High–Order WENO Finite Volume Methods on Cartesian Grids with Adaptive Mesh Refinement

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Pawel Buchmueller
aus Karasuk

Düsseldorf, Oktober 2016

aus dem Mathematischen Institut
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent:	Prof. Dr. Christiane Helzel Heinrich-Heine-Universität Düsseldorf
Korreferent:	Prof. Dr. James Rossmanith Iowa State University

Tag der mündlichen Prüfung: 15.12.2016

Zusammenfassung

Will man ein WENO Finite–Volumen–Verfahren vom eindimensionalen Fall auf den mehrdimensionalen Fall übertragen, so ist die dimensionsweise Anwendung des Verfahren die einfachste Möglichkeit. Bei nichtlinearen Problemen ist das resultierende Verfahren jedoch höchstens von zweiter Ordnung.

In dieser Arbeit wird eine Möglichkeit vorgestellt, diese Beschränkung zu überwinden. Dazu werden Formeln hergeleitet, die es erlauben Mittelwerte in Punktwerte ohne Ordnungsverlust umzurechnen. Mit den Punktwerten kann nun der Fluss im Zentrum der Zellkante berechnet werden. Für ein Finite–Volume–Verfahren wird jedoch der Mittelwert des Flusses über die Kante benötigt. Daher erfolgt anschließend die Umrechnung der Punktwerte in Mittelwerte. Das Verfahren wird detailliert für Erhaltungsgleichungen in Divergenzform beschrieben. Dabei wird sowohl der räumlich zweidimensionale als auch der dreidimensionale Fall betrachtet. Des Weiteren wird ein entsprechendes Verfahren für hyperbolische Systeme, die nicht in Divergenzform vorliegen, entwickelt.

Die Anwendung des Verfahrens ist beschränkt auf kartesische Gitter. Adaptive Gitterverfeinerung (AMR) erhöht jedoch die Flexibilität der Gitterstruktur. Dazu wird das Rechengebiet in Blöcke aufgeteilt und jeder Block wird durch ein kartesisches Gitter diskretisiert.

Das entwickelte Verfahren wird bezüglich Aufwand und Genauigkeit, sowohl mit dem klassischen dimensionsweisen Ansatz als auch mit einem Verfahren, das auf mehrdimensionale Rekonstruktion basiert, verglichen. Mit deutlich geringerem Rechenaufwand erreicht das vorgestellte Verfahren eine Genauigkeit, die vergleichbar ist mit der Genauigkeit der mehrdimensionalen Rekonstruktion. In zahlreichen numerischen Rechnungen werden die Euler–Gleichungen der Gasdynamik und die Gleichungen der Magnetohydrodynamik betrachtet.

Die hier präsentierten Resultate sind zu großen Teilen in den zwei bereits erschienenen Publikationen [1] und [2] zu finden.

Abstract

We present a WENO–based finite volume method for the approximation of hyperbolic conservation laws on adaptively refined Cartesian grids. These retain the high spatial order of accuracy of the one–dimensional discretization when applied to nonlinear multidimensional systems of conservation laws.

We derive formulas that allow us to compute high–order accurate point values of the conserved quantities at grid cell interfaces. Using those point values, we compute a high–order flux at the center of a grid cell interface. Those point values are then used to compute high–order accurate averaged fluxes at cell interfaces, as needed by a finite volume method. The method is described in detail for conservation laws in divergence form in 2D as well as 3D. Furthermore, the method is extended to WENO–type methods for hyperbolic systems that are not in divergence form.

The method is restricted to Cartesian grids. With AMR, we add more flexibility to the grid structure. On each single patch the grid is still Cartesian. So, the method can be applied as it is.

Numerical tests illustrate the accuracy of the new adaptive WENO finite volume method. The method is compared with both, the classical dimension–by–dimension approach as well as the multidimensional reconstruction where a high–order quadrature formula is used to compute the fluxes. The accuracy of the multidimensional reconstruction is comparable with the new method, while it is three times more expensive in 2D and even nine times more expensive in 3D.

Most of the main results presented in this thesis can also be found in two previous publications: [1] and [2].

Contents

Chapter 1

Introduction

High–order WENO (i.e. weighted, essentially non-oscillatory) methods are widely used for the approximation of hyperbolic problems. See, for example, the recent review of Shu [3]. The simplest way to use WENO methods on multidimensional Cartesian grids consists in applying a one–dimensional WENO scheme in each direction. This spatial discretization is typically combined with a Runge–Kutta method in time, i.e. during each stage of the Runge–Kutta method, one–dimensional WENO schemes are used in a dimension–by–dimension fashion.

On uniform Cartesian grids, conservative finite difference WENO methods based on flux interpolation, as introduced by Shu and Osher [4, 5], are very efficient. These finite difference methods are conservative and retain the accuracy of the one–dimensional WENO method in the multidimensional case, both for linear as well as nonlinear problems. An extension to smoothly varying mapped grids is possible (see [6]). In contrast to this, finite volume WENO methods based on a dimension–by–dimension approach retain the full order of accuracy for smooth solutions of linear multidimensional problems but are only 2^{nd}–order accurate for smooth solutions of nonlinear problems (see [3, 7]).

Here, we restrict our focus to finite volume WENO methods. For hyperbolic equations in divergence form, an advantage of finite volume methods is that they approximate the integral form of a conservation law that remains valid at discontinuities, where the differential form of the equation is not valid in the classical sense. In general, it is straightforward to extend finite volume methods to unstructured grids. The dimension–by–dimension approach limits the focus to Cartesian grids, but not necessarily to equidistant Cartesian grids. To avoid loss of accuracy for finite volume WENO methods, one can replace the simple dimension–by–dimension approach using a multidimensional reconstruction of the conserved quantities and use a high–order quadrature

formula for the flux computation. Such methods were used on unstructured and structured grids in [8, 9, 7, 10]. However, such high–order finite volume methods are much more expensive than the dimension–by–dimension approach, as will be shown in the various examples.

We present here a simple modification of finite volume WENO methods, which also leads to the full spatial order of accuracy by using only one–dimensional polynomial reconstructions in a dimension–by–dimension approach. While WENO reconstruction is typically of an odd order (here we consider methods of order five and seven), the corrections introduced in this thesis lead to fluxes of even order (here we present the formulas for order four and six). For the temporal discretization, we use explicit Runge–Kutta methods of order four, five, or seven. The crucial step is a transformation between face–averaged values and point values of the conserved quantities and the interface fluxes. This makes our method only slightly more expensive compared to a dimension–by–dimension finite volume WENO method or a finite difference WENO method. Such a transformation has also been used in the recently proposed 4^{th}–order accurate finite volume method of McCorquodale and Colella [11].

We also extend the modified finite volume WENO method to grids with adaptive mesh refinement. For the implementation we used the parallel, mesh adaptive framework *Racoon*, developed by Dreher and Grauer [12]. The grid is split into blocks and each block can be refined by a regular bisection. In this way we increase the flexibility of the grid structure significantly while local it remains Cartesian, so the new method can be applied as it is.

On uniform as well as on adaptively refined grids, the new method is only slightly more expensive than the dimension–by–dimension approach, but it produces results comparable with those obtained using multidimensional reconstruction.

Most of the results presented in this thesis have been published by Buchmüller and Helzel in [1] and Buchmüller, Dreher, and Helzel in [2]. In particular, parts of the introduction and several sections are adapted from [1] or [2]. The corresponding sections are indicated by footnotes. My contributions to these publications is outlined in Appendix C.

Chapter 2

Motivation for the use of high–order accurate methods

To illustrate the need for high–order accurate methods, we consider the equations of linear acoustics in two space dimensions, given by

$$\begin{pmatrix} p \\ u \\ v \end{pmatrix}_t + \begin{pmatrix} 0 & K_0 & 0 \\ \frac{1}{\rho_0} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} p \\ u \\ v \end{pmatrix}_x + \begin{pmatrix} 0 & 0 & K_0 \\ 0 & 0 & 0 \\ \frac{1}{\rho_0} & 0 & 0 \end{pmatrix} \begin{pmatrix} p \\ u \\ v \end{pmatrix}_y = 0, \quad (2.1)$$

where K_0 is the bulk modulus of compressibility [13]. This first example was suggested by R.J. LeVeque in a private email.

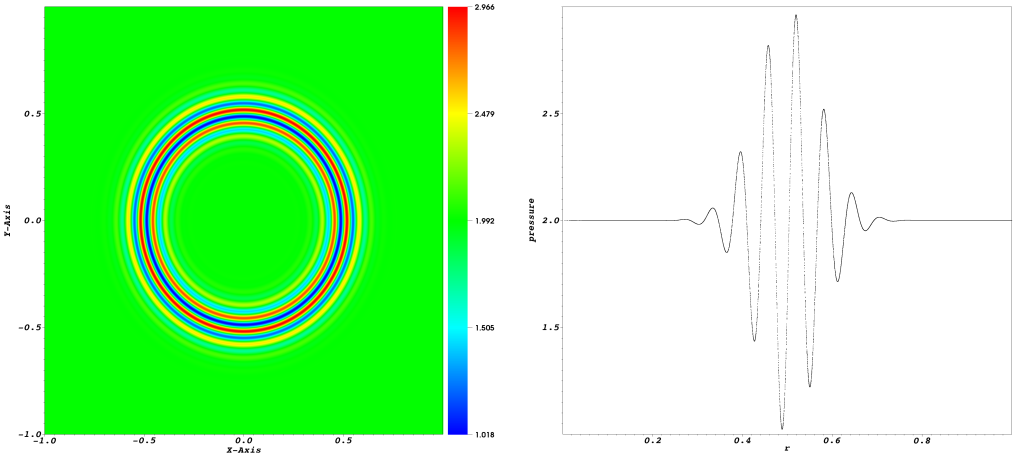
Example 2.1. *We consider the solution of* (2.1) *on the domain* $[-1,1]^2$. *The initial data is given by*

$$\begin{aligned} p(x,y,0) &= 2 + e^{-100(r-0.5)^2} \sin(100r), \\ u(x,y,0) &= v(x,y,0) = 0, \end{aligned}$$

where $r = \sqrt{x^2+y^2}$. *We set* $K_0 = 4$ *and* $\rho_0 = 1$.

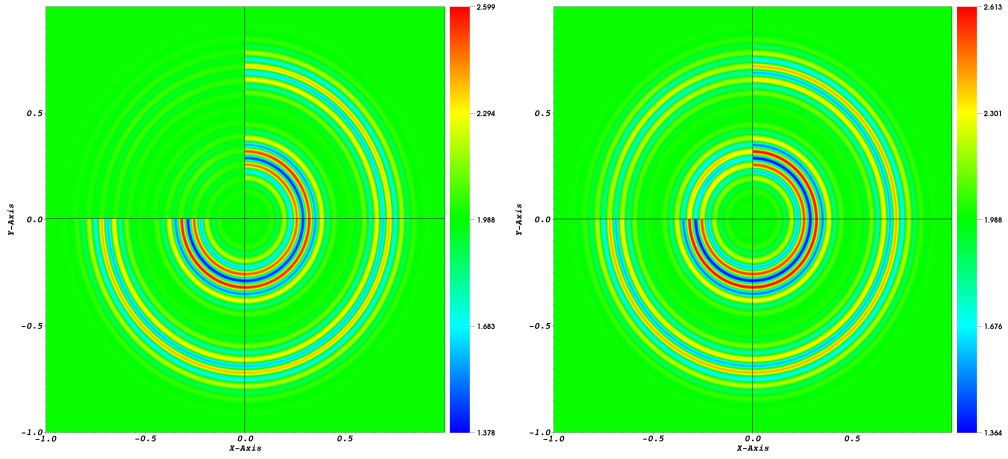
At final time $T = 0.1$, we compare the results with a reference solution computed on a very fine grid. The initial data, shown in Figure 2.1, consists of a wave package located around 0.5 units from the origin. With time, the initial wave package separates in two similar packages moving in opposite directions: one moving toward the origin and one moving away from the origin.

At this point, the particular setup of the used method is not so important; therefore, all details are moved to Appendix A. For now, we simply think of any 3^{rd}–order, 5^{th}–order, and 7^{th}–order method.



(a) Pseudocolor plot (b) Scatter plot

Figure 2.1: Initial pressure distribution, (left) pseudocolor plot, (right) scatter plot with respect to the radius.



(a) 256^2 grid cells (b) 512^2 grid cells

Figure 2.2: Pressure at final time $T = 0.1$. In both plots, 3^{rd}–order method (upper left corner), 5^{th}–order method (upper right corner), 7^{th}–order method (lower right corner), and the reference solution (lower left corner).

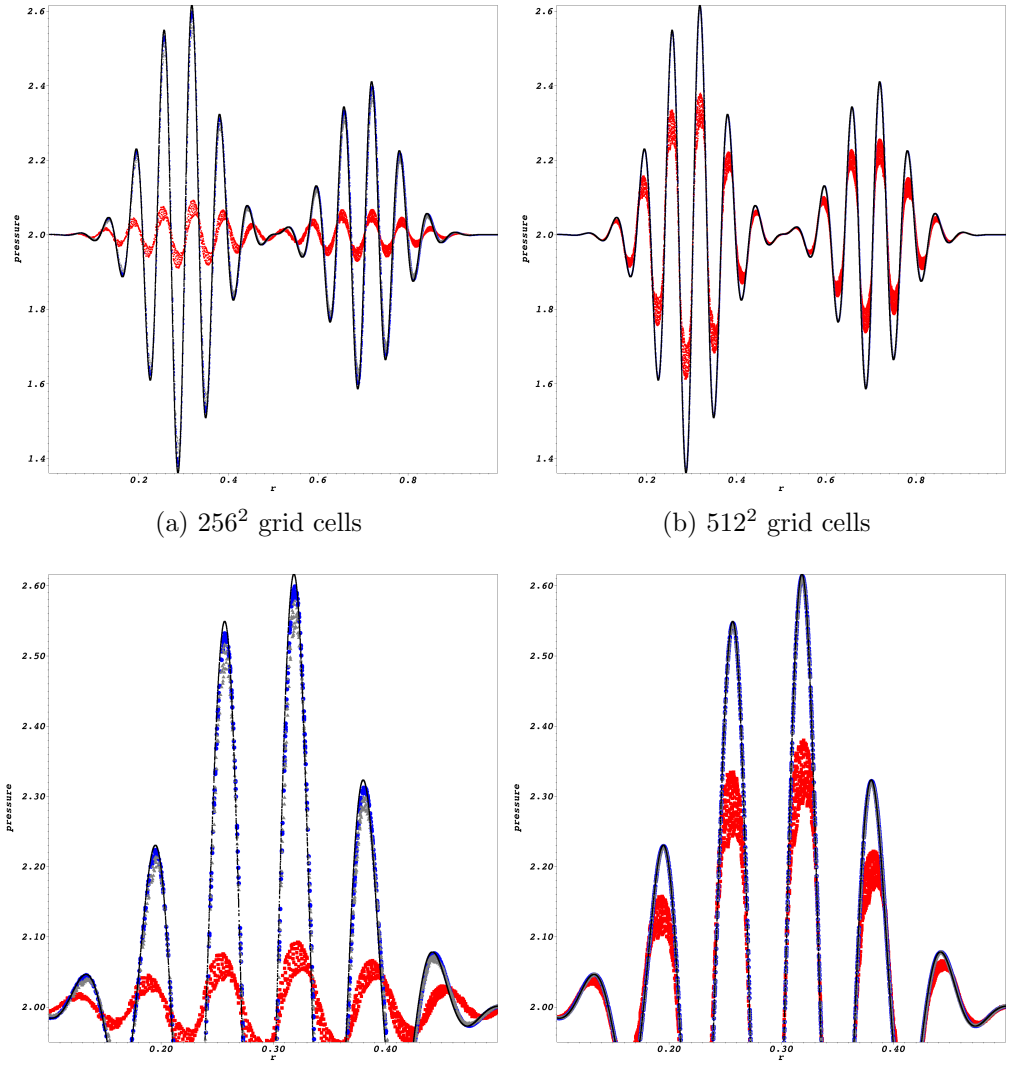
Figure 2.2 shows the results at the final time of the three methods as well as the reference solution. The results of the 3^{rd}–order method (upper left corner) differ the most from the reference solution (lower left corner). On the coarse grid with 256^2 grid cells, the solution structure became almost invisible. The results of the 5^{th}–order method (upper right corner) fits much better. Looking very carefully, one can see some differences at least on the coarse grid. On the fine grid, the solution is identical for the eye with the reference solution. The same holds for the results of the 7^{th}–order method (lower right corner) on both grid resolutions.

Figure 2.3 shows scatter plots for the same results. Again, we see that the 3^{rd}–order method (red squares) resolves poorly the solution structure on both the grids. The two higher–order methods perform much better. On the coarse grid and especially in Figure 2.3c, one can still see the difference for the 5^{th}–order method (grey triangles) and even for the 7^{th}–order method (blue circles). On the fine grid, however, the difference is not visible to the eye.

What we see in Figures 2.2 and 2.3 is also consistent with the measured L1 error shown in Table 2.1. On the coarsest grid, none of the methods converges. On a grid with 256^2 grid cells, the 5^{th}– and 7^{th}–order methods start to converge. But the low–order method still does not do so. On a grid with 512^2 cells, the 5^{th}–order method converges almost with the expected order. For the 7^{th}–order method, we see even superconvergence on this grid while the low–order method still converges very slowly. This example demonstrates the advantage and the need of high–order methods.

	3^{rd}–order		5^{th}–order		7^{th}–order	
grid	$\|p - p_{ex}\|_1$	EOC	$\|p - p_{ex}\|_1$	EOC	$\|p - p_{ex}\|_1$	EOC
64^2	2.042E-02		2.038E-02		2.039E-02	
128^2	3.089E-02	**-0.60**	2.796E-02	**-0.46**	2.347E-02	**-0.20**
256^2	2.725E-02	**0.18**	2.053E-03	**3.77**	4.409E-04	**5.73**
512^2	9.795E-03	**1.48**	7.309E-05	**4.81**	2.405E-06	**7.52**
1024^2	2.528E-03	**1.95**	2.367E-06	**4.95**	1.931E-08	**6.96**

Table 2.1: Convergence study for Example 2.1. The used methods are discussed in Appendix A.



(a) 256^2 grid cells

(b) 512^2 grid cells

(c) 256^2 grid cells

(d) 512^2 grid cells

Figure 2.3: Scatter plot of the pressure with respect to the radius at final time $T = 0.1$. Results of the 3^{rd}–order method (red squares), the 5^{th}–order method (grey triangles), and the 7^{th}–order method (blue circles).

Chapter 3

The WENO reconstruction

In 1994, Liu et al. [14] introduced the first WENO scheme as an improvement over the ENO scheme developed by Harten et al. [15]. A more general framework was proposed by Jiang et al. [16] in 1996. In the more recent review of Shu [3], more applications and a detailed discussion of the WENO scheme can be found.

3.1 The WENO method

In this section, we describe the main idea of a WENO method of order $(2r-1)$. In the following sections, we discuss the particular methods used in this thesis in more detail. For simplicity, we first restrict our considerations to the one–dimensional case, i.e. we consider initial value problems of the form

$$\begin{aligned} \partial_t q + \partial_x f(q) &= 0 \\ q(x,0) &= q_0(x), \end{aligned} \tag{3.1}$$

where $q : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}^m$ is a vector of conserved quantities, and $f : \mathbb{R}^m \to \mathbb{R}^m$ is a vector–valued flux function.

We discretize (3.1) using the method of lines approach on an equidistant mesh with grid cells $C_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ and mesh width Δx. The semi–discrete form can then be written as

$$Q_i'(t) = -\frac{1}{\Delta x}\left(f(q(x_{i+\frac{1}{2}},t)) - f(q(x_{i-\frac{1}{2}},t))\right), \tag{3.2}$$

where $Q_i(t)$ is the cell–average of the conserved quantities in grid cell C_i and $f(q(x_{i\pm\frac{1}{2}},t))$ is the flux at grid cell interfaces. Throughout this thesis, we assume that the temporal discretization is carries out by a sufficiently accurate Runge–Kutta method. We neglect the dependency on t for simplicity.

The numerical flux $F_{i\pm\frac{1}{2}}$ is an approximation to $f(q(x_{i\pm\frac{1}{2}}))$. To obtain an $(2r-1)^{th}$–order accurate approximation of the cell–average Q_i, we need

$$F_{i\pm\frac{1}{2}} = f(q(x_{i\pm\frac{1}{2}})) + \mathcal{O}(\Delta x^{2r-1}). \tag{3.3}$$

Let $q^{\pm}_{i+\frac{1}{2}}$ be an approximation to the limit $\lim_{h\to\pm 0} q(x_{i+\frac{1}{2}}+h)$. Then the flux can be computed using a numerical flux function such as the local Lax-Friedrichs flux, i.e. we compute

$$F_{i+\frac{1}{2}} = \frac{1}{2}\left[f(q^{-}_{i+\frac{1}{2}}) + f(q^{+}_{i+\frac{1}{2}}) - \alpha\left(q^{+}_{i+\frac{1}{2}} - q^{-}_{i+\frac{1}{2}}\right)\right], \tag{3.4}$$

where α is an upper estimate for the largest absolute value of the eigenvalues of the flux Jacobian matrix. Alternatively, we can compute a unique interface value $q^{*}_{i+\frac{1}{2}}$ of the conserved quantities, by exact or approximative solution of the Riemann problem with data $q^{\pm}_{i+\frac{1}{2}}$. The flux can then be computed using $F_{i+\frac{1}{2}} = f(q^{*}_{i+\frac{1}{2}})$. However, to satisfy Equation (3.3), we need

$$q^{\pm}_{i+\frac{1}{2}} = q(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^{2r-1}). \tag{3.5}$$

We first consider an r^{th}–order accurate polynomial reconstruction on cell C_i. There are r obvious choices for a stencil containing cell C_i. These are $\{S^r_{i,0}, ..., S^r_{i,r-1}\}$ with $S^r_{i,k} := \{C_{i+k-(r-1)}, ..., C_{i+k}\}$. Each polynomial $P^r_{i,k}$ reconstructed on $S^r_{i,k}$ satisfies the conditions

$$\int_C P^r_{i,k}(x) = \int_C q(x), \quad \forall C \in S^r_{i,k} \tag{3.6}$$

and provides an r^{th}–order accurate approximation to the conserved quantity in cell C_i. In Figure 3.1, the different stencils for $r = 3$ are illustrated. In the

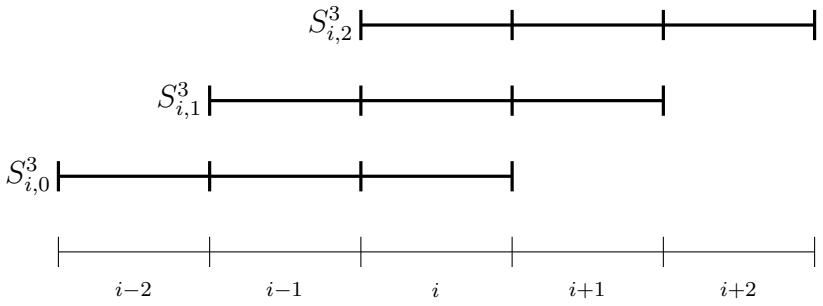


Figure 3.1: Illustration of the stencils for $r = 3$.

context of ENO, one would choose the polynomial that was reconstructed in the "most smooth" region and thus avoid reconstruction over discontinuities. But if the solution is smooth in all of the stencils there is no reason to restrict ourselves to only one. Instead, we can use the union of the r stencils $S^{2r-1}_{i,r-1} := \{C_{i-(r-1)}, ..., C_{i+r-1}\}$ and reconstruct a polynomial $P^{2r-1}_{i,r-1}$, which provides an approximation of order $2r-1$. The idea of the WENO scheme is now to combine the lower–order reconstructions through weights ω_k, which satisfy $\sum_{k=0}^{r-1} \omega_k = 1$. Assuming we are interested in the value of the conserved quantity q at $x_{i+\frac{1}{2}}$,

$$q_{i+\frac{1}{2}} = \sum_{k=0}^{r-1} \omega_k P^r_{i,k}(x_{i+\frac{1}{2}}) \tag{3.7}$$

is the WENO approximation for this quantity.

The crucial step here is to define the weights w_k in such a way that

$$\sum_{k=0}^{r-1} w_k P^r_{i,k}(x_{i+\frac{1}{2}}) = P^{2r-1}_{i,r-1}(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^{2r-1}) \tag{3.8}$$

whenever q is smooth in the large stencil $S^{2r-1}_{i,r-1}$. On the other hand, if $S^r_{i,k}$ contains a discontinuity, then w_k should be close to zero to reduce the influence of $P^r_{i,k}$ and in this way imitate the ENO behavior.

We first note that for $r \geq 2$ there exist optimal weights $\gamma_k > 0$ with $\sum_{k=0}^{r-1} \gamma_k = 1$ in such a way that

$$\sum_{k=0}^{r-1} \gamma_k P^r_{i,k}(x_{i+\frac{1}{2}}) = P^{2r-1}_{i,r-1}(x_{i+\frac{1}{2}}) = q(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^{2r-1}). \tag{3.9}$$

A common way of defining those weights is

$$\omega_k = \frac{\tilde{\omega}_k}{\sum_{k=0}^{r-1} \tilde{\omega}_k}, \quad \tilde{\omega}_k = \frac{\gamma_k}{(\epsilon + IS_k)^p}, \quad k = 0, ..., r-1, \tag{3.10}$$

where ϵ is the sensitivity, p the power parameter, and IS_k are indicators for the smoothness of q in $S^r_{i,k}$. The smoothness indicators are usually defined as in [16],

$$IS_k = \sum_{l=1}^{r-1} \Delta x^{2l-1} \int_{C_i} \left(\frac{d^l}{dx^l} P^r_{i,k}(x) \right)^2. \tag{3.11}$$

Liu et al. [14] originally defined ϵ as a constant to avoid division by zero ($\epsilon = 10^{-5}$). Later [17, 18, 19], it was pointed out that ϵ should be chosen as a function of Δx. Otherwise, an ϵ that is too large may undermine the ENO

behavior of the scheme; meanwhile, an ϵ that is too small may reduce the order of accuracy at critical points. The WENO reconstruction described above is usually denoted as the WENO-JS scheme in the literature.

As an alternative to retaining the optimal order of accuracy, Henrick et al. [17] proposed a mapping for the weights. Their method is known as the WENO-M scheme. According to Don et al. [20], it is about 25% more expensive than the WENO-JS scheme in terms of CPU time. Therefore, we will not further consider the WENO-M scheme in this thesis.

Another definition of the weights ω_k was proposed by Borges et al. [21], and Don and Borges [20]. In order to describe their method we introduce the so-called global optimal–order smoothness indicator

$$\tau_{2r-1} = \left| \sum_{k=0}^{r-1} c_k IS_k \right|, \tag{3.12}$$

where c_k are the given constants discussed in [18]. The weights are now given by

$$\omega_k = \frac{\tilde{\omega}_k}{\sum_{k=0}^{r-1} \tilde{\omega}_k}, \quad \tilde{\omega}_k = \gamma_k \left(1 + \frac{\tau_{2r-1}}{(\epsilon + IS_k)^p} \right), \quad k = 0, ..., r-1, \tag{3.13}$$

where ϵ is the sensitivity and p the power parameter. According to Don and Borges [20], the ϵ in (3.13) should be defined as $\epsilon = \Delta x^m$. Where the optimal choice of m is dependent on r and the power parameter p, the resulting WENO method is known in the literature as the WENO-Z scheme.

In the next subsections, we will provide all the so far omitted coefficients that are necessary for implementing the different WENO methods used in this thesis.

