schemes and differential geometry and its discrete counterpart, first a wider introduction of mimetic schemes is given. After that the important concepts of differential geometry are discussed and it will become clear why these concepts are useful for the development of mimetic schemes.
Although mimetic methods can use differential geometry explicitly, the definition of mimetic methods is a bit wider. In general, mimetic methods try to mimic the real physics in a numerical scheme. In other words: the underlying physics are mimicked in the discrete domain [5]. Hence, the notion of a mimetic method is very broad. However, it is very important to fully understand the underlying physics of the problem at hand.
According to Lipnikov [5], several phases in the research of mimetic (Finite Difference) methods can

be distinguished. The first phase started mid-fifties and focussed on operators with preserving capabilities, orthogonal meshes, and convergence and stable behaviour. In the second phase, starting in the seventies, methods evolved further, as the need for non-orthogonal meshes arose. In this period, discrete operators based on variational principles, and discrete integral identities were developed. Also, the component vector components are used as extra degrees of freedom for vector fields, and practise of conservative staggered discretization became common. Mid-nineties the third period started. The name of a 'mimetic scheme' was first used and the first researchers were consciously trying to mimic important aspects of a physical problem. In this period the mathematical foundation was laid. Mimetic discretization and discrete vector and tensor calculus had a strong development. Methods also focussed on a wider spectrum of grids, such as polygonal and polyhedral grids. In the fourth and current period, starting in 2004 according to Lipnikov, much of the research is aimed at stability and convergence properties. Also arbitrary-order discretization becomes of importance.
As a mimetic method tries to mimic underlying physics, it is now time to discuss differential geometry, which due to its discrete counterpart can mimic continuous physics with discrete points, regardless of spatial orientation. Differential geometry is quite similar to vector calculus, but is much more powerful, as many operators have no metrics involved. The metrics are contained in specific operators, such as the Hodge. Several sources are available in understanding the basics of differential geometry. For a proper introduction the works of Boothby [6] and Tu [47]. Frankel [18] also gives a clear understanding of the mathematics, but puts more effort in the connections of the math and the physics. For an even more practical approach the work of Stone [42] is very useful. Finally the paper of Mansour [32] gives an overview of the tools relevant to fluid dynamics. The important theoretical content of these works is discussed in the remainder of this section.
In order to understand the differential geometry theory, the first thing one should understand are manifolds and k-forms. A manifold is the n-dimensional space where differentiable functions live on. The notion of forms, called k-forms, is important. The definition of a k-form can be formulated as

> a k-form can be described as an entity ready (or designed, if you prefer) to be integrated on a kD (sub)region [14].

In general this means that a 0-form is associated with points, 1-forms with lines, 2-forms with area's and 3-forms with volumes. Another way to put it is that a k-form can be a vector, tensor or a scalar in the domain $\mathbb{R}^n$. When one takes a close look at a physical set of equations, one can notice that differentiation and integration is a change from one form to another. For instance, a line integral associates values in points to lines. Similar, a differentiation of a surface is done in direction of a line. Several operators defined within the differential geometry framework are available for integration and differentiation. Important operators for this are the wedge product, $\wedge$, the exterior derivative and the interior product, d and i_v.
The wedge, $\wedge$, is a skew-symmetric operator and often called the exterior product. The result of a wedge between an k-form and l-form is a $(k+l)$-form. The wedge is used when in integrating function. When integrating over a volume for instance, the wedge should be used: $dx \wedge dy \wedge dz$. Often the $\wedge$-sign is omitted due to the similarity with integral calculus. The wedge however, has orientation build in. This is due to the anti-symmetry property. Other properties of the wedge are that it is associative and multi-linear. These properties are respectively:

$$a^{(k)} \wedge b^{(l)} = (-1)^{kl} b^{(l)} \wedge a^{(k)}, \tag{2.1a}$$

$$(a \wedge b) \wedge c = a \wedge (b \wedge c), \tag{2.1b}$$

$$\alpha a \wedge (\beta b + \gamma c) = \alpha\beta (a \wedge b) + \alpha\gamma (a \wedge c). \tag{2.1c}$$

The exterior derivative is, d, is the differential of a function and maps a k-form onto a $(k+1)$-form. The general form of the d of function f is:

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy + \frac{\partial f}{\partial z} dz. \tag{2.2}$$

The d has several properties, it is additive, it follows the Leibniz rule and the d of the d is zero. These properties are respectively:

$$d\left(a^{(k)}+b^{(k)}\right)=da^{(k)}+db^{(k)}, \tag{2.3a}$$

$$d\left(a^{(k)}\wedge b^{(m)}\right)=da^{(k)}\wedge b^{(m)}+(-1)^{k}a^{(k)}\wedge db^{(m)}, \tag{2.3b}$$

$$d^{2}a^{(k)}=d\left(da^{(k)}\right)=0. \tag{2.3c}$$

The d is similar to the gradient, curl and divergence in vector calculus, depending on which form it is applied. In fact, this translation can be written as a chain:

$$\mathbb{R}\longrightarrow\Lambda^{(0)}(\Omega)\xrightarrow[\nabla]{d}\Lambda^{(1)}(\Omega)\xrightarrow[\nabla\times]{d}\Lambda^{(2)}(\Omega)\xrightarrow[\nabla\cdot]{d}\Lambda^{(3)}(\Omega)\xrightarrow{d}0\ . \tag{2.4}$$

This chain also shows that the d takes a k-form from the k-space Λ^{k}, to a $k+1$-space Λ^{k+1}. Using the generalized Stokes theorem, a connection between the exterior derivative, d, and the discrete equivalent, the boundary operator ∂ can be found. Written as the duality paring this is:

$$\left\langle da^{(k-1)},\mathbb{M}^{k}\right\rangle=\left\langle a^{(k-1)},\partial\mathbb{M}^{k}\right\rangle, \tag{2.5}$$

where $\mathbb{M}^{k}$ is a k-dimensional subdomain. This duality pairing is not depending on space or dimension, hence it becomes also clear why metrics should not play a role of importance. Line, surface and volume integrals are independent of metrics in space [18].
Another important concept is the Hodge-$\star$. Since real physics depends on geometry, the $\star$-operator is a map of a k-form to a $(n-k)$-form. It is important to know where the hodge is doing its work, as metric relation occurs where two different k-forms are computed while this is impossible in a discrete representation. Using the $\star$, one of the most important diagrams can be constructed, namely the deRham Complex:

$$\begin{array}{ccccccccccc}
\mathbb{R} & \longrightarrow & \Lambda^{(0)}(\Omega) & \xrightarrow{d} & \Lambda^{(1)}(\Omega) & \xrightarrow{d} & \Lambda^{(2)}(\Omega) & \xrightarrow{d} & \Lambda^{(3)}(\Omega) & \xrightarrow{d} & 0 \\
 & & \downarrow\star & & \downarrow\star & & \downarrow\star & & \downarrow\star & & \\
0 & \xleftarrow[d]{} & \Lambda^{(3)}(\Omega) & \xleftarrow[d]{} & \Lambda^{(2)}(\Omega) & \xleftarrow[d]{} & \Lambda^{(1)}(\Omega) & \xleftarrow[d]{} & \Lambda^{(1)}(\Omega) & \longleftarrow & \mathbb{R}
\end{array}\ . \tag{2.6}$$

The last important fundamental operator for this research, is the interior or interior product. The interior product is the contraction of a vector field with a tensor and is written as $i_{v}a^{(k)}$. This is an antiderivative, as it maps a k-form to a $(k-1)$-form. When the interior product is taken of a 0-form, the result will thus be zero. Just like the exterior derivative, the interior product satisfies a Leibniz rule, in analogy with equation 2.3b:

$$i_{v}\left(a^{(k)}+b^{(m)}\right)=i_{v}a^{(k)}\wedge b^{(m)}+(-1)^{k}a^{(k)}\wedge i_{v}b^{(m)}. \tag{2.7}$$

When an interior product is taken from a p-form, the following results can be found:

$$i_{v}a^{0}=0, \tag{2.8}$$

$$i_{v}a^{1}=a(v), \tag{2.9}$$

$$i_{v}a^{p}\left(w_{2},...,w_{p}\right)=a^{p}\left(v,w_{2},...,w_{p}\right). \tag{2.10}$$

The interior product is additive as well, hence $i_{A+B}=i_{A}+i_{B}$. When a function f and vector field X are considered, the following is valid as well:

$$i_{fX}=fi_{X}. \tag{2.11}$$

Having discussed the exterior derivative and interior product, it is time to bring up the Lie derivative. The Lie derivative describes the change of differential form in direction of the vector field. In fluid dynamics it describes the convective part of the flow. In [2], an analogy with a fisherman is made:

> The flow carries all possible differential-geometric objects past the fisherman, and the fisherman, sits there and differentiates them. [2]

The fisherman's analogy results in the dynamic definition of the Lie derivative:

$$\mathcal{L}_v a^k = \frac{d}{dt}\Big|_{t=0} \int_{\Omega_t} a^k = \frac{d}{dt}\Big|_{t=0} \int_{\Omega_t} \Phi_t^\star a^k, \tag{2.12}$$

Where Ω_t is $\phi_t(\Omega)$. For this research, the Lie derivative defined by Cartan's magic formula is used. Cartan's magic formula reads:

$$\mathcal{L}_v a^k = d i_v a^k + i_v d a^k. \tag{2.13}$$

The relevance with the interior product and exterior derivative is now clear.
The Lie derivative has some useful properties. Since both the interior product as exterior derivative follow the Leibniz rule, the Lie derivative does so as well. The Lie derivative and the exterior derivative commute with each other. This property can be found in [33] and follows directly from (2.13) and the fact that $d^2 = 0$. This property useful for this thesis research. These two properties are fully expanded written as:

$$\mathcal{L}_v \left(a^{(k)} \wedge b^{(m)}\right) = \left(\mathcal{L}_v a^{(k)}\right) \wedge b^{(m)} + a^{(k)} \wedge \left(\mathcal{L}_v b^{(m)}\right), \tag{2.14}$$

$$d\left(\mathcal{L}_v a^{(k)}\right) = \mathcal{L}_v d a^{(k)}. \tag{2.15}$$

In order to represent a continuous solution in finite amount of elements, reduction operators are used. Reduction operators represent the solution with a set of polynomials. Recent works by Gerritsma [19], showed that a commutation diagram of the Lie derivative and the reduction operator can be constructed, such that a discrete Lie operator can be made. The commutive diagram reads:

$$\begin{array}{ccc} f(\Omega) & \xrightarrow{\quad\mathcal{L}_v\quad} & \mathcal{L}_v f \\ \downarrow{\scriptstyle \mathcal{R}_p^{(1)}} & & \downarrow{\scriptstyle \mathcal{R}_p^{(0)}} \\ \left(f_p, g_{1,p}, \cdots, g_{n,p}\right) & \xrightarrow{\quad \mathrm{L}_v \quad} & c^1 g_{1,p} + \cdots + c^n g_{n,p} \end{array}. \tag{2.16}$$

In other words, we have for all $f \in C^m(\Omega), m \geq 1$:

$$\mathcal{R}_p^{(0)}\left(\mathcal{L}_v f\right) = \mathrm{L}_v \mathcal{R}_p^{(1)}(f) \tag{2.17}$$

For several physical problems, especially those which are non-linear, a connection between k-forms and vector fields is necessary. For this, two operators can be used, the $\flat$ and $\sharp$ operator. The $\flat$ operator transform a vector $v(\Omega)$ to a k-form:

$$\flat v(\Omega) = \Lambda^1(\Omega). \tag{2.18}$$

The $\sharp$ operator does the opposite:

$$\sharp \Lambda^1(\Omega) = v(\Omega). \tag{2.19}$$

2.4. Conclusion

In this chapter, reference methods have been explored. Emphasis was on FV and FE methods, as these are the most common within CFD, but also the most relevant for the method proposed in this research. For FV, the MUSCL scheme with appropriate limiters is good reference method. These schemes are relatively easy to program and a wide selection of limiters is available. The MUSCL scheme have been proven to give good results. Other options, such as WENO reconstruction, are proven as well, but are more complicated to implement.
For comparison with other FE methods, a Galerkin scheme is selected. For this, basis functions are piece-wise linear or quadratic. In cases more stability is required, an upwind Petrov-Galerkin scheme is used. Again, these methods are selected as these are relatively easy to implement. For the more complicated problems, SUPG and DG methods are popular and proven. However, these methods can be complicated and implementing correctly is an art on itself.
In this chapter, the powerful properties of differential geometry have been reviewed as well. Within this framework, operators are available which are independent of geometry or metric. The metric are then inserted using specialized operators, such as the Hodge. In this framework, the Lie derivative describes the advection of any n-form in a vector field. For the new advection operator, the Lie derivative will be used.

3

Relevant Equations

In this chapter, the three sets of equations relevant for this research are discussed. Each set of equations has different characteristics and thus different methods are required in order to solve the equations. The three equation sets are all one-dimensional, but increase in difficulty. The first set, linear advection, is a fully linear problem and should be very easy to solve numerically. The second problem, the inviscid Burgers equation, is non-linear, but still has only advection. Lastly, the Euler equations in one dimension are investigated. This is set of equations consists of three equations and besides advection, also includes pressure terms. This can be solved for simple, one-dimensional problems. However, when these problems become more complicated, the analytical solution becomes extremely complicated or even impossible, hence numerical schemes are necessary. Therefore, the three equations discussed in this chapter, provide an excellent benchmark for new numerical methods.

3.1. Linear Advection

In linear advection, the initial solution is advected with a constant speed a. As a result, the initial solution should keep exactly the same shape and magnitude, only advected with a distance of at. The governing equation is:

$$\frac{\partial \rho}{\partial t} + a \cdot \nabla \rho = 0. \tag{3.1}$$

One could say that the solution is constant along the characteristic $x = at$ in the $x - t$ frame. All characteristics have the same slope, hence never intersect or diverge from each other. The exact solution is easily found in this way:

$$u(x,t) = u(x - at, 0). \tag{3.2}$$

For numerical schemes, the challenge is to keep the shape of the solution intact. Especially when the initial conditions are discontinues, have extremely large gradients, or are non-smooth in other ways, numerical schemes will have difficulties. With discontinuities, either overshoots or dissipative behaviour is likely to occur.
For linear advection problems, one can use the fact that all characteristics are parallel to each other. With this fact, one could select the local value $a\Delta t$ upstream. Another option is to demand that local solutions are within the upwind set of solutions. In that case, the value of a local point, has to be contained in the cell upwind to that point at the previous time level.

3.2. Inviscid Burgers Equations

The Burgers equation is named after Johannes Martinus Burgers. This equation has proven its use, as it is very suitable to get insights into shock formation as well as turbulence. The complete equation consists of a non-linear advection term and a second order spatial derivative. Due to the second order spatial derivative in only one-dimension, the first insights into turbulence can be gained, even though turbulence is an essentially three-dimensional physical phenomenon and the Burgers equation is only

one-dimensional. When the second order derivative is set to zero, the inviscid version of the Burgers equations is attained. This equations is extremely useful as well, as this can be used for the study of shock formation.
In the inviscid Burgers equation, advection happens with the value of the conserved variable. When one would make an analogy to fluid dynamics, this would be the momentum equation with a mass density of one. The governing equation for this is:

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = 0, \tag{3.3}$$

or in the conservative form:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\nabla \cdot (uu) = 0, \tag{3.4}$$

which in one dimension becomes:

$$\frac{\partial u}{\partial t} + \frac{1}{2} frac \partial u^2 \partial x = 0. \tag{3.5}$$

The local characteristics have a slope of u in the $x-t$ plane, which means they are either diverging from, or converging to each other. In case intersection of the characteristics occurs, two or more solutions to the problem exist. Since this is not possible, conservation of u results in a single value. At this point a shock occurs. A sketch of this is shown in Figure 3.1. In order to determine the exact solution at the shock, the conservation of u has to be satisfied. Therefore, the value at the shock is equal to the average of u of all intersecting characteristics.

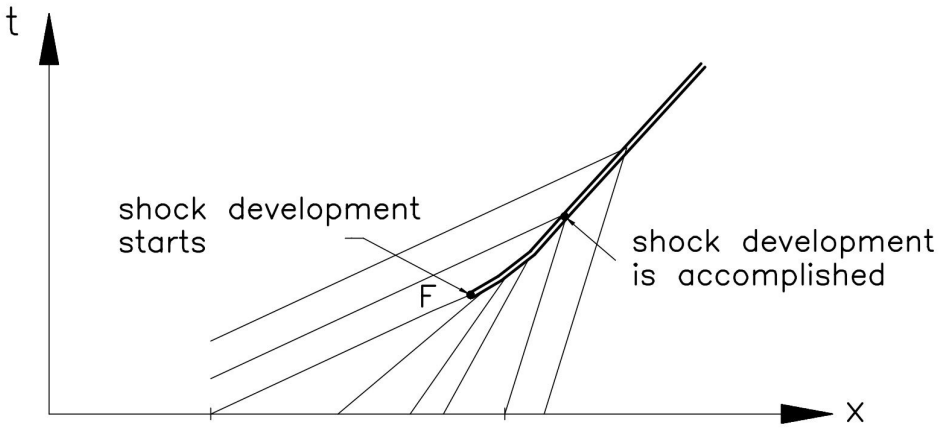


Figure 3.1: Characteristics intersecting in Burgers equations [3]

3.3. One-Dimensional Euler Equations

The Euler equations are the inviscid form of the full Navier-Stokes equations for fluid dynamics. This means the set consist of three equations, one for continuity, one for momentum and one for energy. The continuity and energy equations state that mass and energy cannot be created, nor destroyed. The momentum equation is a force balance.
To setup the equations, a fixed control volume is considered. The change in mass inside the volume has to be equal to the difference between in- and outflow. The resulting equation is [1]:

$$\frac{\partial}{\partial t} \iiint_{\mathcal{V}} \rho d\mathcal{V} + \oiint_S \rho \mathbf{V} \cdot \mathbf{dS} = 0. \tag{3.6}$$

For momentum, the change is not only due to in- and outflow, but also due to forces acting on the control volume. The full momentum equation reads [1]:

$$\frac{\partial}{\partial t} \iiint_{\mathcal{V}} \rho \mathbf{V} d\mathcal{V} + \oiint_S (\rho \mathbf{V} \cdot \mathbf{dS}) = - \oiint_S p \mathbf{dS} + \iiint_{\mathcal{V}} \rho \mathbf{f} d\mathcal{V} + \mathcal{F}_{visc}. \tag{3.7}$$

In the Euler equations it is assumed that the viscous forces as well as the body forces are equal to zero. As a result the last two terms of the right hand are zero. The first term on the left hand side is the time rate of change of momentum in the control volume. The second term is transport of momentum out of the control volume and the pressure term on the right hand side is the work done by pressure on the boundaries.
The last equation is the energy conservation equation. Besides advection, energy can be added in several ways. For the control volume, the full energy equation reads [1]:

$$\iiint_{\mathcal{V}} \dot{q}\rho d\mathcal{V} + \dot{Q}_{visc} - \oiint_S p\mathbf{V}\cdot\mathbf{dS} + \iiint_{\mathcal{V}} \rho\left(\mathbf{f}\cdot\mathbf{V}\right) d\mathcal{V} + \dot{W}_{visc} = \frac{\partial}{\partial t}\iiint_{\mathcal{V}} \rho\left(e+\frac{V^2}{2}\right) d\mathcal{V} + \oiint_S \rho\left(e+\frac{V^2}{2}\right)\mathbf{V}\cdot\mathbf{dS}. \tag{3.8}$$

On the left hand side of this equation, the first term is the energy change of volumetric heat, which is zero when no heat is added and adiabatic boundaries are assumed. The second term is the heat added by viscous effects and is zero since no viscosity is assumed. The third term is work done by pressure, the fourth term is work done by body forces and can often be ignored. Finally $\dot{W}_{visc}$ is work done by viscosity, which is again zero. On the right hand the first term is the time rate of change of total energy and the second term is the in- and outflow of total energy.
With the Equations (3.6) to (3.8), the Euler equations in one dimension can be written in the form:

$$\frac{\partial}{\partial t}U + \frac{\partial}{\partial x}F(U) = 0, \tag{3.9a}$$

in which U and F are defined as:

$$U = \begin{bmatrix}\rho\\ m^x\\ \rho E\end{bmatrix}, \qquad F = \begin{bmatrix}\rho u\\ p+um^x\\ u(\rho E+p)\end{bmatrix}. \tag{3.9b}$$

In these equations, ρ is the mass density, m^x is the x-momentum and ρE is the energy density. In terms of the Lie derivative, this system of equations can be written as:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ m^x\\ \rho E\end{bmatrix} + \mathcal{L}_u\begin{bmatrix}\rho\\ m^x\\ \rho E\end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix}0\\ p\\ up\end{bmatrix} = 0. \tag{3.10a}$$

The system of equations are closed with the velocity and pressure relations:

$$u = \frac{m^x}{\rho}, \tag{3.10b}$$

$$p = (\gamma-1)\left(\rho E - \frac{1}{2}\frac{{m^x}^2}{\rho}\right). \tag{3.10c}$$

Additional relations which are generally valid are:

$$a = \sqrt{\frac{\gamma p}{\rho}}, \tag{3.11a}$$

$$T = \frac{p}{\rho R}, \tag{3.11b}$$

which are the speed of sound a and the temperature T. R is the specific gas constant and is 287 for air. For simple, one-dimensional problems, the Euler equations can be solved manually with the method of characteristics and with local Riemann problems. In the $x-t$ plane, these characteristics run with u, $u+a$ and $u-a$, with a the speed of sound. Depending on initial conditions, the $u+a$ and $u-a$ characteristics are associated with either a shock-wave or an expansion fan, along the u characteristic a contact discontinuity occurs. The complete initial value problem is called a Riemann problem, of which an example can be found in Figure 3.2.
Depending on the occurrence of shock or expansion, the solution is calculated differently. Naturally, both cases should adhere all conservation equations. However, with expansion, the fluid is expected to

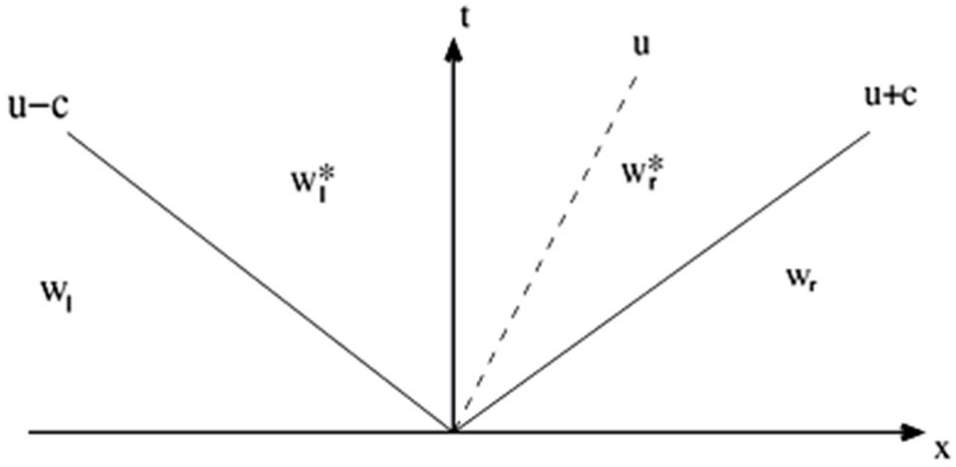


Figure 3.2: sketch of characteristics, taken from [21]

behave isentropic, hence entropy is constant. On the other hand, a shock is not isentropic. In the case of a shock, entropy increases. When the jump is associated with a shock, the conditions behind the shock can be computed using the Hugoniot curve. Conditions after the expansion fan are computed using the Poisson curve.
The Hugoniot curve, for shocks, calculates the mass flux m through the shock as:

$$m = \rho_{pre}|c_s| = \rho_{pre}a_{pre}\sqrt{a + \frac{\gamma+1}{2\gamma}\frac{\Delta p}{p_{pre}}}. \tag{3.12}$$

Also, a relation between pressure difference and velocity difference exist:

$$p_{post} - p_{pre} = \pm m\left(u_{post} - u_{pre}\right). \tag{3.13}$$

With these two equations, all conditions after the shock can be computed as function of the conditions prior to the shock. The Hugoniot curve has also the following property:

$$p_{post} - p_{pre} = \rho_{pre}a_{pre}\sqrt{1 + \frac{\gamma+1}{2\gamma}\frac{p_{post} - p_{pre}}{p_{pre}}}\left(u_{post} - u_{pre}\right). \tag{3.14}$$

Similarly, the Poisson curve can be found as well. In Figure 3.3, the characteristics of an expansion fan are shown. In this, the first characteristic on the high pressure side is called the head. The last characteristic of the fan, which is on the low pressure side, is called the tail. Within a shocktube problem the characteristics are usually straight, since conditions before and after the fan are constant. With the isentropic relations, the states along a characteristic going through the expansion wave gives:

$$u_{post} - u_{pre} = \frac{2}{\gamma-1}\left\{\left(\frac{p_{post}}{p_{pre}}\right)^{\frac{\gamma-1}{2\gamma}} - 1\right\}a_{pre}. \tag{3.15}$$

In this equation, a_{pre} is the speed of sound before the expansion wave has passed. Again, Equation 3.13 is valid. With this, the mass flux through the expansion fan can be expressed as:

$$m = \rho_{pre}a_{pre}\frac{\gamma-1}{2\gamma}\frac{1 - \frac{p_{post}}{p_{pre}}}{1 - \left(\frac{p_{post}}{p_{pre}}\right)^{\frac{\gamma-1}{2\gamma}}}. \tag{3.16}$$

The solution within the expansion fan can also be solved, since for each left running characteristic running through a right running expansion fan, the following is valid:

$$u + \frac{2a}{\gamma-1} = const. \tag{3.17}$$

Also, the slope of each characteristic in an expansion fan is defined as:

$$\frac{\partial x}{\partial t} = u + a. \tag{3.18}$$

The local velocity u can now be calculated:

$$\begin{aligned} u(x,t) &= \frac{2}{\gamma+1}\left(\frac{x^*}{t} + \frac{\gamma-1}{2}u_r + a_r\right). \\ &= \frac{2}{\gamma+1}\left(\frac{x^*}{t} + \frac{\gamma-1}{2}u_l + a_l\right). \end{aligned} \tag{3.19}$$

For the local speed of sound, a similar solution can be found:

$$\begin{aligned} a(x,t) &= \frac{2}{\gamma+1}\left(\frac{x^*}{t} + \frac{\gamma-1}{2}u_r + a_r\right) - \frac{x^*}{t}. \\ &= \frac{2}{\gamma+1}\left(\frac{x^*}{t} + \frac{\gamma-1}{2}u_l + a_l\right) - \frac{x^*}{t}. \end{aligned} \tag{3.20}$$

In both equations, for u and a, x^* is the relative x-location, hence $x^* = x - x_0$. With the isentropic relations, the pressure and density can be computed. The Poisson curve has the following property:

$$\left(\frac{p_{post}}{p_{pre}}\right)^{\frac{\gamma-1}{2\gamma}} = 1 + \frac{\gamma-1}{2}\frac{u_{post} - u_{pre}}{a_{pre}}. \tag{3.21}$$

With the relations between pre and post conditions of shocks and expansion, the complete solution can be found. For this, the solution is split into four area's. The initial conditions w_l and w_r, just like Figure 3.2, and the conditions after the left and respectively right running characteristics $w_l^\star$ and $w_r^\star$. For the complete solution, it is known that both the velocities and pressures of area $w_l^\star$ and $w_r^\star$ are equal. The only variable which is discontinues between both area's, is the density.
Typically, there are several issues found when the Euler equations are solved numerically. Problems can occur with shock velocity, which will result in wrong characteristic wave speeds. Also possible when computing numerically but not in the physical world, is the occurrence of expansion shocks. In that case, the expansion fan has a shock-like discontinuity. Besides these problems, the usual issues can occur, such as oscillations near jumps, lack of conservation and instability. This last issue generally occurs when negative pressure is present in the solution. In the works of Toro [46] and Laney [30], numerical methods for compressible flows are discussed in more detail.

3.4. Conclusions

The presented equations should give sufficient test possibilities for new numerical methods. The first two equations sets, linear and non-linear advection, will give a good insight of the performance of the advection operator. With the Euler equations, interaction of advection with pressure can be investigated. For all equations, an exact solution can be found for one-dimensional problems with the steps discussed in this chapter. Due to that, performance for each method can be compared.

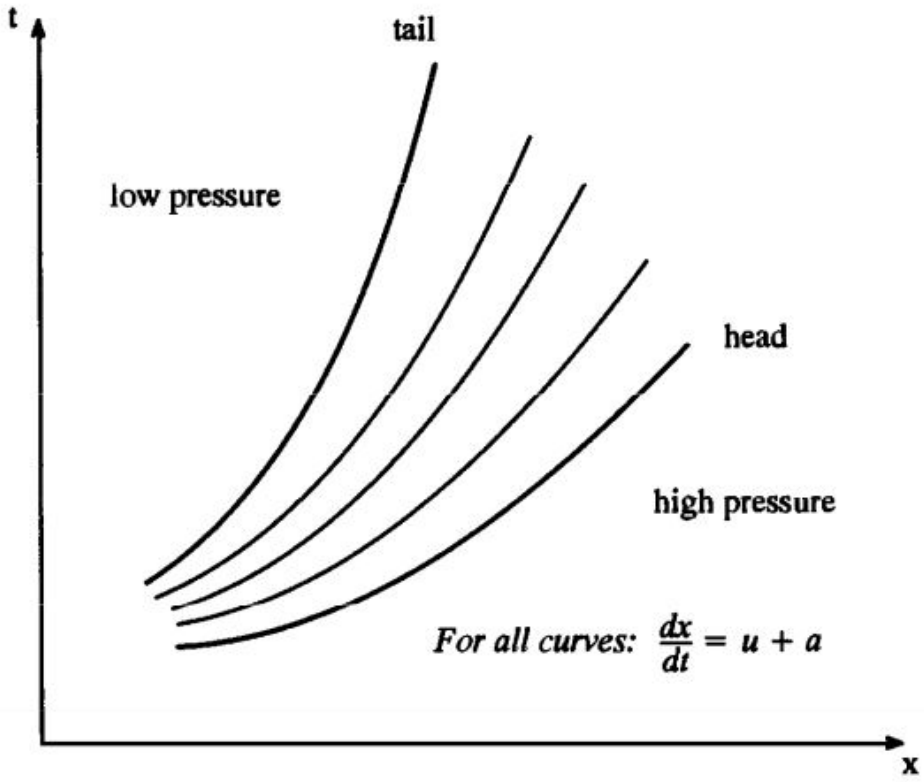


Figure 3.3: sketch of characteristics during expansion, taken from [30]

4

Proposed Method

In this chapter, the foundation of the method is laid. First the polynomials used for the proposed method are discussed. Second, the idea for the method is given, followed by an example in which this is worked out for the one-dimensional linear advection equation. Finally, several possible techniques for stabilization and reducing spurious oscillations are discussed.