3.1.1 The 5^{th}–order WENO method

A 5^{th}–order accurate method is obtained by setting $r = 3$. On a uniform grid, the evaluation of the three polynomials $P^3_{i,k}, k = 0, 1, 2$ at the interface of cell C_i leads to the following formulas

$$\begin{aligned}
q^{(0-)}_{i+\frac{1}{2}} &= \frac{1}{3}Q_{i-2} - \frac{7}{6}Q_{i-1} + \frac{11}{6}Q_i, & q^{(0+)}_{i-\frac{1}{2}} &= -\frac{1}{6}Q_{i-2} + \frac{5}{6}Q_{i-1} + \frac{1}{3}Q_i, \\
q^{(1-)}_{i+\frac{1}{2}} &= -\frac{1}{6}Q_{i-1} + \frac{5}{6}Q_i + \frac{1}{3}Q_{i+1}, & q^{(1+)}_{i-\frac{1}{2}} &= \frac{1}{3}Q_{i-1} + \frac{5}{6}Q_i - \frac{1}{6}Q_{i+1}, \\
q^{(2-)}_{i+\frac{1}{2}} &= \frac{1}{3}Q_i + \frac{5}{6}Q_{i+1} - \frac{1}{6}Q_{i+2}, & q^{(2+)}_{i-\frac{1}{2}} &= \frac{11}{6}Q_i - \frac{7}{6}Q_{i+1} + \frac{1}{3}Q_{i+2}.
\end{aligned} \tag{3.14}$$

The interface values of the conserved quantities are then computed by

$$\begin{aligned} q^{-}_{i+\frac{1}{2}} &= \omega^{-}_{0} q^{(1-)}_{i+\frac{1}{2}} + \omega^{-}_{1} q^{(2-)}_{i+\frac{1}{2}} + \omega^{-}_{2} q^{(3-)}_{i+\frac{1}{2}}, \\ q^{+}_{i-\frac{1}{2}} &= \omega^{+}_{0} q^{(1+)}_{i-\frac{1}{2}} + \omega^{+}_{1} q^{(2+)}_{i-\frac{1}{2}} + \omega^{+}_{2} q^{(3+)}_{i-\frac{1}{2}}. \end{aligned} \tag{3.15}$$

As discussed in the previous section, we have several ways of defining the weights $\omega^{\pm}_{k}$. However, we first need the smoothness indicators IS_k. For $r = 3$, Equation (3.11) leads to

$$\begin{aligned} IS_0 &= \frac{13}{12}\left(Q_{i-2} - 2Q_{i-1} + Q_i\right)^2 + \frac{1}{4}\left(Q_{i-2} - 4Q_{i-1} + 3Q_i\right)^2, \\ IS_1 &= \frac{13}{12}\left(Q_{i-1} - 2Q_i + Q_{i+1}\right)^2 + \frac{1}{4}\left(Q_{i-1} + Q_{i+1}\right)^2, \\ IS_2 &= \frac{13}{12}\left(Q_i - 2Q_{i+1} + Q_{i+2}\right)^2 + \frac{1}{4}\left(3Q_i - 4Q_{i+1} + Q_{i+2}\right)^2. \end{aligned} \tag{3.16}$$

To obtain the 5^{th}–order accurate WENO-JS method, we use Equation (3.10) to define the weights

$$\omega^{\pm}_{k} = \frac{\tilde{\omega}^{\pm}_{k}}{\tilde{\omega}^{\pm}_{0} + \tilde{\omega}^{\pm}_{1} + \tilde{\omega}^{\pm}_{2}}, \quad \tilde{\omega}^{\pm}_{k} = \frac{\gamma^{\pm}_{k}}{(\epsilon + IS_k)^p}, \quad k = 0, 1, 2 \tag{3.17}$$

with $\gamma^{-}_{0} = \gamma^{+}_{2} = \frac{1}{10}$, $\gamma^{-}_{1} = \gamma^{+}_{1} = \frac{3}{5}$, $\gamma^{-}_{2} = \gamma^{+}_{0} = \frac{3}{10}$, $\epsilon = \Delta x^2$ and $p = 2$.

To obtain the 5^{th}–order accurate WENO-Z method, we use Equation (3.13) to define the weights

$$\omega^{\pm}_{k} = \frac{\tilde{\omega}^{\pm}_{k}}{\tilde{\omega}^{\pm}_{0} + \tilde{\omega}^{\pm}_{1} + \tilde{\omega}^{\pm}_{2}}, \quad \tilde{\omega}^{\pm}_{k} = \gamma^{\pm}_{k}\left(1 + \left(\frac{\tau_5}{IS_k + \epsilon}\right)^p\right), \quad k = 0, 1, 2 \tag{3.18}$$

with $\tau_5 = |IS_0 - IS_2|$, $\epsilon = \Delta x^4$, $p = 2$, and $\gamma^{\pm}_{k}$ as above.

3.1.2 The 7^{th}–order WENO method

We will also use 7^{th}–order accurate WENO methods. In this case, we set $r = 4$ and the interfaces values of the conserved quantities are computed by

$$q^{\mp}_{i\pm\frac{1}{2}} = \sum_{k=0}^{3} \omega^{\mp}_{k} q^{(k\mp)}_{i\pm\frac{1}{2}}, \tag{3.19}$$

with

$$q^{(k\mp)}_{i\pm\frac{1}{2}} = \sum_{l=0}^{3} a^{\pm}_{k,l} Q_{i-3+k+l}. \tag{3.20}$$

l	0	1	2	3
$a^+_{0,l}$	$\frac{-3}{12}$	$\frac{13}{12}$	$\frac{-23}{12}$	$\frac{25}{12}$
$a^+_{1,l}$	$\frac{1}{12}$	$\frac{-5}{12}$	$\frac{13}{12}$	$\frac{3}{12}$
$a^+_{2,l}$	$\frac{-1}{12}$	$\frac{7}{12}$	$\frac{7}{12}$	$\frac{-1}{12}$
$a^+_{3,l}$	$\frac{3}{12}$	$\frac{13}{12}$	$\frac{-5}{12}$	$\frac{1}{12}$

Table 3.1: Coefficients for the 7^{th}–order WENO reconstruction.

The coefficients $a^+_{k,l}$ are given in Table 3.1. Due to the symmetry of the scheme, the relation $a^-_{k,l} = a^+_{3-k,3-l}$ holds. Applying Equation (3.11) for $r = 4$, we get (up to a scaling factor)

$$\begin{aligned}
IS_0 &= Q_{i-3}(547Q_{i-3} - 3882Q_{i-2} + 4642Q_{i-1} - 1854Q_i) + \\
&\quad Q_{i-2}(7043Q_{i-2} - 17246Q_{i-1} + 7042Q_i) + \\
&\quad Q_{i-1}(11003Q_{i-1} - 9402Q_i) + 2107Q_i^2, \\
IS_1 &= Q_{i-2}(267Q_{i-2} - 1642Q_{i-1} + 1602Q_i - 494Q_{i+1}) + \\
&\quad Q_{i-1}(2843Q_{i-1} - 5966Q_i + 1922Q_{i+1}) + \\
&\quad Q_i(3443Q_i - 2522Q_{i+1}) + 547Q_{i+1}^2, \\
IS_2 &= Q_{i-1}(547Q_{i-1} - 2522Q_i + 1922Q_{i+1} - 494Q_{i+2}) + \\
&\quad Q_i(3443Q_i - 5966Q_{i+1} + 1602Q_{i+2}) + \\
&\quad Q_{i+1}(2843Q_{i+1} - 1642Q_{i+2}) + 267Q_{i+2}^2, \\
IS_3 &= Q_i(2107Q_i - 9402Q_{i+1} + 7042Q_{i+2} - 1854Q_{i+3}) + \\
&\quad Q_{i+1}(11003Q_{i+1} - 17246Q_{i+2} + 4642Q_{i+3}) + \\
&\quad Q_{i+2}(7043Q_{i+2} - 3882Q_{i+3}) + 547Q_{i+3}^2.
\end{aligned} \tag{3.21}$$

Note that in the definition of the weights a scaling factor for the smoothness indicators is neglectable.

Again, we have two ways of defining the weights. To obtain the 7^{th}–order accurate WENO-JS method, we use Equation (3.10) to define the weights

$$\omega_k^\pm = \frac{\tilde{\omega}_k^\pm}{\sum_{m=0}^{3} \tilde{\omega}_m^\pm}, \quad \tilde{\omega}_k^\pm = \frac{\gamma_k^\pm}{(\epsilon + IS_k)^p}, \quad k = 0, .., 3 \tag{3.22}$$

with $\gamma_0^- = \gamma_3^+ = \frac{1}{35}$, $\gamma_1^- = \gamma_2^+ = \frac{12}{35}$, $\gamma_2^- = \gamma_1^+ = \frac{18}{35}$, $\gamma_3^- = \gamma_0^+ = \frac{4}{35}$, $\epsilon = \Delta x^2$ and $p = 2$.

To obtain the 7^{th}–order accurate WENO-Z method, we use Equation

(3.13) to define the weights

$$\omega_k^{\pm} = \frac{\tilde{\omega}_k^{\pm}}{\sum_{m=0}^{3} \tilde{\omega}_m^{\pm}}, \quad \tilde{\omega}_k^{\pm} = \gamma_k^{\pm} \left(1 + \left(\frac{\tau_7}{IS_k + \epsilon}\right)^p\right), \quad k = 0, .., 3 \tag{3.23}$$

where $\tau_7 = |IS_0 + 3IS_1 - 3IS_2 - IS_3|$, $\epsilon = \Delta x^5$ and $p = 2$ as suggested in [20].

The smoothness indicators, as defined in Equation (3.11), should always be non-negative. But for IS_k close to zero, due to round–off errors, the computations in Equation (3.21) might result in a small negative value. To avoid unexpected behavior, we used $|IS_k|$ in our implementation instead.

3.2 Comparing the methods in 1D

In this section, we test and compare the methods introduced so far by applying them to the Euler equations of gas dynamics. These equations describe the conservation of mass, momentum, and energy. A detailed discussion of the equations can be found in many textbooks, such as [22] or [13]. The one–dimensional Euler equations are given by

$$\partial_t \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} + \partial_x \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{pmatrix} = 0, \tag{3.24}$$

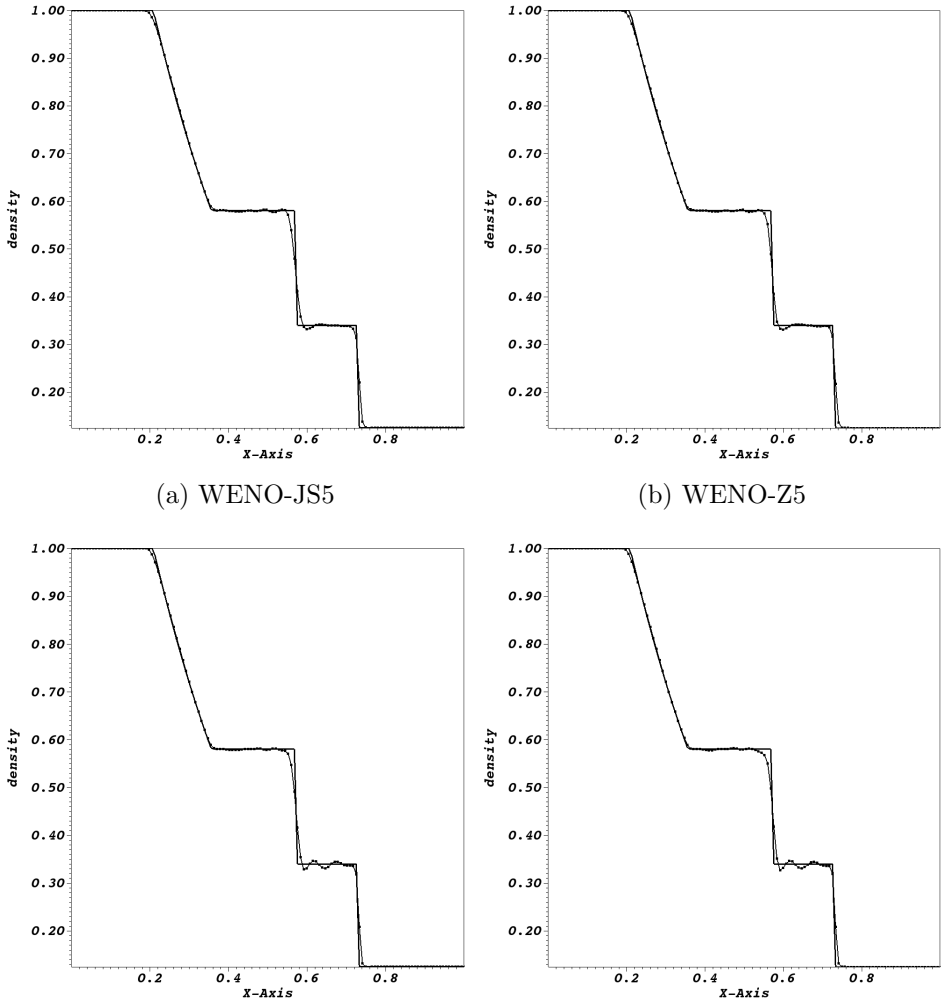
with the total energy density

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2. \tag{3.25}$$

Here, ρ is the density, u the velocity, and p the pressure. The constant γ in the equation of state (4.21) is often called the adiabatic exponent. The value depends on the gas under consideration. In this chapter, we always set $\gamma = 1.4$.

We use adaptive mesh refinement (AMR) realized by a regular bisection of grid blocks. In Chapter 5, the grid structure will be discussed in more detail.

The test cases are taken from the collection of Liska and Wendroff [23]. The setup for the first five tests is always the same. Initially, a discontinuity located at $x_0 \in I$ separates two constant states (ρ_L, u_L, p_L) to the left and (ρ_R, u_R, p_R) to the right of x_0. The computation is performed until the final time T. We apply SSP-RK(10,4) with $CFL \approx 1.5$ in time and use the Roe Riemann solver with an entropy fix, in accordance with Harten and Hyman [24], to compute the fluxes. All computations are compared using the exact solution obtained by the exact Riemann solver presented by Toro [22].



(a) WENO-JS5 (b) WENO-Z5

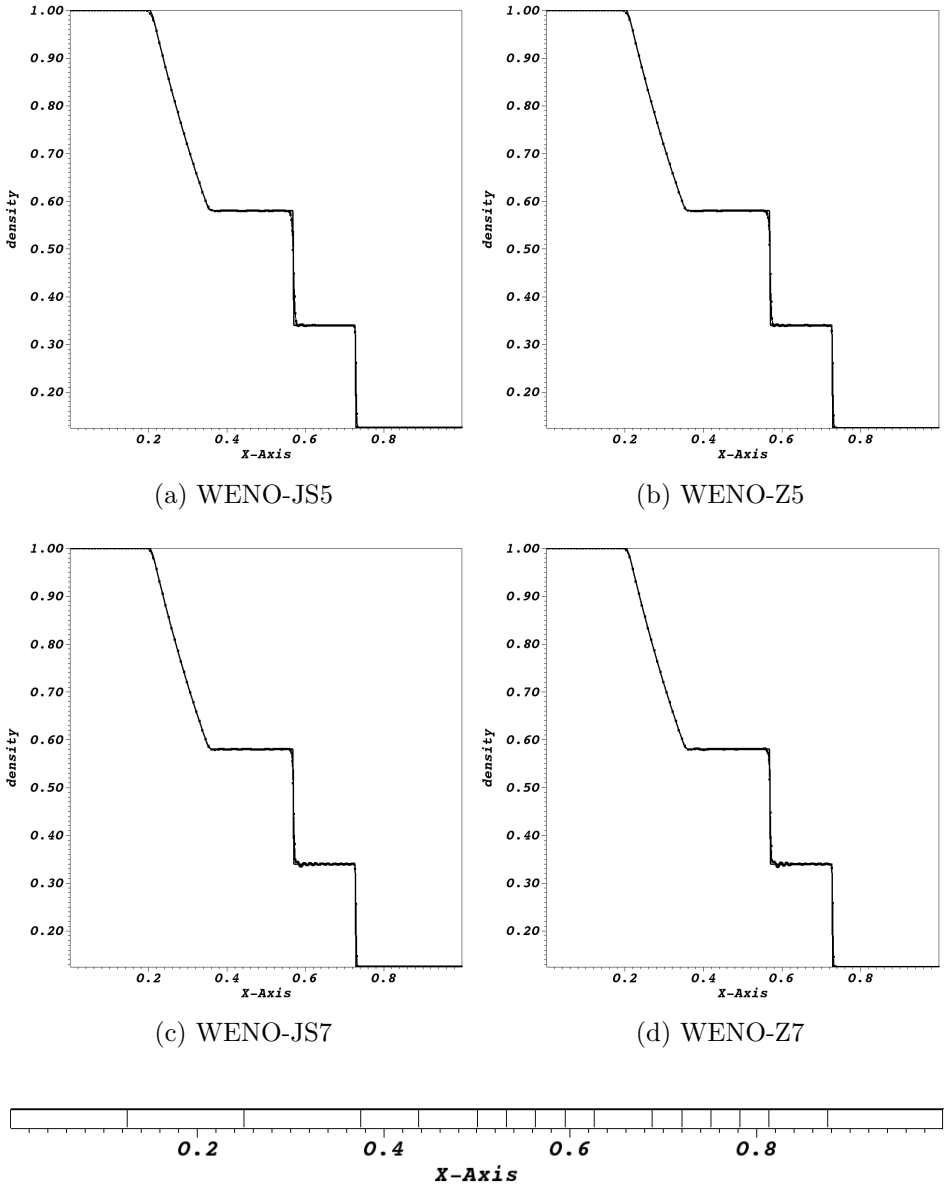
(c) WENO-JS7 (d) WENO-Z7

Figure 3.2: Result for Example 3.1 on a uniform grid with 128 grid cells.

Example 3.1. *This is a variant of Sod's Riemann problem. We set*

$$\begin{aligned} &x_0 = 0.3, \qquad T = 0.2, \qquad I = [0,1], \\ &\rho_l = 1, \qquad u_l = 0.75, \quad p_l = 1, \\ &\rho_r = 0.125, \quad u_r = 0, \qquad p_r = 0.1. \end{aligned} \tag{3.26}$$

Figure 3.2 presents the result at final time for each of the methods on a uniform grid. Both 7^{th}–order accurate methods are more oscillatory than the 5^{th}–order accurate methods. The relative L1 error in Table 3.2 shows that the results are quite similar. In Figure 3.3, we present the result on a



(a) WENO-JS5

(b) WENO-Z5

(c) WENO-JS7

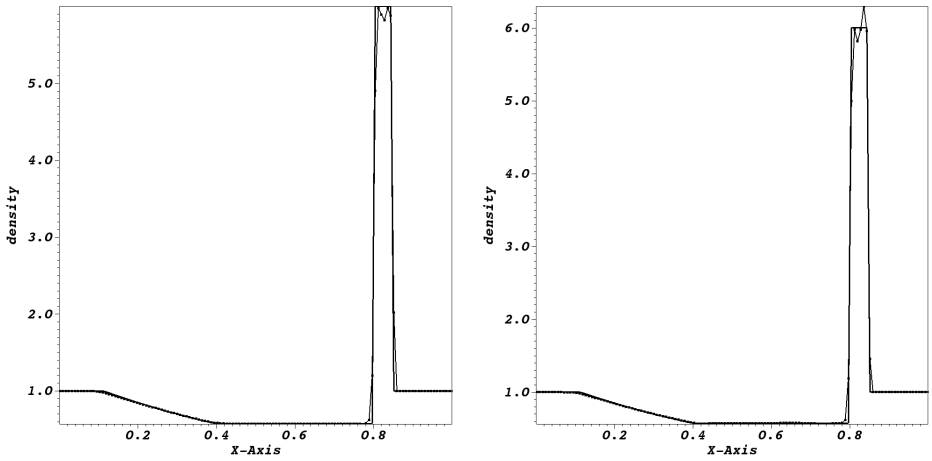
(d) WENO-Z7

(e) Structure of the AMR grid at final time.

Figure 3.3: Result for Example 3.1 on a grid with 128 grid cells and two AMR levels.

grid with two levels of refinement. The cell size is 1/128 on the coarsest grid, 1/256 on the first refinement level, and 1/512 on the second refinement level. The structure of the grid is shown in Figure 3.3e. Both discontinuities are covered by grids with the finest resolution. Therefore, the oscillations reduce

significantly. For both 7^{th}-order methods, tiny oscillations are still visible. However, the oscillations do not increase the L1 error compared with the 5^{th}–order methods. The entire rarefaction wave is outside the refined zone. Nevertheless, the result on the AMR grid agrees much better with the exact solution. This is due to better resolution of the solution at earlier times.



(a) WENO-Z5 (b) WENO-Z7

Figure 3.4: Result for Example 3.3 on a uniform grid with 128 grid cells.

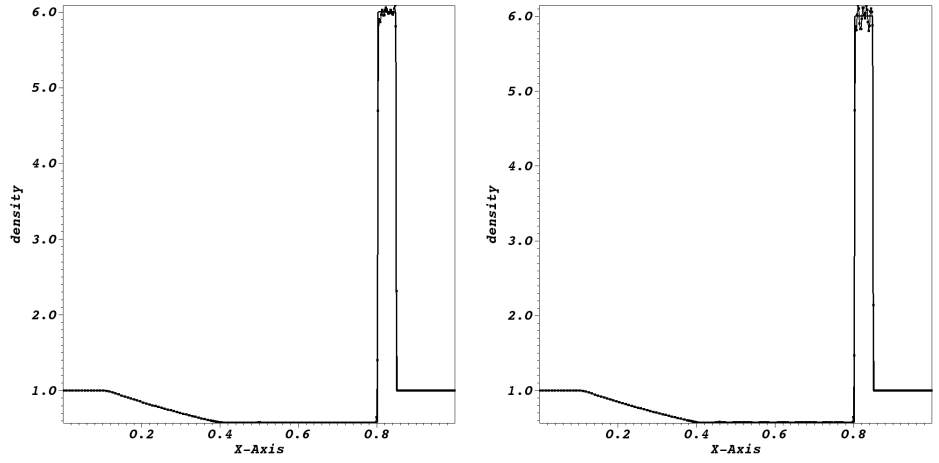
Example 3.2. *This is the 123 problem first proposed by Einfeld et al. [25]. We set*

$$\begin{aligned} &x_0 = 0.5, \quad T = 0.15, \quad I = [0,1], \\ &\rho_l = 1, \quad u_l = -2, \quad p_l = 0.4, \\ &\rho_r = 1, \quad u_r = 2, \quad p_r = 0.4. \end{aligned} \tag{3.27}$$

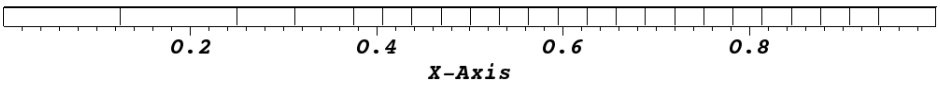
The exact solution for this test consists of two rarefaction waves moving in opposite directions. At the center, a low–density and low–pressure region is created. This leads to difficulties since negative pressure is reconstructed after a few steps by each of the methods. Positivity preserving considerations, as discussed by Zhang et al. [26], could resolve this issue. But they are not within the scope of this work.

Example 3.3. *We set*

$$\begin{aligned} &x_0 = 0.8, \quad T = 0.012, \quad I = [0,1], \\ &\rho_l = 1, \quad u_l = -19.59745, \quad p_l = 1000, \\ &\rho_r = 1, \quad u_r = -19.59745, \quad p_r = 0.01. \end{aligned} \tag{3.28}$$



(a) WENO-Z5 (b) WENO-Z7



(c) Structure of the AMR grid at final time.

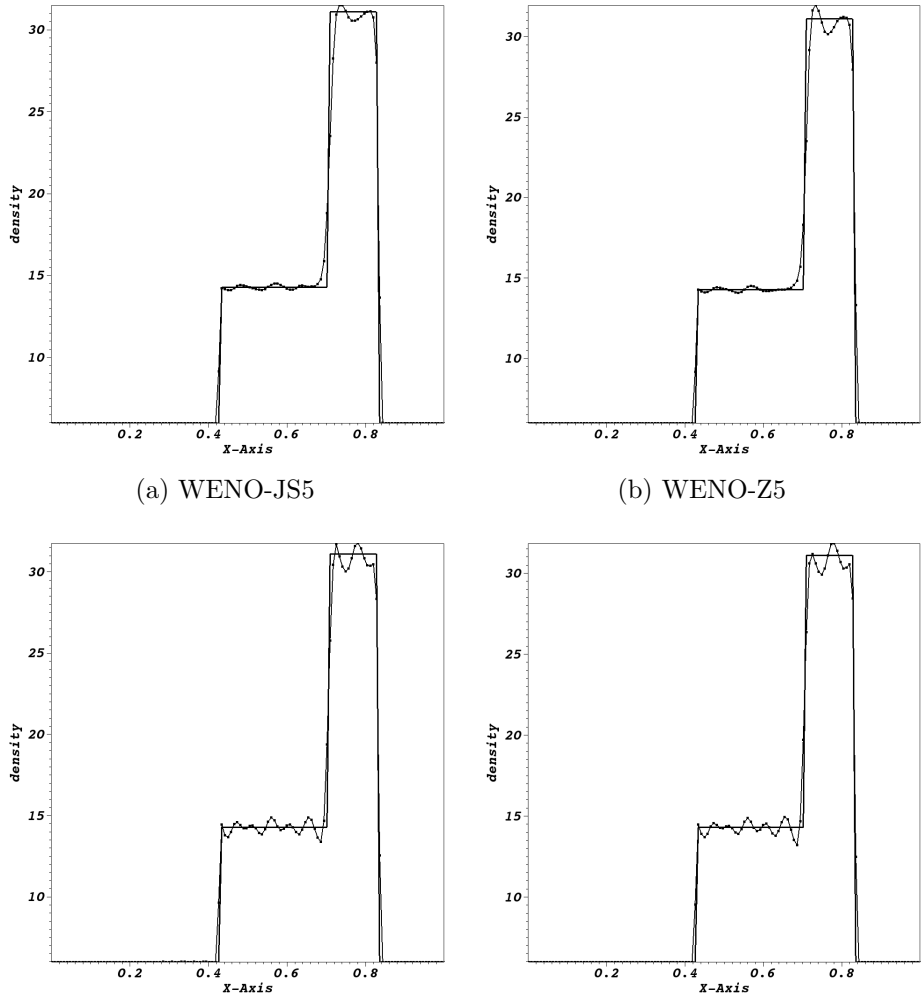
Figure 3.5: Result for Example 3.3 on a grid with 128 grid cells and two AMR levels.

The exact solution for this test consists of a left rarefaction wave, a contact, and a right shock. The results of this test are presented in Figure 3.4. Both WENO-JS methods reconstruct negative pressure after several time steps and, therefore, fail this test. This could be resolved by reducing the CFL number or the sensitivity parameter ϵ. However, this would not be a fair comparison anymore. All results are quite oscillatory, especially those produced by the 7^{th}–order accurate method.

Example 3.4. *We set*

$$\begin{aligned} &x_0 = 0.4, && T = 0.35, && I = [0,1], \\ &\rho_l = 5.99924, && u_l = 19.5975, && p_l = 460.894, \\ &\rho_r = 5.99242, && u_r = -6.19633, && p_r = 46.095. \end{aligned} \tag{3.29}$$

The solution consists of two shock waves and a contact discontinuity. Figure 3.6 presents the results on a uniform grid with 128 grid cells. All methods produce oscillatory results, the 7^{th}–order methods more so than the 5^{th}–order methods. This can also be seen from the error in Table 3.2. With two levels of refinement, the results are much less oscillatory, as shown in



(a) WENO-JS5 (b) WENO-Z5

(c) WENO-JS7 (d) WENO-Z7

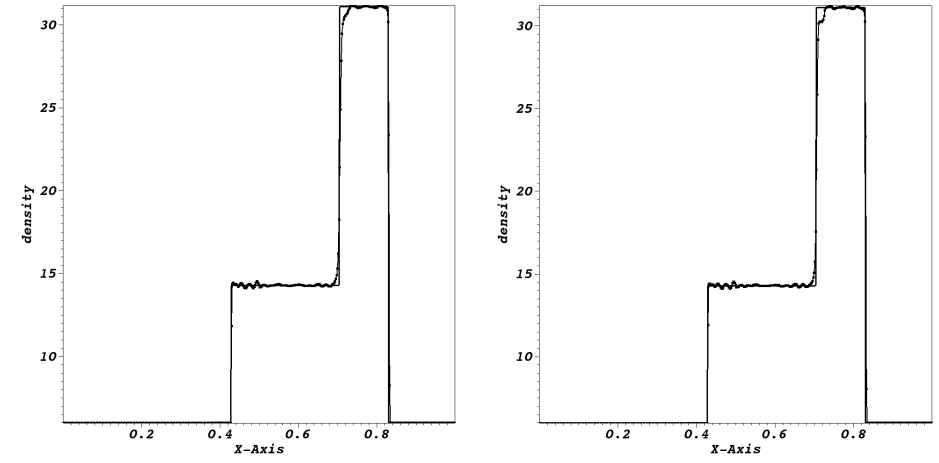
Figure 3.6: Result for Example 3.4 on a uniform grid with 128 grid cells.

Figure 3.7. Due to the nested structure of the grid and the location of the discontinuities, only a small part of the computational domain is covered by the coarsest grid.

Example 3.5. *The peak problem. We set*

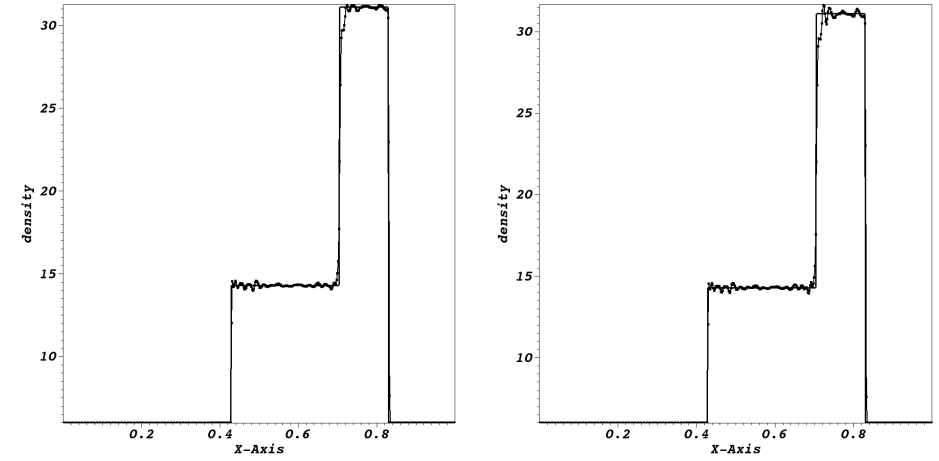
$$\begin{aligned} &x_0 = 0.5, && T = 0.0039, && I = [0.1, 0.6], \\ &\rho_l = 0.1261192, && u_l = 8.9047029, && p_l = 782.92899, \\ &\rho_r = 6.59143, && u_r = 2.2654207, && p_r = 3.1544874. \end{aligned} \tag{3.30}$$

Figure 3.8 presents the solution on a uniform grid with 128 grid cells.



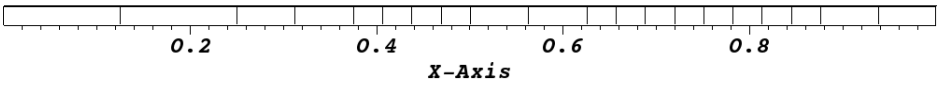
(a) WENO-JS5

(b) WENO-Z5



(c) WENO-JS7

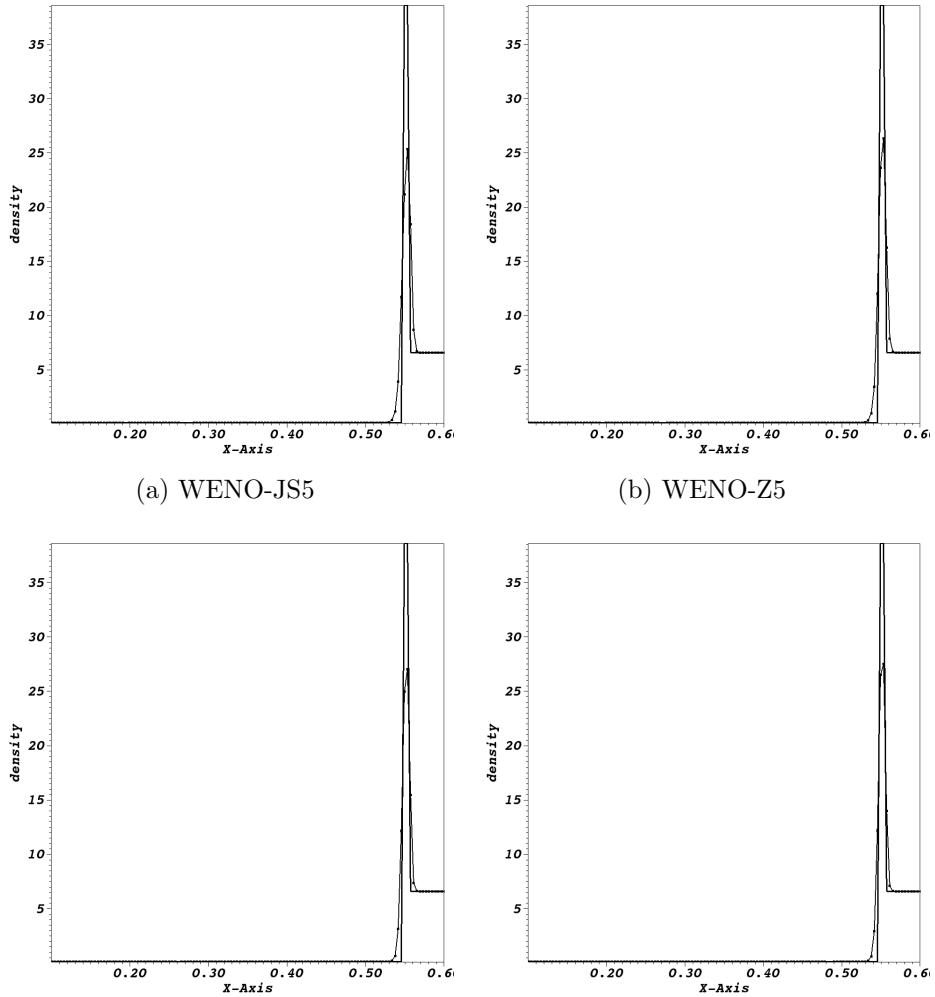
(d) WENO-Z7



(e) Structure of the AMR grid at final time.

Figure 3.7: Result for Example 3.4 on a grid with 128 grid cells and two AMR levels.

The density peak is better resolved by the 7^{th}–order methods. According to the L1 error in Table 3.2, the WENO-Z methods perform better than their WENO-JS counterparts. With two levels of refinement, all methods produce similar results (see Figures 3.9).



(a) WENO-JS5 (b) WENO-Z5

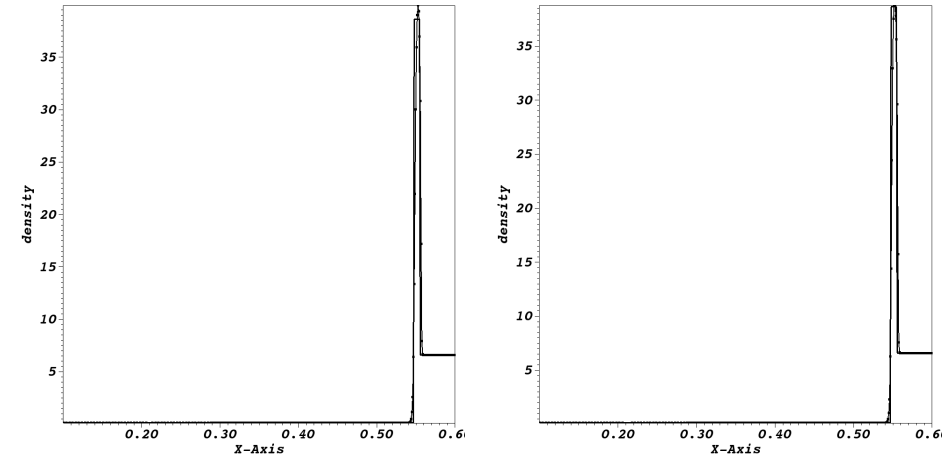
(c) WENO-JS7 (d) WENO-Z7

Figure 3.8: Result for Example 3.5 on a uniform grid with 128 grid cells.

Example 3.6. *Another classical test is the so–called shock entropy wave interaction. The initial data is given by*

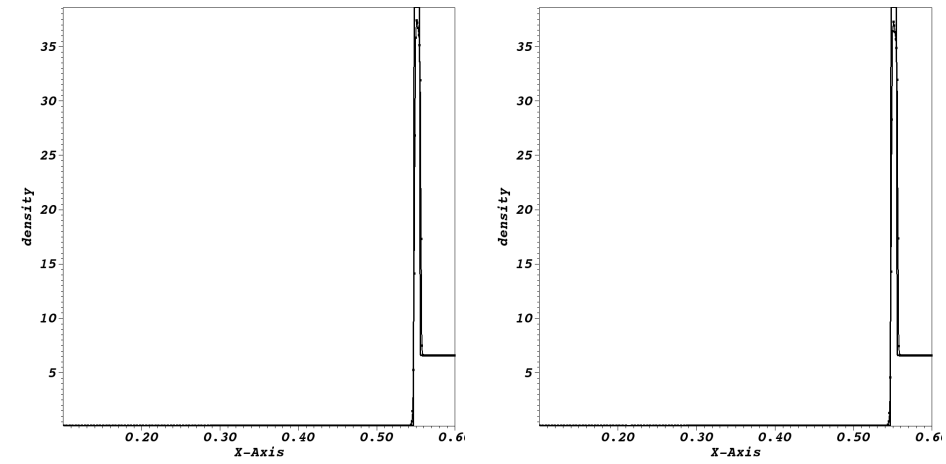
$$\begin{aligned} &\rho = 3.857143, && u = 2.629369, && p = 10.33333 && \text{when } x < 1, \\ &\rho = 1 + 0.2\sin 5x, && u = 0, && p = 1, && \text{when } x \geq 1. \end{aligned}$$

This test starts with a Mach 3 shock moving to the right and interacting with a sine wave in density. Figure 3.10 presents the results on a uniform grid with 400 cells. The solid line in each plot is the result of a computation on 6,400 grid cells using WENO-Z7. None of the methods can resolve



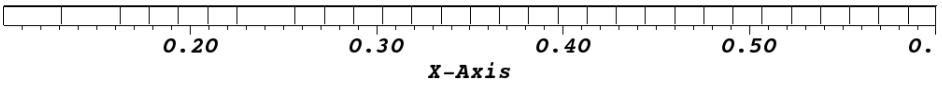
(a) WENO-JS5

(b) WENO-Z5



(c) WENO-JS7

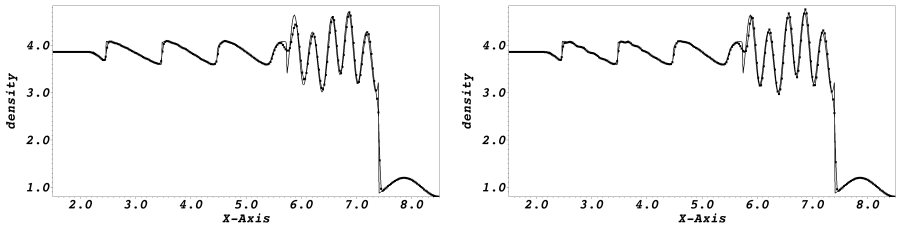
(d) WENO-Z7



(e) Structure of the AMR grid at final time.

Figure 3.9: Result for Example 3.5 on a grid with 128 grid cells and two AMR levels.

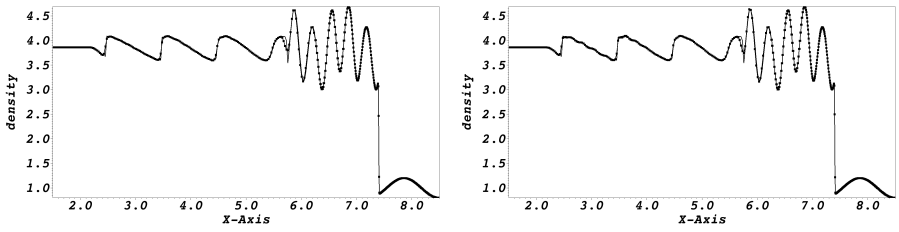
the maximum behind the shock on the coarse grid. The performance is most different in the region around $x = 5.6$. Both 7^{th}–order accurate methods perform better than the 5^{th}–order accurate methods. This becomes even more obvious on comparing the L1 errors in Table 3.2. In Figure 3.11, the results



(a) WENO-JS5 (b) WENO-Z5

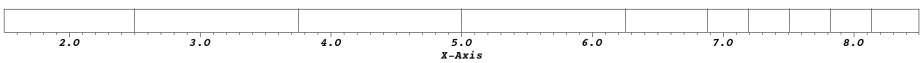
(c) WENO-JS7 (d) WENO-Z7

Figure 3.10: Result for Example 3.6 shock entropy wave interaction on a uniform grid with 400 grid cells.



(a) WENO-JS5 (b) WENO-Z5

(c) WENO-JS7 (d) WENO-Z7



(e) Structure of the AMR grid at final time.

Figure 3.11: Result for Example 3.6 shock entropy wave interaction on a grid with 400 grid cells and two AMR levels.

with two levels of refinement are presented. The shock is inside the refined region. Figure 3.3e shows the grid structure at the final time. Each of the visualized blocks consists of 50 cells in the x direction. Now, the maximum right after the shock is resolved by each method. The high–frequency region agrees better with the highly resolved computation for each method, even though only a part is covered by the fine grid.

Example	grid	WENO-JS5 $\|\rho - \rho_{ex}\|_1$	WENO-Z5 $\|\rho - \rho_{ex}\|_1$	WENO-JS7 $\|\rho - \rho_{ex}\|_1$	WENO-Z7 $\|\rho - \rho_{ex}\|_1$
3.1	128	4.13E-03	3.69E-03	4.02E-03	3.99E-03
	128 + 2L	1.40E-03	1.26E-03	1.24E-03	1.23E-03
	512	1.32E-03	1.21E-03	1.21E-03	1.19E-03
3.2		failed	failed	failed	failed
3.3	128	failed	4.69E-03	failed	3.82E-03
	128 + 2L	failed	1.66E-03	failed	2.06E-03
	512	failed	1.66E-03	failed	2.06E-03
3.4	128	9.44E-03	9.34E-03	1.02E-02	1.02E-02
	128 + 2L	4.14E-03	3.91E-03	4.07E-03	4.29E-03
	512	4.39E-03	4.18E-03	4.35E-03	4.58E-03
3.5	128	1.90E-02	1.63E-02	1.47E-02	1.33E-02
	128 + 2L	5.09E-03	4.67E-03	4.92E-03	4.85E-03
	512	5.09E-03	4.67E-03	4.92E-03	4.86E-03
3.6	400	5.52E-03	5.83E-03	3.83E-03	4.02E-03
	400 + 2L	1.45E-03	1.61E-03	1.28E-03	1.45E-03
	1600	1.04E-03	1.15E-03	1.01E-03	1.15E-03

Table 3.2: Relative L1 error for the 1D Riemann problems on a coarse uniform grid, an AMR grid with two levels of refinement, and a fine uniform grid.

For each example, the L1 errors on the AMR grid correspond very well with the L1 errors on the fine uniform grid (see Table 3.2). The resolution of the fine uniform grid matches the resolution on the highest refinement level of the AMR grid.

3.3 The dimension–by–dimension WENO method

A common and simple way to extend the method to higher dimensions is the dimension–by–dimension approach. We consider the initial value problem of the form

$$\begin{aligned} \partial_t q + \partial_x f(q) + \partial_y g(q) &= 0 \\ q(x,y,0) &= q_0(x,y), \end{aligned} \tag{3.31}$$

where $q : \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}^m$ is a vector of conserved quantities and $f, g : \mathbb{R}^m \to \mathbb{R}^m$ are the vector–valued flux functions.

We discretize (3.31) using the method of lines approach on an equidistant mesh with grid cells $C_{i,j} = [x_{i-\frac{1}{2},j}, x_{i+\frac{1}{2},j}] \times [y_{i,j-\frac{1}{2}}, y_{i,j+\frac{1}{2}}]$ and mesh width $\Delta x, \Delta y$. The semi–discrete form can then be written as

$$Q'_{i,j}(t) = -\frac{1}{\Delta x}\left(\mathscr{F}_{i+\frac{1}{2},j}(t) - \mathscr{F}_{i-\frac{1}{2},j}(t)\right) - \frac{1}{\Delta y}\left(\mathscr{G}_{i,j+\frac{1}{2}}(t) - \mathscr{G}_{i,j-\frac{1}{2}}(t)\right), \tag{3.32}$$

where $Q_{i,j}(t)$ is the cell–average of the conserved quantities in the grid cell $C_{i,j}$ and the terms $\mathscr{F}_{i\pm\frac{1}{2},j}(t)$ and $\mathscr{G}_{i,j\pm\frac{1}{2}}(t)$ are the interface fluxes

$$\begin{aligned} \mathscr{F}_{i\pm\frac{1}{2},j}(t) &= \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(q(x_{i\pm\frac{1}{2}}, y, t))dy, \\ \mathscr{G}_{i,j\pm\frac{1}{2}}(t) &= \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g(q(x, y_{j\pm\frac{1}{2}}, t))dx. \end{aligned} \tag{3.33}$$

Relation (3.32) is an exact formula. As before, we neglect the dependency of t for simplicity and assume that the temporal discretization is carries out by a sufficiently accurate Runge–Kutta method. The numerical fluxes $F_{i\pm\frac{1}{2},j}$ and $G_{i,j\pm\frac{1}{2}}$ are approximations of $\mathscr{F}_{i\pm\frac{1}{2},j}$ and $\mathscr{G}_{i,j\pm\frac{1}{2}}$, respectively. To obtain an p^{th}–order accurate approximation of the cell–average $Q_{i,j}$, we need

$$\begin{aligned} F_{i\pm\frac{1}{2},j} &= \mathscr{F}_{i\pm\frac{1}{2},j} + \mathcal{O}(\Delta x^p + \Delta y^p), \\ G_{i,j\pm\frac{1}{2}} &= \mathscr{G}_{i,j\pm\frac{1}{2}} + \mathcal{O}(\Delta x^p + \Delta y^p). \end{aligned} \tag{3.34}$$

Applying any one–dimensional WENO reconstruction from the previous section, we compute averaged values of the conserved quantities at all grid cell interfaces, which are denoted by $Q^{\pm}_{i+\frac{1}{2},j}$ and $Q^{\pm}_{i,j+\frac{1}{2}}$. These interface

values are p^{th}–order accurate approximations of edge–averaged values of the conserved quantities, i.e.

$$\begin{aligned} Q^{\pm}_{i+\frac{1}{2},j} &= \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} q(x_{i+\frac{1}{2}},y)dy + \mathcal{O}(\Delta x^p) \\ Q^{\pm}_{i,j+\frac{1}{2}} &= \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(x,y_{j+\frac{1}{2}})dx + \mathcal{O}(\Delta y^p), \end{aligned} \tag{3.35}$$

where q is the exact solution. As before, the superscript $\pm$ refers to the limit $x \to x_{i+\frac{1}{2}} \pm 0$ for $Q^{\pm}_{i+\frac{1}{2},j}$ and similar to the limit $y \to y_{j+\frac{1}{2}} \pm 0$ for $Q^{\pm}_{i,j+\frac{1}{2}}$.

These edge–averaged interface values of the conserved quantities can now be used to compute interface fluxes. One choice could be the local Lax-Friedrichs flux formula, i.e. we compute

$$\begin{aligned} F_{i+\frac{1}{2},j} &= \frac{1}{2}\left[f(Q^{-}_{i+\frac{1}{2},j}) + f(Q^{+}_{i+\frac{1}{2},j}) - \alpha\left(Q^{+}_{i+\frac{1}{2},j} - Q^{-}_{i+\frac{1}{2},j}\right)\right] \\ G_{i,j+\frac{1}{2}} &= \frac{1}{2}\left[g(Q^{-}_{i,j+\frac{1}{2}}) + g(Q^{+}_{i,j+\frac{1}{2}}) - \alpha\left(Q^{+}_{i,j+\frac{1}{2}} - Q^{-}_{i,j+\frac{1}{2}}\right)\right], \end{aligned} \tag{3.36}$$

where α is an upper estimate for the largest absolute value of the eigenvalues of the flux Jacobian matrix. Alternatively, we can compute a unique interface value $Q^{*}_{i-\frac{1}{2},j}$ of the conserved quantities by exact or approximative solution of the Riemann problem with data $Q^{\pm}_{i-\frac{1}{2},j}$. The flux can then be computed using $F_{i-\frac{1}{2},j} = f(Q^{*}_{i-\frac{1}{2},j})$. At some point, however, we need to evaluate the flux $f(Q_{i-\frac{1}{2},j})$. For a linear flux, we get

$$\begin{aligned} f(Q_{i-\frac{1}{2},j}) =& f(\frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} q(x_{i+\frac{1}{2}},y)dy + \mathcal{O}(\Delta x^p)) \\ =& \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(q(x_{i+\frac{1}{2}},y))dy + \mathcal{O}(\Delta x^p) \\ =& \mathscr{F}_{i+\frac{1}{2},j} + \mathcal{O}(\Delta x^p), \end{aligned} \tag{3.37}$$

which is exactly what we need, the interface–averaged flux.

Using these fluxes in a finite volume method results in a p^{th}–order accurate approximation. If the flux function is nonlinear, Equation (3.37) does not hold. But since any cell–average is a 2^{nd}–order accurate approximation to

the cell–centered point value, we get

$$\begin{aligned} f(Q_{i-\frac{1}{2},j}) =& f(\frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} q(x_{i+\frac{1}{2}},y)dy + \mathcal{O}(\Delta x^p)) \\ =& f(q(x_{i+\frac{1}{2}},y_j)) + \mathcal{O}(\Delta x^p + \Delta y^2) \\ =& \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(q(x_{i+\frac{1}{2}},y))dy + \mathcal{O}(\Delta x^p + \Delta y^2) \\ =& \mathscr{F}_{i+\frac{1}{2},j} + \mathcal{O}(\Delta x^p + \Delta y^2). \end{aligned} \tag{3.38}$$

We obtained the interface–averaged flux once again, but this time only with 2^{nd}–order accuracy. Therefore, using these fluxes results in a 2^{nd}–order accurate approximation (see [1, 3, 7]). Analogously, the method can be extended to any dimension, resulting always in a 2^{nd}–order accurate approximation for nonlinear fluxes. We will refer to this approach as DIM–BY–DIM.

For sufficiently smooth solutions of hyperbolic systems in divergence form, the results of this section are summarized in the following theorem.

Theorem 3.1. *The* DIM–BY–DIM *WENO finite volume method is 2^{nd}–order accurate for general hyperbolic systems in divergence form. For linear hyperbolic systems, the* DIM–BY–DIM *approach retains the full order of the WENO reconstruction.*

3.4 The multidimensional WENO method

To avoid the loss of accuracy of the DIM–BY–DIM approach, it is crucial to evaluate fluxes only at point values. The averaged flux can then be computed by quadrature formulas at each grid cell interface. This section describes a 5^{th}– and 7^{th}–order accurate multidimensional WENO reconstruction. We will refer to this approach as MULTI–DIM.

3.4.1 The 5^{th}–order WENO method[1]

For a 5^{th}–order accurate method, we approximate

$$\begin{aligned} \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(q(x_{i+\frac{1}{2}},y))dy &\approx \frac{1}{\Delta y}\sum_{s=1}^{3} c_s f_{i+\frac{1}{2},j_s} \\ &=: F_{i+\frac{1}{2},j}, \end{aligned} \tag{3.39}$$

[1] This section was adapted from [2] with minor changes.

where

$$y_{j_1} = y_j - \frac{\Delta y}{2}\sqrt{\frac{3}{5}},\ y_{j_2} = y_j,\ y_{j_3} = y_j + \frac{\Delta y}{2}\sqrt{\frac{3}{5}} \tag{3.40}$$

are the nodes and

$$c_1 = \sqrt{\pi}/6,\ c_2 = 2\sqrt{\pi}/3,\ c_3 = \sqrt{\pi}/6. \tag{3.41}$$

are the weights of a Gaussian quadrature formula.

To get a high–order accurate approximation of the flux via Equation (3.39), we need to compute 5^{th}–order accurate point values of the flux function at the nodes of the quadrature formula. This is again done using a numerical flux formula, such as the local Lax-Friedrichs flux, i.e. we compute

$$f_{i+\frac{1}{2},j_s} = \mathcal{F}(q^-_{i+\frac{1}{2},j_s}, q^+_{i+\frac{1}{2},j_s}), \quad s = 1, 2, 3. \tag{3.42}$$

The values $q^{\pm}_{i+\frac{1}{2},j_1}, \ldots, q^{\pm}_{i+\frac{1}{2},j_3}$ are 5^{th}–order accurate approximations of point values of the conserved quantities at the nodes of the quadrature formula, i.e.

$$q^{\pm}_{i+\frac{1}{2},j_s} = q(x_{i+\frac{1}{2}}, y_{j_s}) + \mathcal{O}(\Delta x^5 + \Delta y^5).$$

To compute these left and right point values, we can use a multidimensional polynomial reconstruction of the conserved quantities. Let $p_{ij}(x, y)$ denote a 5^{th}–order accurate polynomial reconstruction of the conserved quantities in grid cell $C_{i,j}$. At a grid cell edge $(i + \frac{1}{2}, j)$, the reconstructed polynomials can be evaluated at the points $y_{j_1}, \ldots, y_{j_3}$ to compute

$$q^-_{i+\frac{1}{2},j_s} := p_{ij}(x_{i+\frac{1}{2}}, y_{j_s}), \quad q^+_{i+\frac{1}{2},j_s} := p_{i+1,j}(x_{i+\frac{1}{2}}, y_{j_s}), \quad s = 1, 2, 3. \tag{3.43}$$

On Cartesian grids, the reconstruction can be simplified. In 1993, Casper and Atkins [8] proposed a method based on the ENO scheme. But any WENO method can be applied in the same way. Instead of a two–dimensional polynomial reconstruction, one can use two one–dimensional WENO reconstructions at each edge. Let us again consider the interface $(i + \frac{1}{2}, j)$. Using a one–dimensional WENO reconstruction, we obtain a high–order accurate approximation of edge–averaged values of the conserved quantities at grid cell interfaces. In a second reconstruction step, these edge–averaged values $Q^{\pm}_{i+\frac{1}{2},j-2}, \ldots, Q^{\pm}_{i+\frac{1}{2},j+2}$ are used to construct the point values $q^{\pm}_{i+\frac{1}{2},j_1}, \ldots, q^{\pm}_{i+\frac{1}{2},j_3}$. To simplify the notation, we suppress the $\pm$–superscript in the following paragraph.

k	0	1	2
γ_k^1	$\frac{1008+71\sqrt{15}}{5240}$	$\frac{403}{655}$	$\frac{1008-71\sqrt{15}}{5240}$
γ_k^2	$-\frac{9}{80}$	$\frac{49}{40}$	$-\frac{9}{80}$
γ_k^3	$\frac{1008-71\sqrt{15}}{5240}$	$\frac{403}{655}$	$\frac{1008+71\sqrt{15}}{5240}$

k	0	1	2
γ_k^{2+}	$\frac{9}{80}$	$\frac{49}{20}$	$\frac{9}{80}$
γ_k^{2-}	$\frac{9}{40}$	$\frac{49}{40}$	$\frac{9}{40}$

Table 3.3: Weights for WENO reconstruction of point values at Gaussian nodes.

Analogously to the computation of the interface values described in Section 3.1.1, a 5^{th}–order accurate approximation of the point value at the Gaussian nodes y_{j_s} is described in the form

$$q_{i+\frac{1}{2},j_s} = \gamma_0^s q^{(0)}_{i+\frac{1}{2},j_s} + \gamma_1^s q^{(1)}_{i+\frac{1}{2},j_s} + \gamma_2^s q^{(2)}_{i+\frac{1}{2},j_s}, \tag{3.44}$$

with

$$\begin{aligned} q^{(0)}_{i+\frac{1}{2},j_s} &= a^s_{0,0} Q_{i+\frac{1}{2},j-2} + a^s_{0,1} Q_{i+\frac{1}{2},j-1} + a^s_{0,2} Q_{i+\frac{1}{2},j}, \\ q^{(1)}_{i+\frac{1}{2},j_s} &= a^s_{1,0} Q_{i+\frac{1}{2},j-1} + a^s_{1,1} Q_{i+\frac{1}{2},j} + a^s_{1,2} Q_{i+\frac{1}{2},j+1}, \\ q^{(2)}_{i+\frac{1}{2},j_s} &= a^s_{2,0} Q_{i+\frac{1}{2},j} + a^s_{2,1} Q_{i+\frac{1}{2},j+1} + a^s_{2,2} Q_{i+\frac{1}{2},j+2}. \end{aligned} \tag{3.45}$$

The values of γ are given in the left part of Table 3.3 and the coefficients $a^s_{k,l}$ are given in Table 3.4. Since γ_0^2 and γ_2^2 are negative, we cannot apply WENO limiting in the same way as in the situation of Section 3. Instead, we follow the approach of Shi et al. [27] and split γ_k^2 for $k = 0, 1, 2$ into a positive and a negative part, as shown in the right part of Table 3.3. For both parts, we can now apply WENO-JS5 (3.17) or WENO-Z5 (3.18) limiting, i.e. we compute $\omega_k^{\pm}$ using $\gamma_k^{2\pm}$, $k = 0, 1, 2$. Now, we can compute

$$q_{i+\frac{1}{2},j_{2\pm}} = \omega_0^{\pm} q^{(0)}_{i+\frac{1}{2},j_2} + \omega_1^{\pm} q^{(1)}_{i+\frac{1}{2},j_2} + \omega_2^{\pm} q^{(2)}_{i+\frac{1}{2},j_2}, \tag{3.46}$$

and finally we compute

$$q_{i+\frac{1}{2},j_2} = \sigma^+ q_{i+\frac{1}{2},j_{2+}} - \sigma^- q_{i+\frac{1}{2},j_{2-}}, \tag{3.47}$$

with

$$\sigma^+ = \sum_{k=0}^{2} \gamma_k^{2+} = \frac{214}{80} \quad \text{and} \quad \sigma^- = \sum_{k=0}^{2} \gamma_k^{2-} = \frac{67}{40}.$$

At the Gaussian nodes y_{j_1} and y_{j_3}, we can directly compute

$$q_{i+\frac{1}{2},j_s} = \omega_0 q^{(0)}_{i+\frac{1}{2},j_s} + \omega_1 q^{(1)}_{i+\frac{1}{2},j_s} + \omega_2 q^{(2)}_{i+\frac{1}{2},j_s}, \quad s = 1, 3, \tag{3.48}$$

l	0	1	2
$a^1_{0,l}$	$\frac{2-3\sqrt{15}}{60}$	$\frac{-4+12\sqrt{15}}{60}$	$\frac{62-9\sqrt{15}}{60}$
$a^1_{1,l}$	$\frac{2+3\sqrt{15}}{60}$	$\frac{56}{60}$	$\frac{2-3\sqrt{15}}{60}$
$a^1_{2,l}$	$\frac{62+9\sqrt{15}}{60}$	$\frac{-4-12\sqrt{15}}{60}$	$\frac{2+3\sqrt{15}}{60}$
$a^2_{0,l}$	$-\frac{1}{24}$	$\frac{2}{24}$	$\frac{23}{24}$
$a^2_{1,l}$	$-\frac{1}{24}$	$\frac{26}{24}$	$-\frac{1}{24}$
$a^2_{2,l}$	$\frac{23}{24}$	$\frac{2}{24}$	$-\frac{1}{24}$
$a^3_{0,l}$	$\frac{2+3\sqrt{15}}{60}$	$\frac{-4-12\sqrt{15}}{60}$	$\frac{62+9\sqrt{15}}{60}$
$a^3_{1,l}$	$\frac{2-3\sqrt{15}}{60}$	$\frac{56}{60}$	$\frac{2+3\sqrt{15}}{60}$
$a^3_{2,l}$	$\frac{62-9\sqrt{15}}{60}$	$\frac{-4+12\sqrt{15}}{60}$	$\frac{2-3\sqrt{15}}{60}$

Table 3.4: Coefficients for WENO reconstruction of point values at Gaussian nodes.

with $\omega_0, \omega_1, \omega_2$ computed as in (3.17) or (3.18).