4.1. Polynomials

For the method a set of polynomials is derived from Hermite polynomials. The Hermite polynomials, which are from now on written as h-polynomials, are C^1 continuous polynomials. The polynomials are defined as:

$$\begin{aligned}
h_0^0(-1) &= 1, & h_1^0(-1) &= 0, & h_0^1(-1) &= 0, & h_1^1(-1) &= 0, \\
h_0^0(1) &= 0, & h_1^0(1) &= 1, & h_0^1(1) &= 0, & h_1^1(1) &= 0, \\
\frac{\partial}{\partial x} h_0^0(-1) &= 0, & \frac{\partial}{\partial x} h_1^0(-1) &= 0, & \frac{\partial}{\partial x} h_0^1(-1) &= 1, & \frac{\partial}{\partial x} h_1^1(-1) &= 0, \\
\frac{\partial}{\partial x} h_0^0(1) &= 0, & \frac{\partial}{\partial x} h_1^0(1) &= 0, & \frac{\partial}{\partial x} h_0^1(1) &= 0, & \frac{\partial}{\partial x} h_1^1(1) &= 1.
\end{aligned} \tag{4.1}$$

In these notations, the subscript indicates if the polynomial is associated with the left or right boundary, 0 being the left boundary, 1 the right boundary. The superscript associates the polynomials with the derivative order. ξ is defined in the domain $[-1, 1]$. The h-polynomials can be seen in Figure 4.1a. From these polynomials, a special sort of edge polynomials, from now on called e-polynomials, can be derived. The e-polynomials are defined as the derivatives of the h-polynomials. This results in the following relations:

$$e_0(\xi) = dh_0^1(\xi), \qquad e_1(\xi) = dh_1^0(\xi) = -dh_0^0(\xi), \qquad e_2(\xi) = dh_1^1(\xi). \tag{4.2}$$

The resulting polynomials can be seen in Figure 4.1b These polynomials have several useful properties, which can be written as:

$$\begin{aligned}
e_0(-1) &= 1, & e_0(1) &= 0, & \int_{-1}^{1} e_0(\xi) &= 0, \\
e_1(-1) &= 0, & e_1(1) &= 0, & \int_{-1}^{1} e_1(\xi) &= 1, \\
e_2(-1) &= 0, & e_2(1) &= 1, & \int_{-1}^{1} e_2(\xi) &= 0.
\end{aligned} \tag{4.3}$$

The properties listed in the first two columns follow directly from the definition of the e-polynomials and the derivative values of the h-polynomials on the boundary of the cell. The properties listed in the last

column, can be verified using Stokes Theorem:

$$\int_{\partial\Omega} \omega = \int_{\Omega} d\omega. \tag{4.4}$$

This results for the $h-$ and e-polynomials in:

$$\int_{\partial\Omega} h = \int_{\Omega} dh = \int_{\Omega} e. \tag{4.5}$$

The e-polynomials as function of ξ are:

$$e_0(\xi) = \frac{3}{4}(\xi+1)^2 - 2(\xi+1) + 1 \tag{4.6a}$$

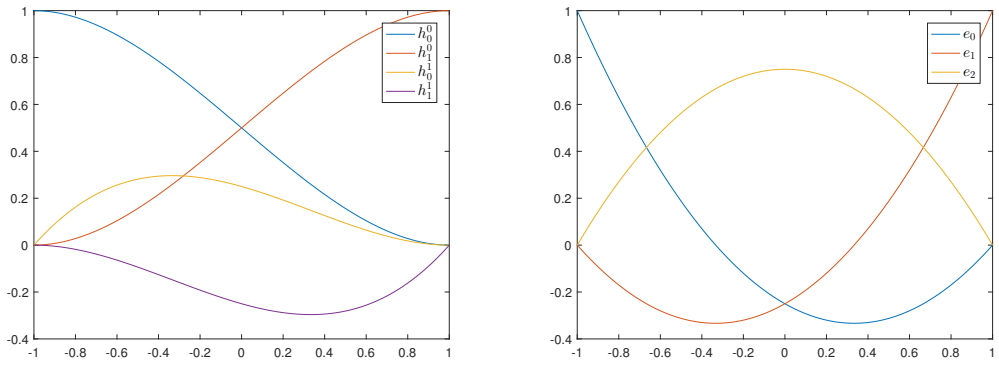
$$e_1(\xi) = \frac{6}{8}(\xi+1)(1-\xi) \tag{4.6b}$$

$$e_2(\xi) = \frac{3}{4}(1-\xi)^2 - 2(1-\xi) + 1 \tag{4.6c}$$

How these functions are derived is discussed in more detail in Appendix A. With the e-polynomials, the solution within a cell can be reconstructed as:

$$u(\xi) = u_L e_0(\xi) + u_I e_1(\xi) + u_R e_2(\xi), \tag{4.7}$$

In this, u can be any conserved variable. The subscripts L and R denote the left respectively right boundary of the cell, subscript I denotes the value of u integrated over the cell. The left and right boundary of the cell are often called cell faces.
Now it becomes important to define the grid. Cell faces are located on $x \pm \frac{1}{2}$, the integrated values on x_i. For scaling of the domain from $[-1,1]$ to $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, e_1 needs to be scaled with $\frac{2}{\Delta x}$, with $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$. This is necessary to ensure the polynomial integrates to 1 over the domain Δx.



(a) h-polynomials (b) e-polynomials

Figure 4.1: Polynomials

4.2. Idea

The e-polynomials as derived in the previous section are used. For any k-form the e-polynomials in the required dimension will be used. In fluid dynamics, the appropriate form for conserved variables is the n-form, with n the spatial dimensions. On the grid, the n-forms need to be defined. For any conserved variable u, the n form is represented as in (4.7).
Any PDE with an advection term is considered. For the advection terms in this PDE the Lie derivative, $\mathcal{L}_a$ is used, in which a is the velocity field. The Lie derivative for any n-form will result in a function of the velocity and the value of the variable the Lie derivative is applied on. Using Cartans magic formula,

see (2.13), on finds:

$$\begin{aligned}\mathcal{L}_a u^n &= d i_a u \\ &= d a u \\ &= \frac{\partial a u}{\partial x} dx \end{aligned} \tag{4.8}$$

Which can be integrated over a domain using Stokes equations in order to get the discrete Lie derivative:

$$\begin{aligned}\int_L^R \mathcal{L}_a u^n &= a u|_L^R \\ &= a_R u_R - a_L u_L. \end{aligned} \tag{4.9}$$

In this equation, u_R and u_L are again the values on the right respectively left cell face. Similarly, a_R and a_L are the advective velocities on these locations. Ideally, this would lead to the following solution:

$$\int_{-1}^{1} \frac{\partial \rho_L}{\partial t} e_0(\xi) + \int_{-1}^{1} \frac{\partial \rho_I}{\partial t} e_1(\xi) + \int_{-1}^{1} \frac{\partial \rho_R}{\partial t} e_2(\xi) + (-\rho_L a_L + \rho_R a_R) + \int_{-1}^{1} f d\xi = 0. \tag{4.10}$$

With the polynomial propreties listed in (4.3), this results in:

$$\frac{\partial \rho_I}{\partial t} = -(-\rho_L a_L + \rho_R a_R) - \int_{-1}^{1} f d\xi, \tag{4.11}$$

with f all other terms of the PDE. The values on the boundaries can be computed with FE. Unfortunately, this gave no satisfactory results and will therefore not be further discussed in this thesis. Example results of why this idea was not further pursuit can be found in Appendix B. Instead, a Galerkin method is applied in order to solve the PDE. This means, the PDE is weighted with test functions, which are the same as the functions used to reconstruct the solution. This leads to a weak solution in the form of:

$$\begin{aligned}&\text{find } u \in V \text{such that for all } v, \\ &\left(\left(\mathcal{L}_a u\right), v\right) + (f, v) = 0. \end{aligned} \tag{4.12}$$

In which V is a finite dimensional subspace. f in this case holds the other terms of the PDE, such as the time derivative, stresses and external forces. Since advection is a skew-symmetrical form when Galerkin is applied, (4.12) can be modify to have half the Lie derivative on u and half the Lie derivative on v:

$$\begin{aligned}&\text{find } u \in V \text{such that for all } v, \\ &\frac{1}{2}\left(\left(\mathcal{L}_a u\right), v\right) - \frac{1}{2}\left(u, \left(\mathcal{L}_a v\right)\right) + (f, v) = -\frac{1}{2}\left[a u v\right]_a^b. \end{aligned} \tag{4.13}$$

The first term of the right hand side will be zero when periodic boundary conditions are applied. When other boundary conditions are used, this term has to be taken into account. The proof of this skew-symmetry is found when the advection term multiplied with a test function is integrated over the domain:

$$\begin{aligned}\int_a^b \frac{\partial\left(a u\right)}{\partial x} v dx &= \left[a u v\right]_a^b - \int_a^b a u \frac{\partial v}{\partial x} dx, \\ &= \frac{1}{2}[a u v]_a^b - \frac{1}{2}\int_a^b a u \frac{\partial v}{\partial x} dx + \frac{1}{2}\int_a^b \frac{\partial\left(a u\right)}{\partial x} v dx. \end{aligned} \tag{4.14}$$

The term on the right hand side is zero when periodic boundary conditions are used, otherwise this term is added to the boundary terms.
The PDE can be written as a system of matrix equations. This looks like:

$$\mathbb{M}\frac{\partial \rho}{\partial t} + \mathbb{F}\rho + \mathbb{G}\rho = 0. \tag{4.15}$$

In this equation $\mathbb{M}$ is the mass element matrix, which is just like any other Galerkin method:

$$\mathbb{M}_{i,j} = \langle \rho_i, v_j \rangle = \int_\Omega \rho_i v_j d\Omega. \tag{4.16}$$

The matrix $\mathbb{G}$ contains all source and non-convection terms of the PDE, as well as the boundary conditions. The matrix $\mathbb{F}$ is the flux matrix. This matrix will contain the $\mathcal{L}_a$ terms for the polynomials associated with the integrated cell value, and for the other polynomials:

$$\mathbb{F}_{i,j} = \left\langle \frac{\partial(a\rho_i)}{\partial x}, v_j \right\rangle, \tag{4.17}$$

or when the skew-symmetric form is used:

$$\mathbb{F}_{i,j} = \frac{1}{2}\left\langle \frac{\partial(a\rho_i)}{\partial x}, v_j \right\rangle - \frac{1}{2}\left\langle a\rho_i, \frac{\partial v_j}{\partial x} \right\rangle, \tag{4.18}$$

In the following section this approach is worked out for the one dimension advection equation.

4.2.1. Boundary Conditions

Boundary conditions can be applied just as any other FE scheme. This usually means rewriting of the equations is required. When diffusion is present in the governing equations, this naturally occurs when the weak form for the diffusion term is made. For pure advection problems, these boundary conditions are simply forced on the boundary of the domain. In this thesis, most of the tests use periodic boundary conditions. In these cases the boundary conditions completely reappear on the right hand side.

4.3. Application on Linear Advection in 1D

Linear advection is usually written in the convective or conservative form, the latter being:

$$\frac{\partial \rho}{\partial t} + \frac{\partial a\rho}{\partial x} = 0. \tag{4.19}$$

In differential geometry this is written as

$$\frac{\partial \rho^{(1)}}{\partial t} + \mathcal{L}_a \rho^{(1)} = 0. \tag{4.20}$$

The superscript denote that ρ is a 1-form in one dimensional space, hence an integration of ρ in each cell is performed. In a n-dimensional space ρ will be a n-form. As can be seen, the spatial derivative is replace with $\mathcal{L}_a$, which is the Lie derivative. The Lie derivative for any k-form is defined as:

$$\mathcal{L}_a \rho^{(k)} = i_a d\rho^{(k)} + d i_a \rho^{(k)}.$$

Because ρ is a n-form in a n-dimensional space applying the d directly is zero the Lie derivative for density ρ simplifies to $d i_a \rho^{(k)}$. This results in:

$$\mathcal{L}_a \rho^{(1)} = \frac{\partial a\rho}{\partial x}. \tag{4.21}$$

Because ρ is a volume-form, it will be reconstructed with the edge function:

$$\rho(x) = \rho_L e_0(x) + \rho_I e_1(x) + \rho_R e_2(x), \tag{4.22}$$

Now the Galerkin method can be applied on (4.20). This results in the following formulation:

$$\begin{gathered} \text{find } \rho \in V \text{such that for all } v, \\ \left(\frac{\partial \rho}{\partial t}, v\right) + \left(\left(\mathcal{L}_a \rho\right), v\right) = 0. \end{gathered} \tag{4.23}$$

Since linear advection is a symmetric form, the skew-symmetric notation will be used. This results in:

$$\text{find } \rho \in V \text{such that for all } v,$$
$$\left(\frac{\partial \rho}{\partial t}, v\right) + \frac{1}{2}\left(\left(\mathcal{L}_a \rho\right), v\right) - \frac{1}{2}\left(\rho, \left(\mathcal{L}_a v\right)\right) = -\frac{1}{2}\left[auv\right]_a^b. \tag{4.24}$$

After weighting of the functions the following system of equations is found, similar to the System 4.15:

$$\mathbb{M}\frac{\partial \rho}{\partial t} + \mathbb{F}\rho = 0. \tag{4.25}$$

In this equation, the elements of $\mathbb{M}$ will be:

$$\mathbb{M}_{i,j} = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} e_i(x) e_j(x) dx, \tag{4.26}$$

and $\mathbb{F}$ will be:

$$\mathbb{F}_{i,j} = \frac{1}{2}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{e_i(x)}{\partial x} e_j(x) - \frac{1}{2}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{e_j(x)}{\partial x} e_i(x) dx \tag{4.27}$$

In this local matrix corresponds with $\rho = \begin{bmatrix}\rho_L & \rho_I & \rho_R\end{bmatrix}^T$. The matrix can be assembled to a global matrix using any FEM construction technique. This usually means a gathering matrix is constructed, in which all cells are connected with its boundaries.
The matrix $\mathbb{F}$ has some remarkable features, which can be found in the row weighted with the e_I polynomial. When a is constant or when a is reconstructed with the same polynomials $F_{1,1}$ becomes

$$\begin{aligned}
F_{1,1} &= \frac{1}{2}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial a e_1(x)}{\partial x} e_1(x) - \frac{1}{2}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial a e_1(x)}{\partial x} e_1(x) dx, \\
&= \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{e_i(x)}{\partial x} e_j(x) dx, \\
&= \left[e_1(x) e_1(x)\right]_L^R - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} e_1(x) \frac{e_1(x)}{\partial x} dx, \\
&= 0.
\end{aligned} \tag{4.28}$$

This is found with the properties defined in 4.3. Since this result is zero, it is independent of scaling. Again with a constant or scaled with the same polynomials and some more effort, a second term can be found:

$$\begin{aligned}
F_{0,1} &= \frac{1}{2}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial a e_0(x)}{\partial x} e_1(x) - \frac{1}{2}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial a e_0(x)}{\partial x} e_1(x) dx, \\
&= \frac{\Delta x}{2}\int_{-1}^{1} \left(\frac{2}{\Delta x}\right) \frac{\partial a e_0(\xi)}{\partial \xi} \left(\frac{2}{\Delta x}\right) e_1(\xi)\, d\xi \\
&= \frac{2}{\Delta x}\int_{-1}^{1} a_L \left[\frac{\partial\left(\frac{3}{4}(\xi+1)^2 - 2(\xi+1) + 1\right)}{\partial x} \frac{6}{8}(\xi+1)(1-\xi)\right] d\xi \\
&= \frac{2}{\Delta x}\int_{-1}^{1} a_L \left[\left(\frac{6}{4}(\xi+1) - 2\right) \frac{6}{8}(\xi+1)(1-\xi)\right] d\xi \\
&= \frac{2a_L}{\Delta x}\left[\frac{36}{32}\int_{-1}^{1} \left(1 + \xi - \xi^2 - \xi^3\right) d\xi - 2\right] \\
&= \frac{2a_L}{\Delta x}\left[\frac{4}{3}\frac{9}{8} - 2\right] = -\frac{a_L}{\Delta x}
\end{aligned} \tag{4.29}$$

The $\frac{2}{\Delta x}$ terms in the second line are due to scaling from $[-1,1]$ to $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$. For $\mathbb{F}_{2,1}$, the result found is $\frac{a_R}{\Delta x}$. The second row of the $\mathbb{F}$ matrix looks quit similar to the discrete Lie derivative of (4.9), but with a $\frac{1}{\Delta x}$ term. This can be shown with:

$$\begin{bmatrix} -\frac{1}{\delta x} & 0 & \frac{1}{\Delta x} \end{bmatrix} \begin{bmatrix} \rho_L \\ \rho_I \\ \rho_R \end{bmatrix} = \frac{a_R \rho_R}{\Delta x} - \frac{a_L \rho_L}{\Delta x} \tag{4.30}$$

The result of (4.29) is the same for both (4.23) and (4.24).
For the time integration an implicit or explicit method can be used, or any other time integration scheme. For now a θ-scheme and a RK scheme are considered. For the θ-scheme (4.25) is extended to:

$$\mathbb{M}\frac{\rho^{n+1} - \rho^n}{\Delta t} + \theta\, \mathbb{F}\rho|^{n+1} + (1-\theta)\, F\rho|^n = 0. \tag{4.31}$$

Usually this scheme is taken with $\theta = 0.5$, which is the Crank-Nicolson scheme. In the case of non-linear problems, the θ scheme is changed to a predictor-corrector type. First an approximation for time level $n+1$ is made by forward Euler. After that, iterations are performed until convergence at time level $n+1$ is found. The reason to use a predictor-corrector time integrator for non-linear problems, is that the velocity field changes each time step, hence the advection matrix needs to be recalculated. Details on the RK time integrator can be found in Appendix C.

4.4. Stabilization Options

For problems in which discontinuities arise, or which are strongly non-linear, stabilization might be necessary. In this section, options to suppress oscillations and enhance stability are discussed. These options are focused on the points left and right of the cell, as the corresponding polynomials integrate to zero. One option is to modify the test functions $e_j(\xi)$ in Equations 4.26 and 4.27 to achieve some upwinding. Upwind methods provide additional stability, although usually at the cost of artificial diffusion. Another option might be correcting overshoots on the cell faces. This method would resembles flux limiting used in Finite Volume methods and can be done since the results found on the cell faces do not contribute to conservation. In this section these two options are discussed.
Since the proposed method is for the spatial discretization of the advection terms within a FEM framework, several options for upwind are available. A popular and proven technique is to use SUPG. This method is easily adapted to multi-dimensions. In general the approach is to add a SUPG term to the normal Galerkin formulation, this leads to:

$$\left\langle \left(\frac{\partial \rho}{\partial t} + u\nabla\rho + f \right), v \right\rangle + SUPG(v,\rho) = 0, \tag{4.32}$$

with $SUPG(v,\rho)$:

$$\left\langle \left(\frac{\partial \rho}{\partial t} + u\nabla\rho + f \right), T\nabla v \right\rangle. \tag{4.33}$$

Since SUPG modifies the basis functions as function of ∇v, this approach has to be performed on all basis function, hence e_0, e_1 and e_2. If one would only do this for the e_0 and e_2 polynomials, the test function do not add up to one any more. In (4.33), T is a constant which modifies the weight function. In order to determine the value of T, many different approaches have been developed. Some simple options are:

$$T = \beta \Delta t, \tag{4.34a}$$

$$T = \beta \frac{2\Delta x}{|a|} \xi(a). \tag{4.34b}$$

$\xi(a)$ is a function of Péclet number and is always 1 for inviscid problems. β is a constant which can be selected, usually a value near a half is taken for this. For more details, the reader can consult work by Hughes [24]. Some other options for T can be found in [45] and [8]. For one dimensional piecewise

linear basis functions, a visual representation of $v + \nabla T$ within the SUPG framework can be found in Figure 4.2a. In this figure, velocity is positive from left to right. The Matrices $\mathbb{M}$ and $\mathbb{F}$ become:

$$\mathbb{M}_{i,j} = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} e_i(x) \left(e_j(x) + T \frac{\partial e_j(x)}{\partial x} \right) dx, \tag{4.35a}$$

$$\mathbb{F}_{i,j} = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial e_i(x)}{\partial x} \left(e_j(x) + T \frac{\partial e_j(x)}{\partial x} \right) dx \tag{4.35b}$$

Another option is the use of bubble functions, which can be added on the left and right polynomials. In that case a function is added which has no value on the boundaries of the cell. The simplest bubble function has the form of:

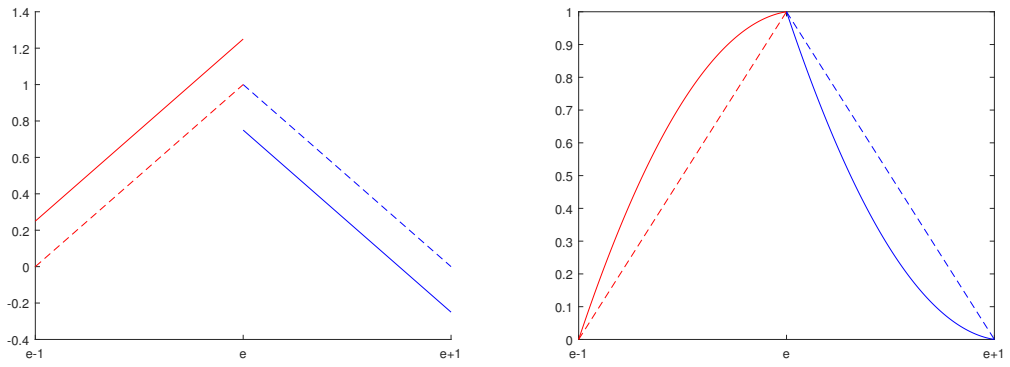
$$B = \frac{3\beta}{4} \left(1 - x^2\right). \tag{4.36}$$

The function is added or subtracted from the test function depending on flow direction. Upstream of the node the bubble function is added, downstream subtracted from the test function. Again for piecewise linear basis functions, a visual representation can be seen in Figure 4.2b, again flow is coming from the left. For the proposed method it is only necessary to modify the basis function associated with the cell faces, so e_0 and e_2. Due to the upwind functions, the test functions associated with the boundaries of the cell do not integrate to zero any more. The row associated with the integrated values does not change. The other terms in $\mathbb{F}$ become:

$$\mathbb{F}_{i,j\neq 2} = \frac{1}{2} \int_{-1}^{1} \frac{\partial a e_i}{\partial x} \left(e_j \pm B \right) dx. \tag{4.37}$$

With plus or minus T depending on location of the node with respect to the flow as discussed before. The full $\mathbb{F}$ matrix becomes the following:

$$\mathbb{F} = \begin{bmatrix} \frac{1}{2}\int_{-1}^{1} \frac{\partial a e_1}{\partial x} \left(e_1 - \frac{a}{|a|} B \right) dx & \frac{1}{2}\int_{-1}^{1} \frac{\partial a e_2}{\partial x} \left(e_1 - \frac{a}{|a|} B \right) dx & \frac{1}{2}\int_{-1}^{1} \frac{\partial a e_3}{\partial x} \left(e_1 - \frac{a}{|a|} B \right) dx \\ \frac{-a_L}{\Delta x} & 0 & \frac{a_R}{\Delta x} \\ \frac{1}{2}\int_{-1}^{1} \frac{\partial a e_1}{\partial x} \left(e_3 + \frac{a}{|a|} B \right) dx & \frac{1}{2}\int_{-1}^{1} \frac{\partial a e_2}{\partial x} \left(e_3 + \frac{a}{|a|} B \right) dx & \frac{1}{2}\int_{-1}^{1} \frac{\partial a e_3}{\partial x} \left(e_3 + \frac{a}{|a|} B \right) dx \end{bmatrix}. \tag{4.38}$$



(a) for SUPG (b) with Bubble functions

Figure 4.2: Modification of weight functions for one dimensional piecewise linear functions. Freely recreated from [8]

Another option in order to stabilize the scheme and prevent strong overshoots, is to introduce flux limiters. Limiters are usually used within the FV framework. For the proposed method, a limiter like method can be used to correct values at the cell boundaries, hence preventing oscillations. In FV schemes, a limiter can be seen as a switch between high and low order flux:

$$F\left(U_{i\pm\frac{1}{2}}\right) = f^{low}_{i\pm\frac{1}{2}} - \phi(r_i) \left(f^{low}_{i\pm\frac{1}{2}} - f^{high}_{i\pm\frac{1}{2}} \right), \tag{4.39}$$

with $f^{low}_{i\pm\frac{1}{2}}$ and $f^{high}_{i\pm\frac{1}{2}}$ the low order and high order flux respectively. ϕ is the limiter function, which is dependent on r, the ratio of successive gradient. For function ϕ, many different options are available, such as minmod, Koren, van Leer and Superbee.
Although the proposed method is within the Finite Element framework, a minmod type correction can easily be implemented. In that case r could be constructed as successive average slopes:

$$r_{i+\frac{1}{2}} = \frac{\rho_{i+\frac{1}{2}} - \rho_{i-\frac{1}{2}}}{\rho_{i+\frac{3}{2}} - \rho_{i+\frac{1}{2}}}. \tag{4.40}$$

This radius can be used in the minmod limiter:

$$\text{minmod}\left(r_{i+\frac{1}{2}}\right) = \max\left\{0, \min(1, r)\right\}. \tag{4.41}$$

This is then used in (4.39), in which the low order flux is used as upwind from the integrated value divided by cell size in the cell and the high order flux the value evolved in time according to the previously discussed for the proposed method. The result is that the values on the cell boundaries are corrected when local extrema are created. In other words, when the values on the cell faces are creating local extrema, a correction is done.
A second correction strategy is developed with this specific scheme and the physical problem it is applied to. For pure advection, the variable ρ moves with the velocity of the flow. The Courant-Friedrichs-Lewy number (CFL) condition states that:

$$C = \frac{a\Delta t}{\Delta x} \leq 1. \tag{4.42}$$

When the value of C becomes higher then 1, instability issues are likely to occur, hence a CFL should always be lower than 1. Using this fact, the solution found for the cell faces should be from the set of local values possible in the cell upwind of that cell face. With the polynomials from Figure 4.1b, the maxima and minima are found at either x_i or at cell faces $x_{i\pm\frac{1}{2}}$. Using (4.6) and (4.7) and the values at time level n, the local maxima and minima of the upwind cell can be computed:

$$\max\left(\rho^n_i(x)\right) = \max\left(\rho^n_{i-\frac{1}{2}}, \rho^n_{i+\frac{1}{2}}, -\frac{\rho^n_{i-\frac{1}{2}}}{4} + \frac{12\overline{\rho_i}^n}{8\Delta x} - \frac{\rho^n_{i+\frac{1}{2}}}{4}\right), \tag{4.43a}$$

$$\min\left(\rho^n_i(x)\right) = \min\left(\rho^n_{i-\frac{1}{2}}, \rho^n_{i+\frac{1}{2}}, -\frac{\rho^n_{i-\frac{1}{2}}}{4} + \frac{12\overline{\rho_i}^n}{8\Delta x} - \frac{\rho^n_{i+\frac{1}{2}}}{4}\right). \tag{4.43b}$$

The solution at each cell face has to be within these bounds. Hence, a correction scheme can be constructed:

$$\rho^{n+1}_{i+\frac{1}{2}} = \min\left[\max\left(\rho^{n+1}_{i+\frac{1}{2}}, \min\left(\rho^n_i(x)\right)\right), \max\left(\rho^n_i(x)\right)\right], \qquad \text{if } a > 0, \tag{4.44a}$$

$$\rho^{n+1}_{i+\frac{1}{2}} = \min\left[\max\left(\rho^{n+1}_{i+\frac{1}{2}}, \min\left(\rho^n_{i+1}(x)\right)\right), \max\left(\rho^n_{i+1}(x)\right)\right], \qquad \text{else.} \tag{4.44b}$$

When the value at the cell face is corrected, the total flux through the cell face changes as well. Normally, the total flux through the cell face is the integral of the flux term, integrated from time level n to $n+1$. If this is a linear function, as would be in a θ scheme, the result will be:

$$\overline{F}\left(\rho_{i+\frac{1}{2}}\right) = \frac{\Delta t}{2}\left(F\left(\rho^{n+1}_{i+\frac{1}{2}}\right) + F\left(\rho^n_{i+\frac{1}{2}}\right)\right). \tag{4.45}$$

Since the value of $\rho_{i+\frac{1}{2}}$ is only corrected on time level $n+1$, the correction for the integrated value ρ_i becomes:

$$\rho^{n+1}_{i,cor} = \overline{\rho}^n_i + \frac{\Delta t}{2}\left[a_{i+\frac{1}{2}}\left(\rho_{i+\frac{1}{2},cor} - \rho_{i+\frac{1}{2}}\right) - a_{i-\frac{1}{2}}\left(\rho_{i-\frac{1}{2},cor} - \rho_{i-\frac{1}{2}}\right)\right]. \tag{4.46}$$

For variables that are not only depending on advection, such as momentum and energy in the Euler equations, one should take into account the effect of pressure gradient, stresses and temperature. From now on this stabilization method will be called 'corrector 1,' since it checks if the solution is

possible for the physical model and corrects accordingly.
Finally, a second correction option is investigated for non-linear problems, hence the Burger's equation and the Euler equations. For this option, oscillations are dampened by demanding that the flux solution lies between the two averages of the neighbouring cells. The correction for the fluxes would then become:

$$\rho_{i+\frac{1}{2}}^{n+1} = \min\left[\max\left(\rho_{i+\frac{1}{2}}^{n+1}, \min\left(\frac{\rho_i^{n+1}}{\Delta x_i}, \frac{\rho_{i+1}^{n+1}}{\Delta x_{i+1}}\right)\right), \max\left(\frac{\rho_i^{n+1}}{\Delta x_i}, \frac{\rho_{i+1}^{n+1}}{\Delta x_{i+1}}\right)\right]. \tag{4.47}$$

Again, a correction for the integrated values of the cell is required. This is, for linear advection, the same as (4.46). The result of the correction of $\overline{\rho_i}$ that might result of an overcorrection of the cell fluxes. Therefore iteration of the corrector is necessary until the steady solution at time level $n+1$ is found. For the remainder of this report, this second correction method will be called 'corrector 2.' It is expected that this corrector method dampens oscillations more, but also adds more diffusion.

5

Linear Advection

In this chapter, the simplest of the one dimensional problems is tested, linear advection. In linear advection, an input condition will move with a fixed velocity, a through the domain. The exact solution has exactly the same shape and magnitude as the initial conditions, only moved a distance of velocity a times the time. Numerical methods however, may not be able to conserve the original shape. Depending on the input several effects can happen due to the discrete representation of the problem. The numerical method can show the wrong propagation velocity, or the solution can damped out due to numerical diffusion. Another issue typically arises when discontinuities are present in the initial conditions. Following the Godunov's theorem, higher order methods usually show spurious oscillations. The solution initially overshoots the discontinuity jump. Due to conservation, the solution will have an correction slightly further from the jump, leading to oscillations around the real solution. Some methods show staircasing, as the height of the discontinuity jump is not matched within a few spatial steps. These are just a few of the errors that might arise due to a numerical method on advection.
In this chapter, first the physical model and the test cases are introduced. After that the base scheme, as proposed in Section 4.3, is compared with several reference schemes. These references are two FE schemes as well as two MUSCL schemes with different limiters applied. After that, the possible improvement that stabilization techniques can give are investigated. Finally a conclusion of the performance on linear advection is made.

5.1. Introducing: Test Cases

Linear advection is a constant advection of a variable, which has the governing equation:

$$\frac{\partial \rho}{\partial t} + a\frac{\partial \rho}{\partial x} = 0, \tag{5.1}$$

Which in terms of differential geometry is written as:

$$\frac{\partial \rho^{(1)}}{\partial t} + \mathcal{L}_a \rho^{(1)} = 0. \tag{5.2}$$

The details of the discrete Lie method on linear advection are previously discussed in Section 4.3. In this case, the mass matrix, $\mathbb{M}$, used is from (4.26). The flux matrix $\mathbb{F}$, is the skew-symmetric form found in (4.31).
To test the new method, a domain with $x\ [0, 1]$ and periodic boundary conditions is used. On this domain several initial conditions are tested. The first test case is a sine input:

$$\rho(x, 0) = \sin(2\pi x). \tag{5.3a}$$

The second test case is a block function input with height 1 and width of $\frac{1}{2}$, which has the initial conditions:

$$\begin{aligned} \rho(x, 0) &= 1, && \text{if } x \geq \frac{L}{4} \text{ and } x \leq \frac{3L}{4}, \\ &= 0, && \text{else.} \end{aligned} \tag{5.3b}$$

The third problem has a constant increasing slope, with a discontinuity in the middle of the domain. Initial conditions for this case are:

$$\begin{aligned} \rho(x,0) &= x, && \text{if } x \leq \frac{1}{2}, \\ &= x-1, && \text{else.} \end{aligned} \tag{5.3c}$$

Finally a hat function is tested. The initial conditions for this case are:

$$\begin{aligned} \rho(x,0) &= 4x-1, && \text{if } x > \frac{1}{4} \text{ and } x \leq \frac{1}{2}, \\ &= 3-4x, && \text{if } x > \frac{1}{2} \text{ and } x \leq \frac{3}{4}, \\ &= 0, && \text{else.} \end{aligned} \tag{5.3d}$$

The method proposed in this research is compared to two Galerkin schemes and two MUSCL schemes. The first Galerkin scheme uses piecewise linear basis functions, the second piecewise quadratic basis functions. The MUSCL schemes are also similar, except that the first scheme has a minmod limiter applied, for the second the superbee limiter is used.