In order to compute the six point values of the conserved quantities $q^{\pm}_{i+\frac{1}{2},j_1}, q^{\pm}_{i+\frac{1}{2},j_2}, q^{\pm}_{i+\frac{1}{2},j_3}$, the above formulas are evaluated both for the left and the right interface–averaged values of the conserved quantities. Finally, these pairs of point values are used to compute the interface flux using (3.39) and (3.42). Analogously, we compute fluxes $G_{i,j+\frac{1}{2}}$ in the y direction.

3.4.2 The 7^{th}–order WENO method[2]

To guarantee 7^{th}–order accuracy, we use the Gauss quadrature rule with four points

$$\begin{aligned}\frac{1}{\Delta x}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f(q(x_{i+\frac{1}{2}},y))dy &\approx \frac{1}{\Delta x}\sum_{s=1}^{4} c_s f_{i+\frac{1}{2},j_s} \\ &=: F_{i+\frac{1}{2},j},\end{aligned} \tag{3.49}$$

[2]This section was adapted from [2] with minor changes.

where

$$y_{j_1} = y_j - \frac{\Delta y}{2}\sqrt{\frac{3}{7} + \frac{2}{7}\sqrt{\frac{6}{5}}}, \quad y_{j_2} = y_j - \frac{\Delta y}{2}\sqrt{\frac{3}{7} - \frac{2}{7}\sqrt{\frac{6}{5}}},$$
$$y_{j_3} = y_j + \frac{\Delta y}{2}\sqrt{\frac{3}{7} - \frac{2}{7}\sqrt{\frac{6}{5}}}, \quad y_{j_4} = y_j + \frac{\Delta y}{2}\sqrt{\frac{3}{7} + \frac{2}{7}\sqrt{\frac{6}{5}}}$$

are the nodes and

$$c_1 = \frac{18 - \sqrt{30}}{36},\ c_2 = \frac{18 + \sqrt{30}}{36},\ c_3 = \frac{18 + \sqrt{30}}{36},\ c_4 = \frac{18 - \sqrt{30}}{36} \tag{3.50}$$

are the weights of the quadrature formula. Analogously to Section 3.1.2, a 7^{th}–order accurate approximation of the point value at the Gaussian node $y_{j_s}, s = 1, ..., 4$ is described in the form

$$q_{i+\frac{1}{2},j_s} = \gamma_0^s q^{(0)}_{i+\frac{1}{2},j_s} + \gamma_1^s q^{(1)}_{i+\frac{1}{2},j_s} + \gamma_2^s q^{(2)}_{i+\frac{1}{2},j_s} + \gamma_3^s q^{(3)}_{i+\frac{1}{2},j_s}$$

with

$$\begin{aligned}
q^{(0)}_{i+\frac{1}{2},j_s} &= a^s_{0,0} Q_{i+\frac{1}{2},j-3} + a^s_{0,1} Q_{i+\frac{1}{2},j-2} + a^s_{0,2} Q_{i+\frac{1}{2},j-1} + a^s_{0,3} Q_{i+\frac{1}{2},j},\\
q^{(1)}_{i+\frac{1}{2},j_s} &= a^s_{1,0} Q_{i+\frac{1}{2},j-2} + a^s_{1,1} Q_{i+\frac{1}{2},j-1} + a^s_{1,2} Q_{i+\frac{1}{2},j} + a^s_{1,3} Q_{i+\frac{1}{2},j+1},\\
q^{(2)}_{i+\frac{1}{2},j_s} &= a^s_{2,0} Q_{i+\frac{1}{2},j-1} + a^s_{2,1} Q_{i+\frac{1}{2},j} + a^s_{2,2} Q_{i+\frac{1}{2},j+1} + a^s_{2,3} Q_{i+\frac{1}{2},j+2},\\
q^{(3)}_{i+\frac{1}{2},j_s} &= a^s_{3,0} Q_{i+\frac{1}{2},j} + a^s_{3,1} Q_{i+\frac{1}{2},j+1} + a^s_{3,2} Q_{i+\frac{1}{2},j+2} + a^s_{3,3} Q_{i+\frac{1}{2},j+3}.
\end{aligned} \tag{3.51}$$

The values of γ_k^s and $a^s_{k,l}$ are given in Tables 3.5 and 3.6. The limited values at the Gaussian nodes are given by

$$q_{i+\frac{1}{2},j_s} = \omega_0 q^{(0)}_{i+\frac{1}{2},j_s} + \omega_1 q^{(1)}_{i+\frac{1}{2},j_s} + \omega_2 q^{(2)}_{i+\frac{1}{2},j_s} + \omega_3 q^{(3)}_{i+\frac{1}{2},j_s}$$

where ω_k is computed analogously as in Equation (3.22) or (3.23). As discussed in Section 3.4.1, we evaluate these formulas for the left and the right interface–averaged values of the conserved quantities. Finally, these pairs of points are used for the flux evaluation in Equation (3.49). Analogously, we compute the fluxes $G_{i,j+\frac{1}{2}}$.

k	0	1	2	3
γ_k^1	0.097897339374826219 5	0.494589300173797349	0.370641008922792546	0.036872351528583886 0
γ_k^2	0.042216057022944471 7	0.348843639132503842	0.430804627949651975	0.178135675894899710
γ_k^3	0.178135675894899710	0.430804627949651975	0.348843639132503842	0.042216057022944471 7
γ_k^4	0.036872351528583886 0	0.370641008922792546	0.494589300173797349	0.097897339374826219 5

Table 3.5: Weights for WENO-7 reconstruction of point values at Gaussian nodes.

l	0	1	2	3
$a^1_{0,l}$	0.087853391504588917	−0.427831293623330335	1.02265572559231028	0.317317228880561163
$a^1_{1,l}$	−0.0763979370214947678	0.495505690689556931	0.668750585482396728	−0.087853391504588914
$a^1_{2,l}$	0.189913942603577861	1.12713820761136533	−0.393450087236437961	0.0763979370214947678
$a^1_{3,l}$	1.8867939780256767 8	−1.53293374285790512	0.836053707435806209	−0.189913942603577861
$a^2_{0,l}$	0.0776175431540304203	−0.345066168275341003	0.627270228881018877	0.640178396240291707
$a^2_{1,l}$	−0.0345959956592193208	0.161564969956836354	0.950648568856413386	−0.0776175431540304203
$a^2_{2,l}$	0.0231809873199590709	1.15822454281172931	−0.216001525790907703	0.0345959956592193208
$a^2_{3,l}$	1.25094849209156560	−0.355087449710662129	0.127319944939055605	−0.0231809873199590709
$a^3_{0,l}$	−0.0231809873199590709	0.127319944939055605	−0.355087449710662129	1.25094849209156560
$a^3_{1,l}$	0.0345959956592193208	−0.216001525790907703	1.15822454281172931	0.0231809873199590709
$a^3_{2,l}$	−0.0776175431540304203	0.950648568856413386	0.161564969956836354	−0.0345959956592193208
$a^3_{3,l}$	0.640178396240291707	0.627270228881018877	−0.345066168275341003	0.0776175431540304203
$a^4_{0,l}$	−0.189913942603577861	0.836053707435806209	−1.53293374285790512	1.8867939780256767 8
$a^4_{1,l}$	0.0763979370214947678	−0.393450087236437961	1.12713820761136533	0.189913942603577861
$a^4_{2,l}$	−0.087853391504588914	0.668750585482396728	0.495505690689556931	−0.0763979370214947678
$a^4_{3,l}$	0.317317228880561163	1.02265572559231028	−0.427831293623330335	0.087853391504588917

Table 3.6: Coefficients for WENO-7 reconstruction of point values at Gaussian nodes.

3.5 WENO reconstruction in 3D

The dimension–by–dimension approach

In 3D, the interface is no longer a one–dimensional object but a two–dimensional one. For the DIM–BY–DIM WENO reconstruction discussed in Section 3.3, nothing changes. A two–dimensional cell–average is still a 2^{nd}–order approximation to the center of the cell and the other way around. Therefore, the extension to 3D or even higher dimensions is straightforward.

The multidimensional approach

The MULTI–DIM WENO reconstruction of Section 3.4 can also be extended to 3D easily. Assuming we use a quadrature rule with n nodes at the interface, the algorithm is

1. Reconstruct interface values $Q^{\pm}_{i+\frac{1}{2},j,k}$ in the x direction.
2. Apply the reconstruction in the y direction on the interfaces to get values at the quadrature nodes $Q^{\pm}_{i+\frac{1}{2},j_m,k}$, $1 \leq m \leq n$.
3. Apply the reconstruction in the z direction on each quadrature node to get the point values $q^{\pm}_{i+\frac{1}{2},j_m,k_s}$, $1 \leq m, s \leq n$.
4. Compute flux point values $f_{i-\frac{1}{2},j_m,k_s} = \mathcal{F}(q^{-}_{i-\frac{1}{2},j_m,k_s}, q^{+}_{i-\frac{1}{2},j_m,k_s})$.
5. Use the quadrature rule to get the flux average $F_{i+\frac{1}{2},j,k} = \sum_{m=1}^{n} \sum_{s=1}^{n} c_m c_s f_{i-\frac{1}{2},j_m,k_s}$.

Analogously, we compute the fluxes G and H in the y and z directions. The difference with the 3D case is that, after Step 2, the values are still averaged in one direction. Therefore, we have to apply another reconstruction step to get the point values. In the end, we have n^2 point values at every interface.

Chapter 4

A new approach for a high–order WENO finite volume method in 2D

We now describe a modification of the DIM–BY–DIM WENO method, which avoids the accuracy drop described in Section 3.5. With this modification, the full order of accuracy can be retained for multidimensional nonlinear problems. The method is computationally less expensive than the methods described in Section 3.4, since the computational expensive WENO reconstruction is performed only once. Furthermore, the new approach requires only one evaluation of the flux function per interface.

The WENO reconstruction provides high–order accurate approximations of averaged values of the conserved quantities at grid cell interfaces. As discussed in Section 3.5 with respect to conservation laws with nonlinear flux functions, we cannot compute high–order accurate averaged values of the interface flux directly from these edge–averaged values.

Instead, we first compute the point values of the conserved quantities at the center of the grid cell interfaces. The numerical fluxes at these point values can then be computed without loss of accuracy. Finally, we compute averaged values of the fluxes at grid cell interfaces. Similar transformations were used by several authors in a different context (see [11, 28, 29, 30]).

These transformations will be based on Theorem 4.1. To ensure that the theorem remains generic, we first introduce some additional notations. In a d–dimensional space, we use the typical multi–index notation $\boldsymbol{i} = (i_1, ..., i_d) \in \mathbb{Z}^d, \boldsymbol{n} = (n_1, ..., n_d) \in \mathbb{N}^d, \boldsymbol{x} = (x_1, ..., x_d) \in \mathbb{R}^d$, with $\boldsymbol{n}! = n_1! \cdots n_d!$, $|\boldsymbol{n}| = n_1 + \cdots + n_d$ and $\boldsymbol{x}^{\boldsymbol{n}} = x_1^{n_1} \cdots x_d^{n_d}$. In this sense, it is $\mathbf{1} = (1, ..., 1) \in \mathbb{N}^d$, $\boldsymbol{\Delta x} = (\Delta x_1, ..., \Delta x_d) \in \mathbb{R}^d$, $\boldsymbol{\Delta x}^{\mathbf{1}} = \Delta x_1 \cdots \Delta x_d$ and so on. Furthermore, we

denote by $C_{\boldsymbol{i}}$ a grid cell of the form $C_{\boldsymbol{i}} = \left[x_{i_1-\frac{1}{2}}, x_{i_1+\frac{1}{2}}\right] \times \cdots \times \left[x_{i_d-\frac{1}{2}}, x_{i_d+\frac{1}{2}}\right]$ with mesh width $\boldsymbol{\Delta x} = (\Delta x_1, ..., \Delta x_d)$.

The Taylor series in several variables is discussed in many textbooks (e.g. [31]). With the notation above, we can write the multidimensional Taylor series expansion of a sufficiently smooth function $f : \mathbb{R}^d \to \mathbb{R}$ for $\boldsymbol{x}, \boldsymbol{a} \in \mathbb{R}^d$. in a very compact form

$$T_f(\boldsymbol{x}) = \sum_{\boldsymbol{n} \in \mathbb{N}^d} \frac{(\boldsymbol{x} - \boldsymbol{a})^{\boldsymbol{n}}}{\boldsymbol{n}!} \frac{\partial^{|\boldsymbol{n}|} f(\boldsymbol{a})}{\boldsymbol{\partial x^n}}. \tag{4.1}$$

Theorem 4.1. *Let $q : \mathbb{R}^d \to \mathbb{R}$ be a sufficient smooth function and $Q_{\boldsymbol{i}} = \frac{1}{\boldsymbol{\Delta x^1}} \int_{C_{\boldsymbol{i}}} q(\boldsymbol{x}) d\boldsymbol{x}$ the averaged value of q over cell $C_{\boldsymbol{i}}$. Then the equation*

$$Q_{\boldsymbol{i}} = q(\boldsymbol{x_i}) + \sum_{\substack{1<|\boldsymbol{n}|<N \\ n_i \text{ even } \forall i}} \frac{\boldsymbol{\Delta x^n}}{\boldsymbol{n}!(\boldsymbol{n}+\boldsymbol{1})^{\boldsymbol{1}}\boldsymbol{2^n}} \frac{\partial^{|\boldsymbol{n}|} q(\boldsymbol{x_i})}{\boldsymbol{\partial x^n}} + \mathcal{O}(\Delta x^N), \tag{4.2}$$

holds with $\Delta x = \max\{\Delta x_1, ..., \Delta x_d\}$.

Proof. Without loss of generality, we assume $\Delta x_1 = \cdots = \Delta x_d = \Delta x$ and $\boldsymbol{x_i} = \boldsymbol{0}$. Then, we first note that for $\boldsymbol{n} = (n_1, ..., n_d) \in \mathbb{Z}^d$ we have

$$\begin{aligned}
\frac{1}{\boldsymbol{\Delta x^1}} \int_{C_{\boldsymbol{i}}} \boldsymbol{x^n} d\boldsymbol{x} &= \frac{1}{\Delta x^d} \int_{-\frac{\Delta x}{2}}^{\frac{\Delta x}{2}} \cdots \int_{-\frac{\Delta x}{2}}^{\frac{\Delta x}{2}} x_1^{n_1} \cdots x_d^{n_d} dx_1 \cdots dx_d \\
&= \prod_{i=1}^{d} \left(\frac{1}{\Delta x} \int_{-\frac{\Delta x}{2}}^{\frac{\Delta x}{2}} x_i^{n_i} dx_i \right) \\
&= \prod_{i=1}^{d} \left(\frac{1}{\Delta x} \frac{1}{n_i + 1} \left(\left(\frac{\Delta x}{2} \right)^{n_i+1} - \left(-\frac{\Delta x}{2} \right)^{n_i+1} \right) \right) \\
&= \begin{cases} \prod_{i=1}^{d} \frac{1}{n_i+1} \left(\frac{\Delta x}{2} \right)^{n_i} & \text{if } n_i \text{ even } \forall i \\ 0 & \text{else} \end{cases} \\
&= \begin{cases} \frac{\boldsymbol{\Delta x^n}}{(\boldsymbol{n}+\boldsymbol{1})^{\boldsymbol{1}}\boldsymbol{2^n}} & \text{if } n_i \text{ even } \forall i \\ 0 & \text{else.} \end{cases}
\end{aligned} \tag{4.3}$$

Using (4.3) and the multidimensional Taylor series expansion (4.1), we obtain

$$
\begin{aligned}
Q_{\boldsymbol{i}} &= \frac{1}{\boldsymbol{\Delta x^1}} \int_{C_{\boldsymbol{i}}} q(\boldsymbol{x}) d\boldsymbol{x} \\
&= \frac{1}{\boldsymbol{\Delta x^1}} \int_{C_{\boldsymbol{i}}} \sum_{0 \le |\boldsymbol{n}| < N} \frac{(\boldsymbol{x}-\boldsymbol{x_i})^{\boldsymbol{n}}}{\boldsymbol{n}!} \frac{\partial^{|\boldsymbol{n}|} q(\boldsymbol{x_i})}{\partial \boldsymbol{x^n}} + \mathcal{O}(\Delta x^N) d\boldsymbol{x} \\
&\overset{(\boldsymbol{x_i}=0)}{=} \sum_{0 \le |\boldsymbol{n}| < N} \frac{1}{\boldsymbol{n}!} \left(\frac{1}{\boldsymbol{\Delta x^1}} \int_{C_{\boldsymbol{i}}} \boldsymbol{x^n} d\boldsymbol{x} \right) \frac{\partial^{|\boldsymbol{n}|} q(\boldsymbol{x_i})}{\partial \boldsymbol{x^n}} + \mathcal{O}(\Delta x^N) \\
&= q(\boldsymbol{x_i}) + \sum_{\substack{1 < |\boldsymbol{n}| < N \\ \boldsymbol{n} \in 2\mathbb{N}^d}} \frac{\boldsymbol{\Delta x^n}}{\boldsymbol{n}!(\boldsymbol{n}+\boldsymbol{1})\boldsymbol{1}\boldsymbol{2^n}} \frac{\partial^{|\boldsymbol{n}|} q(\boldsymbol{x_i})}{\partial \boldsymbol{x^n}} + \mathcal{O}(\Delta x^N).
\end{aligned}
\tag{4.4}
$$

□

Similar transformations are also possible in mapped coordinates, as shown by Colella et al. [29].

4.1 Transformation between average values and point values[3]

We discuss the transformation between average values and point values for functions of one spatial variable. This is in line with the situation that will later be used in our 2D method, since the second variable at grid cell interfaces will just lead to an additional index. To simplify the notation, we suppress the time dependence of the functions in this section.

We denote with Q_i an approximation of the cell–average of the function q in grid cell $C_i = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$, and by q_i an approximation of the point value $q(x_i)$ of the quantities q at the midpoint x_i of the grid cell. Applying Theorem 4.1 in 1D, we obtain the transformation

$$
q_i = Q_i - \frac{\Delta x^2}{24} q''(x_i) - \frac{\Delta x^4}{1920} q^{(4)}(x_i) + \mathcal{O}(\Delta x^6) \tag{4.5}
$$

between point values and cell–averaged values.

Thus, we need expressions for the approximation of the second and fourth derivatives. In order to transform from point values to cell–averaged values, we can approximate these derivatives using standard finite difference formulas. If we transform from cell–averaged values to point values, we use cell–averaged values of the conserved quantities to approximate the second and fourth derivatives at the midpoint of the interval.

[3] This section, except for the last part, was adapted from [1] with some amendments.

Approximation of derivatives from point values

The second derivative $q^{''}(x_i)$ can be approximated using point values of the quantity q via the well–known second–order accurate finite difference formula

$$q^{''}(x_i) = \frac{q_{i-1} - 2q_i + q_{i+1}}{\Delta x^2} + \mathcal{O}(\Delta x^2). \tag{4.6}$$

A 4^{th}–order accurate representation of $q^{''}(x_i)$ can be obtained using the formula

$$q^{''}(x_i) = \frac{-q_{i-2} + 16q_{i-1} - 30q_i + 16q_{i+1} - q_{i+2}}{12\Delta x^2} + \mathcal{O}(\Delta x^4). \tag{4.7}$$

A second–order accurate representation of $q^{(4)}(x_i)$ can be computed from point values using the finite difference formula

$$q^{(4)}(x_i) = \frac{q_{i-2} - 4q_{i-1} + 6q_i - 4q_{i+1} + q_{i+2}}{\Delta x^4} + \mathcal{O}(\Delta x^2). \tag{4.8}$$

All of these formulas can be verified using Taylor series expansion.

Approximation of derivatives from cell–averaged values

Second–order accurate approximations of $q''(x_i)$ and $q^{(4)}(x_i)$ can be obtained using formulas analogously to (4.6) and (4.8) with the point values q_i replaced by average values Q_i.

A 4^{th}–order accurate approximation of $q^{''}(x_i)$ can be computed from cell–averaged values via the formula

$$q^{''}(x_i) = \frac{-Q_{i-2} + 12Q_{i-1} - 22Q_i + 12Q_{i+1} - Q_{i+2}}{8\Delta x^2} + \mathcal{O}(\Delta x^4). \tag{4.9}$$

This formula can also be verified by Taylor series expansion, after using (4.5) to express the cell–averaged values with the help of point values.

Note that Equations (4.6)–(4.9) are only valid for uniform Cartesian grids.

Transformation between average values and point values

Formula (4.6) is sufficient to derive a 4^{th}–order accurate transformation between average values and point values. With Equations (4.5) and (4.6), we get

$$\begin{aligned} Q_i - \frac{Q_{i-1} - 2Q_i + Q_{i+1}}{24} &= Q_i - \frac{\Delta x^2}{24}\left(q^{''}(x_i) + \mathcal{O}(\Delta x^2)\right) \\ &= Q_i - \frac{\Delta x^2}{24} q^{''}(x_i) + \mathcal{O}(\Delta x^4) \\ &= q_i + \mathcal{O}(\Delta x^4). \end{aligned} \tag{4.10}$$

Analogously, we get

$$q_i - \frac{q_{i-1} - 2q_i + q_{i+1}}{24} = Q_i + \mathcal{O}(\Delta x^4). \tag{4.11}$$

To compute a 6^{th}–order accurate approximation of a point value from averaged values, we use the formulas (4.8) and (4.9). In contrast, when computing a 6^{th}–order accurate approximation of an averaged value from point values, we use the formulas (4.8) and (4.7).

4.2 Modification of the dimension–by–dimension WENO method

The considerations of the previous section suggest the following modification of the DIM–BY–DIM WENO method.

Algorithm: Modified dimension–by–dimension WENO method

1. Compute averaged values of the conserved quantities at grid cell interfaces using the DIM–BY–DIM WENO approach, i.e. compute

   $$Q^{\pm}_{i-\frac{1}{2},j},\ Q^{\pm}_{i,j-\frac{1}{2}}$$

   at all grid cell interfaces.

2. Compute point values of the conserved quantities at the center of each grid cell interface, i.e. compute

   $$\begin{aligned} q^{\pm}_{i+\frac{1}{2},j} &= Q^{\pm}_{i+\frac{1}{2},j} - \frac{1}{24}\left(Q^{\pm}_{i+\frac{1}{2},j-1} - 2Q^{\pm}_{i+\frac{1}{2},j} + Q^{\pm}_{i+\frac{1}{2},j+1}\right) \\ q^{\pm}_{i,j+\frac{1}{2}} &= Q^{\pm}_{i,j+\frac{1}{2}} - \frac{1}{24}\left(Q^{\pm}_{i-1,j+\frac{1}{2}} - 2Q^{\pm}_{i,j+\frac{1}{2}} + Q^{\pm}_{i+1,j+\frac{1}{2}}\right). \end{aligned} \tag{4.12}$$

3. Compute fluxes at the center of the grid cell interfaces using a consistent numerical flux function, i.e.

   $$f_{i-\frac{1}{2},j} = \mathcal{F}(q^{-}_{i-\frac{1}{2},j}, q^{+}_{i-\frac{1}{2},j}), \quad g_{i,j-\frac{1}{2}} = \mathcal{F}(q^{-}_{i,j-\frac{1}{2}}, q^{+}_{i,j-\frac{1}{2}})$$

4. Compute averaged values of the flux, i.e. compute

   $$\begin{aligned} F_{i+\frac{1}{2},j} &= f_{i+\frac{1}{2},j} + \frac{1}{24}\left(f_{i+\frac{1}{2},j-1} - 2f_{i+\frac{1}{2},j} + f_{i+\frac{1}{2},j+1}\right) \\ G_{i,j+\frac{1}{2}} &= g_{i,j+\frac{1}{2}} + \frac{1}{24}\left(g_{i-1,j+\frac{1}{2}} - 2g_{i,j+\frac{1}{2}} + g_{i+1,j+\frac{1}{2}}\right). \end{aligned} \tag{4.13}$$

5. Approximate the semi–discrete system (3.2), using a high–order accurate Runge–Kutta method.

Using any WENO method described in Section 3.1, this method will be 4^{th}–order accurate for both linear and nonlinear flux functions. We will refer to this approach as MODIFIED–4. To get an even higher order of accuracy, we can replace Equations (4.12) and (4.13) with

$$q^{\pm}_{i+\frac{1}{2},j} = Q^{\pm}_{i+\frac{1}{2},j} - \sum_{\nu=-2}^{2} \alpha_\nu Q^{\pm}_{i+\frac{1}{2},j+\nu} \tag{4.14}$$

and

$$F_{i+\frac{1}{2},j} = f_{i+\frac{1}{2},j} + \sum_{\nu=-2}^{2} \beta_\nu Q^{\pm}_{i+\frac{1}{2},j+\nu} \tag{4.15}$$

where the coefficients α_ν and β_ν are given in Table 4.1. We will refer to this approach as MODIFIED–6. Note that the formulas in Equations (4.14) and (4.15) include both the approximation of the 2^{nd} as well as the 4^{th} derivative. Combined with one of the 5^{th}–order WENO reconstructions, this leads to a 5^{th}–order accurate method, and using one of the 7^{th}–order WENO reconstructions, this leads to a 6^{th}–order accurate method.

ν	-2	-1	0	1	2
α_ν	$-\frac{3}{640}$	$\frac{29}{480}$	$-\frac{107}{960}$	$\frac{29}{480}$	$-\frac{3}{640}$
β_ν	$-\frac{17}{5760}$	$\frac{77}{1440}$	$-\frac{97}{960}$	$\frac{77}{1440}$	$-\frac{17}{5760}$

Table 4.1: Coefficients for the 6^{th}–order transformation formulas.

Assuming that the solution is sufficiently smooth and the WENO reconstruction sufficiently accurate, we have derived the following result.

Theorem 4.2. *The* MODIFIED–4 *WENO finite volume method is* 4^{th}*–order accurate and the* MODIFIED–6 *WENO finite volume method is* 6^{th}*–order accurate for general hyperbolic systems in divergence form.*

4.3 Approximation of shock waves and contact discontinuities[4]

The derivation of our modified dimension–by–dimension WENO–FV method is based on Taylor series expansion, i.e. it is assumed that the solution we

[4] This section was adapted from [2] with minor changes.

wish to approximate is sufficiently smooth. However, it is well known that the solution of hyperbolic conservation laws can contain shock waves and contact discontinuities along which the solution even becomes discontinuous. For discontinuous solutions, it is easy to construct situations in which the transformation (4.12) or (4.14) from cell–averaged values to point values of the conserved quantities produces unphysical results, for example, a negative density or a negative pressure for the Euler equations of gas dynamics.

To avoid such problems, we used a simple approach that reduces our method to the classical DIM–BY–DIM approach near shock waves. For a grid cell interface in the x direction, we rewrite the transformation (4.12) or (4.14) in the form

$$q^{\pm}_{i+\frac{1}{2},j} = Q^{\pm}_{i+\frac{1}{2},j} - \mathcal{D}^{\pm}_{i+\frac{1}{2},j}. \tag{4.16}$$

If the change described by $\mathcal{D}$ is *too large*, we switch to the DIM–BY–DIM approach. This is implemented in the following way:

1. Compute $\mathcal{D}^{\pm}_{i+\frac{1}{2},j}$ for all components of the conserved quantities (recall that $q \in \mathbb{R}^m$).
2. If $|Q^{\pm}_{i+\frac{1}{2},j}| < 2|\mathcal{D}^{\pm}_{i+\frac{1}{2},j}|$ and $|\mathcal{D}^{\pm}_{i+\frac{1}{2},j}| > 10^{-15}$ (in any component) then
   set $q^{\pm}_{i+\frac{1}{2},j} = Q^{\pm}_{i+\frac{1}{2},j}$
   else set $q^{\pm}_{i+\frac{1}{2},j} = Q^{\pm}_{i+\frac{1}{2},j} - \mathcal{D}^{\pm}_{i+\frac{1}{2},j}$
3. Compute fluxes $f_{i+\frac{1}{2},j} = \mathcal{F}(q^{-}_{i+\frac{1}{2},j}, q^{+}_{i+\frac{1}{2},j})$
4. Use transformation (4.13) or (4.15) only at those grid cell interfaces where we have used the transformation form edge–averaged values to point values in Step 2.

This simple extension of our MODIFIED–4/MODIFIED–6 approach was only used in our test simulations of the double Mach reflection problem in Section 5.2.2 and the cloud–shock interaction problem in Section 6.2.

4.4 The stencil[5]

In Figures 4.1 and 4.2, we show the stencil used in one time stage of our MODIFIED–4 method with 5^{th}–order WENO reconstruction. In the left part of Figure 4.1, we show the stencil that is used to compute edge–averaged values of the conserved quantities marked as two dashed lines. In a standard

[5] This section was adapted from [1] with minor changes.

DIM–BY–DIM approach, those edge–averaged values are used to compute the interface flux. In our MODIFIED–4 method, we compute point values of the conserved quantities using Equation (4.12). For this computation, we need neighboring edge–averaged values. This enlarges the stencil, as indicated in the right part of Figure 4.1. The point values of the conserved quantity (indicated by black dots) are used to compute point values of the flux (indicated by the open ellipse.)