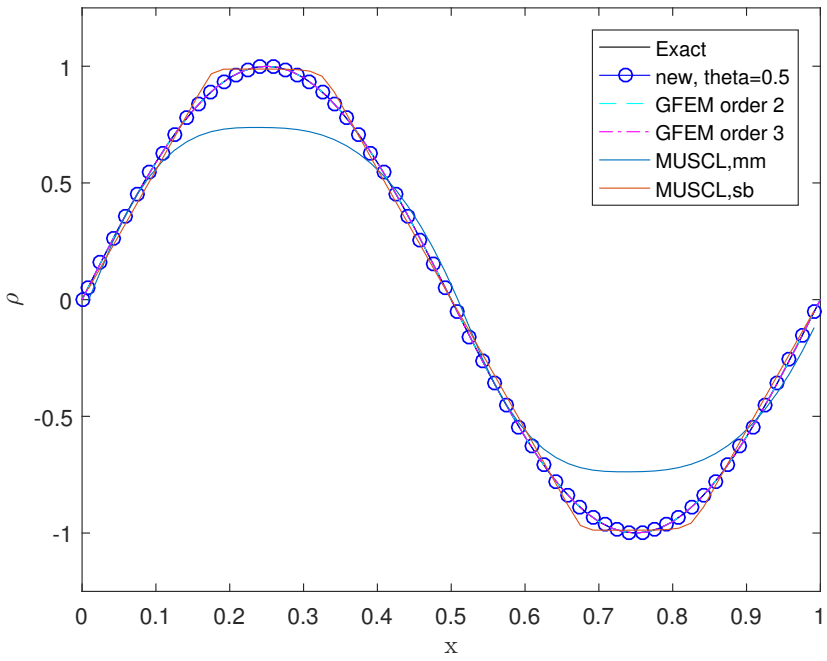
5.2. Qualitative Results

In Figures 5.1 and 5.2, the solutions after 10 full cycles through the domain are shown. For these results 60 cells are used, with a CFL of 0.1. The figures show the solution of the method as discussed in Section 4.3. The results of this scheme are plotted in blue with markers, in which the markers represent the integrated cell values. These results are compared to the exact solution, two Galerkin FE and 2 FV methods. The first Galerkin scheme uses piecewise linear basis functions and is plotted with the cyan dashed line. The second Galerkin scheme uses quadratic polynomials and is plotted with magenta dash-point. The FE reference schemes should be respectively second and third order accurate. The time integration scheme for the Galerkin methods and the new method is a θ scheme, with a θ of 0.5. The reference FV schemes are of the MUSCL type. Because a central scheme has stability issues and an upwind scheme is rather diffusive, limiters are applied. For the first FV scheme the minmod limiter is used, for the second scheme a superbee limiter. The first FV scheme is plotted with a light blue line, the second with a brown line.

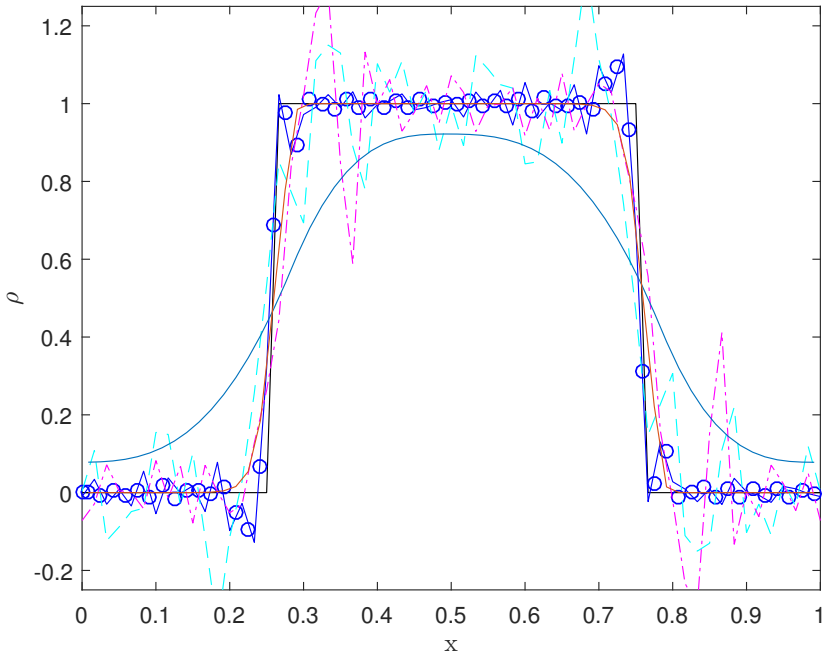
The sine wave of test case 1, shown in Figure 5.1a, is perfectly solved by the new method and both the Galerkin schemes. No differences between the exact solution and the FE methods can be seen. For the FV methods however, the limiter can be seen at work, while the problem is smooth and hence the high order flux should be always taken. Since this is not the case, a clear amount of diffusion can be seen for the minmod limiter. The superbee limiter performs much better, but close to the extrema some flattening occurs. The curvature is clearly less than it should be.

For the second problem, the unit step, oscillations are to be expected. As can be seen in Figure 5.1b this is indeed the case for both Galerkin and the new method. Comparing the new method with the Galerkin methods however, oscillations are a lot less severe. Only small wobbles can be seen in the whole solution and only close to the discontinuities some sharp edges can be seen. The shock itself is resolved in just a few cells. At 3 cells before and after the discontinuity, the variation of the integrated values is approximately 1% of the exact solution. The Galerkin schemes exhibit much sharper and stronger oscillations, while having a wider shock representation. Comparing to both the MUSCL schemes, again the minmod is to diffusive. The superbee limiter smooths out all oscillation, while displaying a strong shock. In approximately six cells the superbee limited scheme has resolved the discontinuity without oscillations, the new method does this in 3 or 4 cells while having some overshoot and wobbles.

For the third problem, see Figure 5.2a, performances are similar. The discontinuity is resolved correctly by the new method. Again a small overshoot after the discontinuity, but this is again smoothed out rapidly and the linear part is resolved correctly with only small wobbles. In comparison, the Galerkin methods require more space to resolve the discontinuity a display strong and sharp oscillations. The minmod limited MUSCL scheme is again much too dissipative. The superbee limited MUSCL scheme has the shock on the correct position and relatively sharp. The peak however is not reached correctly and flat areas are found in the region just before and after the discontinuity.

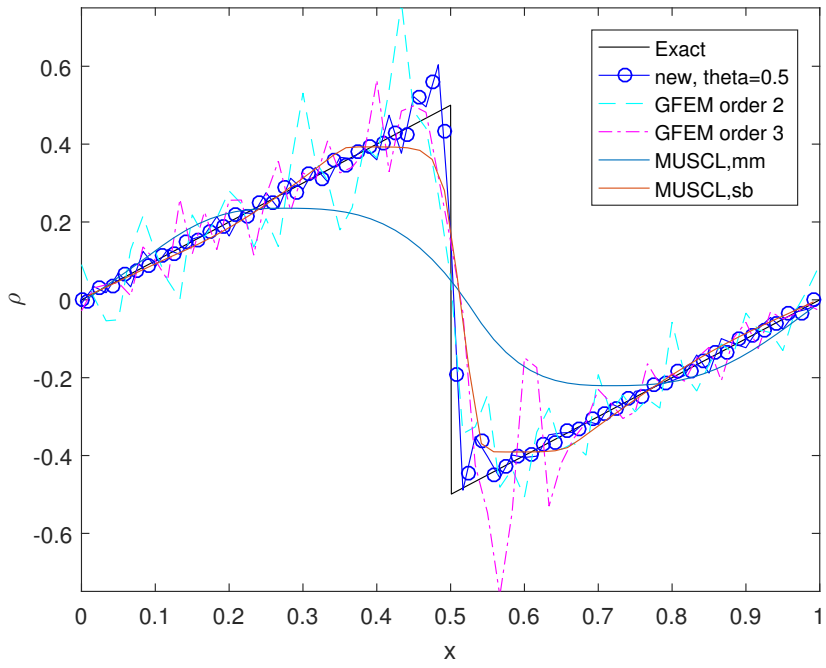


(a) Case 1

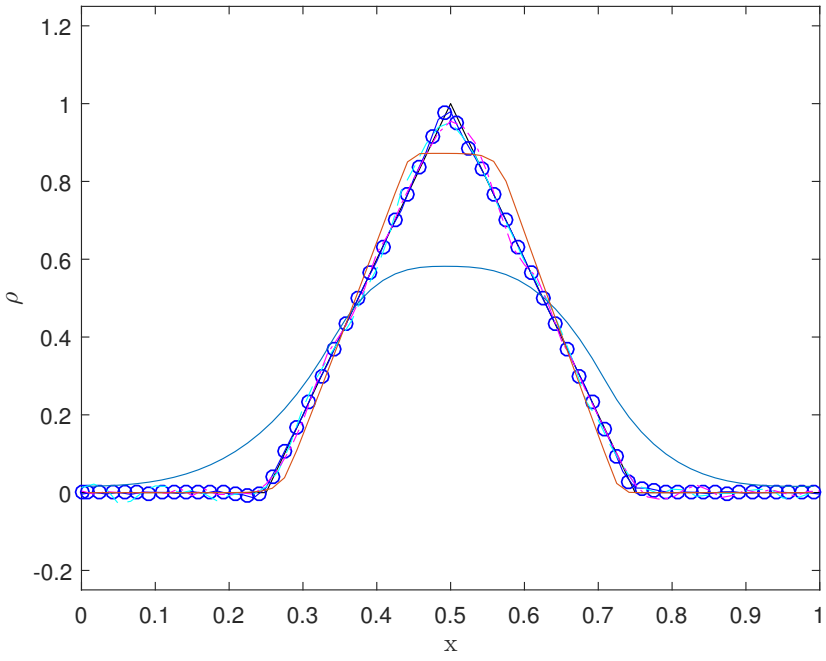


(b) Case 2

Figure 5.1: Results for linear advection test cases, using 60 cells, CFL of 0.1, with $a = 1$ after 10 cycles through the domain.



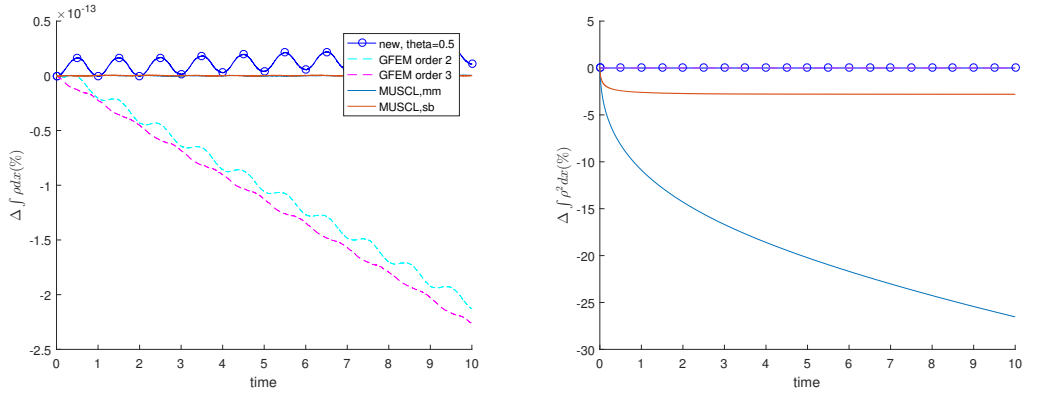
(a) Case 3



(b) Case 4

Figure 5.2: Results for linear advection test cases, using 60 cells, CFL of 0.1, with $a = 1$ after 10 cycles through the domain.

The last problem can be seen in Figure 5.2b. For the hat function, the results for the limited MUSCL schemes are not great. Both do not reach the peak and the minmod limiter smears out over a much wider domain. The FE schemes however, match with the exact solution fairly well. Both Galerkin schemes closely match the corners. The maximum value is approximately 5% below the top for both the second and the third order Galerkin. At any node, the Galerkin schemes are up to 2% off with the exact solution. The new method has flux values only 2% of near the top, but the integrated value is only 1% off. Since the actual polynomials cannot represent such a sharp edge, the value just before the peak is slightly below exact, while after the peak it is slightly above exact. At the lower left corner the integrated value has a slight undershoot. Within three cells away from each corner, the integrated values lie on the exact solution.



(a) $\int_\Omega \rho d\Omega$ (b) $\int_\Omega \rho^2 d\Omega$, as percentage of initial conditions

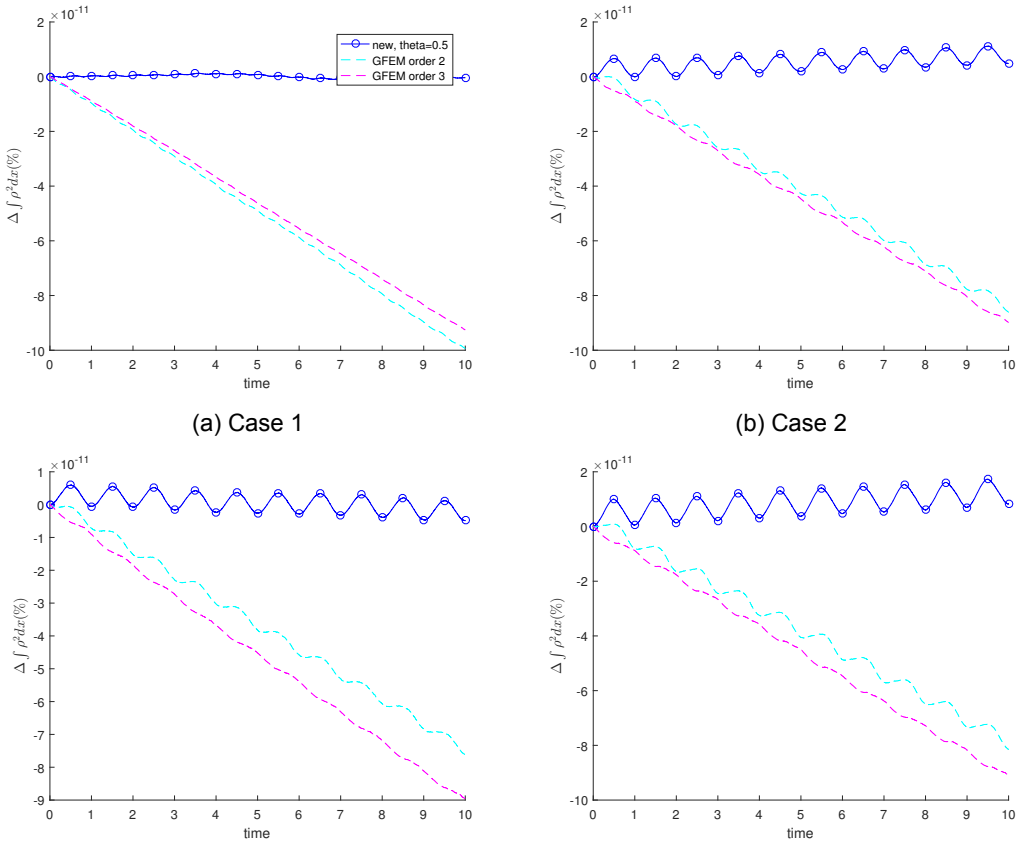
Figure 5.3: Variation of conservation properties over time, for test case 2

5.2.1. Conservative Properties

When we look at the conservation properties of this new approach, it was found that the change in mass, $\Delta \int_\Omega \rho d\Omega$, is close to zero. The results over time for test case 2 are plotted in figure 5.3a. As can be seen from this figure, the maximum variation is in the order of $\mathcal{O}\left(10^{-13}\right)$. This is in the same range for all the reference methods, although the MUSCL schemes, being a FV method, show even less variation. One can see how the explicitly defined conservation of FV methods reduces the error, since the MUSCL schemes have a steady value of 0. The variation of $\Delta \int_\Omega \rho d\Omega$ for both the proposed method and the FE methods show an oscillation, which exactly coincides with the wave passing through the domain. However, where the new method consistently returns back to near 0, the Galerkin schemes lose mass. For all other test cases, the results with respect to $\Delta \int_\Omega \rho d\Omega$ are near identical.
Diffusion can be measured as a change in $\Delta \int_\Omega \rho^2 d\Omega$. In Figure 5.3b this change in $\Delta \int_\Omega \rho^2 d\Omega$ as percentage of initial $\int_\Omega \rho^2 d\Omega$ over time is shown for test case 2. The settings are again 60 cells, a CFL of 0.1 and running time is 10 cycles through the domain. As can be seen from the figure, both the MUSCL schemes show significant diffusion, although the superbee limiter performs significantly better in comparison to the minmod limiter. The minmod limiter reaches a difference of over 25% after 10 cycles, whereas the superbee limiter keeps the maximum decrease at approximately 2.8% of the initial $\int_\Omega \rho^2 d\Omega$. These reduction are a confirmation of the results discussed before. In the Figures 5.4, a close up of $\Delta \int_\Omega \rho^2 d\Omega$ over time is shown for all test cases. In this it can be seen that the FE results and the new method have a change in $\Delta \int_\Omega \rho^2 d\Omega$ in the order of $\mathcal{O}\left(10^{-11}\right)$%. Comparing the result for test case 2, see Figure 5.4b, with the change of mass, see Figure 5.3a, a direct connection between the variation of $\Delta \int_\Omega \rho^2 d\Omega$ and $\Delta \int_\Omega \rho d\Omega$ can be found. Again, this variation is within round-off error domain.

5.2.2. Error Analysis

In Figures 5.5 and 5.6, the L_2 error norms, with respect to the exact solution, are shown. In these figures, N is the number of cells. In Table 5.1, the slopes associated with the error norms are listed. In this table, the slopes for L_2 with respect to the exact solutions, as well as the L_1 norm with respect to



(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 5.4: Close up of the variation in $\int_\Omega \rho^2 d\Omega$ over time per test case

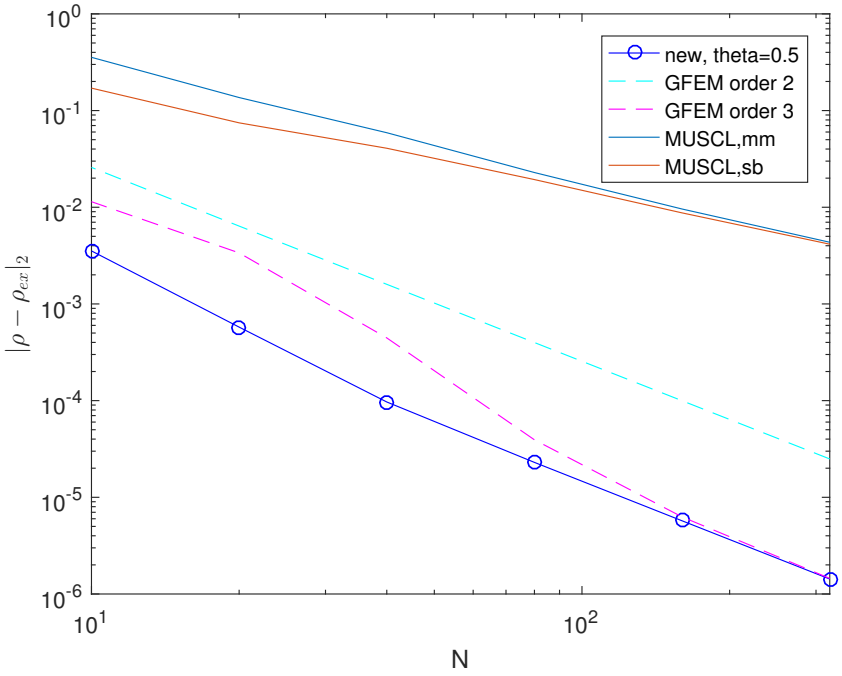
the exact solution and the L_2 norm with respect to a solution on a fine mesh are shown. A grid with 320 cells is selected as the fine mesh. The L_2 norms shows with which rate the method is converging, the L_2 with respect to a fine grid is to validate the convergence rates found. The norms are defined as:

$$|\rho - \rho_{ex}|_n = \left(\int_\Omega |\rho(x) - \rho_{ex}(x)|^n \right)^{\frac{1}{n}}, \tag{5.4}$$

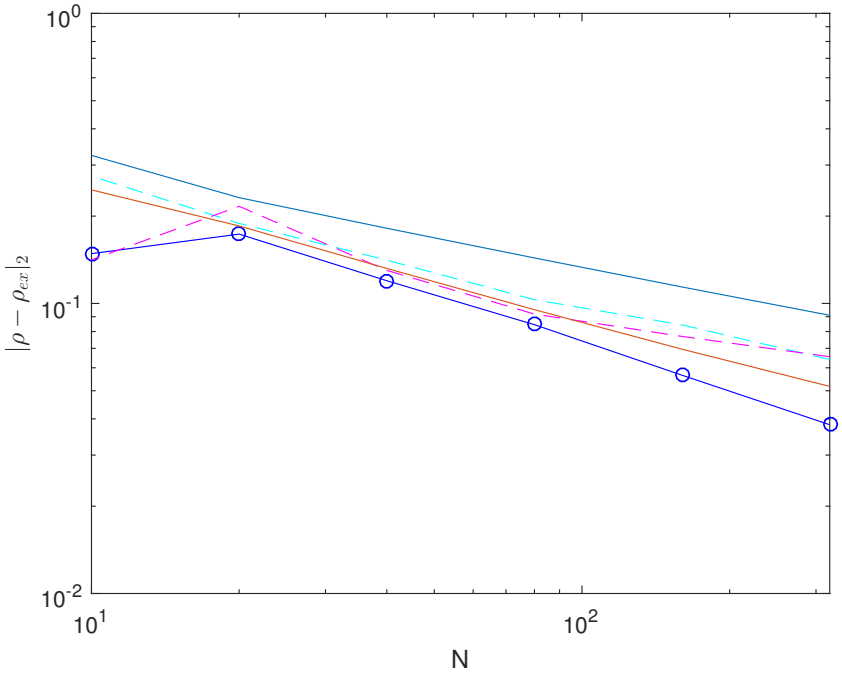
in which subscript n denotes the L_n norm. The norms are integrated with 1000 evenly spaced points on the reconstructed solution, hence $\rho(x) = \sum \rho_i e_i(x)$. This integration is performed using a trapezoidal numerical integration method. In each of the figures, the new scheme is consistently the one with the lowest error. For the completely smooth problems of case 1, the same error is found as the Galerkin schemes. For the other test cases, the L_2 errors are lower and convergence rates are higher. In Table 5.1, the average convergence rates are listed. From this table, it is found that the new method in almost every case has one of the highest error convergence, both for the convergence to the exact solution as to the convergence to the fine mesh solution.
The expected convergence rate should be $\min(s, p+1)$, with s the regularity of the problem and p the polynomial degree. The regularity for the first test case is ∞, for the second and third $\frac{1}{2}$ and for the fourth case $\frac{3}{2}$.
From the L_2 convergence rates, the new method is extremely close to expected convergence rate in the second and third test case, with a convergence rate surpassing the expected $\frac{1}{2}$. The convergence rate to the fine mesh solution is slightly higher, probably due to smoothing of the solution. For the first test case, the new polynomials have a convergence closer to 2, while 3 should be expected with the polynomial degree. The Galerkin scheme with piecewise linear polynomials has exactly the expected rate of 2 and the scheme with piecewise quadratic polynomials is close to 3, but not exactly. Both the MUSCL schemes have a rate slightly over 1. MUSCL schemes do have a first order polynomial approximation, but the limiter does not allow the convergence rate to be the expected value of 2. The convergence rate to the fine mesh solution occurs with a convergence rate one would expect. For the new method, a rate close to 3 should be expected, but this is not reached. The FE2 reaches an order of 2 and FE3 reaches an order of 3 on the convergence to the fine grid solution. For test case 4, a convergence rate of $\frac{3}{2}$ is expected, but the actual rate found is lower.

Table 5.1: Average error convergence rates after $t = 1$ with $CFL = 0.1$.

Scheme	case	$\lvert\rho - \rho_{ex}\rvert$	$\lvert\rho - \rho_{ex}\rvert_2$	$\lvert\rho - \rho_{320}\rvert_2$
new, $\theta = .5$	1	-2.0944	-2.1640	-2.3571
	2	-0.7052	-0.5458	-0.7601
	3	-0.6876	-0.6333	-0.7562
	4	-1.3073	-1.1030	-1.2410
FEM2	1	-2.0015	-2.0003	-2.1123
	2	-0.5080	-0.3897	-0.4189
	3	-0.5128	-0.4102	-0.4463
	4	-1.3692	-1.2307	-1.2577
FEM3	1	-2.7509	-2.7957	-3.1316
	2	-0.5892	-0.4307	-0.5294
	3	-0.5091	-0.3792	-0.4598
	4	-1.3008	-1.1087	-1.2002
MUSCL, mm	1	-1.2559	-1.2430	-1.3623
	2	-0.6612	-0.3368	-0.8160
	3	-0.6874	-0.3484	-0.8210
	4	-1.1008	-0.9693	-1.1485
MUSCL, sb	1	-1.0552	-1.0423	-1.1096
	2	-0.9271	-0.4594	-0.7744
	3	-0.9693	-0.4799	-0.7926
	4	-1.0847	-1.0526	-1.1212

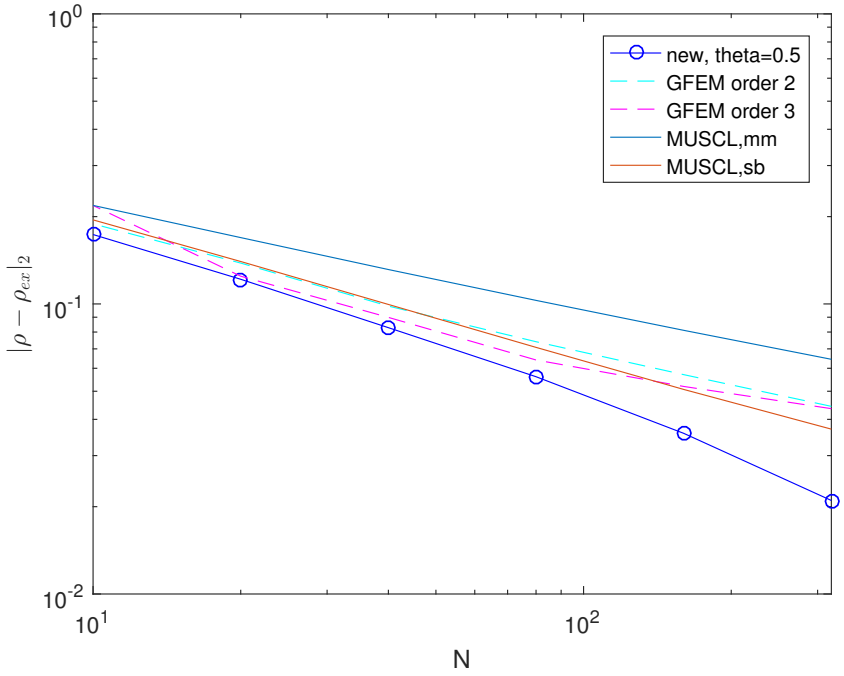


(a) Case 1

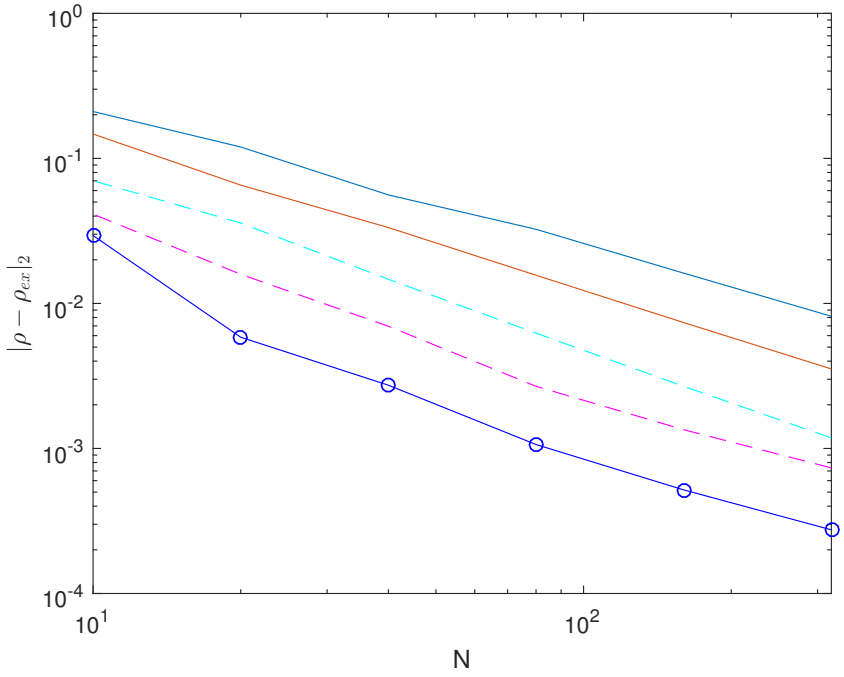


(b) Case 2

Figure 5.5: L_2-error norms, measure after 1 cycle through the domain with a CFL of 0.1



(a) Case 3

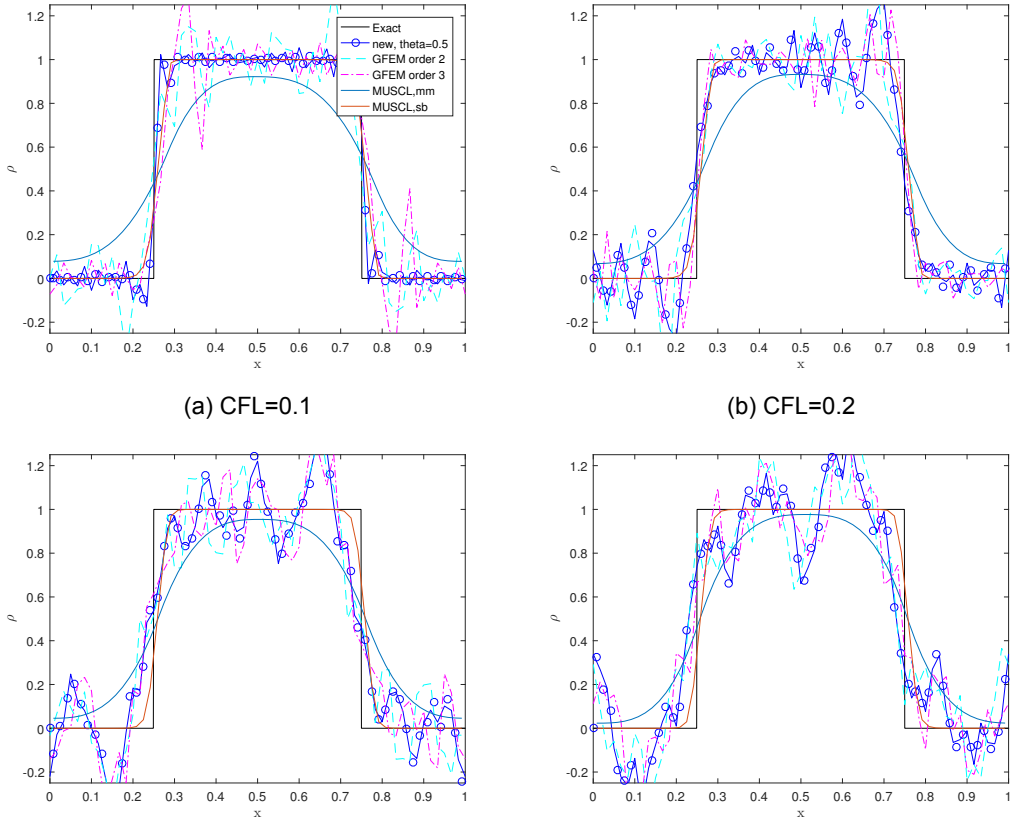


(b) Case 4

Figure 5.6: L_2-error norms, measure after 1 cycle through the domain with a CFL of 0.1

5.2.3. Time Step Dependency

Up to now, all computations shown have been done with a CFL of 0.1, which is rather low. In Figure 5.7, several plots with increasing CFL numbers are shown. For these plots, the other settings are kept constant, hence 60 cells and 10 cycles through the domain. The time integrator is just as before a Crank-Nicolson scheme. Figure 5.7a is the exactly the same figure as 5.4b. However, as soon as the CFL is increased, see Figures 5.7b to 5.7d, oscillations start to occur in the solution of the new method. With CFL of 0.2, the oscillations have a constant period and are large just behind the jump and nearly damped out just before the next jump. At CFL of 0.4, the period seem to be doubled, but the solution is also more random. With a CFL of 0.6, the solution is completely useless, one cannot determine any jump, nor what the equilibrium between jumps should be.
The Galerkin schemes have the same issues, although these scheme have oscillations at a CFL of 0.1 as well. The MUSCL schemes are only mildly affected by CFL, the minmod limited scheme reaches slightly better values between both jumps. In the next Section, a RK4 scheme is used, which is better able to handle larger time steps. Results for the RK time integrator are shown in that section.