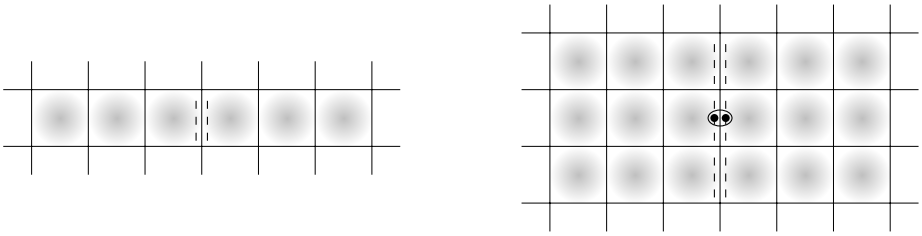


Figure 4.1: The plot on the left shows the stencil for the computation of $Q^{\pm}_{i-\frac{1}{2},j}$. These averaged interface values are indicated by the two dashed lines. The right plot shows the stencil for the computation of the point values $q^{\pm}_{i-\frac{1}{2},j}$, indicated as black dots in the figure. Those point values are used to compute point values of the flux, denoted by $f_{i-\frac{1}{2},j}$. The point value of the flux is marked as an open ellipse.

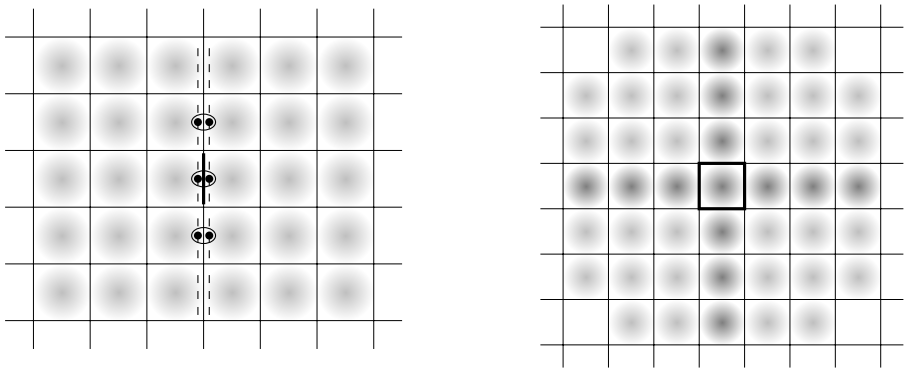


Figure 4.2: The plot on the left shows the stencil used to compute the cell–averaged value of the flux at the interface, i.e. $F_{i-\frac{1}{2},j}$. This flux is marked as a dark solid line. The plot on the right shows the complete stencil used to update the grid cell at the center. The dark shaded cells are used in a classical DIM–BY–DIM approach.

In the plot on the left of Figure 4.2, we show the stencil needed to compute edge–averaged values of the flux according to Equation (4.13). This transformation requires neighboring point values of fluxes, which further enlarges the stencil. In the plot on the right of Figure 4.2, we show the full stencil of cells used to update one cell. The dark shaded grid cells are those used by the DIM–BY–DIM WENO method. After computing all fluxes (of the MODIFIED–4 method) for one cell, most of the work for the neighboring cells is already done. Therefore, the larger stencil only leads to a relatively small increase of the computational costs, as shown below in Table 5.3.

We can replace the flux computation in (4.13) by using a formula of the form

$$F_{i-\frac{1}{2},j} = f_{i-\frac{1}{2},j} + \frac{1}{24}\left(\bar{f}_{i-\frac{1}{2},j-1} - 2\bar{f}_{i-\frac{1}{2},j} + \bar{f}_{i-\frac{1}{2},j+1}\right), \tag{4.17}$$

and analogously for $G_{i,j-\frac{1}{2}}$, where $\bar{f}_{i-\frac{1}{2},k} = \mathcal{F}(Q^-_{i-\frac{1}{2},k}, Q^+_{i-\frac{1}{2},k})$, $k = j \pm 1, j$, is a flux computed using the averaged values of the conserved quantities. The resulting finite volume method has a more local stencil (for WENO-5, the 16 outermost light shaded cells in the plot on the right of Figure 4.2 would not be used). However, this approach requires the computation of two fluxes per interface. We have also tested such versions of the method and obtained good results. These computations will not be presented here.

4.5 Extension to 3D

In 3D, the interfaces between grid cells are two–dimensional objects. Therefore, to apply the new method in 3D, we have to find a transformation between face–averaged values and point values for functions of two spatial variables. Applying Theorem 4.1 in 2D, we obtain the transformation

$$\begin{aligned} q_{ij} =& Q_{ij} - \frac{\Delta x^2}{24} q_{xx}(x_i, y_j) - \frac{\Delta y^2}{24} q_{yy}(x_i, y_j) \\ & - \frac{\Delta x^4}{1920} q_{xxxx}(x_i, y_j) - \frac{\Delta x^2 \Delta y^2}{576} q_{xxyy}(x_i, y_j) - \frac{\Delta y^4}{1920} q_{yyyy}(x_i, y_j) \\ & + \mathcal{O}(\Delta x^6 + \Delta y^6) \end{aligned} \tag{4.18}$$

between point values and cell face–averaged values. Using Equation (4.18), it is straightforward to construct a 4^{th}–order accurate transformation. Just apply the finite difference formula from (4.6) in the x and y directions.

For the 6^{th}–order accurate transformation, we can also compute the approximations to the 2^{nd} derivative in the x and y directions, as well as 4^{th} derivative in the x and y directions, by applying the formulas of Section 4.1 in each direction. But there is another term $\frac{\Delta x^2 \Delta y^2}{576} q_{xxyy}(x_i, y_j)$; therefore,

we need an approximation to $q_{xxyy}(x_i, y_j)$. Since the term is multiplied with $\frac{\Delta x^2 \Delta y^2}{576}$, a 2^{nd}–order accurate approximation is sufficient here. As in Section 4.1, we can compute the derivative using a finite difference formula, i.e.

$$\begin{aligned} q_{xxyy}(x_i, y_j) = & \frac{1}{\Delta x^2 \Delta y^2} \left(q_{i-1,j+1} + q_{i+1,j+1} + q_{i-1,j-1} + q_{i+1,j-1}\right. \\ & \left. -2(q_{i,j+1} + q_{i,j-1} + q_{i+1,j} + q_{i-1,j}) + 4q_{i,j}\right) \\ & + \mathcal{O}(\Delta x^2 + \Delta y^2). \end{aligned} \tag{4.19}$$

For the approximation of the derivative from cell–averaged values, the same formula can be used since we need only 2^{nd}–order accuracy here.

4.6 The methods used in this work

So far, we have presented different reconstruction methods in 1D and different ways of extending these methods to higher dimensions. Several combinations of the methods are possible. In this section, we give an overview of the combinations used in this work. In Section 3.1, we presented two 5^{th}–order methods to which we refer as WENO-JS5 and WENO-Z5, and two 7^{th}–order methods to which we refer as WENO-JS7 and WENO-Z7.

Each of the methods can be applied in a dimension–by–dimension fashion, as discussed in Section 3.3. Such a combination will be denoted as DIM–BY–DIM.

In Section 3.4, we presented a multidimensional reconstruction. The 5^{th}–order methods are combined with a Gaussian quadrature formula of order 6 and the 7^{th}–order methods are combined with a Gaussian quadrature formula of order 8. Thus, we formally retain the full order of accuracy of the underlying WENO reconstruction. These methods will be denoted as MULTI–DIM.

Finally, in Section 4.2, we presented a new approach to retain the high–order of accuracy. The 4^{th}–order transformation will be used mainly with a 5^{th}–order WENO reconstruction and denoted as MODIFIED–4 . The 6^{th}–order transformation will be denoted as MODIFIED–6 and used mainly with a 7^{th}–order WENO reconstruction.

In Table 4.2, an overview is given of the theoretical order of accuracy of the different combinations, when applied to linear problems in higher dimensions. When applied to nonlinear problems in higher dimensions, the theoretical order of accuracy is given in Table 4.3.

	WENO-JS5	WENO-Z5	WENO-JS7	WENO-Z7
DIM–BY–DIM	5^{th}–order	5^{th}–order	7^{th}–order	7^{th}–order
MULTI–DIM	5^{th}–order	5^{th}–order	7^{th}–order	7^{th}–order
MODIFIED–4	4^{th}–order	4^{th}–order	4^{th}–order	4^{th}–order
MODIFIED–6	5^{th}–order	5^{th}–order	6^{th}–order	6^{th}–order

Table 4.2: Theoretical order of accuracy for linear multidimensional problems.

	WENO-JS5	WENO-Z5	WENO-JS7	WENO-Z7
DIM–BY–DIM	2^{nd}–order	2^{nd}–order	2^{nd}–order	2^{nd}–order
MULTI–DIM	5^{th}–order	5^{th}–order	7^{th}–order	7^{th}–order
MODIFIED–4	4^{th}–order	4^{th}–order	4^{th}–order	4^{th}–order
MODIFIED–6	5^{th}–order	5^{th}–order	6^{th}–order	6^{th}–order

Table 4.3: Theoretical order of accuracy for nonlinear multidimensional problems.

4.7 Comparing the methods in 2D

In this section, we test and compare the methods introduced so far by applying them to the Euler equations of gas dynamics in 2D.

The two–dimensional Euler equations are given by

$$\partial_t \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} + \partial_x \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix} + \partial_y \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix} = 0, \tag{4.20}$$

with the total energy density

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho \left(u^2 + v^2\right), \tag{4.21}$$

where ρ is the density, u the velocity in the x direction, v the velocity in the y direction, and p the pressure.

4.7.1 Linear problem

Example 4.1. *We consider periodic solutions of* (4.20) *on the computational domain* $[0,1] \times [0,1]$. *The initial values are given by*

$$\begin{aligned} \rho(x,y,0) &= 1 + 0.5\sin(\kappa 2\pi x)\cos(\kappa 2\pi y) \\ p(x,y,0) &= 1 \\ u(x,y,0) &= v(x,y,0) = 1. \end{aligned} \tag{4.22}$$

In this case, velocity and pressure remain constant for all times and the initial density profile is advected by the velocity field. Thus, we are approximating a problem in the linear regime.
This is a modified version of the linear test case from [1]. The original setup can be retrieved by setting $\kappa = 1$. Here, we set $\kappa = 5$ to increase the variation in the density profile, which makes the problem more challenging. Therefore, we can perform computations with higher resolutions before running into problems with machine precision.
For all computations in this chapter, we will use the HLL Riemann solver proposed by Harten et al. [32], including the considerations of Davis [33]. More details can be found in [22]. The experimental order of convergence (EOC) is computed using the formula

$$EOC = \frac{\log(\|\rho_{n^2} - \rho_{ex}\|_1 / \|\rho_{(2n)^2} - \rho_{ex}\|_1)}{\log(2)},$$

where ρ_{n^2} is the numerical solution in density, computed on an $n \times n$ coarse grid, while ρ_{ex} is the exact value of the density evaluated on the same grid.

Table 4.4 shows the results of a numerical convergence study. We compute the L1 grid function norm of the error in density by comparing the solution obtained on each grid with the exact solution. In all of these computations, we used RK(10,5) of Section B.2 as the time–stepping scheme. The upper part of the table shows the results for 5^{th}–order WENO-JS reconstruction. As expected in the linear case, the simple DIM–BY–DIM approach converges with 5^{th}–order. The full order of convergence of the WENO-JS reconstruction is also retained by the MULTI–DIM approach. In fact, the results are almost identical. But the MODIFIED–4 approach also converges with 5^{th}–order. This is surprising since the method is formally only 4^{th}–order accurate. Only on the finest grid do we see a drop in the convergence rate.

In the lower part of the table, we show the results for the 5^{th}–order WENO-Z reconstruction. Again, the DIM–BY–DIM approach and the MULTI–DIM approaches converge with 5^{th}–order, but the error is lower than in the WENO-JS computations. The convergence rate of the MODIFIED–4 approach is around five on coarse grids and drops down to four on finer grids.

		DIM–BY–DIM		MODIFIED–4		MULTI–DIM	
	grid	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC
WENO-JS5	128^2	1.044E-03		1.047E-03		1.043E-03	
	256^2	3.269E-05	**5.00**	3.285E-05	**4.99**	3.268E-05	**5.00**
	512^2	1.022E-06	**5.00**	1.035E-06	**4.99**	1.022E-06	**5.00**
	1024^2	3.193E-08	**5.00**	3.345E-08	**4.95**	3.193E-08	**5.00**
	2048^2	9.973E-10	**5.00**	1.179E-09	**4.83**	9.973E-10	**5.00**
WENO-Z5	128^2	1.488E-04		1.563E-04		1.488E-04	
	256^2	4.690E-06	**4.99**	5.398E-06	**4.86**	4.690E-06	**4.99**
	512^2	1.470E-07	**5.00**	2.210E-07	**4.61**	1.470E-07	**5.00**
	1024^2	4.597E-09	**5.00**	1.129E-08	**4.29**	4.597E-09	**5.00**
	2048^2	1.432E-10	**5.00**	6.582E-10	**4.10**	1.432E-10	**5.00**

Table 4.4: Convergence study for Example 4.1 with WENO-JS5/WENO-Z5 reconstruction and RK(10,5). The HLL flux and a time step with $CFL \approx 1.5$ were used for these computations.

		DIM–BY–DIM		MODIFIED–6		MULTI–DIM	
	grid	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC
WENO-JS7	128^2	2.680E-05		2.672E-05		2.662E-05	
	256^2	4.134E-07	**6.02**	4.129E-07	**6.02**	4.133E-07	**6.01**
	512^2	5.761E-09	**6.16**	5.767E-09	**6.16**	5.761E-09	**6.16**
	1024^2	6.282E-11	**6.52**	6.302E-11	**6.52**	6.282E-11	**6.52**
WENO-Z7	128^2	1.905E-06		1.981E-06		1.905E-06	
	256^2	1.512E-08	**6.98**	1.684E-08	**6.88**	1.512E-08	**6.98**
	512^2	1.186E-10	**6.99**	1.642E-10	**6.68**	1.186E-10	**6.99**
	1024^2	9.725E-13	**6.93**	2.083E-12	**6.30**	1.029E-12	**6.85**

Table 4.5: Convergence study for Example 4.1 with WENO-JS7/WENO-Z7 reconstruction and RK(11,7). The HLL flux and a time step with $CFL \approx 0.9$ were used for these computations.

In Table 4.5 (top), we show the results for the 7^{th}–order WENO-JS reconstruction in combination with RK(11,7) of Section B.3 as the time–stepping scheme. As for WENO-JS5, any combination leads to almost identical results. But we never see the theoretical order of convergence. The accuracy loss is related to a well–known issue of the classical WENO-JS scheme. As already mentioned in Section 3.1, the choice of ϵ (see (3.17) and (3.22)) might affect the order of convergence. This was discussed by many authors, such as Henrick et al. [17]. This loss of accuracy motivated the development of

the WENO-Z method.

In Table 4.5 (bottom), we show results for the 7^{th}–order WENO-Z reconstruction. As for WENO-Z5, any combination leads to almost identical results. We also see the full theoretical order of convergence for both the DIM–BY–DIM approach as well as for the MULTI–DIM approach. Similar to the results for WENO-Z5, we see superconvergence for the MODIFIED–6 approach on coarse grids decreasing with refinement.

4.7.2 Nonlinear problem

Example 4.2. *We now consider the two-dimensional vortex evolution problem (see e.g. [34]) on the periodic domain* $[-7,7]\times[-7,7]$. *The initial data consists of a mean flow* $\rho = u = v = p = 1$, *which is perturbed by adding*

$$\begin{pmatrix}\delta\rho\\ \delta u\\ \delta v\\ \delta p\end{pmatrix} = \begin{pmatrix}(1+\delta T)^{1/(\gamma-1)}-1\\ -y\frac{\sigma}{2\pi}e^{0.5(1-r)}\\ x\frac{\sigma}{2\pi}e^{0.5(1-r)}\\ (1+\delta T)^{\gamma/(\gamma-1)}-1\end{pmatrix}. \tag{4.23}$$

Here, δT, *the perturbation in the temperature, is given by*

$$\delta T = -\frac{(\gamma-1)\sigma^2}{8\gamma\pi^2}e^{1-r^2}, \tag{4.24}$$

with $r^2 = x^2+y^2$ *and the vortex strength* $\sigma = 5$.

The initial data is also used as a reference solution at time $t = 14$, where it agrees with the exact solution. As in the previous example, we will combine the 5^{th}–order WENO methods with RK(10,5) and the 7^{th}–order WENO reconstruction with RK(11,7).

In Table 4.6 (top), we show the results for the 5^{th}–order WENO-JS reconstruction. As expected in the nonlinear case, the order of convergence of the DIM–BY–DIM approach drops down to two. The MODIFIED–4 approach and the MULTI–DIM approach provide very similar results with a convergence order of around five.

In Table 4.6 (bottom), we show the results for the 5^{th}–order WENO-Z reconstruction. Again, the convergence order of the DIM–BY–DIM approach drops, while the MODIFIED–4 and MULTI–DIM approaches provide similar results. Compared to the WENO-JS5 reconstruction, the results are always better for the MODIFIED–4 and MULTI–DIM approaches. The DIM–BY–DIM approach provides similar results with both reconstructions, since the 2^{nd}–order error term dominates.

		DIM–BY–DIM		MODIFIED–4		MULTI–DIM	
	grid	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC
WENO-JS5	128^2	3.730E-05		1.639E-05		1.632E-05	
	256^2	7.095E-06	**2.39**	5.753E-07	**4.83**	5.707E-07	**4.84**
	512^2	1.758E-06	**2.01**	1.868E-08	**4.95**	1.838E-08	**4.96**
	1024^2	4.396E-07	**2.00**	5.988E-10	**4.96**	5.758E-10	**5.00**
	2048^2	1.099E-07	**2.00**	1.973E-11	**4.92**	1.802E-11	**5.00**
WENO-Z5	128^2	2.953E-05		8.172E-06		8.158E-06	
	256^2	7.038E-06	**2.07**	2.403E-07	**5.09**	2.371E-07	**5.10**
	512^2	1.756E-06	**2.00**	7.716E-09	**4.96**	7.408E-09	**5.00**
	1024^2	4.396E-07	**2.00**	2.569E-10	**4.91**	2.302E-10	**5.01**
	2048^2	1.099E-07	**2.00**	9.958E-12	**4.69**	8.736E-12	**4.72**

Table 4.6: Convergence study for Example 4.2 with WENO-JS5/WENO-Z5 reconstruction and RK(10,5). The HLL flux and a time step with $CFL \approx 1.5$ were used for these computations.

		DIM–BY–DIM		MODIFIED–6		MULTI–DIM	
	grid	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC
WENO-JS7	64^2	1.336E-04		5.728E-05		5.722E-05	
	128^2	2.808E-05	**2.25**	5.232E-07	**6.77**	5.231E-07	**6.77**
	256^2	7.014E-06	**2.00**	3.655E-09	**7.16**	3.652E-09	**7.16**
	512^2	1.757E-06	**2.00**	2.783E-11	**7.04**	2.772E-11	**7.04**
	1024^2	4.396E-07	**2.00**	2.181E-13	**7.00**	2.158E-13	**7.01**
WENO-Z7	64^2	1.182E-04		3.084E-05		3.079E-05	
	128^2	2.805E-05	**2.07**	4.853E-07	**5.99**	4.850E-07	**5.99**
	256^2	7.014E-06	**2.00**	4.286E-09	**6.82**	4.278E-09	**6.82**
	512^2	1.757E-06	**2.00**	3.459E-11	**6.95**	3.439E-11	**6.96**
	1024^2	4.396E-07	**2.00**	2.774E-13	**6.96**	2.712E-13	**6.99**

Table 4.7: Convergence study for Example 4.2 with WENO-JS7/WENO-Z7 reconstruction and RK(11,7). The HLL flux and a time step with $CFL \approx 0.9$ were used for these computations.

In Table 4.7, we show the results obtained for the 7^{th}–order WENO-JS and WENO-Z reconstruction. In both cases, RK(11,7) was used for the time integration. For the DIM–BY–DIM approach, the 2^{nd}–order error term dominates; therefore, the results are similar to those of the 5^{th}–order reconstructions as well. The MODIFIED–6 and the MULTI–DIM approaches provide similar results for each reconstruction. WENO-JS and WENO-Z show similar

results. In all further computations, however, we will only use the WENO-Z reconstructions.

Chapter 5

The computational framework (RACOON)

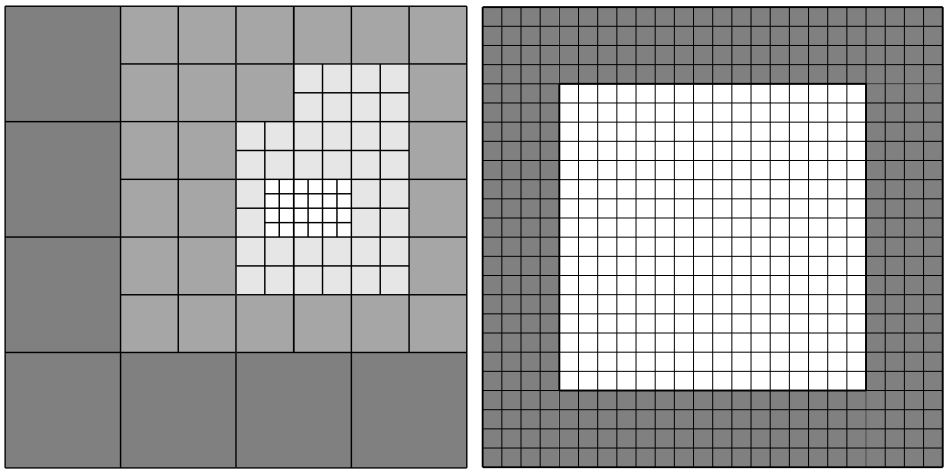
We implemented the methods discussed in Chapters 3 and 4 into the parallel AMR framework *Racoon* developed by Dreher and Grauer [12]. The adaptive mesh refinement and the parallelization are based on blocks, where each block consists of a Cartesian grid. This is a key advantage since all discussed methods from Chapters 3 and 4 are restricted to Cartesian grids. Therefore, using *Racoon*, we can apply any of the methods as it is.

5.1 The general structure of the AMR grid[6]

In *Racoon*, mesh refinement is realized by a regular bisection of grid blocks. By using a refinement ratio of 2, a block of level ℓ is replaced by 2^d blocks of level $\ell + 1$, where d is the spatial dimension. Thus, the grid resolution increases by a factor of 2 with increasing refinement level ℓ. For simplicity, we assume in this chapter that the number of cells in each direction within a block is the same; therefore, a d-dimensional block consists of 8^d or 16^d finite volume mesh cells. But it does not have to be this way. The only restrictions are that the number of cells in each direction has to be even and bigger than the number of ghost cells. In Figure 5.1a, the refinement strategy is illustrated. The coarsest resolution consists of level $\ell = 2$ blocks, i.e. a domain decomposition is used to divide the computational domain into 16 coarse patches. Some of these patches are further refined recursively.

Every few time steps, an error indicator is computed for all grid cells. If a block contains at least one grid cell that is marked for refinement, then the whole block will be refined. Furthermore, neighboring blocks may be

[6] This section was taken from [2] in revised form.



(a) Hierarchy of grid blocks. (b) A single block.

Figure 5.1: (a) Hierarchy of grid blocks in which each grid block consists of a Cartesian mesh with a fixed number of cells. Shown here is an adaptive mesh consisting of blocks of level 2–5. (b) A single block consisting of 16×16 grid cells, and four rows and columns of ghost cells.

refined as well. If a grid cell in the upper right part of a block is marked for refinement, then the block on the right side, the upper block, and the block in the upper right diagonal direction will also be refined, unless these blocks have the same refinement level already. Analogously, additional blocks are refined if a cell in the upper left, lower right, or lower left part of a block is marked for refinement. This is necessary since a usual reason for refinement could be a shock wave. To guarantee that this shock wave does not leave the refined area until the next check, we have to refine these neighboring blocks. Finally, the blocks are required to be *properly nested*, in the sense that neighboring blocks should either both have the same level of refinement or their level of refinement differs by one. This may lead to an additional refinement of some blocks.

5.1.1 Transfer of data between grid patches

Each patch typically consists of 8×8 or 16×16 grid cells, which belong to the physical domain. In addition, each patch is equipped with four rows and columns of ghost cells, which are used to transfer data between patches (see Figure 5.1b). The 5^{th}–order accurate WENO reconstruction can be implemented by using only three ghost cells. However, the implementation of the data transfer between patches can be done more efficiently using four

rows and columns of ghost cells.

If a patch is located at the physical boundary of the domain, then the ghost cells are used to implement the boundary conditions (in the same way as this is typically done for a single grid). Ghost cells that belong to internal boundaries of patches are set by using the grid cell values of the conserved quantities from neighboring patches.

Most of the required data transfer operators are trivial and consist in copying cell–averaged values of the conserved quantities from neighboring patches to ghost cells. Here, we only describe the procedure used to create fine grid values from coarse grid cell–averaged values.

If a patch is refined, then we need to assign cell–averaged values of the conserved quantities to the newly refined grid cells using cell–averages of the conserved quantities from coarser grid cells. With finite volume methods, it is straightforward to do this in a conservative way, without losing accuracy. On the coarser level ℓ, let Q_{ij}^{ℓ} denote the cell–average of the conserved quantities in grid cell C_{ij}^{ℓ}. Furthermore, the geometrical center of this cell is the point (x_i, y_j) and the mesh width is denoted by Δx^{ℓ}, Δy^{ℓ}. On the refined level, we use the index $\ell + 1$.

In grid cell C_{ij}^{ℓ} on the coarse grid level ℓ, we reconstruct a polynomial of the form

$$p(x, y) = \sum_{m,n=0}^{4} c_{mn}(x - x_i)^m (y - y_j)^n. \tag{5.1}$$

The 25 coefficients $c_{00}, \ldots, c_{44}$ are determined using the condition

$$\iint_{C_{kl}} p(x, y)dxdy = \Delta x^{\ell} \Delta y^{\ell} Q_{kl}, \quad k = i-2, \ldots, i+2,\ l = j-2, j+2. \tag{5.2}$$

Now, 5^{th}–order accurate cell–averaged values on the refined grids can be computed by integrating $p(x, y)$ over the four grid cells on the refined level that cover C_{ij}^{ℓ}. For the refined grid cell on the upper right (denoted by ur), we compute

$$Q_{ur}^{\ell+1} = \frac{1}{\Delta x^{\ell+1} \Delta y^{\ell+1}} \int_{y_j}^{y_{j+\frac{1}{2}}} \int_{x_i}^{x_{i+\frac{1}{2}}} p(x, y)\, dxdy, \tag{5.3}$$

and analogously for the other three refined cells. The same approach can be used to fill ghost cells of a fine mesh from coarser grid cells of a neighboring patch.

In order to obtain a polynomial of the desired accuracy, we only need the coefficients c_{mn} with $m + n \leq 4$, i.e. only 15 coefficients. Thus, as an alternative to the approach used here, a conservative reconstruction of

the form presented in [35] can be used, together with a least square fit to determine the coefficients of the polynomial. Analogously, we can reconstruct polynomials of higher degree to obtain higher–order accurate cell–averaged values on the refined grids.

The conservative computation of coarse grid cell–averages from fine grid cells can be performed by simple averaging.

5.1.2 A conservative flux evaluation at grid patch interfaces

For the adaptive method to be conservative, we need to ensure that the same flux is used at interfaces between different grid patches. This requires some exchange of fluxes between neighboring patches and is a well–understood component of adaptive finite volume methods for hyperbolic problems (see, e.g. [36, 11]). Along the boundary of patches with different resolutions, we use the fluxes computed from the fine mesh to update the conserved quantities on both sides of the interface. In a parallel framework, this might add some additional communication across CPUs. However, it is much easier to realize such an exchange efficiently when a low–storage Runge–Kutta method is used for the time integration.

5.1.3 Refinement criteria

To decide whether a block should be marked for refinement, we compute in 2D the quantity

$$\delta = \frac{|q_{i-1,j} - 2q_{i,j} + q_{i+1,j}| + |q_{i,j-1} - 2q_{i,j} + q_{i,j+1}|}{|q_{i,j}|\Delta x \Delta y} \tag{5.4}$$

for each grid cell, where q is the conserved quantity and $\Delta x \Delta y$ is the size of the cell. If δ is larger than a predefined threshold δ_0, then the grid cell (i, j) is marked for refinement. A block will be refined as soon as it contains a single mesh cell that is marked for refinement. The value δ_0 is problem–dependent and will be provided for each test case. Similar refinement criteria were used in [37]. For the Euler equations, it is usually sufficient to check only the mass density ρ. But if there is a discontinuity in one of the other components and not in the density, such as in Example 3.3, we have to check each component.

A more sophisticated refinement criteria, as discussed by Li [38] or Puppo and Semplice [39], could provide better results. But this will not be covered in the present thesis.

5.2 Comparing the methods in 2D with AMR

As in Section 4.7, we test the methods by applying them to the two–dimensional Euler equations of gas dynamics.

5.2.1 Nonlinear problem

We first consider Example 4.2 of Section 4.7.2.

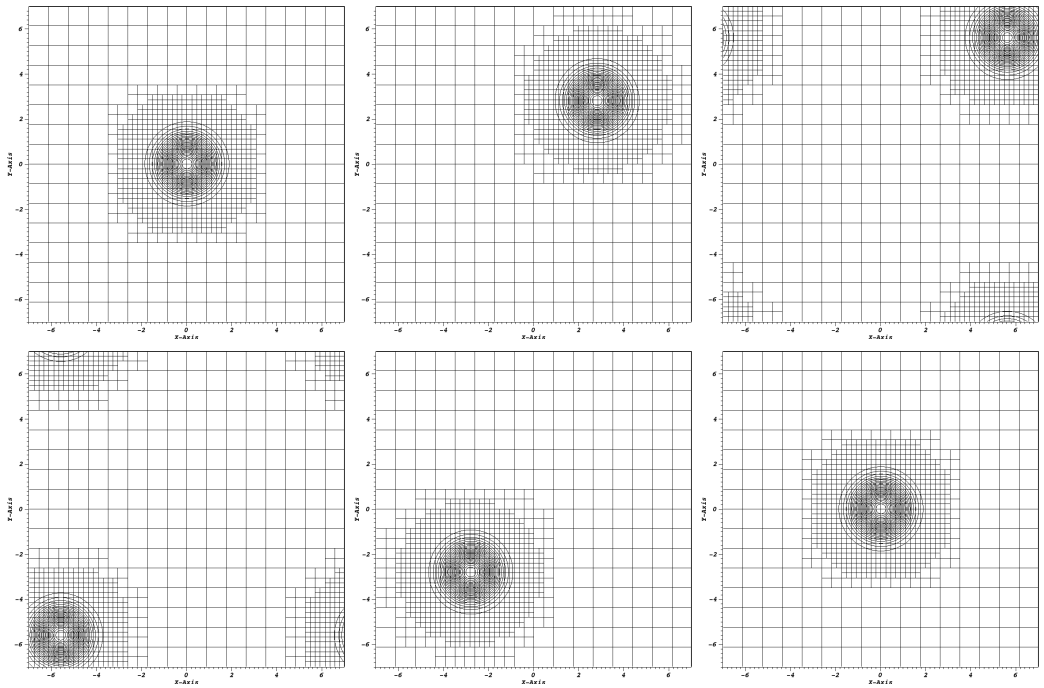


Figure 5.2: Numerical results for the vortex evolution problem using the MODIFIED–4 approach on an adaptively refined grid. The coarse grid corresponds to a resolution with a mesh width of 14/256. The numerical solution is shown at times 0, 2.8, 5.6, 8.4, 11.2, 14.

Figure 5.2 shows the results of a simulation with two levels of refinement starting on a grid with 256×256 mesh cells. In this situation, the coarse grid corresponds to a level 4 grid with 16×16 blocks. Some of these blocks are further refined and contain blocks of refinement level 5 or 6. Each block is discretized with a Cartesian mesh with 16×16 mesh cells plus ghost cells. The 64^2 grid computation starts with a level 2 grid and the 128^2 grid computation starts with a level 3 grid. Thus, in the convergence study of Tables 5.1–5.2, each block is always discretized using a Cartesian mesh with 16×16 mesh cells.

We now present some computations with one or two levels of adaptive mesh refinement (AMR). The threshold for refinements is set to $\delta_0 = 0.01$ (see Section 5.1.3). As mentioned above, we will consider only the WENO-Z reconstruction. All computations will be combined with the low–storage Runge–Kutta method, RK(10,5).

Tables 5.1–5.2 show the results of the convergence studies. The number of grid cells indicates the number of cells used to discretize the domain on the coarsest mesh. Thus, the AMR computation with two levels of refinement, starting with 128^2 grid cells, should be compared with the accuracy obtained by a uniform mesh with 512^2 grid cells.

The error $\|\rho_{n^2} - \rho_{ex}\|$ is computed using the L_1 grid function norm of the difference between the numerical and exact solution on the AMR grid, although n^2 is the number of grid cells that would be used to discretize the problem on the coarsest grid.

		DIM–BY–DIM		MODIFIED–4		MULTI–DIM	
AMR	grid	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC
1 level	64^2	2.95E-005		8.15E-006		8.14E-006	
	128^2	7.03E-006	**2.07**	2.41E-007	**5.08**	2.38E-007	**5.09**
	256^2	1.76E-006	**2.00**	8.17E-009	**4.89**	7.81E-009	**4.93**
	512^2	4.41E-007	**1.99**	3.34E-010	**4.61**	2.96E-010	**4.72**
	1024^2	1.13E-007	**1.97**	1.73E-011	**4.27**	1.28E-011	**4.54**
2 level	64^2	7.03E-006		2.41E-007		2.38E-007	
	128^2	1.76E-006	**2.00**	1.01E-008	**4.57**	9.88E-009	**4.59**
	256^2	4.41E-007	**1.99**	6.93E-010	**3.87**	5.84E-010	**4.08**
	512^2	1.13E-007	**1.96**	9.62E-011	**2.85**	5.96E-011	**3.29**

Table 5.1: Convergence study for Example 4.2 with WENO-Z5 reconstruction and RK(10,5). The HLL flux and a time step with $CFL \approx 1.5$ were used for these computations. We used an AMR grid with one and two levels of refinement. Compare with Table 4.6 (bottom) for the computation on a uniform grid.

In Table 5.1, we show the results for the WENO-Z5 reconstruction with one and two levels of refinement. The DIM–BY–DIM method converges as expected with 2^{nd}–order. The MODIFIED–4 and MULTI–DIM approaches perform very well with one level of refinement. The error on each grid is of the same magnitude as in a corresponding computation on a uniformly refined grid (see Table 4.6 (bottom)). The EOC is lower than on the uniform grid, especially with two levels of refinement, where we see a drop in the order of convergence.

One reason for this loss is the use of different grid structures for different resolutions. It is also possible to enforce the same grid structure for every resolution. Instead of increasing the number of blocks, we could increase the number of cells per block. With this strategy, we are able to retrieve about the same EOC as on uniform grids (see [2]). But this strategy is less efficient in a practical computation and will not be presented here.

		DIM–BY–DIM		MODIFIED–4		MULTI–DIM	
AMR	grid	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC
uniform	64^2	1.182E-04		3.084E-05		3.079E-05	
	128^2	2.805E-05	**2.07**	4.853E-07	**5.99**	4.850E-07	**5.99**
	256^2	7.014E-06	**2.00**	4.285E-09	**6.82**	4.277E-09	**6.83**
	512^2	1.757E-06	**2.00**	3.474E-11	**6.95**	3.454E-11	**6.95**
	1024^2	4.396E-07	**2.00**	2.432E-12	**3.84**	2.432E-12	**3.83**
1 level	64^2	2.805E-05		4.835E-07		4.833E-07	
	128^2	7.011E-06	**2.00**	4.311E-09	**6.81**	4.306E-09	**6.81**
	256^2	1.758E-06	**2.00**	3.897E-11	**6.79**	3.841E-11	**6.81**
	512^2	4.415E-07	**1.99**	2.441E-12	**4.00**	2.440E-12	**3.98**
2 level	64^2	7.011E-06		4.310E-09		4.304E-09	
	128^2	1.757E-06	**2.00**	7.456E-11	**5.85**	7.241E-11	**5.89**
	256^2	4.412E-07	**1.99**	4.115E-12	**4.18**	4.063E-12	**4.16**

Table 5.2: Convergence study for Example 4.2 with WENO-Z7 reconstruction and RK(10,5). The HLL flux and a time step with $CFL \approx 0.9$ were used for these computations.

Table 5.2 presents the results for the WENO-Z7 reconstruction on a uniform grid as well as with one and two levels of refinement. The DIM–BY–DIM method converges as expected with 2^{nd}–order. The MODIFIED–4 and MULTI–DIM approaches perform quite similarly on any grid. Furthermore, the error of computations with one or two levels of refinement is always of the same magnitude as in the corresponding computation on a uniformly refined grid. The drop in the convergence rate on the fines grid is induced by the round–off error of the low–storage Runge–Kutta method. This issue was recently discussed by Ketcheson et al. [40].

Finally, in Table 5.3, we show the results of a performance test for the vortex evolution problem. The CPU time is computed as the average of 10 runs. All computations were performed on equivalent machines with two Intel Xeon E5-2697 CPUs (12 cores each). The setup is the same as before for computations using WENO-Z5 as well as for computations using WENO-Z7. For a better comparison, the CPU time is scaled on each grid with respect

		WENO-Z5		WENO-Z7	
AMR	grid	MODIFIED–4	MULTI–DIM	MODIFIED–6	MULTI–DIM
uniform	128^2	1.12	3.19	1.06	2.44
	256^2	1.09	3.41	1.04	2.76
	512^2	1.10	3.52	1.09	3.11
	1024^2	1.10	3.54	1.09	3.14
1 level	64^2	1.00	2.40	1.02	2.43
	128^2	1.06	2.75	1.08	2.79
	256^2	1.09	3.22	1.08	2.97
	512^2	1.09	3.37	1.08	3.08
2 level	64^2	1.07	2.75	1.07	2.72
	128^2	1.07	3.01	1.08	2.87
	256^2	1.08	3.25	1.08	2.99
	512^2	1.09	3.37	1.08	2.35

Table 5.3: Performance test for the different methods. The test is scaled by the performance of the classical DIM–BY–DIM method, i.e. the classical DIM–BY–DIM method has the value one for each grid and each refinement level.

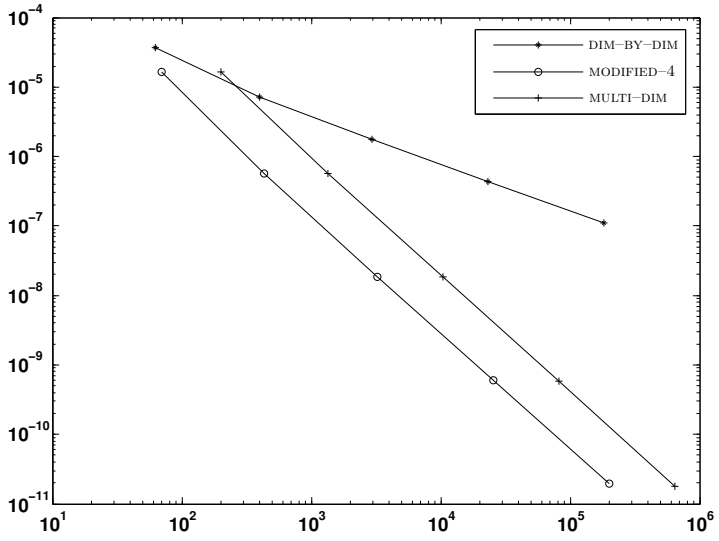


Figure 5.3: Plot of accuracy versus CPU time for the three methods used to approximate the vortex evolution problem.

to the performance of the corresponding DIM–BY–DIM method. In terms of computational time, the DIM–BY–DIM method is most efficient. However,

the MODIFIED–4 approach is only about 1.1 times more expensive, while the accuracy increases significantly. The same holds for the MODIFIED–6 approach. Here, the additional costs are about the same in relation to the DIM–BY–DIM approach using WENO-Z7. In both cases, the MULTI–DIM approach increases the computational costs by a factor of about three. This agrees with the fact that the WENO reconstruction is applied three times more often, which forms the most expensive part of the computation.
Note that these ratios differ from the results in [2]. The reason is that these relations are very sensitive to the implementation of the reconstruction. A more efficient reconstruction may result in a worse ratio for the MODIFIED–4/MODIFIED–6 method. However, the necessary transformations are very simple; therefore, those methods will always be faster than the MULTI–DIM method.

Figure 5.3 shows plots of the accuracy versus the CPU time in s for the three different methods using WENO-Z5 applied to the vortex evolution problem. We use logarithmic scales on both the horizontal and vertical axes. It clearly shows that, for a given accuracy, the two high–order accurate methods require less CPU time than the second–order accurate DIM–BY–DIM approach. The fully multidimensional method is about three times more expensive than the MODIFIED–4 approach, but almost identical results are produced.

5.2.2 Discontinuous problems

We now study the performance of the three different methods for the approximation of two–dimensional Riemann problems proposed by Schulz-Rinne [41, 42]. For all computations of this subsection, we used the 5^{th}–order accurate WENO-Z method.

Example 5.1. *We consider a configuration of four interacting shocks. In this case, the initial values have the form*

$$(\rho, p, u, v)(x, y, 0) = \begin{cases} (1.1, 1.1, 0.0, 0.0) & x > 0.5, y > 0.5 \\ (0.5065, 0.35, 0.8939, 0.0) & x < 0.5, y > 0.5 \\ (1.1, 1.1, 0.8939, 0.8939) & x < 0.5, y < 0.5 \\ (0.5065, 0.35, 0.0, 0.8939) & x > 0.5, y < 0.5 \end{cases} . \quad (5.5)$$

In Figure 5.4 (top), we show the results obtained on adaptively refined grids of level 3–5. In these plots, we also show the boundaries of the different grid patches. In Figure 5.4 (bottom), we show the results for the three different methods on adaptive grids of levels 3–7. The three different methods all produce the same solution structure.

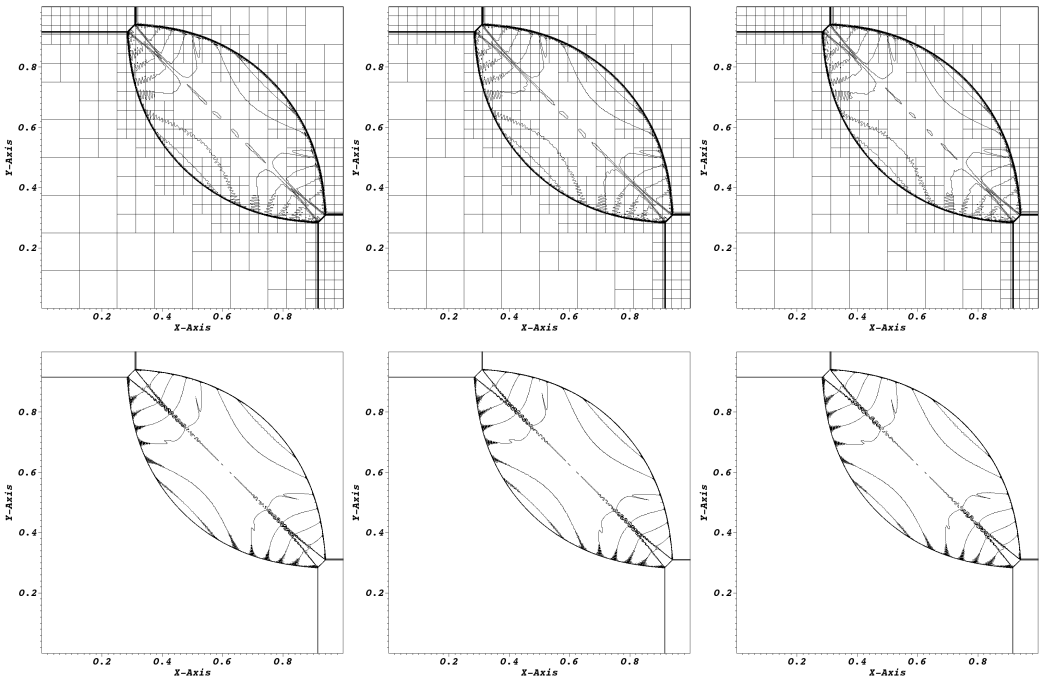


Figure 5.4: Numerical results for the two–dimensional Riemann problem with initial values (5.5) on adaptive grids of levels 3–5 (top) and 3–7 (bottom), where each patch contains 16×16 grid cells. Left: DIM–BY–DIM approach, middle: MODIFIED–4 approach, right: MULTI–DIM method.

Example 5.2. *In the second configuration, the solution structure consists of two contact discontinuities, a rarefaction wave and a shock wave. For this configuration, the initial values have the form*

$$(\rho, p, u, v)(x, y, 0) = \begin{cases} (1.0, 1.0, 0.0, -0.4) & x > 0.5, y > 0.5 \\ (2.0, 1.0, 0.0, -0.3) & x < 0.5, y > 0.5 \\ (1.0625, 0.4, 0.0, 0.2145) & x < 0.5, y < 0.5 \\ (0.5197, 0.4, 0.0, 0.2741) & x > 0.5, y < 0.5 \end{cases} . \quad (5.6)$$

We again consider the results for the three different methods on adaptive grids of levels 3–5 and 3–7. Contour plots of the solution structure are shown in Figure 5.5. For both examples, we set the threshold for refinement to $\delta_0 = 10^4$.

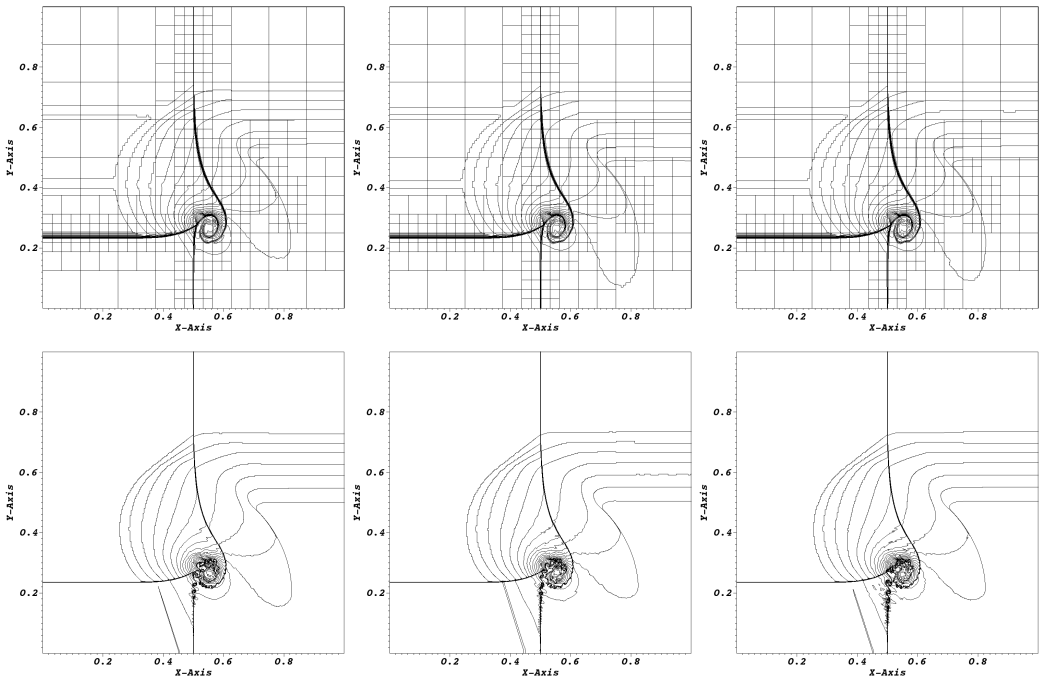


Figure 5.5: Numerical results for the two–dimensional Riemann problem with initial values (5.6) on adaptive grids of levels 3–5 (top) and 3–7 (bottom), where each patch contains 16×16 grid cells. Left: DIM–BY–DIM approach, middle: MODIFIED–4 approach, right: MULTI–DIM method.

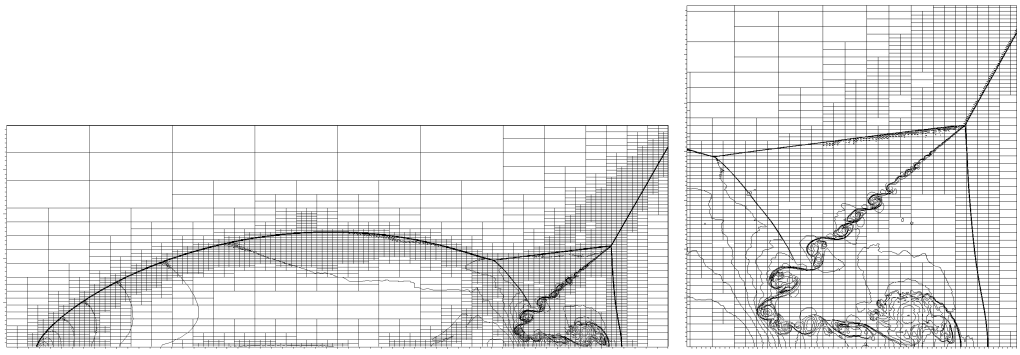
Our numerical results suggest that, for problems with discontinuities in the solution structure, the numerical solutions obtained with the popular DIM–BY–DIM approach compare very well with the results obtained with the two higher–order methods. This was also observed in Zhang et al. [7],

where the DIM–BY–DIM approach was compared with a fully multidimensional method.

We tested all of the two–dimensional Riemann problems proposed by Schulz-Rinne [41, 42] and obtained similar results. Further results have been published in [1] and [2]. For the two examples presented here, it was not necessary to apply the considerations discussed in Section 4.3.

Example 5.3 (Double Mach reflection)**.** *A classical test problem of Woodward and Colella [43] is the reflection of a Mach 10 shock off a 30–degree wedge. Here, the computational domain is the rectangle* $[0, 3] \times [0, 1]$*.*

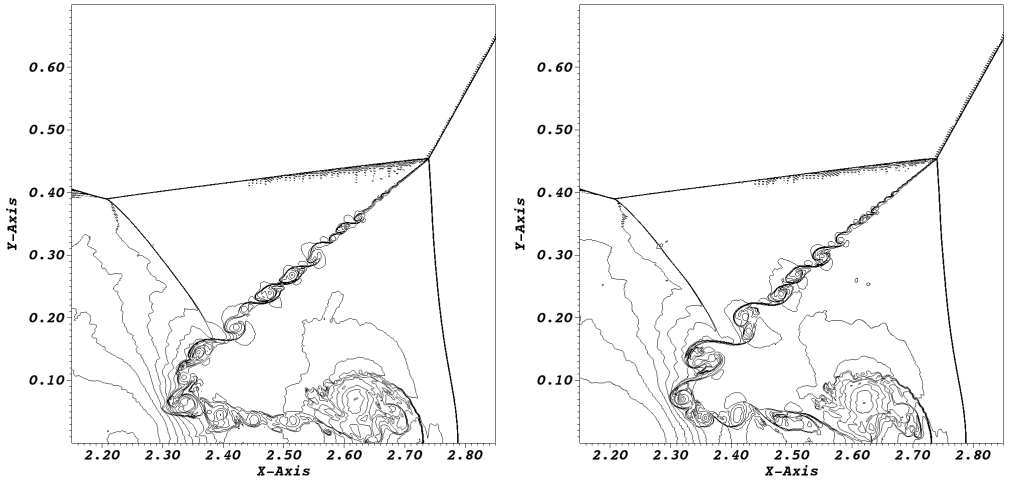
For this computation, each block contains 36×12 mesh cells plus four rows and columns of ghost cells. The grid consists of blocks of levels 3–7. For all computations we set the threshold for refinement to $\delta_0 = 2000$. Level 3 corresponds to a discretization of the whole domain with 288×96 mesh cells, while level 7 corresponds to a discretization with 4608×1536 mesh cells. Figure 5.6 shows the results of the computation, including the block structure using the MODIFIED–4 approach, with a close–up view of the Mach stem region. Figure 5.7 presents a close–up view of the Mach stem



(a) (b)

Figure 5.6: AMR computation of Example 5.3 using the MODIFIED–4 approach.

region using the MODIFIED–4 approach once again. Here, a uniform mesh with 4608×1536 mesh cells was used in (a) and the adaptive mesh was used in (b). The resolution of the finest level for the AMR computation of Figure 5.7 (b) agrees with the resolution used in the whole domain for the computation shown in (a). For this test case, the considerations of Section 4.3 are actually necessary. Otherwise, we would obtain negative pressure across the shock after the transformation to point values. In comparison,



(a) Uniform (b) AMR

Figure 5.7: Close–up view of the double Mach reflection problem computed with the MODIFIED–4 approach. For (a) we used a uniform Cartesian mesh with 4608×1536 mesh cells and in (b) we used the adaptive mesh with blocks at level 3–7.

Figures 5.8 and 5.9 show the same close–up view from computations that employed the MULTI–DIM WENO method of Section 3.4 and the classical DIM–BY–DIM approach of Section 3.3, respectively.

5.3 Comparing the methods in 3D with AMR

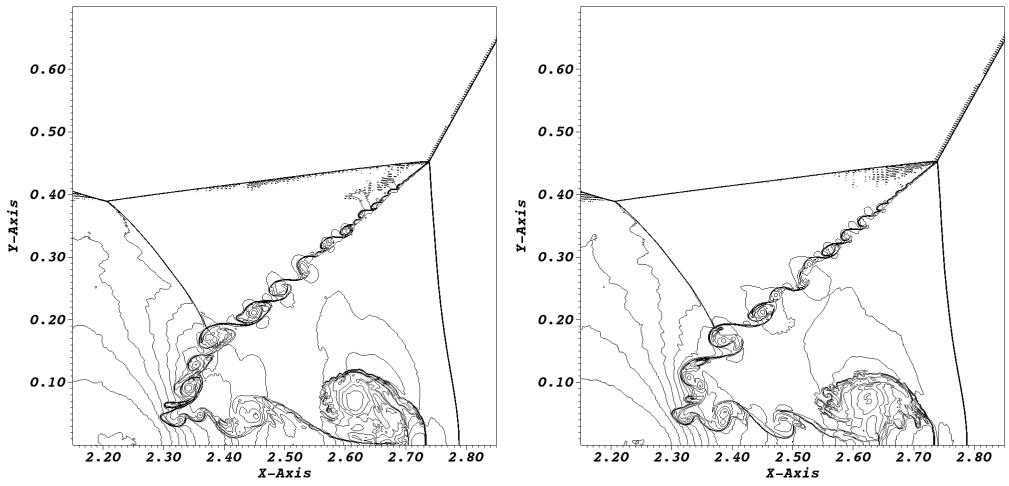
In this section, we apply the methods to the Euler equations of gas dynamics in 3D. The three–dimensional Euler equations are given by

$$\partial_t \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix} + \partial_x \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix} + \partial_y \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix} + \partial_y \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix} = 0, \tag{5.7}$$

with the total energy density

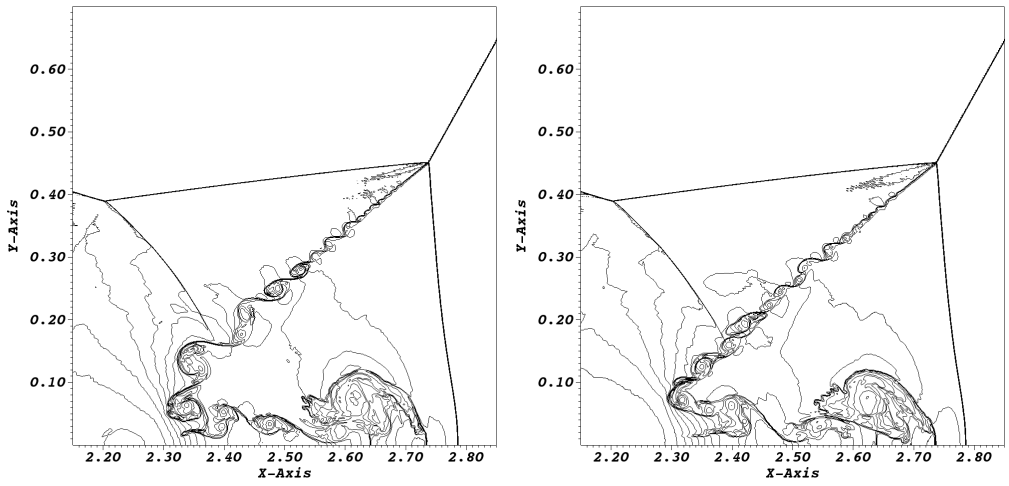
$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho \left(u^2 + v^2 + w^2\right), \tag{5.8}$$

where ρ is the density, u the velocity in the x direction, v the velocity in the y direction, w the velocity in the z direction, and p the pressure.



(a) Uniform (b) AMR

Figure 5.8: Close–up view of the double Mach reflection problem computed with the MULTI–DIM WENO method. For (a) we used a uniform Cartesian mesh with 4608×1536 mesh cells and in (b) we used the adaptive mesh with blocks at level 3–7.



(a) Uniform (b) AMR

Figure 5.9: Close–up view of the double Mach reflection problem computed with the classical DIM–BY–DIM approach. For (a) we used a uniform Cartesian mesh with 4608×1536 mesh cells and in (b) we used the adaptive mesh with blocks at level 3–7.

5.3.1 Rotated vortex

This example is an extension of a 2D vortex introduced by Kadioglu et al. [44].