(a) CFL=0.1 (b) CFL=0.2

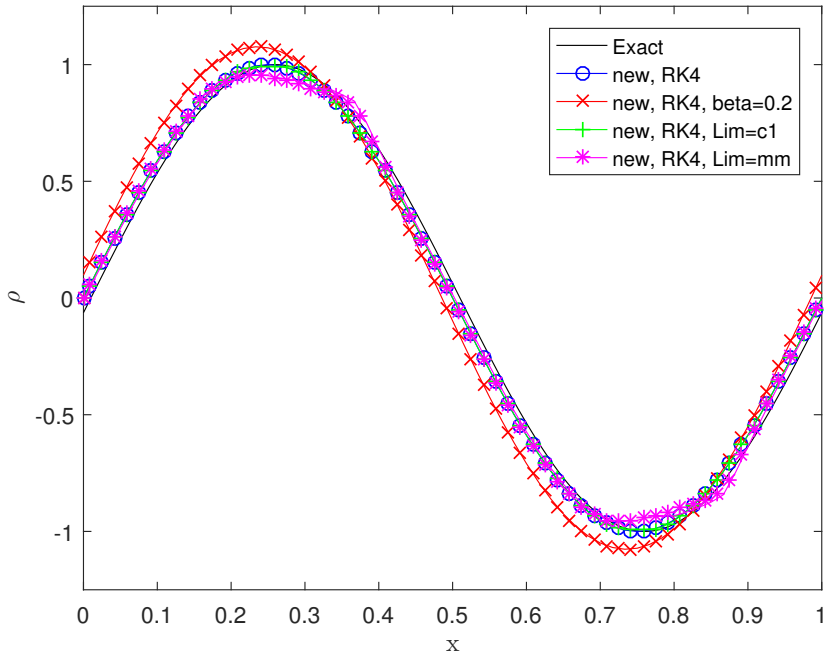
(c) CFL=0.4 (d) CFL=0.6

Figure 5.7: Results for test case 2, with increasing CFL.

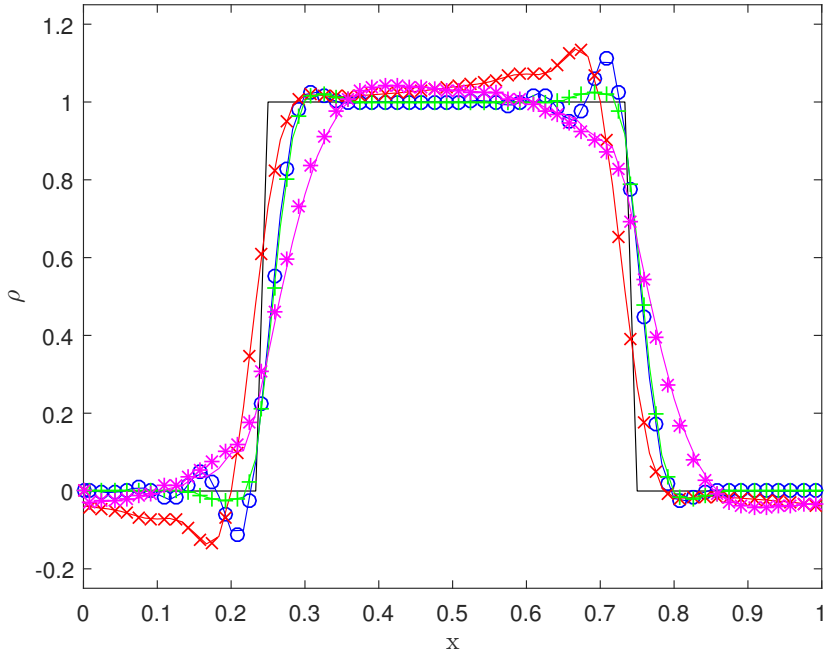
5.3. Stabilization Techniques

It was found in 5.2.3 that for cases with a discontinuity, the solution starts to oscillate when CFL numbers are higher than 0.1. This behaviour is just like any other higher order scheme without limiting, upwind or other stabilization techniques. Using a higher order time integrator such as the Runge-Kutta 4 scheme helps to reduce oscillations. In Section 4.4, a few options to enhance stability into the scheme were discussed. In this section these options are tested for linear advection. The test case 1 to 4 are tested again. For the comparison the normal scheme is used. The basic scheme is again plotted in blue with circles for the integrated values. This is compared to the stabilized methods. In the plots, the red line with crosses for the integrated values, is the scheme with bubble functions added. Only the bubble

function case of (4.38) is considered at this point. Also two correction methods are used. In green with plus signs for integrated values is the corrector 1, in magenta and stars, the minmod limiter. The implementation of the minmod limiter is according to equations (4.39) and (4.41) and is labelled mm in the figures. The corrector 1 scheme is implemented according to equations (4.44) and (4.46) and is labelled with c1 in the figures.
In the plots of Figures 5.8 and 5.9, the results with each of the techniques are shown. From the sine wave, shown in 5.8a, it can be seen that the upwind method and the minmod limiter, labelled with mm, have results differing from the exact solution. The minmod limiter, just like it does in the MUSCL scheme, flattens the top of the sine wave. However, in the MUSCL scheme the minmod limiter shaves of the results equally before and after the extrema. In the new method, more correction is performed at the front of each extrema. In comparison, the diffusion of the minmod limiter is smaller with the new scheme. When upwind is used, the solution lags a small bit behind the exact solution. it seems as if the tops of sine wave have grown. It was found that the solution is rather sensitive to the upwind factor β. In the test performed in this section, β was taken as 0.2, when this factor is increased further, the scheme changes such that it might become unstable. This sounds rather strange, but later on, when non-linear advection is treated, a small β value does stabilize. For the sine wave corrector 1 yields the same results as the uncorrected scheme. This is logical, as no discontinuities are present, hence a smooth solution is found which is advected as expected.
For the block function, the differences of each method becomes more apparent. Also the effect of larger time steps become clearly visible, as the slope which solves the discontinuity is less sharp. Where the results found in the previous section required 3 or 4 cells to resolve the jump, this is now doubled. It was found that this is not due to time integration, but the RK4 time integrator in general performed better. The results are plotted in Figure 5.8b. The new method without additional stabilization has a small undershoot before the discontinuity, a significant overshoot directly after the jump and several oscillations directly afterwards. After approximately 2 waves, which take up to 12 cells in length, the solution tends to the exact height of the unit step. When bubble function upwind is used on the cell faces, all oscillations are removed from the solution, but the overshoots are definitely not. Directly after the jump, an overshoot of approximately 13% of the step height occurs. In this case β is already to high, since the solution is overdamped. The solution after the discontinuity does not reach the exact solution within the step width. At the point the second discontinuity is reached, the solution is still 2% off. The minmod limiter adds significantly more diffusion, but makes sure no oscillations occur and no overshoots are present. This however, comes at a cost of a high artificial diffusion. The added diffusion is of such a scale that the corners of the unit input are completely rounded off. Corrector 1 keeps the solution close to the solution found without corrections, but the overshoot is reduced significantly from approximately 12% to only 2% of the step height. The visible oscillation are removed in approximately 6 cells, which is halve of the cells required with the uncorrected method.
With the third test case, results are very similar to the second test case, as can be seen in Figure 5.9a. Again the uncorrected, non-upwind scheme shows some oscillations near the discontinuity, which damp out after approximately 12 cells after the jump. The overshoot is again in the order of 12% of the jump height. Just before the discontinuity a small dip is shown. With upwind applied, one big overshoot raises the whole solution after the discontinuity. The overshoot itself is slightly larger than the non corrected scheme and is further upstream. With upwind the jump itself is also slightly moved a few cells upstream. It takes nearly half the domain length for the solution to get back to the exact solution, which increases the slope significantly. For the minmod limiter again a large amount of diffusion is added. Due to this, the dip before and the overshoot after the jump move further away. Corrector 1 greatly reduces the overshoot and effectively reduce oscillation length by half. The overshoot is reduced from 12% of the jump height to 2%.
The fourth case, see Figure 5.9b, the minmod limiter and the upwind scheme behave similar to case 1. The upwind scheme slightly trail behind and the maxima is slightly higher. As a result a dip behind the hat function is formed. The minmod limiter flattens the sharp point which effectively reduces the height of the solution with 18%. The missing mass is diffused, which results in a lump in front of the hat function and rounded edged on the base of the hat function. The uncorrected method performs just like it did with low CFL, the top is only of by less than 5%, while the integrated value is off by even less. The base corners are rounded a little bit, but these are resolved in 2 cells.

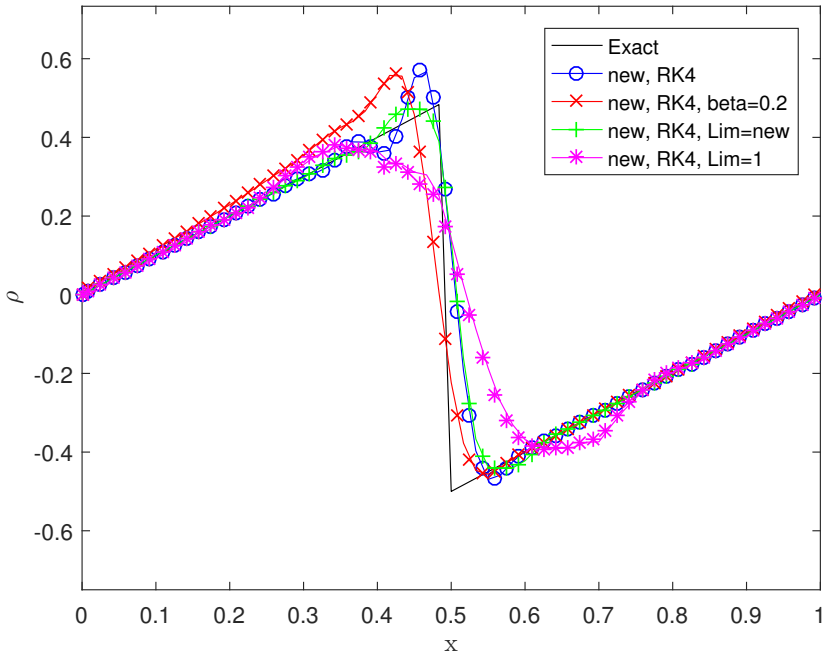


(a) Case 1

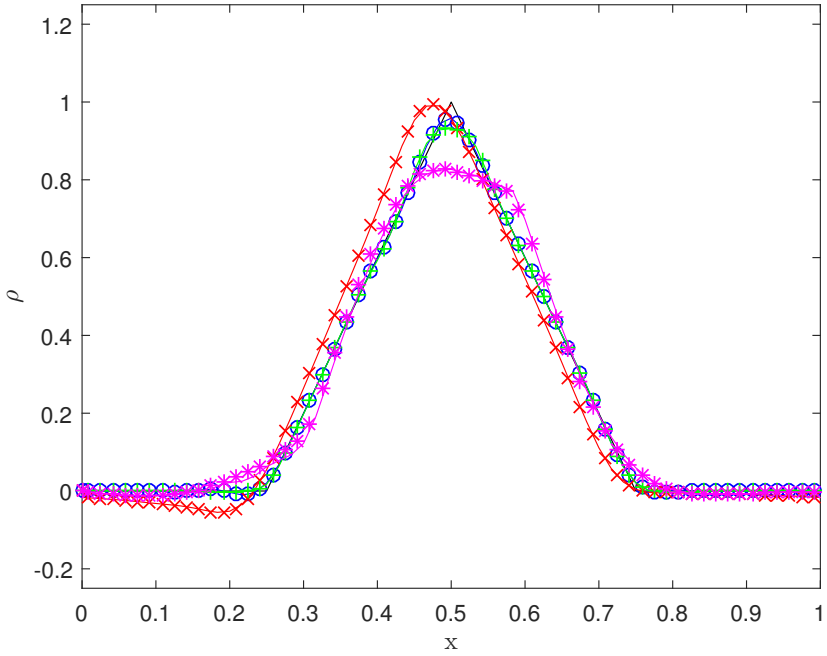


(b) Case 2

Figure 5.8: Results for linear advection test cases, using 60 cells, CFL of 0.6, with $a = 1$ after 10 cycles through the domain, with stabilization techniques.

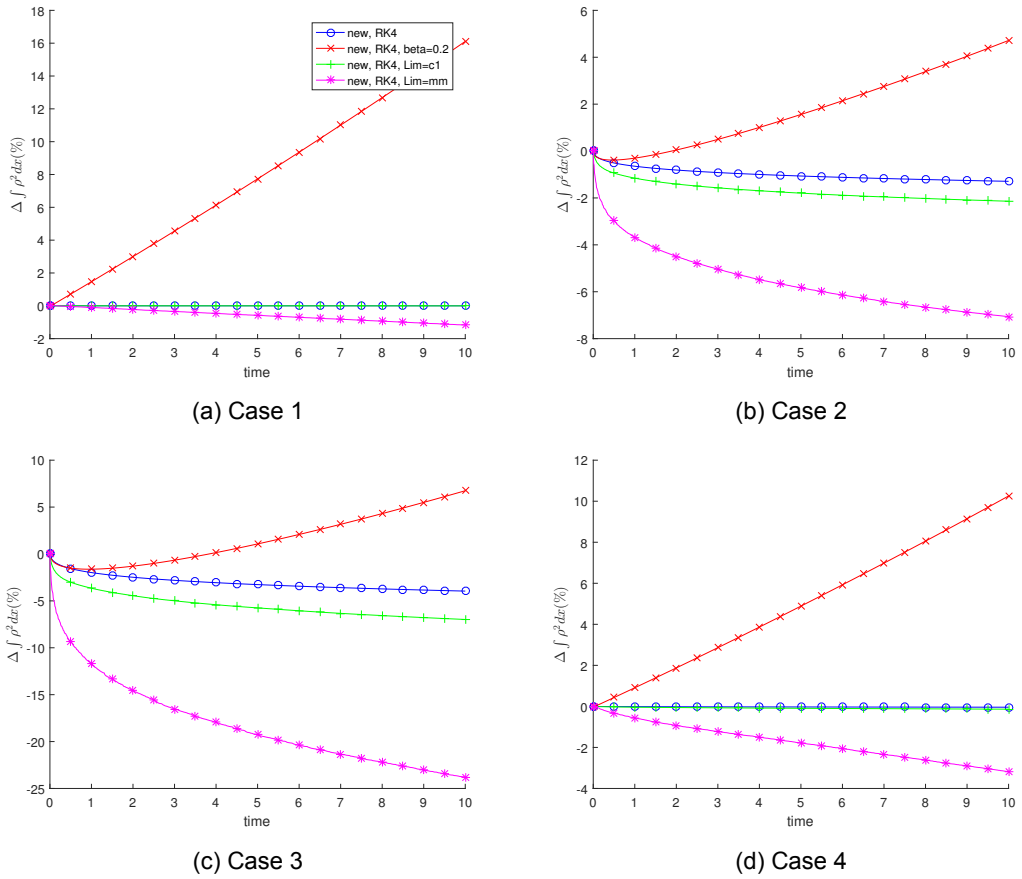


(a) Case 3



(b) Case 4

Figure 5.9: Results for linear advection test cases, using 60 cells, CFL of 0.6, with $a = 1$ after 10 cycles through the domain, with stabilization techniques.



(a) Case 1 (b) Case 2

(c) Case 3 (d) Case 4

Figure 5.10: Variation of $\int_\Omega \rho^2 d\Omega$ over time for linear advection, with stabilization techniques.

5.3.1. Conservative Properties

Especially important for this test, is the amount of numerical diffusion each technique adds. Usually limiting techniques add diffusion, which should be kept at a minimum. Again the change of $\Delta \int_\Omega \rho^2 d\Omega$ over time is investigated. The percentage change with respect respect to the initial conditions represented by the basis functions is taken as measurement for diffusion. It has to be noted that percentage change cannot be compared directly between each of the test cases, as the initial integral $\int_\Omega \rho^2 d\Omega$ differs per case. The percentage do say something about the numerical diffusion. The plots of the results can be found in the Figure 5.10. This time, the base scheme does have some change in this integrated quantity, which is entirely due to the different time integrator. Still, the choice of a Runge-Kutta time integrator is justified as the θ scheme no longer yields results at the time step taken for this test. From each of the figures, it is immediately clear that without correction or stabilization the least amount of diffusion is present. Depending on initial conditions, the error on $\int_\Omega \rho^2 d\Omega$ is in the order of $\mathcal{O}\left(10^{-5}\right)$% for the sine wave and -1.2% for the case 2 and -3.9% for case 3.

Corrector 1 also performs really well. In comparison to the uncorrected method, the diffusion error is one or two orders higher, $\mathcal{O}\left(10^{-3}\right)$% for the sin wave and -2.1% for the unit step function. For test case 3, the change is -6.9%. Given the accuracy and improved damping of unwanted oscillations found before, this is a satisfactory result.

The upwind scheme in has the tendency to increase the $\int_\Omega \rho^2 d\Omega$ over time, which eventually will result in a divergence of the solution. This is opposite of the goal in mind when using upwind. Depending on the case, the upwind method has an increase between 5% and 16% after 10 seconds of running. In the cases with discontinuity, initially a small decrease was found. The lowest points are in the approximately -0.3% and -1.6% for test cases 2 and 3 respectively. The change of sign happens after respectively 2 and 4 passes through the domain.

Overly stable and diffusive, is the minmod limiter. As can be seen from Figure 5.10b and 5.10c, $\int_\Omega \rho^2 d\Omega$ decreases rapidly after simulation starts. For the smooth problems, this decrease is more linear, but

again much stronger compared to the uncorrected scheme and the corrector 1 scheme. The decrease ranges from -1.1% for the sine wave, to nearly -25% for test case three.

5.3.2. Error Analysis

In Figure 5.11, the L_2 error norm is shown with stabilization options. The same settings are used as with the other computations in this section, thus a CFL number of 0.6 and a Runge-Kutta 4 time integrator, but an end time $t = 1$. As can be seen, the corrector reduces the error slightly. The minmod limiter increases the error significantly, which is understandable as the limiter adds a significant amount of diffusion. Using upwind only increases error when a finer grid is used, which is kind of counter-intuitive. The amount of diffusion does not decreases accordingly with the upwind approach. Also, the error of the upwind scheme seems to be lower compared to the base RK4 scheme. This is a contradiction to the results found in Figure 5.8b. The reason for this can be found in Figure 5.10b, in the first two cycles the upwind has a negative $\Delta \int_\Omega \rho^2 d\Omega$, while only after that the solution starts to diverge from the exact solution.
In Table 5.2, the error convergence found for each case is tabulated. Corrector 1, yields almost the same error convergence as the upwind stabilized scheme. The minmod limiter and the upwind scheme show a reduction in convergence rates, just as seen in Figure 5.11. The upwind scheme has the strongest reduction in convergence rates.

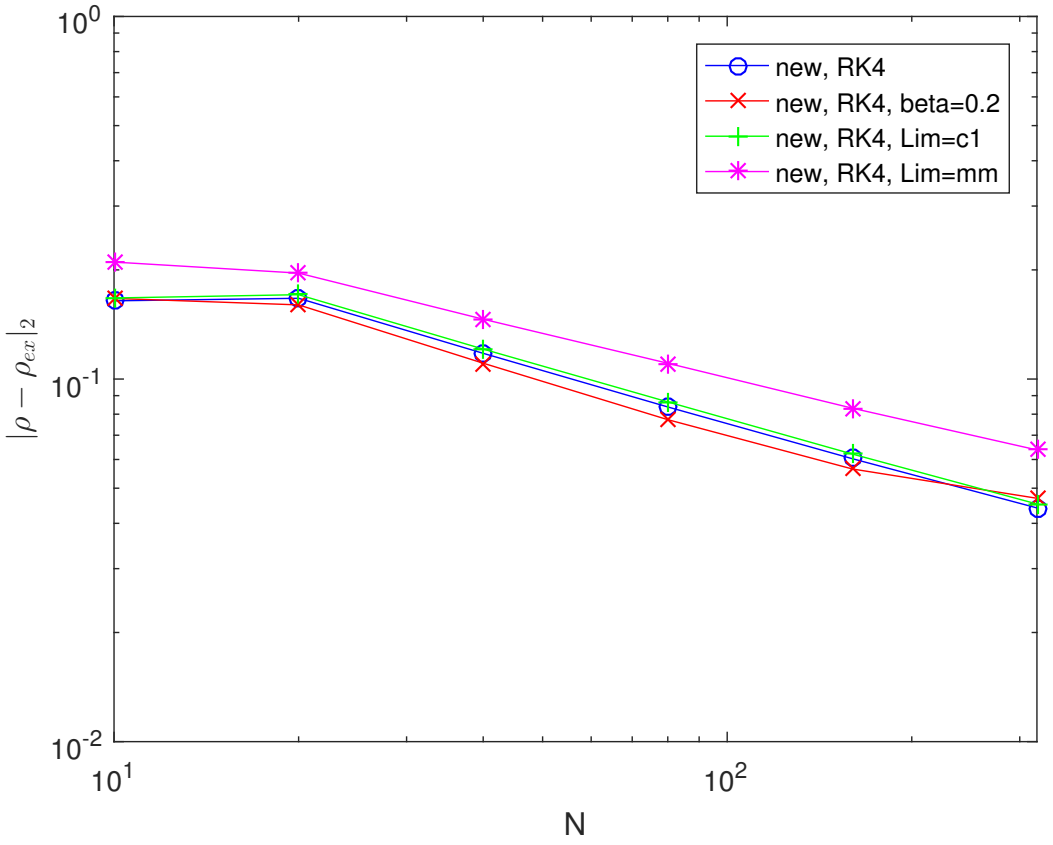


Figure 5.11: L_2 error convergence with limiters applied, for case 2, $t = 1$ and $CFL = 0.6$

5.4. Conclusions and Discussion

From the results for linear advection, the general conclusion is that satisfactory results are obtained. The basic scheme is able to successfully solve each of the test cases. In comparison with standard Galerkin FE using piecewise polynomials and the MUSCL scheme, performance was the same or better. Error rates are low and up to second order error convergence was found. Also, the error is

Table 5.2: Average error convergence rates after $t = 1$ with $CFL = .6$, for stabilized schemes

Scheme	case	$\|\rho - \rho_{ex}\|$	$\|\rho - \rho_{ex}\|_2$	$\|\rho - \rho_{320}\|_2$
New, RK4	1	-3.4338	-3.4526	-3.5753
	2	-0.8359	-0.4801	-0.7731
	3	-0.8497	-0.5043	-0.7732
	4	-1.5249	-1.1720	-1.2791
New, RK4, $\beta = .2$	1	-0.5822	-0.5827	-0.7869
	2	-0.6807	-0.4429	-0.8408
	3	-0.6828	-0.4260	-0.8428
	4	-0.6944	-0.5455	-0.9493
New, RK4, c1	1	-2.2601	-1.9124	-1.9902
	2	-0.8703	-0.4794	-0.8466
	3	-0.8985	-0.5041	-0.8374
	4	-1.5443	-1.1581	-1.2694
New, RK4, mm	1	-1.3049	-1.3048	-1.3675
	2	-0.7701	-0.4050	-0.7723
	3	-0.8041	-0.4395	-0.7926
	4	-1.2753	-1.0867	-1.1982

consistently lower than the reference methods. Even though the scheme is higher order, the oscillations near a discontinuity disappear rapidly. Also artificial diffusion was found to be low.
It was noted that the scheme is rather dependable on CFL number, especially with a Crank-Nicolson time integrator. With Crank-Nicolson, results become noticeable oscillatory when the CFL number is taken higher than 0.1. Using a Runge Kutta time integrator allows a CFL number up to 0.6. Also several methods to decrease oscillations and increase stability were investigated. Upwind was against expectations the least stable for linear advection. Using a limiter was more successful in that respect, but comes at the price of high diffusion rates. Corrector 1 performed excellent in this respect. At a minimal cost on diffusion, the oscillations a reduced effectively. Overshoots reduced significantly by using this approach on correction.
The fact that CFL numbers could not go up to one without stabilization was unexpected. A possible reason is that the scheme can be seen as a three point Finite Element scheme, thus effective cell size is smaller. The Runge Kutta time integrator clearly helped, as this is a multi step method with effectively smaller time step in each stage. Also, the Runge Kutta scheme results in a quadratic reconstruction of the flux, rather a linear construction when a θ scheme is used. This further increased stability.

6

Non-linear advection: Inviscid Burgers' Equation

The Burgers' equation, named after Johannes Martinus Burgers, is a useful PDE to get an understanding in physical as well as mathematical problems. For instance, the complete Burgers' equation can be used in modelling one dimensional turbulence, even though turbulence is inherently a multi-dimensional phenomenon. To test advection schemes, the inviscid Burgers' equation is a perfect one dimensional test case, as it is non-linear. In the Burgers equation, particles move with a velocity equal to their own value. This non-linear behaviour usually results in particle overtaking each other. Since that is not allowed, shocks will form. The non-linearity of the Burgers equation requires proper stabilization techniques.

6.1. Problem Definition

In Burgers' equation a conserved property is advected with it's own value. The one dimensional inviscid Burgers' equation can be written in the advective form:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0, \tag{6.1}$$

or in the conservative form:

$$\frac{\partial u}{\partial t} + \frac{1}{2}\frac{\partial uu}{\partial x} = 0. \tag{6.2}$$

In Lie derivative form this is written as:

$$\begin{aligned} \frac{\partial \rho^{(1)}}{\partial t} + \frac{1}{2}\mathcal{L}_a \rho^{(1)} &= 0, \\ a(x) &= \sharp\rho^{(1)}(x). \end{aligned} \tag{6.3}$$

The $\sharp$ operator transforms the 1-form ρ into a vector a. It can be shown that the same scheme as found in 4.3 is valid, with the only exception that a_L and a_R in (4.27) are not taken a a constant, but instead the local value of ρ. As a result, the $\mathbb{F}$ matrix becomes:

$$\mathbb{F} = \begin{bmatrix} \int_{-1}^{1} \frac{\partial a_L e_1}{\partial x} e_1 dx & \int_{-1}^{1} \frac{\partial a_I e_2}{\partial x} e_1 dx & \int_{-1}^{1} \frac{\partial a_R e_3}{\partial x} e_1 dx \\ \frac{-a_L}{\Delta x} & 0 & \frac{a_R}{\Delta x} \\ \int_{-1}^{1} \frac{\partial a_L e_1}{\partial x} e_3 dx & \int_{-1}^{1} \frac{\partial a_I e_2}{\partial x} e_3 dx & \int_{-1}^{1} \frac{\partial a_R e_3}{\partial x} e_3 dx \end{bmatrix}, \tag{6.4}$$

with $a_L = \rho_L$, $a_R = \rho_R$ and $a_I = \rho_I/\Delta x$. Since the governing equation is non-linear,
Since the scheme is non-linear, some sort of stabilization is required. Therefore two series of tests are performed. In the first series the effect of the two upwind approaches, the SUPG and bubble function approach, is investigated. The SUPG scheme is according to (4.35), with a value of T of (4.34b). The

bubble function are done according to (4.38) and (4.36).
In the second series a minmod limiter and the correctors 1 and 2 are applied to the scheme. More details on both the upwind methods as the corrector methods can be found in Section 4.4. As reference a Galerkin scheme with piecewise linear polynomials is used, with a bubble functions for upwind properties. The bubble functions for this scheme are quadratic functions, which are equal to zero at the nodes. The bubble functions have a positive value upstream, and as a result the weight functions become:

$$W_i = \phi_i - \frac{a}{|a|}\frac{3\beta}{4}\left(1-\xi^2\right), \tag{6.5a}$$

$$W_{i+1} = \phi_{i+1} + \frac{a}{|a|}\frac{3\beta}{4}\left(1-\xi^2\right). \tag{6.5b}$$

With β the upwind parameter, which is set to 0.2, and a the local flow velocity, which is equal to ρ_i. Also used as reference method, is a SUPG scheme with piecewise linear polynomials. For this follows the same method as (4.35), with a value of T of (4.34b).
As time integrator, an explicit Runge-Kutta 4 scheme is used. Runge-Kutta time schemes can be written as:

$$\rho^{n+1} = \rho^n + h\sum_{i=1}^{s} b_i k_i. \tag{6.6}$$

In this the superscript n means time level n. k_i are solutions at intermediate time levels, which are calculated according to:

$$k_i = f\left(t_n + c_i\Delta t, \rho^n + \Delta t\sum_{j+1}^{i-1} a_{i,j}k_j\right). \tag{6.7}$$

Weights $a_{i,j}$ and b_i are taken from the Butcher tableau for the explicit RK4 scheme. The initial conditions are two parabola, connected such that they look similar to a sine wave. To generate a wave running in 1 direction only, the whole equation is increased with a constant such that no negative initial conditions are present. The equation for the initial conditions are:

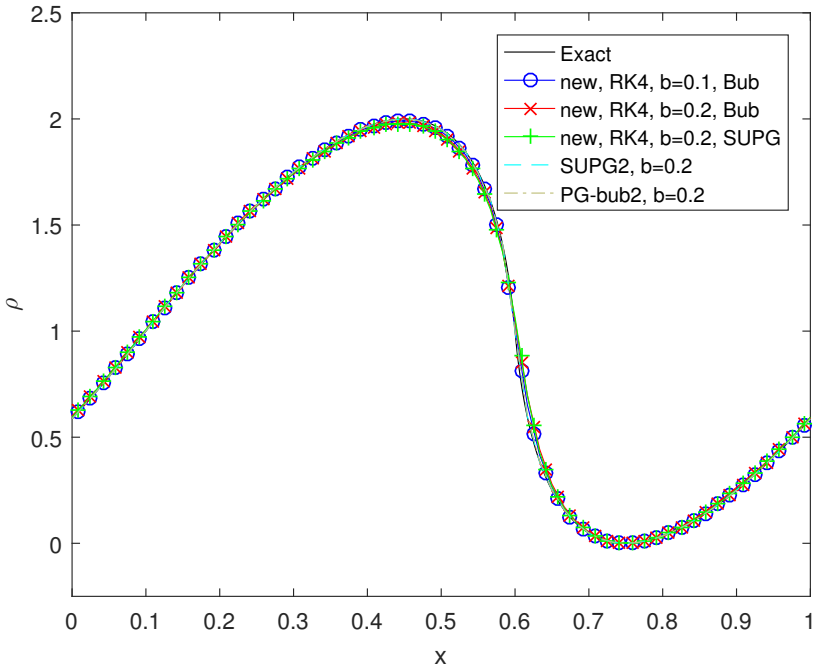
$$\begin{aligned} \rho(x,0) &= 1 - \frac{16}{L^2}\left(x-\frac{L}{2}\right)x, && \textit{if } x \le \frac{L}{2}, \\ \rho(x,0) &= 1 + \frac{16}{L^2}\left(x-\frac{L}{2}\right)(x-1), && \textit{else}. \end{aligned} \tag{6.8}$$

6.2. Results with Upwind

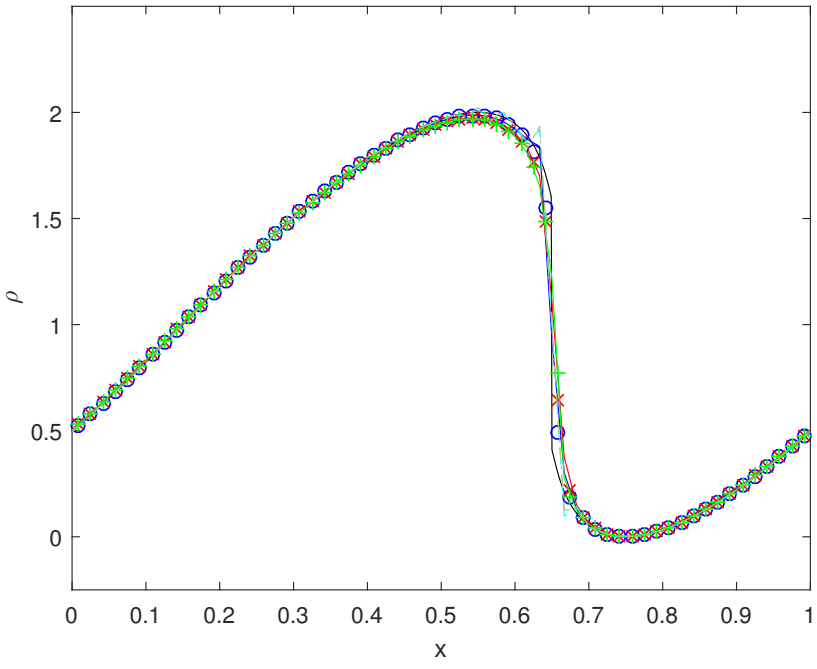
First the results of the upwind methods will be discussed. The upwind methods tested are the bubble functions added to the polynomials associated with the cell faces and the SUPG method applied on all polynomials. In this section, the reference methods are treated as well.