Example 5.4. *We place the vortex described in [44] at the origin and set all values to be constant in the third dimension. A fully three–dimensional version of the problem is obtained by rotating the structure within the domain by* $45°$ *around the y–axis. To obtain a periodic solution in the domain* $[-1,1]^3$*, the data is assumed to be periodic with period* $\sqrt{2}$ *in the rotated coordinates. This is realized as follows. For* $r = \sqrt{\tilde{x}^2 + \tilde{y}^2}/r_0 < 1$*, we set the tangential velocity to*

$$\vartheta(r) = 1024(1-r)^6 r^6,$$

where

$$\tilde{x} = \left(\left(\cos(\frac{\pi}{4})x + \sin(\frac{\pi}{4})z + 1\right) \mod \sqrt{2}\right) - 1,\ \tilde{y} = y.$$

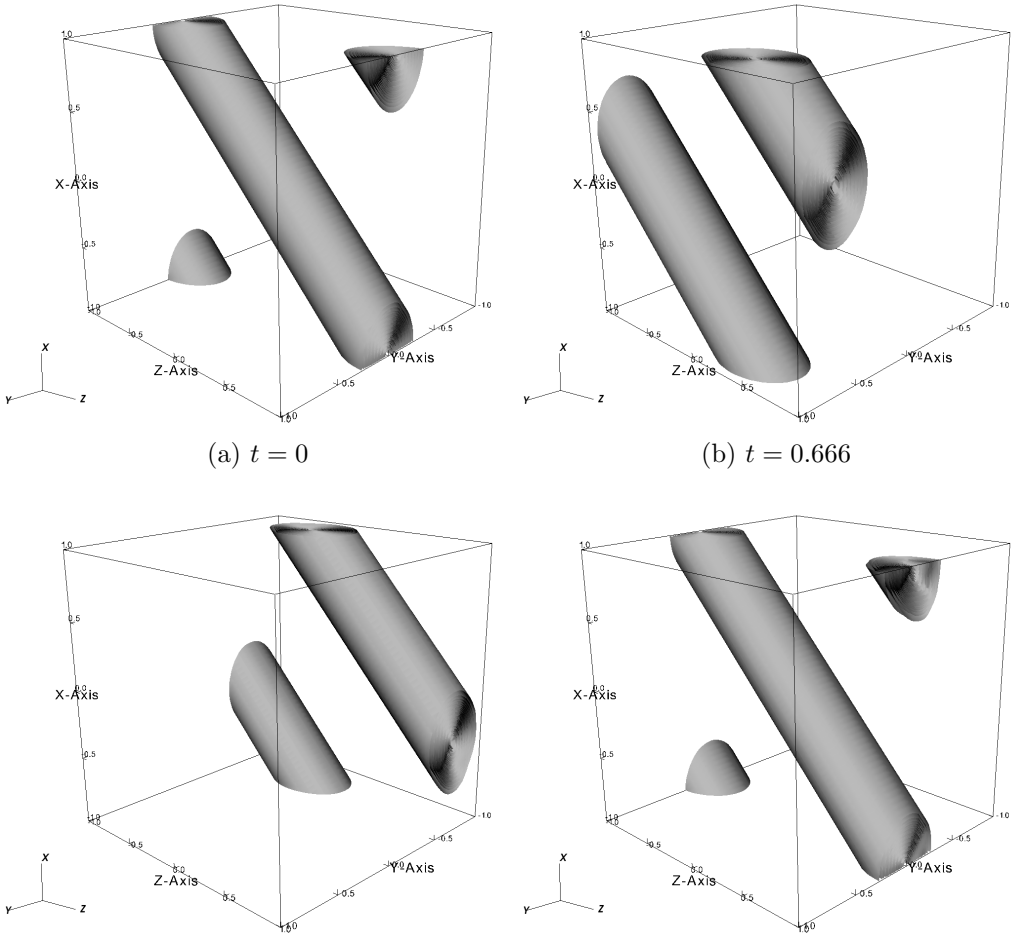
The initial data for the rotated vortex is then given by

$$\begin{pmatrix} \rho \\ u \\ v \\ w \\ p \end{pmatrix}(x,y,z,0) = \begin{pmatrix} \rho_0 + \frac{1}{2}(1-r^2)^6 \\ u_0 - \cos(\frac{\pi}{4})\frac{\tilde{y}}{r}\vartheta(r) \\ v_0 + \frac{\tilde{x}}{r}\vartheta(r) \\ w_0 - \sin(\frac{\pi}{4})\frac{\tilde{y}}{r}\vartheta(r) \\ p_0 + \int_1^r \frac{\rho(s)\vartheta(s)^2}{s} ds \end{pmatrix}. \tag{5.9}$$

Outside of the vortex (i.e. $r \geq 1$*), we set* $(\rho, u, v, w, p) = (\rho_0, u_0, v_0, w_0, p_0)$*.*

For the computation, we set the vortex radius to $r_0 = 0.4$, $\rho_0 = \frac{1}{2}$ and $u_0 = v_0 = w_0 = p_0 = 1$, so that at final time $t = 2.0$, the solution structure is consistent with the initial data. In Figure 5.10, the solution structure is shown at different times. In [44], the pressure p is given explicitly as a polynomial of degree 36. However, according to private correspondence with the authors, one of the coefficients in the original paper was incorrect. It should be $\left(-\frac{269}{15}r^{30}\right)$ instead of $\left(-\frac{259}{15}r^{30}\right)$.

Table 5.4 shows the results of a numerical convergence study. In all of these computations, we used RK(10,5) as time–stepping scheme and the 5^{th}–order WENO-Z reconstruction. In the upper part of the table, we show the results obtained on uniform grids. Similar to the 2D case, the order of convergence of the DIM–BY–DIM approach drops to two, while the MODIFIED–4 and MULTI–DIM approaches provide very similar results with a convergence order of around five. In the lower part of the table, we show the results obtained on an AMR grid with two levels of refinement. The threshold for refinement was set to $\delta_0 = 100$. The number of grid cells indicates the number



(a) $t = 0$ (b) $t = 0.666$

(c) $t = 1.333$ (d) $t = 2$

Figure 5.10: 3D contour plot of the density.

of cells that are used to discretize the domain on the coarsest mesh. Thus, the AMR computation, starting with 64^3 grid cells, should be compared with the accuracy obtained on a uniform mesh with 256^3 grid cells. In fact, each method produces results similar to those of the corresponding uniform grid.

When applying the MULTI–DIM approach in 2D, we first reconstruct interface–averaged values $Q^{\pm}_{i+\frac{1}{2},j}$ in the x direction. In a second step, the reconstruction is applied to $Q^{+}_{i+\frac{1}{2},j}$ and $Q^{-}_{i+\frac{1}{2},j}$ to obtain point values at the quadrature nodes. Analogously, the reconstruction in the y direction is performed. Thus, the reconstruction is applied three times more often than in the DIM–BY–DIM approach. In Section 5.2.1, we showed that the method is also about three times more expensive in terms of CPU time.

		DIM−BY−DIM		MODIFIED−4		MULTI−DIM	
AMR	grid	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC	$\|\rho - \rho_{ex}\|_1$	EOC
uniform	64^3	1.561E-03		1.568E-03		1.575E-03	
	128^3	1.239E-04	**3.65**	1.140E-04	**3.78**	1.113E-04	**3.82**
	256^3	1.197E-05	**3.37**	1.677E-06	**6.09**	1.613E-06	**6.11**
	512^3	2.884E-06	**2.05**	4.516E-08	**5.21**	3.905E-08	**5.37**
2 level	64^3	1.197E-05		1.677E-06		1.612E-06	
	128^3	2.883E-06	**2.05**	4.516E-08	**5.21**	3.906E-08	**5.37**

Table 5.4: Convergence study for Example 5.4 with WENO-Z5 reconstruction and RK(10,5). The HLL flux and a time step with $CFL \approx 1.5$ were used for these computations.

In 3D, the second reconstruction step does not provide point values. Instead, we get the values $Q^{\pm}_{i+\frac{1}{2},j_m,k}$, $1 \leq m \leq n$ at the n quadrature nodes. In a third step, the reconstruction is applied to $Q^{+}_{i+\frac{1}{2},j_m,k}$ and $Q^{-}_{i+\frac{1}{2},j_m,k}$ for $1 \leq m \leq n$ to obtain the point values $q^{\pm}_{i+\frac{1}{2},j_m,k_s}$ for $1 \leq m, s \leq n$. Analogously, the reconstruction in the y and z direction is performed. Thus, the reconstruction is applied $3 + 2n$ times more often than in the DIM−BY−DIM approach. To recover the full order of convergence of the WENO-Z5 method, we use the Gaussian quadrature rule with three points, i.e. the reconstruction is applied nine times more often than by the DIM−BY−DIM approach. In Table 5.5, the results of a performance test confirm the theoretical considerations. The MULTI−DIM method is about nine times more expensive than the DIM−BY−DIM approach. As in 2D, the MODIFIED−4 approach is only slightly more expensive. Thus, in 3D, the advantage of the new method is even more significant.

Figure 5.11 shows plots of the accuracy versus the CPU time in s for the three different methods using WENO-Z5 applied to Example 5.4. We use logarithmic scales on both the horizontal and vertical axes. The MULTI−DIM method is about 10 times more expensive than the MODIFIED−4 approach, but almost identical results are produced. Unlike the 2D case, the DIM−BY−DIM approach seems very efficient compared to the MULTI−DIM approach at least on coarse grids.

AMR	grid	MODIFIED–4	MULTI–DIM
uniform	64^3	0.97	9.54
	128^3	1.14	10.26
	256^3	1.13	10.40
	512^3	1.15	10.58
2 level	64^3	1.12	9.39
	128^3	1.16	9.50

Table 5.5: Performance test of the different methods. The test is scaled by the performance of the classical DIM–BY–DIM method, i.e. the classical DIM–BY–DIM method has the value one for each grid and each refinement level.

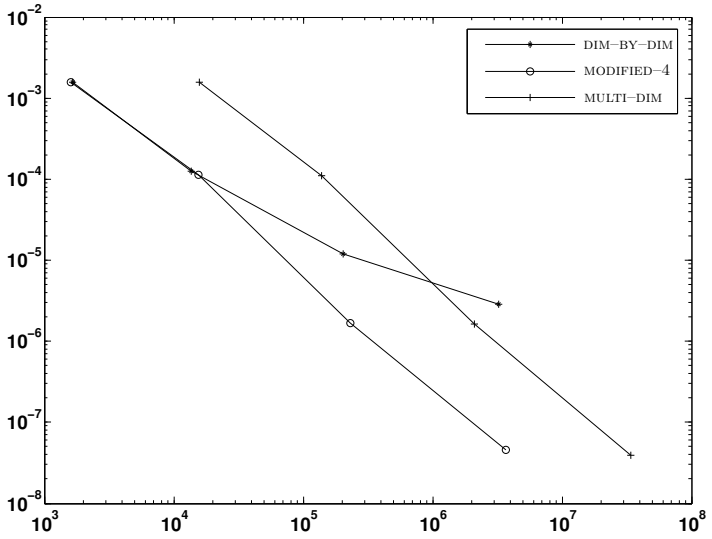


Figure 5.11: Plot of accuracy versus CPU time for the three different methods used to approximate the 3D vortex evolution problem.

Chapter 6

Nonlinear systems in quasilinear form

The WENO reconstruction can also be used to construct high–order accurate methods for hyperbolic equations in the quasilinear form

$$q_t + A(q)q_x + B(q)q_y = 0. \tag{6.1}$$

Recently, Ketcheson et al. [45] developed a high–order wave propagation method based on the WENO reconstruction. Their numerical method can be written in the semi–discrete form

$$\begin{aligned} \frac{d}{dt}Q_{i,j}(t) = &- \frac{1}{\Delta x}\left(\mathcal{A}^-\Delta Q_{i+\frac{1}{2},j} + \mathcal{A}^+\Delta Q_{i-\frac{1}{2},j} + \mathcal{A}\Delta Q_{i,j}\right) \\ &- \frac{1}{\Delta y}\left(\mathcal{B}^-\Delta Q_{i,j+\frac{1}{2}} + \mathcal{B}^+\Delta Q_{i,j-\frac{1}{2}} + \mathcal{B}\Delta Q_{i,j}\right). \end{aligned} \tag{6.2}$$

The fluctuations $\mathcal{A}^{\pm}\Delta Q$ and $\mathcal{B}^{\pm}\Delta Q$ are computed using an eigenvector decomposition of the jump of the reconstructed quantity q at each grid cell interface, as explained in [45, 13]. In particular,

$$\mathcal{A}^{\pm}\Delta Q_{i-\frac{1}{2},j} = \sum_{p=1}^{d}\left(s^p(Q^-_{i-\frac{1}{2},j}, Q^+_{i-\frac{1}{2},j})\right)^{\pm}\mathcal{W}^p(Q^-_{i-\frac{1}{2},j}, Q^+_{i-\frac{1}{2},j}) \tag{6.3}$$

where $\mathcal{W}^p(Q^-_{i-\frac{1}{2},j}, Q^+_{i-\frac{1}{2},j})$ are the waves and $s^p(Q^-_{i-\frac{1}{2},j}, Q^+_{i-\frac{1}{2},j})$ the eigenvalues of $A(q)$ at the interfaces. By $(a)^{\pm}$, we denote the positive $(a)^+ = \max(a, 0)$ or negative $(a)^- = \min(a, 0)$ part of a. The waves can be obtained through decomposition of the jump $Q^+_{i-\frac{1}{2},j} - Q^-_{i-\frac{1}{2},j}$ in terms of eigenvectors $r^p(Q^-_{i-\frac{1}{2},j}, Q^+_{i-\frac{1}{2},j})$

$$Q^+_{i-\frac{1}{2},j} - Q^-_{i-\frac{1}{2},j} = \sum_{p=1}^{d}\alpha^p r^p(Q^-_{i-\frac{1}{2},j}, Q^+_{i-\frac{1}{2},j}) = \sum_{p=1}^{d}\mathcal{W}^p(Q^-_{i-\frac{1}{2},j}, Q^+_{i-\frac{1}{2},j}). \tag{6.4}$$

Analogously to the computation of fluxes for equations in divergence, we need interface–averaged values of the fluctuations. Since both the eigenvalues as well as the waves depend on the reconstructed values, the product will, in general, be only 2^{nd}–order accurate if the averaged interface values $Q^{\pm}_{i-\frac{1}{2},j}$ are used.

The approach to retain the high–order of accuracy can be applied here as well. Analogously to Section 4.2, we modify the computation of the fluctuations $\mathcal{A}^{\pm}\Delta Q$ and $\mathcal{B}^{\pm}\Delta Q$, i.e.

1. Compute averaged values of the conserved quantities at grid cell interfaces using one–dimensional WENO reconstruction, i.e. compute

$$Q^{\pm}_{i+\frac{1}{2},j},\ Q^{\pm}_{i,j+\frac{1}{2}}$$

   at all grid cell interfaces.

2. Compute point values of the conserved quantities at the midpoints of grid cell interfaces, i.e. compute

$$\begin{aligned} q^{\pm}_{i+\frac{1}{2},j} &= Q^{\pm}_{i+\frac{1}{2},j} - \frac{1}{24}\left(Q^{\pm}_{i+\frac{1}{2},j-1} - 2Q^{\pm}_{i+\frac{1}{2},j} + Q^{\pm}_{i+\frac{1}{2},j+1}\right) \\ q^{\pm}_{i,j+\frac{1}{2}} &= Q^{\pm}_{i,j+\frac{1}{2}} - \frac{1}{24}\left(Q^{\pm}_{i-1,j+\frac{1}{2}} - 2Q^{\pm}_{i,j+\frac{1}{2}} + Q^{\pm}_{i+1,j+\frac{1}{2}}\right). \end{aligned} \tag{6.5}$$

3. Compute fluctuations at the midpoints of the grid cell interfaces

$$\begin{aligned} \mathcal{A}^{\pm}\Delta q_{i+\frac{1}{2},j} &= \sum_{p=1}^{d}\left(s^p(q^-_{i+\frac{1}{2},j}, q^+_{i+\frac{1}{2},j})\right)^{\pm}\mathcal{W}^p(q^-_{i+\frac{1}{2},j}, q^+_{i+\frac{1}{2},j}), \\ \mathcal{B}^{\pm}\Delta q_{i,j+\frac{1}{2}} &= \sum_{p=1}^{d}\left(s^p(q^-_{i,j+\frac{1}{2}}, q^+_{i,j+\frac{1}{2}})\right)^{\pm}\mathcal{W}^p(q^-_{i,j+\frac{1}{2}}, q^+_{i,j+\frac{1}{2}}). \end{aligned} \tag{6.6}$$

4. Compute averaged values of the fluctuations, i.e. compute

$$\begin{aligned} \mathcal{A}^{\pm}\Delta Q_{i+\frac{1}{2},j} &= \mathcal{A}^{\pm}\Delta q_{i+\frac{1}{2},j} \\ &\quad + \frac{1}{24}\left(\mathcal{A}^{\pm}\Delta q_{i+\frac{1}{2},j-1} - 2\mathcal{A}^{\pm}\Delta q_{i+\frac{1}{2},j} + \mathcal{A}^{\pm}\Delta q_{i+\frac{1}{2},j+1}\right), \\ \mathcal{B}^{\pm}\Delta Q_{i,j+\frac{1}{2}} &= \mathcal{B}^{\pm}\Delta q_{i,j+\frac{1}{2}} \\ &\quad + \frac{1}{24}\left(\mathcal{B}^{\pm}\Delta q_{i-1,j+\frac{1}{2}} - 2\mathcal{B}^{\pm}\Delta q_{i,j+\frac{1}{2}} + \mathcal{B}^{\pm}\Delta q_{i+1,j+\frac{1}{2}}\right). \end{aligned} \tag{6.7}$$

However, this is not enough. We also have to compute the terms $\mathcal{A}\Delta Q_{i,j}$ and $\mathcal{B}\Delta Q_{i,j}$, which requires some additional transformations. Consider the discretization of

$$\mathcal{A}\Delta Q_{i,j} \approx \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} A(q(x,y,t))q_x(x,y,t)dxdy. \tag{6.8}$$

Using a quadrature formula with nodes $x_{i-\frac{1}{2}} \leq x_{i_1} < \ldots < x_{i_\ell} \leq x_{i+\frac{1}{2}}$ and weights $c_1, \ldots, c_\ell$, one can compute

$$\mathcal{A}\Delta Q_{i,j} \approx \sum_{k=1}^{\ell} c_k \left(\frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} A(q(x_{i_k},y,t))q_x(x_{i_k},y,t)dy \right), \tag{6.9}$$

where we need a high–order approximation to

$$\frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} A(q(x_{i_k},y,t))q_x(x_{i_k},y,t)dy. \tag{6.10}$$

Let $q^1_{i,j}(x)$, $x_{i-\frac{1}{2}} < x < x_{i+\frac{1}{2}}$ denote the p^{th}–order accurate WENO reconstruction in the x direction in cell (i,j). The evaluation of $q^1_{i,j}$ provides us with

$$q^1_{i,j}(x_{i_k}) = \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} q(x_{i_k},y)dy + \mathcal{O}(\Delta x^p + \Delta y^p), \quad k = 1, \ldots, \ell, \tag{6.11}$$

i.e. we obtain the averaged value of q in the y direction at the quadrature nodes. WENO can also be used to approximate the derivative of q. By differentiating the polynomial $q^1_{i,j}$, we obtain an approximation of the averaged derivative, i.e.

$$\left(q^1_{i,j}\right)_x (x_{i_k}) = \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} q_x(x_{i_k},y)dy + \mathcal{O}(\Delta x^{p-1} + \Delta y^p), \quad k = 1, \ldots, \ell. \tag{6.12}$$

Note that a p^{th}–order accurate WENO reconstruction provides only a $(p-1)^{th}$–order accurate approximation to the derivative.

Using these values directly would lead to a 2^{nd}–order accurate method. Instead, we again use the transformation formulas from Section 4.1 to compute approximations of the point values $q(x_{i_k}, y_j)$ and $q_x(x_{i_k}, y_j)$ for $k = 1, \ldots, \ell$ and $y_j = (y_{j-\frac{1}{2}} + y_{j+\frac{1}{2}})/2$. Now, we can evaluate the point values $A(q^1_{i,j}(x_{i_k}, y_j, t))q^1_x(x_{i_k}, y_j, t)$ for $k = 1, \ldots, \ell$. Once again using the transformation of Section 4.1, we compute the averaged values of these quantities in

the y direction and denote them by

$$\overline{A(q^1_{i,j}(x_{i_k},t))q^1_x(x_{i_k})} \approx \frac{1}{\Delta y}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} A(q(x_{i_k},y))q_x(x_{i_k},y)dy \quad k=1,\dots,\ell. \tag{6.13}$$

These values are finally used in the quadrature formula (6.9), i.e.

$$\mathcal{A}\Delta Q_{i,j} = \sum_{k=1}^{\ell} c_k \overline{A(q^1_{i,j}(x_{i_k},t))q^1_x(x_{i_k})}. \tag{6.14}$$

Analogously, we can compute the term

$$\mathcal{B}\Delta Q_{i,j} \approx \frac{1}{\Delta x}\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} B(q(x,y,t))q_y(x,y,t)dxdy. \tag{6.15}$$

The extension to 3D is straightforward. The WENO reconstruction is used in the same way, and the transformation between point values and averaged values is performed as discussed in Section 4.5. The order of accuracy of the resulting method depends on the used WENO reconstruction, the quadrature formula, and the order of the transformation formula. For sufficiently smooth solutions of (6.1) the results of this section are summarized in the following theorem.

Theorem 6.1. *Using a p^{th}–order accurate WENO reconstruction, a quadrature formula of order r, and the* MODIFIED–4 *transformation formula the scheme described above is of order $s = \min(p-1, r, 4)$.*

6.1 Numerical experiments for the equations of ideal magnetohydrodynamics[7]

Finally, we apply the modified WENO method to a more complex application, namely, the approximation of the 3D ideal magnetohydrodynamic (MHD) equations.

The ideal MHD equations can be written in the form

$$\begin{gathered}\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho\mathbf{u}\\ E\\ \mathbf{B}\end{pmatrix} + \nabla\cdot\begin{pmatrix}\rho\mathbf{u}\\ \rho\mathbf{u}\mathbf{u} + \left(p+\frac{1}{2}\|\mathbf{B}\|^2\right)\mathbb{I} - \mathbf{B}\mathbf{B}\\ \mathbf{u}\left(E+p+\frac{1}{2}\|\mathbf{B}\|^2\right) - \mathbf{B}\left(\mathbf{u}\cdot\mathbf{B}\right)\\ \mathbf{u}\mathbf{B}-\mathbf{B}\mathbf{u}\end{pmatrix} = 0,\\ \nabla\cdot\mathbf{B} = 0,\end{gathered} \tag{6.16}$$

[7] This section was adapted from [1] with minor changes.

where ρ, $\rho\mathbf{u} = (\rho u, \rho v, \rho w)^T$ and E are the total mass, momentum, and energy densities, and $\mathbf{B} = (B^1, B^2, B^3)^T$ is the magnetic field. The thermal pressure, p, is related to the conserved quantities through the ideal gas law

$$p = (\gamma - 1)\left(E - \frac{1}{2}\|\mathbf{B}\|^2 - \frac{1}{2}\rho\|\mathbf{u}\|^2\right), \tag{6.17}$$

where $\gamma = 5/3$ is the ideal gas constant.

It is well known that numerical methods for the multidimensional MHD equations must control errors in the discrete divergence of the magnetic field. One way to do this is by using so-called constrained transport (CT) methods. Here, we use an approach that was recently developed by Helzel, Rossmanith, and Taetz [46, 10], and which is based on earlier work by Rossmanith [47]. A FD-WENO method for the ideal MHD equations, using the same kind of constrained transport, was recently proposed by Christlieb, Rossmanith, and Tang [48].

Since $\mathbf{B}$ is divergence–free, we can set $\mathbf{B} = \nabla \times \mathbf{A}$, where $\mathbf{A} \in \mathbb{R}^3$ is the magnetic potential. Inserting this relation in the last line of the MHD equations, we derive an evolution equation for the magnetic potential

$$\partial_t \mathbf{A} + (\nabla \times \mathbf{A}) \times \mathbf{u} = -\nabla\phi. \tag{6.18}$$

Here, ϕ is an arbitrary scalar function. Different choices of ϕ represent different *gauge condition* choices, as explained in [46]. We use the so-called *Weyl gauge*, which means that we set $\nabla\phi = 0$ in (6.18). The resulting evolution equation for the magnetic potential can be written in the form

$$\mathbf{A}_t + N_1(\mathbf{u})\mathbf{A}_x + N_2(\mathbf{u})\mathbf{A}_y + N_3(\mathbf{u})\mathbf{A}_z = 0, \tag{6.19}$$

with

$$N_1 = \begin{pmatrix} 0 & -v & -w \\ 0 & u & 0 \\ 0 & 0 & u \end{pmatrix}, \; N_2 = \begin{pmatrix} v & 0 & 0 \\ -u & 0 & -w \\ 0 & 0 & v \end{pmatrix}, \; N_3 = \begin{pmatrix} w & 0 & 0 \\ 0 & w & 0 \\ -u & -v & 0 \end{pmatrix}. \tag{6.20}$$

The system (6.19) with matrices of the form (6.20) is weakly hyperbolic, i.e. the matrix $N(\mathbf{n}) = n^1 N_1(\mathbf{u}) + n^2 N_2(\mathbf{u}) + n^3 N_3(\mathbf{u})$ has real eigenvalues for all $\mathbf{n} \in S^2$, but there are directions for which $N(\mathbf{n})$ fails to have a complete set of right eigenvectors (see [46]).

To describe the general form of the constrained transport algorithm, we introduce the notation

$$Q'_{MHD}(t) = \mathcal{L}_1(Q_{MHD}(t)), \tag{6.21}$$

for the semi-discrete form of the MHD equations. Here, $Q_{MHD}(t)$ represents the grid function at time t consisting of all cell–averaged values of the conserved quantities from the MHD equation (6.16). Analogously, we introduce

$$Q'_{\mathbf{A}}(t) = \mathcal{L}_2(Q_{\mathbf{A}}(t), Q_{MHD}(t)), \tag{6.22}$$

to describe the semi-discrete form for the evolution equation of the magnetic potential. Note that the evolution of the potential depends on the velocity field, which we take as a given function from the solution step of the MHD equations.

To simplify the notation, we present the numerical method using forward Euler time–stepping.

0. Start with Q^n_{MHD} and $Q^n_{\mathbf{A}}$ (i.e. the solution from the previous time step).

1. Update without regard for the divergence–free condition on the magnetic field, to obtain Q^*_{MHD} and $Q^{n+1}_{\mathbf{A}}$:

$$Q^*_{MHD} = Q^n_{MHD} + \Delta t \mathcal{L}_1(Q^n_{MHD}) \tag{6.23}$$
$$Q^{n+1}_{\mathbf{A}} = Q^n_{\mathbf{A}} + \Delta t \mathcal{L}_2(Q^n_{\mathbf{A}}, Q^n_{MHD}) \tag{6.24}$$

2. Correct the magnetic field components Q^*_{MHD} using the divergence–free values $\mathbf{B}^{n+1} = \nabla \times Q^{n+1}_{\mathbf{A}}$. Set $Q^{n+1}_{MHD} = (\rho^{n+1}, \rho\mathbf{u}^{n+1}, E^{n+1}, \mathbf{B}^{n+1})$.

In Step 1, update (6.23), we use a straightforward extension to the three-dimensional case of our modified WENO method for hyperbolic partial differential equations in divergence form. Here, we used the 5^{th}–order WENO-Z method with a correction that leads to 4^{th}–order accurate flux functions, i.e. MODIFIED–4.

In Step 1, update (6.24), we used a three–dimensional extension of our method from Section 6, to update the evolution equation for the magnetic potential. Due to the weak hyperbolicity of (6.19), the fluctuations $\mathcal{A}^{\pm}\Delta Q_{i+\frac{1}{2},j,k}$, $\mathcal{B}^{\pm}\Delta Q_{i,j+\frac{1}{2},k}$ and $\mathcal{C}^{\pm}\Delta Q_{i,j,k+\frac{1}{2}}$ cannot be computed using an eigenvector decomposition of the jump in $Q_{\mathbf{A}}$ at grid cell interfaces. Instead, we computed the fluctuations using an approach based on the idea of path conservative methods, as explained in [10].

In Step 2, we compute $\mathbf{B}^{n+1} = (B^1, B^2, B^2)$ from the cell–averaged values

of $Q_{\mathbf{A}}^{n+1} = (A^1, A^2, A^3)$, using the formulas

$$\begin{aligned}
B^1_{i,j,k} =& \frac{1}{12\Delta y}\left(A^3_{i,j-2,k} - 8A^3_{i,j-1,k} + 8A^3_{i,j+1,k} - A^3_{i,j+2,k}\right) \\
&- \frac{1}{12\Delta z}\left(A^2_{i,j,k-2} - 8A^2_{i,j,k-1} + 8A^2_{i,j,k+1} - A^2_{i,j,k+2}\right) \\
B^2_{i,j,k} =& \frac{1}{12\Delta z}\left(A^1_{i,j,k-2} - 8A^1_{i,j,k-1} + 8A^1_{i,j,k+1} - A^1_{i,j,k+2}\right) \\
&- \frac{1}{12\Delta x}\left(A^3_{i-2,j,k} - 8A^3_{i-1,j,k} + 8A^3_{i+1,j,k} - A^3_{i+2,j,k}\right) \\
B^3_{i,j,k} =& \frac{1}{12\Delta x}\left(A^2_{i-2,j,k} - 8A^2_{i-1,j,k} + 8A^2_{i+1,j,k} - A^2_{i+2,j,k}\right) \\
&- \frac{1}{12\Delta y}\left(A^1_{i,j-2,k} - 8A^1_{i,j-1,k} + 8A^1_{i,j+1,k} - A^1_{i,j+2,k}\right).
\end{aligned} \tag{6.25}$$

This is a 4^{th}–order accurate approximation of cell–averaged values of $\nabla \times \mathbf{A}$ using cell–averaged values of the magnetic potential.