6.2.1. Qualitative Results

In the plots of Figures 6.1 and 6.2, the solutions with upwind applied, found after a time of 0.1, 0.15, 0.2 and 1 second are shown. In the plots, the proposed polynomials are plotted markers, in which the markers indicate the integrated value of a cell. The blue line with circles has bubble functions with β of 0.1, the red with crosses has bubble functions with β of 0.2. The green line with plus signs is SUPG, again, with β of 0.2. The dashed cyan line corresponds to the SUPG scheme with piecewise linear polynomials. The darker green has piecewise linear polynomials as well, but with bubble functions.
Of the four plots, the first of is just before a discontinuity forms, the second when the discontinuity is forming, the third plot is while the shock is growing in strength, and the last plot is after the wave moved through the domain exactly 1 time. For these results a grid with 60 cells was used and a Δt of $8.333 \cdot 10^{-4}$. With this time step the initial CFL number is equal to 0.1. It has to be noted, results up to a start CFL of 1 are stable when RK4 time integration is used, even without upwind or limiter. The

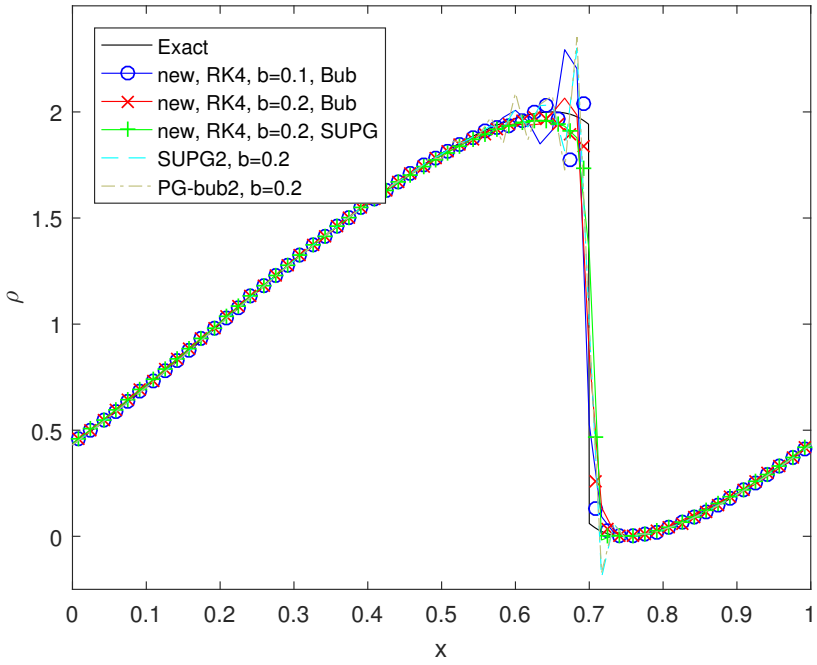


(a) $t = 0.1$

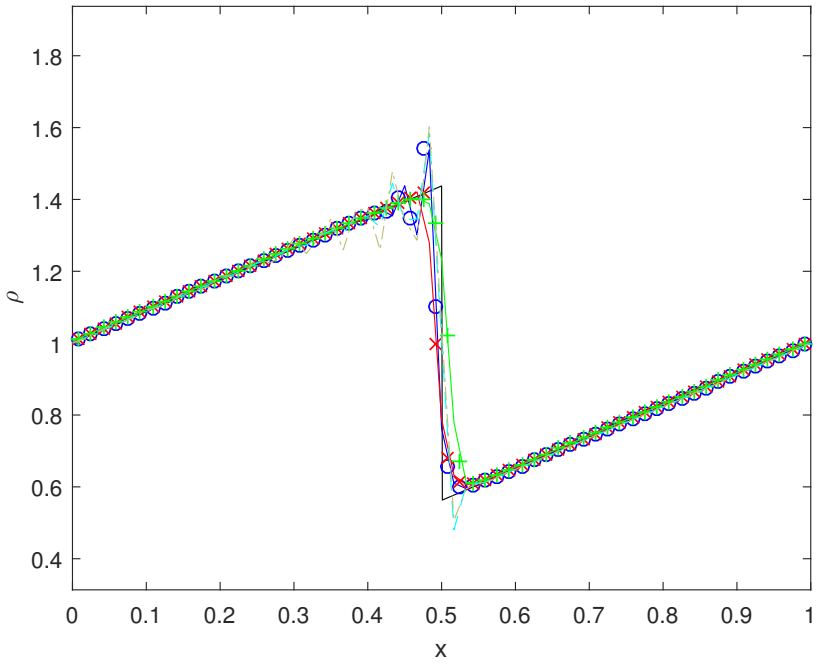


(b) $t = 0.15$

Figure 6.1: Results for non-linear advection with 2 parabola. With 60 cells and $\Delta t = 8.33 \cdot 10^{-4}$ and bubble function upwind applied on fluxes



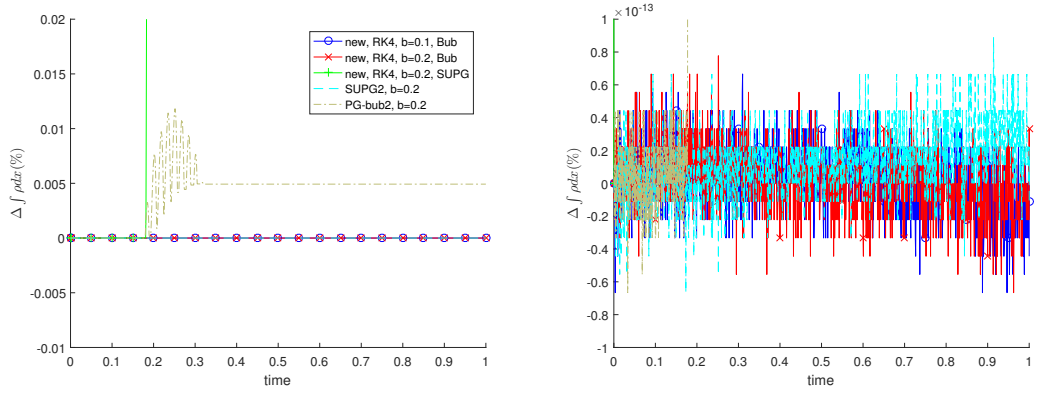
(a) $t = 0.2$



(b) $t = 1$

Figure 6.2: Results for non-linear advection with 2 parabola. With 60 cells and $\Delta t = 8.33 \cdot 10^{-4}$ and bubble function upwind applied on fluxes

results however, are oscillating wildly and are not usable.
In the first Figure 6.1a, the whole solution correctly moved with its own velocity. The integrated values are also close to the exact solution, just like the cell faces and the reference FE schemes. In Figure 6.1b a discontinuity starts to form. At this point each of the schemes capture the shock with a steep slope. Near the top, the schemes slightly underpredict the solution, with the bubble function with $\beta = 0.1$ the closest to exact, the SUPG with new polynomials the furthest. The difference is at most 2%. The reference schemes exhibit oscillations, both before and after the shock a spike can be seen. The rest of the solution is still correctly represented.
A few time steps later, at 0.2 seconds, oscillations start to occur for nearly all schemes. For the new methods, the differences between the different upwind methods start to show. As one could expect, the bubble function method with $\beta = 0.1$ is less damped in comparison with $\beta = 0.2$. With $\beta = 0.1$, the solution oscillates significantly after the shock. These oscillations require up to 12 cells to be damped out. With $\beta = 0.2$, the cell face values do oscillate, but the integrated cell values are less prone to move with the cell faces. Oscillations have disappeared in 8 cells. The SUPG method successfully stops all oscillations, but the solution does not reach the top of the wave. The reference methods have oscillations as well. In this case, the oscillations are present before and after the jump. Before the jump, wiggles are present in 4 cells. After the jump, difference in stabilization is more clear. With the piecewise linear polynomials with bubble functions, require 15 cells before the oscillations have disappeared. The SUPG scheme needs 12 cells to remove the wiggles. Finally, after the wave moved through the whole domain after 1 second, the results of Figure 6.2b are found. All schemes have been able to follow the shock and seem to have a correct solution before and after the discontinuity. For the reference methods, this is paired with oscillations. The piecewise linear polynomials with bubble function upwind shows oscillations up to 17 cells after the discontinuity and 4 before. The SUPG with piecewise linear polynomials also needs 4 cells before the discontinuity, but only 12 after. The new scheme with bubble function upwind yields better results in that respect. No wiggles are present before the jump, after the jump some wiggles can be found, but these are gone within 6 cells. With $\beta = 0.2$, all oscillations have disappeared. The jump location is correct for this method and the discontinuity is solved within 4 cells. The SUPG with the e-polynomials has no oscillations at all, but the jump seem to be slightly further than should be. 5 Cells are required to solve the discontinuity.



(a) (b) Close up

Figure 6.3: change in $\int_\Omega \rho d\Omega$ as function of time, as percentage of initial conditions. Numerical results found with 60 cells, starting CFL of 0.1.

6.2.2. Conservative Properties

In Figures 6.3, the change of total mass, $\int_\Omega \rho d\Omega$, over time is shown. From Figure 6.3a, it can be seen that the SUPG method with the new polynomials and the piecewise linear polynomials with bubble functions is not conservative. Both have a significant change in total mass. For the linear with bubble functions, the change is in the order of $\mathcal{O}\left(5 \cdot 10^{-3}\right)$%. For the SUPG with new polynomials, the result is even worse, a total change of 1.24% is found. All other schemes are perfectly conservative, at least globally. From Figure 6.3b, one can see variations in the order of $\mathcal{O}\left(10^{-13}\right)$% are present, this is within machine error.

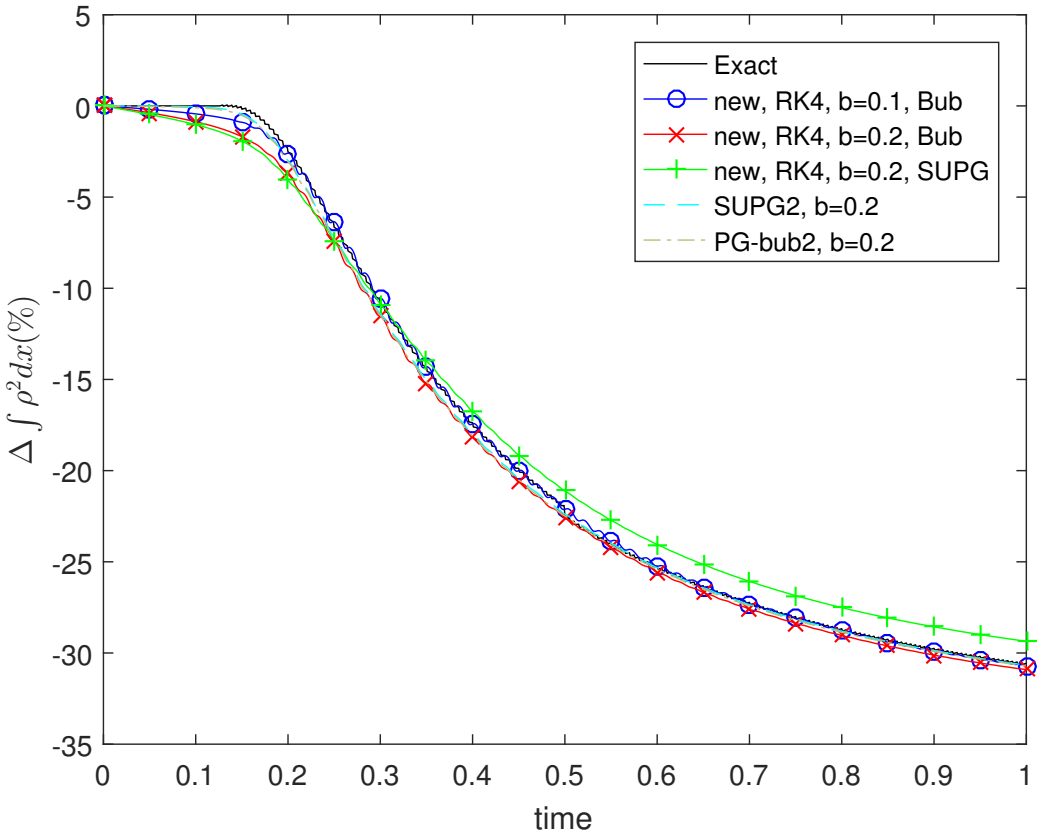
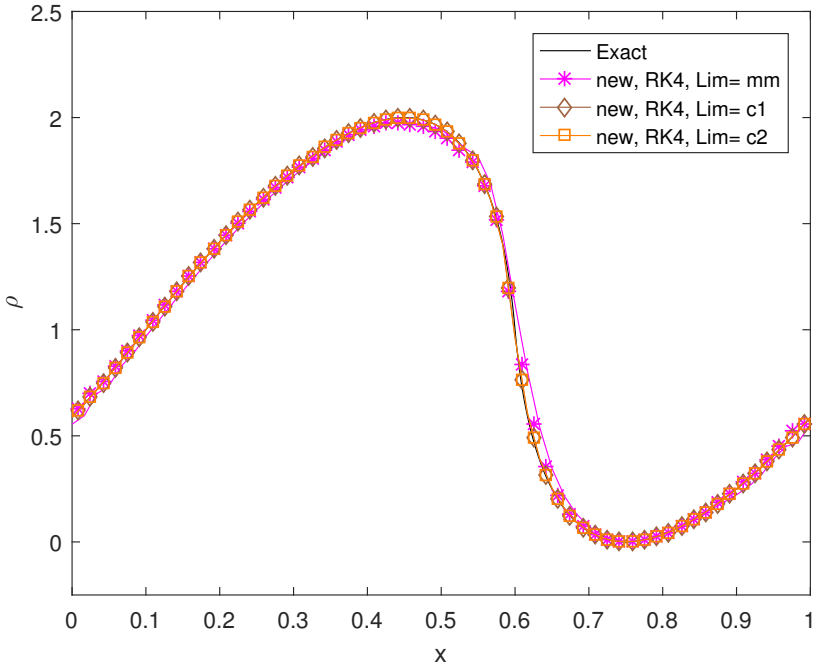
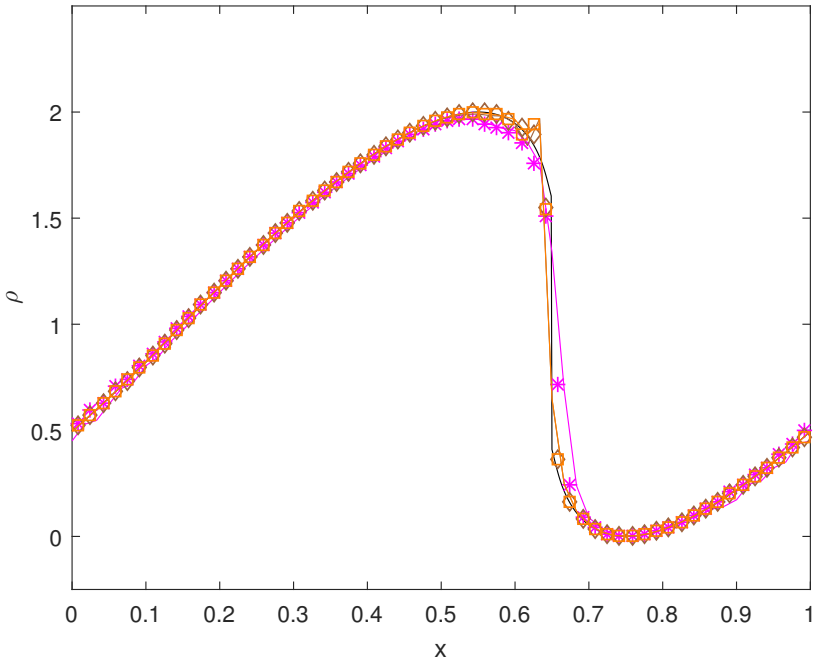


Figure 6.4: change in $\int_\Omega \rho^2 d\Omega$ as function of time, as percentage of initial conditions. Numerical results found with 60 cells, starting CFL of 0.1.

In Figure 6.4, the change in momentum, or $\int_\Omega \rho^2 d\Omega$, over time is shown. In contrast to the results from linear advection, this property changes over time, since pre and post shock values are averaged. As can be seen from the figure, all the tested schemes have diffusion changing the $\int_\Omega \rho^2 d\Omega$ before discontinuities occur. The SUPG method in combination with the new polynomials does not find the correct behaviour of $\int_\Omega \rho^2 d\Omega$. Before the discontinuity formation, this scheme has the highest diffusion of $\int_\Omega \rho^2 d\Omega$, while after the discontinuity formation it is the only scheme which finds a to high value. With bubble function upwind on the new polynomials, the scheme finds the correct decrease of $\int_\Omega \rho^2 d\Omega$. With the upwind parameter of $\beta = 0.1$, the value prior to discontinuity formation is slightly under predicted. With $\beta = 0.2$, the value of $\int_\Omega \rho^2 d\Omega$ is under predicted even more. This scheme has a slight under prediction after the shock formation as well. The comparison scheme with piecewise linear polynomials has only a slight under prediction of $\int_\Omega \rho^2 d\Omega$, both before and after discontinuity formation. For this property, the upwind mechanism has not so much effect.
A small note for the reader with a sharp eye, in the exact solution plot of Figure 6.4, a small jump can be found at a time of 0.5 seconds. The reason this jump is present there, has to do with the calculation of the exact solution at each node. In order to do this, each point is advected with its own solution, and points which reach the discontinuity are discarded. With the found locations at time n, an interpolation is done to get the values at each node. After half a second, the discontinuity has reached the right side of the domain an no polynomial can be constructed. Since the exact solution is already extremely close to linear, a linear interpolation is used after this time level. As a result the exact solution has a small dip.

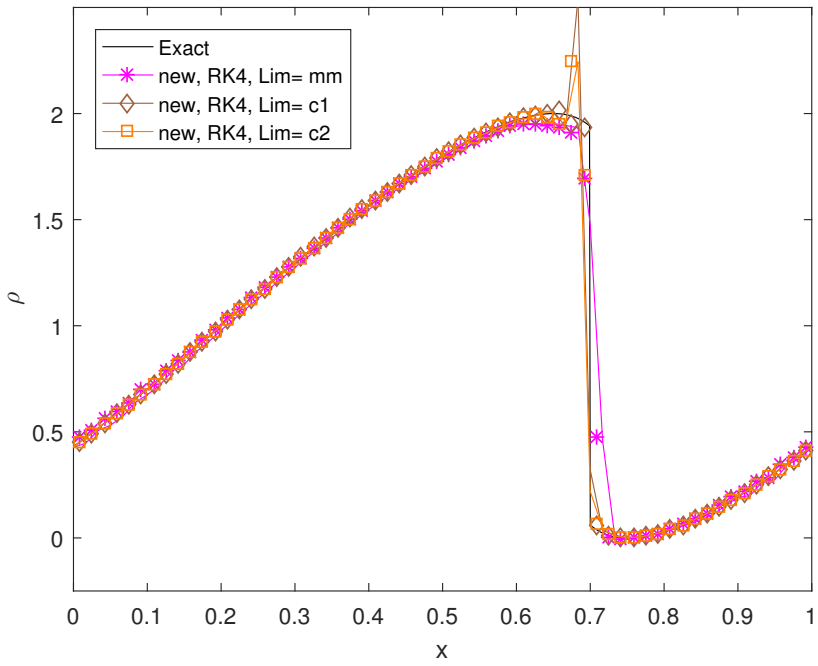


(a) $t = 0.1$

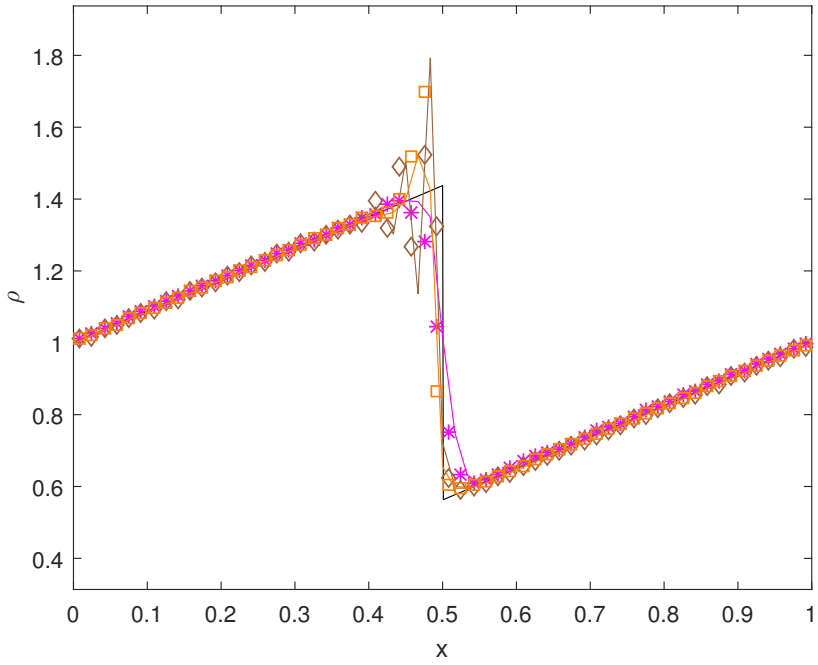


(b) $t = 0.15$

Figure 6.5: Results for non-linear advection with 2 parabola. With 60 cells and $\Delta t = 8.33 \cdot 10^{-4}$ and correctors applied



(a) $t = 0.2$



(b) $t = 1$

Figure 6.6: Results for non-linear advection with 2 parabola. With 60 cells and $\Delta t = 8.33 \cdot 10^{-4}$ and correctors applied

6.3. Results with Correctors

The second series of test is done using the corrector approach. As elaborated upon in Section 4.4, the corrector schemes correct the values of the cell faces. This is possible as these values do not contribute to the conservation properties of the scheme.

6.3.1. Qualitative Results

In the Figures 6.5 and 6.6, the results with correction methods are shown. The green line, with plus markers, is with minmod applied. The magenta line with asterix markers belongs to the corrector 1 scheme, the copper line with diamond markers is the corrector 2 scheme. Again, the markers are the integrated values within the cell. Prior to shock formation, see 6.5a, the results with the corrector 1 scheme (c1) and the corrector 2 scheme (c2) are the same as those found with upwind methods. In this case no correction has been performed and the scheme yields results close to exact results. The case with a minmod type corrector already starts to show deviations from the exact solution, as high gradients are limited and corrected on the cell faces. This effect becomes only stronger in the Figure 6.5b, when shock formation start.

At this time level, the first overshoot in the corrector 1 scheme starts to occur. Since this oscillation is still in the set upwind of that cell face, this is not corrected accordingly. Just before the discontinuity jump, the solutions is still following the exact solution neatly. The corrector 2 shows similar results, but the overshoot of the integrated value is slightly higher in comparison with the corrector 1. The flux value on the other hand, has a slightly less extreme overshoot.

At time $t = 0.2$, the shock is close to its maximum strength. For the corrector 1, this time level is paired with a very strong oscillation just behind the shock. In contrast to the Petrov-Galerkin scheme, these oscillations do not propagate upstream at this point. Also, the solution just in front of the shock is still matching the exact solution. The shock itself is resolved within 4 cells. For corrector 2, the results are very similar, but the overshoot is less severe and the following undershoot is dampened more as well. Again, the shock is resolved in 4 cells. The minmod type correction has a more diffusive behaviour, for one the solution starts to change 2 cells before the shock. No overshoot is occurring, but the solution does not reach the correct height either. A small plateau is created at a level of 1.95, while a value of 2 should be reached at this point.

Finally, the wave has travelled once through the whole domain in Figure 6.6b. The minmod corrector has spread the jump over 7 cells and thereby never reached the maxima or minima of this jump. The corrector 1 only needs 4 cells for the shock, but has a very strong overshoot. 7 cells are required to damp the resulting oscillations. In comparison, corrector 2 has the same amount of cells for the shock, but the overshoot is less extreme and there is only a minor undershoot. The oscillations are damped much more. Although the GFEM solution of Figure 6.2b has a much larger domain of oscillations, the maximum overshoot is less severe.

6.3.2. Conservative Properties

In Figure 6.7 the change of $\int_\Omega \rho^2 dx$ is plotted against time. The minmod corrected method displays some diffusion as soon as simulation is started and in the end the total diffusion is still to much. Both corrector 1 and corrector 2 keep this change close to zero. Before shock formation, the change is found to be in the order of $\mathcal{O}\left(10^{-4}\right)$. After shock formation, these correction methods in general follows the exact change very well. However, one can clearly see the moments corrections are performed, as the line is has oscillations as soon as the shock starts to form. The minmod corrector is not as bad as it is without shock, but still, more diffusion is present for this correction scheme than the other stabilization methods.

6.4. Error Convergence

In this section, the error convergence of each of the tested schemes is discussed. In Table 6.1, the error convergence rates are listed. These are the average slopes of the errors, which are plotted in Figure 6.8a and 6.8b for the L_1 and L_2 respectively. The error is calculated in the same way as done for linear advection, hence (5.4) is integrated using 1000 points for the L_1 and L_2 norms. Also tested again, is the L_2 convergence rate in which the scheme converge to a fine mesh solution. This is again done to validate the convergence rate.

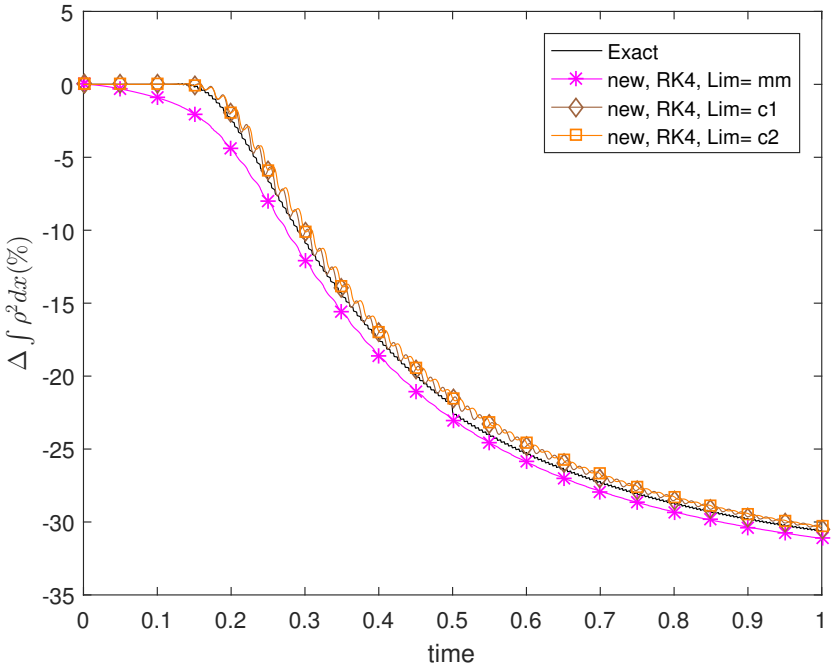


Figure 6.7: Change in $\int_\Omega \rho^2 dx$ over time.

Table 6.1: Average error convergence rates after $t = .1$ with $CFL = .1$

Scheme	$\lvert\rho - \rho_{ex}\rvert$	$\lvert\rho - \rho_{ex}\rvert_2$	$\lvert\rho - \rho_{320}\rvert_2$
new, Bub $\beta = .1$	-0.9406	-0.8695	-1.1697
new, Bub $\beta = .2$	-0.9208	-0.8183	-1.1028
new, SUPG $\beta = 0.2$	-0.9261	-0.8361	-1.1341
new, mm	-1.0029	-0.9351	-1.0403
new, c1	-1.6519	-1.3249	-1.4117
new, c2	-1.6622	-1.3267	-1.4139
SUPG2	-1.8763	-1.6831	-1.7548
PG-Bub2, $\beta = 0.2$	-1.5465	-1.5145	-1.7133

The time level for which Table 6.1 is generated, is at $t = 0.1$. This is just before the discontinuity forms, which happens at $t = 0.125$, hence the solution should be smooth. The new polynomials using upwind achieve approximate error convergence in the neighbourhood of 0.85 for the L_2 and 0.92 for the L_1 error. These convergence rates are lower than one would expect from a third order polynomial. With the correctors 1 and 2, the convergence rate is significantly higher, at approximately 1.32 for the L_2 and 1.65 for the L_1. However, in the case of non-linear problems the SUPG scheme has the highest convergence rate for both the L_2 and L_1 error.
In Figures 6.8a and 6.8b the corresponding error plots are shown. From this plot, it can be seen that the convergence rates for the upwind scheme with e-polynomials is indeed less, but also the initial error is worse. With correctors, the error convergence rate is closer to those found for the piecewise linear polynomials in the SUPG scheme. For the corrector case, the initial error is lower. Somewhere between 80 and 160 cells the errors of both schemes are equal. This is for both the L_2 and L_1 error rate.

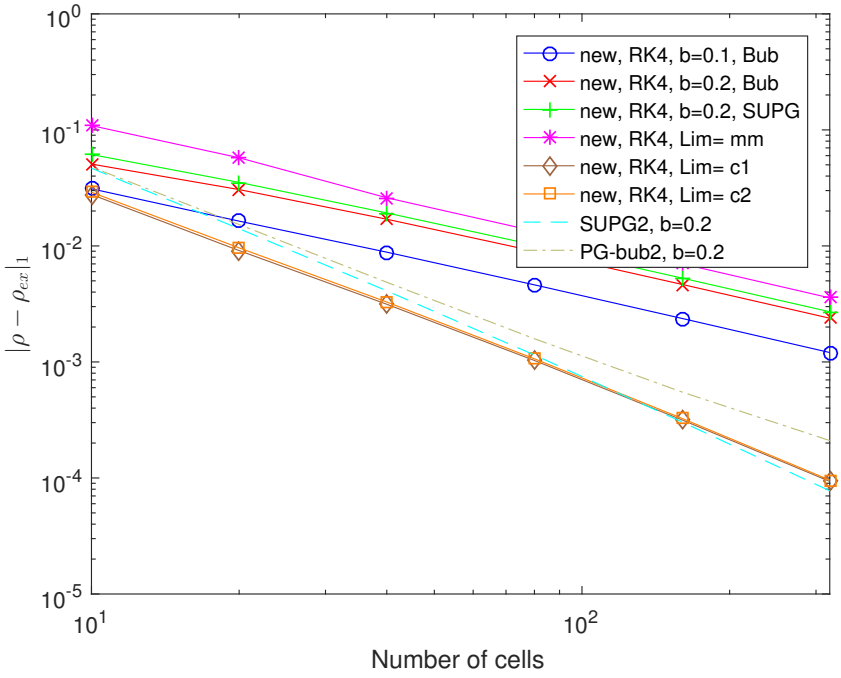
Table 6.2: Average error convergence rates after $t = .5$ with $CFL = .1$

Scheme	$\lvert\rho - \rho_{ex}\rvert$	$\lvert\rho - \rho_{ex}\rvert_2$	$\lvert\rho - \rho_{320}\rvert_2$
new, Bub $\beta = .1$	-1.0629	-0.6292	-0.5364
new, Bub $\beta = .2$	-1.0295	-0.6189	-0.7544
new, SUPG $\beta = 0.2$	-0.3598	0.0908	-0.8196
new, mm	-0.9695	-0.4988	-0.7846
new, c1	-1.0728	-0.5763	-0.4964
new, c2	-1.0897	-0.6430	-0.6359
SUPG2	-1.0562	-0.4912	-0.6064
PG-Bub2, $\beta = 0.2$	-1.0853	-0.5121	-0.6134

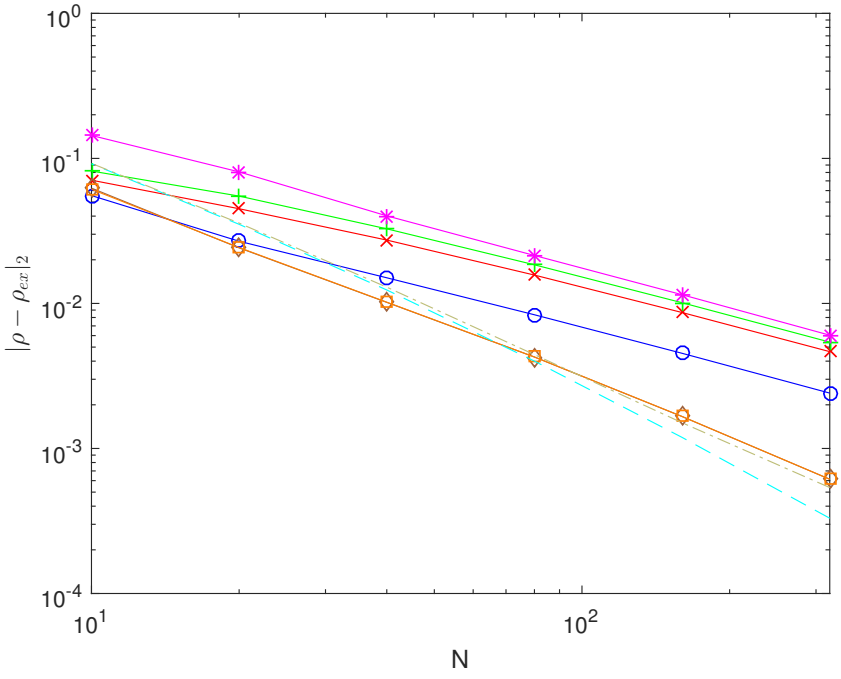
In Figures 6.9a and 6.9b the error convergence study has been performed again, but this time at $t = 0.5$. At this time level, the discontinuity is fully developed. The convergence rate expected would be $\frac{1}{2}$, since the solution is discontinuous. With exception of the e-polynomials with a SUPG upwind, all schemes exhibit similar error levels and convergences rates. For the L_2 norm, it seems as if the convergence rate is higher in comparison to the piecewise linear polynomial schemes. In Table 6.2, the convergence rates at an end time of 0.5 are listed. The e-polynomials show convergence rates of 0.6 and above, which is high considering a discontinuity is present.

6.5. Conclusions and Discussion

Using stabilization, the proposed methods using e-polynomials perform satisfactory for the simulated non-linear advection problem. Two methods of upwind were considered, which strongly affect the acquired results. In the SUPG approach, the scheme with e-polynomials seems to give reasonable results, but on inspection it was found that this approach is not conservative. This might have to do with the fact the SUPG approach has to be applied on all polynomials instead of the polynomials associated with the cell faces. When using SUPG, it is advised not to use e-polynomials.
When bubble functions are used in the test space, the results are much better. Conservation is ensured with this upwind method and with $\beta = 0.2$ all oscillations are smoothed out. A higher β comes at the cost of significantly more diffusion. With $\beta = 0.1$, not all oscillations are removed. This is only a small step in β, so selection of this parameter is rather sensitive. The convergence rates are slightly disappointing with the used upwind methods.
With corrector reasonable results could be obtained as well. However, the only corrector to fully prevent oscillations to occur is the minmod limiter. This method comes at the cost of a much higher numerical diffusion. The two corrector methods do allow oscillations to occur. The diffusive behaviour of this approach is significantly less.

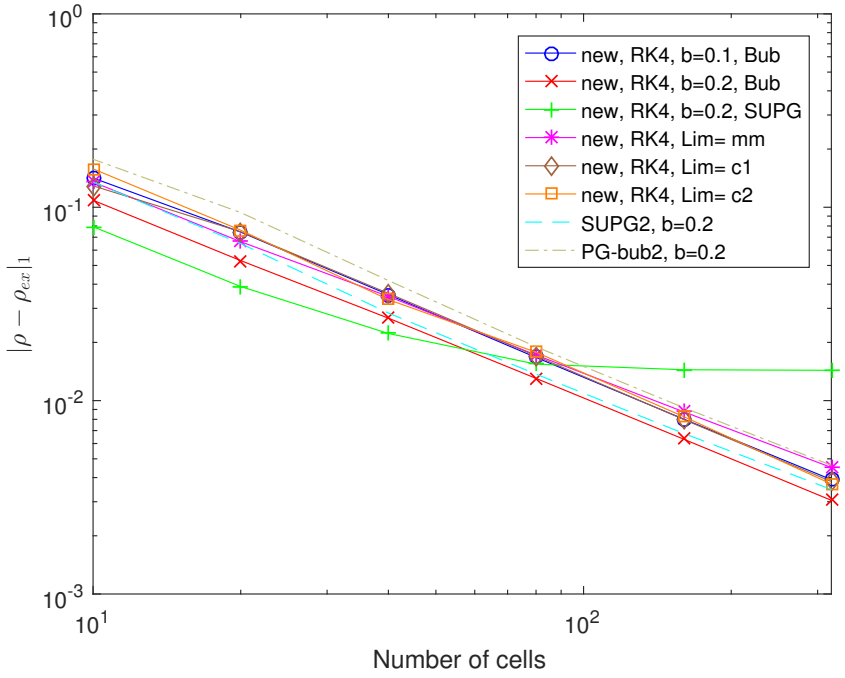


(a) L_1

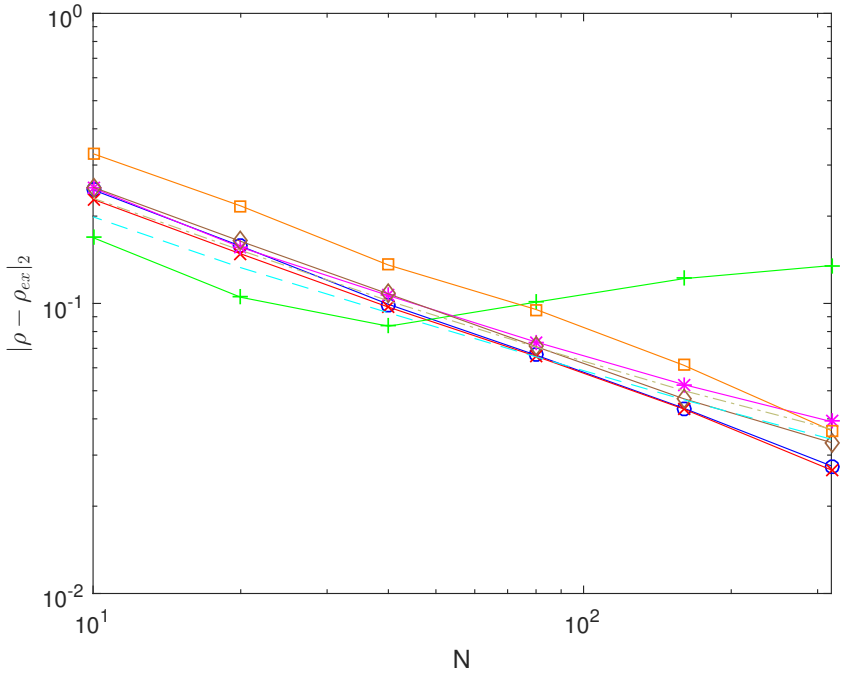


(b) L_2

Figure 6.8: Error convergence for each of the tested schemes, $t = 0.1$



(a) L_1



(b) L_2

Figure 6.9: Error convergence for each of the tested schemes, $t = 0.5$

7

1 Dimensional Euler Equations

The last part of the research is the 1 dimensional Euler equations. The Euler equations are the inviscid version of the full Navier-Stokes equations and are suitable for computation of advection dominated flow at high Mach number, such as supersonic flows. For this set of equations, a so-called shock-tube is modelled. A shock tube is a tube with two sets of initial conditions, separated by a membrane. After removal of the membrane at t_0, a combination of shocks and expansion fans can occur, depending on the initial conditions. The scheme is compared with a MUSCL scheme, with Roe's approximate Riemann solver and a superbee limiter. MUSCL schemes are currently widely used for supersonic flow calculations.

7.1. Problem Definition

In Section 3.3, the Euler equations are given by:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ m^x\\ \rho E\end{bmatrix} + \mathcal{L}_u\begin{bmatrix}\rho\\ m^x\\ \rho E\end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix}0\\ p\\ up\end{bmatrix} = 0. \tag{7.1a}$$

And these are closed with the pressure and velocity relations:

$$u = \frac{m^x}{\rho}, \tag{7.1b}$$

$$p = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho u^2\right), \tag{7.1c}$$

Equation (7.1a) can be further written in differential geometry terms:

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ m^x\\ \rho E\end{bmatrix} + \mathcal{L}_u\begin{bmatrix}\rho\\ m^x\\ \rho E\end{bmatrix} + d\begin{bmatrix}0\\ p\\ up\end{bmatrix} = 0. \tag{7.2}$$

Several shock tube problems will be tested. This means the domain, with a length of 1, is divided into two equal parts. For both the left and right initial domain, different initial conditions are used. Five test cases are defined, which are the same as those from Naber [34] and can also be found in Toro [46]. The initial conditions can be found in Table 7.1.
The first test case is the well known Sod's problem. In Sod's problem, a left running expansion wave and right running shock occur. This is one of the most used test cases for testing numerical schemes, as all the jumps have to be computed correctly in order to get a satisfactory result.
The second test case has two outflow boundaries. In this case, a vacuum is created. This happens due to two expansion waves, which accelerates the gas in different directions. Among the possible issues to occur, is when pressure is underestimated. A direct result of that is locally negative pressures are present which will blow-up the solution. Other issues can be overshoots in momentum, which result in rapid changes of the flow direction, which most likely blow-up the whole solution.

Table 7.1: Initial conditions 1D shock tube test cases[34]

Case	ρ_l	u_l	p_l	ρ_r	u_r	p_r	t_{end}
1	1.0	0	1.0	0.125	0.0	0.1	0.25
2	1.0	-2.0	0.4	1.0	2.0	0.4	0.15
3	1.0	0.0	1000.0	1.0	0.0	0.01	0.012
4	1.0	0	0.01	1.0	0	100	0.035
5	5.99924	19.5975	460.894	5.99242	-6.19633	46.0950	0.035

The third and fourth case is similar to Sod's problem, but with a much higher pressure jump. Due to this strong pressure differential, the smallest overshoots in the solution might result in negative pressures. Another difficulty with these cases, is that the velocity of the shock wave and the contact discontinuity velocity do not differ much. As a result, the region of post-shock conditions is small, but high concentrations of mass and energy are in this region. Since most schemes require some cells to compute jump conditions, it might be that this region is not computed correctly or even completely missed. This test case has high hypersonic flow conditions in the final solution. The difference between the third and fourth test case, is the flow direction.
Finally, the fifth problem has two inflow boundaries, with as a result, two shockwaves. The initial conditions are such that both waves should be right moving. In Figure 7.1, a sketch of the characteristics

Table 7.2: Exact results for the test cases [34, 46]

Case	p^*	u^*	ρ_l^*	ρ_r^*
1	0.30313	0.92745	0.42632	0.26557
2	0.00189	0	0.02185	0.02185
3	460.894	19.5975	0.57506	5.99924
4	46.0950	-6.1933	5.99242	0.57511
5	1691.64	8.68975	14.2823	31.0426

is shown. In this figure, one can find the regions left and right, as well as the *-regions left and right. In Table 7.2, for each of the test cases the exact solutions in the $*$-regions can be found. In the table, the subscripts $_l$ and $_r$ indicate the left and right region respectively. As pressure and velocity on both sides of the contact discontinuity are equal, no separation between left and right is made for these variables. The exact values are computed as discussed in Section 3.3. The pre shock/expansion states are unchanged from the initial conditions, hence can be found in Table 7.1.

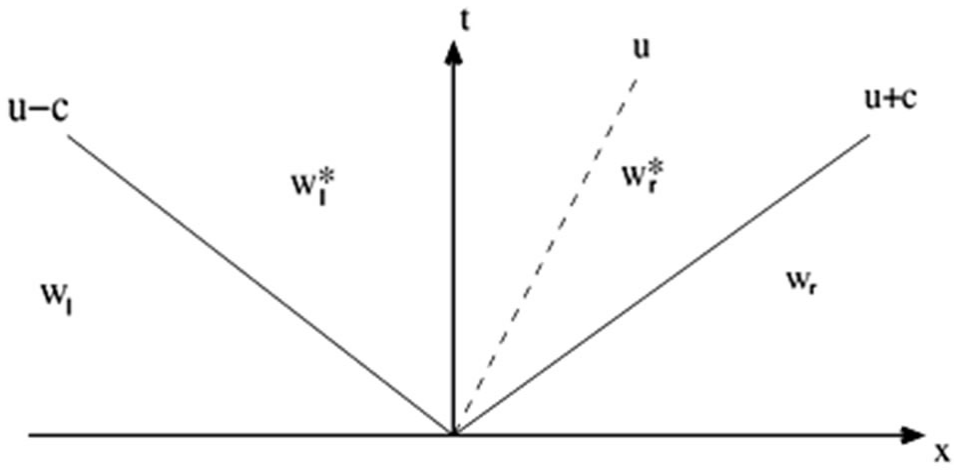


Figure 7.1: sketch of characteristics, taken from [21]

7.2. Numerical Treatment

For the numerical treatment of the Euler equations, the form in conservative variables as written in (7.1a) is used. The discretized system can be written into matrix notation:

$$\mathbb{M}\frac{\partial U}{\partial t} + \mathbb{F}U + \mathbb{P}U = 0. \tag{7.3}$$

In this equation, $\mathbb{M}$ is the mass matrix, $\mathbb{L}$ is the advection matrix and $\mathbb{P}$ is the pressure matrix. The vector U contains the complete solution within a cell:

$$U = \begin{bmatrix}\rho_L & \rho_I & \rho_R & m_L^x & m_I^x & m_R^x & \rho E_L & \rho E_I & \rho E_R\end{bmatrix}^T, \tag{7.4}$$

where the variables with subscript L are the values left of the cell, R the values right of the cell and I the integrated variable. The mass matrix is a diagonal block matrix, containing the local mass matrices as diagonal entries:

$$\mathbb{M} = \begin{bmatrix}\mathbb{M}_{\rho} & 0 & 0\\ 0 & \mathbb{M}_{m^x} & 0\\ 0 & 0 & \mathbb{M}_{\rho E}\end{bmatrix}. \tag{7.5}$$

Submatrices $\mathbb{M}_{\rho,m^x,\rho E}$ are the mass matrices for each variable. Since each variable uses the same polynomial approximation, these are all the same and are calculated as:

$$\mathbb{M}_{\rho,m^x,\rho E}\big|_{i,j} = \int_{-1}^{1} e_i(\xi)e_j(\xi)dx, \tag{7.6}$$

with ξ the transformed x location in the cell.
For advection, a similar matrix is constructed. Just like the mass matrix in 7.5, matrix $\mathbb{L}$ is a diagonal matrix consisting of submatrices. In this instance, each submatrix of $\mathbb{L}$ is in the form of (4.27), hence:

$$\mathbb{F}_{\rho,m^x,\rho E} = \begin{bmatrix}\int_{-1}^{1}\frac{\partial u_R e_1}{\partial x}e_1dx & \int_{-1}^{1}\frac{\partial u_I e_2}{\partial x}e_1dx & \int_{-1}^{1}\frac{\partial u_R e_3}{\partial x}e_1dx\\ \frac{-u_L}{\Delta x} & 0 & \frac{u_R}{\Delta x}\\ \int_{-1}^{1}\frac{\partial u_L e_1}{\partial x}e_3dx & \int_{-1}^{1}\frac{\partial u_I e_2}{\partial x}e_3dx & \int_{-1}^{1}\frac{\partial u_R e_3}{\partial x}e_3dx\end{bmatrix}. \tag{7.7}$$

The convective velocity u is built up with the same polynomials, with:

$$u_L = \frac{m_L^x}{\rho_L}, \qquad u_I = \frac{\overline{m}_I^x}{\overline{\rho}_I}, \qquad u_R = \frac{m_R^x}{\rho_R}.$$

When the pressure is expressed in conserved variables, the pressure terms can be expressed as:

$$d\begin{bmatrix}0\\ p\\ up\end{bmatrix} = d\begin{bmatrix}0\\ (\gamma-1)\left(\rho E - \frac{1}{2}um^x\right)\\ u(\gamma-1)\left(\rho E - \frac{1}{2}um^x\right)\end{bmatrix}. \tag{7.8}$$

Since the d operator is additive, the matrix $\mathbb{P}$ can be formed:

$$\mathbb{P} = \begin{bmatrix}0 & 0 & 0\\ 0 & \mathbb{P}_{m^x|m^x} & \mathbb{P}_{m^x|\rho E}\\ 0 & \mathbb{P}_{\rho E|m^x} & \mathbb{P}_{\rho E|\rho E}\end{bmatrix}. \tag{7.9}$$

In this, the local matrices which take the d of the momentum and energy part of the pressure. This results in the four non-zero local matrices:

$$\mathbb{P}_{m^x|m^x} = \left(\frac{1-\gamma}{2}\right) \begin{bmatrix} u_L \int_{-1}^{1} \frac{\partial e_1}{\partial x} e_1 dx & u_I \int_{-1}^{1} \frac{\partial e_2}{\partial x} e_1 dx & u_R \int_{-1}^{1} \frac{\partial e_3}{\partial x} e_1 dx \\ \frac{-2u_L}{\Delta x} & 0 & \frac{2u_R}{\Delta x} \\ u_L \int_{-1}^{1} \frac{\partial e_1}{\partial x} e_3 dx & u_I \int_{-1}^{1} \frac{\partial e_2}{\partial x} e_3 dx & u_R \int_{-1}^{1} \frac{\partial e_3}{\partial x} e_3 dx \end{bmatrix}, \tag{7.10a}$$

$$\mathbb{P}_{m^x|\rho E} = (\gamma - 1) \begin{bmatrix} \int_{-1}^{1} \frac{\partial e_1}{\partial x} e_1 dx & \int_{-1}^{1} \frac{\partial e_2}{\partial x} e_1 dx & \int_{-1}^{1} \frac{\partial e_3}{\partial x} e_1 dx \\ \frac{-2}{\Delta x} & 0 & \frac{2}{\Delta x} \\ \int_{-1}^{1} \frac{\partial e_1}{\partial x} e_3 dx & \int_{-1}^{1} \frac{\partial e_2}{\partial x} e_3 dx & \int_{-1}^{1} \frac{\partial e_3}{\partial x} e_3 dx \end{bmatrix}, \tag{7.10b}$$

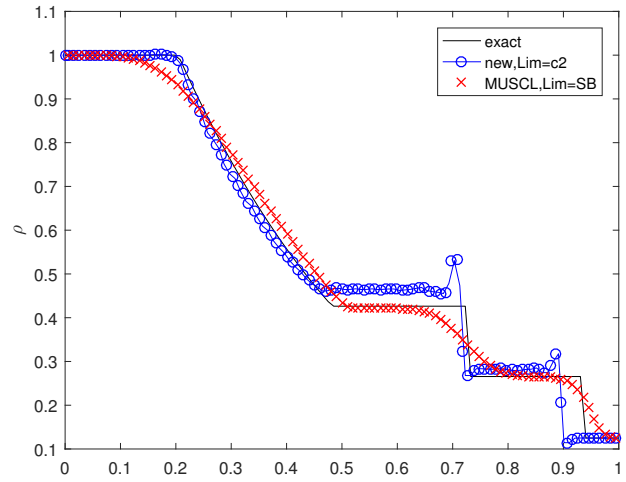
$$\mathbb{P}_{\rho E|m^x} = \left(\frac{1-\gamma}{2}\right) \begin{bmatrix} u_L^2 \int_{-1}^{1} \frac{\partial e_1}{\partial x} e_1 dx & u_I^2 \int_{-1}^{1} \frac{\partial e_2}{\partial x} e_1 dx & u_R^2 \int_{-1}^{1} \frac{\partial e_3}{\partial x} e_1 dx \\ \frac{-2u_L^2}{\Delta x} & 0 & \frac{2u_R^2}{\Delta x} \\ u_L^2 \int_{-1}^{1} \frac{\partial e_1}{\partial x} e_3 dx & u_I^2 \int_{-1}^{1} \frac{\partial e_2}{\partial x} e_3 dx & u_R^2 \int_{-1}^{1} \frac{\partial e_3}{\partial x} e_3 dx \end{bmatrix}, \tag{7.10c}$$

$$\mathbb{P}_{\rho E|\rho E} = (\gamma - 1) \begin{bmatrix} u_L \int_{-1}^{1} \frac{\partial e_1}{\partial x} e_1 dx & u_I \int_{-1}^{1} \frac{\partial e_2}{\partial x} e_1 dx & u_R \int_{-1}^{1} \frac{\partial e_3}{\partial x} e_1 dx \\ \frac{-2u_L}{\Delta x} & 0 & \frac{2u_R}{\Delta x} \\ u_L \int_{-1}^{1} \frac{\partial e_1}{\partial x} e_3 dx & u_I \int_{-1}^{1} \frac{\partial e_2}{\partial x} e_3 dx & u_R \int_{-1}^{1} \frac{\partial e_3}{\partial x} e_3 dx \end{bmatrix}. \tag{7.10d}$$

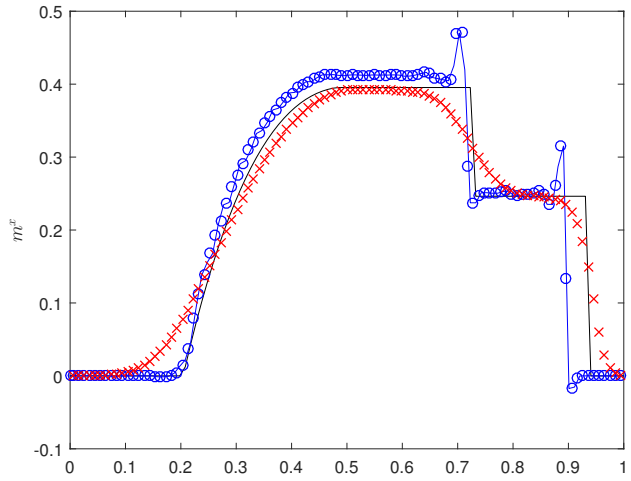
In cases for which upwind is required for a stable solution, bubble functions are applied as before. The direction of the bubble is purely based on flow direction, hence u. At this point, no differentiation between u, $u + a$ and $u - a$ is made. The bubble functions are used in the whole testspace, hence for the mass matrix, the advection matrix and the pressure matrix.
Now the system of equations is created, the time integration can be done. One could use a Crank-Nicolson scheme with iterations until convergence, creating a predictor-corrector scheme. In that case the velocity field is taken at time level $n + \frac{1}{2}$. The pressure matrix can be applied to both levels n and $n + 1$, or only on time level $n + 1$, which gives a fully implicit pressure integration. This is often done to get a stable pressure integration. Another option which works equally well, is to use a Runge-Kutta time scheme. The default time integrator is the predictor-corrector scheme.
After each computation, whether Crank-Nicolson iterations or a Runge-Kutta scheme is used, a correction of the solution is performed. For the Euler equations, this is done by means of corrector 2 approach. The integrated values are only corrected for the modified advection flux, hence the pressure change due to correction of cell faces is neglected at this point. This means the correction scheme is exactly the same for each term and follows Equations (4.46) and (4.47). In the situation density or energy becomes negative, an additional correction is made. For the points on cell faces this is done in the same manner as the flux correction. In case the integrated density or energy becomes negative, the outflow is too large. Therefore, mass or energy is taken from the downstream cell.
For cases with high stability requirements, such as test cases 3 and 4, upwind is applied as well. Bubble functions are used again, which are scaled with a constant c_τ. The function is added to the weight function, as previously used (4.36).
The reference method is a MUSCL scheme. The scheme is limited using a superbee limiter. Fluxes are calculated using Roes approximate Riemann solver. Full details of this scheme can be found in [34].

7.3. Results

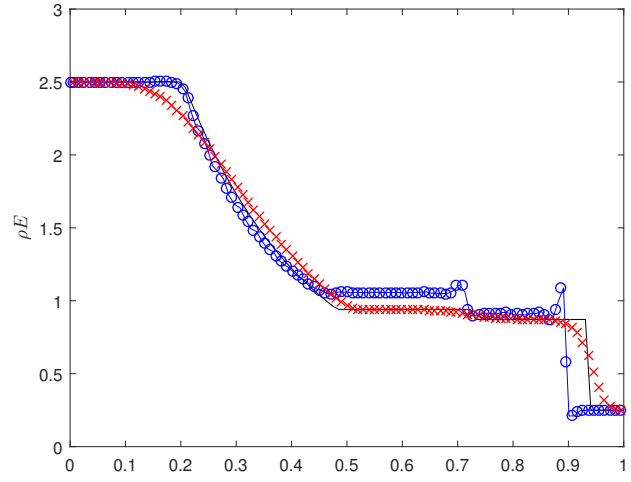
All results are generated on a grid with 101 cells. However, the timestep is different for each case, since flow velocities, and solution features are different. The solutions are shown in conservative variables, since the computations are in these variables as well. In the tables with results, the average conditions in the different regions are given. The averages are taken from all cells which are clearly in one of the $*$-regions. This means that the cells are sufficiently far from the characteristics, such that the influence of oscillations can be neglected. In the cases no average could be computed, for instance because the solution cannot achieve a steady post state, the extrema is taken. Discontinuity locations are estimated from the figures, for which a point between the pre and post jump states is selected.



(a) Density



(b) Momentum



(c) Energy

Figure 7.2: Results for test case 1: Sod's problem

7.3.1. Test Case 1

For the first test case, a time step of $1 \cdot 10^{-3}$ is taken. The results shown in Figures 7.2 are generated using the iteration method, but a Runge-Kutta scheme yields similar results. The most important results found are listed in Table 7.3. In this table, the results for the * regions are average results, the important locations are hand picked. These results have the correct features, a shock to the right, a contact discontinuity and an expansion fan are present.
However, taking a closer look reveals some issues. The first issue is that the values between the contact discontinuity and the expansion fan are overestimated. The density should be 0.4263, the average of the computed results in this domain is 0.4643. The same discrepancy is found for momentum with 0.3954 exact and 0.4116 computed and for energy 0.9412 and 1.0558 respectively. This means an overestimation of 4% for momentum up to 12% for energy density.
The second issue is the shock velocity. From the figures, it can be seen that the shock should be at $x = 0.938$, while the shock location found is at $x \approx 0.90$. The location of the contact discontinuity matches slightly better, the exact solution is at $x = 0.7319$, the found value is at $x \approx 0.72$. On the other hand, the total mass and energy in the system is only changed in the order of $\mathcal{O}\left(10^{-13}\right)$% as it should be since no additional mass or energy is added. Improvement of either the solution after the expansion fan or the location of the shock will probably result in a much more accurate overall solution. in Section 7.5, calculations are performed in order to find the sources of error. A final issue are the overshoots after each discontinuity, and the small wiggles found near every jump. Although damped relatively quickly, in a domain of approximately 0.05, these might cause problems when flow structures are thin, such as a shock with a contact discontinuity shortly after.
In comparison, the MUSCL scheme smears out jumps much more, which is due to the limiter. Jump locations are better estimated, as well as final states. The expansion fan is smeared out as well. When a time level is selected closer to the start of the computations (not shown), one finds that the MUSCL scheme requires more time to compute steady post jump conditions. In comparison with the new methods, the steady state solutions of the MUSCL scheme, are consistently closer to the exact solution.

Table 7.3: Results for test cases 1

	new method	MUSCL	Exact
ρ_l^*	0.4643	0.4216	0.4263
ρ_r^*	0.2815	0.2658	0.2657
$m_l^{x,*}$	0.4116	0.3919	0.3954
$m_r^{x,*}$	0.2489	0.2463	0.2463
ρE_l^*	1.0558	0.9380	0.9412
ρE_r^*	0.9088	0.8706	0.8720
x_{shock}	$\sim$ 0.90	$\sim$ 0.94	0.9380
x_{dc}	$\sim$ 0.72	$\sim$ 0.72	0.7319
$x_{exp,h}$	$\sim$ 0.20	$\sim$ 0.08	0.2042
$x_{exp,t}$	$\sim$ 0.48	$\sim$ 0.51	0.4824

7.3.2. Test Case 2

The second test case is a difficult one. The velocities are not computed correctly, resulting in one cell which is not becoming close to a vacuum. The cells which follow do become vacuum, but overshoot which results in negative pressures. The density just before the solution diverges can be found in Figure 7.3. Within a couple of timesteps the solution diverges and blows up. Unfortunately, no solution using the new polynomials for this problem is found yet. A MUSCL scheme with an exact Riemann solver for flux calculation should be able to solve this problem, although the selected limiter and the use of an entropy fix can severely affect the outcome. See [34] for details.