6.2 Comparing the methods in 3D

Example 6.1. *We consider the three–dimensional smooth Alfvén wave problem (see e.g. [46]). The initial data is given by*

$$\begin{aligned}
\rho(x,y,z,0) &= 1, \\
p(x,y,z,0) &= 0.1, \\
\mathbf{u}(x,y,z,0) &= u^n\mathbf{n} + u^t\mathbf{t} + u^r\mathbf{r}, \\
\mathbf{B}(x,y,z,0) &= B^n\mathbf{n} + B^t\mathbf{t} + B^r\mathbf{r}
\end{aligned}$$

where

$$\begin{aligned}
u^n &= 0, & B^n &= 1, \\
u^t &= 0.1\sin(2\pi\mathbf{n}\cdot\mathbf{x}), & B^t &= 0.1\sin(2\pi\mathbf{n}\cdot\mathbf{x}), \\
u^r &= 0.1\cos(2\pi\mathbf{n}\cdot\mathbf{x}), & B^r &= 0.1\cos(2\pi\mathbf{n}\cdot\mathbf{x})
\end{aligned}$$

and

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \mathbf{n} = \begin{pmatrix} \cos\phi\cos\theta \\ \sin\phi\cos\theta \\ \sin\theta \end{pmatrix}, \ \mathbf{t} = \begin{pmatrix} -\sin\phi \\ \cos\phi \\ 0 \end{pmatrix}, \ \mathbf{r} = \begin{pmatrix} -\cos\phi\sin\theta \\ -\sin\phi\sin\theta \\ \cos\theta \end{pmatrix}.$$

In our computations, we set $\phi = \theta = \tan^{-1}(0.5)$ *and the computational domain is*

$$\Omega = \left[0, \frac{1}{\cos\phi\cos\theta}\right] \times \left[0, \frac{1}{\sin\phi\cos\theta}\right] \times \left[0, \frac{1}{\sin\theta}\right]$$

with periodic boundary conditions. To use the CT method, we also have to specify initial conditions for the magnetic potential $\mathbf{A}$*,*

$$A^1(x,y,z,0) = zn^2 - \frac{1}{2\pi}\sin\phi\sin(2\pi\mathbf{n}\cdot\mathbf{x}),$$
$$A^2(x,y,z,0) = xn^3 + \frac{1}{2\pi}\cos\phi\sin(2\pi\mathbf{n}\cdot\mathbf{x}),$$
$$A^3(x,y,z,0) = yn^1 + \frac{1}{2\pi\cos\phi}\cos(2\pi\mathbf{n}\cdot\mathbf{x}).$$

The boundary conditions are not simple periodic for the magnetic potential; instead, $\mathbf{A}$ *consists of a linear but time–independent part and a periodic time–dependent part. The linear part can be handled by linear extrapolation, which is exact in this case, and the periodic part is handled by applying periodic boundary conditions.*

In all of these computations, we used RK(10,5) as time–stepping scheme and the 5^{th}–order WENO-Z reconstruction. In Table 6.1, we show the results of a numerical convergence study. As expected, the DIM–BY–DIM approach converges with 2^{nd}–order and the DIM–BY–DIM approach with 4^{th}–order. Using the constraint transport method discussed in Section 6.1, the error is bigger. But we still see the theoretical order of convergence. Even so, for

		DIM–BY–DIM		MODIFIED–4	
CT	grid	$\|\rho-\rho_{ex}\|_1$	EOC	$\|\rho-\rho_{ex}\|_1$	EOC
	64×128^2	3.258E-07		3.820E-08	
off	128×256^2	7.602E-08	**2.10**	1.679E-09	**4.51**
	256×512^2	1.884E-08	**2.01**	9.128E-11	**4.20**
	64×128^2	3.528E-06		4.197E-07	
on	128×256^2	8.770E-07	**2.01**	2.661E-08	**3.98**
	256×512^2	2.190E-07	**2.00**	1.675E-09	**3.99**

Table 6.1: Convergence study for Example 6.1 with WENO-Z5 reconstruction and RK(10,5). The HLL flux and a time step with CFL ≈ 1.8 were used for these computations.

this particular example, we do not need to control $\nabla\cdot\mathbf{B}$. This shows that we can easily combine the constraint transport method with the MODIFIED–4 approach to obtain a higher–order method.

Example 6.2. *We consider the three–dimensional cloud–shock interaction problem (see e.g. [49]). On the computational domain* $[0,1]^3$*, the initial data*

consists of a shock located at $x_0 = 0.09375$ *and a density cloud of radius* $r_0 = 0.15$ *centered at* $\mathbf{c} = (0.3, 0.5, 0.5)$

$$\rho(x,y,z,0) = \begin{cases} 3.86859 & x < x_0 \\ 10 & \|(x,y,z) - \mathbf{c}\|_2 < r_0 \\ 1 & else \end{cases},$$

$$p(x,y,z,0) = \begin{cases} 167.345 & x < x_0 \\ 1 & else \end{cases},$$

$$(u,v,w)(x,y,z,0) = \begin{cases} (11.2536, 0, 0) & x < x_0 \\ (0,0,0) & else \end{cases},$$

$$(B^1, B^2, B^3)(x,y,z,0) = \begin{cases} (0, 2.182682, -2.182682) & x < x_0 \\ (0, 0.56418958, 0.56418958) & else \end{cases}.$$

As before, to use the CT method, we also have to specify initial conditions for the magnetic potential $\mathbf{A}$*,*

$$(A^1, A^2, A^3)(x,y,z,0) = \begin{cases} (2.1826182y, 0, -2.1826182(x - x_0)) & x < x_0 \\ (-0.56418958y, 0, -0.56418958(x - x_0)) & else. \end{cases}$$

For the conserved quantities, we impose inflow boundary conditions at $x = 0$ *and outflow boundary conditions on all other sides. The magnetic potential, however, is handled by linear extrapolation.*

For all computations in this section, we use the HLL flux and RK(10,4) with $CFL \approx 1.4$. In Figure 6.1, the solution structure is shown at final time $t = 0.055$. For this computation, we used the WENO-Z5 reconstruction with the constrained transport method and the MODIFIED–4 approach, as discussed in Section 6. The initial data satisfies the constraint $\nabla \cdot \mathbf{B} = 0$. Figure 6.2 shows the evolution of $\|\nabla \cdot \mathbf{B}\|_\infty$ in time for different methods. Without the constraint transport method, $\|\nabla \cdot \mathbf{B}\|_\infty$ grows in time for both the DIM–BY–DIM and the MODIFIED–4 approaches. But also, if we increase the spatial resolution, compare Figures 6.2a and 6.2b. On the other hand, using the constraint transport method, we always get $\|\nabla \cdot \mathbf{B}\|_\infty < 10^{-11}$. If we compare the results of the DIM–BY–DIM method with the results of the MODIFIED–4 method, we see some clear differences. But Figure 6.2 also shows that the general behavior of both methods is very similar.

Note that the considerations of Section 4.3 are crucial for this test case. If the transformation (4.12) is applied across the shock, the resulting point value will be unphysical. Therefore, the MODIFIED–4 method is reduced to the classical DIM–BY–DIM method near the shock, as discussed in Section 4.3.

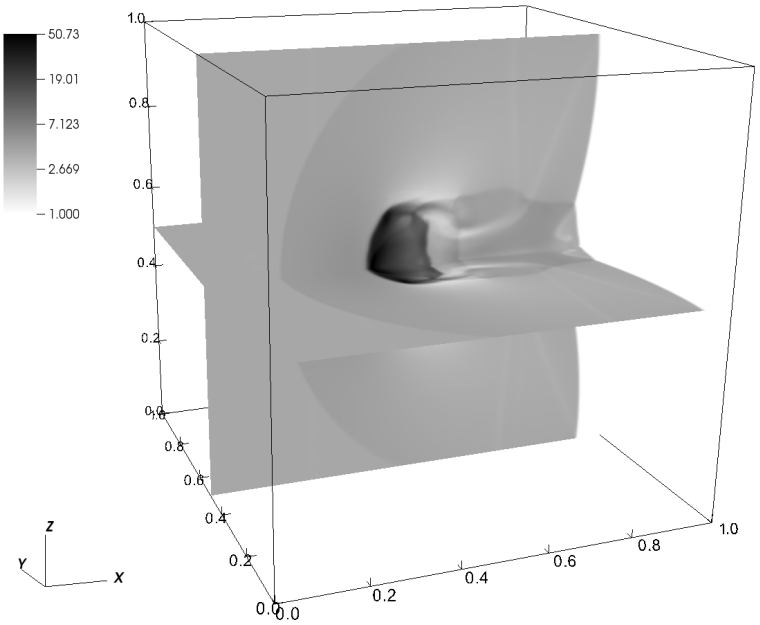
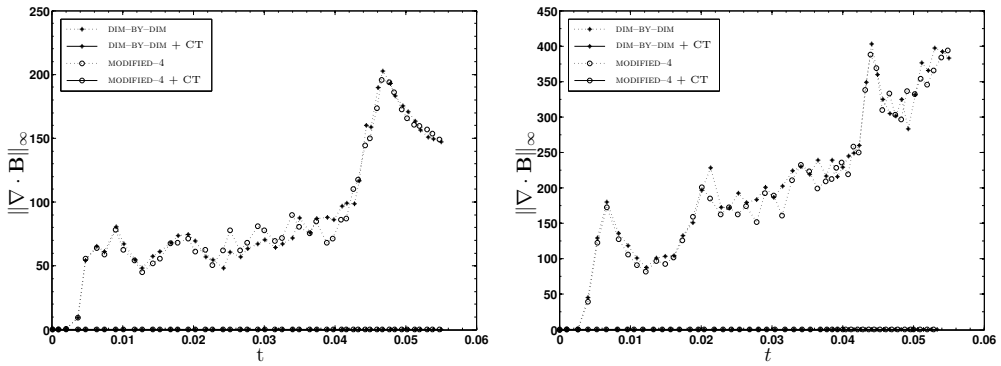


Figure 6.1: Density at final time $t = 0.055$ of Example 6.2.



(a) 128^3 grid cells (b) 256^3 grid cells

Figure 6.2: Evolution of $\|\nabla \cdot \mathbf{B}\|_\infty$ in time, on different grid resolutions for Example 6.2.

Chapter 7

Conclusions and outlook

7.1 New results

In this thesis, a modification of the popular DIM–BY–DIM WENO method for Cartesian grids has been developed. For more flexibility regarding the grid structure, the method has also been extended to adaptively refined grids. The modification improves the order of accuracy, while only slightly increasing the computational costs in two and three spatial dimensions. Several test computations confirmed the gain in accuracy and an increase of less than 20% of the computational costs. The alternative, a MULTI–DIM WENO reconstruction, increases the computational costs by at least a factor of three in 2D. In 3D, it is even a factor of nine. When applied to discontinuous problems, the new method behaves very similarly to the classical DIM–BY–DIM approach.

Furthermore, the method has been extended to nonlinear systems in quasilinear form and applied together with the unstaggered CT method developed by Helzel et al. [10] to the ideal MHD equations in 3D. As for equations in divergence form, the modification improves the order of accuracy, while only slightly increasing the computational costs.

7.2 Future work

One problem with the new method is the limiting at strong discontinuities, as discussed in Section 4.3. Even so, the proposed limiter provided satisfactorily results so far. There might be situations where the limiter fails. However, since only strong discontinuities need to be detected, it should be possible to find an inexpensive and effective limiter for the transformation step.

Also of interest is the extension of the MODIFIED–4/MODIFIED–6 ap-

proach to mapped coordinates. Colella et al. [29] proposed a transformation between average values and point values on mapped grids. Therefore, the question would be whether all results in this thesis can be extended to mapped coordinates.

Appendix A

Initial Example

The initial example of Section 2 was discussed while neglecting the details of the used methods. For completeness, we now specify what methods were used.

A.1 The setup

We are interested in the solution on the domain $[-1,1]^2$. But to avoid interactions with the boundary, the actual computational domain is $[-1.125, 1.125]^2$. For our computation, this surrounding area was sufficient to avoid any interactions with the boundary. But thinking of a long–term computation, this surrounding area can also be used as an absorbing layer of a PML method. Following the lead of Hesthaven [50], we implemented and tested the absorbing layer with similar results. But as mentioned above, it is not necessary for this particular computation.

The error was computed only on the domain $[-1,1]^2$ and the grid resolution always refers to the resolution of this domain. The surrounding area is discretized with additional cells of the same size. The computational domain and the grid resolution are chosen so that the boundary of the domain of interest is always an interface of the grid. For example, if 256^2 grid cells discretize the domain $[-1,1]^2$, then 288^2 grid cells of the same size discretize the computational domain $[-1.125, 1.125]^2$.

For the time integration, we used RK(11,7) of Section B.3 for every method. The HLL flux and a time step with $CFL \approx 0.9$ were also used in all computations. The 3^{rd}–, 5^{th}–, and 7^{th}–order methods refer to a WENO-Z3, WENO-Z5, and WENO-Z7 reconstruction, respectively. While WENO-Z5 and WENO-Z7 are discussed in Section 3.1, WENO-Z3 will be discussed below. To extend the methods to two dimensions, the DIM–BY–DIM approach

of Section 3.3 was used.

A.2 The 3^{rd}–order WENO method

In the case $p = 3$, the interfaces values of the conserved quantities are computed by

$$q^{\mp}_{i\pm\frac{1}{2}} = \omega^{\mp}_0 q^{(0\mp)}_{i\pm\frac{1}{2}} + \omega^{\mp}_1 q^{(1\mp)}_{i\pm\frac{1}{2}}, \tag{A.1}$$

with

$$\begin{aligned} q^{(0-)}_{i+\frac{1}{2}} &= -\frac{1}{2}Q_{i-1} + \frac{3}{2}Q_i, \quad q^{(0+)}_{i-\frac{1}{2}} = \frac{1}{2}Q_{i-1} + \frac{1}{2}Q_i, \\ q^{(1-)}_{i+\frac{1}{2}} &= \frac{1}{2}Q_i + \frac{1}{2}Q_{i+1}, \qquad q^{(1+)}_{i-\frac{1}{2}} = \frac{3}{2}Q_i - \frac{1}{2}Q_{i+1}. \end{aligned} \tag{A.2}$$

The coefficients $\omega^{\pm}_0, \omega^{\pm}_1$ are the weights and they depend basically on the smoothness of the solution in the corresponding stencil. Equation (3.11) leads to

$$\begin{aligned} IS_0 &= (Q_i - Q_{i-1})^2 \\ IS_1 &= (Q_{i+1} - Q_i)^2 . \end{aligned} \tag{A.3}$$

The weights are then defined as suggested in [20] by

$$\omega^{\pm}_j = \frac{\tilde{\omega}^{\pm}_j}{\tilde{\omega}^{\pm}_0 + \tilde{\omega}^{\pm}_1}, \quad \tilde{\omega}^{\pm}_j = \gamma^{\pm}_j \left(1 + \left(\frac{\tau_3}{IS_j + \epsilon}\right)^2\right) \tag{A.4}$$

with $\gamma^{-}_0 = \gamma^{+}_1 = \frac{1}{3}$, $\gamma^{-}_1 = \gamma^{+}_0 = \frac{2}{3}$ and $\tau_3 = |IS_0 - IS_1|$. The ϵ is a small positive number used to avoid division by zero. We set $\epsilon = \Delta x^4$. The description is now complete. We refer to this reconstruction method as WENO-Z3.

Appendix B

Discretization in time

In this chapter, we present the time discretization methods used in this work. As mentioned, we use the method of lines to separate the discretization in space and time. After discretizing the PDE in space, we obtain a system of ODEs of the form

$$\frac{d}{dt}Q(t) = \mathcal{L}(Q(t)), \tag{B.1}$$

where $Q(t)$ is a grid function of cell–averaged values of the conserved quantities at time t. We discretize the resulting ODE system using explicit Runge–Kutta methods of order four, five, or seven.

B.1 The 4^{th}–order Runge–Kutta method

$Q_1^n = Q^n$
$Q_2^n = Q^n$
for $i = 1 : 5$ **do**
    $Q_1^n = Q_1^n + \Delta t \frac{1}{6}\mathcal{L}(Q_1^n)$
end for
$Q_2^n = \frac{1}{25}Q_2^n + \frac{9}{25}Q_1^n$
$Q_1^n = 15Q_2^n - 5Q_1^n$
for $i = 6 : 9$ **do**
    $Q_1^n = Q_1^n + \Delta t \frac{1}{6}\mathcal{L}(Q_1^n)$
end for
$Q^{n+1} = Q_2^n + \frac{3}{5}Q_1^n + \Delta t \frac{1}{10}\mathcal{L}(Q_1^n)$

Algorithm B.1: Low–storage implementation of the strong stability preserving Runge–Kutta scheme of Ketcheson [51]

For computations with shock waves, we use the 10–stage 4^{th}–order strong stability preserving (SSP) Runge–Kutta scheme of Ketcheson [51]. 10 stages may sound computationally expensive and complicated, but the opposite is the case. With this method, it is possible to have a very high CFL number. Furthermore, it is possible to implement the method using only two memory registers, i.e. it is low–storage. Let $Q^n = Q(t_n)$. Then, one time step is described in Algorithm B.1. We will refer to this method as SSP-RK(10,4). For more details on the scheme, we refer to the original paper [51].

B.2 The 5^{th}–order Runge–Kutta method

For convergence tests with a 5^{th}–order WENO reconstruction, we use the 10–stage 5^{th}–order Runge–Kutta scheme of Parsani et al. [52]. With this method, it is also possible to have a very high CFL number. It is also quite efficient in the sense that it can be implemented using only three memory registers, i.e. it is low–storage. Let $Q^n = Q(t_n)$. Then, one time step is described in Algorithm B.2. The coefficients $\delta_i, \gamma_{j,i}, \beta_i$ for $i = 1, ..., 10, j =$

$Q_1^n = Q^n$
$Q_2^n = 0$
$Q_3^n = Q^n$
for $i = 1 : 10$ **do**
    $Q_2^n = Q_2^n + \delta_i Q_1^n$
    $Q_1^n = \gamma_{1,i} Q_1^n + \gamma_{2,i} Q_2^n + \gamma_{3,i} Q_3^n + \beta_i \Delta t \mathcal{L}(Q_1^n)$
end for
$Q^{n+1} = Q_1^n$

Algorithm B.2: Low–storage implementation of the Runge–Kutta scheme of Parsani et al. [52]

$1, ..., 3$ can be found in the Appendix of the original paper [52]. We will refer to this method as RK(10,5).

B.3 The 7^{th}–order Runge–Kutta method

For convergence tests with a 7^{th}–order WENO reconstruction, we use the 11–stage 7^{th}–order Runge–Kutta scheme of Fehlberg [53]. This method is not low–storage. It is also not possible to have a large CFL number, so it might be not very efficient. But in order to avoid any influence from the time–stepping scheme, we will use this method for some convergence tests.

The method is described in the classical way by the Butcher tableau in Tables B.1.

0											
$\frac{2}{27}$	$\frac{2}{27}$										
$\frac{1}{9}$	$\frac{1}{36}$	$\frac{1}{12}$									
$\frac{1}{6}$	$\frac{1}{24}$	0	$\frac{1}{8}$								
$\frac{5}{12}$	$\frac{5}{12}$	0	$-\frac{25}{16}$	$\frac{25}{16}$							
$\frac{1}{2}$	$\frac{1}{20}$	0	0	$\frac{1}{4}$	$\frac{1}{5}$						
$\frac{5}{6}$	$-\frac{25}{108}$	0	0	$\frac{125}{108}$	$-\frac{65}{27}$	$\frac{125}{54}$					
$\frac{1}{6}$	$\frac{31}{300}$	0	0	0	$\frac{61}{225}$	$-\frac{2}{9}$	$\frac{13}{900}$				
$\frac{2}{3}$	2	0	0	$-\frac{53}{6}$	$\frac{704}{45}$	$-\frac{107}{9}$	$\frac{67}{90}$	3			
$\frac{1}{3}$	$-\frac{91}{108}$	0	0	$\frac{23}{108}$	$-\frac{976}{135}$	$\frac{311}{54}$	$-\frac{19}{60}$	$\frac{17}{6}$	$-\frac{1}{12}$		
1	$\frac{2383}{4100}$	0	0	$-\frac{341}{164}$	$\frac{4496}{1025}$	$-\frac{301}{82}$	$\frac{2133}{4100}$	$\frac{45}{82}$	$\frac{45}{164}$	$\frac{18}{41}$	
	$\frac{41}{840}$	0	0	0	0	$\frac{34}{105}$	$\frac{9}{35}$	$\frac{9}{35}$	$\frac{9}{280}$	$\frac{9}{280}$	$\frac{41}{840}$

Table B.1: Butcher tableau of a 7^{th}–order accurate Runge–Kutta method from [53].

Appendix C

Statement about my contribution in previously published work

The theoretical results presented in [1] have been obtained in joint work with my thesis adviser Christiane Helzel. However, all the numerical tests have been perform by myself.

In order to develop our scheme on grids with adaptive mesh refinement as presented in [2] we used the software package *Racoon* developed by Dreher and Grauer. Jürgen Dreher introduced me to this software and provided support for the implementation of the new algorithm. With this help, I carried out all the computations by myself.

Acknowledgements

Computational support and infrastructure was provided by the "Centre for Information and Media Technology" (ZIM) at the University of Düsseldorf (Germany).

Bibliography

[1] Pawel Buchmüller and Christiane Helzel. Improved accuracy of high-order WENO finite volume methods on Cartesian grids. *Journal of Scientific Computing*, 61(2):343–368, 2014.

[2] Pawel Buchmüller, Jürgen Dreher, and Christiane Helzel. Finite volume WENO methods for hyperbolic conservation laws on Cartesian grids with adaptive mesh refinement. *Applied Mathematics and Computation*, 272, Part 2:460 – 478, 2016. Recent Advances in Numerical Methods for Hyperbolic Partial Differential Equations.

[3] Chi-Wang Shu. High order weighted essentially nonoscillatory schemes for convection dominated problems. *SIAM Review*, 51(1):82–126, 2009.

[4] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439 – 471, 1988.

[5] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83(1):32 – 78, 1989.

[6] Barry Merriman. Understanding the Shu–Osher conservative finite difference form. *Journal of Scientific Computing*, 19(1):309–322, 2003.

[7] Rui Zhang, Mengping Zhang, and Chi-Wang Shu. On the order of accuracy and numerical performance of two classes of finite volume WENO schemes. *Communications in Computational Physics*, 9(3):807–827, 2011.

[8] J. Casper and H.L. Atkins. A finite-volume high-order ENO scheme for two-dimensional hyperbolic systems. *Journal of Computational Physics*, 106(1):62 – 76, 1993.

[9] P. Tsoutsanis, V.A. Titarev, and D. Drikakis. WENO schemes on arbitrary mixed-element unstructured meshes in three space dimensions. *Journal of Computational Physics*, 230(4):1585 – 1601, 2011.

[10] Christiane Helzel, James A. Rossmanith, and Bertram Taetz. A high-order unstaggered constrained-transport method for the three-dimensional ideal magnetohydrodynamic equations based on the method of lines. *SIAM Journal on Scientific Computing*, 35(2):A623–A651, 2013.

[11] Peter McCorquodale and Phillip Colella. A high-order finite-volume method for conservation laws on locally refined grids. *Mathematical Sciences Publishers*, 6(1):1–25, 2011.

[12] Jürgen Dreher and Rainer Grauer. Racoon: A parallel mesh-adaptive framework for hyperbolic conservation laws. *Parallel Computing*, 31(8–9):913 – 932, 2005.

[13] Randall J LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge university press, 2002.

[14] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200–212, 1994.

[15] Ami Harten, Bjorn Engquist, Stanley Osher, and Sukumar R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 131(1):3 – 47, 1997.

[16] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202 – 228, 1996.

[17] Andrew K. Henrick, Tariq D. Aslam, and Joseph M. Powers. Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points. *Journal of Computational Physics*, 207(2):542 – 567, 2005.

[18] Marcos Castro, Bruno Costa, and Wai Sun Don. High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 230(5):1766 – 1792, 2011.

[19] F. Aràndiga, A. Baeza, A. M. Belda, and P. Mulet. Analysis of WENO schemes for full and global accuracy. *SIAM Journal on Numerical Analysis*, 49(2):893–915, 2011.

[20] Wai-Sun Don and Rafael Borges. Accuracy of the weighted essentially non-oscillatory conservative finite difference schemes. *Journal of Computational Physics*, 250:347 – 372, 2013.

[21] Rafael Borges, Monique Carmona, Bruno Costa, and Wai Sun Don. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *Journal of Computational Physics*, 227(6):3191 – 3211, 2008.

[22] Eleuterio F Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2009.

[23] Richard Liska and Burton Wendroff. Comparison of several difference schemes on 1D and 2D test problems for the Euler equations. *SIAM Journal on Scientific Computing*, 25(3):995–1017, 2003.

[24] Ami Harten and James M Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics*, 50(2):235 – 269, 1983.

[25] B Einfeldt, C.D Munz, P.L Roe, and B Sjögreen. On Godunov-type methods near low densities. *Journal of Computational Physics*, 92(2):273 – 295, 1991.

[26] Xiangxiong Zhang and Chi-Wang Shu. On positivity-preserving high order discontinuous galerkin schemes for compressible Euler equations on rectangular meshes. *Journal of Computational Physics*, 229(23):8918 – 8934, 2010.

[27] Jing Shi, Changqing Hu, and Chi-Wang Shu. A technique of treating negative weights in WENO schemes. *Journal of Computational Physics*, 175(1):108 – 127, 2002.

[28] P. McCorquodale, M.R. Dorr, J.A.F. Hittinger, and P. Colella. High-order finite-volume methods for hyperbolic conservation laws on mapped multiblock grids. *Journal of Computational Physics*, 288:181 – 195, 2015.

[29] P. Colella, M.R. Dorr, J.A.F. Hittinger, and D.F. Martin. High-order, finite-volume methods in mapped coordinates. *Journal of Computational Physics*, 230(8):2952 – 2976, 2011.

[30] Michael Barad and Phillip Colella. A fourth-order accurate local refinement method for poisson's equation. *J. Comput. Phys.*, 209(1):1–18, 2005.

[31] Lars Hörmander. *The Analysis of Linear Partial Differential Operators I.* Springer Berlin Heidelberg, 1990.

[32] Amiram Harten, Peter D. Lax, and Bram van Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25(1):35–61, 1983.

[33] S. F. Davis. Simplified second-order Godunov-type methods. *SIAM Journal on Scientific and Statistical Computing*, 9(3):445–473, 1988.

[34] Changqing Hu and Chi-Wang Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150(1):97 – 127, 1999.

[35] Carl F. Ollivier-Gooch. Quasi-ENO schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. *Journal of Computational Physics*, 133(1):6 – 17, 1997.

[36] Marsha J. Berger and Randall J. LeVeque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM Journal on Numerical Analysis*, 35(6):2298–2316, 1998.

[37] Michael Dumbser, Olindo Zanotti, Arturo Hidalgo, and Dinshaw S. Balsara. ADER-WENO finite volume schemes with space-time adaptive mesh refinement. *Journal of Computational Physics*, 248:257 – 286, 2013.

[38] Shengtai Li. Comparison of refinement criteria for structured adaptive mesh refinement. *Journal of Computational and Applied Mathematics*, 233(12):3139 – 3147, 2010. Finite Element Methods in Engineering and Science (FEMTEC 2009).

[39] Gabriella Puppo and Matteo Semplice. Numerical entropy and adaptivity for finite volume schemes. *Communications in Computational Physics*, 10(5):1132–1160, 2011.

[40] David I. Ketcheson, Lajos Lóczi, and Matteo Parsani. Internal error propagation in explicit Runge-Kutta methods. *SIAM Journal on Numerical Analysis*, 52(5):2227–2249, 2014.

[41] C. Schulz-Rinne, J. Collins, and H. Glaz. Numerical solution of the Riemann problem for two-dimensional gas dynamics. *SIAM Journal on Scientific Computing*, 14(6):1394–1414, 1993.

[42] C. Schulz-Rinne. Classification of the Riemann problem for two-dimensional gas dynamics. *SIAM Journal on Mathematical Analysis*, 24(1):76–88, 1993.

[43] Paul Woodward and Phillip Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54(1):115 – 173, 1984.

[44] Samet Y. Kadioglu, Rupert Klein, and Michael L. Minion. A fourth-order auxiliary variable projection method for zero-mach number gas dynamics. *Journal of Computational Physics*, 227(3):2012 – 2043, 2008.

[45] D. Ketcheson, M. Parsani, and Randall J LeVeque. High-order wave propagation algorithms for hyperbolic systems. *SIAM Journal on Scientific Computing*, 35(1):A351–A377, 2013.

[46] Christiane Helzel, James A. Rossmanith, and Bertram Taetz. An unstaggered constrained transport method for the 3D ideal magnetohydrodynamic equations. *Journal of Computational Physics*, 230(10):3803 – 3829, 2011.

[47] J. Rossmanith. An unstaggered, high-resolution constrained transport method for magnetohydrodynamic flows. *SIAM Journal on Scientific Computing*, 28(5):1766–1797, 2006.

[48] A. J. Christlieb, James A. Rossmanith, and Qi Tang. Finite difference weighted essentially non-oscillatory schemes with constrained transport for ideal magnetohydrodynamics. *Journal of Computational Physics*, 268:302 – 325, 2014.

[49] Bertram Taetz. *Unstaggered constrained transport methods for ideal magnetohydrodynamic equations*. PhD thesis, Ruhr-Universität Bochum, June 2012.

[50] J.S. Hesthaven. On the analysis and construction of perfectly matched layers for the linearized Euler equations. *Journal of Computational Physics*, 142(1):129 – 147, 1998.

[51] D. Ketcheson. Highly efficient strong stability-preserving Runge-Kutta methods with low-storage implementations. *SIAM Journal on Scientific Computing*, 30(4):2113–2136, 2008.

[52] M. Parsani, David I. Ketcheson, and W. Deconinck. Optimized explicit Runge–Kutta schemes for the spectral difference method applied

to wave propagation problems. *SIAM Journal on Scientific Computing*, 35(2):A957–A986, 2013.

[53] E. Fehlberg. Klassische Runge-Kutta-Formeln fünfter und siebenter Ordnung mit Schrittweiten-Kontrolle. *Computing*, 4(2):93–106, 1969.

List of Figures

List of Tables