7.3.3. Test Case 3

The third test case posed severe difficulties. With a small timestep of $1 \cdot 10^{-5}$ and an upwind parameter chosen as 0.2, a solution could be found. With these settings, the results of Figures 7.4 are found. The

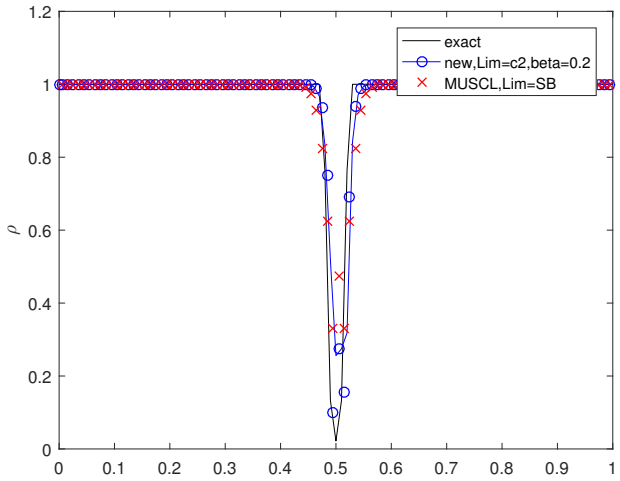
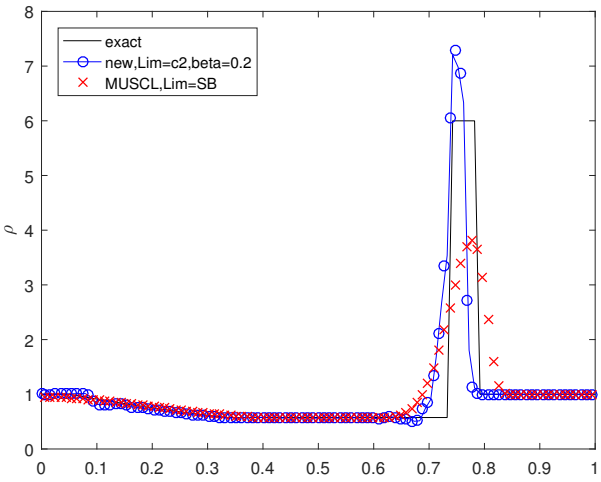


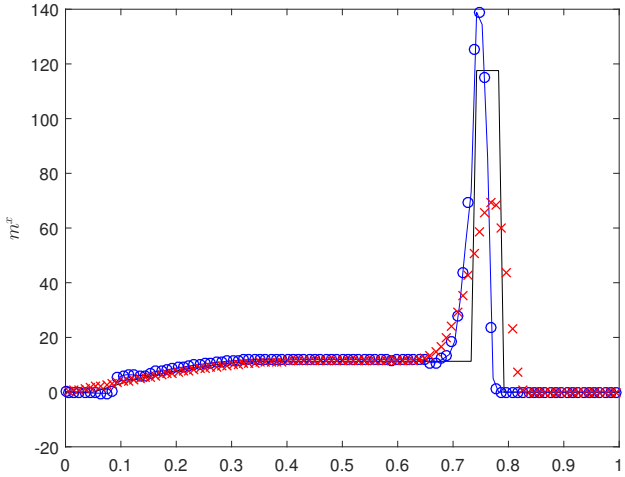
Figure 7.3: Density of the second test case, just before the solution diverges

Table 7.4: Results for test cases 3

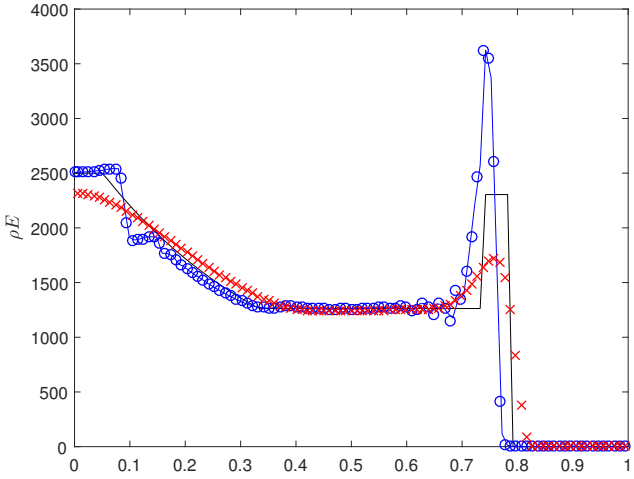
	new method	MUSCL	Exact
ρ_l^*	0.5660	0.5659	0.5751
ρ_r^*	7.294	3.818	5.999
$m_l^{x,*}$	11.7790	11.3202	11.27
$m_r^{x,*}$	138.9	69.85	117.6
ρE_l^*	1268	1245	1263
ρE_r^*	3624	1731	2304
x_{shock}	$\sim$ .76	$\sim$ 0.79	0.7822
x_{dc}	$\sim$.73	$\sim$ 0.71	0.7352
$x_{exp,h}$	$\sim$ 0.085	$\sim$ 0.01	0.05100
$x_{exp,t}$	$\sim$ .335	$\sim$ 0.39	0.3332



(a) Density



(b) Momentum



(c) Energy

Figure 7.4: Results for test case 3

most important features are listed in Table 7.4. The entries for the post shock states are maxima, as no steady condition was found. This is due to the fact the shock and contact discontinuity follow closely.
From the data, one can find a shock, discontinuity and expansion wave, but in comparison to test case 1 more issues exist. As expected, one of the issues is found between the shock and the contact discontinuity. This area is relatively small, only 6 cells in dimension. Without upwind, oscillations occurring at the left side of the discontinuity resulted in negative energy and mass, which blew-up the solution. With the upwind applied, the smearing of the solution is larger than the space between shock and contact discontinuity. From this, the diffusive behaviour of upwind is clear.
The second issue can be found in the expansion fan. The expansion fan is computed correct between $x = 0.15$ and $x = 0.35$. However, left of this the expansion fan transforms into an expansion shock. This is a non-physical phenomenon, which is caused by to the upwind stabilization. The fact this happens due to the upwind method was found when test case 1 was performed with upwind, in which a similar issue arose. The result of that particular test is not shown. Again, the total change of mass and energy in the system is extremely small, in the order of $\mathcal{O}\left(10^{-12}\right)$%. The maximum error found in Table 7.4 is 4.5%.
In contrast to the new methods, the MUSCL scheme is unable to reach the values of the right post shock state. The solution in that area is completely smooth and compression transitions seamlessly in expansion to the left post expansion states. While the new methods under predicts shock velocity, the MUSCL scheme has a slightly higher than correct shock velocity.

Table 7.5: Results for test cases 4

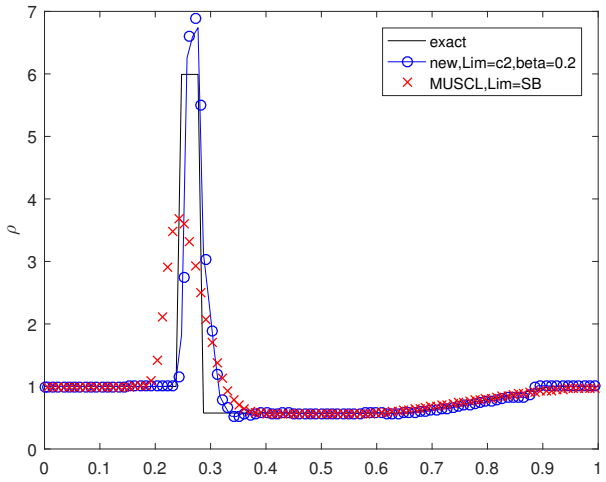
	new method	MUSCL	Exact
ρ_l^*	6.886	3.689	5.992
ρ_r^*	0.5659	0.5662	0.5751
$m_l^{x,*}$	-41.38	-21.28	-37.13
$m_r^{x,*}$	-3.730	-3.586	-3.564
ρE_l^*	342.3	169.7	230.3
ρE_r^*	126.7	124.5	126.3
x_{shock}	$\sim$ 0.25	$\sim$ 0.22	0.2397
x_{dc}	$\sim$ 0.29	$\sim$ 0.30	0.2831
$x_{exp,h}$	$\sim$ 0.89	$\sim$ 0.99	0.9141
$x_{exp,t}$	$\sim$ 0.65	$\sim$ 0.61	0.6539

7.3.4. Test Case 4

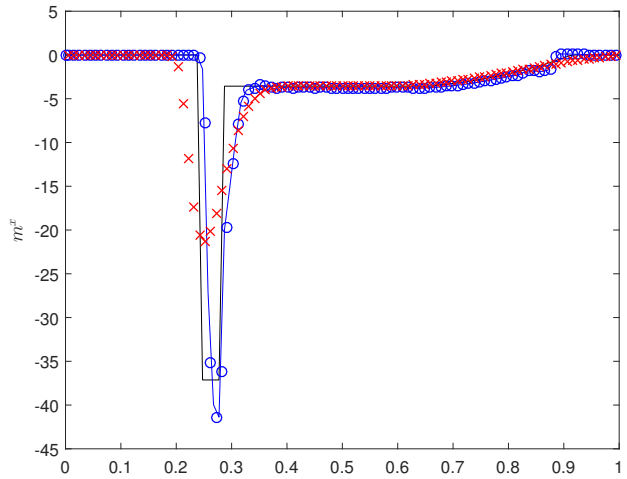
In Figures 7.5 and Table 7.5, the results for the test case 4 are shown. For this test case, the time step is $1 \cdot 10^{-5}$ and the upwind parameter, β is 0.2. As expected, the results are similar to test case 3, but left running instead of right running. Just as with test case 3, the region between shock and contact discontinuity is too small to resolve the solution sufficiently. Due to that, a single strong overshoot is present, which cannot be dampened out before the contact discontinuity. In this test case, the expansion fan again transforms in a non-physical expansion shock. The maximum deviation from the exact solution is 4.6%. In this case, the variation of conservation variables is again in the order of $\mathcal{O}\left(10^{-12}\right)$%.
For the comparison with MUSCL, the findings are similar to the previous test case. The newly developed method has a huge overshoot, while the MUSCL does not reach the true values of the pre shock conditions. The contact discontinuity is smeared out approximately the same width. This time, the discrepancy in shock location is slightly less for the new method, but still lagging behind. Just as in the previous test case, the expansion fan is expanded to much in both directions. The head of the expansion wave is even outside of the domain.

7.3.5. Test Case 5

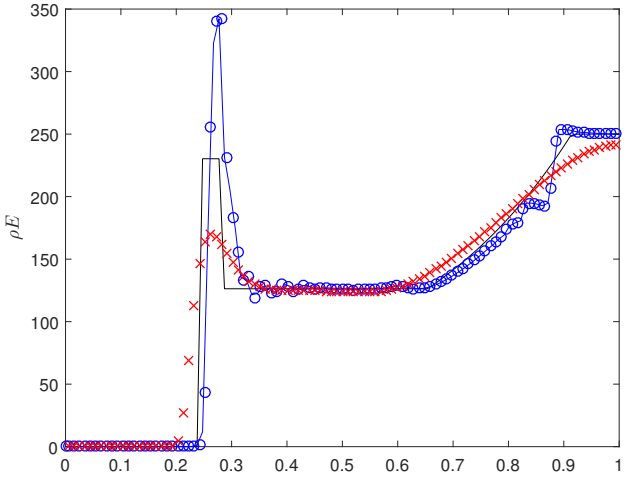
In Figures 7.6, the results for test case 5 are shown, with the important features listed in Table 7.6. For this test case, a timestep of $\Delta t = 10^{-5}$ was chosen. For this test case, even more upwind was necessary, the upwind parameter β was set to 0.4. From the results, it is clear that the upwind adds significant amount of diffusion. Directly after the right running shockwave, the solution gradually drops to the region of the left running wave. Due to this, no average post right running shock condition could



(a) Density

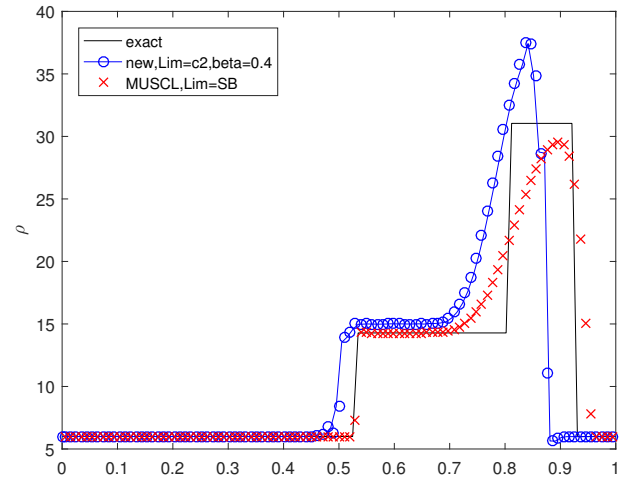


(b) Momentum

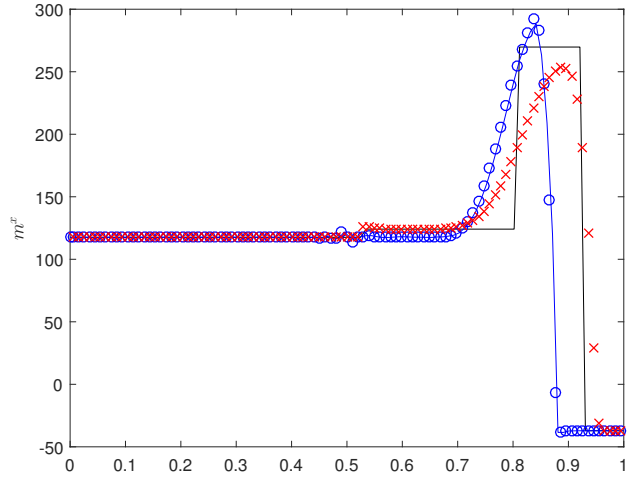


(c) Energy

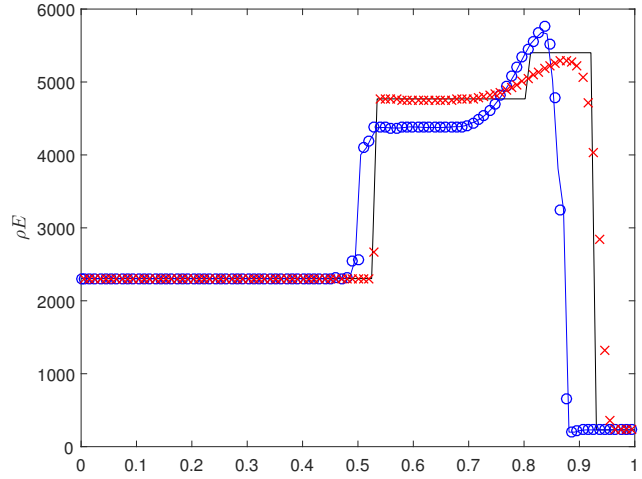
Figure 7.5: Results for test case 4



(a) Density



(b) Momentum



(c) Energy

Figure 7.6: Results for test case 5

be determined. The right running shock wave is also moving with a lower velocity. In this case, the difference in shock location is significant, at $x = 0.928$ for the exact solution and at $x = 0.87$ for the new scheme. The left running shock wave is close to the exact found location. However, the strength of each jump is significantly off. For density, the exact results are 5% overestimated. The momentum however, is 5% underestimated. The jump in momentum should be small, but now there is no jump in momentum. Energy is 8% underestimated.
In this case, the MUSCL scheme approximates all variables with higher accuracy. The right running shockwave is smeared a little, but overall resolved close to the exact location. The peak after the right running shock is underestimated for all variables, and directly starts to transition to the second area. The second area, behind the left running shockwave, is resolved extremely accurately. The results are less than 1% off. The left running shockwave is also resolved in 1 cell.

Table 7.6: Results for test cases 5

	new method	MUSCL	Exact
ρ_l^*	15.01	14.27	14.28
ρ_r^*	37.50	29.53	31.04
$m_l^{x,*}$	117.7	124.4	124.1
$m_r^{x,*}$	292.6	253.4	269.8
ρE_l^*	4377	4757	4768
ρE_r^*	5769	5296	5401
$x_{shock,l}$	$\sim$ 0.51	$\sim$0.53	0.5276
x_{dc}	-	-	.8041
$x_{shock,r}$	$\sim$ 0.87	$\sim$ 0.93	0.9288

7.4. Mesh Refinement

Since test case 1 is solved with the most ease and the different errors are clearly visible, a mesh refinement study is performed for this test case. A mesh with half the amount of cells and a mesh with double the cells are used. In Figure 7.7, the density results for each of the grids are shown. From these results, one can find that the results converge to the wrong solution. The shock and the contact discontinuity locations do not move in direction of the exact solution. Also, the results of the $*$-region between the contact discontinuity and the expansion fan does not significantly change in result, hence is still of from the exact solution.
From the plots, one can also find that the oscillations become sharper with a refined mesh, but are also compressed in a smaller area near the discontinuity.
The reference MUSCL scheme does converge to the exact solution. The discontinuities become more present with a refined mesh as well.

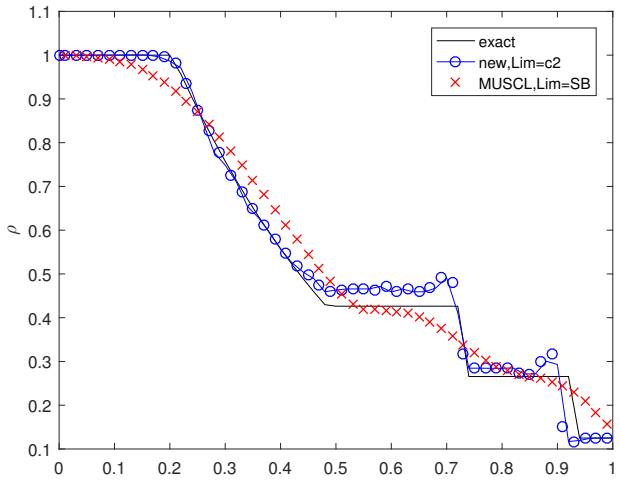
7.5. Error Sources

With the results found for test case 1, as discussed in Section 7.3.1, an effort is made to find the error sources. In order to do so, (3.14) and (3.21) are used to check whether the solution satisfies the Hugoniot shock relations and the Poisson curve. But first, the contact discontinuity is investigated. For the contact discontinuity, no difference in velocity nor pressure is allowed. This is easily checked:

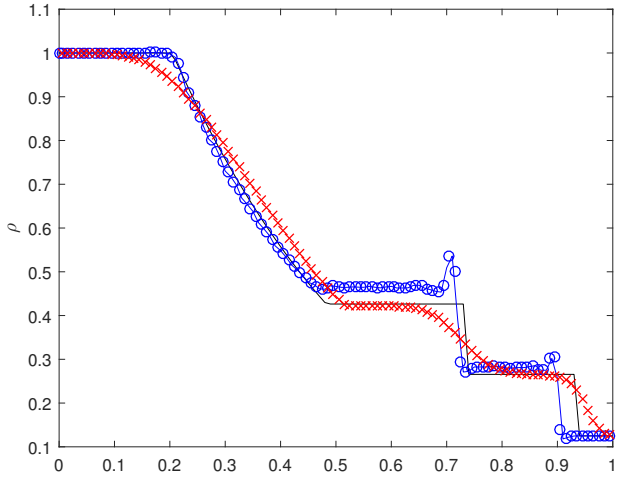
$$u_l^* = \frac{m_l^{x,*}}{\rho_l^*} = \frac{0.4116}{0.4643} = 0.8865, \qquad u_r^* = \frac{m_r^{x,*}}{\rho_r^*} = \frac{0.2489}{0.2815} = 0.8842,$$

$$p_l^* = (1.4-1)\left(1.0558 - \frac{1}{2}0.4643 \cdot 0.8865^2\right) = 0.3493, \quad p_r^* = (1.4-1)\left(0.9088 - \frac{1}{2}0.2815 \cdot 0.8842^2\right) = 0.3195.$$

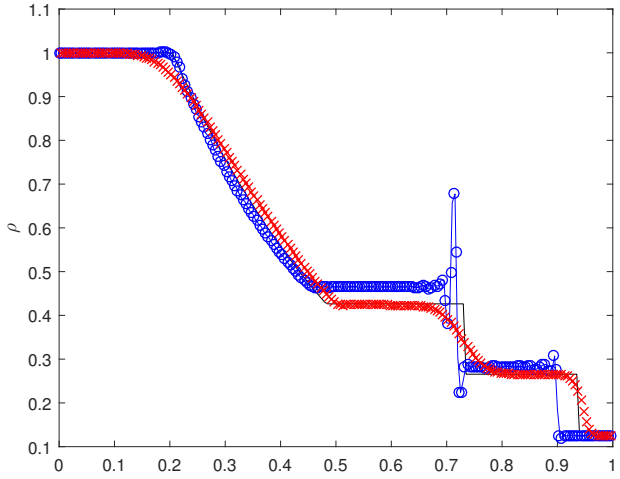
From this, it can be concluded that no significant velocity difference is present, but a small pressure difference is present. This pressure difference cannot be ignored and can be one of the most important contributors to error.



(a) 50 cells



(b) 100 cells



(c) 200 cells

Figure 7.7: Density of test case 1, with mesh refinement

Next, the shock relations are checked. The shock velocity is approximately:

$$\begin{aligned} u_{cs} &= \frac{x_{shock}}{t}, \\ &= \frac{0.90}{0.25} = 3.6ms^{-1}. \end{aligned} \tag{7.11}$$

The pressure after the shock is taken from the average values found:

$$\begin{aligned} p_r^* &= (\gamma - 1)\left(\rho E - \frac{1}{2}\rho u^2\right), \\ &= (1.4 - 1)\left(0.9088 - \frac{1}{2}\frac{0.2489^2}{0.2815}\right), \\ &= 0.3195. \end{aligned} \tag{7.12}$$

The speed of sound in the gas can be found:

$$a_r = \sqrt{\frac{\gamma p_r}{\rho_r}} = \sqrt{\frac{1.4 \cdot 0.1}{0.125}} = 3.3466ms^{-1} \tag{7.13}$$

Reciting (3.14):

$$p_r^* - p_r = \rho_r a_r \sqrt{1 + \frac{\gamma + 1}{2\gamma}\frac{p_r^* - p_r}{p_r}}\left(u_r^* - u_r\right). \tag{7.14}$$

To find out whether or not the Hugoniot relations are satisfied, the results found are put into the equation:

$$\begin{aligned} 0.3195 - 0.1 = 0.125 \cdot 3.3466\sqrt{1 + \frac{1.4 + 1}{2 \cdot 1.4}\frac{0.3195 - 0.1}{0.1}}\left(\frac{0.2489}{0.2815} - 0\right), \\ 0.2195 \neq 0.2526 \end{aligned} \tag{7.15}$$

These results are unequal, hence the Hugoniot shock relations are not complied with.
For the expansion fan a similar calculation can be performed, and then put into (3.21). The only unknown variable to do so, is the speed of sound in the left domain, which is:

$$a_l = \sqrt{\frac{\gamma p_l}{\rho_l}} = \sqrt{\frac{1.4 \cdot 1}{1}} = 1.1832ms^{-1} \tag{7.16}$$

Inserting all variable into (3.21) gives:

$$\begin{aligned} \left(\frac{p_{post}}{p_{pre}}\right)^{\frac{\gamma-1}{2\gamma}} &= 1 + \frac{\gamma - 1}{2}\frac{u_{post} - u_{pre}}{a_{pre}}, \\ \left(\frac{0.3493}{1}\right)^{\frac{\gamma-1}{2\gamma}} &= 1 + \frac{\gamma - 1}{2}\frac{-0.8865 + 0}{1.1832}, \\ 0.8605 &= 0.8502 \end{aligned} \tag{7.17}$$

In this, a negative sign for velocity is used, as the expansion fan is left moving. In this case, the Poisson relations are satisfied within a margin of 0.01.
Two main sources of error are found. The first error was found in the Hugoniot shock relations, which resulted in an error of 0.033. The second important error source is found near the contact discontinuity. Although no velocity difference was found, there is a pressure difference. To the authors believe, the main contribution of error is the pressure difference near the contact discontinuity, since the contact discontinuity determines the balance of post shock and expansion conditions.

7.6. Conclusions

The Euler equations have proved to be a challenge. Although results can be found, these are far from perfect. The quality of the results is highly depending on the problem. A solution for Sod's shock tube

problem can be found, but there is no convergence to the exact solution. In case of Sod's problem, the flux values need to be bounded by the cell averages of the neighbouring cells, which is done with the corrector 2 method. No additional upwind is required to solve this test case. In certain aspects, the solution is good, as the expansion fan is solved perfectly and discontinuities appear sharp. However, significant overshoots are found. Much worse is the fact that shock velocity is not correct. In several areas in the domain, the solution is off. The conservation properties are indeed locally conserved, these are linked to each other.
From Sod's shock tube problem, it was also found that refinement of the computational grid does not lead to convergence to the exact solution. Also found, was that the shock and the contact discontinuity are sources of error. The contact discontinuity results in errors as a pressure differential is present, which should not be possible. The shock wave is a source of error as the shock wave velocity is wrong and the Hugoniot jump conditions are not completely satisfied.
Test cases with higher pressure differences, resulting in greater velocity differences, proved to be a bigger challenge. For these test cases, additional stabilization was required, hence upwind was applied. However, with the upwind applied, the expansion fan became locally an expansion shock, which is physically impossible. With jumps closely near each other, the new scheme has severe troubles with solving correctly as well. This was to be expected, as the jumps require some cells to resolve, while the space between jumps has less cell in-between. However, since for these cases the location of the shock wave was again slightly lagging behind, it can be concluded that the shock handling needs improvement.
The test case with two shocks was solved poorly, the case with two expansion fans was not solvable with the method. From all test cases, it can be concluded that the current method used to use upwind stabilization is insufficient. In order to improve the method, better solution has to be found to solve high velocity shocks. Also required to improve the results is how to deal with pressure. Pressure is the forcing mechanism of shock tube problems, hence should be treated correctly.
Another notable issue is found in the timestep. For the scheme as discussed in Section 7.2, a rather small timestep is required. In the first test case, MUSCL can be used with a timestep twice as large. For test cases 3 to 5, the required timestep for the proposed method is 20 to 40 times smaller compared to the timestep for MUSCL.

8

Conclusions

In this research, special edge polynomials, called e-polynomials, were developed and used in a FE method. These polynomials are new in the sense that both point values as well as integrated values are used. Using this approach, a clear distinction between conserved part and the flux part can be made. Although originally the idea was to solve the integrated values in a FV way and the point values as a FE, schemes using that approach did not yield satisfactory results. Examples of these results can be found in Appendix B. When FE is used to solve both integrated values and point values, effective schemes were constructed.

The scheme using these polynomials have been tested on several equation sets and compared with a set of reference FV and/or FE methods. In general, the newly proposed methods compared well with the reference methods. In all cases, local conservation, and thus global conservation, is assured as long as no SUPG upwind method is used. The overall results show great potential for the proposed polynomials, but many improvements have to be done in order to have a successful method.

An extremely useful property of the new polynomials is that the polynomials associated with the cell face values have no integrated value. Due to this, fluxes can be easily corrected, which can be used as a simple technique to damp spurious oscillations without adding significant diffusion. For cases with more stringent stability requirements, one can add upwind on the fluxes, or use a SUPG upwind method on the complete set of polynomials. The SUPG upwind method was found to be non-conservative and thus should not be used in combination with the new edge polynomials. Usually upwind techniques have downsides and for the new method this is not any different. Most noticeable with upwind, is the numerical diffusion. In all tests in which upwind was used, numerical diffusion was found

From the linear advection and Burgers equations tests, it was found that the numerical diffusion of the scheme is insignificant. In linear advection, the numerical diffusion added, was only in the order of $\mathcal{O}\left(10^{-11}\right)$%, which is extremely small. In the linear case, the new method was consistently the best performing scheme of all the tested methods. Not only the numerical diffusion was the lowest found, but also overshoots and oscillations were the smallest and quickly damped. This combined with the fact that discontinuities in the solutions are solved in the least amount of cells, justifies the conclusion that for linear advection, the proposed method was the best.

In the inviscid Burgers' equation, the total $\int_\Omega \rho^2 d\Omega$ does change and therefore correction and/or upwind was required to stabilize. The method of stabilization greatly affects the found solution. SUPG was not conservative, but when bubble functions are used to achieve upwind, conservation is assured. Depending on the upwind parameter, oscillations could be suppressed effectively. However, if the upwind parameter β is taken to high, numerical diffusion starts to take its toll. For Burgers' equation, β between 0.1 and 0.2 is enough to find stable results. Correction methods using the property of cell face values not contributing to conservation yield better results regarding numerical diffusion. However, these corrector methods are unable to fully damp spurious oscillations. With correctors, error convergence is significantly higher compared to upwind methods. For the inviscid Burgers' equation, the new polynomials do not perform better compared to SUPG with piecewise linear polynomials. On the other hand, the for several stabilization methods the found performance does not perform worse.

In case of the Euler equations, the results are mixed. The Sod problem can be solved, but convergence is not in direction of the exact solution. For Sod's problem, the MUSCL scheme yields better results.

Cases with higher pressure differences became more challenging, but could be solved. For these cases both the MUSCL schemes as the FE method with e-polynomials have troubles. For these cases, additional upwind was required. It was found that upwind results in expansion shocks, which are physically impossible. The test case with two shocks required even more upwind and proved further that upwind should be avoided as much as possible. A test case which should result in two expansion could not be solved. With the current method, shock wave velocity is not yet computed correctly. Besides that, another source of error was found at the contact discontinuity. Although the velocity difference between both sides is insignificant, the pressure difference cannot be neglected.
A general conclusion can be made regarding the time step dependency. For proper results, the time step needs to be small, in many occasions a CFL number of 0.1 is used. For the Euler equations, time steps are even further decreased. Using a RK time integrator allows for larger time steps in several cases. Still, for linear advection the maximum CFL for which good results could be found is 0.6. One possible explanation for this, is that the polynomials are similar to three point piecewise cubic polynomials, hence a cell is virtually split in two.
Using e-polynomials can have significant benefits. A clear differentiation between conserved quantities and fluxes can be made. This research showed that numerical schemes can be constructed using such polynomials. This research also showed that for various problems, the found results are above all expectations. Some downsides were found, such as time step dependency and handling pressure in the Euler equation posed a challenge as well. With further research, these problems can be overcome.

9

Recommendations

The results up to now are satisfactory and it was concluded that this method has potential. Further research is required to improve results. For further research, this chapter provides a list of topics. Some topics are expected to be more difficult then others, but all are important to really determine the capabilities of this new method. The recommendations are as follows:

- Expansion in multiple dimensions. In order to further develop the method, the scheme needs to be expanded to multiple dimensions. This is however, easier said then done. In one dimension, one only need to account for lines and points. If necessary, values of points can be corrected when these get non-physical values. In higher dimension however, one has to account for volumes and surfaces as well. If for example, a two-dimensional space is considered, so no volumes, surfaces will be updated using the surrounding lines. The points on the corners of each surface can be updated using standard Galerkin FE approach, as done in this research. Updating the lines, is however a tedious job.
- Develop a time integrator matching the performance of the spatial discretization. One of the conclusions is that performance of the discretization is extremely depended on time step. Possible solution to increase the performance is to stager cell faces and cell integral values in time. This could mean: solve the cell faces on time level $n + \frac{1}{2}$, the integrated values on n and $n + 1$. This can be done by changing the system of equations such that the integrated values are on one side of the equation, the cell faces on the other side of the equal sign.
  Another idea which might improve the time step accuracy, is to develop a mimetic third order time integrator. The current spatial discretization is a second order polynomial, hence third order accurate discretization. The time integrators used in this research are the Crank-Nicolson scheme and the RK4 scheme. Crank-Nicolson is a second order scheme, the RK4 method is fourth order. However, if a scheme could be constructed which solves the fluxes on n, $n + \frac{1}{2}$ and $n + 1$, the polynomial in time would be similar to the polynomial in space. With this and the discrete Lie derivative, an accurate integration from n to $n + 1$ could be done.
- Improve performance of the scheme for the Euler equations. This probably requires improvement in determination of the velocity field and/or how the pressure terms are dealt with. Some of the shock tube problems resulted in unwanted flow reversal or negative pressure. Especially the negative pressure immediately blows up the solution.
- Expand the correction methods. The corrector 1 ensures that the solution found for the cell faces comes from the solution set available in the upwind cell. Corrector 2 is bit more stringent, as it ensures the cell face solution lies between the averages found of the cells the face lies in between. An option to investigate is to use corrections based on characteristics. This might be an effective solution, especially for the compressible Euler equations, since exact solutions for the compressible Euler equations are found using characteristics.
- Stability needs to be improved. Some of the test cases for the shock tube problem could not be solved, due to stability issues. This issue might be solved if a fitting time integration scheme is found, or when an improved corrector scheme is developed.

- Include viscosity terms. The Euler equations were solved with reasonable, although far from perfect, results. To fully solve fluid problems, viscosity needs to be included. This will lead to the full Navier-Stokes equations. It is advised to follow the development of AF methods [16, 17, 36], since these methods are closely related, but are FV oriented. At this point, the AF methods can solve similar advection problems as presented in this thesis. One can expect that these methods are further developed.
- Investigation on higher order expansion. Higher order methods get significant amount of attention, it would be useful if expanded polynomials can be derived using the same principles as discussed in Chapter 4. In order to achieve higher order reconstruction, new edge polynomials have to be derived. This might be possible from higher order Hermite polynomials, which would give edge polynomials with compact support. Advantages of methods with compact support, are that these have less stringent mesh requirements. Other options might be to imply the conditions of Hermite polynomials on polynomials spanning multiple cells. This would result in a wider support. A simple example for wider support higher order e-polynomials is given in Appendix A Section A.3.

A

Derivation of the Polynomials

In this appendix, the derivation of the polynomials is elaborated upon. First, the complete cubic Hermite polynomials are shown and how to derive these. Second, the e-polynomials as given in (4.6) are shown.

A.1. Cubic Hermite Polynomials

On a domain $\xi \in [-1, 1]$, cubic Hermite polynomials have the following properties:

$$\begin{aligned}
h_0^0(-1) &= 1, & h_1^0(-1) &= 0, & h_0^1(-1) &= 0, & h_1^1(-1) &= 0,\\
h_0^0(1) &= 0, & h_1^0(1) &= 1, & h_0^1(1) &= 0, & h_1^1(1) &= 0,\\
\frac{\partial}{\partial x} h_0^0(-1) &= 0, & \frac{\partial}{\partial x} h_1^0(-1) &= 0, & \frac{\partial}{\partial x} h_0^1(-1) &= 1, & \frac{\partial}{\partial x} h_1^1(-1) &= 0,\\
\frac{\partial}{\partial x} h_0^0(1) &= 0, & \frac{\partial}{\partial x} h_1^0(1) &= 0, & \frac{\partial}{\partial x} h_0^1(1) &= 0, & \frac{\partial}{\partial x} h_1^1(1) &= 1.
\end{aligned} \tag{A.1}$$

This set of conditions can be satisfied using third order functions:

$$h_i^j(\xi) = a\xi^3 + b\xi^2 + c\xi + d. \tag{A.2}$$

Differentiation with respect to ξ gives:

$$\frac{\partial h_i^j(\xi)}{\partial \xi} = 3a\xi^2 + 2b\xi + c \tag{A.3}$$

For each of the four polynomials, a matrix can be created to solve the constants a, b, c and d:

$$\begin{aligned}
h_0^0: &\quad \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 3 & -2 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad h_1^0: \quad \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 3 & -2 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix},\\
h_0^1: &\quad \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 3 & -2 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \qquad h_1^1: \quad \begin{bmatrix} -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 3 & -2 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.
\end{aligned} \tag{A.4}$$

When these systems are solved, the h-polynomials are found:

$$h_0^0(\xi) = \frac{1}{4}\left(\xi^3 - 3\xi + 2\right), \tag{A.5a}$$

$$h_1^0(\xi) = \frac{1}{4}\left(-\xi^3 + 3\xi + 2\right), \tag{A.5b}$$

$$h_0^1(\xi) = \frac{1}{4}\left(\xi^3 - \xi^2 - \xi + 1\right), \tag{A.5c}$$

$$h_1^1(\xi) = \frac{1}{4}\left(\xi^3 + \xi^2 - \xi - 1\right). \tag{A.5d}$$

Using this approach, higher order Hermite polynomials can be easily created. A set of fifth order polynomials can be created by setting a second derivative. However, if one seeks to find higher order e-polynomials, this approach would probably not give the desired results. A better options is to divide the cell in subcells. This would give a requirement on $\xi = 0$ or $\xi \pm \frac{1}{3}$ with 2 or 3 subcells respectively. An example of this is shown in Section A.3 of this appendix

A.2. e-Polynomials

As stated in (4.2), de e-polynomials are difined as:

$$e_0(\xi) = dh_0^1(\xi), \qquad e_1(\xi) = dh_1^0(\xi) = -dh_0^0(\xi), \qquad e_2(\xi) = dh_1^1(\xi). \tag{A.6}$$

When (A.5) are inserted into these relations, the following equations is found for e_0

$$\begin{aligned} e_0(\xi) &= dh_0^1(\xi), \\ &= \frac{\partial}{\partial x}\frac{1}{4}(\xi^3 - \xi^2 - \xi + 1), \\ &= \frac{1}{4}(3\xi^2 - 2\xi - 1), \\ &= \frac{3}{4}(\xi+1)^2 - 2(\xi+1) + 1 \end{aligned} \tag{A.7a}$$

Similarly for e_1:

$$\begin{aligned} e_1(\xi) &= dh_1^0(\xi), \\ &= \frac{\partial}{\partial x}\frac{1}{4}(\xi^3 - 3\xi + 2), \\ &= \frac{1}{4}(3\xi^2 - 3), \\ &= \frac{3}{4}(\xi+1)(1-\xi). \end{aligned} \tag{A.7b}$$

and for e_2:

$$\begin{aligned} e_2(\xi) &= dh_1^1(\xi), \\ &= \frac{\partial}{\partial x}\frac{1}{4}(\xi^3 + \xi^2 - \xi - 1), \\ &= \frac{1}{4}(3\xi^2 + 2\xi - 1), \\ &= \frac{3}{4}(1-\xi)^2 - 2(1-\xi) + 1 \end{aligned} \tag{A.7c}$$

A.3. Higher Order with Wider Support

An expansion to higher order can be achieved by expanding the polynomials to span several cells. When this is done over two cells for instance, the list of conditions of A.1 has to be expanded to include the following conditions:

$$h_{0,1}^{0,1}(0) = 0, \qquad dh_{0,1}^{0,1}(0) = 0. \tag{A.8}$$

Also, two additional polynomials have to be added:

$$\begin{aligned} h_2^0(-1) &= 0, & h_2^1(-1) &= 0, \\ h_2^0(0) &= 1, & h_2^1(0) &= 0, \\ h_2^0(1) &= 0, & h_2^1(1) &= 0, \\ \frac{\partial}{\partial x}h_2^0(-1) &= 0, & \frac{\partial}{\partial x}h_2^1(-1) &= 0, \\ \frac{\partial}{\partial x}h_2^0(0) &=, & \frac{\partial}{\partial x}h_2^1(0) &= 1, \\ \frac{\partial}{\partial x}h_2^0(1) &= 0, & \frac{\partial}{\partial x}h_2^1(1) &= 0. \end{aligned} \tag{A.9}$$

To span two cells, a fifth order function is required:

$$h_i^j(\xi) = a\xi^5 + b\xi^4 + c\xi^3 + d\xi^2 + e\xi + f. \tag{A.10}$$

And differentiation with respect to ξ gives:

$$\frac{\partial h_i^j(\xi)}{\partial \xi} = 5a\xi^4 + 4b\xi^3 + 3c\xi^2 + 2d\xi + e. \tag{A.11}$$

Just as done in (A.4), the constants can be solved. The following equations can be found:

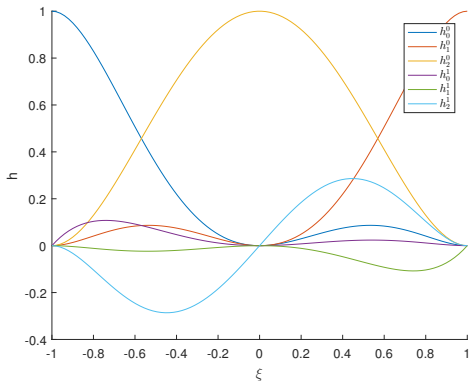
$$\begin{aligned}
h_0^0(\xi) &= \frac{1}{4}\left(3\xi^5 - 2\xi^4 - 5\xi^3 + 4\xi^2\right) \\
h_1^0(\xi) &= \frac{1}{4}\left(-3\xi^5 - 3\xi^4 + 5\xi^3 + 4\xi^2\right) \\
h_2^0(\xi) &= \frac{1}{4}\left(4\xi^4 - 8\xi^2 + 4\right) \\
h_0^1(\xi) &= \frac{1}{4}\left(\xi^5 - \xi^4 - \xi^3 + \xi^2\right) \\
h_0^1(\xi) &= \frac{1}{4}\left(\xi^5 + \xi^4 - \xi^3 - \xi^2\right) \\
h_2^1(\xi) &= \frac{1}{4}\left(4\xi^5 - 8\xi^3 + 4\xi\right)
\end{aligned} \tag{A.12}$$

These polynomials are shown in Figure A.1a. With these polynomials, extend e-polynomials can be derived:

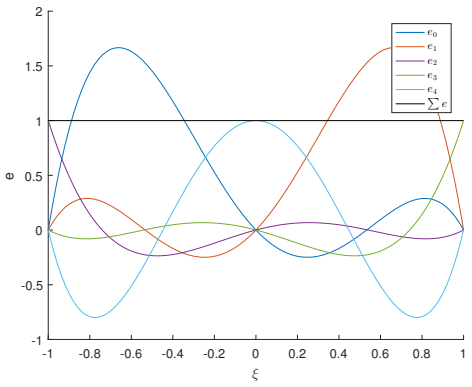
$$e_0(\xi) = -dh_0^0(\xi), \quad e_1(\xi) = dh_2^0(\xi), \quad e_2(\xi) = dh_0^1(\xi), \quad e_3(\xi) = dh_1^1(\xi), \quad e_4(\xi) = dh_2^1(\xi), \tag{A.13}$$

These polynomials can be found in Figure A.1b. The properties of these polynomials are:

$$\begin{aligned}
&e_0(-1) = 1, \quad e_0(0) = 0, \quad e_0(1) = 0, \quad \int_{-1}^{0} e_0(\xi) = 1, \quad \int_{0}^{1} e_0(\xi) = 0, \\
&e_1(-1) = 1, \quad e_1(0) = 0, \quad e_1(1) = 0, \quad \int_{-1}^{0} e_1(\xi) = 0, \quad \int_{0}^{1} e_1(\xi) = 1, \\
&e_2(-1) = 1, \quad e_2(0) = 0, \quad e_2(1) = 0, \quad \int_{-1}^{0} e_2(\xi) = 0, \quad \int_{0}^{1} e_2(\xi) = 0, \\
&e_3(-1) = 0, \quad e_3(0) = 0, \quad e_3(1) = 1, \quad \int_{-1}^{0} e_3(\xi) = 0, \quad \int_{0}^{1} e_3(\xi) = 0, \\
&e_4(-1) = 0, \quad e_4(0) = 1, \quad e_4(1) = 0, \quad \int_{-1}^{0} e_4(\xi) = 0, \quad \int_{0}^{1} e_4(\xi) = 0,
\end{aligned} \tag{A.14}$$



(a) Expansion of h-polynomials over 2 cells



(b) Expansion of e-polynomials over 2 cells

B

Original Idea and Results

The original idea of the e-polynomials, as discussed in Section 4.1 and derived in Appendix A, was to develop a AF scheme by combining FE and FV methods. In that approach, the integrated variable is treated as a a FV using the discrete Lie derivative. The values on the cell faces are updated using a FE approach. During the time this research was conducted, no satisfying results could be obtained with this idea. In this appendix, some of the results are shown which resulted in the change of direction to Galerkin. To start, using the discrete Lie derivative, one can find:

$$\begin{aligned}\int_{-1}^{1} \mathcal{L}_a \rho^{(1)}(\xi) = \int_{-1}^{1} di_a \rho^{(k)} &= \int_{-1}^{1} \rho_L \frac{\partial}{\partial x} a_L e_0(\xi) + c\rho_I \frac{\partial}{\partial x} a_I e_1(\xi) + \int_{-1}^{1} \rho_R \frac{\partial}{\partial x} a_R e_2(\xi), \\ &= \left(-\rho_L a_L + \rho_R a_R\right). \end{aligned} \tag{B.1}$$

When linear advection is considered for the purpose of this example, one would find

$$\begin{aligned}\int_{-1}^{1} \frac{\partial \rho}{\partial t} + \left(-\rho_L a_L + \rho_R a_R\right) = \\ \frac{\partial}{\partial t}\left[\int_{-1}^{1} \rho_L e_0\left(\xi\right) + \int_{-1}^{1} \rho_I e_1\left(\xi\right) + \int_{-1}^{1} \rho_R e_2\left(\xi\right)\right] + \left(-\rho_L a_L + \rho_R a_R\right) = \\ \frac{\partial \rho_I}{\partial t} + \left(-\rho_L a_L + \rho_R a_R\right) = 0. \end{aligned} \tag{B.2}$$

When Galerkin is applied on the fluxes, one would get a system of equations:

$$\mathbb{M}\frac{\partial \rho}{\partial t} + \mathbb{F}\rho = 0, \tag{B.3}$$

With

$$\mathbb{M} = \begin{bmatrix} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} e_0 e_0 dx & \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} e_0 e_1 dx & \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} e_0 e_2 dx \\ 0 & 1 & 0 \\ \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} e_2 e_0 dx & \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} e_2 e_1 dx & \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} e_2 e_2 dx \end{bmatrix}, \tag{B.4}$$

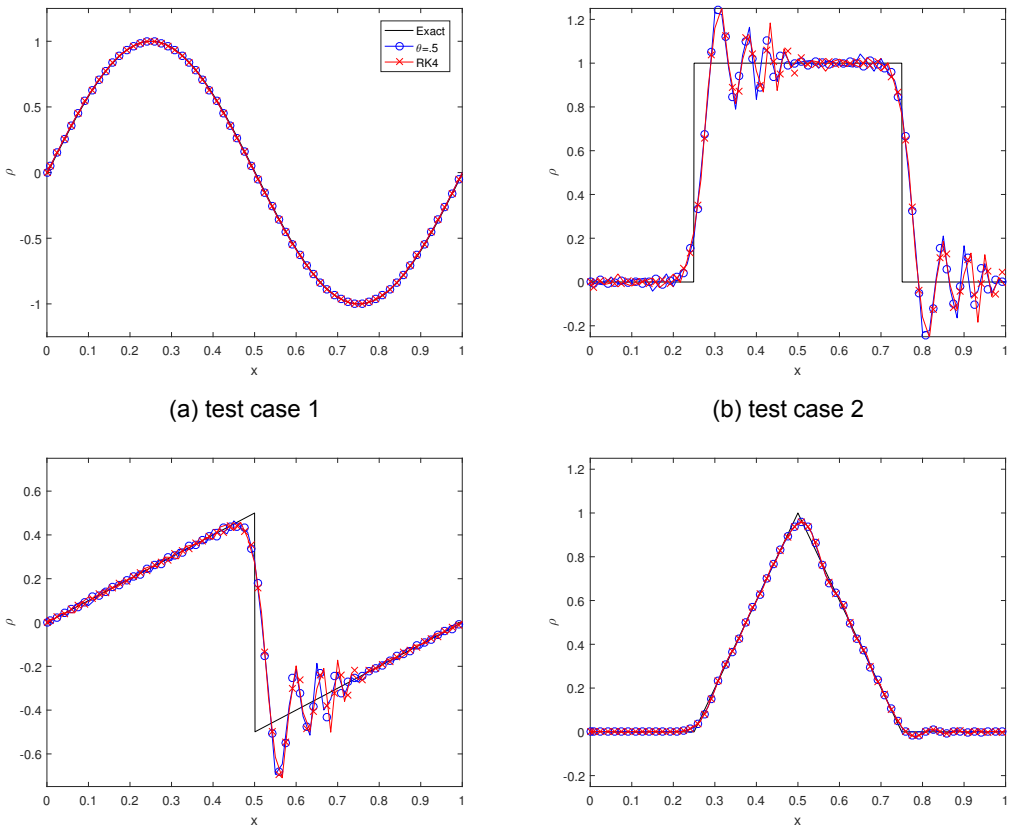
and for $\mathbb{F}$:

$$\mathbb{F} = \begin{bmatrix} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial a e_0}{\partial x} e_0 dx & \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial a e_0}{\partial x} e_1 dx & \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial a e_0}{\partial x} e_2 dx \\ -a & 0 & a \\ \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial a e_2}{\partial x} e_0 dx & \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial a e_2}{\partial x} e_1 dx & \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial a e_2}{\partial x} e_2 dx \end{bmatrix}. \tag{B.5}$$

The system of equations of B.3

B.1. Result on Linear Advection

The same test cases as introduced in Section 5.1 are used. This time, the matrices (B.4) and (B.5) are used to solve system of equations B.3. In Figures B.1, the results with both a Crank-Nicolson and a RK4 time scheme. The Crank-Nicolson time integrator is plotted in blue with circle markers, the RK4 time integrator in red with crosses. For these figures, 60 cells are used, with a CFL of 0.1 and the initial condtions are run once trough the domain.
The results on test case 1, being a smooth problem are close to exact, as can be seen in Figure B.1a. The results of the test cases with a discontinuity in the problem, test cases 2 and 3, are poor. Large overshoots are present in the solution and the discontinuity is far from sharp. In Figure B.1d, the results of test case 4 are shown. These last results are close to the exact solution, although small wiggles are present.



(a) test case 1

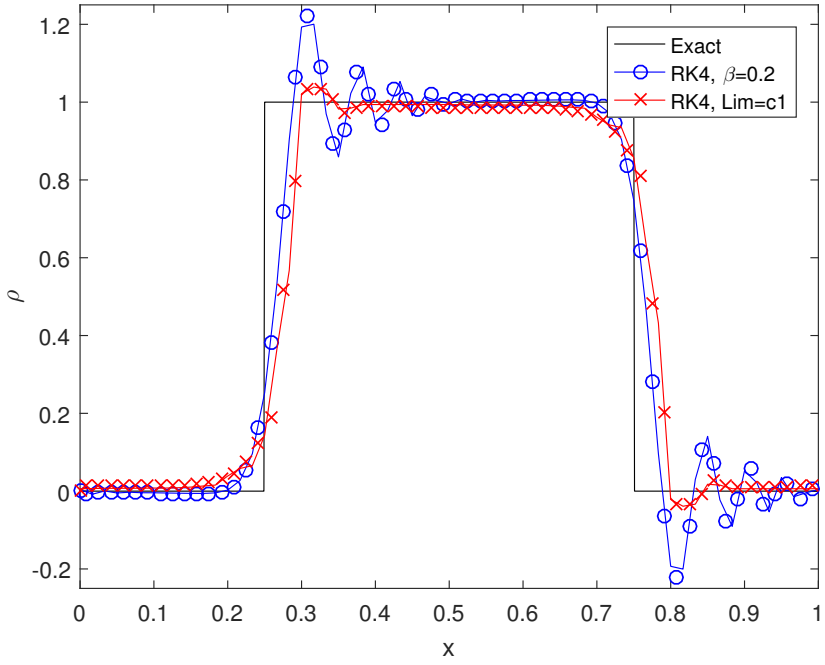
(b) test case 2

(c) test case 3

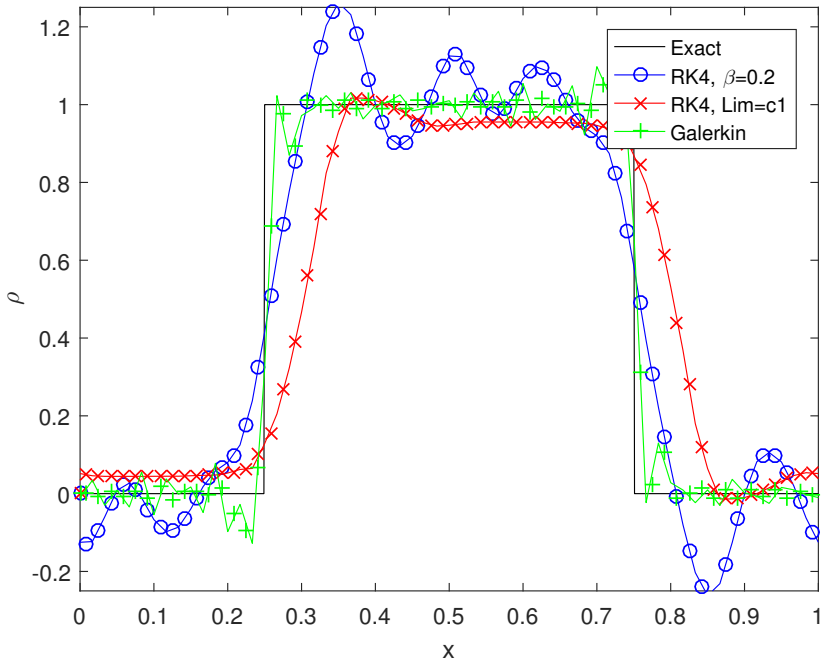
(d) test case 4

Figure B.1: Results for original idea, with 60 cells, CFL of 1 and 1 cycle trough the domain.

For this idea, both upwind with bubble functions on the left and right polynomial, as well as the corrector 1 scheme have been tested. In Figure B.2 the results of these techniques for the second test case are shown. The upwind approach does not seem to improve the results. The corrector 1 approach does reduce oscillations and overshoot height. However, with the corrector in play, the correct solution value between discontinuities is not reached. In Figure B.2b, the simulation is run for 10 cycles through the domain. In this figure, the base results as discussed in Section 5.2 are plotted as well. From this figure, it becomes even more clear that the original idea does not yield as good results as the scheme used in the rest of this thesis. With upwind, the results have become a series of oscillations. The corrected scheme seems to move slightly faster than the advective velocity and results deviate significantly from the exact results.



(a) After 1 cycle



(b) After 10 cycles

Figure B.2: Results for test case 2, with stabilization applied. 60 cells, CFL of 1.

B.2. Conclusions

Compared to the results found when Galerkin is used for all variables, the original idea underperformed significantly. These results resulted in the decision to use Galerkin for all both the integrated cell values and the point values on the cell faces.

C

Runge-Kutta Time Integration

Runge-Kutta (RK) schemes are multi-step iterative methods, which can be implicit and/or explicit. Using multiple steps, a higher order time integration can be achieved, which is self starting since no values of previous time levels are required.

C.1. Basics of Runge-Kutta Schemes

The final time step of a Runge-Kutta scheme is computed as:

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i k_i, \tag{C.1}$$

in which h is the final step and s is the number of stages. For an explicit scheme, each sub-stage is computed as:

$$k_i = f\left(t_n + c_i h, y_n + h \sum_{j=1}^{i} a_{i,j} k_j\right) \tag{C.2}$$

In case an implicit RK method is used, this equation changes to:

$$k_i = f\left(t_n + c_i h, y_n + h \sum_{j=1}^{s} a_{i,j} k_j\right) \tag{C.3}$$

In these (C.2) and (C.3), c_i is a measure for the intermediate level a function evaluation is performed. $a_{i,j}$ is a weight for the contribution of intermediate level j to level i. In that respect $a_{i,j}$ and b_i are extremely similar, as the values b_i are weights of intermediate level i to the final solution at $n+1$. Values for a, b and c are taken from the so called Butcher tableau. A Butcher tableau has the following lay-out:

$$\begin{array}{c|cccc} c_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,s} \\ c_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} . \tag{C.4}$$

C.2. Runge-Kutta 4 Butcher Tableau

The Butcher tableau for the often used RK4 is:

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} . \tag{C.5}$$

Bibliography

[1] Jd Anderson Jr. *Fundamentals of Aerodynamics*, volume Fourth. McGraw-Hilll, 2007. ISBN 0071254080.

[2] V.I Arnold. *Mathematical methods of classical mechanics*. Springer-Verlag New York, 2 edition, 1989. ISBN 978-1-4757-2063-1. doi: 10.1007/978-1-4757-2063-1.

[3] Peter Bakker and Bram van Leer. Lecture notes on Gasdynamics, AE4-140. Technical report, Tu Delft, 2005.

[4] Richard M. Beam and R. F. Warming. An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *Journal of Computational Physics*, 22(1):87–110, 1976. ISSN 10902716. doi: 10.1016/0021-9991(76)90110-8.

[5] Lourenco Beirao da Veiga, Konstantin Lipnikov, and Gianmarco Manzini. *The Mimetic Finite Difference Method for Elliptic Problems*, volume 11. 2014. ISBN 9783319026626.

[6] William M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. 2003. ISBN 0121160505. doi: 10.1016/S0079-8169(08)61020-X.

[7] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*, volume 15. 2008. ISBN 978-0-387-75933-3. doi: 10.1007/978-0-387-75934-0.

[8] Alexander N. Brooks and Thomas J R Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1-3):199–259, 1982. ISSN 00457825. doi: 10.1016/0045-7825(82)90071-8.

[9] A. Burbeau, P. Sagaut, and Ch.-H. Bruneau. A Problem-Independent Limiter for High-Order Runge–Kutta Discontinuous Galerkin Methods. *Journal of Computational Physics*, 169(1):111–150, 2001. ISSN 00219991. doi: 10.1006/jcph.2001.6718.

[10] B. Cockburn and C.-W. Shu. The Runge-Kutta Local Projection P1- Discontinuous Galerkin Method for Scalar Conservation Laws, 1991.

[11] Bernardo Cockburn, Chi-Wang Shu, San-Yih Lin, and Chi-Wang Shu. TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws II. General Framework. *Mathematics of Computation*, 52(186):411–435, 1988. ISSN 0025-5718. doi: 10.1090/S0025-5718-1989-0983311-4.

[12] Bernardo Cockburn, San Yih Lin, and Chi Wang Shu. TVB runge-kutta local projection discontinuous galerkin finite element method for conservation laws III: One-dimensional systems. *Journal of Computational Physics*, 84(1):90–113, 1989. ISSN 10902716. doi: 10.1016/0021-9991(89)90183-6.

[13] Bernardo Cockburn, Suchung Hou, and Chi-Wang Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV: The multidimensional case. *Mathematics of Computation*, 54(190):545–581, 1990. ISSN 0025-5718. doi: 10.1090/S0025-5718-1990-1010597-0.

[14] Mathieu Desbrun, Eva Kanso, and Yiying Tong. Discrete differential forms for computational modeling. *ACM SIGGRAPH 2006 Courses on SIGGRAPH 06*, 38:39, 2006. doi: 10.1145/1185657.1185665.

[15] Bjorn Engquist and Stanley Osher. One-Sided Difference Approximations for Nonlinear Conservation Laws. *Mathematics of Computation*, 36(154):321, 1981. ISSN 00255718. doi: 10.2307/2007646.

[16] T A Eymann and P L Roe. Active Flux Schemes for Systems. *20th AIAA Computational Fluid Dynamics Conference, 27 - 30 June, Honolulu, Hawaii. AIAA Paper 2011-3840*, (June):1–11, 2011.

[17] Timothy Andrew Eymann and Philip L. Roe. Multidimensional Active Flux Schemes. *21st AIAA Computational Fluid Dynamics Conference*, pages 1–24, 2013. doi: 10.2514/6.2013-2940. URL http://arc.aiaa.org/doi/abs/10.2514/6.2013-2940.

[18] Theodore Frankel. *The Geometry of Physics*. Cambridge University Press, 2 edition. ISBN 9780521833301.

[19] Marc Gerritsma, Jeroen Kunnen, and Boudewijn de Heij. *Discrete Lie Derivative*, pages 635–643. Springer International Publishing, Cham, 2016. ISBN 978-3-319-39929-4. doi: 10.1007/978-3-319-39929-4_61. URL http://dx.doi.org/10.1007/978-3-319-39929-4{_}61.

[20] Sergei K. Godunov. A Difference Scheme for Numerical Solution of Discontinuous Solution of Hydrodynamic Equations. *Math. Sbornik*, 47:357–393, 1959.

[21] Sarah Hank. A Hyperbolic Eulerian Model for Dilute Two-Phase Suspensions. *Journal of Modern Physics*, 02(September):997–1011, 2011. ISSN 2153-1196. doi: 10.4236/jmp.2011.29120.

[22] A. Harten, P. Lax, and B. van Leer. On upstream differencing and Godunov type methods for hyperbolic conservation laws. *SIAM review*, 25(1):35–61, 1983.

[23] Ami Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49(3):357–393, 1983. ISSN 10902716. doi: 10.1016/0021-9991(83)90136-5.

[24] Thomas J R Hughes, Guglielmo Scovazzi, and Tayfun E. Tezduyar. Stabilized methods for compressible flows. In *Journal of Scientific Computing*, volume 43, pages 343–368, 2010. ISBN 0885-7474. doi: 10.1007/s10915-008-9233-5.

[25] T.J.R. Hughes, L.P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Computer Methods in Applied Mechanics and Engineering*, 54(2):223–234, 1986. ISSN 00457825. doi: 10.1016/0045-7825(86)90127-1.

[26] Bo-Nan Jiang. *The Least-Squares Finite Element Method, Theory and Applications in Computational Fluid Dynamics and Electromagnetics*. Springer-Verlag Berlin Heidelberg, 1 edition, 1998. ISBN 3-540-63934-9.

[27] C Kong. *Comparison of Approximate Riemann Solvers*. PhD thesis, 2011. URL http://wap.rdg.ac.uk/web/FILES/maths/CKong-riemann.pdf.

[28] Edward M. Kraft. After 40 Years Why Hasn't the Computer Replaced the Wind Tunnel? *ITEA Journal*, 31:329–346, 2010. URL http://www.dtic.mil/dtic/tr/fulltext/u2/a530355.pdf.

[29] Lilia Krivodonova. Limiters for high-order discontinuous Galerkin methods. *Journal of Computational Physics*, 226(1):879–896, 2007. ISSN 00219991. doi: 10.1016/j.jcp.2007.05.011.

[30] Culbert B. Laney. *Computational Gasdynamics*. Cambridge University Press, 1 edition, 1998. ISBN 978-0521625586.

[31] Meng-Sing Liou and Christopher J. Steffen. A new flux splitting scheme. *Journal of Computational Physics*, 107:23–39, 1993. ISSN 00219991. doi: 10.1006/jcph.1993.1122.

[32] Hassan Mansour Mansour and M. A. Bashir. A Differential Geometric Approach to Fluid Mechanics. *International Journal of Scientific and Research Publication*, 5(9):1–6, 2015.

[33] P. Mullen, A. McKenzie, D. Pavlov, L. Durant, Y. Tong, E. Kanso, J. E. Marsden, and M. Desbrun. Discrete Lie Advection of Differential Forms. *Foundations of Computational Mathematics*, 11(2): 131–149, 2011. ISSN 16153375. doi: 10.1007/s10208-010-9076-y.

[34] J. Naber. Building your own shock tube. Technical report, Centrum voor Wiskunde en Informatica, 2005. URL http://ftp.cwi.nl/CWIreports/MAS/MAS-E0705.pdf.

[35] NASA. Wave Clouds Near Amsterdam Island, 2005. URL http://earthobservatory.nasa.gov/IOTD/view.php?id=6151.

[36] H. Nishikawa. Active flux schemes for advection diffusion. *22nd AIAA Computational Fluid Dynamics Conference*, (June):1–11, 2015. doi: 10.2514/6.2015-2450.

[37] Hiroaki Nishikawa and Philip L. Roe. Third-order active-flux scheme for advection diffusion: Hyperbolic diffusion, boundary condition, and Newton solver. *Computers and Fluids*, 125:71–81, 2016. ISSN 00457930. doi: 10.1016/j.compfluid.2015.10.020. URL http://dx.doi.org/10.1016/j.compfluid.2015.10.020.

[38] W. H. Reed and T. R. Hill. Triangular Mesh Methods for the Neutron Transport Equation. *Los Alamos Report LA-UR-73-479*, (836):10, 1973.

[39] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981. ISSN 10902716. doi: 10.1016/0021-9991(81)90128-5.

[40] Chi-Wang Shu. Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws. *Icase*, (97):1–79, 1997. doi: 10.1007/BFb0096351.

[41] Gary A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, 1978. ISSN 10902716.

[42] Michael Stone and Paul Goldbart. Mathematics For Physics. *Book*, 67(1):1–919, 2011. doi: 10.1017/CBO9780511627040. URL http://usir.salford.ac.uk/16202/.

[43] P. K. Sweby. High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws. *SIAM Journal on Numerical Analysis*, 21(5):995–1011, 1984. ISSN 0036-1429. doi: 10.1137/0721062.

[44] Tayfun E. Tezduyar and Thomas J R Hughes. Finite Element Formulations for Convection Dominated Flows With Particular Emphasis on the Compressible Euler Equations. *AIAA paper*, 1983.

[45] Tayfun E Tezduyar and Yasuo Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190: 411–430, 2000. ISSN 00457825. doi: 10.1016/S0045-7825(00)00211-5.

[46] E F Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. 2009. ISBN 9783540252023. doi: 10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C.

[47] Loring W. Tu. *An Introduction to Manifolds*. Springer, 2 edition, 2011. ISBN 9781441973993. doi: 10.1007/978-1-4419-7400-6.

[48] Bram Van Leer. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *Journal of Computational Physics*, 23(3):276–299, 1977. ISSN 10902716. doi: 10.1016/0021-9991(77)90095-X.

[49] Bram van Leer. Upwind and High-Resolution Methods for Compressible Flow : From Donor Cell to Residual-Distribution Schemes. *Communications in Computational Physics*, 1(2):192–206, 2006. doi: 10.2514/6.2003-3559.

[50] Paul Woodward and Phillip Colella. The numerical simulation of two-dimensional fluid flow with strong shocks, 1984. ISSN 10902716.

[51] Xinghui Zhong and Chi Wang Shu. A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods. *Journal of Computational Physics*, 232(1):397–415, 2013. ISSN 00219991. doi: 10.1016/j.jcp.2012.08.028.